



LibreOffice



Base Beispiele

Elektronische Rechnungen mit Base

XRechnungen, XRechnungen mit Extension und ZUGFeRD-Rechnungen mit LibreOffice 24.8 erstellen, einlesen und eingelesene Rechnungen weiterleiten

Copyright

Dieses Dokument unterliegt dem Copyright © 2024. Die Beitragenden sind unten aufgeführt. Sie dürfen dieses Dokument unter den Bedingungen der GNU General Public License (<http://www.gnu.org/licenses/gpl.html>), Version 3 oder höher, oder der Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0/>), Version 3.0 oder höher, verändern und/oder weitergeben.

Warennamen werden ohne Gewährleistung der freien Verwendbarkeit benutzt.

Fast alle Hardware- und Softwarebezeichnungen und weitere Stichworte und sonstige Angaben, die in diesem Buch verwendet werden, sind als eingetragene Marken geschützt.

Da es nicht möglich ist, in allen Fällen zeitnah zu ermitteln, ob ein Markenschutz besteht, wird das Symbol (R) in diesem Buch nicht verwendet.

Mitwirkende/Autoren

Robert Großkopf

Rückmeldung (Feedback)

Kommentare oder Vorschläge zu diesem Dokument bzw. der beigefügten Datenbank können Sie in deutscher Sprache an die Adresse discuss@de.libreoffice.org oder direkt an den Autor robert@familiengrosskopf.de senden.

Vorsicht

Alles, was an eine Mailingliste geschickt wird, inklusive der E-Mail-Adresse und anderer persönlicher Daten, die die E-Mail enthält, wird öffentlich archiviert und kann nicht gelöscht werden. Also, schreiben Sie mit Bedacht!

Datum der Veröffentlichung und Softwareversion

Veröffentlicht am 25.11.2024. Basierend auf der Version LibreOffice 24.8. Datenbank zu dieser Beschreibung ist XRechnung_V2408_034.odt.

Inhalt

Einführung	8
Vorbereitende Schritte	9
Formulare in der Übersicht	12
Das Rechnungsformular	12
Das Lieferformular	17
Das Formular zur Änderung und Weiterleitung von Lieferungen	19
Das Einstellungs-Formular	20
Das Konto-Formular	22
Navigation durch die Formulare	22
Neueingabe von Daten	23
Übersicht über alte Rechnungen	24
Rechnung erstellen	25
Einstellungsmöglichkeiten	29
Benutzeroberfläche in den Formularen	29
Formularhintergrund	30
Felder positionieren	32
Felder entfernen	32
Ausdruck der Rechnung	32
Spalten im Rechnungsinhalt	34
Schriftformatierungen	35
Adressfeld	36
Platzhalter	37
Archivierung	37
Einstellungsmöglichkeiten direkt im Makrocode	38
Automatischer Start eines Formulars	38
Beenden von Base nach dem Schließen des letzten Formulars	38
Darstellungsgröße des Formulars	38
Vergrößerung von Tabellenfeldern	38
Update von einer Vorgängerversion	39
Erforderliche Einträge in der XRechnung	42
Tabellen	46
Rechnungserstellung im Kundenbereich	46
tbl_Kunde	48
tbl_Kunde_Lieferadresse	49
tbl_Rechnung	50
tbl_Ausgang	52
tbl_Rechnung_Zusatz	54
tbl_Rechnung_Anhang	54
tbl_rel_Rechnung	55
tbl_Rechnung_Zahlung	55
tbl_Anzahl_Code	56
tbl_Land	56
tbl_mimeCode	57
tbl_Mitarbeiter	57
Eingang von Waren über Lieferanten	58
tbl_Lieferant	59
tbl_Lieferung	60

tbl_Eingang	62
tbl_Lieferung_Zusatz	63
tbl_Lieferung_Anhang	64
tbl_Lieferung_Zahlung	64
Bearbeitung und Weiterleitung des Eingangs	65
tbl_Lieferung_Aenderung	66
tbl_Eingang_Aenderung	68
tbl_Lieferung_Zusatz_Aenderung	69
tbl_Lieferung_Anhang_Aenderung	69
tbl_Lieferung_Zahlung_Aenderung	70
Einzeltablette ohne definierte Verbindung	70
tbl_Firma	71
tbl_Steuer_Anmerkung	73
tbl_Druck	73
tbl_Filter	74
Verknüpfung von Tabellen und Löschen von Datensätzen	74
Ansichten	77
viw_Filter	77
viw_Lieferung_Aenderung	77
viw_Lieferung_DruckSpalten	79
viw_Lieferung_Eingang_Aenderung	81
viw_Lieferung_Spalten_Aenderung	83
viw_Lieferung_Steuer	84
viw_Lieferung_Steuer_Aenderung	85
viw_Lieferung_Summe	86
viw_Lieferung_Summe_Aenderung	87
viw_Rechnung	88
viw_Rechnung_Ausgang	92
viw_Rechnung_DruckSpalten	97
viw_Rechnung_Spalten	98
viw_Rechnung_Steuer	99
viw_Rechnung_Summe	100
Abfragen	103
qry_Aenderung_Listbox	103
qry_Forms	103
qry_frmAusgang	104
qry_frmEingang	105
qry_frmEingangAenderung	105
qry_frmKontoRechnung	105
qry_frmLieferungRechnung	105
qry_frmLieferungRechnung_Aenderung	106
qry_frmZusatz-Einstellungen	106
qry_frmZusatz-Kunde	107
qry_frmZusatz-Lieferant	107
qry_frmZusatz-Lieferung	107
qry_frmZusatz-Lieferung_Aenderung	107
qry_frmZusatz-Rechnung	108
qry_Konto	108

qry_Land_Listbox	109
qry_Lieferung_Zahlung	110
qry_Lieferung_Zahlung_Aenderung	111
qry_Maßeinheit_Listbox	111
qry_Rechnung_Zahlung	112
qry_Rechnungstyp_Listbox	113
qry_Steuercode_Listbox	113
Formulare	115
frm_Einstellungen	115
frm_Rechnung	118
frm_Zusatz	121
frm_Zusatz_Kunde	122
frm_Lieferadresse	122
frm_Lieferung	124
frm_Zusatz_Lieferant	126
frm_Lieferung_Aenderung	127
frm_Konto	130
Ausdruck	132
Makros	137
Anhang	137
AnhangEinlesen	137
AnhangSpeichern	137
AnhangPfad	137
Backup	138
DatabaseStart	138
DatabaseClose	138
DatabaseClose1	138
DatabaseBackup	138
BackupTarget	139
FilterStart	139
DataWrite	139
Base64Binaer	139
Base64StringToByteStream	140
GetBitsLeft	140
GetBitsRight	140
EncodedToSixBit	140
SixBitToEncoded	140
ByteStreamToBase64Stream	140
Design	141
ToolbarsHide1	141
ToolbarsHide	141
ToolbarsShow	141
ShowAttachment	141
ChangeColumnName	142
ShowColumns	142
MainNurTitel	142
TitelDruckSetzen	142
DefaultSteuercodeErstellen	143
SelectContent	143
Druck	143
FillTableCarryOver	143

SQLDateToDate	144
SQLDateToDateYY	144
CellSetNumberFormat	144
FillTextfields	144
ExportPDF	144
ExportImageIntern	145
ImportImage	145
EingangBearbeiten	145
Import_to_Aenderung	145
ListboxSave	145
Filter	145
FormParallelReload	145
ListboxFilter	146
Listenfeldfilter_allround	146
Listenfeldfilter	146
ListenfeldfilterRechnungsreferenz	146
FormsReload	146
Rechnungsfilter	147
Formularwechsel	147
Navigation	147
FormStart	147
KundeLieferadresse	147
LieferadresseBeenden	147
KundeRechnungszusatz	148
RechnungszusatzBeenden	148
RechnungenGesperrt	148
Kundezusatz	148
KundezusatzBeenden	148
Lieferanzusatz	149
LieferanzusatzBeenden	149
Einstellungenzusatz	149
EinstellungenzusatzBeenden	149
Import	150
InsertSQL	150
XMLCharToChar	150
DecimalPoint	150
Import	150
Inputcontrol	151
FormVars	151
FormChange	151
FieldRequired	151
SaveRequired	151
MailValid	152
MailValid_FieldRequired	152
IBANValid	152
IBANValid_FieldRequired	152
SaveSubRow	152
BeforeSaveSubRow	153
SaveMainRow	153
MainRowSaved	153
ChangeSubToMain	153
ChangeMainToSub	153
ValueUnique	154
Kopieren	154
RechnungCopyInsert	154

Update	154
SelectSQL	154
Fields	154
Import_all	155
Import_Values	155
Wartung	155
TableRestart	155
ViewsErstellen	156
TableContentDeleteAll	156
XRechnung	156
SaveXRechnung	156
Round2Decimalplaces	157
CharToXMLChar	157
StartMail	157
ZUGFeRD	158
SaveZUGFeRD	158

Einführung

Ab Januar 2025 wird die Rechnungsstellung in elektronischer Form für Rechnungen von Firmen untereinander Pflicht. Im Kontakt mit Behörden ist eine solche Rechnungsstellung schon seit längerer Zeit Voraussetzung¹.

Eine elektronische Rechnungsstellung ist nicht die Zusendung einer Rechnung z. B. im PDF-Format oder als Bilddatei. Aus dieser Rechnung müssten ja immer noch die Daten ausgelesen und separat wieder manuell eingegeben werden. Echte E-Rechnungen müssen automatisch bearbeitbar sein. Dies erfordert das Einhalten entsprechender Formate, die zentral festgelegt sind.

Als E-Rechnung sind zwei Formate zugelassen:

1. Die XRechnung als reine XML-Datei

Sie ist direkt verarbeitbar, aber nur schlecht lesbar. Deswegen empfiehlt sich bei diesem Format eine zusätzliche Bereitstellung als PDF-Datei.

2. Die ZUGFeRD-Rechnung

ZUGFeRD steht für «Zentraler User Guide des Forums elektronische Rechnung Deutschland» und ist eine Hybridrechnungsform, bei der die XML-Datei in eine für den Empfänger leicht lesbare PDF-Datei gepackt wird. Beim Öffnen so einer Datei in einem PDF-Betrachtungsprogramm kann der enthaltene Inhalt abgespeichert werden. Eine solche PDF-Datei entspricht allerdings nicht den XRechnungs-Vorschriften. Deshalb exportiert ZUGFeRD mit Version 2.1.1 vom 1. Juli 2020 eine XML-Datei, die dann als XRechnung versandt werden kann.

Diese Dokumentation sowie die entsprechende Datenbank mit Vorlagendatei behandelt sowohl die XRechnung und die dafür zugelassene Erweiterung XRechnung-Extension als auch die Erstellung von ZUGFeRD-Rechnungen. Es geht um die Erstellung einer Rechnung (Kundeneingabe, Rechnungseingabe), die Abspeicherung der Rechnung und den Versand der PDF- und XML-Rechnung. Die erstellten ZUGFeRD-Rechnungen entsprechen im XML-Teil den Vorgaben der XRechnung nach dem UN/CEFACT Cross Industry Invoice – Standard (CII).

Die Datenbank erstellt XRechnungen mit allen Pflichtfeldern der XRechnung. Hinzu kommen je nach Wunsch der Nutzer weitere Felder, die in der Spezifikation zur XRechnung beschrieben sind. Je aktueller die Datenbankdatei ist, desto mehr Felder werden also unterstützt.

Das Einlesen einer XRechnung ist in zwei zugelassenen Formaten möglich, und zwar der UBL-Version und der oben bereits genannten CII-Version.

Die XRechnung besteht bereits aus einer XML-Datei. Bei ZUGFeRD-Rechnungen muss die enthaltene XML-Datei separat abgespeichert werden.

Zum Einlesen werden nur die Felder ausgewertet, die in der vorliegenden Datenbank auch abgebildet werden können. Dies sind auf jeden Fall alle Pflichtfelder der XRechnung und auch Anhänge, die in der XRechnung stecken.

Eingelesene Rechnungen können auch mit Korrekturen versehen und weiter geleitet werden.

Hinweis

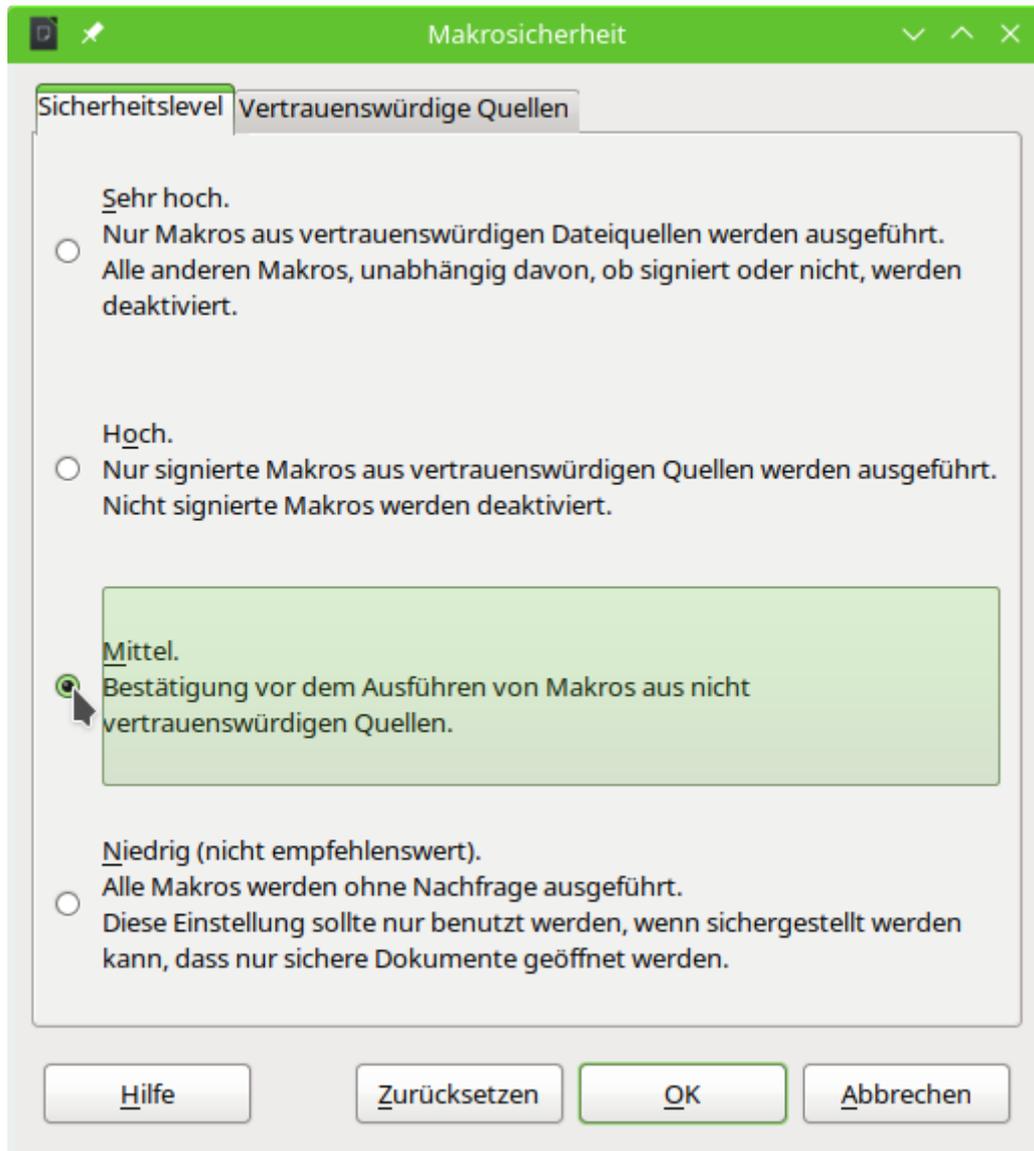
Die vorliegende Datenbankdatei arbeitet mit der internen FIREBIRD-Datenbank von LibreOffice zusammen. Manche Versionen von Linux-Distributionen sind so gepackt, dass diese interne Datenbank nicht unterstützt wird, da die Erstellung der Datenbank zur Zeit noch ein experimentelles Feature ist. In dem Falle hilft es dann nur, die Originalpakete von der Downloadseite von LibreOffice selbst zu installieren: <https://de.libreoffice.org/download/download/>

¹ Siehe auch: <https://www.e-rechnung-bund.de/e-rechnung/>

Vorbereitende Schritte

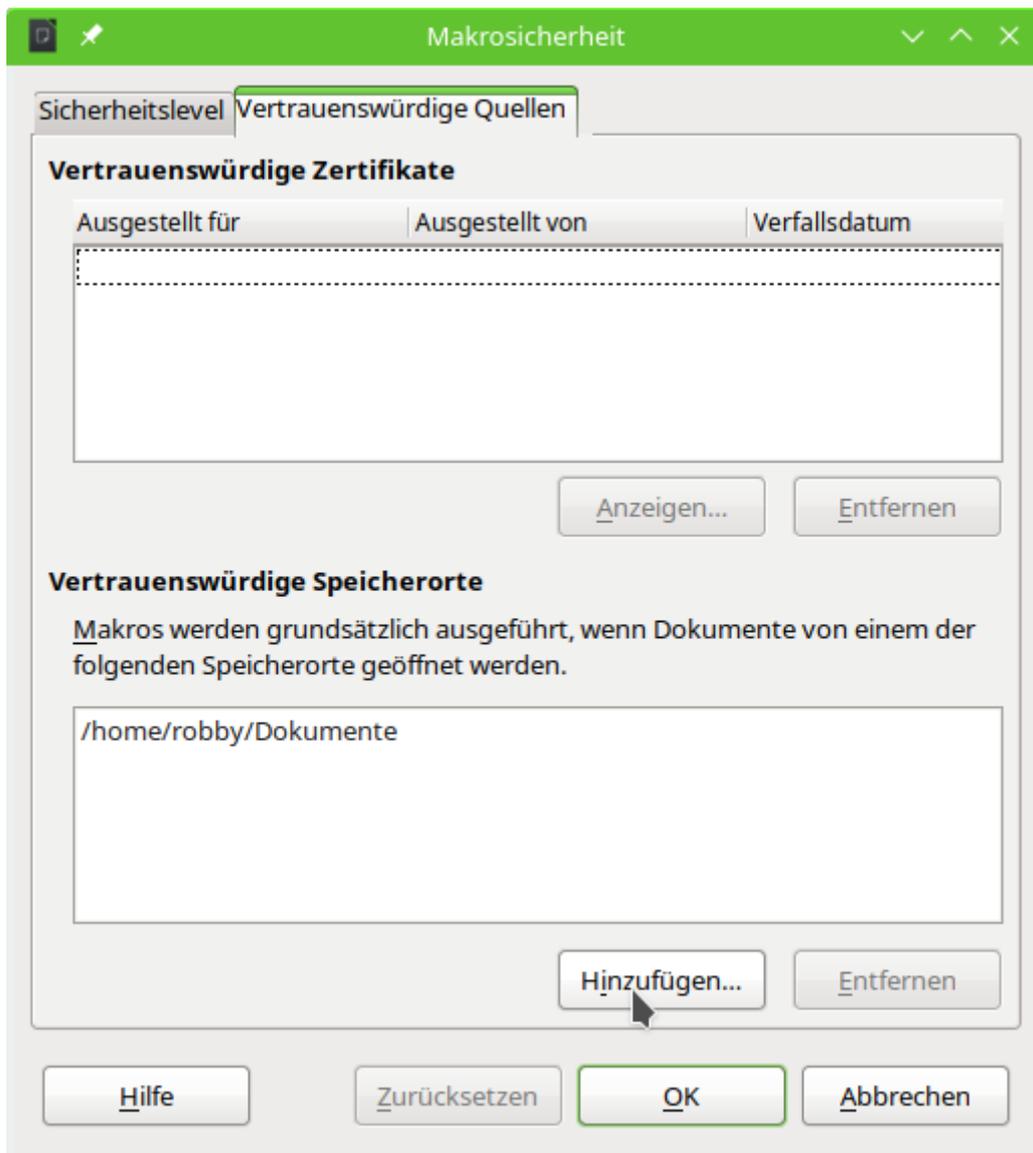
Die Datei «XRechnung_V2408_034.odb²» sowie die Datei «Vorlage_Rechnung_Extension.ott» werden in ein Verzeichnis kopiert. In diesem Verzeichnis wird beim Erstellen der Rechnung ein Unterverzeichnis **Archiv** mit entsprechenden Jahres- und Monatsverzeichnissen automatisch erstellt. Werden Rechnungen eingelesen und enthalten diese Anhänge, so werden sie in einem Unterverzeichnis **Archiv → Lieferungen**, ebenfalls mit entsprechenden Jahres- und Monatsverzeichnissen, gespeichert und mit der Lieferung verknüpft.

Sämtliche Formulare und auch die Vorlagendatei sind mit der Schriftart «DejaVu Sans» erstellt worden. Sollte diese frei verfügbare Schriftart nicht vorhanden sein, so muss gegebenenfalls die Vorlagendatei angepasst werden.



In LibreOffice muss über **Extras → Optionen → Sicherheit → Makrosicherheit → Sicherheitslevel** die Makrosicherheit auf 'Mittel' eingestellt werden. Nur dann ist es möglich, dass die eingebauten Makros ausgeführt werden. Ohne Makros kann kein XRechnungs-Export stattfinden.

² Die ODB-Datei nutzt die in LibreOffice eingebaute FIREBIRD-Datenbank. Die Benennung der Datei enthält neben der Versionsnummer von LO auch eine weitere Versionsnummer des Updates.



Soll zum Start der ODB-Datei die Sicherheitsnachfrage zu den Makros nicht mehr auftauchen, so kann unter Makrosicherheit im Reiter **Vertrauenswürdige Quellen** → **Vertrauenswürdige Speicherorte** der Speicherort der ODB-Datei herausgesucht werden. Die Ausführung von Makros ist dann in diesem Verzeichnis und allen Unterverzeichnissen erlaubt.

Wird jetzt die ODB-Datei geöffnet, so startet sie mit der Ansicht des Kundenformulars für eine erweiterte XRechnung. Für den ersten Start ist es aber wichtig, zuerst einmal die Grundeingaben für die eigene Firma zu erledigen. Zum Einstellungsformular kann über das Listenfeld ganz oben im Formular gewechselt werden.

Die Felder, die im Einstellungsformular mit einem «*» gekennzeichnet sind, sind Pflichtfelder. Das Formular lässt sich nicht abspeichern, wenn kein Eintrag in diese Felder erfolgt. Für eine korrekte XRechnung muss neben den Einstellungen für die Firma auch mindestens eine Person als MitarbeiterIn in dem entsprechenden Tabellenkontrollfeld angegeben werden. Im Tabellenkontrollfeld erfolgt die Abspeicherung durch einen Wechsel von der aktuellen Tabellenzeile zur nächsten Tabellenzeile.

Damit sind alle Voraussetzungen geschaffen, um im Kundenformular die Eingabe einer Rechnung zu testen.

✓ Hinweis

Datenbanken verhalten sich bei der Verwaltung der Daten anders als Tabellenkalkulationen. Durch Verknüpfung von Tabellen müssen Datenbanken die Integrität der Daten sicher stellen. Ein Löschen von Daten ist also nicht so einfach möglich, damit nicht nachher Rechnungen existieren, zu denen z. B. der Kunde gelöscht worden ist.

Die Base-Datei enthält bereits Testdaten und sollte auch ruhig erst weiter getestet werden. Soll die vorliegende Datenbank für den produktiven Gebrauch geleert werden, so müssten zuerst die Datensätze in "tbl_Rechnung" und "tbl_Lieferung" gelöscht werden, bevor die Datensätze in "tbl_Kunde" und "tbl_Lieferant" sicher gelöscht werden können. Vor dem Löschen von Daten in den Tabellen müssen alle offenen Formulare geschlossen werden.

Um das Löschen zu vereinfachen, kann über **Extras → Makros → Makro ausführen** in der Base-Datei das Modul **Wartung → TableContentDeleteAll** ausgeführt werden. Das Modul löscht alle Daten bis auf die Code-Daten für XRechnungen und setzt die Tabellen in den Startzustand zurück.

Siehe hierzu auch: [Verknüpfung von Tabellen und Löschen von Datensätzen](#).

Sollen **ZUGFeRD**-Rechnungen erstellt werden, so ist in LibreOffice 24.8 noch zusätzlich die Installation von **PDFtk** (<https://www.pdfabs.com/tools/pdftk-the-pdf-toolkit/>) notwendig. Linux-Distributionen bieten dieses Tool häufig direkt innerhalb der Repositories zur Installation an. Der PDF-Export von LibreOffice erlaubt zur Zeit noch nicht, dass eine XML-Datei an eine PDF-Datei angehängt wird. Diese Funktionalität stellt das kostenlose PDFtk sowohl für Windows als auch für Linux zur Verfügung. Dass die PDF-Datei nach der Ausführung von PDFtk über ein BASIC-Makro einen Anhang hat, lässt sich nur im PDF-Betrachtungsprogramm erkennen.³

! Vorsicht

Die Datenbankdatei arbeite mit einer internen Datenbank. Die komplette Datei wird beim Start in das Backup-Verzeichnis von LibreOffice gesichert.

Interne Datenbanken sichern leider die eingegebenen Inhalte erst in der Datenbankdatei, wenn die Datenbankdatei geschlossen wird. Kommt es dabei zu Problemen (Absturz des Rechners o.ä.), dann sind alle neu eingegebenen Daten nach dem letzten Öffnen der Datenbankdatei weg.

Für den Gebrauch im professionellen Umfeld ist daher besser zu empfehlen, aus der internen Datenbankdatei eine Datenbankdatei mit Kontakt zu einer externen Firebird-Datei zu machen. Das habe ich im Base-Handbuch beschrieben.⁴

Für den Mehrbenutzerbetrieb wäre dann ein Firebird-Server zu empfehlen. Der ist, zumindest unter Linux, recht problemlos zu installieren. Dann können verschiedene Nutzer über die Datenbankdatei von LibreOffice gleichzeitig auf die Daten zugreifen, neue Rechnungen erstellen, eingehende Rechnungen einlesen oder in den Rechnungen suchen.

³ Siehe dazu auch https://bugs.documentfoundation.org/show_bug.cgi?id=160475 Sollte das Packen von Dateien in eine PDF-Datei über LibreOffice möglich werden, so wird zeitnah ein Update dieser Datenbanklösung für die elektronische Rechnungserstellung erfolgen.

⁴ Das Handbuch für Base ist unter <https://de.libreoffice.org/get-help/documentation/> zu finden. Dort sind auch Schritte für den Aufbau eines Firebird Servers erklärt.

Formulare in der Übersicht

Die Datenbank enthält zur Zeit zwölf Formulare, wobei sechs der Formulare lediglich für die unterschiedlichen Rechnungserstellungen gedacht sind. Zur eigenen Rechnungserstellung wird das Formular "XRechnung" in einer der drei Varianten benötigt. Dort werden Kundendaten erstellt oder aus den bestehenden Daten herausgesucht. Den Kunden können dann Rechnungen zugewiesen werden. Die Rechnungen können von dem Formular aus direkt gespeichert und als PDF-Datei sowie als XRechnung-Datei an eine E-Mail angehängt werden.

Über den Button  in jedem der XRechnungs-Formulare ist ein Formular greifbar, das auch auf die bereits erstellten und komplett gezahlten Rechnungen zugreift. Diese Formulare erlauben dann keine Änderung der Daten mehr, sondern sind nur zur Recherche und zur Kopie von Rechnungsdaten gedacht.

Das Rechnungsformular

XRechnung erweitert

Filter: Kunde Dreaming & Co. → E-Mail: hierhin@beispiel.de

ID: 1

Firma*: Dreaming & Co. AnsprechP: Lisa Ichwillwissen AnsprechPGeschl.: weiblich Kürzel:

StraßeNr: Hamburger Damm 12a Plz*: 90919 Ort*: Irgendwo Land*: Deutschland

E-Mail*: hierhin@beispiel.de Telefon: E-Mail (CC): UST-IdNr:

Datensatz 1 von 1

Spaltenüberschriften für Waren in der Rechnung:
 1. Spalte
 2. Spalte
 3. Spalte
 nur Titel in «Rechnungsposten» eintragen

Rechnungserstellung

ID: 3 Rechnungsnummer: RE-2024-0001 4711 RechnungsNr-Zusatz: 4711 Rechnungstyp*: Handelsrechnung Lieferdatum:

Datum*: 25.03.24 KundeZeichen*: Mitarbeiter der Firma*: Peter Tüftel abw. Lieferadr.:

Zahl Not.1: Zahl Not.2: Anmerkung:

Re.Nr. vorgeben als:

Datensatz 2 von 4

Rechnung erstellen ZeilenNr Titel

Ermäßigungen und Aufschläge auf Rechnungsebene

Zusatz*	Betrag*	Steuersatz*	Steuercode*
		19 % Normalsatz	

Netto	Steuer	Brutto	bereits gezahlt
620,00 €	117,80 €	737,80 €	

Datensatz 1 von 1

Rechnungsposten

ZeilenNr*	Anzahl*	Einheit*	Ware*	Preis*	Steuersatz*	Steuercode*	StartDatum	EndDatum
A	1	Transport	Transporte NRW	240,00 €	19 % Normalsatz		01.03.24	31.03.24
B	1	Transport	Transporte Niedersachsen	380,00 €	19 % Normalsatz		01.03.24	31.03.24

Datensatz 1 von 2

Untergliederung Rechnungsposten

ZeilenNr*	Anzahl*	Einheit*	Ware*	Preis*	Steuersatz*	Steuercode*	StartDatum	EndDatum
A1	1	Transport	Rheine → Münster	60,00 €	19 % Normalsatz		01.03.24	
A2	2	Transport	Rheine → Steinfurt	45,00 €	19 % Normalsatz		05.03.24	
A3	1	Transport	Rheine → Ahaus	90,00 €	19 % Normalsatz		07.03.24	

Datensatz 1 von 3

Startformular "XRechnung erweitert": hier werden neue Kunden eingegeben und Rechnungen für Kunden erstellt.

Über ein Listenfeld wird oben links von einem Formular zum anderen gewechselt. Dieses Formular liegt der Datenbankdatei in drei Varianten bei. Oben ist die Variante «XRechnung erweitert» dargestellt. Sie zeigt im unteren Bereich 2 nahezu identische Tabellenkontrollfelder für die XRechnung-Extension an. Die Variante «XRechnung» sowie die Variante «ZUGFeRD» verfügen nicht über das untere Tabellenkontrollfeld für die «Untergliederung Rechnungsposten» und haben deshalb auch nicht die Möglichkeit, «nur Titel in Rechnungsposten eintragen».

Das neben dem Formularauswahlfeld liegende Listenfeld im Kasten **Filter** dient dazu, bestimmte Kunden aus der Liste auszuwählen, die anschließend im darunterliegenden Formular angezeigt werden. Dann ist die Nutzung der Navigationsfunktion in der darunterliegenden Navigationsleiste nicht mehr notwendig.

Im Bereich bis zur ersten Navigationsleiste werden die Stammdaten für einen Kunden verwaltet. Neben der kompletten Adresse (mit Ausnahme des Straßennamens) ist hier auch der Eintrag der E-Mail notwendig, da sonst die XRechnung nicht verschickt werden kann. Über die Navigationsleiste kann zu anderen Datensätzen navigiert werden. Das Feld «Kürzel» dient dazu, die generierten Rechnungen ggf. mit einem Firmenkürzel zu beginnen, um die Rechnungen direkt einer entsprechenden Firma zuzuordnen zu können. Hier sind maximal 10 Buchstaben zulässig. Von Umlauten sollte dabei abgesehen werden. Für Kunden kann auch in «UST-IdNr» die Umsatzsteuernummer des Kunden, über «Reg-ID» ggf. so etwas wie die nationale Identifikationsnummer (persönliche Steuernummer) sowie ein Freitext «Druck Zus.» für den Ausdruck formuliert werden. Dieser Freitext wird nur für die Erstellung der PDF-Datei genutzt und nicht in die XML-Datei der XRechnung übertragen.

Das Formular für den Kunden ist bereits reichlich gefüllt. Über den Button im Kundenbereich wird ein zusätzliches Formular gestartet, das weitere Einträge im Kundenbereich ermöglicht. Hier geht es besonders um die Erhebung von Lastschriften.

Der Bereich «Spaltenüberschriften für Waren in der Rechnung» wird weiter unten erklärt.

Im Bereich über der zweiten Navigationsleiste befinden sich die Grunddaten für die Rechnung. Das Feld «ID» kann nicht bearbeitet werden (gelber Hintergrund). Drei Felder sind Pflichtfelder (durch '*' gekennzeichnet) für die XRechnung: Das Rechnungsdatum, ein Zeichen des Kunden und die Benennung eines Mitarbeiters/einer Mitarbeiterin, mit der der Kunde Kontakt aufnehmen kann. Das Zeichen des Kunden ist hier standardmäßig mit '-' gesetzt. Hier könnte so etwas wie eine Bestellnummer stehen. Bei Behörden ist dies die **Leitweg-ID**. In dem genannten Bereich sind auch noch ein Feld für einen «Rechnungsnummer-Zusatz» (Zusatzinformationen wie z. B. bisherige eigene Rechnungsnummer, die an die generierte Nummer angehängt wird) und zwei Felder für besondere Zahlungsnotizen enthalten. Wird «Zahl Not. 1» ausgefüllt, so wird damit die standardmäßige Zahlungsnotiz aus dem Formulare «Einstellungen» für die aktuelle Rechnung überschrieben.

Hinweis

Bei der Eingabe der Zahlungsnotiz kann mit der Kombination **?Datum?** das zu der Rechnung passende Datum eingesetzt werden, bis zu dem die Zahlung zu erfolgen hat.

Zusätzlich ist hier noch die Möglichkeit eingebaut, ein Lieferdatum und eine gegebenenfalls abweichende Lieferadresse zu vermerken. Neue Lieferadressen können über den Button direkt neben dem Listenfeld eingegeben werden.

Allgemeine Anmerkungen, die Rechnung betreffend, können als Freitext eingegeben werden. Der Freitext wird auch in die XRechnung übertragen.

abw. Lieferadresse **Filter**
Kunde: Dreaming & Co. → E-Mail: hierhin@beispie

ID:

Firma*:

StraßeNr: Plz*: Ort*: Land*:

Datensatz 1 von 2 Eingabe beenden

Über die Schaltfläche  lassen sich abweichende Lieferadressen eingeben. Bereits vorher eingegebene abweichende Lieferadressen für einen Kunden können in dem daneben befindlichen Listenfeld ausgewählt werden.

zus. Einträge auf Rechnungsebene

ID: Rechnungsnummer:

Zustellzeitraum

StartDatum:

EndDatum:

Referenzen

ProjektNr:

VertragsNr:

BestellNr:

AuftragsNr:

AusschreibungsNr:

ObjektNr:

Kontoreferenz Kunde:

Eingabe beenden

abweichender Dateiname:

Abschlagszahlungen

Rechnungen, auf die Bezug genommen werden soll

<input type="text"/>	

Datensatz 1 von 1 Eingabe beenden

Über den Button  können zusätzliche Einträge auf Rechnungsebene eingegeben werden. Mit der Entwicklung der Datenbank kamen immer mehr Wünsche auf, so dass der Platz für diese Wünsche nicht mehr ausreichte. Die in diesen zusätzlichen Einträgen gemachten Angaben sind nicht erforderlich. Sie werden aber sowohl im Rechnungsdruck als auch bei der Erstellung der XRechnung selbst entsprechend aufgenommen.

Hier wird auch die Zuordnung einer Rechnung als Endabrechnung möglich. Alle Rechnungen, die bei der Erstellung der Rechnung berücksichtigt werden sollen, werden hier ausgewählt und eventuell erfolgte Zahlungen von dem Betrag der Gesamtrechnung abgezogen.

In den drei Tabellen mit weißem Hintergrund können die Rechnungsdaten eingegeben werden. Dies sind Eingaben auf Rechnungsebene (wie in dem Beispiel die Portokosten) und Eingaben der Rechnungsposten. Eine Eingabe von Ermäßigungen und Aufschlägen ist nicht erforderlich. Die Aufführung von Rechnungsposten ist bei der XRechnung hingegen Pflicht.

Die Untergliederung der Rechnungsposten ist Teil der XRechnung-Extension. Mit Hilfe solcher Untergliederung kann beispielsweise in einer Rechnung als Hauptrechnungsposten erscheinen: 'Demontage der Heizungsanlage'. In der Untergliederung wird dann unterschieden in 'Demontage Brenner', 'Demontage Wasserbehälter' usw. Die Untergliederung kann genutzt werden. Wird sie genutzt, so sind auch dort entsprechende Pflichtfelder wie in den darüber liegenden Rechnungsposten.

Bis auf die Felder «Bestellnummer», «Rabatt», «StartDatum» und «EndDatum» in den unteren beiden Tabellenfeldern sind alle Felder Pflichtfelder. Sobald also ein neuer Datensatz begonnen wird, muss auch der gesamte andere Inhalt ausgefüllt werden. Geschieht dies nicht, so kommt beim Versuch des Abspeicherns eine Fehlermeldung. Einige der Felder sind allerdings schon durch das Formular mit Eingabevorschlägen versehen, so dass die Eingabe etwas erleichtert wird.

Das gelb hinterlegte Tabellenfeld wird während der Eingabe von Elementen auf Rechnungsebene und einzelnen Rechnungsposten nach jedem abgespeicherten Datensatz aktualisiert, so dass der Gesamtpreis sichtbar wird.

Anhänge sind nur in bestimmten binären Formaten erlaubt. Sie werden in die XML-Datei eingebaut und von dort auch wieder ausgelesen. Für die Anhänge muss zuerst ein «mimeType» ausgesucht werden.

Über einen Doppelklick auf das grafische Kontrollfeld wird ein neuer Anhang aus dem eigenen Dateisystem ausgesucht. Der Anhang wird anschließend in dem grafischen Kontrollfeld dargestellt.

Manche Anhänge sind für das grafische Kontrollfeld klein genug, so dass sie deutlich erkannt werden können. Bei kompletten Dokumenten wie z. B. einer PDF-Datei zeigt das Kontrollfeld nur die erste Seite in der kompletten Höhe an. Bei Tabellendokumenten wird gar kein Inhalt angezeigt. Mit **Anhang anzeigen** wird die Datei an das im Betriebssystem dafür als Standardprogramm vorgesehene Programm geschickt und damit geöffnet.

Der Anhang wird über die darunter liegende Navigationsleiste mit dem Speichersymbol gespeichert. Dabei wird der Dateiname des Anhangs automatisch in das gelbe Feld «Dateiname» übertragen und mit abgespeichert.

Ist die Rechnung fertig, so lässt sie sich über **Rechnung erstellen** archivieren und verschicken. Dabei wird eine Writer-Datei, eine PDF-Datei und eine XML-Datei (für die eigentliche XRechnung) erstellt und in einem Archiv abgespeichert. Zusätzlich wird das Mailprogramm gestartet und die PDF-Datei sowie die XML-Datei an eine vorgefertigte Mail angehängt. Hier muss nur noch auf **Senden** gedrückt werden.

Für das Erstellen der Writer-Rechnung sind zusätzliche Einstellungen möglich.

- «ZeilenNr»: nach den Rechnungszeilen kann durchnummeriert werden. Diese Zeilennummerierung folgt der Untergliederung wie bei der Nummerierung von Kapiteln. So kann der Haupteintrag z. B. die Nummer '3' haben, der erste Untereintrag die Nummer '3.1' usw.
- «Titel»: Einträge, die folgende Untereinträge haben, können so nur mit dem Titel ohne die Benennung von Anzahl, Preis usw. erscheinen. Die Benennung vor allem des Preises kann sonst auch leicht zur Verwirrung führen, da der Haupteintrag den Gesamtpreis für die Untereinträge bereits enthält.

✓ Hinweis

Das Formular ist mit allen Feldern ausgestattet, die abgespeichert und zur weiteren Verarbeitung genutzt werden können. Es lässt sich also nicht einfach ein zusätzliches Feld einbauen, das dann automatisch in die Rechnung übertragen würde.

Bei manchen Anwendungsfällen mag es aber sein, dass bestimmte Einträge nie benutzt werden. Verzichtet eine Firma komplett auf Ermäßigungen und Aufschläge auf Rechnungsebene, so kann das Tabellenkontrollfeld dafür in der Formularbearbeitung entfernt werden. Auch der gesamte Bereich zu den in die XRechnung eingebauten «Anhang» kann entfernt werden. Die XRechnung bleibt damit weiter gültig.

Innerhalb der einzelnen Formularbereiche können gefahrlos Einzelfelder entfernt werden, die keine Pflichtfelder sind.

Pflichtfelder können nur mit Hilfe von SQL als automatisch vorbelegte Felder definiert werden. Hier ist also Vorsicht geboten.

Siehe hierzu auch im Kapitel «Einstellungsmöglichkeiten» [Benutzeroberfläche in den Formularen](#)

Rechnungsposten

Anzahl*	Ware*	Preis*
2	Open-Source DVD (LibreOffice, Gimp, Scr	12,45 €
3	Handbuch Base 7.6, gedruckt (alles über	21,90 €
1	Stunden für die Arbeit an XRechnung	1.125,00 €
4	Tafeln Schokolade mit Mandeln	1,75 €

Datensatz 3 von 16

Untergliederung Rechnungsposten

Anzahl*	Ware*	Preis*
10	Arbeit für Untergliederung	25,00 €
1	Arbeit am Gesamtkonzept	875,00 €

Datensatz 1 von 2

Einheit*
Bestellnummer
Warenzusatz
Rabatt
Steuersatz*
Steuercode*
StartDatum
EndDatum
Spalten einblenden >

Bei den Rechnungsposten und der Untergliederung der Rechnungsposten können über das Einstellungsformular viele Felder von der Anzeige ausgeschlossen werden. Hier zeigen beide Tabellenkontrollfelder nur die verbleibenden drei Felder an, ohne die eine Rechnung nicht erstellt werden kann.

Werden vorübergehend weitere Felder benötigt, so sind die Felder über die rechte Maustaste (über dem Titelbereich der Tabelle) in einem Kontextmenü erreichbar. Diese Einstellung wird aber beim Neustart des Formulars wieder rückgängig gemacht.

Spaltenüberschriften für Waren in der Rechnung

1. Spalte

2. Spalte

3. Spalte

nur Titel in «Rechnungsposten» eintragen

Für jeden Kunden können insgesamt drei Spalten («Ware*», «Bestellnummer» und «Warenzusatz») mit einer eigenen Überschrift für die Erstellung der Rechnung versehen werden. Die Überschrift erscheint dann auch für den entsprechenden Kunden bei den Tabellenkontrollfeldern. Darüber hinaus ist es möglich, in der Tabelle «Rechnungsposten» nur ein Feld zur Titelein-gabe zu aktivieren. Dieser Titel wird dann in der gedruckten Rechnung als Titel ausgegeben.

Rechnungsposten			
	Titel*		
	Open-Source DVD (LibreOffice, Gimp, Scr		
	Handbuch Base 7.6, gedruckt (alles über		
▶	Stunden für die Arbeit an XRechnung		
	Tafeln Schokolade mit Mandeln		
Datensatz 3 von 16			
Untergliederung Rechnungsposten			
	Anzahl*	Leistung*	Preis*
▶	10	Arbeit für Untergliederung	25,00 €
	1	Arbeit am Gesamtkonzept	875,00 €
+			
Datensatz 1 von 2			

Die vorhergehende Einstellung zeigt jetzt im Tabellenkontrollfeld «Rechnungsposten» nur noch ein Feld an, das zudem nicht mehr «Ware*» sondern «Titel*» heißt. Es erscheint also in der Rechnung ein Titel, dessen Untergliederung die Anzahl der Stunden für verschiedene Bereiche aufzeigt.

Im Tabellenkontrollfeld «Untergliederung Rechnungsposten» ist das Feld «Ware*» jetzt mit der Bezeichnung «Leistung*» versehen worden. Diese Bezeichnung erscheint auch in der ausgedruckten Rechnung.

Die XRechnung selbst wird unabhängig von diesen Einstellungen erstellt. Dort gelten völlig andere Bezeichnungen als im Formular.

Das Lieferformular

Lieferung

Filter: Lieferant Base XRechnung Extended → E-Mail: info@...

ID: 8

Firma*: Base XRechnung Extended

Kontaktperson: Name* Peter Tüftel, Geschlecht, E-Mail* tueftel@beispiel.de, Telefon* 012457910

StraßenNr: Open Source Allee 42, Plz*: 05040, Ort*: Überall, Land*: Deutschland

E-Mail*: info@beispiel.de, Telefon: 012457910, IBAN*: nicht angegeben, Zahlungsempfänger

Datensatz 1 von 1

Lieferungsaufnahme

ID: 16, Rechnungstyp*: Handelsrechnung, Lieferdatum, abw. Lieferadr., Anhang* mimeCode*, Anhang* Anhang anzeigen

Datum*: 25.03.24, LieferantZeichen*: R-2024-0001, UnserZeichen, Zahlungsfrist: 08.04.24

Zahlungsnotiz: Zahlbar bis zum 08.04.2024, Anmerkung

Summen lt. Rechnung

Netto	Steuer	Brutto	
620,00 €	117,80 €	737,80 €	
bereits gezahlt	noch offen		
0,00 €	737,80 €		

Datensatz 2 von 2

Zusatz*	Betrag*	Steuersatz*
		19 %

Datensatz 1 von 1

Summen berechnet

Netto	Steuer	Brutto	bereits gezahlt
620,00 €	117,80 €	737,80 €	0,00 €

Datensatz 1 von 1

Rechnungsposten

XRech-ZeilenID	Anzahl*	Einheit	Ware*	Preis*	Rabatt	Steuersatz*	Steuercode*	StartDatum	EndDatum
1	1	Transport	Transporte NRW	240,00 €	0,00 €	19 % Normalsatz		19.08.24	19.08.24
2	1	Transport	Transporte Niedersachsen	380,00 €	0,00 €	19 % Normalsatz			

Datensatz 1 von 2

Untergliederung der Rechnungsposten

XRech-ZeilenID	Anzahl*	Einheit	Ware*	Preis*	Rabatt	Steuersatz*	Steuercode*	StartDatum	EndDatum
1.1	1	Transport	Rheine → Münster	60,00 €	0,00 €	19 % Normalsatz		01.03.24	01.03.24
1.2	2	Transport	Rheine → Steinfurt	45,00 €	0,00 €	19 % Normalsatz		05.03.24	05.03.24
1.3	1	Transport	Rheine → Ahaus	90,00 €	0,00 €	19 % Normalsatz		07.03.24	07.03.24

Datensatz 1 von 3

Das Lieferformular bietet neben den Feldern aus dem Rechnungsformular u.a. eine Möglichkeit zum Einlesen von XRechnung-Dateien

Der Aufbau des Lieferformulars entspricht dem Prinzip des Rechnungsformulars. Dabei sind zwangsläufig in einigen Formularbereichen mehr Felder eingebaut. So erscheint bei der Rechnung eine Zahlungsfrist (sofern vom Lieferanten angegeben) und die vom Lieferanten angegebenen Summen für den Nettobetrag, den Steuerbetrag und den Bruttobetrag. Unter den Angaben für die Summen des Lieferanten wird noch einmal berechnet, ob die Angaben den aufgeführten Rechnungsposten entsprechen. Hat der Lieferant, wie es erlaubt ist, keine prozentuale Steuerangabe für die einzelnen Rechnungsposten hinterlegt, so stimmt allerdings in der berechneten Summe aus der Datenbank nur der Nettoanteil. Dies kann bereinigt werden, indem den einzelnen Lieferposten auch ein Steuersatz nachträglich hinzugefügt wird.

Wichtigstes Element für die schnelle Verarbeitung in diesem Formular ist die Importfunktion. Über **Durchsuchen...** wird das Dateisystem nach der erhaltenen XML-Datei durchsucht. Über **Einlesen** erfolgt dann das Einlesen der gesamten Daten aus der Rechnung. Ist der Lieferant noch nicht vorhanden, so werden die Daten aus der Rechnung bezogen. Wurde die Rechnung des Lieferanten mit der entsprechenden Nummer noch nicht eingelesen, so wird die gesamte Rechnung eingelesen.

✓ Hinweis

Das Auslesen der Anhänge funktioniert zur Zeit noch recht langsam. Hier muss die Praxis zeigen, ob bei vielen Anhängen eventuell andere Wege notwendig sind.

Das Auslesen der Rechnung setzt eine Rechnung voraus, die konform der XRechnung ist. Siehe dazu auch das entsprechende Kapitel [Erforderliche Einträge in der XRechnung](#). Fehlen erforderliche Einträge, so werden solche Rechnungen im Validator mit Warnungen versehen. Rechnungen mit fehlenden Pflichteinträgen werden in diese Datenbank nicht aufgenommen.

✓ Hinweis

Grundsätzlich ist die Datenbank so angelegt, dass alle Informationen, die als XRechnung ausgegeben werden können, auch als Lieferung eingelesen werden können. Deshalb lässt sich auch in einem zusätzlichen Formular eine eingelesene Lieferung in entsprechende Tabellen kopieren, die eine Bearbeitung und Weiterleitung als Rechnung ermöglichen.

zus. Einträge zum Lieferanten

ID	8
UST-IdNr	DE42424242
ZahlungTage	
RegisterID	
Gläubigernummer	DE98ZZZ09999999999
UST-Ausnahme	
SteuerNr	

Eingabe beenden

Auch bei Lieferformular gibt es zusätzliche Einträge zum Lieferanten, die nicht mehr im eigentlichen Formular darstellbar sind.

Das Formular zur Änderung und Weiterleitung von Lieferungen

Lieferung Änderung

Filter: Lieferant Testverkäufer → E-Mail: test@test.de

Lieferung: [Auswahl] Lieferung übertragen

ID: 4	Firma*: Testverkäufer	E-Mail*: test@test.de	Kontaktperson: Name* Test
StraßeNr: 12345	Telefon: 123456789	IBAN*: DE21081508151234123412	Geschl.:
Ort*: Testhausen	Zahlungsempf.: Testkäufer	E-Mail*: test@test.de	E-Mail*: test@test.de
Land*: Deutschland		Telefon*: 123456789	

Datensatz 1 von 1

Lieferungsänderung

ID: 6	ID (alt): 6	Mitarbeiter:	Lieferdatum:	abw. Lieferadr.:
Datum*: 23.04.21	LieferantZeichen*: 112233	UnserZeichen: 991-33333TEST	Zahlungsfrist: 28.04.21	Änderung:
Zahlungsnotiz:	Anmerkung:	Rechnungstyp*: Handelsrechnung		

Datensatz 1 von 1

Zusatz*	Betrag*	Steuersatz*	Änderung
		19 %	

Datensatz 1 von 1

Netto	Steuer	Brutto	bereits gezahlt
1.500,00 €	304,00 €	1.804,00 €	
	bereits gezahlt	0,00 €	1.804,00 €
1.500,00 €	304,00 €	1.804,00 €	1.804,00 €

Rechnungsposten

XRech-ZeilenID	Anzahl*	Einheit	Ware*	Preis*	Rabatt	Steuersatz*	Steuercode*	StartDatum	EndDatum	Änderung
1	1	ein	Neumotor	1.000,00 €		19 %	Normalsatz			
2	1	ein	Bemessungsgrundlage und Umsatzsteuer auf Altteil	100,00 €		19 %	Normalsatz			
3	-1	ein	Korrektur/Stornierung Bemessungsgrundlage der	100,00 €		0 %	Null bewertet			

Datensatz 1 von 4

Untergliederung der Rechnungsposten

XRech-ZeilenID	Anzahl*	Einheit	Ware*	Preis*	Rabatt	Steuersatz*	Steuercode*	StartDatum	EndDatum	Änderung
						19 %				

Datensatz 1 von 1

Manche Berufe dienen als Schnittstelle zwischen Handwerksbetrieben und Kunden. Bauingenieure überprüfen Rechnungen und leiten anschließend die überprüften Rechnungen an den Kunden weiter. Mit diesem Formular können sie aus eingegangenen Lieferungen bearbeitbare Rechnungen machen. Dabei werden die ursprünglichen Lieferungen nicht geändert. Stattdes-

sen wird mit einer Kopie gearbeitet, die anschließend an den Kunden weiter geleitet werden kann.

In der obersten Zeile neben dem Filter wird eine Lieferung ausgewählt, die dann mit dem Button **Lieferung übertragen** zu einer bearbeitbaren Lieferung kopiert wird. Die Daten für den Kunden können teilweise direkt in diesem Formular bearbeitet werden. Die Gesamtdaten zu dem Kunden sind in den Rechnungsformularen bearbeitbar.

Bei der eingelesenen Lieferung wird anschließend auch angezeigt, auf welche ursprüngliche Lieferung sich die Änderung bezieht. Die entsprechende interne Tabellenummer befindet sich im Feld «ID (alt)» in der Lieferungsänderung.

Das Einstellungs-Formular

Einstellungen

Firma* Base XRechnung Extended

StraßeNr* Open Source Allee 42 Plz* 05040 Ort* Überall

E-Mail* info@beispiel.de Website de.libreoffice.org Telefon* 0203040512345

Firma_Registriert* Base XRechnung USt-IdNr DE42424242 SteuerNr 123/456/7890 Steuerbefreiung (Grund)

Bank* Spasskasse Überall IBAN* DE75512108001245126199 BIC* SPKAUEBL42

RegisterE1 RegisterE2 RegisterE3

Zahlungsnotiz Zahlbar bis zum ?Datum? Logo

ZahlungTage 14 Zahlungsempfänger

Lastschrift Rechnung Stellenanzahl 4

GläubigerNr. DE98ZZZ09999999999 Netto, Steuer und Brutto in Rechnungszeilen

Zeitraumtext

MitarbeiterInnen

ID	Name*	Geschlecht	E-Mail*	Telefon*
1	Elly Workoholic	weiblich	work@beispiel.de	013457902
2	Peter Tüftel	männlich	tueftel@beispiel.de	012457910

Datensatz 1 von 2

im Rechnungsformular zur Warenangabe anzeigen

- Zeilennummer
- Warennummer
- Warezzusatz
- Rabatt
- StartDatum
- EndDatum
- Einheit
- Kontoreferenz Kunde
- Steuersatz und Steuercode

Vorgabe Steuersatz* 19 %

Vorgabe Steuercode* Normalsatz

Das Einstellungs-Formular enthält im oberen Teil nur einen Datensatz - den Datensatz aus der Tabelle "tbl_Firma", in den die zentralen Daten der eigenen Firma eingegeben werden. Auch hier sind alle Pflichtfelder mit einem '*' gekennzeichnet. Sie dienen zur Zusammenstellung der Rechnung sowohl als ODT- bzw. PDF-Datei als auch als XRechnung. Die Einträge in den Stammdaten können durch den Button **Firmeneinstellungen speichern** gespeichert werden. Sollen einzelne Einträge nur entfernt werden, so muss das entsprechende Feld geleert werden. Beim Feld für das «Logo» geht das über die rechte Maustaste:



In der unteren Tabelle ist für die XRechnung erforderlich, dass dort mindestens ein Datensatz (eine Person mit Namen, Telefonnummer und E-Mail-Adresse) steht. Die Einträge in der Tabelle werden durch Verlassen des Datensatzes gespeichert.

Die Felder unterhalb des Firmenlogos dienen dazu, die Rechnungserstellung zu beeinflussen. So wird intern eine Rechnung mit entsprechender Stellenzahl erstellt. Bei einer Stellenzahl von 4 Stellen reicht also das Ganze für 9999 Rechnungen in einem Jahr aus.

Die Markierfelder zeigen an, welche Spalten in dem Kundenformular angezeigt werden sollen. Sind alle Felder angeklickt, so erscheinen alle verfügbaren Felder. Für die Felder «Warennummer», «Warenzusatz», «Rabatt», «StartDatum» und «EndDatum» gilt: Werden sie nicht angezeigt wird auch kein Wert gespeichert. Bei der «Einheit» wird bei Nichtanzeige das XRechnungskürzel für 'Stück' gespeichert. Das Kürzel lässt sich für jeden Rechnungseintrag nutzen. Werden «Steuersatz und Steuercode» nicht angezeigt, so wird stattdessen der entsprechende Vorgabewert gespeichert. Die Nichtanzeige sollte nur bei Rechnungen mit einheitlichem Steuersatz genutzt werden. Der hier eingetragene Wert gilt schließlich für alle danach erstellten Rechnungszeilen.

Ist im oberen Bereich «Kleinunternehmer» markiert, so sollte bei den Vorgaben für die Steuer der Steuersatz auf '0 %' und der Steuercode auf 'Von der Steuer befreit' gesetzt werden. Anschließend kann «Steuersatz und Steuercode» problemlos von der Anzeige ausgeschlossen werden – der Steuersatz und der Steuercode sind in dem Moment vorgegeben. Ohne die Auswahl «Kleinunternehmer» wären hier '19 %' und 'Normalsatz' als Angaben sinnvoll.

zus. Einträge zu den Einstellungen des Ausdrucks

Rechnung ID

ID*	Tabellenkopf	Feld*	Format*	Breite	Reihenfolge*
1	Nr.	RechnungAusgang_ID	String	0,90	1
2	Datum	StartDatum	Date	0,90	2
3	Bis	EndDatum	Date	0,90	3
4		Anzahl	Number	0,95	4
5		Name_DE	String	1,00	5
6	Ware	Ware	String		8
7		Warennummer	String	2,00	7
8	Zusatz	Warenzusatz	String	2,15	6
9	Einzelpreis	Preis	Currency6	1,10	9
10	Rabatt	Rabatt	Percent	0,75	10
11	Steuer	Steuersatz	Percent	0,70	11
12	Summe	AnzahlPreis	Currency	1,10	12

Datensatz 1 von 12

Zusätzlich lässt sich über **Einstellungen zum Druck...** die Tabelle für die PDF-Ausgabe gestalten. Hier sind der Tabellenkopf, die Breite der Spalten und die Reihenfolge der Spalten einstellbar. Bleibt der Eintrag für den Tabellenkopf leer, so wird auch im Druck kein entsprechender Kopf angezeigt.

Bei den Breitereinstellungen ist die Maximalgröße für 100% Breite gleich dem Wert '10,00'. Die Breite für die Ware wird grundsätzlich über den verbleibenden Rest bestimmt. Bleibt ein anderes Feld leer, so wird per Abfrage dort erst einmal der Wert '1' gesetzt. Bei der Addition sämtlicher möglichen Felder wird klar, dass diese nicht komplett erscheinen können. Je nach Gestaltung der Rechnung kann es dann auch sein, dass viele Felder gar nicht benötigt werden. Die Felder "Anzahl", "Ware", "Preis" und "AnzahlPreis" stellen wohl die Mindestkonfiguration für eine Rechnung dar.

Die Formate für die verschiedenen Felder sind im Makrocode festgelegt. Hier muss lediglich erwähnt werden, dass neben dem normalen Feld für die Währung (2 Stellen nach dem Komma) auch eine Formatierung für mindestens 2 Stellen, maximal aber 6 Stellen nach dem Komma

existiert. Verbrauchsrechnungen, bei denen auch mit Werten bis zur 6. Nachkommastelle handiert wird, werden so korrekt dargestellt.

Die Reihenfolge muss nicht stufenlos notiert werden. Hier darf nur nicht ein Feld doppelt vorkommen. Das wird aber schon durch die Tabellenkonstruktion mit einem entsprechenden Index unterbunden.

Wird in den Einstellungen «Netto, Steuer und Brutto in Rechnungszeilen» angewählt, so stehen auch Datensätze mit der ID=13 und ID=14 zur Verfügung. Es erscheint dann unterhalb der Rechnungstabelle nur die Summierungszeile mit den Summen für Netto, Steuer und Brutto.

Das Konto-Formular

Konto

Filter
 Rechnungsnummer Kunde/Lieferant

Offene Zahlungen von Lieferungen

	Lieferant	Datum	LieferantZeichen	UnserZeichen	Anmerkung	Brutto	NochOffen
▶	[Seller name] → E-Mail: seller@em	11.12.17	0000123456	11002002-98765		10.686,20 €	10.686,20 €
	Dolmetscher → E-Mail: info@dolm	03.07.24	RE1983668	04011000-900X9		1.666,00 €	1.666,00 €
	Base XRechnung Extended → E-M.	25.03.24	R-2024-0001	-		737,80 €	737,80 €

Datensatz 1 von 3

	Datum	Betrag	lfd. Summe	lfd. off. Betrag
▶+	[?]			

Datensatz 1 von 1

Offene Zahlungen von geänderten Lieferungen

	Lieferant	Datum	LieferantZeichen	UnserZeichen	Anmerkung	Brutto	NochOffen
▶	[Seller name] → E-Mail: xxx@schu	24.04.15	PRG1502112	99000000-18188		10.555,30 €	10.555,30 €

Datensatz 1 von 1

	Datum	Betrag	lfd. Summe	lfd. off. Betrag
▶+	[?]			

Datensatz 1 von 1

Offene Eingänge von Rechnungen

	Kunde	Datum	RechnungsNr.	Teilrechnungen	Anmerkung	Brutto	NochOffen
▶	Dreaming & Co. → E-Mail: hierhin@	08.12.23	R-2023-0002		Arbeitsleistung	1.606,32 €	1.106,32 €
	Dreaming & Co. → E-Mail: hierhin@	25.03.24	R-2024-0001			737,80 €	737,80 €

Datensatz 1 von 2

	Datum	Betrag	lfd. Summe	lfd. off. Betrag
▶	10.12.23 [?]	500,00 €	500,00 €	1.106,32 €
+				

Datensatz 1 von 1

Kontostand

	Zahldatum	Titel	Firma	Datum	Lief-Zeichen/R-Nummer	Anmerkung	Brutto	Wert	lfdSumme
▶	01.09.19	Lieferung	[Seller trading name]	20.08.19	1234567		13,03 €	-13,03 €	-13,03 €
	30.04.21	Lieferung	Testverkäufer	23.04.21	112233		1.804,00 €	1.804,00 €	1.790,97 €
	15.11.23	Rechnung	Base XRechnung Extended	11.11.23	R-2023-0001		5.795,46 €	2.000,00 €	3.790,97 €
	20.11.23	Rechnung	Base XRechnung Extended	11.11.23	R-2023-0001		5.795,46 €	3.000,00 €	6.790,97 €
	01.12.23	Rechnung	Base XRechnung Extended	11.11.23	R-2023-0001		5.795,46 €	795,46 €	7.586,43 €

Datensatz 1 von 6

Dieses recht einfach gehaltene Formular soll nur aufzeigen, wie auch entsprechende Zahlungen verbucht werden können. Beträge an die Lieferanten müssen entrichtet werden, machen sich beim Kontostand also negativ bemerkbar. Der Kunde hingegen zahlt auf das Konto ein. Das Aufzeigen eines laufenden Kontostandes ist mit Base hier kein Problem. Das Formular lässt nur Eingaben in den rechts liegenden weiß hinterlegten Tabellenkontrollfeldern zu. Dabei können auch Teilzahlungen mit entsprechendem Datum vermerkt werden. Nur wenn hier auch Zahlungen vermerkt werden können diese Zahlungen auch bei einer Rechnungsstellung als bereits erbrachte Zahlungen angegeben werden.

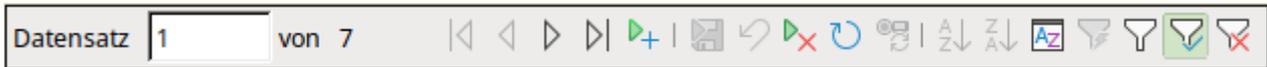
Eine Rechnung, die hier als komplett gezahlt gebucht wird, erscheint anschließend nicht mehr unter der Liste der Rechnungen, die bearbeitet werden können. Zur Anzeige nicht mehr bearbeitbarer Rechnungen dient der Button in dem Rechnungsformular.

Navigation durch die Formulare

Grundsätzlich sind die Formulare so eingestellt, dass von einem Formularfeld zum nächsten Formularfeld mit der Tabulatortaste gesprungen werden kann. Wird ein Datensatz verlassen, so wird er automatisch abgespeichert.

In den Formularen befinden sich auch Tabellenkontrollfelder. Bei einem Tabellenkontrollfeld geht das Abspeichern ebenfalls durch Verlassen des gerade geänderten oder neu eingegebenen Datensatzes. Das kann durch den Tabulator erfolgen (der dann zum nächsten Datensatz springt). Das geht auch mit den Pfeiltasten nach oben oder unten. Schließlich geht es auch durch Verlassen des Tabellenkontrollfeldes. Wird mit der Tastatur gearbeitet, so lassen die Tabellenkontrollfelder sich erst einmal nicht ansteuern. Lediglich das Springen aus dem Feld heraus mit **Strg** + **Tab** funktioniert in den Formularen.

Im Rechnungsformular und im Lieferformular ist ein Filter für die Auswahl der Kunden bzw. Lieferanten eingebaut:



Navigationsleiste mit aktiver Filterung (2. Button von rechts)

Die Filterung ist standardmäßig aktiv. Wird über das Listenfeld zum Filtern der oberste (leere) Eintrag gewählt, so zeigt das Formular trotz aktiver Filterung alle Datensätze an. Hier sind das insgesamt 7 Datensätze.

In dem Formular sind einzelne Navigationsleisten unter dem entsprechenden Formulareteil eingebaut. Sie speichern den Bereich ab, der direkt über ihnen bzw. auch neben ihnen liegt.

Unterhalb des gesamten Formulars befindet sich oft ebenfalls eine Navigationsleiste. Sie ist immer mit dem Teilformular verbunden, in dem gerade der Cursor steht.

Neueingabe von Daten

Im Formulardokument für die Kundeneingabe ist ein neuer Datensatz ausgewählt worden, weil ein neuer Kunde aufgenommen werden soll. In der Navigationsleiste steht «Datensatz 8 von 8». Pflichtfelder für die Eingabe sind zum einen in den Beschriftungen mit «*» gekennzeichnet. Sie sind außerdem durch ein Makro mit einer roten Umrandung versehen, solange in den Feldern nichts steht. Das Pflichtfeld «Land*» ist nicht rot umrandet, weil das Feld bereits bei der Neueingabe auf einen Standardwert eingestellt ist.

Das gelb hinterlegte Feld zeigt in hellgrau **<AutoFeld>** an. Dieses Feld wird automatisch durch die Datenbank mit dem entsprechenden Inhalt, einer für jeden Datensatz unterschiedlichen Zahl, versorgt. Diese Zahl kann anschließend auch als Kundennummer verwandt werden.

Ist in den Feldern zur Kundeneingabe Inhalt eingetragen, dann ist auch der Speicherbutton in der Navigationsleiste darunter aktiv. Die Speicherung wird allerdings mit einer entsprechenden Meldung verweigert, wenn eines der Pflichtfelder keinen Inhalt enthält.

Erst wenn ein neuer Kunde eingegeben und abgespeichert wurde, oder ein bestehender Kunde ausgewählt wurde, steht das Unterformular zur Rechnungserstellung zur Verfügung. Die Navigationsleiste ist dort zur Zeit inaktiv und steht auf «Datensatz 0 von 0». Auch sind die Pflichtfelder zur Zeit noch nicht rot hinterlegt.

Ein Unterformular braucht immer zwingend eine Verbindung zum darüber liegenden Formular, hier dem Formular für die Kundeneingabe. Nur wenn der entsprechende Wert, hier für das Unterformular der Wert aus dem Feld «ID» des Hauptformulars, bereits vorhanden ist, kann der Inhalt im Unterformular geschrieben werden.

✓ Hinweis

Die Sprachregelung bei Formularen ist für neue Nutzer gewöhnungsbedürftig. Einmal wird das, was in Base geöffnet wird, als Formular bezeichnet. Und dann liegen in diesem Formular wieder Formulare.

Das Formular, das in Base geöffnet wird, ist ein **Formulardokument**. Ein Formulardokument kann beliebig viele Formulare enthalten. Oft sind diese Formulare hierarchisch angeordnet: **Hauptformular** → **Unterformular** → **UnterUnterformular**

Übersicht über alte Rechnungen

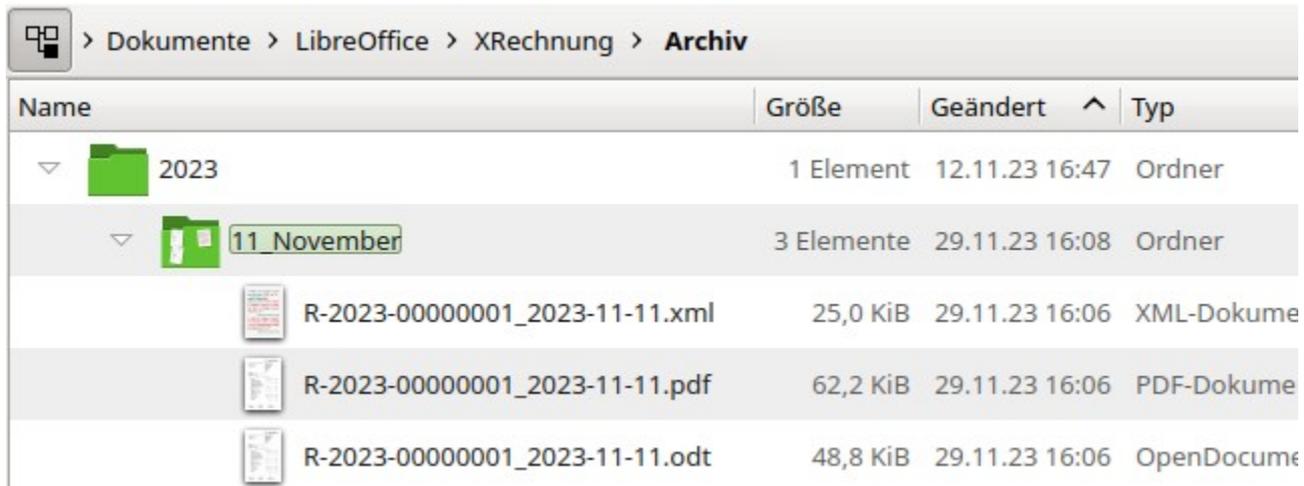
Das Formular zur Eingabe neuer Rechnung zeigt Rechnungen, die bereits als bezahlt notiert wurden, nicht mehr an. Diese Rechnungen dürfen nicht mehr geändert werden.

Über das Eingabeformular sind aber die alten Rechnungen weiter verfügbar und können dort auch bei Bedarf nach entsprechendem Inhalt durchsucht werden. Dafür befindet sich neben dem Listenfeld im Filterbereich ein freies Eingabefeld, in das einfach ein Begriff, unabhängig von Groß- oder Kleinschreibweise, angegeben werden kann. Taucht dieser Begriff bei den Kundendaten oder bei den Rechnungsinhalten auf, so wird der entsprechende Datensatz dargestellt.

Das Formular ist schreibgeschützt, damit nicht bereits bezahlte Rechnungen anschließend noch geändert werden.

Rechnung erstellen

Das Erstellen der Rechnung erzeugt grundsätzlich drei Dokumente.



Name	Größe	Geändert	Typ
2023	1 Element	12.11.23 16:47	Ordner
11_November	3 Elemente	29.11.23 16:08	Ordner
R-2023-00000001_2023-11-11.xml	25,0 KiB	29.11.23 16:06	XML-Dokume
R-2023-00000001_2023-11-11.pdf	62,2 KiB	29.11.23 16:06	PDF-Dokume
R-2023-00000001_2023-11-11.odt	48,8 KiB	29.11.23 16:06	OpenDocume

Die Dokumente werden mit dem Druckbefehl im Archiv-Ordner abgelegt.

Alle drei Dokumente werden gespeichert. Dabei wird das Archivverzeichnis für den entsprechenden Monat automatisch erzeugt. Bei einer Abspeicherung im ZUGFeRD-Modus endet der Dateiname der *.xml-Datei mit dem Zusatz «..._zug.xml». Außerdem kommt ein viertes PDF-Dokument hinzu, das die bestehende PDF-Datei und eine Kopie der «..._zug.xml» - Datei (als «xrechnung.xml») zu einer PDF-Datei mit dem Zusatz «..._zug.pdf» zusammenführt.

Das ODT-Dokument bleibt zur Ansicht geöffnet. Sollten hier Unstimmigkeiten auftauchen, so muss entsprechend im Rechnungsformular nachgebessert werden und die Erstellung erneut angestoßen werden. Nur so kann gewährleistet werden, dass alle Dokumente den gleichen Datenstand haben.

Mit dem Erstellen wird auch das im Betriebssystem festgelegte Standardmailprogramm aufgerufen. Es wird eine Mail mit Betreff und Inhalt sowie den beiden Dateien im PDF-Format und im XML-Format im Anhang erzeugt. Beim Versand von ZUGFeRD-Rechnungen wird hier nur die eine PDF-Datei angehängt, in der die passende XML-Datei eingebettet ist.

Tipp

Sollten mehrere Identitäten im Standardmailprogramm vorhanden sein, so wird die Standardidentität genutzt. Ist dies nicht die gewünschte Identität, so kann es sinnvoll sein, ein weiteres Mailprogramm als Standardmailprogramm mit der gewünschten Standardidentität laufen zu lassen.

Alternativ kann natürlich jedes Mal der Absender des Mailprogramms neu eingestellt werden.

Hinweis

Bisher nur bei der Kombination Ubuntu + Thunderbird bekannt: Der eigentlich als Inhalt gedachte Text erscheint nur bis zum ersten Komma oder ersten Absatz in dem Inhaltsbereich. Danach wird der Inhalt in den Betreff verschoben.

Andere Mailprogramme in Ubuntu haben diesen Fehler nicht. Thunderbird in anderen Umgebungen (OpenSUSE, Windows ...) bisher anscheinend auch nicht.

Gegebenenfalls muss hier leider der Text in dem Makro *StartMail* auf die Anrede gekürzt oder ohne Komma und Absatz vorgegeben werden.

Die Rechnungserstellung (Writer-Datei und PDF-Datei) beruht auf einer nach DIN 5008 aufgebauten Vorlagendatei⁵, die ggf. angepasst werden kann. Der Inhalt wird komplett aus der Datenbank gelesen. Am Ende einer Seite erfolgt, wenn die Rechnungszeilen nicht auf eine Seite passen, ein Umbruch zusammen mit der Ermittlung des Übertrags.



Base XRechnung Extended - Open Source Allee 42 - 05040 Überall

Dreaming & Co.
Hamburger Damm 12a
90919 Irgendwo

Base XRechnung Extended
Open Source Allee 42
05040 Überall
0203040512345
info@beispiel.de

Mitarbeiterin:
Elly Workoholic
work@beispiel.de
013457902

Rechnung
Rechnung Nummer: R-2023-0001
Kundennummer: K-000001
von: 01.09.2023
Abweichende Lieferadresse:

Datum: 11.11.2023
Lieferdatum: 02.11.2023
bis: 30.09.2023
 Dream at Home
 Bachstr. 234
 45678 Nowhere

Projektnummer: 4711 zum Ersten
Bestellnummer: 42GKO68

Vertragsnummer: Dies ist eine Vertragsnummer
Auftragsnummer: Müll333-Issus

UMSATZSTEUER-NR. **ZÄHLUNGSART**
 DE42424242 Zahlbar bis zum 25.11.2023

	Ware	Einzelpreis	Rabatt	Steuer	Summe
2	Stücke Open-Source DVD (LibreOffice, Gimp, Scribus usw.)	12,45 €	10 %	19 %	22,41 €
1	Stück Handbuch Base 7.6, gedruckt (alles über Base auf über 700 Seiten)	25,40 €		19 %	25,40 €
11	Stücke Stunden für die Arbeit an XRechnung, weitere Anmerkungen siehe unten. Ist natürlich im Zusammenhang mit dieser Rechnung nur scherzhaft gedacht, damit die Rechnung auch einen Seitenumbruch erzeugt.	35,00 €		19 %	385,00 €
4	Stücke Tafeln Schokolade mit Mandeln	1,75 €		7 %	7,00 €
1	Stück Dose Brustkaramellen für den schmutzigen Herbst	4,50 €		7 %	4,50 €
1	Stück Laufschuhe vor allem für Straßenlauf geeignet starke Dämpfung, für breite Füße	142,00 €	5 %	19 %	134,90 €
1	Stück Wanderrucksack geeignet für Touren mit Übernachtung 60l, mit zusätzlichem Sitzkissen und Regenschutz	120,00 €		19 %	120,00 €
1	Stück Tintenkartusche black	65,00 €		19 %	65,00 €
1	Stück Tintenkartusche cyan	66,53 €		19 %	66,53 €
1	Stück Tintenkartusche magenta	67,15 €		19 %	67,15 €
1	Stück Tintenkartusche yellow	64,12 €		19 %	64,12 €
2	Stücke A4-Papier Weißegrad 90% blauer Engel	7,15 €		19 %	14,30 €
1	Stück Digitalkamera NieCan Vollformatsensor 10mm bis 1000mm Zoom bei maximaler Blendenöffnung von 1,8 Mit Stativunterstützung automatische Motivsuche Geringste Speichergröße nicht unter 256 GB!	2.035,00 €		19 %	2.035,00 €
Übertrag:					3.011,31 €

FIRMA (REG. NAME)
Base XRechnung
Open Source Allee 42
05040 Überall

TELEFON/MAIL/WEB
0203040512345
info@beispiel.de
de.libreoffice.org

BANK/IBAN/BIC
Spasskasse Überall
DE75512108001245126199
SPKAUEBL42

Erste Seite des Rechnungsdrucks. Die Rechnung hat so viel Inhalt, dass ein Übertrag entsteht.

5 Dem Paket liegt eine Vorlage «Vorlage_Rechnung_Extension.ott» bei. Die Vorlage ist vom Adressfeld her DIN 5008 Form A angepasst. Wer lieber die Form B nutzen möchte kopiert die Vorlage Form B und gibt der Kopie den Namen «Vorlage_Rechnung_Extension.ott».

Seite 2 von 2 **Base XRechnung Extended** · Open Source Allee 42 · 05040 Überall

		Übertrag:		3.011,31 €
1 Stück	Per Anhalter durch den PC Wo ist der Weg zur ultimativen Lösung? Ist '42' der Aus-Knopf?	13,25 €	7 %	13,25 €
1 Stück	Zurück zum Analogen Der Weg von Tastatur und Bildschirm zu Keil und Steinplatte	45,12 €	7 %	45,12 €
1 Stück	Fotolabor komplett, gebraucht, mit allen Schikanen Farbvergrößerung, Zeitschaltuhr, Wannen in verschiedener Größe	250,00 €	19 %	250,00 €
5 Stücke	Arbeitsstunden Makros XRechnung	25,00 €	19 %	125,00 €
12 Stücke	Arbeitsstunden Beschreibung XRechnung	25,00 €	19 %	300,00 €
36 Stücke	Bugssuche und Bugbeseitigung XRechnung	25,00 €	19 %	900,00 €
10 Stücke	Benutzerdefinierte Anpassung XRechnung	25,00 €	19 %	250,00 €
Netto:				4.894,68 €
zugl. Portokosten:				1,50 €
abzgl. Rabatt:				25,00 €
MWSt 7%:				4,89 €
MWSt 19%:				913,39 €
Gesamt:				5.795,46 €

Der Übertrag erscheint auch auf der 2. Seite oben. Unter der Rechnung dann die Summierung sowie die Aufführung der Steuern und zusätzlicher Beträge auf Rechnungsebene.

Sämtliche Dokumente neben den Rechnungen werden ebenfalls im Archivordner gebündelt:

> Dokumente > LibreOffice > XRechnung > **Archiv**

Name	Größe	Typ
▶ Lieferungen	2 Elemente	Ordner
▶ 2024	1 Element	Ordner
▼ 2023	2 Elemente	Ordner
▼ 12_Dezember	4 Elemente	Ordner
▼ Anhang	4 Elemente	Ordner
R-2023-00000002_2023-12-08_A_4.b64	17,9 KiB	Base64-kodierte Daten
R-2023-00000002_2023-12-08_A_4.pdf	13,2 KiB	PDF-Dokument
R-2023-00000002_2023-12-08_A_3.b64	12,4 KiB	Base64-kodierte Daten
R-2023-00000002_2023-12-08_A_3.png	9,1 KiB	PNG-Bild
R-2023-00000002_2023-12-08.xml	42,9 KiB	XML-Dokument
R-2023-00000002_2023-12-08.pdf	58,4 KiB	PDF-Dokument
R-2023-00000002_2023-12-08.odt	45,9 KiB	OpenDocument Text
▶ 11_November	4 Elemente	Ordner

Archivordner mit Anhängen, die in die Rechnungen eingebaut werden (Base64). Auch zu sehen: Unterordner Lieferungen für Anhänge bei den Lieferungen.

Base XRechnung Extended - Open Source Allee 42 - 05040
Überall

Dreaming
Hamburger Damm 12a
90919 Irgendwo

Base XRechnung Extended

Open Source Allee 42
05040 Überall
0203040512345
info@beispiel.de

Mitarbeiter:
Peter Tüftel
tueftel@beispiel.de
012457910

Rechnung

Rechnung Nummer: R-2024-0001_4711 Datum: 25.03.2024
Kundennummer: K-000001

UMSATZSTEUER-NR. ZAHLUNGSART
DE42424242 Zahlbar innerhalb von 14 Tagen ab Rechnungsdatum

Datum		Ware	Einzelpreis	Summe
Transporte NRW				
01.03.24	1	Transport Rhelne → Münster	60,00 €	60,00 €
05.03.24	2	Transporte Rhelne → Steinfurt	45,00 €	90,00 €
07.03.24	1	Transport Rhelne → Ahaus	90,00 €	90,00 €
Transporte Niedersachsen				
14.03.24	2	Transporte Rhelne → Oldenburg	160,00 €	320,00 €
19.03.24	1	Transport Rhelne → Osnabrück	60,00 €	60,00 €
Netto:				620,00 €
MWSt 19%:				117,80 €
Gesamt:				737,80 €

FIRMA (REG. NAME)
Base XRechnung
Open Source Allee 42
05040 Überall

TELEFON/MAIL/WEB
0203040512345
info@beispiel.de
de.libreoffice.org

BANK/IBAN/BIC
Spasskasse Überall
DE75512108001245126199
SPKAUEBL42

Über untergeordnete Rechnungszeilen sind mit XRechnung-Extension auch Untergliederungen in Rechnungen möglich.

Die oben stehende Rechnung hat ein zusätzliches Feld für das Datum. Dafür konnten die Felder für den Rabatt (kein Rabatt auf Rechnungsebene) und für die einzeln aufgeführte Steuer (jeder Posten hat die gleiche Steuer) entfallen. Die Einblendung notwendiger Felder wird über die Abfragen und Makros automatisch geregelt.

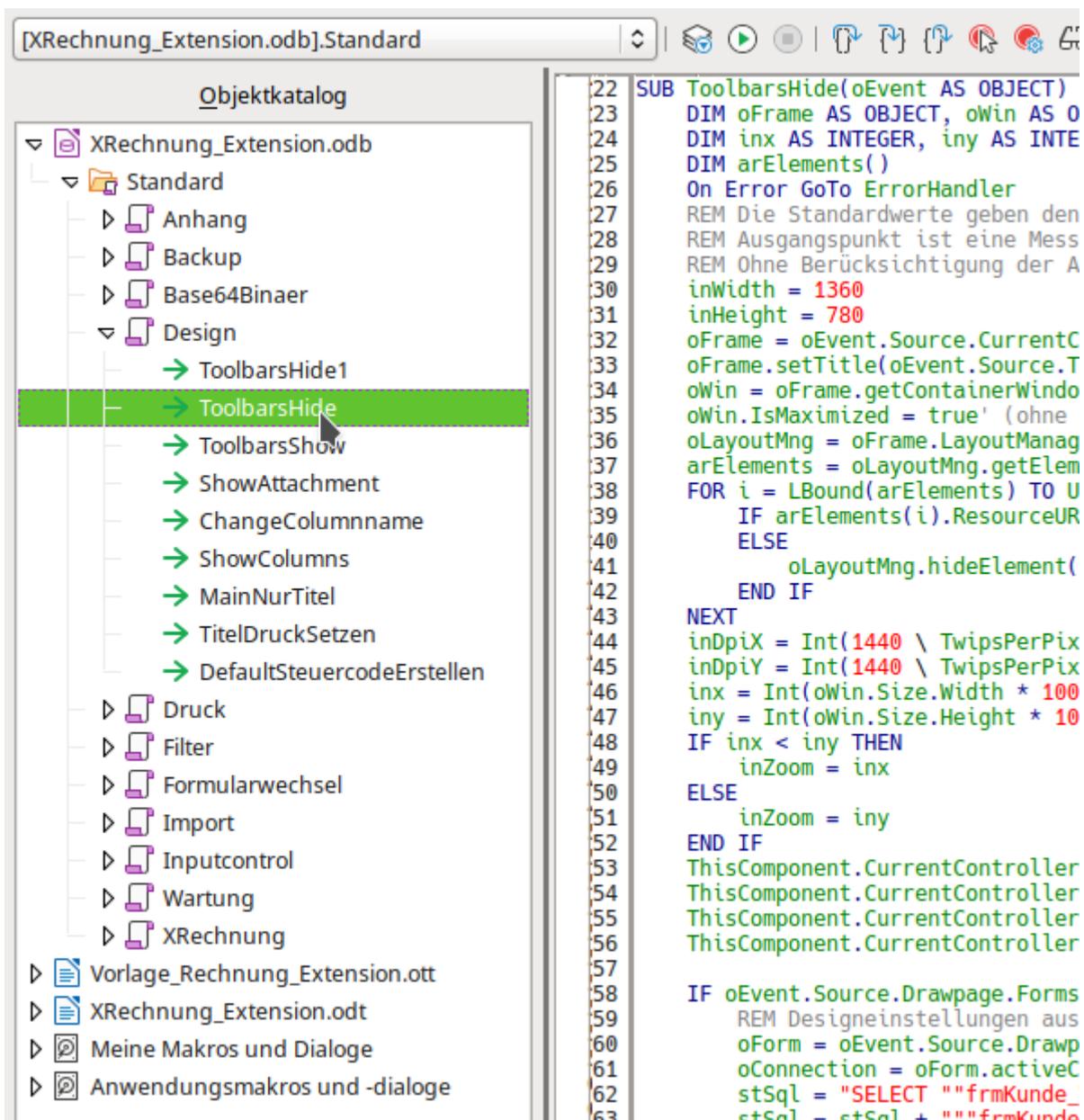
Bei genauem Blick auf die oben abgebildete Rechnung fällt ein «Schönheitsfehler» auf: Der Absender erscheint in dem Adressfenster mit einem Umbruch, weil die Schriftgröße nicht zu dem angegebenen Absender passt. Dies ist in der Vorlage nachjustiert worden.

Einstellungsmöglichkeiten

Die hier aufgezeigten Einstellungsmöglichkeiten sollen helfen, die Datenbank XRechnung_V2408_034.odt an die Bedürfnisse in der jeweiligen Firma anzupassen. Dabei wird immer berücksichtigt, dass eine Anpassung nicht dazu führt, dass die eigentliche XRechnung in Mitleidenschaft gezogen wird. Die XRechnung bleibt also gültig, auch die Makros funktionieren nach der Anpassung weiter.

Benutzeroberfläche in den Formularen

Die Formulare führen beim Start ein Makro aus, das das Erscheinen der Menüleiste unterbindet. Hier sollte vor der Bearbeitung des Formulars eine Änderung vorgenommen werden: **Extras → Makros → Makros bearbeiten...**



Im Objektkatalog wird **XRechnung_Extension.odt → Standard → Design → ToolbarsHide** aufgesucht. Rechts im Bild ein Teil des Codes, beginnen mit der Benennung der Prozedur.

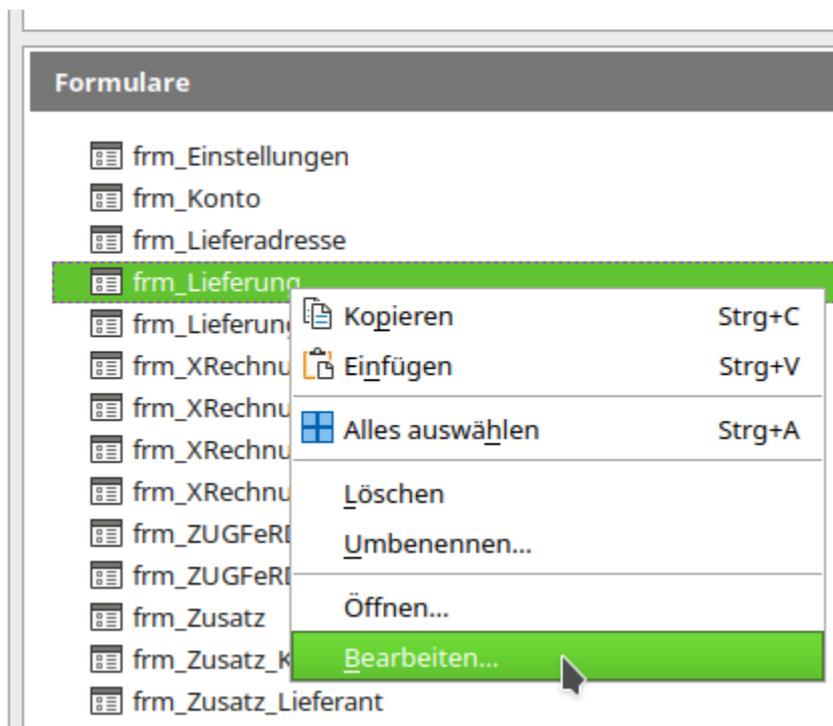
1. «ToolbarsHide» wird geändert in «ToolbarsHide2».

2. «ToolbarsHide1» wird aufgesucht. Die Prozedur enthält keinen Inhalt.
3. «ToolbarsHide1» wird geändert in «ToolbarsHide» und die Datenbankdatei abgespeichert.

Nach dieser Änderung startet statt des Makros zum Ausblenden der Werkzeugleisten und der Menüleiste ein leeres Makro. Die Leisten erscheinen.

Dieser Vorgang muss nach der Bearbeitung und dem Abspeichern des Formulars wieder rückgängig gemacht werden.

Um ein Formular zu ändern, muss es im Bearbeitungsmodus gestartet werden:



Beim Klick mit der rechten Maustaste öffnet sich ein Kontextmenü. Hier wird **Bearbeiten...** ausgewählt. Alternativ kann auch im Menü der Base-Datei **Bearbeiten** → **Bearbeiten...** gewählt werden.

✓ Hinweis

Je nach LibreOffice-Version und Betriebssystemoberfläche kann es sein, dass die Formularfelder im Bearbeitungsmodus nicht den standardmäßig weißen Hintergrund aufweisen. Der Hintergrund erscheint aber beim Ausführen der Formulare.

Formularhintergrund

Formulare sind vom Prinzip her in die Base-Datei eingebettete Writer-Dateien. Der Formularhintergrund lässt sich über **Format** → **Seitenvorlage** ändern.

Der voreingestellte Formularhintergrund nutzt einen Farbverlauf. Links oben im Formular sieht das Ganze so aus:



Wird jetzt der Farbhintergrund geändert, so passiert an dieser Stelle folgendes:



Das Listenfeld für die Navigation von einem Formular zum anderen ist mit einer Farbe hinterlegt, die der Farbe des umgebenden Hintergrundes angepasst ist. Mit einer rechten Maustaste auf dem markierten Listenfeld steht **Steuerelement-Eigenschaften...** zur Verfügung. **Eigenschaften: Listenfeld → Allgemein → Hintergrundfarbe** ermöglicht die Einstellung der Farbe auf die gewünschte Farbe, die mit dem Hintergrund vereinbar ist.

Wird der Hintergrund geändert, so kann das ggf. auch zu Problemen mit der Umrandung der Formularfelder geben. So ist in dem ersten Screenshot noch eine hellgraue Umrandung zu sehen, die beim zweiten Screenshot natürlich nicht mehr auffällt.

✓ Hinweis

Der Dialog für die Eigenschaften von Feldern sowie der Dialog zum Ändern von Formularen speichert die Einstellungen automatisch beim Schließen des Dialogs. Es sind keine Buttons zum Speichern oder Abbrechen des Dialogs vorhanden.

Die Farbe der Umrandung kann auch in dem allgemeinen Eigenschaften jedes Kontrollfeldes geändert werden. Um ein Kontrollfeld zu markieren reicht aber ein einfacher Mausklick auf das Feld nicht aus. Da die Eingabefelder zusammen mit den Beschriftungsfeldern gruppiert sind muss erst die Gruppe betreten und dann das entsprechende Feld markiert werden. Um ein einzelnes Feld einer Gruppierung direkt zu betreten, hilft **Strg** + linke Maustaste. Sollen gleich mehrere Felder aus unterschiedlichen Gruppen aufgerufen werden, so muss **Strg** + **Shift** zuerst gedrückt und gehalten werden und dann mit der linken Maustaste die Auswahl erfolgen. Jetzt können alle gewünschten Felder markiert werden und in einem Zug die Umrandung bei allen Feldern in den allgemeinen Eigenschaften geändert werden. Bei einer Mehrfachauswahl ist aber zu Bedenken, dass nur die Eigenschaften zur Auswahl gestellt werden, die bei allen gewählten Feldern vorhanden sind. Wird also aus Versehen z. B. eine Navigationsleiste mit ausgewählt, dann erscheint nicht die Möglichkeit, eine Umrandung für alle Felder einzustellen.

Neben der Umrandung kann in einer Mehrfachauswahl natürlich auch die Schriftart oder andere Eigenschaften für alle Felder geändert werden.

! Vorsicht

Bei einer Mehrfachsortierung sollte nie die Eigenschaft **PositionX** oder **PositionY** eingestellt werden. Haben plötzlich alle gewählten Felder die gleiche Position, so liegen sie übereinander!

Auch die Einstellung des Namens der Felder sollte nicht hier erfolgen. Viele Formularfelder werden durch Makros über den Namen identifiziert. Wenn plötzlich alle Felder den gleichen Namen haben, versagen die Makros.

Felder positionieren

Natürlich können auch Felder neu positioniert werden. Hierbei sollte aber, sofern eine Navigation mit Tabulator angestrebt wird, auch die Tabulatorreihenfolge neu eingestellt werden.

In den Tabellenkontrollfeldern kann die Position der Felder und auch die Breite der Felder beeinflusst werden. Wenn bestimmte Felder grundsätzlich auf unsichtbar geschaltet wurden (Formular "Einstellungen"), dann entsteht auf jeden Fall in den Tabellenkontrollfeldern für die Rechnungszeileneingabe zusätzlicher freier Platz. Eine Beeinflussung der Breite der Felder ist auch während der Eingabe selbst möglich. Nur wird diese Verbreiterung nicht gespeichert und geht mit dem Schließen des Formulars verloren.

Felder entfernen

Felder, die keine Pflichtfelder sind, können gelöscht werden. Der Bereich für die Erstellung der einzelnen Rechnungszeile in den Kunden-Formularen sollte allerdings unberührt bleiben. Hier wird über die Einstellungen einfach die Anzeige von Feldern geregelt.

Wer keine Anhänge für Rechnungen einlesen möchte, kann den gesamten Bereich «Anhänge» markieren und entfernen. Gleiches gilt für die «Ermäßigungen und Aufschläge auf Rechnungsebene».

Wer den Bereich für die «Untergliederung Rechnungsposten» im Kundenformular für die erweiterte XRechnung nicht braucht sollte nicht dieses Formular ändern sondern auf das Formular für die einfache XRechnung oder für ZUGFeRD nutzen.

✓ Hinweis

Das Formular, mit dem die Datenbank startet, wird in einem Makro festgelegt.

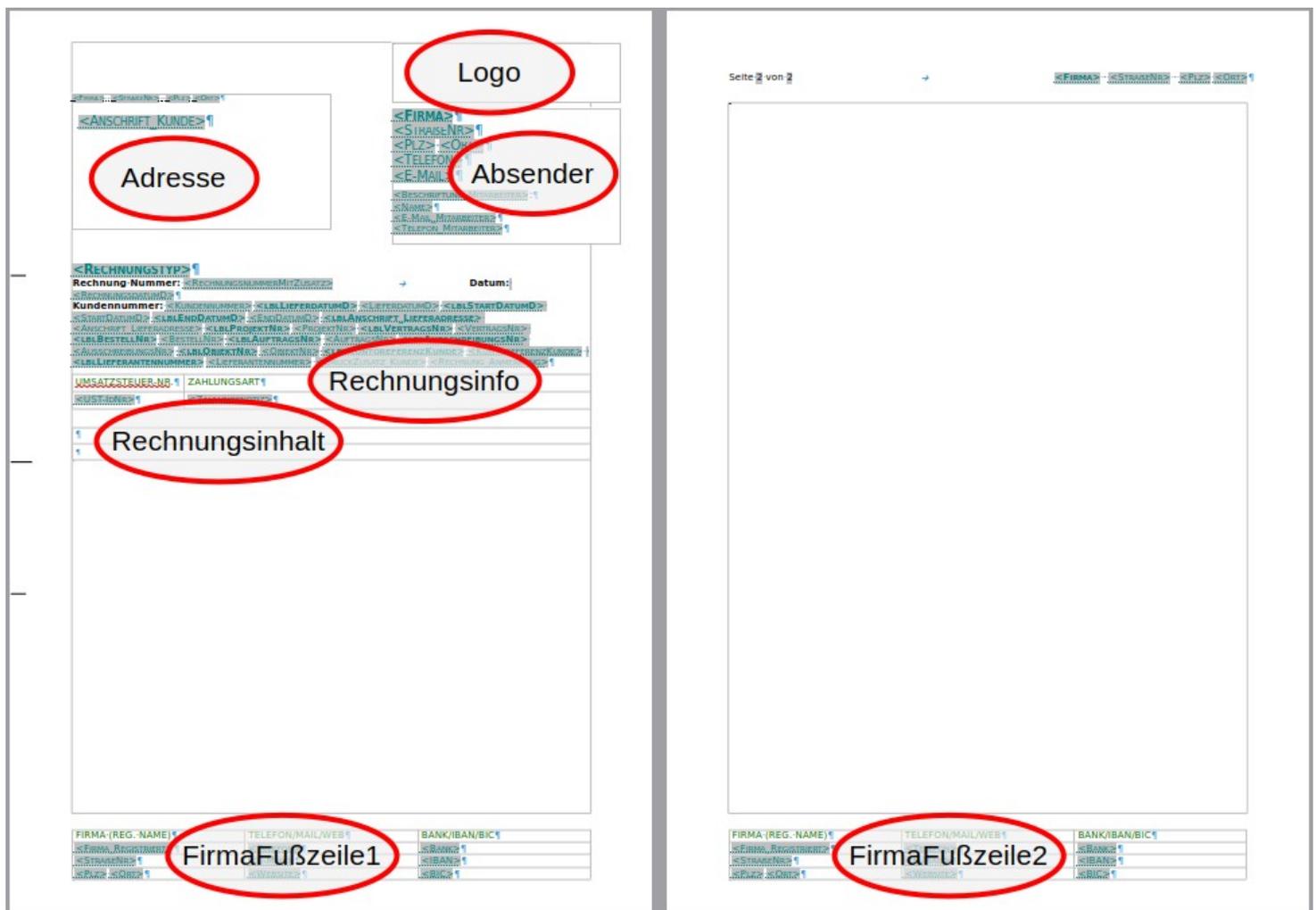
In **Extras** → **Makros** → **Makros bearbeiten** → **XRechnung_V2408_034.odt** → **Standard** → **Backup** → **FilterStart** steht zu Anfang

```
001 stForm = "frm_XRechnung-Extension"
```

Hier kann jetzt ein anderes Formular angegeben werden. Das wird dann das zukünftige Startformular.

Ausdruck der Rechnung

Das Aussehen des Ausdrucks wird überwiegend mit einer Vorlage gesteuert. Die Vorlage besteht dabei aus einer ersten Seite mit Adressfeld und der Vorlage für die Folgeseiten ohne Adressfeld.



Um die zweite Seite der Vorlage sichtbar zu machen, muss direkt nach der Tabelle «Rechnungsinhalt» ein Seitenumbruch eingefügt werden (**Einfügen** → **Seitenumbruch**). Dieser Seitenumbruch muss vor dem Abspeichern der Vorlage wieder entfernt werden, da sonst der Druck grundsätzlich aus mindestens 2 Seiten besteht.

Rahmen und Tabellen sind mit Namen versehen. Die Namen sind über die Rahmeneigenschaften bzw. die Tabelleneigenschaften zu erfahren.

- Rahmen «Logo»
Der Rahmen enthält das Bild, was im Formular «Einstellungen» als Logo der Firma eingegeben wurde. Fehlt dort eine Eingabe, so bleibt der Rahmen leer. Der Rahmen kann frei positioniert werden. Auch die Größeneinstellung ist beliebig. Der Rahmen sollte allerdings nicht entfernt werden, wenn nicht innerhalb der Prozedur «FillTableCarryOver» der Code an der entsprechend gekennzeichneten Stelle auskommentiert wird.
- Rahmen «Adresse»
Der Rahmen ist je nach Vorlage nach DIN 5008, Form A (siehe oben) oder Form B positioniert und von der Größe her festgelegt. Position und Größe sind deswegen in **Eigenschaften** → **Rahmen** → **Optionen** → **Schützen** vor zufälligen Veränderungen geschützt. So bleibt gewährleistet, dass sich an der entsprechenden Position auch das Fenster eines Fensterumschlags befindet.
Der Rahmen wird nur zur korrekten Positionierung benötigt. Die in dem Rahmen enthaltenen Platzhalter sind genauso benannt wie die Felder aus «vivi_Rechnung». Das Makro vergleicht nur Felder in der Tabellenansicht und Platzhalter in der Vorlage und füllt dann gleichnamige Platzhalter aus.

- Rahmen «Absender»
Hier werden alle Absenderinformationen zusammengefasst. Dieser Rahmen kann nach unten ausgedehnt werden.
- Tabelle «Rechnungsinfo»
Diese Tabelle dient nur zur Positionierung und Formatierung des Inhaltes. Diese Tabelle enthält entweder die Umsatzsteuernummer (mit entsprechender Überschrift) oder bei Kleinunternehmern die einfache Steuernummer (ebenfalls mit entsprechender Überschrift). Daneben ist dann eine allgemeine Zahlungsinformation enthalten.
- Tabelle «Rechnungsinhalt»
Die Tabelle enthält nur zwei Zeilen und eine Spalte. In der ersten Spalte sollen die Tabellenköpfe stehen, ab der zweiten Zeile wird dann der eigentliche Inhalt jeder Rechnungszeile ausgegeben.
- Tabelle «FirmaFußzeile1»
Dieser Tabelle sollte keine weitere Spalte hinzugefügt werden. Per Makro wird die Tabelle angesprochen und mit einer weiteren Spalte versehen, wenn im Formular «Einstellungen» Registereinträge gemacht wurden. Dann enthält die hinzugefügte Spalte den Namen «Registereinträge». Dieser Name kann im Makro «FillTableCarryOver» ggf. geändert werden.
- Tabelle «FirmaFußzeile2»
Gleich der Tabelle FirmaFußzeile1. Diese Tabelle ist auf den Folgeseiten der ersten Seite in der Fußzeile für den gleichen Inhalt gedacht. Wird die Tabelle auf der zweiten Seite entfernt, so erfolgen die Informationen nur auf der ersten Seite.

Spalten im Rechnungsinhalt

	ID	Tabellenkopf	Feld	Format	Breite	Reihenfolge
	1	Nr.	RechnungAusgang_ID	String	0,90	1
	2	Datum	StartDatum	Date	0,90	2
	3	Bis	EndDatum	Date	0,90	3
	4		Anzahl	Number	0,95	4
	5		Name_DE	String	1,00	5
	6	Ware	Ware	String		8
	7		Warennummer	String	2,00	7
	8	Zusatz	Warenzusatz	String	2,15	6
	9	Einzelpreis	Preis	Currency6	1,10	9
	10	Rabatt	Rabatt	Percent	0,75	10
	11	Steuer	Steuersatz	Percent	0,70	11
	12	Summe	AnzahlPreis	Currency	1,10	12

Die Tabelle "tbl_Druck" regelt den Inhalt der Spalten in der Tabelle «Rechnungsinhalt» der obigen Vorlage. Insgesamt sind 12 verschiedene Einträge möglich. In der Regel werden für eine Rechnung aber längst nicht alle Einträge benötigt. Standardfelder dürften hier die Felder mit der ID=4 ("Anzahl"), ID=6 ("Ware"), ID=9 ("Preis") und ID=12 ("AnzahlPreis") sein. Das Primärschlüsselfeld "ID" sollte auf keinen Fall geändert werden, ebenso wenig die Position der Einträge in der Spalte "Feld" mit der dazugehörigen Spalte "Format". Am besten wird die Tabelle nur über das entsprechende Formular bearbeitet. Hier ist die Änderung nur in dem Bereich erlaubt, der keine größeren Probleme verursachen kann.

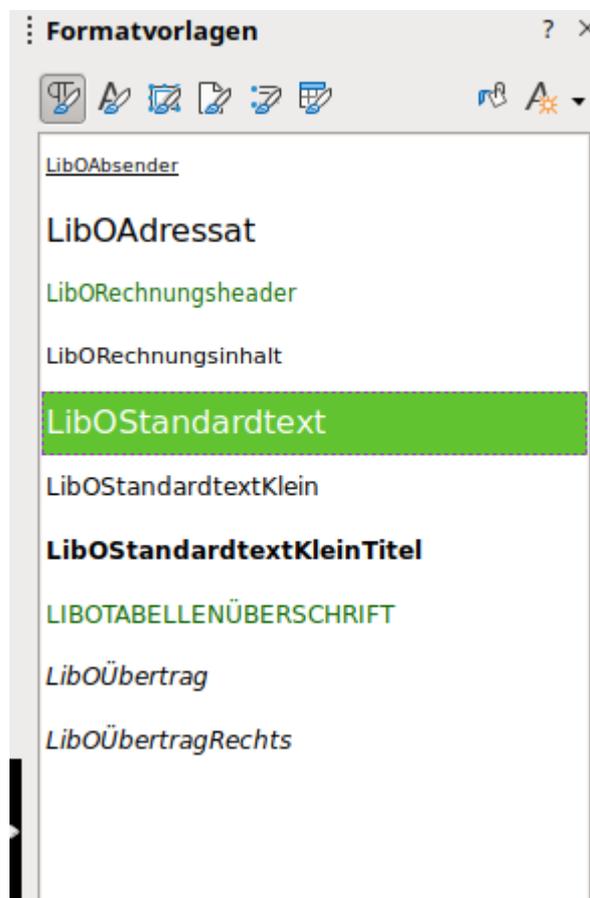
- «Tabellenkopf»
Hier stehen die Einträge, die in der Tabelle des Ausdrucks als Titel erscheinen. So erscheint hier für das Feld "RechnungAusgangID» aus der Ansicht "viw_Rechnung_Aus-

gang" im Kopf der Eintrag «Nr.». Für einige Felder ist gar kein Eintrag vorgesehen. Gerade bei der Anzahl und der Maßeinheit dazu ("Name_DE") würden solche Einträge mehr Platz einnehmen als die darunterliegenden Zahlen und Maßeinheiten. Der Inhalt ist in der Regel auch so klar.

- «Format» (nicht ändern!)
Für den Preis ist ein Währungsformat mit 2 bis zu 6 Nachkommastellen vorgesehen. Kleine Preisangaben gibt es z. B. bei Energieabrechnungen, so dass mit der einfachen Angabe von 2 Nachkommastellen die Rechnung nicht stimmig erscheint.
- «Breite»
Es steht für die gesamte Breite der Tabelle (100%) eine Breitenangabe von 10 zur Verfügung. Ist die Tabelle also 17 cm breit, so entspricht '1' einer Breite von 1,7 cm. Die Breitenangaben für eine Rechnung dürfen also auf keinen Fall 10 übersteigen. Die Breitenangaben für die "Ware" ist frei gehalten und wird berechnet. Einträge in dieser Liste werden für die Ware nicht berücksichtigt.
Werden viele Felder in der Tabelle benötigt, so muss ggf. die Schriftgröße in der Tabelle verkleinert werden oder eine ganz neue Vorlage erstellt werden, die den Brief im Querformat beschreiben lässt.
- «Reihenfolge»
Mit der entsprechenden Zahl wird angegeben, in welcher Reihenfolge die einzelnen Felder im Ausdruck erscheinen sollen. Hier darf kein Wert doppelt vorkommen. Dies wird in der Tabelle durch einen Index abgesichert. Abstände zwischen den Nummerierungen sind egal, so dass gegebenenfalls auch deutlich größere Zahlen einzelne Posten weiter nach hinten setzen können.

Schriftformatierungen

Sämtliche Schriften sind durch Formatvorlagen festgelegt. Im Sinne eines einheitlichen Aussehens sollten nur die Vorlagen geändert werden, nicht aber Inhalt hart formatiert werden.

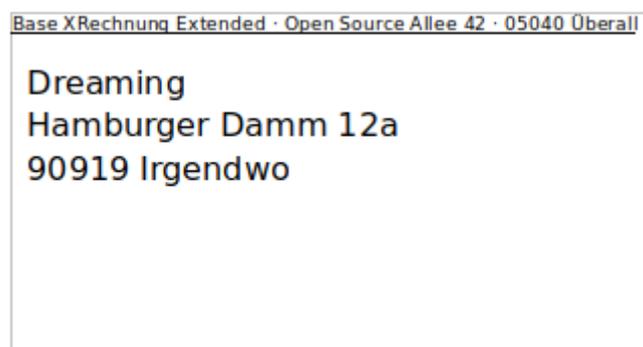


Die benutzerdefinierten Formatvorlagen beginnen alle mit dem Kürzel «LibO». Diese Vorlagen können entsprechend angepasst werden. Wem also z. B. die Farbgebung nicht gefällt, der/die wird hier fündig. Bei der Vorlage «LibOAbsender» sollte darauf geachtet werden, dass der komplette Absender tatsächlich in den entsprechenden Rahmen passt und nicht einfach umgebrochen wird.

Die Vorlage «LibORechnungsinhalt» wird natürlich auch die Umbrüche in den Spalten der Rechnungsposten beeinflussen. Wird die Schrift zu groß gewählt, so passt gegebenenfalls ein Datum oder der Preis einer Ware nicht komplett in eine Spalte.

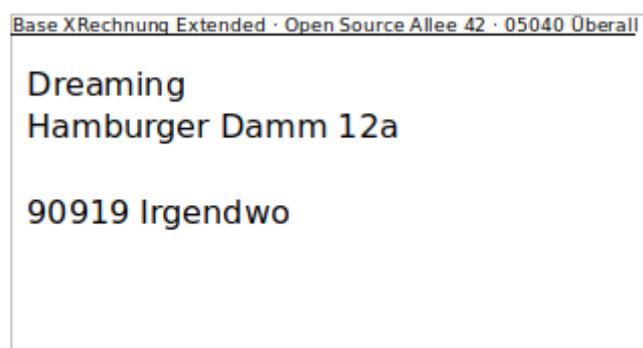
Wird eine andere Schriftart gewählt, so hängt natürlich auch von der Laufweite der Schriftart die entsprechende Breite der benötigten Spalten ab. Mit «DejaVu Sans» ist eine Schriftart mit etwas größerer Laufweite in den Absatzvorlagen gewählt. Würde die Schriftart durch «Liberation Sans» oder «Arial» ersetzt, dann dürften bei gleicher Schriftgröße auf keinen Fall Probleme mit der Textbreite auftauchen.

Adressfeld



Im Adressfeld ist oben der Absender und unten den Adressat enthalten. Die Größe des Rahmens gibt dabei die Größe für Adressfelder an, die auch in einem Brief im entsprechenden Sichtfenster zu sehen sind.

Enthalten Adressen nur drei Zeilen, so ist sehr viel Platz unterhalb der Adressen frei. Manche Geschäftspost sieht dann eher so aus:



Der Ort wird hier durch einen doppelten Return von der Straße getrennt. Es bleibt auch noch genügend Platz, um unterhalb bei Schreiben in andere Länder das jeweiligen Land auszugeben.



Der Rahmen sieht in der Vorlagendatei allerdings so aus, dass dort lediglich ein einziger Platzhalter für Text, nämlich <Anschrift_Kunde>, vorgesehen ist. Die Anschrift wird also direkt in der Datenbank zusammengestellt, nicht aus einzelnen Feldern ausgelesen.

```
"tbl_Kunde"."Firma"||ASCII_CHAR(13)||"tbl_Kunde"."StraßeNr"||ASCII_CHAR(13)||"tbl_Kunde"."Plz"||' '||"tbl_Kunde"."Ort"||
CASE WHEN "tbl_Kunde"."Land" = 'DE' OR "tbl_Kunde"."Land" IS NULL THEN '' ELSE ASCII_CHAR(13)||UPPER("tbl_Land"."Land") END AS "Anschrift_Kunde",
```

Die Anschrift wird in der Ansicht «viw_Rechnung» erstellt. Der obige Codeausschnitt zeigt, dass der Anschrift zum Schluss der Name «Anschrift_Kunde» zugewiesen wird. Dies ist auch der Name des Platzhalters in der Writer-Vorlagendatei. Mit **ASCII_CHAR(13)** sind in dieser Zusammenstellung die Absatzmarken gesetzt. Mit || werden einzelne Elemente verbunden.

```
"tbl_Kunde"."Firma"||ASCII_CHAR(13)||"tbl_Kunde"."StraßeNr"||ASCII_CHAR(13)||ASCII_CHAR(13)||"tbl_Kunde"."Plz"||' '||"tbl_Kunde"."Ort"||
CASE WHEN "tbl_Kunde"."Land" = 'DE' OR "tbl_Kunde"."Land" IS NULL THEN '' ELSE ASCII_CHAR(13)||UPPER("tbl_Land"."Land") END AS "Anschrift_Kunde",
```

Hier wird jetzt zwischen "tbl_Kunde"."StraßeNr" und "tbl_Kunde"."Plz" ein zusätzlicher Umbruch eingefügt: ||**ASCII_CHAR(13)**. Danach werden die Anschriften mit einem doppelten Return nach der Straßenbezeichnung ausgegeben.

Platzhalter

Neben den Tabellen und dem Logo werden Inhalte auch über Platzhalter in die Vorlage übertragen. Die Platzhalter können, wenn nicht gewünscht, problemlos gelöscht werden. Der Inhalt muss dann gegebenenfalls direkt in die Vorlage eingetragen werden, wenn er notwendig in der Rechnung ist. Worauf auf jeden Fall geachtet werden sollte: Sind Inhalte der Datenbank und der Rechnung anschließend nicht identisch, so stimmt die gedruckte Rechnung nicht mit der XRechnung überein. Für die XRechnung gelten die Firmendaten aus der Datenbank, nicht die aus der Writer-Vorlage.

Archivierung

Die Form der Archivierung wird in einer entsprechenden Prozedur *ExportPDF* festgelegt. In Formularen werden

- die Anzahl der Rechnungsstellen festgelegt (Formular «Einstellungen»)
- ein eventuell zusätzliches Kürzel bei der Erzeugung des Dateinamens erstellt (Formular «Kunde», direkt beim Kunden im ersten Formularteil)
- eine interne (bisher verwendete) Rechnungsformatierung in der Rechnung mitgeführt (Formular «Kunde», «Rechnungsnummer-Zusatz» bei jeder neuen Rechnung).

Diese Vorgaben betreffen nur den Dateinamen, nicht die Art der Archivierung.

In **Extras** → **Makros** → **Makros bearbeiten** → **XRechnung_V2408_034.odt** → **Standard** → **Druck** → **ExportPDF** wird in dieser Zeile der Pfad zum Archiv zusammengestellt:

```
001 stFile = stPrintDir & "Archiv/" & stYear & "/" & stMonth & "/" & stFilename
```

Der Pfad, in dem die Base-Datei liegt, wird als **stPrintDir** genutzt. An diesen wird das Unterverzeichnis **Archiv** angehängt. Dieses Verzeichnis kann hier problemlos anders benannt werden. Die Anordnung sollte aber nicht so geändert werden, dass der Pfad nicht unterhalb der Base-Datei liegt. Relative Pfade können in Windows zu Problemen führen.

Aus dem Rechnungsdatum wird das Jahr und der entsprechende Monat ermittelt. Dabei wird über ein Array dem Monat noch der Monatsnamen hinzugefügt, so dass alle Rechnungen eines bestimmten Monats immer zusammen in einem Verzeichnis landen. Die Rechnungen sind dann nach der Rechnungsnummer des entsprechenden Jahres geordnet, wenn nicht ein «Kürzel» für den Kunden hinzugefügt wurde.

Enthalten eingehende Rechnungen Anhänge, so werden diese nach **Standard → Import → Import** in ein Unterverzeichnis «Lieferungen» geschrieben:

```
001 stAttachmentPath = "Archiv/Lieferungen/" & stYear & "/" & stMonth & "/" &
    arAttachFilename(i)
```

Die eingehenden Rechnungen sollten also möglichst auch in diesem Pfad gespeichert werden, damit das Archiv übersichtlich zu handhaben ist.

Die obige Makrozeile befindet sich im Import-Modul an zwei unterschiedlichen Stellen. Dies hängt damit zusammen, dass es eben zwei zulässige unterschiedliche Standards für die XML-Dateien der XRechnung gibt, siehe das folgende Kapitel [Erforderliche Einträge in der XRechnung](#).

Einstellungsmöglichkeiten direkt im Makrocode

Einige Einstellungen wurden fest in den Makrocode übernommen, sind aber vielleicht nicht so gewünscht.

Automatischer Start eines Formulars

Aufruf aus
Makro: FilterStart

Beenden von Base nach dem Schließen des letzten Formulars

Aufruf aus
Makro: DatabaseClose

Dies ist zu Beginn ausgeschaltet, indem «DatabaseClose» als Bezeichnung einem leeren Dummy zugeordnet ist. Das automatische Schließen kann eingeschaltet werden, indem der Dummy umbenannt wird zu «DatabaseClose1» und die andere Prozedur «DatabaseClose1» zu «DatabaseClose» umbenannt wird.

Darstellungsgröße des Formulars

Aufruf aus
Makro: ToolbarsHide

In dem Makro befinden sich die Zeilen

```
001 inWidth = 1470
002 inHeight = 840
```

Hier wird die Größe des größten Formulars bei einem Zoom von 100 % notiert. Sollten Scrollbars an der rechten Seite oder unterhalb des Formulars auftauchen, so sind diese Werte größer einzustellen.

Vergrößerung von Tabellenfeldern

Eine Bearbeitung der Tabelleneigenschaften ist **mit äußerster Vorsicht** zu handhaben. Dies sei hier nur an einem Beispiel verdeutlicht:

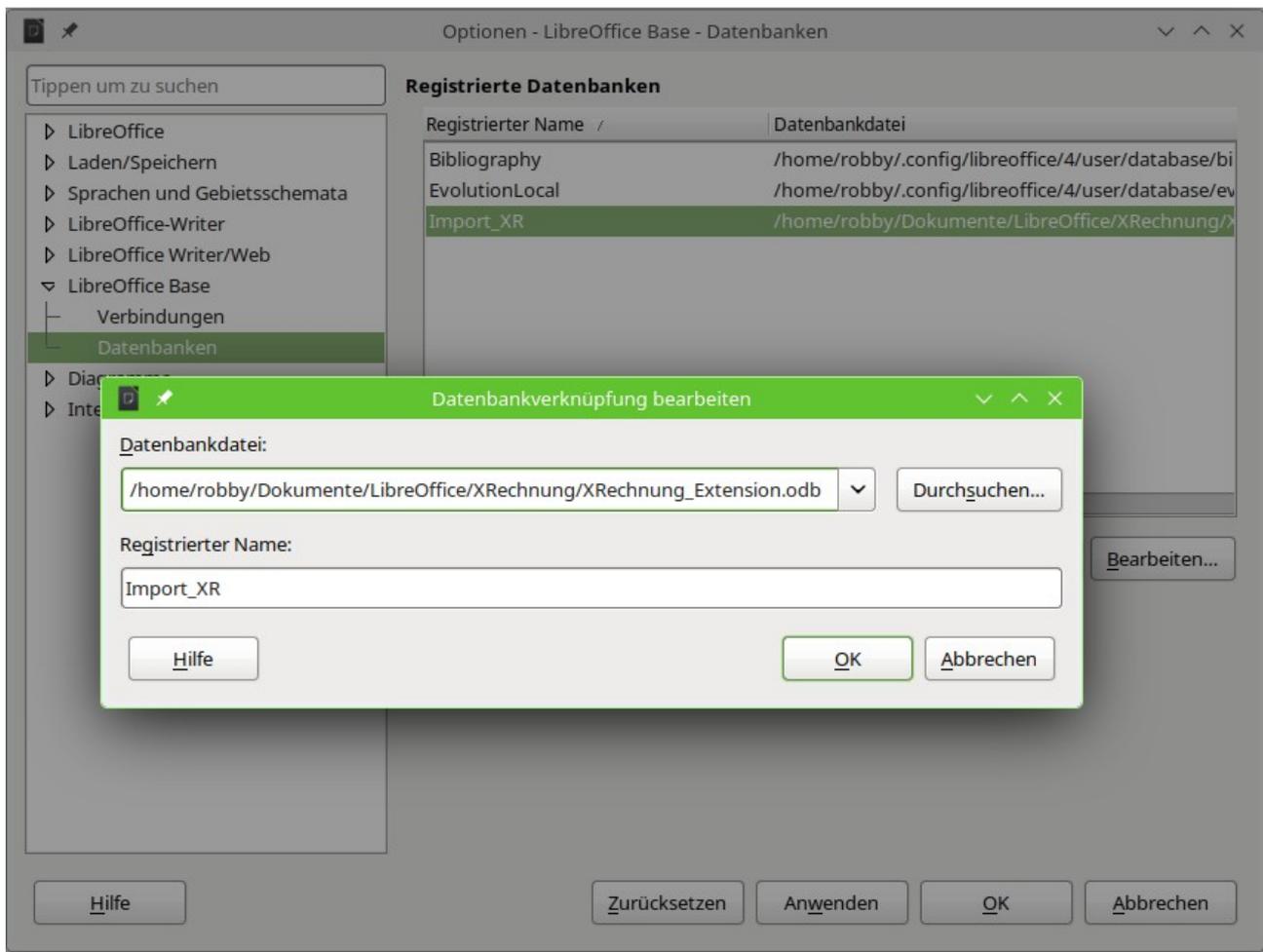
In der Tabelle "tbl_Ausgang" befindet sich das Feld "Ware". Das Feld ist zur Zeit auf eine Größe von 250 Zeichen eingestellt, was ungefähr 3 Zeilen in dieser Anleitung entspricht. Reicht einer Firma die Größe nicht aus, so kann problemlos auf 1000 Zeichen erhöht werden. Wird dann der erste Datensatz mit über 250 Zeichen eingegeben, so funktioniert plötzlich der Rechnungsausdruck nicht mehr, weil in der Ansicht "viw_Rechnung_Ausgang" das Feld "Ware" noch mit der alten Definition von 250 Zeichen für die Datenbank abgespeichert ist. Folgende Punkte sind bei einer Änderung von Feldern zu beachten:

- Eine Änderung der Feldgröße ist nur zu einem größeren Inhalt hin möglich. Eine Verkleinerung lässt die Firebird-Datenbank nicht zu.
- Nach einer Änderung muss nicht nur die Tabelle abgespeichert werden, sondern auch über **Extras → Makros → Makro ausführen → Wartung → ViewsErstellen** ein Neuschreiben der Ansichten ausgelöst werden.
- Bei einem Update von so einer geänderten Version zu einer neueren Version der XRechnungs-Datenbank muss zuerst die in der alten Version vollzogene Änderung erneut durchgeführt werden. Ansonsten können gegebenenfalls einige Daten nicht importiert werden, da die Felder in der Originalversion ja kleiner sind als die in der selbst bearbeiteten Version.

Update von einer Vorgängerversion

Diese Base-Datenbankversion zur XRechnung wird beständig weiter entwickelt. Wer bereits eine vorherige Version genutzt hat, fragt sich sicher: Wie kann ich an die Funktionen der neueren Version kommen?

Das Update-Verfahren macht sich dabei zu eigen, dass die vorhergehenden Versionen grundsätzlich weniger Felder in den Tabellen aufwiesen als die aktuellen Versionen, da ja für jede zusätzliche Information, die für eine XRechnung gespeichert werden soll, zusätzliche Felder erstellt werden. Um auf die Version zugreifen zu können, die die aktuell genutzten Daten enthält, muss die Datenbankdatei in LibreOffice registriert werden: **Extras → Optionen → LibreOffice Base → Datenbanken → Neu**



Als Datenbankdatei wird der Pfad für die bisher genutzte Base-Datei angegeben, hier also zur Datei XRechnung_Extension.odt. Wichtig ist jetzt der registrierte Name. Wird einfach nur eine automatische Registrierung durchgeführt, so würde hier der Name der Datenbankdatei ohne die Erweiterung «.odt» stehen. Der **Registrierte Name** soll aber 'Import_XR' sein.

Diese Verbindung wird für das Makro zum Update genutzt.

Aufruf aus

Makro: *Import_all*

Dabei werden die Tabellen der neuen Datenbankdatei zuerst geleert, anschließend die Inhalte der alten Datenbankdatei in die neue Datenbankdatei kopiert und schließlich die Datenbank so eingestellt, dass auch die automatisch erstellten Primärschlüsselfelder einwandfrei funktionieren.

Wurde in den Firmeneinstellungen ein Bild (als Logo) genutzt, so muss dieses Bild neu eingefügt werden. Binäre Inhalte werden bei der Importfunktion nämlich nicht mit übertragen.

Wurde in älteren Versionen bereits Rechnungen mit einem Zahldatum versehen, so muss dies im Konto-Formular neu erfolgen. Das Zahldatum konnte nur eine Komplettzahlung verbuchen. Anzahlungen waren so nicht möglich, so dass für diese Funktion neue Tabellen erforderlich wurden.

Die neue Datenbankdatei muss im gleichen Pfad wie die vorhergehende Datenbankdatei liegen, da sonst die bisherige Archivierung nicht genutzt werden kann. Dies wird zuerst dann auffallen, wenn Bilder in die Rechnungen eingefügt wurden und jetzt nicht mehr verfügbar sind.

✓ Hinweis

Da bei einem Update alle Testinhalte aus der neuen Datenbankdatei gelöscht werden sollte das Kommando dazu nur zum Start der neuen Datenbankdatei einmal ausgeführt werden. War das Update erfolgreich, so wird die Registrierung der Datenbankdatei nicht mehr benötigt. Der Eintrag zu Registrierung kann gelöscht werden. Danach wäre auch das Makro nicht mehr ausführbar, da eben diese Registrierung zum Zugriff notwendig ist.

Mit der Version XRechnung_V2408_024.odt wird die **fortlaufende Nummerierung für die Rechnungsnummer** in der tbl-Rechnung separat abgespeichert. Ohne diese Abspeicherung war es möglich, dass beim Löschen einer alten Rechnung die neueren Rechnungen von der Nummerierung her aufrückten. Es änderte sich also plötzlich die Nummer für alle Folgerechnungen.

Bisher erstellte Rechnungen erhalten in jedem Jahr eine einheitliche Endnummer, wenn an dieser Stelle keine Nummer steht. Die Endnummer muss beim Update von älteren Versionen als XRechnung_V2408_024.odt in dem Feld "ReNummer_lfdJahr" nachgetragen werden, damit die Nummerierung wieder stimmt. Es handelt sich dabei um eine einfache fortlaufende Nummerierung, die jedes Jahr neu beginnt und bisher anhand des Feldes "ID" und des Jahres des Rechnungsdatensatzes ermittelt wurde.

Die Vorlage für die XRechnung ist ab XRechnung_V2408_033.odt in der Tabelle «Rechnungsinhalt» nur mit einer einzigen Spalte versehen. Die Spalten werden dort erst durch ein Makro erstellt. Deshalb müssen existierende Vorlagen hier angepasst werden oder eine neue Vorlage genutzt werden.

Erforderliche Einträge in der XRechnung

Die folgende Tabelle listet alle Felder auf, die für eine XRechnung⁶ erforderlich sind. In der ersten Spalte die Vorgaben aus der Spezifikation für die XRechnung 3.0.1, veröffentlicht 23.09.2023. In der 2. Spalte dann die UBL-Version für die Erstellung einer XML-Datei. Diese Version wird auch zum Erstellen der XRechnung in dieser Datenbank genutzt. In der 3. Spalte die zweite erlaubte XML-Version, hier mit dem Kürzel CII bezeichnet.

Die Spalte "Datenbank" zeigt auf, in welcher Tabelle oder Ansicht die entsprechenden Felder in der Datenbank zu finden sind. Ist das Feld bei der Datenbank leer, so ist der Code für die XRechnung direkt im Makro eingegeben.

XRechnung	XML-Version UBL	XML-Version CII	Datenbank
INVOICE	Invoice	CrossIndustryInvoice	
Invoice number [BT-1]	ID	ExchangedDocument → ID	"viw_Rechnung"."Rechnungsnummer" bzw. "tbl_Rechnung"."Datum" + "tbl_Rechnung"."ReNummer_lfdJahr"
Invoice issue date [BT-2]	IssueDate	ExchangedDocument → IssueDateTime	"tbl_Rechnung"."Datum"
Invoice type code [BT-3]	InvoiceTypeCode	ExchangedDocument → TypeCode	tbl_Rechnung"."Rechnungstyp_Code_ID" ('380': Handelsrechnung)
Invoice currency code [BT-5]	DocumentCurrencyCode	ApplicableHeaderTradeSettlement → InvoiceCurrencyCode	Macro Xrechnung → SaveXRechnung ('EUR': Euro)
Buyer reference [BT-10]	BuyerReference	BuyerReference	"tbl_Rechnung"."KundeZeichen"
PROCESS CONTROL [BG-2]	Direkt auf Invoice-Ebene	ExchangedDocument-Context	
Business process type [BT-23]	ProfileID	BusinessProcessSpecifiedDocumentContextParameter → ID	Macro Xrechnung → SaveXRechnung 'urn:fdc:peppol.eu:2017:poacc:billing:01:1.0'
Specification identifier [BT-24]	CustomizationID	GuidelineSpecifiedDocumentContextParameter → ID	Macro Xrechnung → SaveXRechnung 'urn:cen.eu:en16931:2017#compliant#urn:xeinkauf.de:kosit:xrechnung_3.0'
SELLER [BG-4]	AccountingSupplier-Party	SellerTradeParty	
Seller name [BT-27]	PartyLegalEntity → RegistrationName	Name	"tbl_Firma"."Firma"
Seller electronic address [BT-34]	EndpointID schemeID="EM"	URIUniversalCommunication → URIID schemeID="EM"	"tbl_Firma"."E-Mail"
SELLER POSTAL ADDRESS [BG-5]	PostalAddress	PostalTradeAddress	
Seller city [BT-37]	CityName	CityName	"tbl_Firma"."Ort"
Seller postcode [BT-38]	PostalZone	PostcodeCode	"tbl_Firma"."Plz"
Seller country code [BT-40]	IdentificationCode	CountryID	Macro Xrechnung → SaveXRechnung ('DE': Deutschland)

6 Siehe <https://xeinkauf.de/dokumente/> → dort nach XRechnung suchen...

XRechnung	XML-Version UBL	XML-Version CII	Datenbank
SELLER CONTACT [BG-6]	Contact	DefinedTradeContact	
Seller contact point [BT-41]	Name	PersonName	"tbl_Mitarbeiter"."Name"
Seller contact telephone number [BT-42]	Telephone	TelephoneUniversalCom- munication → Complete- Number	"tbl_Mitarbeiter"."Telefon" "tbl_Firma"."Telefon"
Seller contact email address [BT-43]	ElectronicMail	EmailURIUniversalCommu- nication → URIID	"tbl_Mitarbeiter"."E-Mail"
BUYER [BG-7]	AccountingCustomer- Party	BuyerTradeParty	
Buyer name [BT-44]	PartyLegalEntity → Regis- trationName	Name	"tbl_Kunde"."Firma"
Buyer electronic address [BT-49]	EndpointID schemeID="EM"	URIUniversalCommunica- tion → URIID schemeID="EM"	"tbl_Kunde"."E-Mail"
BUYER POSTAL ADDRESS [BG-8]	PostalAddress	PostalTradeAddress	
Buyer city [BT-52]	CityName	CityName	"tbl_Kunde"."Ort"
Buyer postcode [BT-53]	PostalZone	PostcodeCode	"tbl_Kunde"."Plz"
Buyer country code [BT-55]	IdentificationCode	CountryID	"tbl_Kunde"."Land"
PAYMENT INSTRUCTI- ONS⁷ [BG-16]	PaymentMeans	SpecifiedTradeSettle- mentPaymentMeans	
Payment means type code [BT-81]	PaymentMeansCode	TypeCode	Macro Xrechnung → SaveXrechnung ('58': nur IBAN (innerhalb des SEPA-Raumes) oder '59': Abbuchungsermächti- gung)
DOCUMENT TOTALS [BG-22]	LegalMonetaryTotal	SpecifiedTradeSettle- mentHeaderMonetary- Summation	
Sum of Invoice line net amount [BT-106]	LineExtensionAmount	LineTotalAmount	"viw_Rechnung_Summe"." Netto" (Summe Rechnungszeilen)
Invoice total amount without VAT [BT-109]	TaxExclusiveAmount	TaxBasisTotalAmount	"viw_Rechnung_Summe"." Netto" (Gesamt)
Invoice total amount with VAT [BT-112]	TaxInclusiveAmount	GrandTotalAmount	"viw_Rechnung_Summe"." Brutto" (Gesamt)
Amount due for payment [BT-115]	PayableAmount	DuePayableAmount	"viw_Rechnung_Summe"." NochOffen" (Gesamt)
VAT⁸ BREAKDOWN (1... n) [BG-23]	TaxTotal	ApplicableTradeTax	
VAT category taxable amount [BT-116]	TaxSubtotal → TaxableA- mount	BasisAmount	"viw_Rechnung_Summe"." Netto" (Summe Steuer- satz)

7 Es wird ein Verzeichnis zur Zahlungsvereinbarung erstellt. Es wird aber nicht zur Bedingung gemacht, welche Form der Zahlung erscheinen soll. In der Datenbank ist die Banküberweisung als einzige Zahlungsform implementiert.

8 VAT: Value Added Tax → Mehrwertsteuer

XRechnung	XML-Version UBL	XML-Version CII	Datenbank
VAT category tax amount [BT-117]	TaxSubtotal → TaxAmount	CalculatedAmount	"viw_Rechnung_Summe"."Steuer" (Summe Steuersatz)
VAT category code [BT-118]	TaxSubtotal → TaxCategory → ID	CategoryCode	"viw_Rechnung_Summe"."Steuer_Code_ID" (Summe Steuersatz)
VAT category rate [BT-119]	TaxSubtotal → TaxCategory → Percent	RateApplicablePercent	"viw_Rechnung_Summe"."Steuersatz" (Summe Steuersatz)
INVOICE LINE (1...n) [BG-25]	InvoiceLine	IncludedSupplyChain-TradeLineItem	
Invoice line identifier [BT-126]	ID	AssociatedDocumentLine-Document → LineID	"viw_Rechnung_Ausgang"."RechnungAusgang_ID"
Invoiced quantity [BT-129]	InvoicedQuantity	BilledQuantity	"viw_Rechnung_Ausgang"."Anzahl"
Invoiced quantity unit of measure code [BT-130]	unitCode="XPP"	unitCode="XPP"	"viw_Rechnung_Ausgang"."Anzahl_Code_ID" (Auch die deutschsprachige Bezeichnung in "Name_DE")
Invoice line net amount [BT-131]	LineExtensionAmount	SpecifiedTradeSettlement-LineMonetarySummation → LineTotalAmount	"viw_Rechnung_Ausgang"."AnzahlPreis"
PRICE DETAILS [BG-29]	Price	NetPriceProductTrade-Price	
Item net price [BT-146]	PriceAmount	ChargeAmount	"viw_Rechnung_Ausgang"."Preis"
LINE VAT INFORMATION [BG-30]	ClassifiedTaxCategory	ApplicableTradeTax	
Invoiced item VAT category code ⁹ [BT-151]	ID	CategoryCode	"viw_Rechnung_Ausgang"."Steuer_Code_ID"
ITEM INFORMATION [BG-31]	Item	SpecifiedTradeProduct	
Item name [BT-153]	Name	Name	"viw_Rechnung_Ausgang"."Ware"

Diese Liste enthält nur die Pflichteinträge, die in jeder XRechnung vorhanden sein müssen. Die ursprüngliche Spezifikation für UBL definiert etwas weniger Pflichteinträge, die Spezifikation für CII deutlich weniger Einträge. XRechnungen sind aber nur gültig, wenn sie die notwendigen Felder von XRechnung enthalten. Andere Rechnungen gibt die beigefügte Datenbank nicht aus und liest sie auch nicht ein.

Die Anordnung und Untergliederung der UBL-Version ist deutlich näher an der XRechnung orientiert als die CII-Version. Bei der CII-Version sind Einträge zu einigen Daten völlig anderen Einträgen untergeordnet. Aus dem Grund erstellt die beigefügte Datenbank nur das UBL-Format, kann aber neben dem UBL-Format auch das CII-Format einlesen.

Erst mit der XRechnung-Extension ist es möglich, einer Rechnungszeile untergeordnete Zeilen hinzuzufügen. Diese Zeilen sind in dem Formularen «Kunde» und «Lieferant» in den Tabellenkontrollfeldern ganz unten im Formular verzeichnet. Die XRechnung-Extension wird zur Zeit nur von der XML-Version UBL unterstützt.

9 Das Feld für die prozentuale Angabe der Steuer in der Rechnungszeile ist nicht Pflicht. Die prozentuale Angabe ist nur in der Steuerangabe für die Gesamtrechnung Pflicht. Aus dem Code für die Steuer (S für den Standard-Steuersatz) lässt sich ggf. der entsprechende Prozentsatz über [BT-118] und [BT-119] ermitteln.

✓ Hinweis

XRechnungen können im Internet z.B. über <https://erechnungsv validator.service-bw.de/> auf ihre Korrektheit hin überprüft werden. Der Prüfbericht gibt bei einer XRechnung, die die XRechnung-Extension mit untergeordneten Rechnungszeilen nutzt, zur Zeit noch eine Warnung aus:

Konformitätsprüfung: Das geprüfte Dokument enthält 0 Fehler / 1 Warnungen. Es ist nicht konform zu den formalen Vorgaben.

Übersicht der Validierungsergebnisse:

Prüfschritt	Fehler	Warnungen	Informationen
XML Schema for UBL 2.1 Invoice (val-xsd)	0	0	0
Schematron rules for EN16931 (UBL) (val-sch.1)	0	1	0
Schematron rules for Invoice - XRechnung (UBL) (val-sch.2)	0	0	0
(val-xml)	0	0	0

Validierungsergebnisse im Detail:

Pos	Code	Adj. Grad	Text
val-sch.1.1	UBL-CR-646	warning	[UBL-CR-646]-A UBL invoice should not include the InvoiceLine SubInvoiceLine Pfad: /ubl:Invoice

Bewertung: Es wird empfohlen das Dokument anzunehmen und weiter zu verarbeiten.

Die SubInvoiceLine ist Hauptbestandteil der XRechnung-Extension. Die Warnung erscheint einfach deshalb, weil dieser Bestandteil noch nicht in den erstellten Validator übernommen wurde. Der Inhalt innerhalb der SubInvoiceLine wird deshalb nicht weiter überprüft. Entscheidend ist: Die eigentliche XRechnung passt und wird zur weiteren Verarbeitung empfohlen.

! Vorsicht

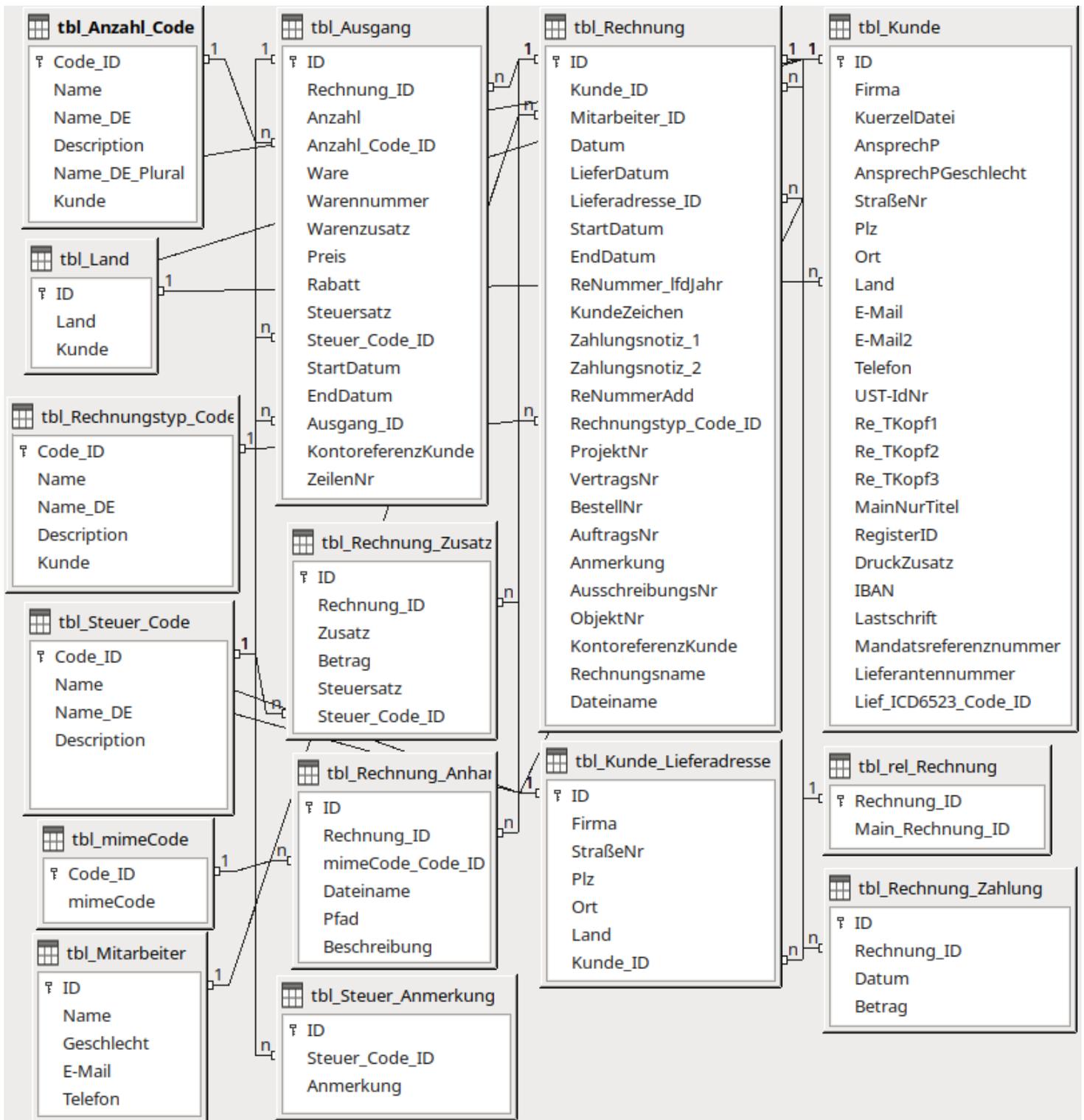
Der Validator lässt leider auch Rechnungen passieren, bei denen aus den oben genannten Pflichtfeldern Felder fehlen. Er gibt dann lediglich eine Warnung aus. So ist z.B. die **Buyer electronic address** (E-Mail des Käufers) ein Pflichtfeld. Eine bekannte Buchhaltungssoftware wirft aber genau solche XRechnungen aus – ohne Mailadresse. Da die Pflichtfelder in dieser Datenbank auch über die Tabellenkonstruktion fest verankert sind, ist so eine Rechnung über das Liefermodul leider nicht einlesbar.

Tabellen

Die Tabellenstruktur der Datenbank ist darauf ausgelegt, dass XRechnungen erstellt und XRechnungen ausgelesen werden können. Darüber hinaus sollen noch normale Papierrechnungen erstellt werden können. Dies stellt nur einen Rahmen für eine Komplettlösung dar, bei der eingehende Waren und ausgehende Waren miteinander verbunden würden und auch eine Kas senführung Bestandteil des Pakets wäre.

Rechnungserstellung im Kundenbereich

Für den Kundenbereich sind neben "tbl_Kunde" und "tbl_Rechnung" weitere Tabellen notwendig, um die Detailinformationen zu speichern. Grundlage aller relationalen Datenbank ist, dass sich Inhalte von Feldern in einer Tabelle möglichst selten wiederholen. Würde alles in eine Tabelle gepackt, so müsste für jede Rechnung Name und Adresse für den Kunden neu eingegeben werden, auch wenn der Kunde bereits eine Rechnung bezogen hat. Würde die Rechnung nicht von den Rechnungsinhalten in "tbl_Ausgang" getrennt, so müsste zu jedem Inhalt Datum, Kundenzeichen und Rechnungsnummer neu eingegeben werden. Deswegen auch hier eine Trennung zwischen "tbl_Rechnung" und "tbl_Ausgang".



Der Zusammenhang zwischen Tabellen wird über Extras → Relationen sichtbar. Hier die Tabellen, die die Rechnungserstellung ermöglichen.

tbl_Kunde

Datenziel
Ansicht: <i>viw_Rechnung, viw_Lieferung_Aenderung, viw_Lieferung_Eingang_Aenderung</i>
Abfrage: <i>qry_frmZusatz-Kunde</i>
Formular: <i>frm_Rechnung, frm_Lieferung_Aenderung, frm_Lieferadresse</i>
Makro: <i>Import_all, Import_Values, TableContentDeleteAll, TableRestart</i>

Feldname	Feldtyp	Beschreibung
ID	Integer	Primärschlüssel der Tabelle. Der Primärschlüssel muss eindeutig sein. Das Feld ist als Auto-Wert-Feld gesetzt. [BT-46]
Firma	Text	Der Name der Firma/Person Eingabe erforderlich: Ja [BT-44] ¹⁰
KuerzelDatei	Text(10)	Firmenkürzel, mit der der Dateiname beim Abspeichern versehen wird – maximal 10 Zeichen, Umlaute vermeiden Eingabe erforderlich: Nein
AnsprechP	Text	Der Name der Ansprechperson in der Firma Eingabe erforderlich: Nein
AnsprechPGeschlecht	Text	Nur für die richtige Anrede der Ansprechperson – lediglich ein Buchstabe Eingabe erforderlich: Nein
StraßeNr	Text	Straße und Hausnummer der Firma/Person Eingabe erforderlich: Ja (obwohl für XRechnung nicht erforderlich) [BT-50]
Plz	Text	Postleitzahl für Waren an die Firma/Person Eingabe erforderlich: Ja [BT-53]
Ort	Text	Lieferort für Waren an die Firma/Person Eingabe erforderlich: Ja [BT-52]
Land	Text	Landesangabe für Waren an die Firma/Person – 2 Buchstaben Fremdschlüssel aus Tabelle "tbl_Land"."ID" Eingabe erforderlich: Ja [BT-55]
E-Mail	Text	E-Mail-Adresse der Firma zur Zusendung der XRechnung Eingabe erforderlich: Ja [BT-49]
E-Mail2	Text	Zusätzliche E-Mail-Adresse der Firma zur Zusendung der XRechnung, erscheint bei der Mail im CC Eingabe erforderlich: Nein
Telefon	Text	Telefonkontakt zu der Firma Eingabe erforderlich: Nein
UST-IdNr	Text	Angabe zur Umsatzsteuernummer Eingabe erforderlich: Nein [BT-48]
Re_TKopf1	Text	Beschriftung für den Inhalt, der standardmäßig als «Beschreibung» im Tabellenkopf der Rechnung auftaucht. Überschreibt den Begriff und auch den Begriff «Ware*» in den Eingabefeldern des Kundenformulars. Eingabe erforderlich: Nein [BT-127]

¹⁰ Den Beschreibungen ist die jeweilige ID aus der Beschreibung zur XRechnung in [] hinzugefügt. Felder mit diesem Vermerk werden auch zur XRechnung exportiert.

Feldname	Feldtyp	Beschreibung
Re_TKopf2	Text	Beschriftung für den Inhalt, der standardmäßig als «Bestellnr.» im Tabellenkopf der Rechnung auftaucht. Überschreibt den Begriff und auch den Begriff «Warennummer» in den Eingabefeldern des Kundenformulars. Eingabe erforderlich: Nein [BT-127]
Re_TKopf3	Text	Beschriftung für den Inhalt, der standardmäßig als «Zusatz» im Tabellenkopf der Rechnung auftaucht. Überschreibt den Begriff und auch den Begriff «Warenzusatz» in den Eingabefeldern des Kundenformulars. Eingabe erforderlich: Nein [BT-127]
MainNurTitel	Ja/Nein	Im Tabellenkontrollfeld «Rechnungsposten» des Kundenformulars wird nur das Feld «Titel*» angeboten, eventuell auch noch die Steuereingabe. Das Feld dient dazu, die Rechnungsinhalte aus «Untergliederung Rechnungsposten» jeweils mit einer Überschrift zu versehen. Die XRechnung ist davon nicht berührt. Defaultwert: False (Nein) Eingabe erforderlich: Nein
RegisterID	Text	Manche Rechnungsempfänger benötigen neben der Angabe der UmsatzsteuerID auch die Angabe ihrer nationalen Identifikationsnummer. Für so eine Nummer ist dieses Feld vorgesehen. Eingabe erforderlich: Nein [BT-47]
DruckZusatz	Text	In manchen Ausdrucken benötigen Kunden ganz bestimmte Informationszusammenhänge. Dieses Feld nimmt solche Einträge auf, wird aber nur in den Ausdruck, nicht in die XRechnung, übernommen. Eingabe erforderlich: Nein
IBAN	Text	IBAN des Kunden, für Lastschrift erforderlich Eingabe erforderlich: Nein [BT-91]
Lastschrift	Ja/Nein	Abbuchungen per Lastschrift? Eingabe erforderlich: Nein
Mandatsreferenznummer	Text	Für Lastschrift erforderlich, wird durch Bankingsoftware ermittelt Eingabe erforderlich: Nein [BT-89]
Lieferantennummer	Text	Kennung der eigenen Firma beim Kunden Eingabe erforderlich: Nein [BT-29]
Lief_ICD6523_-Code_ID	Text (4)	Code für die Lieferantennummer, zusammen mit Lieferantennummer ggf. erforderlich Eingabe erforderlich: Nein

tbl_Kunde_Lieferadresse

Datenziel
Ansicht: viw_Lieferung_Aenderung
Formular: frm_Rechnung , frm_Lieferung , frm_Lieferung_Aenderung , frm_Lieferadresse
Makro: Import_Values , TableRestart

Feldname	Feldtyp	Beschreibung
ID	Integer	Primärschlüssel der Tabelle. Der Primärschlüssel muss eindeutig sein. Das Feld ist als Auto-Wert-Feld gesetzt.

Feldname	Feldtyp	Beschreibung
Firma	Text	Der Name der Firma/Person Eingabe erforderlich: Ja [BT-70]
StraßeNr	Text	Straße und Hausnummer der Firma/Person Eingabe erforderlich: Ja (obwohl für XRechnung nicht erforderlich) [BT-75]
Plz	Text	Postleitzahl für Waren an die Firma/Person Eingabe erforderlich: Ja [BT-78]
Ort	Text	Lieferort für Waren an die Firma/Person Eingabe erforderlich: Ja [BT-77]
Land	Text	Landesangabe für Waren an die Firma/Person - 2 Buchstaben Fremdschlüssel aus Tabelle "tbl_Land"."ID" Eingabe erforderlich: Ja [BT-80]
Kunde_ID	Integer	Fremdschlüssel zur Tabelle "tbl_Kunde" Eingabe erforderlich: Ja

tbl_Rechnung

Datenziel
Ansicht: <i>viw_Rechnung</i>
Abfrage: <i>qry_Rechnung_Zahlung, qry_frmZusatz-Rechnung</i>
Formular: <i>frm_Rechnung</i>
Makro: <i>AnhangPfad, RechnungCopyInsert, Import_all, Import_Values, TableContentDeleteAll, TableRestart</i>

Feldname	Feldtyp	Beschreibung
ID	Integer	Primärschlüssel der Tabelle. Der Primärschlüssel muss eindeutig sein. Das Feld ist als Auto-Wert-Feld gesetzt.
Kunde_ID	Integer	Hiermit wird die Rechnung dem Kunden zugewiesen Fremdschlüssel aus "tbl_Kunde"."ID" Eingabe erforderlich: Ja
Mitarbeiter_ID	Integer	Hiermit wird der Rechnung eine Ansprechperson der Firma zugewiesen Fremdschlüssel aus "tbl_Mitarbeiter"."ID" Eingabe erforderlich: Ja
Datum	Datum	Rechnungsdatum Eingabe erforderlich: Ja [BT-1][BT-2]
KundeZeichen	Text	Zeichen des Kunden (Bestellung?) - bei Behörden die Leitweg-ID; kann auf «-» gesetzt werden, falls kein Zeichen vorhanden ist Eingabe erforderlich: Ja [BT-10]
Zahlungsnotiz_1	Text	Reicht die Standard-Zahlungsnotiz aus "tbl_Firma" nicht aus, so kann hier eine Zahlungsnotiz erstellt werden. Diese Notiz wird dann statt der Standard-Zahlungsnotiz gezeigt. Eingabe erforderlich: Nein [BT-20]
Zahlungsnotiz_2	Text	Ist eine weitere Zahlungsnotiz erforderlich, so wird diese Notiz, durch Zeilenumbruch von der ersten getrennt, in der Rechnung aufgeführt. Eingabe erforderlich: Nein [BT-20]

Feldname	Feldtyp	Beschreibung
ReNum-merAdd	Text	Zusätzlich zur automatisch generierten Rechnungsnummer kann hier eine Nummer angegeben werden, so dass die Rechnungsnummern auch in dem bisherigen Bestand einsortiert werden können. Eingabe erforderlich: Nein [BT-1]
Rechnungs-tyt_Code_ID	Text (3)	Hiermit wird der Rechnung ein Rechnungstyp nach XRechnung-Standard zugewiesen Fremdschlüssel aus "tbl_Rechnungstyp_Code"."Code_ID" Eingabe erforderlich: Ja [BT-3]
LieferDatum	Datum	Das Lieferdatum für die komplette Rechnung. Ist durch den Code 001 ALTER TABLE "tbl_Rechnung" 002 ADD CONSTRAINT "Rechnungsdatum>=Lieferdatum" CHECK ("Datum" >= "LieferDatum") abgesichert, so dass das Lieferdatum auf keinen Fall nach dem Rechnungsdatum liegt. Eingabe erforderlich: Nein [BT-72]
Lieferadres-se_ID	Integer	Hiermit wird der Rechnung eine alternative Lieferadresse zugewiesen. Fremdschlüssel aus "tbl_Kunde_Lieferadresse"."ID" Eingabe erforderlich: Nein
StartDatum	Datum	Das Startdatum für die komplette Rechnung. Ist durch den Code 001 ALTER TABLE "tbl_Rechnung" 002 ADD CONSTRAINT "Enddatum>=Startdatum" CHECK ("EndDatum" >= "StartDatum") abgesichert, so dass das Startdatum auf keinen Fall nach dem Enddatum liegt. Eingabe erforderlich: Nein [BT-73]
EndDatum	Datum	Das Enddatum für die komplette Rechnung. Eingabe erforderlich: Nein [BT-74]
ProjektNr	Text	Projektnummer, auch Freitext Eingabe erforderlich: Nein [BT-11]
VertragsNr	Text	Vertragsnummer, auch Freitext Eingabe erforderlich: Nein [BT-12]
BestellNr	Text	Bestellnummer, auch Freitext Eingabe erforderlich: Nein [BT-13]
AuftragsNr	Text	Auftragsnummer, auch Freitext Eingabe erforderlich: Nein [BT-14]
Ausschrei-bungsNr	Text	Ausschreibungsnummer, auch Freitext Eingabe erforderlich: Nein [BT-17]
ObjektNr	Text	Objektnummer, auch Freitext, (z. B. Zählernummer, Telefonnummer, Kfz-Kennzeichen, versicherte Person, Abonnementnummer, Rufnummer) Eingabe erforderlich: Nein [BT-18]
Anmerkung	Text	Allgemeine Anmerkungen für die gesamte Rechnung Eingabe erforderlich: Nein [BT-22]

Feldname	Feldtyp	Beschreibung
ReNum- mer_lfdJahr	Integer	<p>Sichert die Reihenfolge der Rechnungsnummer für das laufende Jahr ab. Ein Datensatz, der hier einen Eintrag hat, kann nicht gelöscht werden.</p> <p>Rechnungsnummern müssen eindeutig sein und dürfen nicht einfach nachträglich zu ändern sein. Die Ziffer aus "ReNummer_lfdJahr" wird deswegen mit der in "tbl_Firma" vermerkten Stellenzahl an die Jahreszahl angehängt, um eine Rechnungsnummer zu bilden. Durch den folgenden Code wird verhindert, dass ein Datensatz mit eingetragener Nummer gelöscht werden kann:</p> <pre> 001 CREATE EXCEPTION EX_NAME_WICHTIG 002 'Löschen des Datensatzes nicht möglich. "ReNummer_lfdJahr" hat einen Eintrag für die laufende Rechnungsnummer.'; </pre> <p>Zuerst wird ein Text definiert, der bei dem Versuch, den Datensatz zu löschen, erscheinen soll.</p> <pre> 001 CREATE TRIGGER DELETE_TBL_RECHNUNG 002 ACTIVE BEFORE DELETE ON "tbl_Rechnung" 003 AS 004 BEGIN 005 IF (NOT OLD."ReNummer_lfdJahr" IS NULL) THEN 006 EXCEPTION EX_NAME_WICHTIG; 007 END; </pre> <p>Anschließend wird ein Trigger definiert, der dann ausgelöst wird, wenn ein Datensatz in der "tbl_Rechnung" gelöscht werden soll (Zeile 2). Wenn das bestehende Feld "ReNummer_lfdJahr" nicht leer ist (Zeile 5), dann wird der Text aufgerufen und das Löschen abgebrochen.</p> <p>Eingabe erforderlich: Nein [BT-1]</p>

tbl_Ausgang

Datenziel
Ansicht: <i>viw_Rechnung_Ausgang, viw_Rechnung_Spalten</i>
Abfrage: <i>qry_frmAusgang</i>
Formular: <i>frm_Rechnung</i>
Makro: <i>ChangeSubToMain, ChangeMainToSub, RechnungCopyInsert, Import_Values, TableRestart, ValueUnique</i>

Feldname	Feldtyp	Beschreibung
ID	Integer	Primärschlüssel der Tabelle. Der Primärschlüssel muss eindeutig sein. Das Feld ist als Auto-Wert-Feld gesetzt.
Rechnung_ID	Integer	Hiermit wird der Ausgang der Rechnung zugewiesen Fremdschlüssel aus "tbl_Rechnung"."ID" Eingabe erforderlich: Ja
Anzahl	Dezimal (16,8)	Anzahl der Ware, wird bei anderen Maßeinheiten auch mit Nachkommastellen angegeben Eingabe erforderlich: Ja [BT-129]
Anzahl_Co- de_ID	Text (3)	Maßeinheit für die Anzahl der Ware; ist bereits in der Tabelle mit dem Standardwert 'XPP' für die Stückzahl voreingestellt. Kann auch so etwas wie Paletten, Stunden usw. sein. Fremdschlüssel aus "tbl_Anzahl_Code"."Code_ID" Eingabe erforderlich: Ja [BT-130]

Feldname	Feldtyp	Beschreibung
Ware	Text	Ausführliche Beschreibung der Ware auch über mehrere Zeilen (250 Zeichen) Eingabe erforderlich: Ja [BT-127][BT-128]
Warennummer	Text	ID für die Ware, Bestellnummer... Eingabe erforderlich: Nein [BT-126][BT-127]
Warenzusatz	Text	Zusätzliches Eingabefeld, falls Rechnungen 3 Inhaltsspalten benötigen Eingabe erforderlich: Nein [BT-127]
Preis	Dezimal (10,4)	Einzelpreis der Ware; Angabe bis 4 Nachkommastellen möglich, (bei Energierechnungen üblich) Eingabe erforderlich: Ja [BT-131][BT-146]
Rabatt	Dezimal (3,2)	Rabattangabe in Prozent Eingabe erforderlich: Nein [BT-136]
Steuersatz	Dezimal (3,2)	Steuersatz für die Ware, Angabe in Prozent Eingabe erforderlich: Ja (obwohl für XRechnung nicht erforderlich; Steuercode zählt, Angabe Steuersatz in der Gesamtrechnung) [BT-152]
Steuer_Code_ID	Text (3)	Hiermit wird der Steuercode aus XRechnung der Rechnung zugewiesen Fremdschlüssel aus "tbl_Steuer_Code"."Code_ID" Eingabe erforderlich: Ja [BT-151] Standard auf 'S' (normaler Steuersatz) gesetzt.
StartDatum	Datum	Datum für den Rechnungsposten oder Beginn einer Periode Eingabe erforderlich: Nein [BT-134]
EndDatum	Datum	Ende einer Periode Eingabe erforderlich: Nein [BT-135]
Ausgang_ID	Integer	Hiermit wird der Ausgang einem anderen Ausgang untergeordnet (XRechnung-Extension) Fremdschlüssel aus "tbl_Ausgang"."ID" Eingabe erforderlich: Nein
KontoreferenzKunde	Text	An welcher Stelle in den Finanzkonten des Erwerbers ist die Rechnungszeile zu buchen? Muss vom Kunden vor der Rechnungserstellung mitgeteilt sein. Eingabe erforderlich: Nein [BT-133]
ZeilenNr	Text	Die Zeilen müssen für die XRechnung mit einer eindeutigen Kennung versehen werden. Soll dies nicht durch eine Abfrage geschehen, so kann hier eine eindeutige Bezeichnung für jede Zeile einer Rechnung vorgegeben werden. Eingabe erforderlich: Nein, aber: Wenn ein Eintrag in einer Rechnung existiert müssen alle anderen Zeilen auch gekennzeichnet werden. [BT-126]

Im Feld Anzahl wird die Anzahl gespeichert, mit der der Preis multipliziert werden muss, um auf den Gesamtpreis zu kommen.

tbl_Rechnung_Zusatz

Datenziel
Ansicht: <i>viw_Rechnung_Spalten</i>
Formular: <i>frm_Rechnung</i>
Makro: <i>FillTableCarryOver, SaveXRechnung, SaveZUGFeRD, RechnungCopyInsert, Import_Values, TableRestart</i>

Feldname	Feldtyp	Beschreibung
ID	Integer	Primärschlüssel der Tabelle. Der Primärschlüssel muss eindeutig sein. Das Feld ist als Auto-Wert-Feld gesetzt.
Rechnung_ID	Integer	Hiermit wird der Zusatz der Rechnung zugewiesen Fremdschlüssel aus "tbl_Rechnung"."ID" Eingabe erforderlich: Ja
Zusatz	Text	Grund für Ermäßigung oder Aufschlag auf die Rechnung wie «Portokosten», «Rabatt» Eingabe erforderlich: Ja [BT-97][BT-104]
Betrag	Dezimal (8,2)	Bei Aufschlägen positiver Betrag, bei Rabatt negativer Betrag Eingabe erforderlich: Ja [BT-92][BT-99]
Steuersatz	Dezimal (3,2)	Steuersatz für den Zusatz, Angabe in Prozent Eingabe erforderlich: Ja [BT-96][BT-103]
Steuer_Code_ID	Text (3)	Hiermit wird der Steuercode aus XRechnung der Rechnung zugewiesen Fremdschlüssel aus "tbl_Steuer_Code"."Code_ID" Eingabe erforderlich: Ja [BT-95][BT-102] Standard auf 'S' (normaler Steuersatz) gesetzt.

Bei dem Zusatz wird in der XRechnungen zwischen Nachlässen und Aufschlägen unterschieden. Die Unterscheidung erfolgt hier über den eingetragenen Betrag. Die entsprechenden ID-Nummern aus der XRechnung beziehen sich zuerst auf einen Nachlass, dann auf einen Aufschlag.

tbl_Rechnung_Anhang

Datenziel
Formular: <i>frm_Rechnung</i>
Makro: <i>FillTableCarryOver, SaveXRechnung, SaveZUGFeRD, RechnungCopyInsert, Import_Values, TableRestart</i>

Feldname	Feldtyp	Beschreibung
ID	Integer	Primärschlüssel der Tabelle. Der Primärschlüssel muss eindeutig sein. Das Feld ist als Auto-Wert-Feld gesetzt.
Rechnung_ID	Integer	Hiermit wird der Zusatz der Rechnung zugewiesen Fremdschlüssel aus "tbl_Rechnung"."ID" Eingabe erforderlich: Ja
mimeCode_Code_ID	Text(4)	Fremdschlüssel aus "tbl_mimeCode"."Code_ID" Eingabe erforderlich: Ja [BT-125] (Mimecode Binärdatei)

Feldname	Feldtyp	Beschreibung
Dateiname	Text	Dateiname der eingelesenen Datei - wird automatisch im Formular eingetragen Eingabe erforderlich: Nein (obwohl in der XRechnung erforderlich in der Tabelle erst einmal nicht erforderlich. Der Dateiname wird über das Formular beim Speichern nachgetragen.) [BT-122]
Pfad	Text	Pfad zu der eingelesenen Datei im Binärformat. Die Datei wird in einem grafischen Kontrollfeld angezeigt. Eingabe erforderlich: Nein [BT-125] (Binärdatei)
Beschreibung	Text	Kurze Beschreibung des Anhangs Eingabe erforderlich: Nein [BT-123]

tbl_rel_Rechnung

Datenziel
Formular: frm_Zusatz
Makro: ListenfeldfilterRechnungsreferenz , Import_Values

Feldname	Feldtyp	Beschreibung
Rechnung_ID	Integer	Hier kann eine Rechnung aus der "tbl_Rechnung"."ID" gesucht werden. Jeder Rechnung kann nur eine Hauptrechnung zugeordnet werden. Deshalb ist dieses Feld der Primärschlüssel Eingabe erforderlich: Ja
Main_Rechnung_ID	Integer	Auch dieses Feld bezieht sich auf "tbl_Rechnung"."ID". Dies ist dann die Rechnung, der die Rechnung aus dem vorhergehenden Feld untergeordnet ist. Das Feld ist für die Sammelrechnung gedacht. Eingabe erforderlich: Ja [BT-20][BT-25][BT-26]

tbl_Rechnung_Zahlung

Datenziel
Abfrage: qry_frmKontoRechnung
Makro: FillTableCarryOver , Import_all , Import_Values , TableContentDeleteAll , TableRestart

Feldname	Feldtyp	Beschreibung
ID	Integer	Primärschlüssel der Tabelle. Der Primärschlüssel muss eindeutig sein. Das Feld ist als Auto-Wert-Feld gesetzt.
Rechnung_ID	Integer	Hiermit wird die Zahlung der Rechnung zugewiesen Fremdschlüssel aus "tbl_Rechnung"."ID" Eingabe erforderlich: Ja [BT-20]
Datum	Datum	Datum für den Eingang der Zahlung Eingabe erforderlich: Ja [BT-20]
Betrag	Dezimalzahl	Höhe des eingegangenen Betrags Eingabe erforderlich: Ja [BT-20]

Vorausgegangene Zahlungen müssen bei der Abschlussrechnung berücksichtigt werden. Sie werden auch in der Zahlungsnotiz erwähnt, auf die sich die beigefügte XRechnung-ID jeweils bezieht.

tbl_Anzahl_Code

Datenziel
Ansicht: viw_Rechnung_Ausgang , viw_Lieferung_Eingang_Aenderung
Abfrage: qry_Maßeinheit_Listbox

Feldname	Feldtyp	Beschreibung
Code_ID	Text (3 Zeichen)	Primärschlüssel der Tabelle. Der Primärschlüssel muss eindeutig sein. Das Feld ist durch den Code für die Maßeinheit der Anzahl ¹¹ bei XRechnungen vorgegeben.
Name	Text	Bezeichnung des Codes Eingabe erforderlich: Ja
Name_DE	Text	Deutschsprachige Übersetzung der Bezeichnung Eingabe erforderlich: Nein
Description	Text (250 Zeichen)	Beschreibung des Codes Eingabe erforderlich: Nein
Name_-DE_Plural	Text	Mehrzahl für die deutschsprachige Übersetzung, wird für die ausdrückbare Rechnungserstellung bei "Anzahl" > 1 genutzt, sofern vorhanden. Eingabe erforderlich: Nein
Kunde	Ja/Nein	Hier erfolgt eine Codes, die für Rechnungen der Firma benötigt werden. Ausgewählte Codes werden im Listenfeld oben positioniert. Als Standard wird durch Abfrage 'Stück' angeboten. Eingabe erforderlich: Nein

Die Tabelle ist mit Inhalten aus den XRechnung-Vorgaben in der aktuellen Version 3 (Gültigkeit ab 18.06.2021) gefüllt. Die deutschsprachige Übersetzung wurde mittels DEEPL hinzugefügt und nur teilweise angepasst.

Den gleichen Aufbau haben die Tabellen "tbl_Rechnungstyp_Code", "tbl_Steuer_Code", "tbl_Zusatz_Code" (Ermäßigungen und Zuschläge) und "tbl_ICD6523_Code".

Der Zusatz-Code wird zur Zeit nicht genutzt, da er nach XRechnung-Standard nicht erforderlich ist.

tbl_Land

Datenziel
Ansicht: viw_Rechnung , viw_Lieferung_Aenderung
Abfrage: qry_Land_Listbox

Feldname	Feldtyp	Beschreibung
ID	Text	Primärschlüssel der Tabelle. Der Primärschlüssel muss eindeutig sein. Hier steht das Landeskürzel aus XRechnung
Land	Text	Der ausgeschriebene Landesname, erforderlich bei Rechnungen, die ins Ausland gehen Eingabe erforderlich: Ja

¹¹ https://www.xrepository.de/details/urn:xoev-de:kosit:codeliste:rec21_3

Feldname	Feldtyp	Beschreibung
Kunde	Ja/Nein	Hier erfolgt eine Vorauswahl der Länder, in die Rechnungen der Firma gehen. Ausgewählte Länder werden im Listenfeld oben positioniert. Als Standard wird durch Abfrage 'Deutschland' angeboten. Eingabe erforderlich: Nein

Die Tabelle ist mit Inhalten aus den XRechnungs-Vorgaben in der aktuellen Version 3 (Gültigkeit ab 18.06.2021) gefüllt.

tbl_mimeCode

Datenziel
Formular: frm_Rechnung , frm_Lieferung
Makro: SaveXRechnung , SaveZUGFeRD

Feldname	Feldtyp	Beschreibung
Code_ID	Text (4 Zeichen)	Primärschlüssel der Tabelle. Der Primärschlüssel muss eindeutig sein. Das Feld enthält die zulässigen Dateieindungen.
mimeCode	Text	Bezeichnung des Codes Eingabe erforderlich: Ja

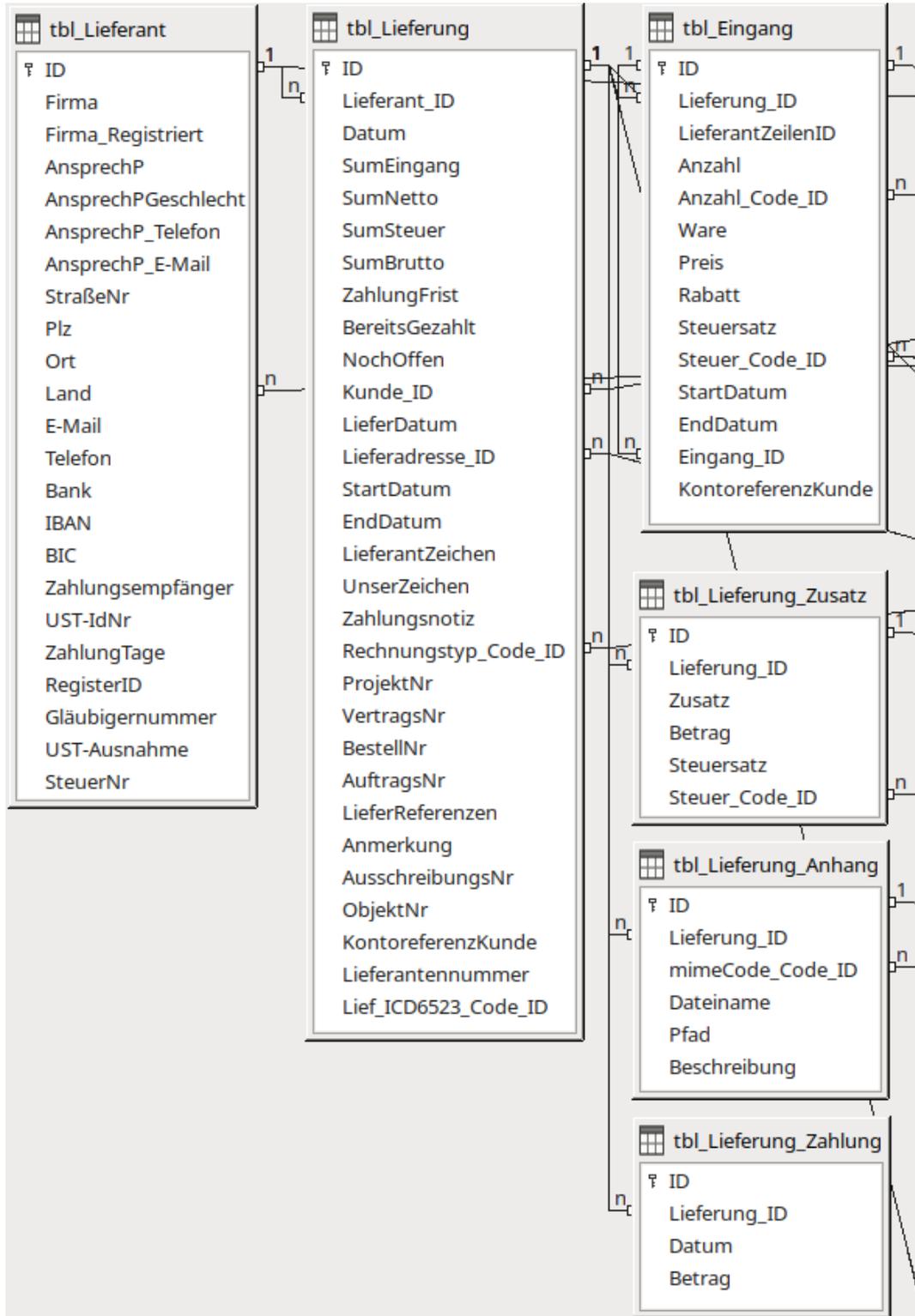
Die Tabelle ist mit Inhalten aus den XRechnungs-Vorgaben in der aktuellen Version 3 (Gültigkeit ab 18.06.2021) gefüllt.

tbl_Mitarbeiter

Datenziel
Ansicht: viw_Rechnung , viw_Lieferung_Aenderung
Formular: frm_Einstellungen
Makro: Import_all , Import_Values , TableContentDeleteAll , TableRestart

Feldname	Feldtyp	Beschreibung
ID	Integer	Primärschlüssel der Tabelle. Der Primärschlüssel muss eindeutig sein. Das Feld ist als Auto-Wert-Feld gesetzt.
Name	Text	Der Name Person Eingabe erforderlich: Ja [BT-41]
Geschlecht	Text	Geschlecht der Person Eingabe erforderlich: Nein
E-Mail	Text	E-Mail-Kontakt zu der Person Eingabe erforderlich: Ja [BT-43]
Telefon	Text	Telefonkontakt zu der Person Eingabe erforderlich: Ja [BT-42]

Eingang von Waren über Lieferanten



Hier ist nur der Block für die Lieferanten abgebildet. "tbl_mimeCode", "tbl_Steuer_Code", "tbl_Anzahl_Code", "tbl_Land", "tbl_Rechnungstyp_Code" und "tbl_Kunde" sind bereits bei dem Block für die Kunden aufgeführt.

tbl_Lieferant

Datenziel
Formular: <i>frm_Lieferung</i>
Ansicht: <i>viw_Lieferung_Aenderung</i>
Abfrage: <i>qry_Konto, qry_Lieferung_Zahlung, qry_frmZusatz-Lieferant</i>
Makro: <i>Import, Import_all, Import_Values, TableContentDeleteAll, TableRestart</i>

Feldname	Feldtyp	Beschreibung
ID	Integer	Primärschlüssel der Tabelle. Der Primärschlüssel muss eindeutig sein. Das Feld ist als Auto-Wert-Feld gesetzt.
Firma	Text	Der Name der Firma/Person Eingabe erforderlich: Ja [BT-27]
Firma-Registriert	Text	Registrierter Name der Firma - kann anders als der Rufname sein Eingabe erforderlich: Ja [BT-28]
AnsprechP	Text	Der Name der Ansprechperson in der Firma Eingabe erforderlich: Ja [BT-41]
AnsprechPGeschlecht	Text	Nur für die richtige Anrede der Ansprechperson - lediglich ein Buchstabe Eingabe erforderlich: Nein
AnsprechP_E-Mail	Text	E-Mail-Adresse der Ansprechperson Eingabe erforderlich: Ja [BT-43]
AnsprechP_Telefon	Text	Telefonnummer der Ansprechperson Eingabe erforderlich: Ja [BT-42]
StraßeNr	Text	Straße und Hausnummer der Firma/Person Eingabe erforderlich: Ja (obwohl für XRechnung nicht erforderlich) [BT-35]
Plz	Text	Postleitzahl für Waren von der Firma/Person Eingabe erforderlich: Ja [BT-38]
Ort	Text	Lieferort für Waren von der Firma/Person Eingabe erforderlich: Ja [BT-37]
Land	Text	Landesangabe für Waren von der Firma/Person - 2 Buchstaben Fremdschlüssel aus Tabelle "tbl_Land" Eingabe erforderlich: Ja [BT-40]
E-Mail	Text	E-Mail-Adresse der Firma zur Zusendung der XRechnung Eingabe erforderlich: Ja [BT-34]
Telefon	Text	Telefonkontakt zu der Firma Eingabe erforderlich: Ja (ggf. Nummer des Ansprechpartners) [BT-42]
Bank	Text	Name der Bank Eingabe erforderlich: Nein
IBAN	Text	IBAN Eingabe erforderlich: Ja [BT-84]
BIC	Text	BIC Eingabe erforderlich: Nein [BT-86]
Zahlungsempfänger	Text	Empfänger der Zahlung bei der Bank Eingabe erforderlich: Nein [BT-85]

Feldname	Feldtyp	Beschreibung
UST-IdNr	Text	Umsatzsteuernummer Eingabe erforderlich: Ja [BT-31]
ZahlungTage	Integer	Zahlungsfrist in Tagen nach Rechnungsstellung Eingabe erforderlich: Nein
RegisterID	Text	Manche Rechnungsempfänger benötigen neben der Angabe der UmsatzsteuerID auch die Angabe ihrer nationalen Identifikationsnummer. Für so eine Nummer ist dieses Feld vorgesehen. Eingabe erforderlich: Nein [BT-47]
Gläubiger-nummer	Text	Für Lastschriften ist die Gläubigernummer der Firma erforderlich, damit sie die Lastschriften einziehen kann Eingabe erforderlich: Nein [BT-90]
UST-Aus-nahme	Text	Kleinunternehmer sind von der Umsatzsteuerangabe befreit. Dann ist hier ein Eintrag erforderlich. Eingabe erforderlich: Nein [BT-33]
SteuerNr	Text	Normale Steuernummer, wenn Kleinunternehmer Eingabe erforderlich: Nein [BT-32]

Es wird davon ausgegangen, dass die Nutzung von IBAN die Standardnutzung ist. Das Feld wird auch beim Einlesen aus der XRechnung vorausgesetzt. Hier könnten auch andere Zahlungsweisen implementiert werden.

Es wird nur eine Ansprechperson aufgeführt. Das kann natürlich dazu führen, dass bei einem Wechsel der Ansprechperson die neue Person nicht übernommen wird. Es ist zur Zeit auch nur möglich, pro Firma eine Ansprechperson auf zu nehmen. An bestimmte Rechnungen gebundene Ansprechpersonen sind so nicht möglich.

tbl_Lieferung

Datenziel
Formular: <i>frm_Lieferung</i>
Abfrage: <i>qry_Konto, qry_Lieferung_Zahlung, qry_frmZusatz-Lieferung</i>
Makro: <i>Import, Import_to_Aenderung, Import_all, Import_Values, TableContentDeleteAll, TableRestart</i>

Feldname	Feldtyp	Beschreibung
ID	Integer	Primärschlüssel der Tabelle. Der Primärschlüssel muss eindeutig sein. Das Feld ist als Auto-Wert-Feld gesetzt.
Lieferant_ID	Integer	Hiermit wird der Lieferant der Lieferung zugewiesen Fremdschlüssel aus "tbl_Lieferant"."ID" Eingabe erforderlich: Ja
Datum	Datum	Datum der Rechnung Eingabe erforderlich: Ja [BT-2]
LieferantZeichen	Text	Rechnungsnummer oder Liefernummer Eingabe erforderlich: Ja [BT-1]
UnserZeichen	Text	Bestellzeichen oder ähnliches Eingabe erforderlich: Nein [BT-10]
ZahlungFrist	Datum	Lieferort für Waren an die Firma/Person Eingabe erforderlich: Nein [BT-9]
Zahlungsnotiz	Text	Notizen auf der Rechnung zur Zahlungsweise Eingabe erforderlich: Nein [BT-20]

Feldname	Feldtyp	Beschreibung
SumEingang	Dezimal (8,2)	Summe aus "tbl_Eingang" für die Lieferung Eingabe erforderlich: Nein [BT-106]
SumNetto	Dezimal (8,2)	Summe "tbl_Eingang" und "tbl_Lieferung_Zusatz" für die Lieferung Eingabe erforderlich: Nein [BT-109]
SumSteuer	Dezimal (8,2)	Summe der Steuer insgesamt Eingabe erforderlich: Nein [BT-110]
SumBrutto	Dezimal (8,2)	Brutto insgesamt Eingabe erforderlich: Nein [BT-112]
BereitsGezahlt	Dezimal (8,2)	Bisher erfolgte Zahlen als Abschlagszahlungen oder Raten Eingabe erforderlich: Nein [BT-113]
NochOffen	Dezimal (8,2)	Offener Betrag nach dieser Lieferung (Brutto - BereitsGezahlt) Eingabe erforderlich: Nein [BT-115]
Kunde_ID	Integer	Hiermit wird ein Kunde der Lieferung zugewiesen, sofern die Lieferung zur Weiterleitung vorgesehen sein soll. Fremdschlüssel aus "tbl_Kunde"."ID" Eingabe erforderlich: Nein
LieferDatum	Datum	Datum der Lieferung (nicht der Rechnung zur Lieferung) Eingabe erforderlich: Nein [BT-72]
Lieferadresse_ID	Integer	Fremdschlüssel zu "tbl_Kunde_Lieferadresse"."ID" Eingabe erforderlich: Nein
StartDatum	Datum	Startdatum für den angegebenen Zeitraum Eingabe erforderlich: Nein [BT-73]
EndDatum	Datum	Enddatum für den angegebenen Zeitraum Eingabe erforderlich: Nein [BT-74]
Rechnungstyp_Code_ID	Text (3)	Hiermit wird der Rechnung ein Rechnungstyp nach XRechnung-Standard zugewiesen Fremdschlüssel aus "tbl_Rechnungstyp_Code"."Code_ID" Eingabe erforderlich: Ja [BT-3]
ProjektNr	Text	Projektnummer, auch Freitext Eingabe erforderlich: Nein [BT-11]
VertragsNr	Text	Vertragsnummer, auch Freitext Eingabe erforderlich: Nein [BT-12]
BestellNr	Text	Bestellnummer, auch Freitext Eingabe erforderlich: Nein [BT-13]
AuftragsNr	Text	Auftragsnummernummer, auch Freitext Eingabe erforderlich: Nein [BT-14]
AusschreibungsNr	Text	Ausschreibungsnummer, auch Freitext Eingabe erforderlich: Nein [BT-17]
ObjektNr	Text	Objektnummer, auch Freitext, (z. B. Zählernummer, Telefonnummer, Kfz-Kennzeichen, versicherte Person, Abonnementnummer, Rufnummer) Eingabe erforderlich: Nein [BT-18]
LieferReferenzen	Text	Beziehung zu vorhergehenden Lieferungen Eingabe erforderlich: Nein [BT-25]
Anmerkung	Text	Allgemeine Anmerkungen für die gesamte Lieferung Eingabe erforderlich: Nein [BT-22]

Das Feld "UnserZeichen" wird automatisch mit ausgefüllt, da es über den XRechnungs-Eingang befüllt wird.

Die Summen werden aus dem Import ausgelesen. Daneben werden die Summen allerdings über eine Ansicht noch einmal separat ermittelt und in dem Formular aufgezeigt.

tbl_Eingang

Datenziel
Ansicht: <i>Diese Ansicht dient zum Druck der erstellten geänderten Lieferung. Sie ist vom Prinzip her genauso aufgebaut wie die viw_Rechnung_Spalten.</i>
Abfrage: <i>qry_frmEingang</i>
Formular: <i>frm_Lieferung</i>
Makro: <i>Import, Import_to_Aenderung, Import_Values, TableRestart</i>

Feldname	Feldtyp	Beschreibung
ID	Integer	Primärschlüssel der Tabelle. Der Primärschlüssel muss eindeutig sein. Das Feld ist als Auto-Wert-Feld gesetzt.
Lieferung_ID	Integer	Hiermit wird der Eingang der Lieferung zugewiesen Fremdschlüssel aus "tbl_Lieferung"."ID" Eingabe erforderlich: Ja
LieferantZeilenID	Text	Zeilenbezeichnung aus der XRechnung. Kann auch eine einfach laufende Nummerierung sein. Eingabe erforderlich: Nein [BT-126]
Anzahl	Dezimal (16,8)	Anzahl der Ware, wird bei anderen Maßeinheiten auch mit Nachkommastellen angegeben Eingabe erforderlich: Ja [BT-129]
Anzahl_Code_ID	Text (3 Zeichen)	Maßeinheit für die Anzahl der Ware; ist bereits in der Tabelle mit dem Standardwert 'XPP' für die Stückzahl voreingestellt. Kann auch so etwas wie Paletten, Stunden usw. sein. Fremdschlüssel aus "tbl_Anzahl_Code"."Code_ID" Eingabe erforderlich: Nein [BT-130]
Ware	Text	Ausführliche Beschreibung der Ware auch über mehrere Zeilen (250 Zeichen) Eingabe erforderlich: Ja [BT-127][BT-128]
Preis	Dezimal (10,4)	Einzelpreis der Ware, Angabe bis 4 Stellen nach dem Komma in XRechnung möglich Eingabe erforderlich: Ja [BT-131][BT-146]
Rabatt	Dezimal (3,2)	Rabattangabe in Prozent Eingabe erforderlich: Nein [BT-136]
Steuersatz	Dezimal (3,2)	Steuersatz für die Ware, Angabe in Prozent Eingabe erforderlich: Ja [BT-152]
Steuer_Code_ID	Text (3)	Hiermit wird der Steuercode aus XRechnung der Rechnung ausgelesen Fremdschlüssel aus "tbl_Steuer_Code"."Code_ID" Eingabe erforderlich: Nein [BT-151]
StartDatum	Datum	Datum für den Lieferposten oder Beginn einer Periode Eingabe erforderlich: Nein [BT-134]
EndDatum	Datum	Ende einer Periode Eingabe erforderlich: Nein [BT-135]

Feldname	Feldtyp	Beschreibung
Eingang_ID	Integer	Hiermit wird der Eingang einem anderen Eingang untergeordnet (XRechnung-Extension) Fremdschlüssel aus "tbl_Eingang"."ID" Eingabe erforderlich: Nein
Kontoreferenzkunde	Text	An welcher Stelle in den Finanzkonten des Erwerbers ist die Rechnungszeile zu buchen? Muss vom Kunden vor der Rechnungserstellung mitgeteilt sein. Eingabe erforderlich: Nein [BT-133]

✓ Hinweis

Die Preis können in den Rechnungen ebenso wie die Anzahl mit beliebig viel Nachkommastellen angegeben werden. Hier werden beim Preis nur 4 Nachkommastellen, bei der Anzahl 8 Nachkommastellen berücksichtigt. Ist ein Preis sehr klein, hat aber mehr als die vorgesehene Anzahl an Nachkommastellen und ist die Anzahl besonders hoch, so kann dies zu Differenzen zwischen dem berechneten Betrag der Datenbank und der XRechnung führen. Maßgebend für die Zahlung ist also immer, was in der Lieferung angegeben wird, da eben nicht unbedingt alle Nachkommastellen übernommen werden.

tbl_Lieferung_Zusatz

Datenziel
Ansicht: <i>Diese Ansicht dient zum Druck der erstellten geänderten Lieferung. Sie ist vom Prinzip her genauso aufgebaut wie die viw_Rechnung_Spalten.</i>
Formular: <i>frm_Lieferung</i>
Makro: <i>Import, Import_to_Aenderung, Import_Values, TableRestart</i>

Feldname	Feldtyp	Beschreibung
ID	Integer	Primärschlüssel der Tabelle. Der Primärschlüssel muss eindeutig sein. Das Feld ist als Auto-Wert-Feld gesetzt.
Lieferung_ID	Integer	Hiermit wird der Zusatz der Lieferung zugewiesen Fremdschlüssel aus "tbl_Lieferung"."ID" Eingabe erforderlich: Ja
Zusatz	Text	Grund für Ermäßigung oder Aufschlag auf die Rechnung wie «Portokosten», «Rabatt» Eingabe erforderlich: Ja [BT-97][BT-104]
Betrag	Dezimal (8,2)	Bei Aufschlägen positiver Betrag, bei Rabatt negativer Betrag Eingabe erforderlich: Ja [BT-92][BT-99]
Steuersatz	Dezimal (3,2)	Steuersatz für den Zusatz, Angabe in Prozent Eingabe erforderlich: Ja [BT-96][BT-103]
Steuer_Code_ID	Text (3)	Hiermit wird der Steuercode aus XRechnung der Rechnung ausgelesen Fremdschlüssel aus "tbl_Steuer_Code"."Code_ID" Eingabe erforderlich: Nein [BT-95][BT-102]

tbl_Lieferung_Anhang

Datenziel
Formular: <i>frm_Lieferung</i>
Makro: <i>SaveXRechnung, Import_to_Aenderung, Import_Values, TableRestart</i>

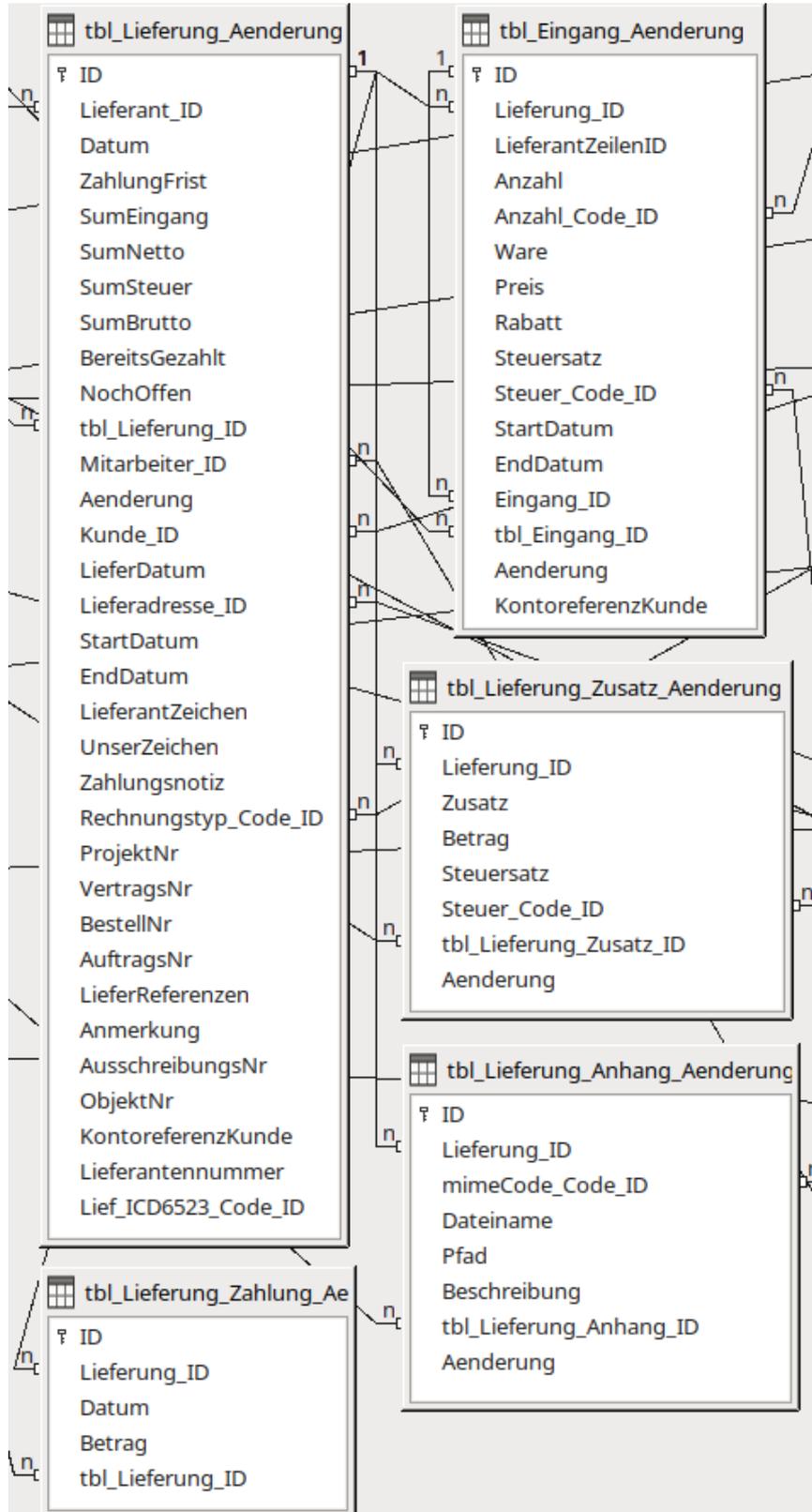
Feldname	Feldtyp	Beschreibung
ID	Integer	Primärschlüssel der Tabelle. Der Primärschlüssel muss eindeutig sein. Das Feld ist als Auto-Wert-Feld gesetzt.
Lieferung_ID	Integer	Hiermit wird der Zusatz der Rechnung zugewiesen Fremdschlüssel aus "tbl_Lieferung"."ID" Eingabe erforderlich: Ja
mimeCode_ Code_ID	Text(4)	Fremdschlüssel aus "tbl_mimeCode"."Code_ID" Eingabe erforderlich: Ja [BT-125] (Mimecode Binärdatei)
Dateiname	Text	Dateiname der ausgelesenen Datei Eingabe erforderlich: Nein [BT-122]
Pfad	Text	Pfad zu der ausgelesenen Datei im Binärformat. Die Datei wird in einem grafischen Kontrollfeld angezeigt. Eingabe erforderlich: Nein [BT-125] (Binärdatei)
Beschreibung	Text	Kurze Beschreibung des Anhangs Eingabe erforderlich: Nein [BT-123]

tbl_Lieferung_Zahlung

Datenziel
Abfrage: <i>qry_frmLieferungRechnung</i>
Makro: <i>Import_all, Import_Values, TableContentDeleteAll, TableRestart</i>

Feldname	Feldtyp	Beschreibung
ID	Integer	Primärschlüssel der Tabelle. Der Primärschlüssel muss eindeutig sein. Das Feld ist als Auto-Wert-Feld gesetzt.
Lieferung_ID	Integer	Hiermit wird die Zahlung der Liegerung zugewiesen Fremdschlüssel aus "tbl_Lieferung"."ID" Eingabe erforderlich: Ja
Datum	Datum	Datum für den Eingang der Zahlung Eingabe erforderlich: Ja
Betrag	Dezimalzahl	Höhe des eingegangenen Betrags Eingabe erforderlich: Ja

Bearbeitung und Weiterleitung des Eingangs



Hier ist nur der Block für die Bearbeitung der Lieferung abgebildet. Die vielen Verzweigungen zur eingehenden Lieferung, zur ausgehenden Rechnung und zu den Code-Tabellen sind weiter im Bild zu sehen.

Alle Tabellen werden durch den Inhalt aus den entsprechenden Liefertabellen über das entsprechende Formular gefüllt. Änderungen der Inhalte erfolgen in diesen Tabellen, nicht in den Originalen. Die Änderungen können dabei als "gesehen" und "geändert" gekennzeichnet werden.

tbl_Lieferung_Aenderung

Datenziel
Formular: <i>frm_Lieferung_Aenderung</i>
Ansicht: <i>viw_Lieferung_Aenderung, viw_Lieferung_Eingang_Aenderung</i>
Abfrage: <i>qry_Lieferung_Zahlung_Aenderung, qry_frmZusatz-Lieferung_Aenderung</i>
Makro: <i>Import_to_Aenderung, Import_all, Import_Values, TableContentDeleteAll, TableRestart</i>

Feldname	Feldtyp	Beschreibung
ID	Integer	Primärschlüssel der Tabelle. Der Primärschlüssel muss eindeutig sein. Das Feld ist als Auto-Wert-Feld gesetzt.
Lieferant_ID	Integer	Hiermit wird der Lieferant der Lieferung zugewiesen Fremdschlüssel aus "tbl_Lieferant"."ID" Eingabe erforderlich: Ja
Datum	Datum	Datum der Rechnung Eingabe erforderlich: Ja
LieferantZeichen	Text	Rechnungsnummer oder Liefernummer Eingabe erforderlich: Ja
UnserZeichen	Text	Bestellzeichen oder ähnliches Eingabe erforderlich: Nein
ZahlungFrist	Datum	Lieferort für Waren an die Firma/Person Eingabe erforderlich: Nein
Zahlungsnotiz	Text	Notizen auf der Rechnung zur Zahlungsweise Eingabe erforderlich: Nein
SumEingang	Dezimal (8,2)	Summe aus "tbl_Eingang" für die Lieferung Eingabe erforderlich: Nein
SumNetto	Dezimal (8,2)	Summe "tbl_Eingang" und "tbl_Lieferung_Zusatz" für die Lieferung Eingabe erforderlich: Nein
SumSteuer	Dezimal (8,2)	Summe der Steuer insgesamt Eingabe erforderlich: Nein
SumBrutto	Dezimal (8,2)	Brutto insgesamt Eingabe erforderlich: Nein
BereitsGezahlt	Dezimal (8,2)	Bisher erfolgte Zahlen als Abschlagszahlungen oder Raten Eingabe erforderlich: Nein
NochOffen	Dezimal (8,2)	Offener Betrag nach dieser Lieferung (Brutto - BereitsGezahlt) Eingabe erforderlich: Nein
tbl_Lieferung_ID	Integer	Hiermit wird die tbl_Lieferung der entsprechenden Änderungstabelle zugewiesen Fremdschlüssel aus "tbl_Lieferung"."ID" Eingabe erforderlich: Ja
Mitarbeiter_ID	Integer	Hiermit wird der Mitarbeiter der Firma der zu erstellenden Rechnung zugewiesen Fremdschlüssel aus "tbl_Mitarbeiter"."ID" Eingabe erforderlich: Nein

Feldname	Feldtyp	Beschreibung
Aenderung	Integer	Über eine Abfrage definiert: 1 → "gesehen", 2 → "geändert". Damit soll festgehalten werden, ob eine Rechnungszeile zur Kenntnis genommen oder ggf. angepasst wurde. Eingabe erforderlich: Nein
Kunde_ID	Integer	Hiermit wird ein Kunde der Lieferung zugewiesen, sofern die Lieferung zur Weiterleitung vorgesehen sein soll. Fremdschlüssel aus "tbl_Kunde"."ID" Eingabe erforderlich: Nein
LieferDatum	Datum	Datum der Lieferung (nicht der Rechnung zur Lieferung) Eingabe erforderlich: Nein
Lieferadresse_ID	Integer	Fremdschlüssel zu "tbl_Kunde_Lieferadresse"."ID" Eingabe erforderlich: Nein
StartDatum	Datum	Startdatum für den angegebenen Zeitraum Eingabe erforderlich: Nein
EndDatum	Datum	Enddatum für den angegebenen Zeitraum Eingabe erforderlich: Nein
Rechnungstyp_Code_ID	Text (3)	Hiermit wird der Rechnung ein Rechnungstyp nach XRechnungs-Standard zugewiesen Fremdschlüssel aus "tbl_Rechnungstyp_Code"."Code_ID" Eingabe erforderlich: Ja
ProjektNr	Text	Projektnummer, auch Freitext Eingabe erforderlich: Nein
VertragsNr	Text	Vertragsnummer, auch Freitext Eingabe erforderlich: Nein
BestellNr	Text	Bestellnummer, auch Freitext Eingabe erforderlich: Nein
AuftragsNr	Text	Auftragsnummenummer, auch Freitext Eingabe erforderlich: Nein
AusschreibungsNr	Text	Ausschreibungsnummer, auch Freitext Eingabe erforderlich: Nein [BT-17]
ObjektNr	Text	Objektnummer, auch Freitext, (z. B. Zählernummer, Telefonnummer, Kfz-Kennzeichen, versicherte Person, Abonnementnummer, Rufnummer) Eingabe erforderlich: Nein [BT-18]
LieferReferenzen	Text	Beziehung zu vorhergehenden Lieferungen Eingabe erforderlich: Nein
Anmerkung	Text	Allgemeine Anmerkungen für die gesamte Lieferung Eingabe erforderlich: Nein

Das Feld "UnserZeichen" wird automatisch mit ausgefüllt, da es über den XRechnungs-Eingang befüllt wird.

Die Summen werden aus dem Import ausgelesen. Daneben werden die Summen allerdings über eine Ansicht noch einmal separat ermittelt und in dem Formular aufgezeigt.

tbl_Eingang_Aenderung

Datenziel	
Ansicht:	<i>viw_Lieferung_Spalten_Aenderung, viw_Lieferung_Steuer_Aenderung, viw_Lieferung_Eingang_Aenderung</i>
Abfrage:	<i>qry_frmEingangAenderung</i>
Makro:	<i>Import_to_Aenderung, Import_Values, TableRestart</i>

Feldname	Feldtyp	Beschreibung
ID	Integer	Primärschlüssel der Tabelle. Der Primärschlüssel muss eindeutig sein. Das Feld ist als Auto-Wert-Feld gesetzt.
Lieferung_ID	Integer	Hiermit wird der Eingang der Lieferung zugewiesen Fremdschlüssel aus "tbl_Lieferung"."ID" Eingabe erforderlich: Ja
LieferantZeilenID	Text	Zeilenbezeichnung aus der XRechnung. Kann auch eine einfach laufende Nummerierung sein. Eingabe erforderlich: Nein
Anzahl	Dezimal (16,8)	Anzahl der Ware, wird bei anderen Maßeinheiten auch mit Nachkommastellen angegeben Eingabe erforderlich: Ja
Anzahl_Code_ID	Text (3 Zeichen)	Maßeinheit für die Anzahl der Ware; ist bereits in der Tabelle mit dem Standardwert 'XPP' für die Stückzahl voreingestellt. Kann auch so etwas wie Paletten, Stunden usw. sein. Fremdschlüssel aus "tbl_Anzahl_Code"."Code_ID" Eingabe erforderlich: Nein
Ware	Text	Ausführliche Beschreibung der Ware auch über mehrere Zeilen (250 Zeichen) Eingabe erforderlich: Ja
Preis	Dezimal (10,4)	Einzelpreis der Ware, Angabe bis 4 Stellen nach dem Komma in XRechnung möglich Eingabe erforderlich: Ja
Rabatt	Dezimal (3,2)	Rabattangabe in Prozent Eingabe erforderlich: Nein
Steuersatz	Dezimal (3,2)	Steuersatz für die Ware, Angabe in Prozent Eingabe erforderlich: Ja
Steuer_Code_ID	Text (3)	Hiermit wird der Steuercode aus XRechnung der Rechnung ausgelesen Fremdschlüssel aus "tbl_Steuer_Code"."Code_ID" Eingabe erforderlich: Nein
StartDatum	Datum	Datum für den Lieferposten oder Beginn einer Periode Eingabe erforderlich: Nein
EndDatum	Datum	Ende einer Periode Eingabe erforderlich: Nein
Eingang_ID	Integer	Hiermit wird der Eingang einem anderen Eingang untergeordnet (XRechnung-Extension) Fremdschlüssel aus "tbl_Eingang"."ID" Eingabe erforderlich: Nein
tbl_Eingang_ID	Integer	Hiermit wird die tbl_Eingang der entsprechenden Änderungstabelle zugewiesen Fremdschlüssel aus "tbl_Eingang"."ID" Eingabe erforderlich: Ja

Feldname	Feldtyp	Beschreibung
Aenderung	Integer	Über eine Abfrage definiert: 1 → "gesehen", 2 → "geändert". Damit soll festgehalten werden, ob eine Rechnungszeile zur Kenntnis genommen oder ggf. angepasst wurde. Eingabe erforderlich: Nein
KontoreferenzKunde	Text	An welcher Stelle in den Finanzkonten des Erwerbers ist die Rechnungszeile zu buchen? Muss vom Kunden vor der Rechnungserstellung mitgeteilt sein. Eingabe erforderlich: Nein [BT-133]

tbl_Lieferung_Zusatz_Aenderung

Datenziel
Ansicht:
Formular: <i>frm_Lieferung_Aenderung</i>
Makro: <i>FillTableCarryOver, Import_to_Aenderung, Import_Values, TableRestart</i>

Feldname	Feldtyp	Beschreibung
ID	Integer	Primärschlüssel der Tabelle. Der Primärschlüssel muss eindeutig sein. Das Feld ist als Auto-Wert-Feld gesetzt.
Lieferung_ID	Integer	Hiermit wird der Zusatz der Lieferung zugewiesen Fremdschlüssel aus "tbl_Lieferung_Aenderung"."ID" Eingabe erforderlich: Ja
Zusatz	Text	Grund für Ermäßigung oder Aufschlag auf die Rechnung wie «Portokosten», «Rabatt» Eingabe erforderlich: Ja
Betrag	Dezimal (8,2)	Bei Aufschlägen positiver Betrag, bei Rabatt negativer Betrag Eingabe erforderlich: Ja
Steuersatz	Dezimal (3,2)	Steuersatz für den Zusatz, Angabe in Prozent Eingabe erforderlich: Ja
Steuer_Code_ID	Text (3)	Hiermit wird der Steuercode aus XRechnung der Rechnung ausgelesen Fremdschlüssel aus "tbl_Steuer_Code"."Code_ID" Eingabe erforderlich: Nein
tbl_Lieferung_Zusatz_ID	Integer	Hiermit wird die tbl_Lieferung_Zusatz der entsprechenden Änderungstabelle zugewiesen Fremdschlüssel aus "tbl_Lieferung_Zusatz"."ID" Eingabe erforderlich: Ja
Aenderung	Integer	Über eine Abfrage definiert: 1 → "gesehen", 2 → "geändert". Damit soll festgehalten werden, ob eine Rechnungszeile zur Kenntnis genommen oder ggf. angepasst wurde. Eingabe erforderlich: Nein

tbl_Lieferung_Anhang_Aenderung

Datenziel
Formular: <i>frm_Lieferung_Aenderung</i>
Makro: <i>FillTableCarryOver, Import_to_Aenderung, Import_Values, TableRestart</i>

Feldname	Feldtyp	Beschreibung
ID	Integer	Primärschlüssel der Tabelle. Der Primärschlüssel muss eindeutig sein. Das Feld ist als Auto-Wert-Feld gesetzt.
Lieferung_ID	Integer	Hiermit wird der Zusatz der Rechnung zugewiesen Fremdschlüssel aus "tbl_Lieferung_Aenderung"."ID" Eingabe erforderlich: Ja
mimeCode_ Code_ID	Text(4)	Fremdschlüssel aus "tbl_mimeCode"."Code_ID" Eingabe erforderlich: Ja
Dateiname	Text	Dateiname der ausgelesenen Datei Eingabe erforderlich: Nein
Pfad	Text	Pfad zu der ausgelesenen Datei im Binärformat. Die Datei wird in einem grafischen Kontrollfeld angezeigt. Eingabe erforderlich: Nein
Beschreibung	Text	Kurze Beschreibung des Anhangs Eingabe erforderlich: Nein
tbl_Lieferung_ Anhang_ID	Integer	Hiermit wird die tbl_Lieferung_Anhang der entsprechenden Änderungstabelle zugewiesen Fremdschlüssel aus "tbl_Lieferung_Anhang"."ID" Eingabe erforderlich: Ja
Aenderung	Integer	Über eine Abfrage definiert: 1 → "gesehen", 2 → "geändert". Damit soll festgehalten werden, ob eine Rechnungszeile zur Kenntnis genommen oder ggf. angepasst wurde. Eingabe erforderlich: Nein

tbl_Lieferung_Zahlung_Aenderung

Datenziel
Abfrage: qry_frmLieferungRechnung_Aenderung
Makro: FillTableCarryOver , Import_all , Import_Values , TableContentDeleteAll , TableRestart

Feldname	Feldtyp	Beschreibung
ID	Integer	Primärschlüssel der Tabelle. Der Primärschlüssel muss eindeutig sein. Das Feld ist als Auto-Wert-Feld gesetzt.
Lieferung_ID	Integer	Hiermit wird die Zahlung der Lieferung zugewiesen Fremdschlüssel aus "tbl_Lieferung_Aenderung"."ID" Eingabe erforderlich: Ja
Datum	Datum	Datum für den Eingang der Zahlung Eingabe erforderlich: Ja
Betrag	Dezimalzahl	Höhe des eingegangenen Betrags Eingabe erforderlich: Ja
tbl_Lieferung_ ID	Integer	Fremdschlüssel aus "tbl_Lieferung"."ID" Eingabe erforderlich: Nein

Einzeltable ohne definierte Verbindung

Einige Tabellen sind so erstellt, dass sie keine Verbindung zu den oben genannten Tabellen haben und auch keine Verbindung untereinander aufweisen.

tbl_Firma

Datenziel
Ansicht: <i>viw_Rechnung, viw_Lieferung_Spalten_Aenderung</i>
Abfrage: <i>qry_Steuercode_Listbox</i>
Formular: <i>frm_Einstellungen</i>
Makro: <i>ToolbarsHide, Import_all, Import_Values</i>

Feldname	Feldtyp	Beschreibung
ID	Ja/Nein	Primärschlüssel der Tabelle. Hier ist ein Ja/Nein-Feld gewählt, weil sowieso nur ein Eintrag erfolgen soll.
Firma	Text	Der Name der eigenen Firma Eingabe erforderlich: Ja [BT-27]
StraßeNr	Text	Straße und Hausnummer der Firma Eingabe erforderlich: Ja (obwohl für XRechnung nicht erforderlich) [BT-35]
Plz	Text	Postleitzahl für die eigene Firma Eingabe erforderlich: Ja [BT-38]
Ort	Text	Ortsangabe für die eigene Firma Eingabe erforderlich: Ja [BT-37]
E-Mail	Text	E-Mail-Adresse der eigenen Firma Eingabe erforderlich: Ja [BT-34]
Website	Text	Website der eigenen Firma Eingabe erforderlich: Nein
Telefon	Text	Telefonkontakt zur eigenen Firma Eingabe erforderlich: Ja [BT-42] (ggf. statt Mitarbeiter)
UST-IdNr	Text	Umsatzsteuernummer Eingabe erforderlich: Ja [BT-31]
Firma-Registriert	Text	Registrierter Name der Firma - kann anders als der Rufname sein. Muss für die XRechnung ausgefüllt sein. Eingabe erforderlich: Ja [BT-28]
Bank	Text	Name der Bank Eingabe erforderlich: Ja (für die Information in der Fußzeile der ausdrückbaren Rechnung)
IBAN	Text	IBAN Eingabe erforderlich: Ja [BT-84]
BIC	Text	BIC Eingabe erforderlich: Ja (für die Information in der Fußzeile der ausdrückbaren Rechnung) [BT-86]
RegisterE1	Text	Amtsregistereintrag 1, der auf der Rechnung erscheinen soll, Kennung der Rechtsform Eingabe erforderlich: Nein [BT-30]
RegisterE2	Text	Amtsregistereintrag 2, der auf der Rechnung erscheinen soll; Sonstige rechtliche Informationen des Lieferanten wie z.B. "Kein Ausweis von Umsatzsteuer, da Kleinunternehmer gemäß § 19UStG" Eingabe erforderlich: Nein [BT-33]
RegisterE3	Text	Amtsregistereintrag 3, der auf der Rechnung erscheinen soll Eingabe erforderlich: Nein

Feldname	Feldtyp	Beschreibung
Logo	Bild	Logo, das in dem Briefkopf auftaucht Eingabe erforderlich: Nein
Zahlungsnotiz	Text	Notizen auf der Rechnung zur Zahlungsweise usw. Allgemeine Notiz, die durch Notizen auf Rechnungsebene ersetzt werden kann. Eingabe erforderlich: Nein [BT-20]
ZahlungTage	Integer	Zahlungsfrist in Tagen nach Rechnungsstellung Eingabe erforderlich: Nein [BT-9]
Zahlungsempfänger	Text	Empfänger der Zahlung bei der Bank Eingabe erforderlich: Nein [BT-85]
ReStellen	SmallInteger	Anzahl der Stellen in der Rechnungsnummer Eingabe erforderlich: Nein
frmKunde_ - Warennummer	Ja/Nein	Warennummer in dem Kundenformular bei der Rechnungseingabe anzeigen Eingabe erforderlich: Nein
frmKunde_ - Warenzusatz	Ja/Nein	Warenzusatz in dem Kundenformular bei der Rechnungseingabe anzeigen Eingabe erforderlich: Nein
frmKunde_ Ra- batt	Ja/Nein	Rabatt in dem Kundenformular bei der Rechnungseingabe anzeigen Eingabe erforderlich: Nein
frmKunde_ - StartDatum	Ja/Nein	StartDatum in dem Kundenformular bei der Rechnungseingabe anzeigen Eingabe erforderlich: Nein
frmKunde_ - EndDatum	Ja/Nein	EndDatum in dem Kundenformular bei der Rechnungseingabe anzeigen Eingabe erforderlich: Nein
frmKunde_ - Steuer	Ja/Nein	Steuer in dem Kundenformular bei der Rechnungseingabe anzeigen Eingabe erforderlich: Nein
frmKunde_ De- faultSteuer- satz	Decimal (3,2)	Steuersatz, der eingefügt wird, wenn die Auswahl der Steuer im Formular Kunde ausgeschaltet wird. Eingabe erforderlich: Nein
frmKunde_ De- faultSteuer- CodeID	Text(3)	Steuercode, der eingefügt wird, wenn die Auswahl der Steuer im Formular Kunde ausgeschaltet wird. Eingabe erforderlich: Nein
frmKunde_ An- zahlCode	Ja/Nein	AnzahlCode (Maßeinheit) in dem Kundenformular bei der Rechnungseingabe anzeigen. Ohne Anzeige wird 'Stück', Code 'XPP', standardmäßig gewählt. Eingabe erforderlich: Nein
frmKunde_ - Kontoreferenz	Ja/Nein	Kontoreferenz für das Kundenformular anzeigen Eingabe erforderlich: Nein
Gläubiger- nummer	Text	Für Lastschriften ist die Gläubigernummer erforderlich Eingabe erforderlich: Nein [BT-90]
SteuerNr	Text	Hier die einfache Steuernummer, nicht die für die Umsatzsteuer, zur Angabe in der Rechnung vor allem bei Kleinunternehmen. Eingabe erforderlich: Nein [BT-32]
Steuer_An- merkung_ID	Integer	Fremdschlüssel aus "tbl_Steuer_Anmerkung", Begründung für Steuerfreiheit Eingabe erforderlich: Nein

Feldname	Feldtyp	Beschreibung
Zeitraumtext	Text	Die Zeitraumangabe im PDF-Dokument erscheint als 'von : ... bis: ...' Hier kann ein Text vorangestellt werden wie 'Lieferzeitraum', 'Mietzeitraum', 'Abrechnungszeitraum' Eingabe erforderlich: Nein
Druck_Netto-SteuerBruttoSum	Ja/Nein	Sollen im Druck in den Rechnungszeilen die Beträge für Netto, Steuer und Brutto nebeneinander erscheinen und anschließend unterhalb auch summiert werden? Eingabe erforderlich: Nein

tbl_Steuer_Anmerkung

Datenziel
Ansicht: viw_Rechnung
Formular: frm_Einstellungen

Feldname	Feldtyp	Beschreibung
ID	Integer	Primärschlüssel der Tabelle. Der Primärschlüssel muss eindeutig sein. Das Feld ist als Auto-Wert-Feld gesetzt.
Steuer_Code_ID	Text (3)	Hiermit wird die Befreiung dem Steuercode zugewiesen Fremdschlüssel aus "tbl_Steuer_Code"."ID" Eingabe erforderlich: Ja
Anmerkung	Text (200)	Anmerkung, die auf der Rechnung und in der XRechnung als Befreiungsgrund erscheint Eingabe erforderlich: Ja

tbl_Druck

Datenziel
Ansicht: viw_Lieferung_DruckSpalten , viw_Rechnung_DruckSpalten
Formular: frm_Einstellungen

Feldname	Feldtyp	Beschreibung
ID	Integer	Primärschlüssel der Tabelle. Der Primärschlüssel muss eindeutig sein. Eingabe erforderlich: Ja
Tabellenkopf	Text (100)	Erscheint als Tabellenüberschrift im Ausdruck. Wenn das Feld leer bleibt, dann erscheint für die Spalte keine Überschrift.. Eingabe erforderlich: Nein
Feld	Text (100)	Feldname der Spalte aus den Ansichten "viw_Rechnung_Ausgang" bzw. "viw_Lieferung_Eingang_Aenderung" Eingabe erforderlich: Ja
Format	Text (100)	Format der Spalten. Die Texte dafür sind vorgegeben. Nur bei den entsprechenden Angaben erfolgt eine einwandfreie Formatierung als Datum, Währung, Währung mit ggf. 6 Nachkommastellen usw. Eingabe erforderlich: Ja

Feldname	Feldtyp	Beschreibung
Breite	Dezimal (4,2)	Breite der Spalten in der Drucktabelle. Der Eintrag bei der Ware wird nicht berücksichtigt, da er aus dem Rest berechnet wird. Bleiben andere Felder frei, so werden sie mit einem Wert von '1' in den Druck übernommen. Maximalangabe ist hier 10 für alle den aktuellen Druck betreffenden Zeilen zusammen. Eingabe erforderlich: Nein
Reihenfolge	SmallInteger	Die Reihenfolge muss eindeutig sein. Durch einen Index wird verhindert, dass bereits existierende Zahlen eingegeben werden. Die Reihenfolge muss aber nicht lückenlos sein. Eingabe erforderlich: Ja

tbl_Filter

Datenziel
Ansicht: <i>viw_Filter</i>
Formular: <i>frm_Einstellungen, frm_Rechnung, frm_Lieferung, frm_Lieferadresse</i>
Makro: <i>FilterStart, KundeLieferadresse, KundeRechnungszusatz, Kundezusatz, Lieferantzusatz, Einstellungenzusatz</i>

Feldname	Feldtyp	Beschreibung
ID	Integer	Primärschlüssel der Tabelle. Der Schlüssel wird beim Start der ODB-Datei gesetzt. Er entspricht der Verbindungsnummer der Datenbank, so dass der Filter auch in Netzwerkumgebungen bei Nutzung einer Serverdatenbank seine Funktion erfüllt
Formular	Text	Name des Formulars, das gerade aktiv ist. Dient zum Sprung von einem Formular zum nächsten und wird über Makro beschrieben.
Lieferant_ID	Integer	Hier wird die ID des Lieferanten eingetragen, dessen Lieferung gerade eingelesen wird. So kann immer die aktuell eingelesene Lieferung sofort angezeigt werden. Auch die Filterung nach einem Lieferanten mittels Listenfeld nutzt dieses Feld.
Kunde_ID	Integer	Hiermit wird das Kundenformular nach einem bestimmten Kunden gefiltert. Der Wert dazu wird durch das Listenfeld zum Filtern im Kundenformular festgelegt.
Lieferung_ID	Integer	Filterung nach Lieferungen für frm_Zusatz
Rechnung_ID	Integer	Filterung nach Rechnungen für frm_Zusatz
Lieferung_Aenderung_ID	Integer	Filterung nach zu ändernden Lieferungen für frm_Zusatz
Listbox_ID	Integer	Voreinstellung für Listbox
ShowRowNr	Ja/Nein	Soll beim Druck die Zeilennummer erscheinen, so wird hier 'Ja' eingestellt.

Verknüpfung von Tabellen und Löschen von Datensätzen

Die Tabellen im Kundenbereich und im Lieferantenbereich sind untereinander und über einige Tabellen auch miteinander verknüpft. Gibt eine Tabelle ihren Primärschlüssel an eine andere

Tabelle als Fremdschlüssel weiter, so ist es standardmäßig nicht möglich, dass ein Datensatz dieser Tabelle gelöscht werden kann, wenn er als Fremdschlüssel noch in Benutzung ist:

Relationen

Beteiligte Tabellen

tbl_Rechnung | tbl_Kunde

Beteiligte Felder

tbl_Rechnung	tbl_Kunde
Kunde_ID	ID

Optionen aktualisieren

- Keine Aktion
- Kask. aktualisieren
- NULL setzen
- Standard setzen

Löschoptionen

- Keine Aktion
- Kask. löschen
- NULL setzen
- Standard setzen

Hilfe | OK | Abbrechen

In der Tabelle "tbl_Rechnung" befindet sich das Feld "Kunde_ID". Wird jetzt in der Tabelle "tbl_Kunde" der Wert für die "ID" geändert, so weist die Datenbank das zurück, wenn dieser Datensatz aus "tbl_Kunde" bereits mit "tbl_Rechnung" verbunden ist. Bei Tabellen wie der "tbl_Kunde" mit einem AutoWert-Feld als Primärschlüssel macht eine Änderung sowieso nicht viel Sinn.

Existieren zu einem Kunden bereits Rechnungen, so weist die Datenbank eine Löschung des Datensatzes in der "tbl_Kunde" zurück. Es müssten zuerst alle Rechnungen gelöscht werden. Dann erst kann der Kunde aus dem Datenbestand entfernt werden.

Relationen

Beteiligte Tabellen

tbl_Ausgang tbl_Rechnung

Beteiligte Felder

tbl_Ausgang	tbl_Rechnung
Rechnung_ID	ID

Optionen aktualisieren

Keine Aktion
 Kask. aktualisieren
 NULL setzen
 Standard setzen

Löschoptionen

Keine Aktion
 Kask. löschen
 NULL setzen
 Standard setzen

Hilfe OK Abbrechen

Die Tabelle "tbl_Ausgang" ist über einen Fremdschlüssel ("Rechnung_ID") mit der Tabelle "tbl_Rechnung" verbunden. Bei dieser Relation ist es möglich, den Primärschlüssel in "tbl_Rechnung" zu ändern. Erfolgt eine Änderung des Wertes in "tbl_Rechnung", dann wird auch der Wert in "tbl_Ausgang" angepasst.

Bei dieser Form der Verknüpfung ist es auch möglich, einen Datensatz in der "tbl_Rechnung" zu löschen. Wird dort ein Datensatz gelöscht, so werden automatisch alle Datensätze in "tbl_Ausgang" gelöscht, die mit dieser Rechnung verbunden sind. Da die Verbindung von "tbl_Rechnung_Zusatz" zu "tbl_Rechnung" nach dem gleichen Muster erstellt ist, kann problemlos ein Datensatz aus "tbl_Rechnung" gelöscht werden. Alle betroffenen Datensätze in "tbl_Rechnung_Zusatz" werden dann ebenfalls entfernt.

Im Lieferantenbereich ist die Verknüpfung ähnlich: Ein Datensatz aus "tbl_Lieferant" kann nicht gelöscht werden, wenn er mit einem Datensatz aus "tbl_Lieferung" verbunden ist. In "tbl_Lieferung" können aber problemlos Datensätze gelöscht werden, da entsprechende dazu gehörende Daten aus "tbl_Eingang" und "tbl_Lieferung_Zusatz" ebenfalls gelöscht werden.

Soll also die vorliegende Datenbank für den produktiven Gebrauch geleert werden, so müssen zuerst die Datensätze in "tbl_Rechnung" und "tbl_Lieferung" gelöscht werden, bevor die Datensätze in "tbl_Kunde" und "tbl_Lieferant" sicher gelöscht werden können.

Ansichten

Ansichten sind in der Datenbank abgespeicherte feste Abfrage. Die Ansichten existieren in dieser Datei gleichzeitig als Abfragen. Soll eine Ansicht geändert werden, auf die eine andere Ansicht zugreift, so ist dies nicht möglich. Dafür wird dann die entsprechende Abfrage geändert und anschließend werden über ein Makro einfach alle Ansichten in der richtigen Reihenfolge neu geschrieben.

viw_Filter

ID	Formular	Lieferant_ID	Kunde_ID	Lieferung_ID	Rechnung_ID	Lieferung_Aenderung_ID	Listbox_ID	Rechnungsnummer	KundeLieferant
9	frm_XRechnung-Ext	8		11	3	6	10		

Datenquelle

Tabelle: *tbl_Filter*

Datenziel

Ansicht: *viw_Lieferung_DruckSpalten, viw_Rechnung_DruckSpalten*

Abfrage: *qry_Forms, qry_frmZusatz-Rechnung, qry_frmZusatz-Lieferung_Aenderung, qry_frmZusatz-Lieferung, qry_frmZusatz-Kunde, qry_frmZusatz-Lieferant, qry_frmZusatz-Lieferant*

Formular: *frm_Einstellungen*

Makro: *FilterStart, ViewsErstellen, Kundezusatz, Lieferantzusatz, Einstellungenzusatz*

```
001 SELECT * FROM "tbl_Filter" WHERE "ID" = CURRENT_CONNECTION
```

Hier geht es nur darum, aus "tbl_Filter" den Datensatz auszulesen, der direkt mit der aktuellen Verbindung der FIREBIRD Datenbank zusammenhängt. Dadurch ist die Filterung auch im Mehrbenutzerbetrieb möglich, falls die Datenbank einmal zur externen Datenbank und von dort zur Serverdatenbank geändert werden sollte.

viw_Lieferung_Aenderung

Firma	StraßeNr	Plz	Ort	E-Mail	Website	Telefon	UST-IdNr
Testverkäufer	Teststraße	12345	Testha	test@test.		123456789	DE123454648 T
[Seller name]	[Seller address	12345	[Seller c	xxx@schu		+49 123456	DE123456789 [S

Ausschnitt der Ausgabe der Ansicht "viw_Lieferung_Aenderung". Die Ansicht zeigt alle Felder, die zum Rechnungsausdruck notwendig sind.

Datenquelle

Tabelle: *tbl_Lieferant, tbl_Kunde, tbl_Mitarbeiter, tbl_Land, tbl_Lieferung_Aenderung, tbl_Kunde_Lieferadresse*

Ansicht: *viw_Lieferung_Summe_Aenderung*

Datenziel

Formular: *frm_Lieferung_Aenderung*

Makro: *FillTableCarryOver, ViewsErstellen*

```
001 SELECT "tbl_Lieferant"."Firma", "tbl_Lieferant"."StraßeNr", "tbl_Lieferant"."Plz",
"tbl_Lieferant"."Ort", "tbl_Lieferant"."E-Mail", NULL AS "Website",
"tbl_Lieferant"."Telefon", "tbl_Lieferant"."UST-IdNr",
```

```

"tbl_Lieferant"."Firma_Registriert", "tbl_Lieferant"."Bank",
"tbl_Lieferant"."IBAN", "tbl_Lieferant"."BIC", NULL AS "RegisterE1",
NULL AS "RegisterE2", NULL AS "RegisterE3",
002 COALESCE("a"."Zahlungsnotiz", '')||COALESCE('Vorhergehende Rechnungen zu dieser
Rechnung: '||"LieferReferenzen",'')||COALESCE(ASCII_CHAR(13)||CASE WHEN
"tbl_Kunde"."Lastschrift" THEN 'Überweisen Sie nicht. Die Zahlung erfolgt per
Lastschrift.' ELSE NULL END,') AS "Zahlungsnotiz",
003 "tbl_Lieferant"."ZahlungTage",
004 COALESCE("tbl_Lieferant"."Zahlungsempfänger",
"tbl_Lieferant"."Firma_Registriert") AS "Zahlungsempfänger", NULL AS "Logo",
005 "tbl_Lieferant"."AnsprechP"||COALESCE(', '||"tbl_Mitarbeiter"."Name",'') AS
"Name", NULL AS "Geschlecht",
006 'Mitarbeiter' AS "Beschriftung_Mitarbeiter",
007 "tbl_Lieferant"."AnsprechP_E-Mail"||COALESCE(', '||"tbl_Mitarbeiter"."E-
Mail",'') AS "E-Mail_Mitarbeiter",
008 "tbl_Lieferant"."AnsprechP_Telefon"||COALESCE(', '||"tbl_Mitarbeiter"."Telefon",
') AS "Telefon_Mitarbeiter",
009 "tbl_Kunde"."ID" AS "Kunde_ID",
010 'K-'||RIGHT('000000'||"tbl_Kunde"."ID",6) AS "Kundenummer",
011 "tbl_Kunde"."KuerzelDatei",
012 "tbl_Kunde"."Firma" AS "Firma_Kunde", "tbl_Kunde"."AnsprechP",
"tbl_Kunde"."AnsprechPGeschlecht", "tbl_Kunde"."StraßeNr" AS "StraßeNr_Kunde",
"tbl_Kunde"."Plz" AS "Plz_Kunde", "tbl_Kunde"."Ort" AS "Ort_Kunde",
"tbl_Kunde"."Land" AS "LandKurz",
013 UPPER("tbl_Land"."Land") AS "Land", "tbl_Kunde"."E-Mail" AS "E-Mail_Kunde",
"tbl_Kunde"."Telefon" AS "Telefon_Kunde",
"tbl_Kunde"."E-Mail2" AS "E-Mail_Kunde2",
014 "tbl_Kunde"."Firma"||ASCII_CHAR(13)||"tbl_Kunde"."StraßeNr"||
ASCII_CHAR(13)||"tbl_Kunde"."Plz"||' '||"tbl_Kunde"."Ort"||CASE WHEN
"tbl_Kunde"."Land" = 'DE' OR "tbl_Kunde"."Land" IS NULL THEN ' ' ELSE
ASCII_CHAR(13)||UPPER("tbl_Land"."Land") END AS "Anschrift_Kunde",
015 "tbl_Kunde"."UST-IdNr" AS "UST-IdNr_Kunde",
"tbl_Kunde"."RegisterID" AS "RegisterID_Kunde",
ASCII_CHAR(13)||"tbl_Kunde"."DruckZusatz" AS "DruckZusatz_Kunde",
016 CASE WHEN "tbl_Kunde"."Lastschrift" THEN "tbl_Kunde"."IBAN" ELSE NULL END AS
"IBAN_Kunde",
017 "tbl_Kunde"."Mandatsreferenznummer",
018 "a"."ID" AS "Rechnung_ID",
019 "a"."ID" AS "Lieferung_ID",
020 RIGHT('0'||EXTRACT(DAY FROM "a"."Datum"),2)||'.'||RIGHT('0'||EXTRACT(MONTH FROM
"a"."Datum"),2)||'.'||EXTRACT(YEAR FROM "a"."Datum") AS "RechnungsdatumD",
021 "a"."Datum" AS "Rechnungsdatum",
022 "a"."Datum" + "tbl_Lieferant"."ZahlungTage" AS "ZahlungBis",
023 "a"."UnserZeichen" AS "KundeZeichen",
024 "a"."LieferantZeichen" AS "Rechnungsnummer",
025 "a"."LieferantZeichen" AS "RechnungsnummerMitZusatz",
026 "viw_Lieferung_Summe_Aenderung"."Netto",
"viw_Lieferung_Summe_Aenderung"."Steuer",
"viw_Lieferung_Summe_Aenderung"."Brutto",
027 "a"."Rechnungstyp_Code_ID", (SELECT "Name_DE" FROM "tbl_Rechnungstyp_Code" WHERE
"Code_ID" = "a"."Rechnungstyp_Code_ID") AS "Rechnungstyp",
028 "a"."LieferDatum",
029 CASE WHEN "a"."LieferDatum" IS NULL THEN NULL ELSE 'Lieferdatum: ' END AS
"lblLieferdatumD",
030 RIGHT('0'||EXTRACT(DAY FROM "a"."LieferDatum"),2)||'.'||RIGHT('0'||EXTRACT(MONTH
FROM "a"."LieferDatum"),2)||'.'||EXTRACT(YEAR FROM "a"."LieferDatum") AS
"LieferdatumD",
031 "l"."Firma" AS "Firma_KundeLiefer", "l"."StraßeNr" AS "StraßeNr_KundeLiefer",
"l"."Plz" AS "Plz_KundeLiefer", "l"."Ort" AS "Ort_KundeLiefer", "l"."Land" AS
"LandKurzLiefer", (SELECT UPPER("Land") FROM "tbl_Land" WHERE "ID" = "l"."Land")
AS "LandLiefer",
032 CASE WHEN "l"."Plz" IS NULL THEN NULL ELSE ASCII_CHAR(13)||'Abweichende
Lieferadresse: ' END AS "lblAnschrift_Lieferadresse",
033 ASCII_CHAR(13)||ASCII_CHAR(9)||"l"."Firma"||COALESCE(ASCII_CHAR(13)||
ASCII_CHAR(9)||"l"."StraßeNr",'')||ASCII_CHAR(13)||
ASCII_CHAR(9)||"l"."Plz"||' '||"l"."Ort"||CASE WHEN "l"."Land" = 'DE' OR

```

```

"l"."Land" IS NULL THEN '' ELSE ASCII_CHAR(13)||ASCII_CHAR(9)||UPPER((SELECT
"Land" FROM "tbl_Land" WHERE "ID" = "l"."Land")) END AS
"Anschrift_Lieferadresse",
034 "a"."StartDatum",
035 CASE WHEN "a"."StartDatum" IS NULL THEN NULL ELSE 'von: ' END AS
"lblStartDatumD",
036 RIGHT('0' || EXTRACT(DAY FROM "a"."StartDatum"),2) || '.' || RIGHT('0' || EXTRACT(MONTH
FROM "a"."StartDatum"),2) || '.' || EXTRACT(YEAR FROM "a"."StartDatum") AS
"StartDatumD",
037 "a"."EndDatum",
038 CASE WHEN "a"."EndDatum" IS NULL THEN NULL ELSE 'bis: ' END AS "lblEndDatumD",
039 RIGHT('0' || EXTRACT(DAY FROM "a"."EndDatum"),2) || '.' || RIGHT('0' || EXTRACT(MONTH
FROM "a"."EndDatum"),2) || '.' || EXTRACT(YEAR FROM "a"."EndDatum") AS
"EndDatumD",
040 "a"."ProjektNr", "a"."VertragsNr", "a"."BestellNr", "a"."AuftragsNr", NULL AS
"VorRechnungen_ID"
041 FROM "tbl_Kunde"
042 LEFT JOIN "tbl_Lieferung_Aenderung" AS "a" ON "a"."Kunde_ID" = "tbl_Kunde"."ID"
043 LEFT JOIN "tbl_Kunde_Lieferadresse" AS "l" ON "l"."ID" = "a"."Lieferadresse_ID"
044 LEFT JOIN "tbl_Mitarbeiter" ON "a"."Mitarbeiter_ID" = "tbl_Mitarbeiter"."ID"
045 LEFT JOIN "tbl_Land" ON "tbl_Kunde"."Land" = "tbl_Land"."ID"
046 LEFT JOIN "viw_Lieferung_Summe_Aenderung" ON "a"."ID" =
"viw_Lieferung_Summe_Aenderung"."Lieferung_ID"
047 LEFT JOIN "tbl_Lieferant" ON "a"."Lieferant_ID" = "tbl_Lieferant"."ID"
048 WHERE LEFT("viw_Lieferung_Summe_Aenderung"."Sort",1) = 6

```

Diese Ansicht dient zum Druck der erstellten geänderten Lieferung. Sie ist vom Prinzip her genauso aufgebaut wie die [viw_Rechnung](#), hat aber auch einige Felder, die wegen der Austauschbarkeit innerhalb eines Makros einfach **NULL** sind.

viw_Lieferung_DruckSpalten

	Lieferung_ID	Reihenfolge	ID	Tabellenkopf	Feld	Format	Breite
	6	4	4		Anzahl	Number	0,95
	6	5	5		Name_DE	String	1,00
	6	8	6	Ware	Ware	String	
	6	9	9	Einzelpreis	Preis	Currency6	1,10
	6	11	11	Steuer	Steuersatz	Percent	0,70
	6	12	12	Summe	AnzahlPreis	Currency	1,10
	7	4	4		Anzahl	Number	0,95
	7	5	5		Name_DE	String	1,00

Datenquelle

Tabelle: [tbl_Druck](#)

Ansicht: [viw_Lieferung_Spalten_Aenderung](#), [viw_Filter](#)

Datenziel

Makro: [FillTableCarryOver](#)

```

001 SELECT "b"."Lieferung_ID", "a"."Reihenfolge", "a"."ID", "a"."Tabellenkopf",
"a"."Feld", "a"."Format", COALESCE("a"."Breite",1) AS "Breite" FROM "tbl_Druck" AS
"a", "viw_Lieferung_Spalten_Aenderung" AS "b", "viw_Filter" WHERE "a"."ID" = 1 AND
"viw_Filter"."ShowRowNr" = TRUE
002 UNION
003 SELECT "b"."Lieferung_ID", "a"."Reihenfolge", "a"."ID", "a"."Tabellenkopf",
"a"."Feld", "a"."Format", COALESCE("a"."Breite",1) AS "Breite" FROM "tbl_Druck" AS

```

```

"a", "viw_Lieferung_Spalten_Aenderung" AS "b" WHERE "a"."ID" = 2 AND NOT
"b"."colStartDatum" IS NULL
004 UNION
005 SELECT "b"."Lieferung_ID", "a"."Reihenfolge", "a"."ID", "a"."Tabellenkopf",
"a"."Feld", "a"."Format", COALESCE("a"."Breite",1) AS "Breite" FROM "tbl_Druck" AS
"a", "viw_Lieferung_Spalten_Aenderung" AS "b" WHERE "a"."ID" = 3 AND NOT
"b"."colEndDatum" IS NULL
006 UNION
007 SELECT "b"."Lieferung_ID", "a"."Reihenfolge", "a"."ID", "a"."Tabellenkopf",
"a"."Feld", "a"."Format", COALESCE("a"."Breite",1) AS "Breite" FROM "tbl_Druck" AS
"a", "viw_Lieferung_Spalten_Aenderung" AS "b" WHERE "a"."ID" = 4
008 UNION
009 SELECT "b"."Lieferung_ID", "a"."Reihenfolge", "a"."ID", "a"."Tabellenkopf",
"a"."Feld", "a"."Format", COALESCE("a"."Breite",1) AS "Breite" FROM "tbl_Druck" AS
"a", "viw_Lieferung_Spalten_Aenderung" AS "b" WHERE "a"."ID" = 5 AND NOT
"b"."colAnzahlCode" IS NULL
010 UNION
011 SELECT "b"."Lieferung_ID", "a"."Reihenfolge", "a"."ID",
COALESCE("d"."Re_TKopf1", "a"."Tabellenkopf") AS "Tabellenkopf", "a"."Feld",
"a"."Format", NULL AS "Breite" FROM "tbl_Druck" AS "a",
"viw_Lieferung_Spalten_Aenderung" AS "b", "tbl_Lieferung_Aenderung" AS "c",
"tbl_Kunde" AS "d" WHERE "a"."ID" = 6 AND "b"."Lieferung_ID" = "c"."ID" AND
"c"."Kunde_ID" = "d"."ID"
012 UNION
013 SELECT "b"."Lieferung_ID", "a"."Reihenfolge", "a"."ID",
COALESCE("d"."Re_TKopf2", "a"."Tabellenkopf") AS "Tabellenkopf", "a"."Feld",
"a"."Format", COALESCE("a"."Breite",1) AS "Breite" FROM "tbl_Druck" AS "a",
"viw_Lieferung_Spalten_Aenderung" AS "b", "tbl_Lieferung_Aenderung" AS "c",
"tbl_Kunde" AS "d" WHERE "a"."ID" = 7 AND "b"."Lieferung_ID" = "c"."ID" AND
"c"."Kunde_ID" = "d"."ID" AND NOT "b"."colWarennummer" IS NULL
014 UNION
015 SELECT "b"."Lieferung_ID", "a"."Reihenfolge", "a"."ID",
COALESCE("d"."Re_TKopf3", "a"."Tabellenkopf") AS "Tabellenkopf", "a"."Feld",
"a"."Format", COALESCE("a"."Breite",1) AS "Breite" FROM "tbl_Druck" AS "a",
"viw_Lieferung_Spalten_Aenderung" AS "b", "tbl_Lieferung_Aenderung" AS "c",
"tbl_Kunde" AS "d" WHERE "a"."ID" = 8 AND "b"."Lieferung_ID" = "c"."ID" AND
"c"."Kunde_ID" = "d"."ID" AND NOT "b"."colWarenzusatz" IS NULL
016 UNION
017 SELECT "b"."Lieferung_ID", "a"."Reihenfolge", "a"."ID", "a"."Tabellenkopf",
"a"."Feld", "a"."Format", COALESCE("a"."Breite",1) AS "Breite" FROM "tbl_Druck" AS
"a", "viw_Lieferung_Spalten_Aenderung" AS "b" WHERE "a"."ID" = 9
018 UNION
019 SELECT "b"."Lieferung_ID", "a"."Reihenfolge", "a"."ID", "a"."Tabellenkopf",
"a"."Feld", "a"."Format", COALESCE("a"."Breite",1) AS "Breite" FROM "tbl_Druck" AS
"a", "viw_Lieferung_Spalten_Aenderung" AS "b" WHERE "a"."ID" = 10 AND NOT
"b"."colRabatt" IS NULL
020 UNION
021 SELECT "b"."Lieferung_ID", "a"."Reihenfolge", "a"."ID", "a"."Tabellenkopf",
"a"."Feld", "a"."Format", COALESCE("a"."Breite",1) AS "Breite" FROM "tbl_Druck" AS
"a", "viw_Lieferung_Spalten_Aenderung" AS "b" WHERE "a"."ID" = 11 AND NOT
"b"."colSteuersatz" IS NULL
022 UNION
023 SELECT "b"."Lieferung_ID", "a"."Reihenfolge", "a"."ID", "a"."Tabellenkopf",
"a"."Feld", "a"."Format", COALESCE("a"."Breite",1) AS "Breite" FROM "tbl_Druck" AS
"a", "viw_Lieferung_Spalten_Aenderung" AS "b" WHERE "a"."ID" = 12
024 UNION
025 SELECT "b"."Lieferung_ID", "a"."Reihenfolge", "a"."ID", "a"."Tabellenkopf",
"a"."Feld", "a"."Format", COALESCE("a"."Breite",1) AS "Breite" FROM "tbl_Druck" AS
"a", "viw_Lieferung_Spalten_Aenderung" AS "b", "tbl_Firma" AS "c" WHERE "a"."ID" =
13 AND "c"."Druck_NettoSteuerBruttoSum" = True
026 UNION
027 SELECT "b"."Lieferung_ID", "a"."Reihenfolge", "a"."ID", "a"."Tabellenkopf",
"a"."Feld", "a"."Format", COALESCE("a"."Breite",1) AS "Breite" FROM "tbl_Druck" AS
"a", "viw_Lieferung_Spalten_Aenderung" AS "b", "tbl_Firma" AS "c" WHERE "a"."ID" =
14 AND "c"."Druck_NettoSteuerBruttoSum" = True

```

Die Tabelle "tbl_Druck" hat 14 Datensätze. Für jeden Datensatz wird hier eine Abfrage gestellt. Anschließend werden die Informationen über **UNION** miteinander verbunden.

Für jede eventuell zur Verfügung stehende Spalte wird abgeklärt, ob sie in der Rechnung vorkommt. Dies ist aus der Ansicht "viw_Lieferung_Spalten_Aenderung" für viele Spalten zu erfahren. Für die Rechnungszeilennummer muss hier zusätzlich die Ansicht "viw_Filter" bemüht werden, da die Zeilennummer in den Formularen separat aus- und eingeschaltet werden kann.

In "Lieferung_ID" = '6' werden nur 6 Spalten angezeigt. Zwei Spalten werden keine Überschrift haben, bei den restlichen Spalten gibt es Spalten, die eine andere Überschrift als einen Feldnamen aus der betreffenden Ansicht "viw_Lieferung_Eingang_Aenderung" haben. Die Breitenangabe für die Ware bleibt leer. Sie wird durch eine anschließende Abfrage über die Summe der Spalten für die Rechnung ermittelt ($0,95 + 1 + 1,1 + 0,7 + 1,1 = 4,85$; $10 - 4,85 = 5,15$).

viw_Lieferung_Eingang_Aenderung

	ID	AusID	Lieferu...	MainSub	RowID	RechnungAusga...	Row	Titel	Anzahl	Anzahl_Code_ID	Name_...	Ware
▶	17	17	6	1	1	1	Main		1,0000	C62	ein	Neumotc
	18	18	6	1	2	2	Main		1,0000	C62	ein	Bemessu
	19	19	6	1	3	3	Main		-1,0000	C62	ein	Korrektur
	20	20	6	1	4	4	Main		1,0000	HUR	Stunde	Montage
	21	21	7	1	1	Seminar: [...]	Main		5,0000	DAY	Tag	Seminar
	22	22	7	1	2	Raumkosten Sch	Main		5,0000	DAY	Tag	Raumkos
	23	23	7	1	3	Reisekostenpau	Main		5,0000	XPP	Stücke	Reisekost
	24	24	7	1	4	Seminarunterlag	Main		11,0000	XPP	Stücke	Seminaru

Erster Teil der Ausgabe der Ansicht "viw_Lieferung_Eingang_Aenderung".

Datenquelle

Tabelle: [tbl_Eingang_Aenderung](#), [tbl_Anzahl_Code](#), [tbl_Lieferung_Aenderung](#), [tbl_Kunde](#)

Datenziel

Ansicht: [viw_Lieferung_Steuer_Aenderung](#)

Makro: [FillTableCarryOver](#), [ViewsErstellen](#)

```

001 SELECT "a"."ID", "a"."ID" AS "AusID", "a"."Lieferung_ID", 1 AS "MainSub",
002     RANK ( ) OVER ( PARTITION BY "a"."Lieferung_ID" ORDER BY "a"."ID" ) AS "RowID",
003     COALESCE ( "a"."LieferantZeilenID", RANK ( ) OVER ( PARTITION BY
004         "a"."Lieferung_ID" ORDER BY "a"."ID" ) ) AS "RechnungAusgang_ID",
005     COALESCE ( ( SELECT DISTINCT 'MainToSub' FROM "tbl_Eingang_Aenderung" WHERE
006         "Eingang_ID" = "a"."ID" ), 'Main' ) AS "Row",
007     CASE WHEN ( SELECT MAX( "ID" ) FROM "tbl_Eingang_Aenderung" WHERE "Eingang_ID" =
008         "a"."ID" ) > 0 THEN "a"."Ware" ELSE NULL END AS "Titel",
009     "a"."Anzahl", "a"."Anzahl_Code_ID",
010     CASE WHEN "a"."Anzahl" > 1 THEN COALESCE ( "tbl_Anzahl_Code"."Name_DE_Plural",
011         "tbl_Anzahl_Code"."Name_DE" ) ELSE "tbl_Anzahl_Code"."Name_DE" END AS
012         "Name_DE",
013     "a"."Ware", "a"."LieferantZeilenID" AS "Warennummer",
014     "a"."LieferantZeilenID" AS "Warenzusatz",
015     COALESCE ( "tbl_Kunde"."Re_TKopf1" || ': ', '' ) || "Ware" || COALESCE(' ' ||
016         COALESCE ( "tbl_Kunde"."Re_TKopf2" || ': ', '' ) || "LieferantZeilenID", '' )
017         || COALESCE(' ' || COALESCE ( "tbl_Kunde"."Re_TKopf3" || ': ', '' ) ||
018         "LieferantZeilenID", '' ) AS "xWare",
019     "Preis", "Rabatt", "Steuersatz", "Steuer_Code_ID", "a"."StartDatum",
020     "a"."EndDatum", "a"."KontoreferenzKunde",
021     ROUND("Anzahl" * "Preis" * (1 - COALESCE("Rabatt",0)),2) AS "AnzahlPreis",
022     ROUND((ROUND("Anzahl" * "Preis" * (1 - COALESCE("Rabatt",0)),2) *
023         "Steuersatz"),2) AS "Steuer",

```

```

013 ROUND("Anzahl" * "Preis" * (1 - COALESCE("Rabatt",0)),2) + ROUND((ROUND("Anzahl"
    * "Preis" * (1 - COALESCE("Rabatt",0)),2) * "Steuersatz"),2) AS "Brutto",
014 COALESCE ( "a"."EndDatum", "a"."StartDatum" ) AS "xEndDate",
015 "a"."Eingang_ID"
016 FROM "tbl_Eingang_Aenderung" AS "a", "tbl_Anzahl_Code", "tbl_Lieferung_Aenderung",
    "tbl_Kunde"
017 WHERE "a"."Anzahl_Code_ID" = "tbl_Anzahl_Code"."Code_ID"
018 AND "a"."Lieferung_ID" = "tbl_Lieferung_Aenderung"."ID"
019 AND "tbl_Lieferung_Aenderung"."Kunde_ID" = "tbl_Kunde"."ID"
020 AND "a"."Eingang_ID" IS NULL
021 UNION
022 SELECT "a"."Eingang_ID", "a"."ID" AS "AusID", "a"."Lieferung_ID", 2 AS "MainSub",
023 (SELECT COUNT("ID") FROM "tbl_Eingang_Aenderung" WHERE "ID" <= "a"."Eingang_ID"
    AND "Lieferung_ID" = "a"."Lieferung_ID" AND "Eingang_ID" IS NULL)||'.'||
    RANK() OVER (PARTITION BY "a"."Eingang_ID" ORDER BY "a"."ID") AS "RowID",
024 COALESCE((SELECT "LieferantZeilenID" FROM "tbl_Eingang_Aenderung" WHERE "ID" =
    "a"."Eingang_ID"), (SELECT COUNT("ID") FROM "tbl_Eingang_Aenderung" WHERE
    "ID" <= "a"."Eingang_ID" AND "Lieferung_ID" = "a"."Lieferung_ID" AND
    "Eingang_ID" IS NULL)||'.'||COALESCE("a"."LieferantZeilenID",RANK() OVER
    (PARTITION BY "a"."Eingang_ID" ORDER BY "a"."ID"))) AS "RechnungEingang_ID",
025 CASE WHEN "a"."ID" IN (SELECT "Eingang_ID" FROM "tbl_Eingang_Aenderung" WHERE
    "Eingang_ID" IS NOT NULL AND "Lieferung_ID" = "a"."Lieferung_ID") THEN
    'SubToSubSub' WHEN RANK() OVER (PARTITION BY "a"."Eingang_ID" ORDER BY
    "a"."ID") = (SELECT COUNT("ID") FROM "tbl_Eingang_Aenderung" WHERE
    "Eingang_ID" = "a"."Eingang_ID") THEN 'SubToMain' ELSE 'Sub' END AS "Row",
026 CASE WHEN (SELECT MAX("ID") FROM "tbl_Eingang_Aenderung" WHERE "Eingang_ID" =
    "a"."ID") > 0 THEN "Ware" ELSE NULL END AS "Titel",
027 "a"."Anzahl", "a"."Anzahl_Code_ID",
028 CASE WHEN "a"."Anzahl" > 1 THEN COALESCE("tbl_Anzahl_Code"."Name_DE_Plural",
    "tbl_Anzahl_Code"."Name_DE") ELSE "tbl_Anzahl_Code"."Name_DE" END
    AS "Name_DE",
029 "a"."Ware", "a"."LieferantZeilenID" AS "Warennummer",
    "a"."LieferantZeilenID" AS "Warenzusatz",
030 COALESCE ( "tbl_Kunde"."Re_TKopf1" || ': ', '' ) || "Ware" || COALESCE('| '||
    COALESCE ( "tbl_Kunde"."Re_TKopf2" || ': ', '' ) || "LieferantZeilenID", '' )
    || COALESCE('| '||COALESCE ( "tbl_Kunde"."Re_TKopf3" || ': ', '' ) ||
    "LieferantZeilenID", '' ) AS "xWare",
031 "Preis", "Rabatt", "Steuersatz", "Steuer_Code_ID", "a"."StartDatum",
    "a"."EndDatum", "a"."KontoreferenzKunde",
032 ROUND("Anzahl" * "Preis" * (1 - COALESCE("Rabatt",0)),2) AS "AnzahlPreis",
033 ROUND((ROUND("Anzahl" * "Preis" * (1 - COALESCE("Rabatt",0)),2) *
    "Steuersatz"),2) AS "Steuer",
034 ROUND("Anzahl" * "Preis" * (1 - COALESCE("Rabatt",0)),2) + ROUND((ROUND("Anzahl"
    * "Preis" * (1 - COALESCE("Rabatt",0)),2) * "Steuersatz"),2) AS "Brutto",
035 COALESCE("a"."EndDatum", "a"."StartDatum") AS "xEndDate",
036 "a"."Eingang_ID"
037 FROM "tbl_Eingang_Aenderung" AS "a",
038 (SELECT "ID" AS "NID", "Lieferung_ID" AS "NLieferung_ID" FROM
    "tbl_Eingang_Aenderung" WHERE "Eingang_ID" IS NULL) AS "b",
039 "tbl_Anzahl_Code", "tbl_Lieferung_Aenderung", "tbl_Kunde"
040 WHERE "a"."Lieferung_ID" = "b"."NLieferung_ID"
041 AND "a"."Eingang_ID" = "b"."NID"
042 AND "a"."Anzahl_Code_ID" = "tbl_Anzahl_Code"."Code_ID"
043 AND "a"."Lieferung_ID" = "tbl_Lieferung_Aenderung"."ID"
044 AND "tbl_Lieferung_Aenderung"."Kunde_ID" = "tbl_Kunde"."ID"
045 AND NOT "a"."Eingang_ID" IS NULL
046 UNION
047 SELECT "NEingang_ID", "a"."ID" AS "AusID", "a"."Lieferung_ID", 3 AS "MainSub",
048 (SELECT COUNT("ID") FROM "tbl_Eingang_Aenderung" WHERE "ID" <= "b"."NEingang_ID"
    AND "Lieferung_ID" = "a"."Lieferung_ID")||'.'||(SELECT COUNT("ID") FROM
    "tbl_Eingang_Aenderung" WHERE "Eingang_ID" = "b"."NEingang_ID")||'.'||RANK()
    OVER (PARTITION BY "a"."Eingang_ID" ORDER BY "a"."ID") AS "RowID",
049 COALESCE((SELECT "LieferantZeilenID" FROM "tbl_Eingang_Aenderung" WHERE "ID" =
    "b"."NEingang_ID"),(SELECT COUNT("ID") FROM "tbl_Eingang_Aenderung" WHERE
    "ID" <= "b"."NEingang_ID" AND "Lieferung_ID" = "a"."Lieferung_ID")||'.'||
    COALESCE((SELECT "LieferantZeilenID" FROM "tbl_Eingang_Aenderung" WHERE "ID"

```

```

= "a"."Eingang_ID"),(SELECT COUNT("ID") FROM "tbl_Eingang_Aenderung" WHERE
"Eingang_ID" = "b"."Neingang_ID"))||'. '||
COALESCE("a"."LieferantZeilenID",RANK() OVER (PARTITION BY "a"."Eingang_ID"
ORDER BY "a"."ID")) AS "RechnungEingang_ID",
050 CASE WHEN (RANK() OVER (PARTITION BY "a"."Eingang_ID" ORDER BY "a"."ID") =
(SELECT COUNT("ID") FROM "tbl_Eingang_Aenderung" WHERE "Eingang_ID" =
"a"."Eingang_ID")) AND ((SELECT MAX("ID") FROM "tbl_Eingang_Aenderung" WHERE
"Lieferung_ID" = "a"."Lieferung_ID" AND "Eingang_ID" = "b"."Neingang_ID") =
"Eingang_ID" ) THEN 'SubSubToMain' WHEN RANK() OVER (PARTITION BY
"a"."Eingang_ID" ORDER BY "a"."ID") = (SELECT COUNT("ID") FROM
"tbl_Eingang_Aenderung" WHERE "Eingang_ID" = "a"."Eingang_ID") THEN
'SubSubToSub' ELSE 'SubSub' END AS "Row",
051 CASE WHEN (SELECT MAX("ID") FROM "tbl_Eingang_Aenderung" WHERE "Eingang_ID" =
"a"."ID") > 0 THEN "Ware" ELSE NULL END AS "Titel",
052 "a"."Anzahl", "a"."Anzahl_Code_ID",
053 CASE WHEN "a"."Anzahl" > 1 THEN COALESCE("tbl_Anzahl_Code"."Name_DE_Plural",
"tbl_Anzahl_Code"."Name_DE") ELSE "tbl_Anzahl_Code"."Name_DE" END
AS "Name_DE",
054 "a"."Ware", "a"."LieferantZeilenID" AS "Warennummer", "a"."LieferantZeilenID"
AS "Warenzusatz",
055 COALESCE ( "tbl_Kunde"."Re_TKopf1" || ': ', '' ) || "Ware" || COALESCE(' ' ||
COALESCE ( "tbl_Kunde"."Re_TKopf2" || ': ', '' ) || "LieferantZeilenID", '' )
|| COALESCE(' ' || COALESCE ( "tbl_Kunde"."Re_TKopf3" || ': ', '' ) ||
"LieferantZeilenID", '' ) AS "xWare",
056 "Preis", "Rabatt", "Steuersatz", "Steuer_Code_ID", "a"."StartDatum",
"a"."EndDatum", "a"."KontoreferenzKunde",
057 ROUND("Anzahl" * "Preis" * (1 - COALESCE("Rabatt",0)),2) AS "AnzahlPreis",
058 ROUND((ROUND("Anzahl" * "Preis" * (1 - COALESCE("Rabatt",0)),2) *
"Steuersatz"),2) AS "Steuer",
059 ROUND("Anzahl" * "Preis" * (1 - COALESCE("Rabatt",0)),2) + ROUND((ROUND("Anzahl"
* "Preis" * (1 - COALESCE("Rabatt",0)),2) * "Steuersatz"),2) AS "Brutto",
060 COALESCE("a"."EndDatum", "a"."StartDatum") AS "xEndDate",
061 "a"."Eingang_ID"
062 FROM "tbl_Eingang_Aenderung" AS "a",
063 (SELECT "ID" AS "NID", "Lieferung_ID" AS "NLieferung_ID", "Eingang_ID" AS
"NEingang_ID" FROM "tbl_Eingang_Aenderung" WHERE "Eingang_ID" IS NOT NULL)
AS "b",
064 "tbl_Anzahl_Code", "tbl_Lieferung_Aenderung", "tbl_Kunde"
065 WHERE "a"."Lieferung_ID" = "b"."NLieferung_ID"
066 AND "a"."Eingang_ID" = "b"."NID"
067 AND "a"."Anzahl_Code_ID" = "tbl_Anzahl_Code"."Code_ID"
068 AND "a"."Lieferung_ID" = "tbl_Lieferung_Aenderung"."ID"
069 AND "tbl_Lieferung_Aenderung"."Kunde_ID" = "tbl_Kunde"."ID"
070 AND NOT "a"."Eingang_ID" IS NULL

```

Diese Ansicht dient zum Druck der erstellten geänderten Lieferung. Sie ist vom Prinzip her genauso aufgebaut wie die [viw_Rechnung_Ausgang](#).

viw_Lieferung_Spalten_Aenderung

	Lieferung_ID	colRabatt	colSteuersatz	colStartDatum	colEndDatum	colWarennumm...	colWarenzusatz	colAnzahlCode
▶	6		0,19					C62
	7							DAY

Ausgabe der Ansicht "viw_Lieferung_Spalten_Aenderung". Die Ansicht dient dazu, notwendige Felder für den Druck zu identifizieren

Datenquelle	
Tabelle:	tbl_Eingang_Aenderung , tbl_Firma

Datenziel
Ansicht: <i>viw_Lieferung_DruckSpalten</i>
Makro: <i>FillTableCarryOver, ViewsErstellen</i>

```

001 SELECT "a"."Lieferung_ID",
002     SUM("Rabatt") AS "colRabatt",
003     CASE WHEN (SELECT MIN("Steuersatz") FROM "tbl_Eingang_Aenderung" WHERE
                "Lieferung_ID" = "a"."Lieferung_ID") = (SELECT MAX("Steuersatz") FROM
                "tbl_Eingang_Aenderung" WHERE "Lieferung_ID" = "a"."Lieferung_ID") THEN NULL
            ELSE MAX("Steuersatz") END AS "colSteuersatz",
004     MIN("StartDatum") AS "colStartDatum",
005     MIN("EndDatum") AS "colEndDatum",
006     NULL AS "colWarennummer",
007     NULL AS "colWarenzusatz",
008     CASE WHEN (SELECT COUNT("ID") FROM "tbl_Eingang_Aenderung" WHERE "Lieferung_ID"
                = "a"."Lieferung_ID" AND NOT "Anzahl_Code_ID" = 'XPP') > 0 OR (SELECT
                "frmKunde_AnzahlCode" FROM "tbl_Firma") = TRUE THEN MIN("Anzahl_Code_ID")
            ELSE NULL END AS "colAnzahlCode"
009 FROM "tbl_Eingang_Aenderung" AS "a"
010 GROUP BY "a"."Lieferung_ID"

```

Diese Ansicht dient zum Druck der erstellten geänderten Lieferung. Sie ist vom Prinzip her genauso aufgebaut wie die *viw_Rechnung_Spalten*.

viw_Lieferung_Steuer

	Lieferung_ID	Sort	Steuersatz	Steuer_Code_ID	Netto	Steuer	Brutto
▶	6	1 - Rechnungszeilen	0,00	Z	-100,0000	0	-100
	6	1 - Rechnungszeilen	0,19	S	1600,000	304	1904

Ausgabe der Ansicht "viw_Lieferung_Steuer". Die Felder für die Summierung sind hier mit allen Nachkommastellen angegeben, damit nicht über eine Währungsformatierung der tatsächliche Wert unsichtbar wird.

Datenquelle
Tabelle: <i>tbl_Eingang, tbl_Lieferung_Zusatz</i>

Datenziel
Ansicht: <i>viw_Lieferung_Summe</i>
Makro: <i>ViewsErstellen</i>

```

001 SELECT "Lieferung_ID", '1 - Rechnungszeilen' AS "Sort", "Steuersatz",
002     SUM( "AnzahlPreis" ) AS "Netto",
003     ROUND(SUM( "AnzahlPreis" ) * ( 1 + "Steuersatz" ),2) - SUM( "AnzahlPreis" )
            AS "Steuer",
004     ROUND(SUM( "AnzahlPreis" ) * ( 1 + "Steuersatz" ),2) AS "Brutto"
005 FROM
006     ROUND(( SELECT "Anzahl" * "Preis" - COALESCE ( "Rabatt", 0 ),2)
            AS "AnzahlPreis",
            "Steuersatz", "Lieferung_ID"
007 FROM "tbl_Eingang" WHERE "Eingang_ID" IS NULL)
008 AS "a"
009 GROUP BY "Lieferung_ID", "Steuersatz"
010 UNION
011 SELECT "Lieferung_ID", '2 - Lieferung pauschal' AS "Sort", "Steuersatz",
012     ROUND(SUM( "Betrag" ),2) AS "Netto",
013     ROUND(SUM( "Betrag" ) * ( 1 + "Steuersatz" ),2) - ROUND(SUM( "Betrag" ),2)
            AS "Steuer",
014     ROUND(SUM( "Betrag" ) * ( 1 + "Steuersatz" ),2) AS "Brutto"

```

```
016 FROM "tbl_Lieferung_Zusatz"
017 GROUP BY "Lieferung_ID", "Steuersatz"
```

In der Unterabfrage von Zeile 6 bis Zeile 8 wird für die "tbl_Eingang" aus "Anzahl" und "Preis" sowie gegebenenfalls dem Abzug von "Rabatt" der Nettobetrag für jede Tabellenzeile erstellt. Dieser Nettobetrag ist über "AnzahlPreis" in der darüber liegenden Abfrage abrufbar.

Wäre der Rabatt ein leeres Feld (**NULL**), so würde "AnzahlPreis" auch leer sein. Deshalb wird mit **COALESCE** hier bei einem leeren Feld "Rabatt" stattdessen '0' genutzt.

In den Zeilen 1 bis 5 wird diese Unterabfrage genutzt, um "AnzahlPreis" für die gesamte Lieferung auf zu summieren. Die einfache Summe ergibt des Nettopreis, die Summe multipliziert mit dem Prozentsatz der Steuer ergibt die fällige Steuer und die Summe multipliziert mit (1 + "Steuersatz") schließlich den Bruttobetrag. Für die Berechnung wird nur die Hauptrechnungszeile ausgewertet. Alle untergeordneten Rechnungszeilen werden in Zeile 8 ausgeschlossen, wie es XRechnung-Extension auch vorsieht.

Summiert wird hier nach der Gruppierung in Zeile 10 für jede Lieferung und dort dann in Untergruppen für jeden Steuersatz, wenn eben mehr als ein Steuersatz existiert.

An diese Abfrage wird über **UNION** (Zeile 11) eine weitere Abfrage angehängt. Hier kann direkt auf die betreffende Tabelle "tbl_Lieferung_Zusatz" Bezug genommen werden, da dort die Beträge nicht von einem Rabatt oder einer Anzahl wie in "tbl_Eingang" abhängig sind. Das Verfahren für diese Abfrage ist bei der Summierung gleich. Es wird lediglich in der Spalte "Sort" ein anderer Text ausgegeben, damit die Daten voneinander getrennt erscheinen und entsprechend zugeordnet werden können.

✓ Hinweis

Werden Abfragen mit **UNION** verknüpft, so erfolgt die Sortierung der Elemente, wenn nicht anders angegeben, nach der Reihenfolge der Spalten. Im obigen Beispiel also zuerst nach "Rechnung_ID", dann nach "Sort", anschließend nach dem "Steuersatz". Standardmäßig schließt **UNION** die Anzeige von Duplikaten ausfallen. Sollte also in einer der verknüpften Abfragen eine Zeile gleich einer der vorhergehenden Zeilen sein, dann wird der Inhalt nicht angezeigt. Diese Funktion wird hier aber nicht benötigt, da sowieso alle Zeilen der beiden betroffenen Abfragen einen unterschiedlichen Inhalt haben.

viw_Lieferung_Steuer_Aenderung

	Lieferung_ID	Sort	Steuersatz	Steuer_Code_ID	Netto	Steuer	Brutto
▶	6	1 - Rechnungszeilen	0,00	Z	-100,0000	0	-100
	6	1 - Rechnungszeilen	0,19	S	1600,000	304	1904

Ausgabe der Ansicht "viw_Lieferung_Steuer". Die Felder für die Summierung sind hier mit allen Nachkommastellen angegeben, damit nicht über eine Währungsformatierung der tatsächliche Wert unsichtbar wird.

Datenquelle

Tabelle: [tbl_Lieferung_Zusatz_Aenderung](#)

Ansicht: [viw_Lieferung_Eingang_Aenderung](#)

Datenziel

Ansicht: [viw_Lieferung_Summe_Aenderung](#)

Makro: [ViewsErstellen](#)

```
001 SELECT "Lieferung_ID", '1 - Rechnungszeilen' AS "Sort", "Steuersatz",
        "Steuer_Code_ID",
```

```

002     SUM( "AnzahlPreis" ) AS "Netto",
003     SUM( "Steuer" ) AS "Steuer",
004     SUM( "Brutto" ) AS "Brutto"
005 FROM
006     ( SELECT "AnzahlPreis", "Steuer", "Brutto", "Steuersatz", "Steuer_Code_ID",
              "Lieferung_ID" FROM "viw_Lieferung_Eingang_Aenderung" WHERE "Eingang_ID" IS NULL
            ) AS "a"
007 GROUP BY "Lieferung_ID", "Steuersatz", "Steuer_Code_ID"
008 UNION
009 SELECT "Lieferung_ID", '2 - Lieferung pauschal' AS "Sort", "Steuersatz",
         "Steuer_Code_ID",
010     SUM( "Betrag" ) AS "Netto",
011     SUM( "Betrag" ) * "Steuersatz" AS "Steuer",
012     SUM( "Betrag" ) * ( 1 + "Steuersatz" ) AS "Brutto"
013 FROM "tbl_Lieferung_Zusatz_Aenderung"
014 GROUP BY "Lieferung_ID", "Steuersatz", "Steuer_Code_ID"

```

Diese Ansicht dient zum Druck der erstellten geänderten Lieferung. Sie ist vom Prinzip her genauso aufgebaut wie die [viw_Rechnung_Steuer](#).

viw_Lieferung_Summe

	Lieferun...	Sort	Steuers...	Steuer_Cod...	Netto	Steuer	Brutto	BereitsGezahlt	NochOffen
▶	6	1 - Rechnungszeilen	0,00	Z	-100,000	0	-100		
	6	1 - Rechnungszeilen	0,19	S	1600,000	304	1904		
	6	3 - Summe Rechnungszeilen			1500,000	304	1804		
	6	5 - Summe Steuersatz	0,00	Z	-100,000	0	-100		
	6	5 - Summe Steuersatz	0,19	S	1600,000	304	1904		
	6	6 - Gesamt			1500,000	304	1804	0	1804

Datenquelle

Ansicht: [viw_Lieferung_Steuer](#)

Datenziel

Formular: [frm_Lieferung](#)

Makros: [ViewsErstellen](#)

```

001 SELECT "viw_Lieferung_Steuer".*, NULL AS "BereitsGezahlt", NULL AS "NochOffen"
002 FROM "viw_Lieferung_Steuer"
003 UNION
004 SELECT "Lieferung_ID", '3 - Summe Rechnungszeilen' AS "Sort", NULL AS "Steuersatz",
         NULL AS "Steuer_Code_ID", SUM( "Netto" ) AS "Netto",
         SUM( "Steuer" ) AS "Steuer", SUM( "Brutto" ) AS "Brutto",
         NULL AS "BereitsGezahlt", NULL AS "NochOffen"
005 FROM "viw_Lieferung_Steuer"
006 WHERE LEFT("Sort",1) = '1'
007 GROUP BY "Lieferung_ID"
008 UNION
009 SELECT "Lieferung_ID", '4 - Summe Rechnung pauschal' AS "Sort",
         NULL AS "Steuersatz", NULL AS "Steuer_Code_ID", SUM( "Netto" ) AS "Netto",
         SUM( "Steuer" ) AS "Steuer", SUM( "Brutto" ) AS "Brutto",
         NULL AS "BereitsGezahlt", NULL AS "NochOffen"
010 FROM "viw_Lieferung_Steuer"
011 WHERE LEFT("Sort",1) = '2'
012 GROUP BY "Lieferung_ID"
013 UNION
014 SELECT "Lieferung_ID", '5 - Summe Steuersatz' AS "Sort", "Steuersatz",
         "Steuer_Code_ID", SUM( "Netto" ) AS "Netto", SUM( "Steuer" ) AS "Steuer",
         SUM( "Brutto" ) AS "Brutto", NULL AS "BereitsGezahlt", NULL AS "NochOffen"

```

```

015 FROM "viw_Lieferung_Steuer"
016 GROUP BY "Lieferung_ID", "Steuersatz", "Steuer_Code_ID"
017 UNION
018 SELECT "Lieferung_ID", '6 - Gesamt' AS "Sort", NULL AS "Steuersatz",
      NULL AS "Steuer_Code_ID", SUM( "Netto" ) AS "Netto",
      SUM( "Steuer" ) AS "Steuer", SUM( "Brutto" ) AS "Brutto",
      (SELECT "BereitsGezahlt" FROM "tbl_Lieferung" WHERE "ID" = "a"."Lieferung_ID") +
      COALESCE((SELECT SUM("Betrag") FROM "tbl_Lieferung_Zahlung" WHERE
      "Lieferung_ID" = "a"."Lieferung_ID"),0) AS "BereitsGezahlt",
      ROUND(SUM( "Brutto" ) - COALESCE((SELECT "BereitsGezahlt" FROM "tbl_Lieferung"
      WHERE "ID" = "a"."Lieferung_ID") + COALESCE((SELECT SUM("Betrag") FROM
      "tbl_Lieferung_Zahlung" WHERE "Lieferung_ID" = "a"."Lieferung_ID"),0),0),2)
      AS "NochOffen"
019 FROM "viw_Lieferung_Steuer" AS "a"
020 GROUP BY "Lieferung_ID"

```

In Zeile 1 wird lediglich die vorher erstellte Ansicht "viw_Lieferung_Steuer" noch einmal eingelesen.

Von Zeile 3 bis 8 wird eine Abfrage an "viw_Lieferung_Steuer" gestellt, die den gesamten Inhalt nach der "Lieferung_ID" und dem "Steuersatz" zusammenfasst. So kommen Zwischensummen für die Steuersätze zustande.

Von Zeile 10 bis Zeile 15 wird dann "viw_Lieferung_Steuer" nur nach der "Lieferung_ID" gruppiert und summiert. So steht in der letzten Zeile zu der jeweiligen Lieferung die Gesamtsumme in Netto, von den Steuern her und in Brutto.

viw_Lieferung_Summe_Aenderung

	Lieferung_ID	Sort	Steuersatz	Steuer_Code_ID	Netto	Steuer	Brutto	BereitsGezahlt	NochOffen
▶	6	1 - Rech	0,00	Z	-100,0000	0	-100		
	6	1 - Rech	0,19	S	1600,0000	304	1904		
	6	3 - Sumr			1500,0000	304	1804		
	6	5 - Sumr	0,00	Z	-100,0000	0	-100		
	6	5 - Sumr	0,19	S	1600,0000	304	1904		
	6	6 - Gesar			1500,0000	304	1804	1804	0
	7	1 - Rech	0,19	S	8870,0000	1685,3	10555,3		
	7	3 - Sumr			8870,0000	1685,3	10555,3		
	7	5 - Sumr	0,19	S	8870,0000	1685,3	10555,3		
	7	6 - Gesar			8870,0000	1685,3	10555,3	0	10555,3

Datenquelle

Ansicht: [viw_Lieferung_Steuer_Aenderung](#)

Datenziel

Ansicht: [viw_Lieferung_Aenderung](#), [qry_Lieferung_Zahlung_Aenderung](#)

Makro: [FillTableCarryOver](#), [ViewsErstellen](#)

```

001 SELECT "viw_Lieferung_Steuer_Aenderung".*, NULL AS "BereitsGezahlt",
      NULL AS "NochOffen"
002 FROM "viw_Lieferung_Steuer_Aenderung"
003 UNION
004 SELECT "Lieferung_ID", '3 - Summe Rechnungszeilen' AS "Sort", NULL AS "Steuersatz",
      NULL AS "Steuer_Code_ID", SUM( "Netto" ) AS "Netto",
      SUM( "Steuer" ) AS "Steuer", SUM( "Brutto" ) AS "Brutto",
      NULL AS "BereitsGezahlt", NULL AS "NochOffen"

```

```

005 FROM "viw_Lieferung_Steuer_Aenderung"
006 WHERE LEFT("Sort",1) = '1'
007 GROUP BY "Lieferung_ID"
008 UNION
009 SELECT "Lieferung_ID", '4 - Summe Rechnung pauschal' AS "Sort",
      NULL AS "Steuersatz", NULL AS "Steuer_Code_ID", SUM( "Netto" ) AS "Netto",
      SUM( "Steuer" ) AS "Steuer", SUM( "Brutto" ) AS "Brutto",
      NULL AS "BereitsGezahlt", NULL AS "NochOffen"
010 FROM "viw_Lieferung_Steuer_Aenderung"
011 WHERE LEFT("Sort",1) = '2'
012 GROUP BY "Lieferung_ID"
013 UNION
014 SELECT "Lieferung_ID", '5 - Summe Steuersatz' AS "Sort", "Steuersatz",
      "Steuer_Code_ID", SUM( "Netto" ) AS "Netto", SUM( "Steuer" ) AS "Steuer",
      SUM( "Brutto" ) AS "Brutto", NULL AS "BereitsGezahlt", NULL AS "NochOffen"
015 FROM "viw_Lieferung_Steuer_Aenderung"
016 GROUP BY "Lieferung_ID", "Steuersatz", "Steuer_Code_ID"
017 UNION
018 SELECT "Lieferung_ID", '6 - Gesamt' AS "Sort", NULL AS "Steuersatz",
      NULL AS "Steuer_Code_ID", SUM( "Netto" ) AS "Netto",
      SUM( "Steuer" ) AS "Steuer", SUM( "Brutto" ) AS "Brutto",
      (SELECT "BereitsGezahlt" FROM "tbl_Lieferung_Aenderung" WHERE "ID" =
      "a"."Lieferung_ID") + COALESCE((SELECT SUM("Betrag") FROM
      "tbl_Lieferung_Zahlung_Aenderung" WHERE "Lieferung_ID" =
      "a"."Lieferung_ID"),0) AS "BereitsGezahlt",
      ROUND(SUM( "Brutto" ) - COALESCE((SELECT "BereitsGezahlt" FROM
      "tbl_Lieferung_Aenderung" WHERE "ID" = "a"."Lieferung_ID") + COALESCE((SELECT
      SUM("Betrag") FROM "tbl_Lieferung_Zahlung_Aenderung" WHERE "Lieferung_ID" =
      "a"."Lieferung_ID"),0),0),2) AS "NochOffen"
019 FROM "viw_Lieferung_Steuer_Aenderung" AS "a"
020 GROUP BY "Lieferung_ID"

```

Diese Ansicht dient zum Druck der erstellten geänderten Lieferung. Sie ist vom Prinzip her genauso aufgebaut wie die [viw_Rechnung_Summe](#).

viw_Rechnung

	Rechnungsdatum	ZahlungBis	KundeZeichen	Rechnungsnummer	Netto	Steuer	Brutto
▶	11.11.23	25.11.23	-	R-2023-00000001	4095,9300	768,2283	4864,1583

Die Ansicht "viw_Rechnung" enthält sehr viele Felder, hier nur einige Felder mit u.a. dem berechneten Datum "ZahlungBis" und der zusammengestellten "Rechnungsnummer". Seit Version V2408_032 die Berechnung von Netto, Steuer und Brutto auf die 2. Nachkommastelle gerundet.

Felder in Bezug zur XRechnung: Kombination für Zahlungsnotiz vorhergehender Rechnungen [BT-20], "RechnungsnummerMitZusatz" [BT-25], "Rechnungsdatum" [BT-26], "Kundennummer" [BT-46]

Datenquelle	
Tabelle:	tbl_Kunde , tbl_Rechnung , tbl_Mitarbeiter , tbl_Land , tbl_Firma , tbl_Steuer_Anmerkung
Ansicht:	viw_Rechnung_Summe
Datenziel	
Formular:	frm_Rechnung , frm_Zusatz
Abfrage:	qry_Konto , qry_Rechnung_Zahlung , qry_frmZusatz-Rechnung
Makro:	FillTableCarryOver , SaveXRechnung , SaveZUGFeRD , ListenfeldfilterRechnungsreferenz , ViewsErstellen

```

001 SELECT "d"."Firma", "d"."StraßeNr", "d"."Plz", "d"."Ort", "d"."E-Mail",
"d"."Website", "d"."Telefon", "d"."UST-IdNr", "d"."Firma_Registriert",
002 "d"."Bank", "d"."IBAN", "d"."BIC", "d"."RegisterE1",
CASE WHEN NOT "d"."Steuer_Anmerkung_ID" IS NULL AND
"viw_Rechnung_Summe"."Steuer" = 0 THEN (SELECT "Anmerkung" FROM
"tbl_Steuer_Anmerkung" WHERE "ID" = "d"."Steuer_Anmerkung_ID") ELSE
003 "d"."RegisterE2" END AS "RegisterE2",
004 "d"."RegisterE3",
REPLACE(COALESCE("a"."Zahlungsnotiz_1","d"."Zahlungsnotiz"))||
COALESCE(ASCII_CHAR(13)||"a"."Zahlungsnotiz_2",'')||COALESCE(ASCII_CHAR(13)||
CASE WHEN "tbl_Kunde"."Lastschrift" THEN 'Überweisen Sie nicht. Die Zahlung
erfolgt per Lastschrift.' ELSE NULL END, ''), '?Datum?', RIGHT('0' || EXTRACT(DAY
FROM "a"."Datum" + "d"."ZahlungTage"), 2) || '.' || RIGHT('0' ||
EXTRACT(MONTH FROM "a"."Datum" + "tbl_Firma"."ZahlungTage"), 2) || '.' ||
EXTRACT(YEAR FROM "a"."Datum" + "d"."ZahlungTage"))
AS "Zahlungsnotiz",
"d"."ZahlungTage",
005 COALESCE("d"."Zahlungsempfänger", "d"."Firma_Registriert")
AS "Zahlungsempfänger",
006 "d"."Logo", "d"."Gläubigernummer", "d"."SteuerNr",
007 CASE WHEN NOT "d"."Steuer_Anmerkung_ID" IS NULL AND
"viw_Rechnung_Summe"."Steuer" = 0 THEN (SELECT "Anmerkung" FROM
"tbl_Steuer_Anmerkung" WHERE "ID" = "d"."Steuer_Anmerkung_ID") ELSE NULL END
AS "Steuerfreiheit",
008 "tbl_Mitarbeiter"."Name", "tbl_Mitarbeiter"."Geschlecht",
009 "tbl_Mitarbeiter"."E-Mail" AS "E-Mail_Mitarbeiter",
010 "tbl_Mitarbeiter"."Telefon" AS "Telefon_Mitarbeiter",
011 'K-' || RIGHT('000000' || "tbl_Kunde"."ID", 6) AS "Kundenummer",
012 "tbl_Kunde"."Firma" AS "Firma_Kunde", "tbl_Kunde"."AnsprechP",
"tbl_Kunde"."AnsprechPGeschlecht", "tbl_Kunde"."StraßeNr" AS "StraßeNr_Kunde",
"tbl_Kunde"."Plz" AS "Plz_Kunde", "tbl_Kunde"."Ort" AS "Ort_Kunde",
"tbl_Kunde"."Land" AS "LandKurz",
013 UPPER("tbl_Land"."Land") AS "Land",
"tbl_Kunde"."E-Mail" AS "E-Mail_Kunde",
"tbl_Kunde"."Telefon" AS "Telefon_Kunde",
014 "tbl_Kunde"."Firma" || ASCII_CHAR(13) || "tbl_Kunde"."StraßeNr" ||
ASCII_CHAR(13) || "tbl_Kunde"."Plz" || ' ' || "tbl_Kunde"."Ort" ||
CASE WHEN "tbl_Kunde"."Land" = 'DE' OR "tbl_Kunde"."Land" IS NULL THEN ''
ELSE ASCII_CHAR(13) || UPPER("tbl_Land"."Land") END
AS "Anschrift_Kunde",
015 "tbl_Kunde"."UST-IdNr" AS "UST-IdNr_Kunde",
"tbl_Kunde"."RegisterID" AS "RegisterID_Kunde",
ASCII_CHAR(13) || "tbl_Kunde"."DruckZusatz" AS "DruckZusatz_Kunde",
016 CASE WHEN "tbl_Kunde"."Lastschrift" THEN "tbl_Kunde"."IBAN" ELSE NULL END AS
"IBAN_Kunde",
017 "tbl_Kunde"."Mandatsreferenznummer", "a"."ID" AS "Rechnung_ID",
018 RIGHT('0' || EXTRACT(DAY FROM "a"."Datum"), 2) || '.' || RIGHT('0' || EXTRACT(MONTH FROM
"a"."Datum"), 2) || '.' || EXTRACT(YEAR FROM "a"."Datum") AS "RechnungsdatumD",
019 "a"."Datum" AS "Rechnungsdatum",
020 "a"."Datum" + "tbl_Firma"."ZahlungTage" AS "ZahlungBis",
021 "a"."KundeZeichen",
022 'R-' || "c"."Jahr" || '-' || RIGHT('0000000000' || "c"."lfdJahr",
COALESCE("tbl_Firma"."ReStellen", 6)) AS "Rechnungsnummer",
023 'R-' || "c"."Jahr" || '-' || RIGHT('0000000000' || "c"."lfdJahr",
COALESCE("d"."ReStellen", 6)) || COALESCE('_', 'a"."ReNummerAdd', '')
AS "RechnungsnummerMitZusatz",
024 "c"."lfdJahr", "viw_Rechnung_Summe"."Netto", "viw_Rechnung_Summe"."Steuer",
"viw_Rechnung_Summe"."Brutto", "viw_Rechnung_Summe"."BereitsGezahlt",
"viw_Rechnung_Summe"."NochOffen", "a"."Rechnungstyp_Code_ID", (SELECT "Name_DE"
FROM "tbl_Rechnungstyp_Code" WHERE "Code_ID" = "a"."Rechnungstyp_Code_ID") AS
"Rechnungstyp", "a"."LieferDatum",
025 CASE WHEN "a"."LieferDatum" IS NULL THEN NULL ELSE ASCII_CHAR(9) || 'Lieferdatum:'
END AS "tblLieferdatumD",
026 RIGHT('0' || EXTRACT(DAY FROM "a"."LieferDatum"), 2) || '.' || RIGHT('0' || EXTRACT(MONTH
FROM "a"."LieferDatum"), 2) || '.' || EXTRACT(YEAR FROM "a"."LieferDatum") AS
"LieferdatumD",

```

```

027 "l"."Firma" AS "Firma_KundeLiefer", "l"."StraßeNr" AS "StraßeNr_KundeLiefer",
"l"."Plz" AS "Plz_KundeLiefer", "l"."Ort" AS "Ort_KundeLiefer", "l"."Land" AS
"LandKurzLiefer",
028 (SELECT UPPER("Land") FROM "tbl_Land" WHERE "ID" = "l"."Land") AS "LandLiefer",
029 CASE WHEN "l"."Plz" IS NULL THEN NULL ELSE ASCII_CHAR(13)||'Abweichende
Lieferadresse:' END AS "lblAnschrift_Lieferadresse",
030 ASCII_CHAR(9)||"l"."Firma"||COALESCE(ASCII_CHAR(13)||ASCII_CHAR(9)||
"l"."StraßeNr",')||ASCII_CHAR(13)||ASCII_CHAR(9)||"l"."Plz"||' '||
"l"."Ort"||
CASE WHEN "l"."Land" = 'DE' OR "l"."Land" IS NULL THEN '' ELSE
ASCII_CHAR(13)||ASCII_CHAR(9)||UPPER((SELECT "Land" FROM "tbl_Land" WHERE
"ID" = "l"."Land")) END AS "Anschrift_Lieferadresse",
031 "a"."StartDatum",
032 CASE WHEN "a"."StartDatum" IS NULL THEN NULL ELSE ASCII_CHAR(13)||'von:' END
AS "lblStartDatumD",
033 RIGHT('0' || EXTRACT(DAY FROM "a"."StartDatum"),2) || '.' || RIGHT('0' || EXTRACT(MONTH
FROM "a"."StartDatum"),2) || '.' || EXTRACT(YEAR FROM "a"."StartDatum") AS
"StartDatumD",
034 "a"."EndDatum",
035 CASE WHEN "a"."EndDatum" IS NULL THEN NULL ELSE ASCII_CHAR(9)||'bis:' END
AS "lblEndDatumD",
036 RIGHT('0' || EXTRACT(DAY FROM "a"."EndDatum"),2) || '.' || RIGHT('0' || EXTRACT(MONTH
FROM "a"."EndDatum"),2) || '.' || EXTRACT(YEAR FROM "a"."EndDatum")
AS "EndDatumD",
037 CASE WHEN "a"."ProjektNr" IS NULL THEN NULL ELSE
ASCII_CHAR(13)||'Projektnummer:' END AS "lblProjektNr",
038 "a"."ProjektNr",
039 CASE WHEN "a"."VertragsNr" IS NULL THEN NULL WHEN "a"."ProjektNr" IS NULL THEN
ASCII_CHAR(13)||'Vertragsnummer:' ELSE ASCII_CHAR(9)||'Vertragsnummer:' END
AS "lblVertragsNr",
040 "a"."VertragsNr",
041 CASE WHEN "a"."BestellNr" IS NULL THEN NULL WHEN ("a"."ProjektNr" IS NULL AND
"a"."VertragsNr" IS NULL) OR (NOT "a"."ProjektNr" IS NULL AND NOT
"a"."VertragsNr" IS NULL) THEN ASCII_CHAR(13)||'Bestellnummer:' ELSE
ASCII_CHAR(9)||'Bestellnummer:' END AS "lblBestellNr",
042 "a"."BestellNr",
043 CASE WHEN "a"."AuftragsNr" IS NULL THEN NULL WHEN ("a"."ProjektNr" IS NULL AND
"a"."VertragsNr" IS NULL AND "a"."BestellNr" IS NULL) OR (NOT "a"."ProjektNr"
IS NULL AND NOT "a"."VertragsNr" IS NULL AND "a"."BestellNr" IS NULL) OR (NOT
"a"."ProjektNr" IS NULL AND "a"."VertragsNr" IS NULL AND NOT "a"."BestellNr"
IS NULL) OR ("a"."ProjektNr" IS NULL AND NOT "a"."VertragsNr" IS NULL AND NOT
"a"."BestellNr" IS NULL) THEN ASCII_CHAR(13)||'Auftragsnummer:' ELSE
ASCII_CHAR(9)||'Auftragsnummer:' END AS "lblAuftragsNr",
044 "a"."AuftragsNr",
045 CASE WHEN "a"."AusschreibungsNr" IS NULL THEN NULL ELSE
ASCII_CHAR(13)||'Ausschreibungsnummer:' END AS "lblAusschreibungsNr",
046 "a"."AusschreibungsNr",
047 CASE WHEN "a"."ObjektNr" IS NULL THEN NULL WHEN ("a"."AusschreibungsNr" IS NULL)
THEN ASCII_CHAR(13)||'Objektnummer:' ELSE ASCII_CHAR(9)||'Objektnummer:' END
AS "lblObjektNr",
048 "a"."ObjektNr",
049 CASE WHEN "a"."KontoreferenzKunde" IS NULL THEN NULL ELSE
ASCII_CHAR(13)||'Kontoreferenz:' END AS "lblKontoreferenzKunde",
050 "a"."KontoreferenzKunde",
051 CASE WHEN "tbl_Kunde"."Lieferantennummer" IS NULL THEN NULL ELSE
ASCII_CHAR(13)||'Lieferantennummer:' END AS "lblLieferantennummer",
052 COALESCE(ASCII_CHAR(13)||"a"."Anmerkung",') || COALESCE(ASCII_CHAR(13)||CASE
WHEN NOT "d"."Steuer_Anmerkung_ID" IS NULL AND "vwi_Rechnung_Summe"."Steuer"
= 0 THEN (SELECT "Anmerkung" FROM "tbl_Steuer_Anmerkung" WHERE "ID" =
"d"."Steuer_Anmerkung_ID") ELSE NULL END, '') AS "Rechnung_Anmerkung",
053 (SELECT LIST("Rechnung_ID") FROM "tbl_rel_Rechnung" WHERE "Main_Rechnung_ID" =
"a"."ID") AS "VorRechnungen_ID"
054 FROM "tbl_Kunde"
055 LEFT JOIN "tbl_Rechnung" AS "a" ON "a"."Kunde_ID" = "tbl_Kunde"."ID"
056 LEFT JOIN "tbl_Kunde_Lieferadresse" AS "l" ON "l"."ID" = "a"."Lieferadresse_ID"
057 LEFT JOIN "tbl_Mitarbeiter" ON "a"."Mitarbeiter_ID" = "tbl_Mitarbeiter"."ID"

```

```

058 LEFT JOIN "tbl_Land" ON "tbl_Kunde"."Land" = "tbl_Land"."ID"
059 LEFT JOIN "viw_Rechnung_Summe" ON "a"."ID" = "viw_Rechnung_Summe"."Rechnung_ID"
060 LEFT JOIN (SELECT "ID", EXTRACT(YEAR FROM "Datum") AS "Jahr",
      COALESCE("ReNummer_lfdJahr", (SELECT MAX("ReNummer_lfdJahr") + 1 FROM
      "tbl_Rechnung" WHERE EXTRACT(YEAR FROM "Datum") = EXTRACT(YEAR FROM
      "a"."Datum")),1) AS "lfdJahr" FROM "tbl_Rechnung" AS "a") AS "c"
      ON "a"."ID" = "c"."ID",
061 "tbl_Firma" AS "d"
062 WHERE "d"."ID" = True AND LEFT("viw_Rechnung_Summe"."Sort",1) = 6

```

Für die Rechnung werden hier die Inhalte vieler Tabellen und auch einer Ansicht zusammengefasst. Dabei ist die zentrale Tabelle "tbl_Kunde" die Tabelle, nach der sich alle weiteren Tabellen orientieren.

Zeile 2: Für Kleinunternehmen muss im Feld "RegisterE2" der Text "Kein Ausweis von Umsatzsteuer, da Kleinunternehmer gemäß § 19 UStG" vermerkt sein. Kleinunternehmen werden in "tbl_Firma" vermerkt. Gleichzeitig muss aber auch keine Mehrwertsteuer auf die Ware erhoben werden. Durch die Doppelabfrage nach dem Eintrag in der Firmentabelle und der Steuer in der Rechnung wird vermieden, dass bei einer Änderung in der Firmentabelle (vom Unternehmen zum Kleinunternehmen oder umgekehrt) plötzlich falsche Einträge bei vorhergehenden Rechnungen erscheinen. Siehe auch die Zeilen 7 und 45.

Zeile 5: Hier wird, wenn im Feld "Zahlungsempfänger" kein Inhalt ist, der Eintrag aus "Firma_Registriert" genommen. In der Ansicht erscheint das Feld dann trotzdem als "Zahlungsempfänger", da die über **AS** als Alias angehängt wurde.

Zeilen 9 und 10: In einer Ansicht dürfen keine Felder mit gleichem Namen vorkommen. Sonst kann der Wert nicht einwandfrei ausgelesen werden. Da die Felder "E-Mail" und "Telefon" bereits durch die "tbl_Firma" besetzt sind, wird diesen Feldern jetzt "E-Mail_Mitarbeiter" und "Telefon_Mitarbeiter" zugewiesen.

In Zeile 11 wird eine interne Kundennummer konstruiert. Die Kundennummer beginnt mit 'K-' und bekommt anschließend eine 6-stellige Ziffer angehängt. Diese Ziffer wird aus dem Primärschlüsselfeld "tbl_Kunde"."ID" ausgelesen und mit führenden Nullen bis auf 6 Stellen aufgefüllt. Das Verbinden geht mit ||, das Begrenzen auf 6 Stellen von rechts aus mit **RIGHT(... , 6)**.

Die Landesbezeichnung auf Briefen sollte nach DIN in Großbuchstaben geschrieben werden. In Zeile 13 wird deswegen das Feld "tbl_Land"."Land" mit **UPPER** in Großschreibweise umgewandelt.

In Zeile 14 wird aus den Daten für den Kunden die komplette Anschrift zusammengesetzt. Mit || werden einzelne Textelemente verbunden, mit **ASCII_CHAR(13)** wird ein Zeilenumbruch erzeugt. Schließlich soll noch vermieden werden, dass beim Eintrag 'DE' in die Adresszeile 'DEUTSCHLAND' eingetragen wird. Dies wird über eine **CASE WHEN ... THEN ... ELSE ... END** - Konstruktion bewerkstelligt.

Zeile 17: Hier taucht eine Tabellenbezeichnung auf, die in den ursprünglichen Tabellen nicht vorkommen: "a". Dies ist nur ein Alias, der der Tabelle "tbl_Rechnung" zugewiesen wird, damit der Code entsprechend verkürzt darstellbar ist.

Für den Serienbrief muss ein Datum erstellt werden, das der deutschen Schreibweise entspricht. Der Inhalt in Platzhaltern würde sonst der Schreibweise entsprechen, die bei Datenbanken üblich ist: YYYY-MM-DD. Das Datum wird mit der Funktion **EXTRACT(DAY FROM ...)** und entsprechend dazu dann eben **MONTH** und **YEAR** ausgelesen. Die Werte für den Tag und den Monat werden mit einer führenden Null versehen und auf 2 Stellen von rechts aus (**RIGHT(..., 2)**) beschnitten.

In Zeile 20 wird das Datum ermittelt, zu dem die Zahlung erfolgen soll. **FIREBIRD** ermöglicht es, zu einem Datum einfach einen **INTEGER**-Wert zu addieren. Der **INTEGER**-Wert (hier: "Zahlung-Tage") wird dann als Anzahl an Tagen verstanden und das zukünftige Datum wieder gegeben. Das Datum wird als "ZahlungBis" angezeigt.

In Zeile 22 wird eine Rechnungsnummer ähnlich wie die Kundennummer konstruiert. Hier ist als führendes Element ein 'R-', gefolgt von dem Jahr des Rechnungsdatums und einer maximal 10-

stelligen Zahl mit führenden Nummern, die in jedem Jahr neu hoch gezählt wird. Die erste Rechnung im Jahr hat so die Nummer 'R-2024-000001'. Die maximal zulässige Stellenzahl kann in dem Formular «Einstellungen» oder direkt in "tbl_Firma"."ReStellen" gesetzt werden. Die Werte für die laufende Nummer und das Jahr werden aus der Unterabfrage geholt, die mit dem Alias "c" ganz unten in dieser Ansicht "viw_Rechnung" steht (Zeile 53).

Von Zeile 29 bis Zeile 30 wird eine abweichende Lieferadresse zusammengestellt, sofern diese angegeben wurde.

Ab Zeile 31 kommen dann spezielle Konstruktionen für Startdatum, Enddatum, Projektnummer usw. dazu, die zusammen mit entsprechenden Formulierungen in die ausdrückbare Rechnungsdatei mit einfließen.

Ab Zeile 54 werden die einzelnen Tabellen aufgeführt. Sie sind nahezu komplett über einen **LEFT JOIN** mit den vorherigen Tabellen gekoppelt. Ist ein Datensatz in "tbl_Kunde" enthalten, so wird der auch angezeigt – selbst wenn zur Zeit noch keine Rechnung existiert.

Die "tbl_Firma" in Zeile 61 steht ohne Beziehung zu den anderen Tabellen mit in der Tabellenübersicht. Diese Tabelle hat genau einen Datensatz, der auch nicht über das Formular gelöscht werden kann. Deswegen benötigt die Tabelle keine weitere Verbindung.

In Zeile 62 wird für die "tbl_Firma" noch einmal klar eingeschränkt: Es wird nur der Datensatz genommen, der mit der **"ID" = TRUE** verknüpft ist. Sollte also jemand einen zusätzlichen Datensatz in die Tabelle schreiben (**"ID" = FALSE**), dann wird dieser Datensatz auf keinen Fall eingelesen. Aus "viw_Rechnung_Summe" wird schließlich nur der Datensatz mit der Gesamtsumme genutzt, der im Feld "Sort" einen Text mit einer führenden '6' enthält.

viw_Rechnung_Ausgang

ID	Rechnung_ID	MainSub	RowID	RechnungAusgan...	Row	Titel	Anzahl	Anzahl_Co...	Name_DE	Ware	Warennum...	Waren...	xWare	Preis	Rabatt	Steuersatz	Steuer_C...	StartDat...	EndDat...	AnzahlPreis	xEndDate
1	1	1	1	123 456	Main		2,0000	XPP	Stücke	Open-Soi	123 456		Leistung: Ope	12,4500	0,10	0,19	S			22,41	
2	1	1	2	920 B 64	Main		3,0000	XPP	Stücke	Handbuc	920 B 64		Leistung: Han	21,9000		0,19	S			65,7	
3	1	1	3	3	MainToSub	Stunden	1,0000	XPP	Stück	Stunden			Leistung: Stun	1125,000		0,19	S			1125	
3	1	2	3.1	3.1	Sub		10,0000	XPP	Stücke	Arbeit für			Leistung: Arbe	25,0000		0,19	S			250	
3	1	2	3.2	3.X-Test 27	SubToSubSub	Arbeit an	1,0000	XPP	Stück	Arbeit an	X-Test 27		Leistung: Arbe	875,0000		0,19	S			875	
3	1	3	3.2.1	3.X-Test 27.1	SubSub		10,0000	XPP	Stücke	Tabellene			Leistung: Tab	25,0000		0,19	S			250	
3	1	3	3.2.2	3.X-Test 27.Teil mit	SubSub		12,0000	XPP	Stücke	Formular	Teil mit Num		Leistung: Forn	25,0000		0,19	S			300	
3	1	3	3.2.3	3.X-Test 27.3	SubSubToMain		13,0000	XPP	Stücke	Makroers			Leistung: Mak	25,0000		0,19	S			325	
4	1	1	4	4	Main		4,0000	XPP	Stücke	Tafeln Sci			Leistung: Tafel	1,7500		0,07	S			7	
5	1	1	5	5	Main		1,0000	XPP	Stück	Dose Bru			Leistung: Dose	4,5000		0,07	S			4,5	

Die Darstellung dieser Ansicht zeigt einmal die korrekt formatierten Felder für "Preis", "Rabatt", "Anzahl-Preis" und "Steuersatz". In der "tbl_Ausgang" wird der Rabatt prozentual angegeben. Je nach Rabatthöhe kann natürlich das Feld "AnzahlPreis" in diesem Fall für die Anzeige gerundet erscheinen. Die weiteren Berechnungen erfolgen aber mit ungerundeten Zahlen. Gegenüber der einfachen XRechnung ist für die XRechnung-Extension die Ermittlung von Hauptrechnungszeilen und Zusatzzeilen in "MainSub" wichtig.

Felder in Bezug zur XRechnung: "RechnungAusgang_ID" [BT-126], "Anzahl" [BT-129], "Anzahl-Code_ID" [BT-130], "AnzahlPreis" [BT-131], "Preis" [BT-146], "Steuer_Code_ID" [BT-151], "Ware" [BT-153]

Datenquelle

Tabelle: [tbl_Ausgang](#), [tbl_Anzahl_Code](#)

Datenziel

Ansicht: [viw_Rechnung_Steuer](#)

Makro: [FillTableCarryOver](#), [SaveXRechnung](#), [SaveZUGFeRD](#), [ViewsErstellen](#)

```
001 SELECT "a"."ID", a"."ID" AS "AusID", "a"."Rechnung_ID", 1 AS "MainSub",
002     RANK ( ) OVER ( PARTITION BY "a"."Rechnung_ID" ORDER BY "a"."ID" ) AS "RowID",
003     COALESCE ( "a"."ZeilenNr", "a"."Warennummer", RANK ( ) OVER ( PARTITION BY
        "a"."Rechnung_ID" ORDER BY "a"."ID" ) ) AS "RechnungAusgang_ID",
```

```

004 COALESCE ( ( SELECT DISTINCT 'MainToSub' FROM "tbl_Ausgang" WHERE "Ausgang_ID" =
005 "a"."ID" ), 'Main' ) AS "Row",
006 CASE WHEN ( SELECT MAX( "ID" ) FROM "tbl_Ausgang" WHERE "Ausgang_ID" =
007 "a"."ID" ) > 0 THEN "a"."Ware" ELSE NULL END AS "Titel",
008 "a"."Anzahl", "a"."Anzahl_Code_ID",
009 CASE WHEN "a"."Anzahl" > 1 THEN COALESCE ( "tbl_Anzahl_Code"."Name_DE_Plural",
"tbl_Anzahl_Code"."Name_DE" ) ELSE "tbl_Anzahl_Code"."Name_DE" END AS
"Name_DE",
010 "a"."Ware", "a"."Warennummer", "a"."Warenzusatz",
011 COALESCE ( "tbl_Kunde"."Re_TKopf1" || ': ', '' ) || "Ware" || COALESCE(' ' ||
COALESCE ( "tbl_Kunde"."Re_TKopf2" || ': ', '' ) || "Warennummer", '' ) ||
COALESCE(' ' || COALESCE ( "tbl_Kunde"."Re_TKopf3" || ': ', '' ) ||
"Warenzusatz", '' ) AS "xWare",
012 "Preis", "Rabatt", "Steuersatz", "Steuer_Code_ID", "StartDatum", "EndDatum",
013 "a"."KontoreferenzKunde",
014 ROUND("Anzahl" * "Preis" * (1 - COALESCE("Rabatt",0)),2) AS "AnzahlPreis",
015 ROUND((ROUND("Anzahl" * "Preis" * (1 - COALESCE("Rabatt",0)),2) *
"Steuersatz"),2) AS "Steuer",
016 ROUND("Anzahl" * "Preis" * (1 - COALESCE("Rabatt",0)),2) + ROUND((ROUND("Anzahl"
* "Preis" * (1 - COALESCE("Rabatt",0)),2) * "Steuersatz"),2) AS "Brutto",
017 COALESCE ( "EndDatum", "StartDatum" ) AS "xEndDate",
018 "a"."Ausgang_ID"
019 FROM "tbl_Ausgang" AS "a", "tbl_Anzahl_Code", "tbl_Rechnung", "tbl_Kunde"
020 WHERE "a"."Anzahl_Code_ID" = "tbl_Anzahl_Code"."Code_ID"
021 AND "a"."Rechnung_ID" = "tbl_Rechnung"."ID"
022 AND "tbl_Rechnung"."Kunde_ID" = "tbl_Kunde"."ID"
023 AND "a"."Ausgang_ID" IS NULL
024 UNION
025 SELECT "a"."Ausgang_ID", "a"."ID" AS "AusID", "a"."Rechnung_ID", 2 AS "MainSub",
(SELECT COUNT("ID") FROM "tbl_Ausgang" WHERE "ID" <= "a"."Ausgang_ID" AND
"Rechnung_ID" = "a"."Rechnung_ID" AND "Ausgang_ID" IS NULL) || '.' || RANK() OVER
(PARTITION BY "a"."Ausgang_ID" ORDER BY "a"."ID") AS "RowID",
026 COALESCE("a"."ZeilenNr", "a"."Warennummer", (SELECT COUNT("ID") FROM
"tbl_Ausgang" WHERE "ID" <= "a"."Ausgang_ID" AND "Rechnung_ID" =
"a"."Rechnung_ID" AND "Ausgang_ID" IS NULL)) || CASE WHEN NOT "a"."ZeilenNr" IS
NULL THEN ' ' ELSE '.' || COALESCE("a"."Warennummer", RANK() OVER (PARTITION BY
"a"."Ausgang_ID" ORDER BY "a"."ID")) END AS "RechnungAusgang_ID",
027 CASE WHEN "a"."ID" IN (SELECT "Ausgang_ID" FROM "tbl_Ausgang" WHERE "Ausgang_ID"
IS NOT NULL AND "Rechnung_ID" = "a"."Rechnung_ID") THEN 'SubToSubSub' WHEN
RANK() OVER (PARTITION BY "a"."Ausgang_ID" ORDER BY "a"."ID") = (SELECT
COUNT("ID") FROM "tbl_Ausgang" WHERE "Ausgang_ID" = "a"."Ausgang_ID") THEN
'SubToMain' ELSE 'Sub' END AS "Row",
028 CASE WHEN (SELECT MAX("ID") FROM "tbl_Ausgang" WHERE "Ausgang_ID" = "a"."ID") >
0 THEN "Ware" ELSE NULL END AS "Titel",
029 "a"."Anzahl", "a"."Anzahl_Code_ID",
030 CASE WHEN "a"."Anzahl" > 1 THEN COALESCE("tbl_Anzahl_Code"."Name_DE_Plural",
"tbl_Anzahl_Code"."Name_DE") ELSE "tbl_Anzahl_Code"."Name_DE" END AS
"Name_DE",
031 "a"."Ware", "a"."Warennummer", "a"."Warenzusatz",
032 COALESCE ( "tbl_Kunde"."Re_TKopf1" || ': ', '' ) || "Ware" || COALESCE(' ' ||
COALESCE ( "tbl_Kunde"."Re_TKopf2" || ': ', '' ) || "Warennummer", '' ) ||
COALESCE(' ' || COALESCE ( "tbl_Kunde"."Re_TKopf3" || ': ', '' ) ||
"Warenzusatz", '' ) AS "xWare",
033 "Preis", "Rabatt", "Steuersatz", "Steuer_Code_ID", "StartDatum", "EndDatum",
034 "a"."KontoreferenzKunde",
035 ROUND("Anzahl" * "Preis" * (1 - COALESCE("Rabatt",0)),2) AS "AnzahlPreis",
036 ROUND((ROUND("Anzahl" * "Preis" * (1 - COALESCE("Rabatt",0)),2) *
"Steuersatz"),2) AS "Steuer",
037 ROUND("Anzahl" * "Preis" * (1 - COALESCE("Rabatt",0)),2) + ROUND((ROUND("Anzahl"
* "Preis" * (1 - COALESCE("Rabatt",0)),2) * "Steuersatz"),2) AS "Brutto",
038 COALESCE("EndDatum", "StartDatum") AS "xEndDate",
039 "a"."Ausgang_ID"
040 FROM "tbl_Ausgang" AS "a",
041 (SELECT "ID" AS "NID", "Rechnung_ID" AS "NRechnung_ID" FROM "tbl_Ausgang" WHERE
"Ausgang_ID" IS NULL) AS "b",
042 "tbl_Anzahl_Code", "tbl_Rechnung", "tbl_Kunde"

```

```

040 WHERE "a"."Rechnung_ID" = "b"."NRechnung_ID"
041     AND "a"."Ausgang_ID" = "b"."NID"
042     AND "a"."Anzahl_Code_ID" = "tbl_Anzahl_Code"."Code_ID"
043     AND "a"."Rechnung_ID" = "tbl_Rechnung"."ID"
044     AND "tbl_Rechnung"."Kunde_ID" = "tbl_Kunde"."ID"
045     AND NOT "a"."Ausgang_ID" IS NULL
046 UNION
047 SELECT "NAusgang_ID", a"."ID" AS "AusID", "a"."Rechnung_ID", 3 AS "MainSub",
048     (SELECT COUNT("ID") FROM "tbl_Ausgang" WHERE "ID" <= "b"."NAusgang_ID" AND
"Rechnung_ID" = "a"."Rechnung_ID")||'. '||(SELECT COUNT("ID") FROM
"tbl_Ausgang" WHERE "Ausgang_ID" = "b"."NAusgang_ID")||'. '||RANK() OVER
(PARTITION BY "a"."Ausgang_ID" ORDER BY "a"."ID") AS "RowID",
049     COALESCE("a"."ZeilenNr", (SELECT "Warennummer" FROM "tbl_Ausgang" WHERE "ID" =
"b"."NAusgang_ID"),(SELECT COUNT("ID") FROM "tbl_Ausgang" WHERE "ID" <=
"b"."NAusgang_ID" AND "Rechnung_ID" = "a"."Rechnung_ID"))||CASE WHEN NOT
"a"."ZeilenNr" IS NULL THEN ' ' ELSE '. '||COALESCE((SELECT "Warennummer" FROM
"tbl_Ausgang" WHERE "ID" = "a"."Ausgang_ID"),(SELECT COUNT("ID") FROM
"tbl_Ausgang" WHERE "Ausgang_ID" = "b"."NAusgang_ID"))||'. '||
COALESCE("a"."Warennummer",RANK() OVER (PARTITION BY "a"."Ausgang_ID" ORDER
BY "a"."ID")) END AS "RechnungAusgang_ID",
050     CASE WHEN (RANK() OVER (PARTITION BY "a"."Ausgang_ID" ORDER BY "a"."ID") =
(SELECT COUNT("ID") FROM "tbl_Ausgang" WHERE "Ausgang_ID" =
"a"."Ausgang_ID")) AND ((SELECT MAX("ID") FROM "tbl_Ausgang" WHERE
"Rechnung_ID" = "a"."Rechnung_ID" AND "Ausgang_ID" = "b"."NAusgang_ID") =
"Ausgang_ID" ) THEN 'SubSubToMain' WHEN RANK() OVER (PARTITION BY
"a"."Ausgang_ID" ORDER BY "a"."ID") = (SELECT COUNT("ID") FROM "tbl_Ausgang"
WHERE "Ausgang_ID" = "a"."Ausgang_ID") THEN 'SubSubToSub' ELSE 'SubSub' END
AS "Row",
051     CASE WHEN (SELECT MAX("ID") FROM "tbl_Ausgang" WHERE "Ausgang_ID" = "a"."ID") >
0 THEN "Ware" ELSE NULL END AS "Titel",
052     "a"."Anzahl", "a"."Anzahl_Code_ID",
053     CASE WHEN "a"."Anzahl" > 1 THEN COALESCE("tbl_Anzahl_Code"."Name_DE_Plural",
"tbl_Anzahl_Code"."Name_DE") ELSE "tbl_Anzahl_Code"."Name_DE" END AS
"Name_DE",
054     "a"."Ware", "a"."Warennummer", "a"."Warenzusatz",
055     COALESCE ( "tbl_Kunde"."Re_TKopf1" || ': ', '' ) || "Ware" || COALESCE(' ' ||
COALESCE ( "tbl_Kunde"."Re_TKopf2" || ': ', '' ) || "Warennummer", '') ||
COALESCE(' ' ||COALESCE ( "tbl_Kunde"."Re_TKopf3" || ': ', '' ) ||
"Warenzusatz", '') AS "xWare",
056     "Preis", "Rabatt", "Steuersatz", "Steuer_Code_ID", "StartDatum", "EndDatum",
"a"."KontoreferenzKunde",
057     ROUND("Anzahl" * "Preis" * (1 - COALESCE("Rabatt",0)),2) AS "AnzahlPreis",
058     ROUND((ROUND("Anzahl" * "Preis" * (1 - COALESCE("Rabatt",0)),2) *
"Steuersatz"),2) AS "Steuer",
059     ROUND("Anzahl" * "Preis" * (1 - COALESCE("Rabatt",0)),2) + ROUND((ROUND("Anzahl"
* "Preis" * (1 - COALESCE("Rabatt",0)),2) * "Steuersatz"),2) AS "Brutto",
060     ROUND("Anzahl" * "Preis" * (1 - COALESCE("Rabatt",0)),2) AS "AnzahlPreis",
061     COALESCE("EndDatum", "StartDatum") AS "xEndDate",
062     "a"."Ausgang_ID"
063 FROM "tbl_Ausgang" AS "a",
064     (SELECT "ID" AS "NID", "Rechnung_ID" AS "NRechnung_ID", "Ausgang_ID" AS
"NAusgang_ID" FROM "tbl_Ausgang" WHERE "Ausgang_ID" IS NOT NULL) AS "b",
065     "tbl_Anzahl_Code", "tbl_Rechnung", "tbl_Kunde"
066 WHERE "a"."Rechnung_ID" = "b"."NRechnung_ID"
067     AND "a"."Ausgang_ID" = "b"."NID"
068     AND "a"."Anzahl_Code_ID" = "tbl_Anzahl_Code"."Code_ID"
069     AND "a"."Rechnung_ID" = "tbl_Rechnung"."ID"
070     AND "tbl_Rechnung"."Kunde_ID" = "tbl_Kunde"."ID"
071     AND NOT "a"."Ausgang_ID" IS NULL

```

Die Ansicht ist in drei Unterabfragen aufgeteilt, die durch **UNION** (Zeile 21 und 46) zu einer Abfrage zusammen geführt werden.

Die erste Unterabfrage ermittelt nur die Datensätze der "tbl_Ausgang", die nicht von einem anderen Datensatz abhängig sind. Das sind die Datensätze, die auch in der einfachen XRech-

nung erfasst werden, da Untergliederungen dort nicht erlaubt sind. Diese Daten werden dadurch herausgefiltert, dass bei ihnen die "Ausgang_ID" leer ist (**IS NULL**, Zeile 22).

In der zweiten Unterabfrage werden die Datensätze ermittelt, die direkt auf einen der Datensätze zugreifen, die bereits in der ersten Unterabfrage ermittelt wurden. Diese Datensätze sind die direkt untergeordneten Datensätze. Diese Daten werden dadurch herausgefiltert, dass bei ihnen die "Ausgang_ID" nicht leer ist (**NOT IS NULL**, Zeile 39). Außerdem muss die "Ausgang_ID" dieser Datensätze in den Daten enthalten sein (Zeile 35), bei denen die "Ausgang_ID" leer ist (Zeile 32).

In der dritten Unterabfrage werden die Datensätze ermittelt, die auf die Datensätze der zweiten Unterabfrage zugreifen. Dabei handelt es sich dann um die zweite Unterordnung, die, wie auch weitere Unterordnungen, nach der XRechnung-Extension erlaubt sind. Diese Daten werden dadurch herausgefiltert, dass bei ihnen die "Ausgang_ID" nicht leer ist (**NOT IS NULL**, Zeile 61). Außerdem muss die "Ausgang_ID" dieser Datensätze in den Daten enthalten sein (Zeile 57), bei denen die "Ausgang_ID" nicht leer ist (Zeile 54).

In Zeile 2 wird eine laufende Nummerierung für die Rechnungsposten mit einer spezifischen Funktion von FIREBIRD erzeugt. Dabei ist die Reihenfolge der Nummerierung abhängig von der "ID" in "tbl_Ausgang": **ORDER BY "ID"**. Die Nummerierung startet für jede Rechnung neu. Dies wird durch die Formulierung **PARTITION BY "Rechnung_ID"** bewirkt. Diese Nummer ist für die XRechnung notwendig. Sie erhält hier die Bezeichnung "RowID" und in der Folgezeile, zusammen mit einer gegebenenfalls vergebenen "Warennummer", die Bezeichnung "RechnungAusgang_ID". Die "RowID" wird nur für die erste Unterabfrage fortlaufend ermittelt.

Datensätze in den beiden anderen Unterabfragen sind ja einem bestimmten Datensatz in der ersten Unterabfrage untergeordnet. Sie müssen also an erster Stelle die gleiche Bezeichnung für die "RowID" haben wie der Haupteintrag. In Zeile 20 wird der übergeordneten "RowID" durch einen Punkt getrennt die Durchnummerierung angehängt, die zu der entsprechenden Unterabfrage passt. Die führende Nummer wird dabei über eine korrelierende Unterabfrage ermittelt. Mit "a"."Ausgang_ID" wird auf die äußere Abfrage zugegriffen, die in Zeile 31 mit dem Alias "a" versehen wurde. Dadurch können alle Datensätze mit der gleichen Bedingung wie in Zeile 2 gezählt werden.

Bei der zweiten Unterabfrage muss entsprechend vorgegangen werden (Zeile 42). Um die erste Ziffer zu ermitteln wurde hier eine gesonderte Abfrage in die Tabellenbenennung mit eingebaut (Zeile 54), die die Felder aus der Hauptabfrage beinhalten, die hierfür von Bedeutung sind. Entsprechend er "RowID" in der ersten Unterabfrage wird auch hier wieder mittels korrelierenden Unterabfrage sowohl die erste als auch die zweite Stelle der "RowID" ermittelt. Das führt dann zu Bezeichnungen wie '3.2.3' im oberen Screenshot.

Um so eine Konstruktion wie die Zuordnung einer Rechnungszeile als Unterrechnungszeile zu einer vorhergehenden Rechnungszeile zu erreichen ist bei XML-Dateien wichtig, im vornherein zu wissen, ob die aktuelle Rechnungszeile eine Unterzeile hat.

```
001 <cac:InvoiceLine>
002 </cac:InvoiceLine>
```

Eine Rechnungszeile beginnt mit dem Tag **cac:InvoiceLine** und endet mit dem Tag **/cac:InvoiceLine**. Die untergeordneten Rechnungszeilen müssen innerhalb dieses Rahmens eingebaut werden:

```
001 <cac:InvoiceLine>
002   <cac:SubInvoiceLine>
003   </cac:SubInvoiceLine>
004   <cac:SubInvoiceLine>
005   </cac:SubInvoiceLine>
006 </cac:InvoiceLine>
```

Es ist also notwendig, bereits beim Erstellen der **InvoiceLine** zu wissen, ob noch eine Unterordnung erfolgt. Sonst würde die Rechnungszeile im Code enden und die Zuordnung wäre nicht gewährleistet.

```
001 <cac:InvoiceLine>
```

```

002 <cac:SubInvoiceLine>
003 </cac:SubInvoiceLine>
004 <cac:SubInvoiceLine>
005 <cac:SubInvoiceLine>
006 </cac:SubInvoiceLine>
007 <cac:SubInvoiceLine>
008 </cac:SubInvoiceLine>
009 </cac:SubInvoiceLine>
010 </cac:InvoiceLine>

```

Eine Unterordnung kann eine weitere Unterordnung enthalten. Auch für diese Unterordnung einer Unterordnung muss vorher klar sein, ob beim nächsten Datensatz eine Unterordnung erfolgen soll. Und zum Schluss muss auch noch klar sein, ob es von der zweiten Unterordnung zur ersten Unterordnung zurück geht oder sogar, wie in dem vorhergehenden Beispiel, direkt zur nächsten Hauptrechnungszeile.

Die Spalte "Row" (Zeile 4, 22 und 44) in dieser Ansicht soll ermöglichen, dass aus dem aktuellen Datensatz darauf geschlossen werden kann, ob der folgende Datensatz eben eine Unterordnung oder einen Sprung weiter nach oben enthält.

Zeile 4 - oberste Ordnung: Enthält die Tabelle "tbl_Ausgang" einen Datensatz, bei dem die "ID" des aktuellen Datensatzes als "Ausgang_ID" vorhanden ist, dann folgt dem aktuellen Datensatz eine Unterordnung. Der Begriff 'MainToSub' wird angezeigt. Ansonsten handelt es sich um eine alleinstehende Rechnungszeile. Der Begriff 'Main' wird angezeigt.

Zeile 22 - erste Unterordnung: Ist die "ID" des aktuellen Datensatzes in der Menge der Datensätze aus "tbl_Ausgang" enthalten, bei denen die "Ausgang_ID" nicht leer ist, dann folgt dem aktuellen Datensatz eine Unterordnung. Der Begriff 'SubToSubSub' wird angezeigt. Wird über die Reihenfolge innerhalb dieser Unterordnung der gleiche Wert ermittelt wie der maximale Wert der entsprechenden Unterordnung, dann folgt danach kein weiterer Wert zu den Unterordnung. Es geht zurück zur obersten Ordnung. Der Begriff 'SubToMain' wird angezeigt. Für alle anderen Einträge in der ersten Unterordnung wird 'Sub' angezeigt.

Zeile 44 - zweite Unterordnung: Wird über die Reihenfolge innerhalb dieser UnterUnterordnung der gleiche Wert ermittelt wie der maximale Wert der entsprechenden UnterUnterordnung, dann folgt danach kein weiterer Wert zu den UnterUnterordnung. Entspricht darüber hinaus noch der maximale Wert der einfachen Unterordnung dem Zeilenwert der einfachen Unterordnung, dann handelt es sich um den letzten Eintrag der UnterUnterordnung, dem direkt die oberste Ordnung folgt. Der Begriff 'SubSubToMain' wird angezeigt. Trifft nur die erste Bedingung zur UnterUnterordnung zu, so folgt dem aktuellen Datensatz der UnterUnterordnung ein weiterer Datensatz der Unterordnung. Der Begriff 'SubSubToSub' wird angezeigt. Alle sonstigen Datensätze in dieser dritten Unterabfrage starten als UnterUnterordnung und enden als UnterUnterordnung. Der Begriff 'SubSub' wird angezeigt.

In Zeile 11 sowie 29 und 51 wird der Nettobetrag für die jeweilige Rechnungszeile ermittelt. Dabei ist zu berücksichtigen, dass es eventuell Ermäßigungen in Form eines prozentualen Rabattes gibt. Da diese Ermäßigung aber auch leer sein kann muss mit **COALESCE("Rabatt", 0)** ein leerer Wert durch die Zahl '0' ersetzt werden. Der Rabatt wird von 100 % subtrahiert und mit dem Produkt aus "Anzahl" * "Preis" multipliziert. Das Ergebnis kann mehr Nachkommastellen haben, als aus dem Screenshot ersichtlich ist. Die tatsächliche weitere Berechnung erfolgt mit dem Ergebnis, das alle Nachkommastellen berücksichtigt.

Für die Berechnung der Summe einer Rechnung sind letztlich nur die Felder der Ansicht von Bedeutung, die im Feld "Ausgang_ID" keinen Eintrag haben.

viw_Rechnung_DruckSpalten

	Rechnung_ID	Reihenfolge	ID	Tabellenkopf	Feld	Format	Breite
	1	4	4		Anzahl	Number	0,95
	1	5	5		Name_DE	String	1,00
	1	8	6	Ware	Ware	String	
	1	9	9	Einzelpreis	Preis	Currency6	1,10
	1	10	10	Rabatt	Rabatt	Percent	0,75
	1	11	11	Steuer	Steuersatz	Percent	0,70
	1	12	12	Summe	AnzahlPreis	Currency	1,10
	2	4	4		Anzahl	Number	0,95
	2	5	5		Name_DE	String	1,00
	2	8	6	Ware	Ware	String	

Datenquelle

Tabelle: *tbl_Druck*

Ansicht: *viw_Rechnung_Spalten, viw_Filter*

Datenziel

Makro: *FillTableCarryOver*

```

001 SELECT "b"."Rechnung_ID", "a"."Reihenfolge", "a"."ID", "a"."Tabellenkopf",
    "a"."Feld", "a"."Format", COALESCE("a"."Breite",1) AS "Breite" FROM "tbl_Druck" AS
    "a", "viw_Rechnung_Spalten" AS "b", "viw_Filter" WHERE "a"."ID" = 1 AND
    "viw_Filter"."ShowRowNr" = TRUE
002 UNION
003 SELECT "b"."Rechnung_ID", "a"."Reihenfolge", "a"."ID", "a"."Tabellenkopf",
    "a"."Feld", "a"."Format", COALESCE("a"."Breite",1) AS "Breite" FROM "tbl_Druck" AS
    "a", "viw_Rechnung_Spalten" AS "b" WHERE "a"."ID" = 2 AND NOT "b"."colStartDatum"
    IS NULL
004 UNION
005 SELECT "b"."Rechnung_ID", "a"."Reihenfolge", "a"."ID", "a"."Tabellenkopf",
    "a"."Feld", "a"."Format", COALESCE("a"."Breite",1) AS "Breite" FROM "tbl_Druck" AS
    "a", "viw_Rechnung_Spalten" AS "b" WHERE "a"."ID" = 3 AND NOT "b"."colEndDatum" IS
    NULL
006 UNION
007 SELECT "b"."Rechnung_ID", "a"."Reihenfolge", "a"."ID", "a"."Tabellenkopf",
    "a"."Feld", "a"."Format", COALESCE("a"."Breite",1) AS "Breite" FROM "tbl_Druck" AS
    "a", "viw_Rechnung_Spalten" AS "b" WHERE "a"."ID" = 4
008 UNION
009 SELECT "b"."Rechnung_ID", "a"."Reihenfolge", "a"."ID", "a"."Tabellenkopf",
    "a"."Feld", "a"."Format", COALESCE("a"."Breite",1) AS "Breite" FROM "tbl_Druck" AS
    "a", "viw_Rechnung_Spalten" AS "b" WHERE "a"."ID" = 5 AND NOT "b"."colAnzahlCode"
    IS NULL
010 UNION
011 SELECT "b"."Rechnung_ID", "a"."Reihenfolge", "a"."ID",
    COALESCE("d"."Re_TKopf1","a"."Tabellenkopf") AS "Tabellenkopf", "a"."Feld",
    "a"."Format", NULL AS "Breite" FROM "tbl_Druck" AS "a", "viw_Rechnung_Spalten" AS
    "b", "tbl_Rechnung" AS "c", "tbl_Kunde" AS "d" WHERE "a"."ID" = 6 AND
    "b"."Rechnung_ID" = "c"."ID" AND "c"."Kunde_ID" = "d"."ID"
012 UNION
013 SELECT "b"."Rechnung_ID", "a"."Reihenfolge", "a"."ID",
    COALESCE("d"."Re_TKopf2","a"."Tabellenkopf") AS "Tabellenkopf", "a"."Feld",
    "a"."Format", COALESCE("a"."Breite",1) AS "Breite" FROM "tbl_Druck" AS "a",
    "viw_Rechnung_Spalten" AS "b", "tbl_Rechnung" AS "c", "tbl_Kunde" AS "d" WHERE
  
```

```

    "a"."ID" = 7 AND "b"."Rechnung_ID" = "c"."ID" AND "c"."Kunde_ID" = "d"."ID" AND NOT
    "b"."colWarennummer" IS NULL
014 UNION
015 SELECT "b"."Rechnung_ID", "a"."Reihenfolge", "a"."ID",
    COALESCE("d"."Re_TKopf3", "a"."Tabellenkopf") AS "Tabellenkopf", "a"."Feld",
    "a"."Format", COALESCE("a"."Breite",1) AS "Breite" FROM "tbl_Druck" AS "a",
    "viw_Rechnung_Spalten" AS "b", "tbl_Rechnung" AS "c", "tbl_Kunde" AS "d" WHERE
    "a"."ID" = 8 AND "b"."Rechnung_ID" = "c"."ID" AND "c"."Kunde_ID" = "d"."ID" AND NOT
    "b"."colWarenzusatz" IS NULL
016 UNION
017 SELECT "b"."Rechnung_ID", "a"."Reihenfolge", "a"."ID", "a"."Tabellenkopf",
    "a"."Feld", "a"."Format", COALESCE("a"."Breite",1) AS "Breite" FROM "tbl_Druck" AS
    "a", "viw_Rechnung_Spalten" AS "b" WHERE "a"."ID" = 9
018 UNION
019 SELECT "b"."Rechnung_ID", "a"."Reihenfolge", "a"."ID", "a"."Tabellenkopf",
    "a"."Feld", "a"."Format", COALESCE("a"."Breite",1) AS "Breite" FROM "tbl_Druck" AS
    "a", "viw_Rechnung_Spalten" AS "b" WHERE "a"."ID" = 10 AND NOT "b"."colRabatt" IS
    NULL
020 UNION
021 SELECT "b"."Rechnung_ID", "a"."Reihenfolge", "a"."ID", "a"."Tabellenkopf",
    "a"."Feld", "a"."Format", COALESCE("a"."Breite",1) AS "Breite" FROM "tbl_Druck" AS
    "a", "viw_Rechnung_Spalten" AS "b" WHERE "a"."ID" = 11 AND NOT "b"."colSteuersatz"
    IS NULL
022 UNION
023 SELECT "b"."Rechnung_ID", "a"."Reihenfolge", "a"."ID", "a"."Tabellenkopf",
    "a"."Feld", "a"."Format", COALESCE("a"."Breite",1) AS "Breite" FROM "tbl_Druck" AS
    "a", "viw_Rechnung_Spalten" AS "b" WHERE "a"."ID" = 12
024 UNION
025 SELECT "b"."Rechnung_ID", "a"."Reihenfolge", "a"."ID", "a"."Tabellenkopf",
    "a"."Feld", "a"."Format", COALESCE("a"."Breite",1) AS "Breite" FROM "tbl_Druck" AS
    "a", "viw_Rechnung_Spalten" AS "b", "tbl_Firma" AS "c" WHERE "a"."ID" = 13 AND
    "c"."Druck_NettoSteuerBruttoSum" = True
026 UNION
027 SELECT "b"."Rechnung_ID", "a"."Reihenfolge", "a"."ID", "a"."Tabellenkopf",
    "a"."Feld", "a"."Format", COALESCE("a"."Breite",1) AS "Breite" FROM "tbl_Druck" AS
    "a", "viw_Rechnung_Spalten" AS "b", "tbl_Firma" AS "c" WHERE "a"."ID" = 14 AND
    "c"."Druck_NettoSteuerBruttoSum" = True

```

Die Tabelle "tbl_Druck" hat 14 Datensätze. Für jeden Datensatz wird hier eine Abfrage gestellt. Anschließend werden die Informationen über **UNION** miteinander verbunden.

Für jede eventuell zur Verfügung stehende Spalte wird abgeklärt, ob sie in der Rechnung vorkommt. Dies ist aus der Ansicht "viw_Rechnung_Spalten" für viele Spalten zu erfahren. Für die Rechnungszeilennummer muss hier zusätzlich die Ansicht "viw_Filter" bemüht werden, da die Zeilennummer in den Formularen separat aus- und eingeschaltet werden kann.

In "Rechnung_ID" = '1' werden nur 7 Spalten angezeigt. Zwei Spalten werden keine Überschrift haben, bei den restlichen Spalten gibt es Spalten, die eine andere Überschrift als einen Feldnamen aus der betreffenden Ansicht "viw_Rechnung_Ausgang" haben. Die Breitenangabe für die Ware bleibt leer. Sie wird durch eine anschließende Abfrage über die Summe der Spalten für die Rechnung ermittelt ($0,95 + 1 + 1,1 + 0,75 + 0,7 + 1,1 = 5,6$; $10 - 5,6 = 4,4$).

viw_Rechnung_Spalten

	Rechnung_ID	colRabatt	colSteuersatz	colStartDatum	colEndDatum	colWarennummer	colWarenzusatz	colAnzahlCode
▶	1	0,15	0,19					
	2							
	3			01.03.24				SX

Datenquelle

Tabelle: *tbl_Ausgang*

Datenziel
Ansicht: <i>viw_Rechnung_DruckSpalten</i>
Makro: <i>FillTableCarryOver, ViewsErstellen</i>

```

001 SELECT "a"."Rechnung_ID",
002     SUM("Rabatt") AS "colRabatt",
003     CASE WHEN (SELECT MIN("Steuersatz") FROM "tbl_Ausgang" WHERE "Rechnung_ID" =
    "a"."Rechnung_ID") = (SELECT MAX("Steuersatz") FROM "tbl_Ausgang" WHERE
    "Rechnung_ID" = "a"."Rechnung_ID") THEN NULL ELSE MAX("Steuersatz") END AS
    "colSteuersatz",
004     MIN("StartDatum") AS "colStartDatum",
005     MIN("EndDatum") AS "colEndDatum",
006     CASE WHEN (SELECT "frmKunde_Warennummer" FROM "tbl_Firma") = TRUE OR (SELECT
    "tbl_Kunde"."Re_TKopf2" FROM "tbl_Kunde", "tbl_Rechnung" WHERE
    "tbl_Kunde"."ID" = "tbl_Rechnung"."Kunde_ID" AND "tbl_Rechnung"."ID" =
    "a"."Rechnung_ID") IS NOT NULL THEN MIN("Warennummer") ELSE NULL END AS
    "colWarennummer",
007     CASE WHEN (SELECT "frmKunde_Warenzusatz" FROM "tbl_Firma") = TRUE OR (SELECT
    "tbl_Kunde"."Re_TKopf3" FROM "tbl_Kunde", "tbl_Rechnung" WHERE
    "tbl_Kunde"."ID" = "tbl_Rechnung"."Kunde_ID" AND "tbl_Rechnung"."ID" =
    "a"."Rechnung_ID") IS NOT NULL THEN MIN("Warenzusatz") ELSE NULL END AS
    "colWarenzusatz",
008     CASE WHEN (SELECT COUNT("ID") FROM "tbl_Ausgang" WHERE "Rechnung_ID" =
    "a"."Rechnung_ID" AND NOT "Anzahl_Code_ID" = 'XPP') > 0 OR (SELECT
    "frmKunde_AnzahlCode" FROM "tbl_Firma") = TRUE THEN MIN("Anzahl_Code_ID")
    ELSE NULL END AS "colAnzahlCode"
009 FROM "tbl_Ausgang" AS "a"
010 GROUP BY "a"."Rechnung_ID"

```

Die Ansicht dient dazu, innerhalb der Rechnungsstellung zu entscheiden, ob bestimmte Spalten in der Rechnung notwendigerweise erscheinen müssen. So muss in "Rechnung_ID" = '1' eine Spalte für den zeilenbezogenen Rabatt erscheinen, weil dort mindestens ein Eintrag vorliegt. Die Spalte ist nicht leer (Zeile 2). Ebenso muss eine Spalte für den Steuersatz erscheinen, da der Eintrag für den minimalen Steuersatz unterschiedlich zum Eintrag für den maximalen Steuersatz ist. Es gibt also Artikel, die sich im Steuersatz unterscheiden (Zeile 3). Für das Startdatum (Zeile 4) und das Enddatum (Zeile 5) sind keine Spalten bei "Rechnung_ID" = '1' notwendig, weil dort kein Eintrag bei einem der Rechnungsposten existiert.

"Rechnung_ID" = '2' benötigt keine der abgefragten Spalten. Dort steht dann mehr Platz für die Warenbezeichnung zur Verfügung. Bei "Rechnung_ID" = '3' existiert mindestens eine Startdatum-Eingabe, so dass das Feld in der Rechnung erscheinen muss. Bei der Benennung der Anzahl hat mindestens eine Spalte einen Code, der von 'XPP' verschieden ist (Zeile 8). 'XPP' steht hier für 'Stück'. Unterscheidet sich eine dieser Maßeinheiten von 'XPP', dann müssen die Maßeinheiten auch in der Rechnung erscheinen.

viw_Rechnung_Steuer

Rechnung_ID	Sort	Steuersatz	Steuer_Code_ID	Netto	Steuer	Brutto
1	1 - Rechnungszeilen	0,07	S	69,87	4,89	74,76
1	1 - Rechnungszeilen	0,19	S	4824,81	916,71	5741,52
1	2 - Rechnung pauschal	0,19	S	-17,50	-3,33	-20,83

Datenquelle
Tabelle: <i>tbl_Rechnung_Zusatz</i>
Ansicht: <i>viw_Rechnung_Ausgang</i>

Datenziel
Ansicht: <i>viw_Rechnung_Summe</i>
Makro: <i>ViewsErstellen</i>

```

001 SELECT "Rechnung_ID", '1 - Rechnungszeilen' AS "Sort", "Steuersatz",
      "Steuer_Code_ID",
002     SUM( "AnzahlPreis" ) AS "Netto",
003     SUM( "Steuer" ) AS "Steuer",
004     SUM( "Brutto" ) AS "Brutto"
005 FROM
006     ( SELECT "AnzahlPreis", "Steuersatz", "Steuer", "Brutto", "Steuer_Code_ID",
      "Rechnung_ID" FROM "viw_Rechnung_Ausgang" WHERE "Ausgang_ID" IS NULL )
007     AS "a"
008 GROUP BY "Rechnung_ID", "Steuersatz", "Steuer_Code_ID"
009 UNION
010 SELECT "Rechnung_ID", '2 - Rechnung pauschal' AS "Sort", "Steuersatz",
      "Steuer_Code_ID",
011     ROUND(SUM( "Betrag" ),2) AS "Netto",
012     ROUND((SUM( "Betrag" ) * "Steuersatz" ),2) AS "Steuer",
013     ROUND(SUM( "Betrag" ),2) + ROUND((SUM( "Betrag" ) * "Steuersatz" ),2) AS
      "Brutto"
014 FROM "tbl_Rechnung_Zusatz"
015 GROUP BY "Rechnung_ID", "Steuersatz", "Steuer_Code_ID"

```

Diese Ansicht entspricht in weiten Teilen der Ansicht "viw_Lieferung_Steuer".

In der Unterabfrage von Zeile 6 bis Zeile 8 wird für die "tbl_Ausgang" aus "Anzahl" und "Preis" sowie gegebenenfalls dem prozentualen Abzug von "Rabatt" der Nettobetrag für die Tabellenzeilen erstellt, die keinen Eintrag in der Spalte "Ausgang_ID" enthalten. Dieser Nettobetrag ist über "AnzahlPreis" in der darüber liegenden Abfrage abrufbar.

Wäre der Rabatt ein leeres Feld (**NULL**), so würde "AnzahlPreis" auch leer sein. Deshalb wird mit **COALESCE** bei einem leeren Feld "Rabatt" stattdessen '0' genutzt.

Alle Betragsangaben werden gerundet. Dabei wird die Steuer über die Differenz von Netto zu Brutto berechnet. So sollen eventuelle Rundungsfehler bei der Rechnungssummierung ausgeschlossen werden.

In den Zeilen 1 bis 5 wird diese Unterabfrage genutzt, um "AnzahlPreis" für die gesamte Rechnung auf zu summieren. Die einfache Summe ergibt des Nettopreis, die Summe multipliziert mit dem Prozentsatz der Steuer ergibt die fällige Steuer und die Summe multipliziert mit (1 + "Steuersatz") schließlich den Bruttobetrag.

Summiert wird hier nach der Gruppierung in Zeile 10 für jede Rechnung und dort dann in Untergruppen für jeden Steuersatz, wenn eben mehr als ein Steuersatz existiert.

An diese Abfrage wird über **UNION** (Zeile 11) eine weitere Abfrage angehängt. Hier kann direkt auf die betreffende Tabelle "tbl_Rechnung_Zusatz" Bezug genommen werden, da dort die Beträge nicht von einem Rabatt oder einer Anzahl wie in "tbl_Ausgang" abhängig sind. Das Verfahren für diese Abfrage ist bei der Summierung gleich. Es wird lediglich in der Spalte "Sort" ein anderer Text ausgegeben, damit die Daten voneinander getrennt erscheinen und entsprechend zugeordnet werden können.

viw_Rechnung_Summe

	Rechnu...	Sort	Steuers...	Steuer_C...	Netto	Steuer	Brutto	BereitsGe...	NochOffen
▶	2	1 - Rechnungszeilen	0,19	S	1367,350	259,8	1627,15		
	2	2 - Rechnung pauschal	0,19	S	-17,50000	-3,33	-20,83		
	2	3 - Summe Rechnungszeilen			1367,350	259,8	1627,15		
	2	4 - Summe Rechnung pauschal			-17,50000	-3,33	-20,83		
	2	5 - Summe Steuersatz	0,19	S	1349,850	256,47	1606,32		
	2	6 - Gesamt			1349,850	256,47	1606,32	500,00	1106,32

Felder in Bezug zur XRechnung: "Netto" (Summe Rechnungszeilen) [BT-106], "Netto" (Summe Gesamt) [BT-109], "Steuer" (Summe Gesamt) [BT-110], "Brutto" (Summe Gesamt) [BT-112], "BereitsGezahlt" (Summe Gesamt) [BT-113], "NochOffen" (Summe Gesamt) [BT-115], "Netto" (Summe Steuersatz) [BT-116], "Steuer" (Summe Steuersatz) [BT-117], "Steuer_Code_ID" (Summe Steuersatz) [BT-118], "Steuersatz" (Summe Steuersatz) [BT-119]

Datenquelle

Ansicht: [viw_Rechnung_Spalten](#)

Datenziel

Ansicht: [viw_Rechnung](#)

Formular: [frm_Rechnung](#)

Makro: [FillTableCarryOver](#), [SaveXRechnung](#), [SaveZUGFeRD](#), [ViewsErstellen](#)

```

001 SELECT "viw_Rechnung_Steuer".*, NULL AS "BereitsGezahlt", NULL AS "NochOffen"
002 FROM "viw_Rechnung_Steuer"
003 UNION
004 SELECT "Rechnung_ID", '3 - Summe Rechnungszeilen' AS "Sort", NULL AS "Steuersatz",
      NULL AS "Steuer_Code_ID", SUM( "Netto" ) AS "Netto",
      SUM( "Steuer" ) AS "Steuer", SUM( "Brutto" ) AS "Brutto",
      NULL AS "BereitsGezahlt", NULL AS "NochOffen"
005 FROM "viw_Rechnung_Steuer"
006 WHERE LEFT("Sort",1) = '1'
007 GROUP BY "Rechnung_ID"
008 UNION
009 SELECT "Rechnung_ID", '4 - Summe Rechnung pauschal' AS "Sort",
      NULL AS "Steuersatz", NULL AS "Steuer_Code_ID", SUM( "Netto" ) AS "Netto",
      SUM( "Steuer" ) AS "Steuer", SUM( "Brutto" ) AS "Brutto",
      NULL AS "BereitsGezahlt", NULL AS "NochOffen"
010 FROM "viw_Rechnung_Steuer"
011 WHERE LEFT("Sort",1) = '2'
012 GROUP BY "Rechnung_ID"
013 UNION
014 SELECT "Rechnung_ID", '5 - Summe Steuersatz' AS "Sort", "Steuersatz",
      "Steuer_Code_ID", SUM( "Netto" ) AS "Netto", SUM( "Steuer" ) AS "Steuer",
      SUM( "Brutto" ) AS "Brutto", NULL AS "BereitsGezahlt", NULL AS "NochOffen"
015 FROM "viw_Rechnung_Steuer"
016 GROUP BY "Rechnung_ID", "Steuersatz", "Steuer_Code_ID"
017 UNION
018 SELECT "Rechnung_ID", '6 - Gesamt' AS "Sort", NULL AS "Steuersatz",
      NULL AS "Steuer_Code_ID", SUM( "Netto" ) AS "Netto",
      SUM( "Steuer" ) AS "Steuer", SUM( "Brutto" ) AS "Brutto",
      COALESCE((SELECT SUM("Betrag") FROM "tbl_Rechnung_Zahlung" WHERE "Rechnung_ID" =
        "a"."Rechnung_ID"), (SELECT SUM("Betrag") ) FROM "tbl_Rechnung_Zahlung" WHERE
        "Rechnung_ID" IN (SELECT "tbl_Rechnung"."ID" FROM "tbl_Rechnung",
        "tbl_rel_Rechnung" WHERE "tbl_Rechnung"."ID" =
        "tbl_rel_Rechnung"."Rechnung_ID" AND "tbl_rel_Rechnung"."Main_Rechnung_ID" =
        "a"."Rechnung_ID"))) AS "BereitsGezahlt",
      ROUND(SUM( "Brutto" ) - COALESCE((SELECT SUM("Betrag") FROM
        "tbl_Rechnung_Zahlung" WHERE "Rechnung_ID" = "a"."Rechnung_ID"), (SELECT

```

```
SUM("Betrag" ) FROM "tbl_Rechnung_Zahlung" WHERE "Rechnung_ID" IN (SELECT
"tbl_Rechnung"."ID" FROM "tbl_Rechnung", "tbl_rel_Rechnung" WHERE
"tbl_Rechnung"."ID" = "tbl_rel_Rechnung"."Rechnung_ID" AND
"tbl_rel_Rechnung"."Main_Rechnung_ID" = "a"."Rechnung_ID"),0),2)
AS "NochOffen"
```

```
019 FROM "viw_Rechnung_Steuer" AS "a"
020 GROUP BY "Rechnung_ID"
```

In Zeile 1 wird lediglich die vorher erstellte Ansicht "viw_Rechnung_Steuer" noch einmal eingelesen.

Von Zeile 3 bis 9 wird eine Abfrage an "viw_Rechnung_Steuer" gestellt, die den gesamten Inhalt nach der "Rechnung_ID" und den Einträgen für die Rechnungszeile über **LEFT("Sort", 1) = '1'** zusammenfasst – unabhängig vom Steuersatz. Das Feld für den Steuersatz bleibt leer.

Von Zeile 11 bis Zeile 17 wird dann "viw_Rechnung_Steuer" nach der "Rechnung_ID" gruppiert und lediglich für die Einträge aus "tbl_Rechnung_Zusatz" summiert. Da in "tbl_Rechnung_Zusatz" im Beispiel keine unterschiedlichen Steuersätze vorgegeben waren zeigt die Ansicht hier die gleiche Zeile für '4 – Summe Rechnung pauschal' wie für '2 – Rechnung pauschal'. Nur der Eintrag für den Steuersatz bleibt leer.

Zeile 19 bis 24 summieren "viw_Rechnung_Steuer" nach gleichen Einträgen in der "Rechnung_ID" und dem "Steuersatz". Hier ist also im Screenshot von '1 – Rechnungszeilen' für einen Steuersatz von 19 % der Betrag von '2 – Rechnung pauschal' abgezogen worden, da der eine Betrag eben durch den gegebenen Rabatt negativ war.

Zeile 26 bis 30 erstellt schließlich den Gesamtbetrag für Netto, Steuer und Brutto, wie er schließlich auch in der "viw_Rechnung" erscheint.

Abfragen

Die Datenbank enthält viele Abfrage, die mit «v_qry_...» beginnen. Diese Abfragen entsprechen den Ansichten, die mit «viw_» beginnen. Sie dienen dazu, die Ansichten zu überarbeiten und dann mittel Makro in der richtigen Reihenfolge die Ansichten neu zu erstellen.

Daneben existieren einige weitere Abfragen.

qry_Aenderung_Listbox

	Field	Value
▶	gesehen	1
	geändert	2

Datenquelle

Ansicht: *viw_Filter*

Datenziel

Formular: *frm_Lieferung_Aenderung*

```
003 SELECT 'gesehen' AS "Field", 1 AS "Value" FROM "viw_Filter"
004 UNION
005 SELECT 'geändert' AS "Field", 2 AS "Value" FROM "viw_Filter"
```

Diese Abfrage dient lediglich dazu, in einer Listbox die Begriffe in der Spalte «Field» an zu zeigen und die die Begriffe in «Value» zu speichern. Mit der Listbox soll die Bearbeitung einer eingehenden Rechnung, die weiter geleitet werden soll, nachvollzogen werden.

qry_Forms

	Anzeige	Formular	Sort
▶	 Einstellungen	frm_Einstellungen	Einstellungen
	 Konto	frm_Konto	Konto
	 Kunde	frm_Kunde	Kunde
	 Lieferant	frm_Lieferant	Lieferant

Datenquelle

Ansicht: *viw_Filter*

Datenziel

Formular: *frm_Einstellungen, frm_Rechnung, frm_Lieferung, frm_Konto*

```
001 SELECT
002 TRIM(BOTH FROM "Anzeige") AS "Anzeige",
003 TRIM(BOTH FROM "Formular") AS "Formular", "Sort"
004 FROM
005 (SELECT ' Kunde XRechnung' AS "Anzeige", 'frm_Kunde_XRechnung'
      AS "Formular". 'KundeXRechnung' AS "Sort"
      FROM "viw_Filter"
006 UNION
```

```

007 SELECT '🛒 Kunde XRechnung erweitert' AS "Anzeige", 'frm_Kunde_XRechnung-Extension'
      AS "Formular". 'KundeXRechnungerweitert' AS "Sort"
      FROM "viw_Filter"
008 UNION
009 SELECT '🛒 Kunde ZUGFeRD' AS "Anzeige", 'frm_Kunde_ZUGFeRD'
      AS "Formular". 'KundeZUGFeRD' AS "Sort"
      FROM "viw_Filter"
010 UNION
011 SELECT '🔧 Einstellungen' AS "Anzeige", 'frm_Einstellungen' AS "Formular",
      'Einstellungen' AS "Sort" FROM "viw_Filter"
012 UNION
013 SELECT '🏢 Lieferant' AS "Anzeige", 'frm_Lieferant' AS "Formular",
      'Lieferant' AS "Sort" FROM "viw_Filter"
014 UNION
015 SELECT '📄 Lieferung Änderung' AS "Anzeige", 'frm_Lieferung_Aenderung' AS
      "Formular", 'Lieferung' AS "Sort" FROM "viw_Filter"
016 UNION
017 SELECT '💰 Konto' AS "Anzeige", 'frm_Konto' AS "Formular", 'Konto' AS "Sort"
      FROM "viw_Filter")
018 ORDER BY "Sort"

```

Die Abfrage dient dazu, den Inhalt für ein Listenfeld zu erstellen, bei dem das Feld "Anzeige" gezeigt und das Feld "Formular" gespeichert wird. Mit diesem Listenfeld wird von Formular zu Formular navigiert.

Die einzelnen Unterabfragen enthalten je einen Datensatz. Da Abfragen sich auf eine Tabelle oder eine Ansicht beziehen müssen, ist hier einfach die Ansicht "viw_Filter" genommen worden, die eben aus einer Zeile besteht. Die Einträge sind reine Texteinträge, denen jeweils der Alias "Anzeige" bzw. "Formular" zugewiesen wurde. Die Icons in "Anzeige" sind UTF-8-Symbole, sind also mit der Schriftart verbunden und als Text einfach zu handhaben. Die Einträge in "Formular" entsprechen der Formularnamen innerhalb der ODB-Datei.

Firebird ist leider bei den Zeichenlängen noch nicht optimal integriert, so dass es bei so einer Konstruktion passieren kann, dass im Feld "Anzeige" und im Feld "Formular" in den Zeilen 2 und 3 Texte mit zusätzlichen Leerzeichen enthalten sind. Diese werden dort mit der Funktion **TRIM(BOTH FROM ...)** entfernt.

Für die Sortierung muss ein gesonderter Eintrag erscheinen, da sonst nach den Symbolen von "Anzeige" sortiert würde. Und das ist ja nur schwer durchschaubar. Natürlich wäre auch eine Sortierung nach dem Formularnamen möglich. Eine gesonderte Spalte macht es aber auch möglich, ggf. auch nach Wichtigkeit der Formulare zu sortieren.

qry_frmAusgang

Datenquelle

Tabelle: *tbl_Ausgang*

Datenziel

Formular: *frm_Rechnung*

```

001 SELECT "tbl_Ausgang".*,
002 COALESCE ( "tbl_Ausgang"."Rabatt", 0 ) AS "linkRabatt"
003 FROM "tbl_Ausgang"

```

Durch die XRechnung werden untergeordnete Rechnungszeilen in mehreren Bereichen gleich den übergeordneten Rechnungszeilen ausgefüllt. Um auch ein leeres Feld im Bereich "Rabatt" an das Unterformular weitergeben zu können wird eine Spalte "linkRabatt" erstellt, die für ein leeres Feld eine '0' schreibt. Feldinhalte, die **NULL** sind, können nicht zu einer Verknüpfung von Hauptformular und Unterformular genutzt werden.

qry_frmEingang

Datenquelle

Tabelle: *tbl_Eingang*

Datenziel

Formular: *frm_Lieferung*

```
001 SELECT "tbl_Eingang".*,  
002 COALESCE ( "tbl_Eingang"."Rabatt", 0 ) AS "linkRabatt"  
003 FROM "tbl_Eingang"
```

Diese Abfrage erfüllt den gleichen Zweck wie qry_frmAusgang, diesmal allerdings für das Lieferanten-Formular.

qry_frmEingangAenderung

Datenquelle

Tabelle: *tbl_Eingang_Aenderung*

Datenziel

Formular: *frm_Lieferung_Aenderung*

```
001 SELECT "tbl_Eingang_Aenderung".*,  
002 COALESCE ( "tbl_Eingang_Aenderung"."Rabatt", 0 ) AS "linkRabatt"  
003 FROM "tbl_Eingang_Aenderung"
```

Diese Abfrage erfüllt den gleichen Zweck wie qry_frmAusgang, diesmal allerdings für das Lieferanten-Formular.

qry_frmKontoRechnung

Datenquelle

Tabelle: *tbl_Rechnung_Zahlung*

Datenziel

Formular: *frm_Konto*

```
001 SELECT "ID", "Rechnung_ID", "Datum", "Betrag",  
002 ( SELECT SUM( "Betrag" ) FROM "tbl_Rechnung_Zahlung" WHERE "Rechnung_ID" =  
"a"."Rechnung_ID" AND "Datum" <= "a"."Datum" ) AS "lfdSumme",  
003 ( SELECT "Brutto" FROM "viw_Rechnung" WHERE "Rechnung_ID" = "a"."Rechnung_ID" )  
- ( SELECT SUM( "Betrag" ) FROM "tbl_Rechnung_Zahlung" WHERE "Rechnung_ID" =  
"a"."Rechnung_ID" AND "Datum" <= "a"."Datum" ) AS "NochOffen"  
004 FROM "tbl_Rechnung_Zahlung" AS "a" ORDER BY "Datum"
```

In einem Tabellenkontrollfeld wird neben den Eingabefeldern für das Datum und den Betrag die laufende Summe für die Bezahlungen der Rechnung und der noch offene Restbetrag dargestellt. Die Abfrage ist editierbar, da der Primärschlüssel der Tabelle "tbl_Rechnung_Zahlung» in der Abfrage enthalten ist.

qry_frmLieferungRechnung

Datenquelle

Tabelle: *tbl_Lieferung_Zahlung*

DatenzielFormular: *frm_Konto*

```
001 SELECT "ID", "Lieferung_ID", "Datum", "Betrag",
002   ( SELECT SUM( "Betrag" ) FROM "tbl_Lieferung_Zahlung" WHERE "Lieferung_ID" =
      "a"."Lieferung_ID" AND "Datum" <= "a"."Datum" ) AS "lfdSumme",
003   ( SELECT "SumBrutto" FROM "tbl_Lieferung" WHERE "ID" =
      "a"."Lieferung_ID" ) - ( SELECT SUM( "Betrag" ) FROM "tbl_Lieferung_Zahlung"
      WHERE "Lieferung_ID" = "a"."Lieferung_ID" AND "Datum" <= "a"."Datum" )
      AS "NochOffen"
004 FROM "tbl_Lieferung_Zahlung" AS "a" ORDER BY "Datum"
```

Diese Abfrage entspricht der Abfrage [qry_frmKontoRechnung](#), dient aber zur Eingabe von Beträgen, die für eine Lieferung gezahlt wurden.

qry_frmLieferungRechnung_Aenderung

DatenquelleTabelle: *tbl_Lieferung_Zahlung_Aenderung***Datenziel**Formular: *frm_Konto*

```
001 SELECT "ID", "Lieferung_ID", "Datum", "Betrag",
002   ( SELECT SUM( "Betrag" ) FROM "tbl_Lieferung_Zahlung_Aenderung"
      WHERE "Lieferung_ID" = "a"."Lieferung_ID" AND "Datum" <= "a"."Datum" )
      AS "lfdSumme",
003   ( SELECT "SumBrutto" FROM "tbl_Lieferung_Aenderung" WHERE "ID" =
      "a"."Lieferung_ID" ) - ( SELECT SUM( "Betrag" ) FROM
      "tbl_Lieferung_Zahlung_Aenderung" WHERE "Lieferung_ID" = "a"."Lieferung_ID"
      AND "Datum" <= "a"."Datum" ) AS "NochOffen"
004 FROM "tbl_Lieferung_Zahlung_Aenderung" AS "a" ORDER BY "Datum"
```

Diese Abfrage entspricht der Abfrage [qry_frmKontoRechnung](#), dient aber zur Eingabe von Beträgen, die für eine geänderte Lieferung eingegangen sind.

qry_frmZusatz-Einstellungen

Datenquelle

Tabelle:

Ansicht:

Datenziel

Formular:

Makro: *Einstellungenzusatz*

```
001 SELECT "tbl_Druck".*, ( SELECT "Rechnung_ID" FROM "viw_Filter" ) AS "Rechnung_ID"
002 FROM "tbl_Druck"
003 WHERE "ID" IN ( SELECT "ID" FROM "viw_Rechnung_DruckSpalten" WHERE "Rechnung_ID" =
      ( SELECT "Rechnung_ID" FROM "viw_Filter" ) )
```

Für die Einstellung der Druckparameter soll das Formular nur die Datensätze anzeigen, die in der jeweiligen Rechnung auch erscheinen. Dadurch lässt sich besser beurteilen, wie die Aufteilung der Spaltenbreiten in der Tabelle für die Rechnungspausen erfolgt.

qry_frmZusatz-Kunde

Datenquelle
Tabelle: <i>tbl_Kunde</i>
Ansicht: <i>viw_Filter</i>

Datenziel
Formular: <i>frm_Zusatz_Kunde</i>

```
001 SELECT *
002 FROM "tbl_Kunde"
003 WHERE "ID" = ( SELECT "Kunde_ID" FROM "viw_Filter" )
```

qry_frmZusatz-Lieferant

Datenquelle
Tabelle: <i>tbl_Lieferant</i>
Ansicht: <i>viw_Filter</i>

Datenziel
Formular: <i>frm_Zusatz_Lieferant</i>

```
001 SELECT *
002 FROM "tbl_Lieferant"
003 WHERE "ID" = ( SELECT "Lieferant_ID" FROM "viw_Filter" )
```

qry_frmZusatz-Lieferung

Datenquelle
Tabelle: <i>tbl_Lieferung</i>
Ansicht: <i>viw_Filter</i>

Datenziel
Makro: <i>KundeRechnungszusatz</i>

```
001 SELECT "tbl_Lieferung".*, "LieferantZeichen" AS "RNummer"
002 FROM "tbl_Lieferung"
003 WHERE "ID" = ( SELECT "Lieferung_ID" FROM "viw_Filter" )
```

Diese Abfrage entspricht der Abfrage [qry_frmZusatz-Rechnung](#), weist aber dem Feld "LieferantZeichen" den Alias «RNummer» zu, der in dem Formular «frm_Zusatz» benötigt wird.

qry_frmZusatz-Lieferung_Aenderung

Datenquelle
Tabelle: <i>tbl_Lieferung_Aenderung</i>
Ansicht: <i>viw_Filter</i>

Datenziel
Makro: <i>KundeRechnungszusatz</i>

```
001 SELECT "tbl_Lieferung_Aenderung".*, "LieferantZeichen" AS "RNummer"
```

```

002 FROM "tbl_Lieferung_Aenderung"
003 WHERE "ID" = ( SELECT "Lieferung_Aenderung_ID" FROM "viw_Filter" )

```

Diese Abfrage entspricht der Abfrage [qry_frmZusatz-Rechnung](#), weist aber dem Feld "Lieferant-Zeichen" den Alias «RNummer» zu, der in dem Formular «frm_Zusatz» benötigt wird.

qry_frmZusatz-Rechnung

Datenquelle	
Tabelle:	tbl_Rechnung
Ansicht:	viw_Rechnung , viw_Filter

Datenziel	
Formular:	frm_Zusatz
Makro:	KundeRechnungszusatz

```

001 SELECT "a".*,
002     'siehe Abschlagszahlungen' AS "LieferReferenzen",
003     ( SELECT "RechnungsnummerMitZusatz" FROM "viw_Rechnung"
        WHERE "Rechnung_ID" = "a"."ID" ) AS "RNummer"
004 FROM "tbl_Rechnung" AS "a"
005 WHERE "a"."ID" = ( SELECT "Rechnung_ID" FROM "viw_Filter" )

```

Es wird der gesamte Inhalt der "tbl_Rechnung" dargestellt. Daneben werden noch zwei Felder mit Alias versehen abrufbar: «LieferReferenzen» mit fester Texteingabe und «Rnummer» für ein Feld, in dem innerhalb des «frm_Zusatz» die Nummer der betreffenden Rechnung bzw. Lieferung dargestellt werden soll.

qry_Konto

	Titel	Firma	Datum	LieferantZeichen	Brutto	ZahlDatum	Wert	lfdSumme
▶	Lieferung	[Seller name]	11.12.17	0000123456	10.686,20 €	27.01.18	-10.686,20 €	-10.686,20 €
	Lieferung	[Seller name]	13.04.18	1234567	12.829,69 €	28.04.18	-12.829,69 €	-23.515,89 €
	Lieferung	Betriebsstätte	25.01.21	1234567890	357,00 €	01.02.21	-357,00 €	-23.872,89 €
	Rechnung	Base - XRechn	11.11.23	R-2023-00000001	4.864,16 €	20.11.23	4.864,16 €	-19.008,73 €

Datenquelle	
Tabelle:	tbl_Lieferant , tbl_Lieferung
Ansicht:	viw_Rechnung

Datenziel	
Formular:	frm_Konto

```

001 SELECT "a".*, SUM("Wert") OVER (ORDER BY "Datum") AS "lfdSumme"
002 FROM (SELECT 'Lieferung' AS "Titel", "tbl_Lieferant"."Firma",
              'L-'||"tbl_Lieferant"."ID" AS "KundeLieferant", "tbl_Lieferung"."Datum" AS
              "Rechnungsdatum", "tbl_Lieferung"."LieferantZeichen",
              "tbl_Lieferung"."Anmerkung", "tbl_Lieferung"."NochOffen" AS "Brutto",
              "tbl_Lieferung_Zahlung"."Datum", "tbl_Lieferung"."ID",
              "tbl_Lieferung_Zahlung"."Betrag" * -1 AS "Wert"
003 FROM "tbl_Lieferung", "tbl_Lieferant", "tbl_Lieferung_Zahlung"
004 WHERE "tbl_Lieferung"."Lieferant_ID" = "tbl_Lieferant"."ID" AND
        "tbl_Lieferung"."ID" = "tbl_Lieferung_Zahlung"."Lieferung_ID"
005 UNION

```

```

006 SELECT 'Lieferung' AS "Titel", "tbl_Lieferant"."Firma",
      'L-'||"tbl_Lieferant"."ID" AS "KundeLieferant",
      "tbl_Lieferung_Aenderung"."Datum" AS "Rechnungsdatum",
      "tbl_Lieferung_Aenderung"."LieferantZeichen",
      "tbl_Lieferung_Aenderung"."Anmerkung", "tbl_Lieferung_Aenderung"."NochOffen"
      AS "Brutto", "tbl_Lieferung_Zahlung_Aenderung"."Datum",
      "tbl_Lieferung_Aenderung"."ID", "tbl_Lieferung_Zahlung_Aenderung"."Betrag" AS
      "Wert"
007 FROM "tbl_Lieferung_Aenderung", "tbl_Lieferant",
      "tbl_Lieferung_Zahlung_Aenderung"
008 WHERE "tbl_Lieferung_Aenderung"."Lieferant_ID" = "tbl_Lieferant"."ID" AND
      "tbl_Lieferung_Aenderung"."ID" =
      "tbl_Lieferung_Zahlung_Aenderung"."Lieferung_ID"
009 UNION
010 SELECT 'Rechnung' AS "Titel", "viw_Rechnung"."Firma",
      'K-'||"viw_Rechnung"."Kunde_ID" AS "KundeLieferant",
      "viw_Rechnung"."Rechnungsdatum", "viw_Rechnung"."Rechnungsnummer",
      "viw_Rechnung"."Rechnung_Anmerkung", ROUND("viw_Rechnung"."Brutto",2) AS
      "Brutto", "tbl_Rechnung_Zahlung"."Datum", "viw_Rechnung"."Rechnung_ID",
      ROUND("tbl_Rechnung_Zahlung"."Betrag",2) AS "Wert"
011 FROM "viw_Rechnung", "tbl_Rechnung_Zahlung"
012 WHERE "viw_Rechnung"."Rechnung_ID" = "tbl_Rechnung_Zahlung"."Rechnung_ID"
013 ORDER BY 6, 3, 7) AS "a"
014 WHERE "LieferantZeichen" = COALESCE((SELECT "Rechnungsnummer" FROM "viw_Filter"),
      "LieferantZeichen") AND "KundeLieferant" = COALESCE ( ( SELECT
      "KundeLieferant" FROM "viw_Filter" ), "KundeLieferant" )

```

Eine Unterabfrage zu den Lieferungen und eine Unterabfrage zu den Rechnungen werden über **UNION** miteinander kombiniert. Beide Unterabfragen zeigen nur die Datensätze an, bei denen das Feld "ZahlDatum" nicht leer ist (**IS NOT NULL**). Für die Lieferung werden die angegebenen Bruttobeträge mit -1 multipliziert, da sie ja gezahlt werden müssen. Diese Kombination aus 2 Abfragen wird nach den Spalten 6, 3 und 7 nacheinander sortiert, also zuerst nach "ZahlDatum", dann bei gleichem "ZahlDatum" nach dem Datum der Lieferung bzw. Rechnung und wenn das auch noch gleich ist nach der "ID" von "tbl_Lieferung" bzw. "tbl_Rechnung".

In der äußeren Abfrage werden in Zeile 1 zuerst alle Daten aus der Unterabfrage noch einmal aufgeführt. Zusätzlich erfolgt mit der Window-Funktion (**FIREBIRD**) **SUM("Wert") OVER (ORDER BY "ZahlDatum")** eine Summierung der Beträge bis einschließlich des jeweiligen Zahldatums. So wird ein laufender Kontostand dargestellt.

qry_Land_Listbox

	Land	ID	Sort
▶	Deutschland	DE	1
	Belgien	BE	2
	Bosnien und Herzegowina	BA	2
	Dänemark	DK	2
	Estland	EE	2
	Finnland	FI	2
	Frankreich	FR	2

Datenquelle

Tabelle: [tbl_Land](#)

Datenziel

Formular: [frm_Rechnung](#), [frm_Lieferung](#), [frm_Lieferung_Aenderung](#), [frm_Lieferadresse](#)

```

001 SELECT * FROM
002     (SELECT "Land", "ID", 1 AS "Sort" FROM "tbl_Land" WHERE "ID" = 'DE'
003     UNION
004     SELECT "Land", "ID", 2 AS "Sort" FROM "tbl_Land" WHERE "ID" <> 'DE'
        AND "Kunde" = TRUE
005     UNION
006     SELECT "Land", "ID", 3 AS "Sort" FROM "tbl_Land" WHERE NOT "Kunde" = TRUE)
007 ORDER BY 3, 1 COLLATE UNICODE ASC

```

Hier wird eine Vorsortierung für das Listenfeld «Land» vorgenommen. 'Deutschland' ist als erster Datensatz positioniert und kann so leicht als Standarddatensatz in dem Kundenformular aufgerufen werden. Danach folgen alphabetisch die Länder, bei denen in der Tabelle "tbl_Land" das Feld "Kunde" angeklickt ist. Hier wird also vorausgewählt, zu welchen Ländern von der Firma aus überhaupt (öfter) Kontakt besteht. Zum Schluss dann, wieder alphabetisch, die Länder, zu denen bisher kein oder nur wenig Kontakt besteht.

Bei insgesamt 251 Datensätzen in dieser Tabelle wird so eine sinnvolle Vorsortierung für die jeweilige Firma getroffen.

✓ Hinweis

Bei den Abfragen, die Listboxen vorsortieren, können durch Änderung der Einträge im Feld "Kunde" die Datensätze ausgewählt werden, die für die eigenen Projekte sinnvoll erscheinen. Alle anderen Datensätze werden dann weiter unten präsentiert. Für die obige Abfrage ist also "tbl_Land" auf zu suchen und die entsprechenden Länder zu markieren, mit denen öfter Kontakte bestehen.

qry_Lieferung_Zahlung

SumBrutto	BereitsGezahlt	NochOffen	ZahlDatum	Lieferant
10.686,20 €	0,00 €	10.686,20 €	27.01.18	[Seller name] → E-Mail: contact@seller.de
357,00 €	0,00 €	357,00 €	01.02.21	Betriebsstätte → E-Mail: seller@email.de
233,00 €	0,00 €	233,00 €		Mustermann GmbH → E-Mail: mail@muster.de
12.829,69 €	0,00 €	12.829,69 €	28.04.18	[Seller name] → E-Mail: rechnungsausgang@test.com
2.576,41 €	0,00 €	2.576,41 €		[Seller name] → E-Mail: rechnungsausgang@test.com
804,86 €	1.030,00 €	-225,14 €		[Seller name] → E-Mail: kundencenter@sellder.de

Datenquelle

Tabelle: *tbl_Lieferant, tbl_Lieferung*

Datenziel

Formular: *frm_Konto*

```

001 SELECT "a".*,
002     ( SELECT "Firma" || ' → E-Mail: ' || "E-Mail" FROM "tbl_Lieferant" WHERE "ID" =
        "a"."Lieferant_ID" ) AS "Lieferant",
003     "b"."NochOffen" AS "aktuellNochOffen"
004 FROM "tbl_Lieferung" AS "a", "viw_Lieferung_Summe" AS "b"
005 WHERE "a"."ID" = "b"."Lieferung_ID" AND LEFT( "b"."Sort", 1 ) = 6
006     AND "ID" NOT IN ( SELECT "tbl_Lieferung_ID" FROM "tbl_Lieferung_Aenderung" )
007     AND "a"."LieferantZeichen" = COALESCE ( ( SELECT "Rechnungsnummer" FROM
        "viw_Filter" ), "a"."LieferantZeichen" )
008     AND 'L-' || "a"."Lieferant_ID" = COALESCE ( ( SELECT "KundeLieferant" FROM
        "viw_Filter" ), 'L-' || "a"."Lieferant_ID" )

```

Diese Abfrage ist grundsätzlich noch für die Eingabe von Daten brauchbar. Dies zeigt auch der Cursor an, der in der 3. Zeile im Feld "ZahlDatum" steht. Der "tbl_Lieferung" wird bei der Abfrage der Alias "a" zugewiesen (Zeile 4). In Zeile 2 erfolgt eine korrelierende Unterabfrage, die nur den Datensatz anzeigt, der genau zur "Lieferant_ID" des aktuellen Datensatzes der Abfrage passt. Dies wird durch "ID" = "a"."Lieferant_ID" bewerkstelligt. Dadurch kann die Kombination aus Firmenname und E-Mail direkt in einer Abfrage angezeigt werden.

Die Bedingungen der Abfrage in Zeile 7 und Zeile 8 ermöglichen die Filterung der Daten nach der Rechnungsnummer bzw. dem Lieferanten.

qry_Lieferung_Zahlung_Aenderung

SumBrutto	BereitsGezahlt	NochOffen	ZahlDatum	Lieferant
10.686,20 €	0,00 €	10.686,20 €	27.01.18	[Seller name] → E-Mail: contact@seller.de
357,00 €	0,00 €	357,00 €	01.02.21	Betriebsstätte → E-Mail: seller@email.de
233,00 €	0,00 €	233,00 €		Mustermann GmbH → E-Mail: mail@muster.de
12.829,69 €	0,00 €	12.829,69 €	28.04.18	[Seller name] → E-Mail: rechnungsausgang@test.com
2.576,41 €	0,00 €	2.576,41 €		[Seller name] → E-Mail: rechnungsausgang@test.com
804,86 €	1.030,00 €	-225,14 €		[Seller name] → E-Mail: kundencenter@seller.de

Datenquelle

Tabelle: [tbl_Lieferung_Aenderung](#)

Ansicht: [viw_Lieferung_Summe_Aenderung](#)

Datenziel

Formular: [frm_Konto](#)

```

001 SELECT "a".*,
002     ( SELECT "Firma" || ' → E-Mail: ' || "E-Mail" FROM "tbl_Lieferant" WHERE "ID" =
        "a"."Lieferant_ID" ) AS "Lieferant", "b"."NochOffen" AS "aktuellNochOffen"
003 FROM "tbl_Lieferung_Aenderung" AS "a", "viw_Lieferung_Summe_Aenderung" AS "b"
004 WHERE "a"."ID" = "b"."Lieferung_ID" AND LEFT( "b"."Sort", 1 ) = 6
005     AND "LieferantZeichen" = COALESCE ( ( SELECT "Rechnungsnummer" FROM
        "viw_Filter" ), "LieferantZeichen" )
006     AND 'L-' || "a"."Lieferant_ID" = COALESCE ( ( SELECT "KundeLieferant" FROM
        "viw_Filter" ), 'L-' || "a"."Lieferant_ID" )

```

Diese Abfrage entspricht der Abfrage [qry_Rechnung_Zahlung](#), nur dass sie sich auf die Inhalte bezieht, die für Lieferungen gedacht sind, die nach Bearbeitung weiter geleitet werden.

qry_Maßeinheit_Listbox

	Name_DE	Code_ID	Sort
▶	Stück	XPP	1
	m ² /Monat	MKM	2
	Transport	SX	2
	1/4 EURO-Palette	XOJ	3
	1/8 EURO-Palette	XOL	3

Datenquelle

Tabelle: [tbl_Anzahl_Code](#)

Datenziel

Formular: [frm_Rechnung](#), [frm_Lieferung](#), [frm_Lieferung_Aenderung](#)

```
001 SELECT * FROM
002   (SELECT "Name_DE", "Code_ID", 1 AS "Sort" FROM "tbl_Anzahl_Code"
      WHERE "Code_ID" = 'XPP'
003 UNION
004   SELECT "Name_DE", "Code_ID", 2 AS "Sort" FROM "tbl_Anzahl_Code"
      WHERE "Kunde" = True
005 UNION
006   SELECT "Name_DE", "Code_ID", 3 AS "Sort" FROM "tbl_Anzahl_Code"
      WHERE NOT "Kunde" = True )
007 ORDER BY 3, 1 COLLATE UNICODE ASC
```

Die Liste der Maßeinheiten, die mit Inhalten aus den XRechnungs-Vorgaben in der aktuellen Version 3 (Gültigkeit ab 18.06.2021) gefüllt wurde, beträgt 2161 Datensätze. In dieser Liste lässt sich durch ein Listenfeld nicht so schnell ein entsprechender Begriff finden. Damit als Default bei einem Listenfeld ein Wert angegeben werden kann, muss seine Position innerhalb der Auflistung klar sein. Durch diese Abfrage wird der Code für die Stückzahlen als erster Datensatz dargestellt und kann so auch Standarddatensatz eines Listenfeldes werden. Er wird dadurch am Beginn positioniert, dass die entsprechende Unterabfrage in Zeile 2 die Zahl '1' als Sortierungsspalte anzeigt.

Als weitere separate Menge werden in Zeile 4 die "Code_ID"-Werte aufgelistet, die für die Rechnungen, die erstellt werden sollen, häufiger vorkommen. Dies wird über das anklicken des Feldes "Kunde" in "tbl_Anzahl_Code" geregelt.

Erst in der dritten Unterabfrage wird schließlich der Inhalte der gesamten Tabelle "tbl_Anzahl_Code" ohne die obigen Werte angezeigt.

Siehe auch [qry_Land_Listbox](#).

qry_Rechnung_Zahlung

ZahlDatum	Rechnungsnummer	Kunde	Brutto
20.11.23	R-2023-00000001	Dreaming → E-Mail: hierhin@beispiel.de	4.864,16 €
	R-2023-00000002	Dreaming → E-Mail: hierhin@beispiel.de	1.593,74 €

Datenquelle

Tabelle: [tbl_Rechnung](#)

Ansicht: [viw_Rechnung](#)

Datenziel

Formular: [frm_Konto](#)

```
001 SELECT "a".*, "b"."Rechnungsnummer",
002   ( SELECT "Firma" || ' → E-Mail: ' || "E-Mail" FROM "tbl_Kunde" WHERE "ID" =
      "a"."Kunde_ID" ) AS "Kunde",
003   "b"."Brutto", "b"."NochOffen",
004   ( SELECT LIST( "Rechnungsnummer", ' ↴ ' || ASCII_CHAR( 13 ) ) FROM
      "viw_Rechnung", "tbl_rel_Rechnung" WHERE "viw_Rechnung"."Rechnung_ID" =
      "tbl_rel_Rechnung"."Rechnung_ID" AND "tbl_rel_Rechnung"."Main_Rechnung_ID" =
      "a"."ID" ) AS "TeilRechnungen"
```

```

005 FROM "tbl_Rechnung" AS "a", "viw_Rechnung" AS "b"
006 WHERE "a"."ID" = "b"."Rechnung_ID" AND NOT "a"."ReNumer_lfdJahr" IS NULL
007 AND "b"."Rechnungsnummer" = COALESCE ( ( SELECT "Rechnungsnummer" FROM
"viw_Filter" ), "b"."Rechnungsnummer" )
008 AND 'K-' || "a"."Kunde_ID" = COALESCE ( ( SELECT "KundeLieferant" FROM
"viw_Filter" ), 'K-' || "a"."Kunde_ID" )

```

Wie bei der Abfrage "qry_Lieferung_Zahlung" ist diese Abfrage auch weiter mit Daten befüllbar. Hier sind allerdings insgesamt 3 korrelierende Unterabfragen eingebaut, da die "tbl_Rechnung" weder den Kundennamen noch die Rechnungsnummer und auch nicht den Bruttobetrag der Rechnung enthält.

Durch die 3 Unterabfragen können aber mit dieser Abfrage auch die Felder aus den Rechnungen dargestellt werden, die zur Bestätigung des Zahlungseingangs wichtig erscheinen.

qry_Rechnungstyp_Listbox

Datenquelle
Tabelle: tbl_Rechnungstyp_Code

Datenziel
Formular: frm_Rechnung , frm_Lieferung , frm_Lieferung_Aenderung

```

001 SELECT * FROM
002 (SELECT "Name_DE", "Code_ID", 1 AS "Sort" FROM "tbl_Rechnungstyp_Code"
WHERE "Code_ID" = '380'
003 UNION
004 SELECT "Name_DE", "Code_ID", 2 AS "Sort" FROM "tbl_Rechnungstyp_Code"
WHERE "Code_ID" <> '380' AND "Kunde" = TRUE
005 UNION
006 SELECT "Name_DE", "Code_ID", 3 AS "Sort" FROM "tbl_Rechnungstyp_Code"
WHERE NOT "Kunde" = TRUE)
007 ORDER BY 3, 1 COLLATE UNICODE ASC

```

Die Code-Listboxen sind so aufgebaut, dass der Standardcode zuerst angezeigt wird, danach die für das eigene Profil notwendigen Codes und zum Schluss schließlich alle weiteren Codes.

Code '380', wie in Zeile 2 beschrieben, steht für die allgemeine Handelsrechnung. Er wird durch die Zuordnung von **1 AS "Sort"** an erster Stelle angezeigt.

Es folgen in Zeile 4 mit **2 AS "Sort"** und dem Feld **"Kunde" = True** alle Datensätze, die in "tbl_Rechnungstyp_Code" im Feld "Kunde" mit einem Haken versehen sind.

Anschließend werden alle weiteren Codes angegeben.

Siehe auch [qry_Land_Listbox](#).

qry_Steuercode_Listbox

Datenquelle
Tabelle: tbl_Steuer_Code , tbl_Firma

Datenziel
Formular: frm_Rechnung , frm_Lieferung , frm_Lieferung_Aenderung

```

001 SELECT "Name_DE", "Code_ID", '1' AS "Sort"
002 FROM "tbl_Steuer_Code"
003 WHERE "Code_ID" = COALESCE((SELECT "frmKunde_DefaultSteuerCodeID" FROM "tbl_Firma"
WHERE "ID" = TRUE), 'S')
004 UNION

```

```
005 SELECT "Name_DE", "Code_ID", '2' AS "Sort"
006 FROM "tbl_Steuer_Code"
007 WHERE "Code_ID" <> COALESCE((SELECT "frmKunde_DefaultSteuerCodeID" FROM "tbl_Firma"
    WHERE "ID" = TRUE), 'S')
008 ORDER BY 3, 1
```

Über die Bedingung in Zeile 3 wird der vorgewählte Standardcode aus der "tbl_Firma" oder, wenn dort keiner vorgewählt wurde, Code 'S' für den Normalsatz als Standardcode für Steuern angenommen. Die Listbox stellt sich also in der Vorauswahl auf diesen Code ein. Sämtliche anderen Codes werden dem in alphabetischer Reihe untergeordnet.

Formulare

Jedes Formulardokument wird zum Start auf Vollbildansicht gestellt. Außerdem werden sämtliche Symbolleisten ausgeblendet, da sie für das Formular nur störend sind.

Die Einstellung erfolgt bei der Erstellung des Formulardokument über **Extras → Anpassen → Ereignisse**.

Makros in den Ereignissen der Formulardokumente	
Dokument öffnen	Design. <i>ToolbarsHide</i>
Dokument wird geschlossen	Design. <i>ToolbarsShow</i>

frm_Einstellungen

Einstellungen

Firma* Base XRechnung Extended

StraßeNr* Open Source Allee 42 Plz* 05040 Ort* Überall

E-Mail* info@beispiel.de Website de.libreoffice.org Telefon* 0203040512345

Firma_Registriert* Base XRechnung USt-IdNr DE42424242 SteuerID 123/456/7890 Steuerbefreiung (Grund)

Bank* Spasskasse Überall IBAN* DE75512108001245126199 BIC* SPKAUEBL42

RegisterE1 RegisterE2 RegisterE3

Zahlungsnotiz Zahlbar bis zum ?Datum? Logo

ZahlungTage 14 Zahlungsempfänger

Lastschrift GläubigerNr. DE98ZZZ09999999999 Rechnung Stellenanzahl 4 Netto, Steuer und Brutto in Rechnungszeilen

Zeitraumtext Einstellungen zum Druck...

MitarbeiterInnen

ID	Name*	Geschlecht	E-Mail*	Telefon*
1	Elly Workoholic	weiblich	work@beispiel.de	013457902
2	Peter Tüftel	männlich	tueftel@beispiel.de	012457910

im Rechnungsformular zur Warenangabe anzeigen

Zeilennummer
 Warennummer
 Warezzusatz
 Rabatt
 StartDatum
 EndDatum
 Einheit
 Kontoreferenz Kunde
 Steuersatz und Steuercode
Vorgabe Steuersatz* 19 %
Vorgabe Steuercode* Normalsatz

Firmeneinstellungen speichern

1	frmFilter (Tabelle: <i>tbl_Filter</i>)
2	frmFirma (Tabelle: <i>tbl_Firma</i>)
3	frmMitarbeiter (Tabelle: <i>tbl_Mitarbeiter</i>)

Makros in Formulareigenschaften		
1	Beim Laden	Formularwechsel. <i>FormStart</i>
2	Beim Laden	Inputcontrol. <i>FormVars</i>
2	Vor der Datensatzaktion	Inputcontrol. <i>SaveRequired</i>
2	Nach dem Datensatzwechsel	Inputcontrol. <i>FormChange</i>

Makros in Feldeigenschaften		
1	Listenfeld Formularwechsel (Status geändert)	Formularwechsel. <i>Navigation</i>
2	Feld «Firma» (bei Fokusverlust) ebenso «StraßeNr», «Plz», «Ort», «Telefon», «Bank», «BIC»	Inputcontrol. <i>FieldRequired</i>
2	Feld «E-Mail» (bei Fokusverlust)	Inputcontrol. <i>MailValid_FieldRequired</i>
2	Feld «IBAN» (bei Fokusverlust)	Inputcontrol. <i>IBANValid_FieldRequired</i>
2	Feld «Steuersatz und Steuercode» (Status geändert)	Design. <i>DefaultSteuercodeErstellen</i>
2	<input type="button" value="Einstellungen zum Druck..."/> (Aktion ausführen)	Formularwechsel. <i>Einstellungenzusatz</i>

Formularfilter	
1	<input type="text" value="001"/> ("tbl_Filter"."ID" = (SELECT "ID" FROM "viw_Filter"))
2	<input type="text" value="001"/> ("tbl_Firma"."ID" = TRUE)

Im Formular 1 liegt lediglich ein Listenfeld, mit dem von einem Formular zum anderen navigiert werden kann.

Formular 2 zeigt einen einzigen Datensatz aus "tbl_Firma" an. Dies wird durch das Setzen des Filters im Formular festgeschrieben. Außerdem ist das Formular so eingestellt, dass nur eine Datenänderung, nicht aber eine Datenlöschung oder das Hinzufügen von neuen Datensätzen erlaubt ist. Der Button ist direkt über **Steuerelement-Eigenschaften → Allgemein → Aktion → Datensatz speichern** mit der Speicherfunktion verbunden.

In diesem Formular sind auch Einstellungen unter gebracht, mit denen Felder im Kundenformular sichtbar bzw. unsichtbar geschaltet werden können. Werden «Einheit» oder «Steuersatz und Steuercode» unsichtbar geschaltet, so wird stattdessen ein Standardwert in die Datensätze eingefügt. Der Standardwert für «Einheit» ist dann das universelle 'Stück', der Standardwert für «Steuersatz und Steuercode» wird ebenfalls in diesem Formular eingestellt.

Im 3. Formular können MitarbeiterInnen eingegeben werden. Hier muss mindestens eine Person eingetragen sein, da dieser Eintrag in den XRechnungen zwingend ist.

Die für die Eingabe notwendigen Felder sind durchweg mit einem «*» gekennzeichnet. Sie werden, sofern es sich um Einzelfelder wie in Formular 2 handelt, rot umrandet dargestellt, wenn sie vor dem Abspeichern noch leer sind. Die Abspeicherung ohne Eingabe in dies Felder ist nicht möglich.

zus. Einträge zu den Einstellungen des Ausdrucks

Rechnung ID

ID*	Tabellenkopf	Feld*	Format*	Breite	Reihenfolge*
1	Nr.	RechnungAusgang_ID	String	0,90	1
2	Datum	StartDatum	Date	0,90	2
3	Bis	EndDatum	Date	0,90	3
4		Anzahl	Number	0,95	4
5		Name_DE	String	1,00	5
6	Ware	Ware	String		8
7		Warennummer	String	2,00	7
8	Zusatz	Warenzusatz	String	2,15	6
9	Einzelpreis	Preis	Currency6	1,10	9
10	Rabatt	Rabatt	Percent	0,75	10
11	Steuer	Steuersatz	Percent	0,70	11
12	Summe	AnzahlPreis	Currency	1,10	12

Datensatz 1 von 12

Eingabe beenden

1 frmFilter (Tabelle: *tbl_Druck*)

Makros in Feldeigenschaften

1	<input type="button" value="Eingabe beenden"/> (Aktion ausführen)	Formularwechsel. <i>Einstellungenzu-</i> <i>satzBeenden</i>
---	---	--

Dieses Formular zeigt bei den Einstellungen alle möglichen Einträge an. Es können Einträge nur in den existierenden Datensätzen geändert, nicht aber neue Einträge hinzugefügt werden.

Wird das Formular nicht aus dem Formular «Einstellungen» sondern aus einem der Rechnungsfomulare gestartet, so zeigt das Formular nur die Felder an, die für das jeweilige Formular im Druck erscheinen werden. Dadurch lässt sich dann aus besser einschätzen, wie groß die verbleibende Breite für das Feld "Ware" sein wird.

frm_Rechnung

Das Kundenformular für die Rechnungseingabe existiert in drei Varianten. Der obige Screenshot zeigt das Formular «frm_X-Rechnung-Extension». In den Formularen «frm_XRechnung» und «frm_ZUGFeRD» fehlt vor allem das mit «2.1.2.1» gekennzeichnete Tabellenkontrollfeld zur Untergliederung der Rechnungsposten.

In der aktuellen Variante zeigen die Formulare nur die Datensätze an, bei denen noch keine Zahlung erfolgt ist. Diese Daten werden schreibgeschützt in dem jeweiligen Formular mit der Endung «..._all» angezeigt.

1	frmFilter (Tabelle: tbl_Filter)						
2	frmKunde (Tabelle: tbl_Kunde)	2.1	frmRechnung (SQL, siehe unten))	2.1.1	frmRechnungZusatz (Tabelle: tbl_Rechnung_Zusatz)		
				2.1.2	frmAusgang (Abfrage: qry_frmAusgang)	2.1.2.1	frmAusgangSub (Abfrage: qry_frmAusgang) (nur XRechnung Extension)
				2.1.3	frmSumme (Ansicht: viw_Rechnung_Summe)		
				2.1.4	frmDruck (Ansicht: viw_Rechnung)		
				2.1.5	frmAnhang (Tabelle: tbl_Rechnung_Anhang)		

Makros in Formulareigenschaften		
1	Beim Laden	Formularwechsel. FormStart
2, 2.1, 2.1.5	Beim Laden	Inputcontrol. FormVars
2, 2.1	Vor der Datensatzaktion	Inputcontrol. SaveRequired
2, 2.1, 2.1.5	Nach dem Datensatzwechsel	Inputcontrol. FormChange
2	Nach der Datensatzaktion	Design. ChangeColumnName

Makros in Formulareigenschaften		
2.1.1,	Nach der Datensatzaktion	Filter. <i>FormParallelReload</i>
2.1.2	Vor der Datensatzaktion	Inputcontrol. <i>SaveMainRow</i>
2.1.2	Nach der Datensatzaktion	Inputcontrol. <i>MainRowSaved</i>
2.1.5	Vor der Datensatzaktion	Anhang. <i>AnhangEinlesen</i>
2.1.5	Nach der Datensatzaktion	Anhang. <i>AnhangSpeichern</i>
2.1.2.1	Vor der Datensatzaktion	Inputcontrol. <i>BeforeSaveSubRow</i>
2.1.2.1	Nach der Datensatzaktion	Inputcontrol. <i>SaveSubRow</i>

Makros in Feldeigenschaften		
1	Listenfeld Formularwechsel (Status geändert)	Formularwechsel. <i>Navigation</i>
1	Listenfeld Filter «Kunde» (Status geändert)	Filter. <i>ListboxFilter</i>
2	Feld «Firma» (Bei Fokusverlust) ebenso «StraßeNr», «Plz», «Ort», «Land»	Inputcontrol. <i>FieldRequired</i>
2	Feld «E-Mail» (bei Fokusverlust)	Inputcontrol. <i>MailValid_FieldRequired</i>
2	 (weitere Kundenangaben, Aktion ausführen)	Formularwechsel. <i>Kundezusatz</i>
2.1	Feld «Datum» (bei Fokusverlust) ebenso «KundeZeichen», «MitarbeiterID»	Inputcontrol. <i>FieldRequired</i>
2.1	 (Lieferadresse bearbeiten, Aktion ausführen)	Formularwechsel. <i>KundeLieferadresse</i>
2.1	 (weitere Rechnungsangaben, Aktion ausführen)	Formularwechsel. <i>KundeRechnungszusatz</i>
2.1	 (alle Rechnungen, auch bereits gezahlte, schreibgeschützt anzeigen, Aktion ausführen)	Formularwechsel. <i>RechnungenGespart</i>
2.1	 (Rechnungskopie, Aktion ausführen)	Kopieren. <i>RechnungCopyInsert</i>
2.1.4	 Rechnung drucken (Aktion ausführen)	Druck. <i>FillTableCarryOver</i>
2.1	 (weitere Druckangaben, Aktion ausführen)	Formularwechsel. <i>Einstellungenzusatz</i>
2.1.5	 Anhang anzeigen (Aktion ausführen)	Design. <i>ShowAttachment</i>
2.1.1, 2.1.2, 2.1.2.1	Alle numerischen Felder in Tabellenkontrollfeldern (bei Fokuserhalt) (Die Inhalte werden beim Fokuserhalt markiert wie bei Textfeldern)	Design. <i>SelectContent</i>

Formularfilter	
1	001 ("tbl_Filter"."ID" = (SELECT "ID" FROM "viw_Filter"))
2	001 ("ID" = COALESCE((SELECT "Kunde_ID" FROM "viw_Filter"),"ID"))
2.1.3	001 ("viw_Rechnung_Summe"."Sort" = '6 - Gesamt')

SQL-Code Formular	
2.1	001 SELECT * 002 FROM "tbl_Rechnung" 003 WHERE "ID" IN (SELECT "Rechnung_ID" FROM "viw_Rechnung_Summe" WHERE LEFT("Sort",1) = '6' AND "NochOffen" <> 0)

SQL-Code Listenfelder	
1	Listenfeld Filter «Lieferant» <pre>001 SELECT "Firma" ' → E-Mail: ' "E-Mail" AS "Field", "ID" AS "Key" 002 FROM "tbl_Kunde" 003 ORDER BY "Field"</pre>
2	Listenfeld «Land», Datenquelle: qry_Land_Listbox
2	Listenfeld «Geschlecht», Werteliste: 'd';'m';'w', Listeneinträge: 'divers';'männlich';'weiblich'
2.1	Listenfeld «abw. Lieferadr» <pre>001 SELECT "Firma" COALESCE (' , ' "StraßeNr", '') COALESCE (' , ' "Plz", '') COALESCE ("Ort", '') AS "Field", "ID" 002 FROM "tbl_Kunde_Lieferadresse" 003 WHERE "Kunde_ID" = (SELECT "Listbox_ID" FROM "viw_Filter") 004 ORDER BY "Field" ASC</pre>
2.1	Listenfeld «Rechnungstyp*», Datenquelle: qry_Rechnungstyp_Listbox
2.1.2, 2.1.2.1	Listenfeld «Einheit», Datenquelle: qry_Maßeinheit_Listbox
2.1.2, 2.1.2.1	Listenfeld «Steuercode*», Datenquelle: qry_Steuercode_Listbox

Im Formular 1 liegt ein Listenfeld, mit dem von einem Formular zum anderen navigiert werden kann. Darüber hinaus ist ein Listenfeld enthalten, mit dem der Datenstand für bestimmte Kunden gesucht werden kann.

Formular 2 dient dazu, neue Kunden auf zu nehmen und bestehende Kunden aufzurufen und zu bearbeiten. Zum Aufrufen bestehender Kunden ist die Navigationsleiste direkt unterhalb der Eingabefelder gedacht. Hier lassen sich die Kundendatensätze nach Feldern sortieren oder auch filtern, so dass selbst bei vielen Kunden die Kundendaten schnell gefunden werden können. Über das Listenfeld «Kunde» im Filter-Bereich lässt sich auch direkt ein Kunde auswählen. Das Formular wird dann auf diesen Kunden eingestellt.

Formular 2.1 ist ein Unterformular von Formular 1. Es werden nur die Daten in Formular 2.1 angezeigt, die auch zu dem Kunden in Formular 2 passen. Auch dort sind natürlich wieder mehrere Rechnungen möglich, so dass eine Navigationsleiste dabei hilft, durch die Rechnungen durch zu schauen.

Formular 2.1.1 dient zur Eingabe von Aufschlägen und Ermäßigungen für die gesamte Rechnung, nicht nur für eine Rechnungszeile. Als Beispiele sind dort Portokosten und pauschale Rabatte eingetragen. Wird hier ein Eintrag geändert und abgespeichert, so wird das Ergebnis direkt in Formular 2.1.3 sichtbar. Die Abspeicherung in Tabellenkontrollfeldern erfolgt am einfachsten durch das Verlassen eine Datenzeile.

In Formular 2.1.2 werden die Rechnungsposten eingegeben. Bis auf das Feld für den Rabatt sind alle Felder Pflichtfelder. Die Anzahl ist mit bis zu 4 Nachkommastellen eingebbar. Die Maßeinheit für die Anzahl ist in der Standarddefinition 'Stück'. Dies ist bereits durch die Abfrage als erstes Feld ausgewählt und auch in den Eigenschaften des Listenfeldes mit der Position '0' (erster Datensatz der Abfrage – Zählung beginnt bei 0) als Vorschlagsfeld definiert. Auch Änderungen in diesem Formular werden nach Abspeicherung sofort in Formular 2.1.3 sichtbar. Die Felder in diesem Tabellenkontrollfeld können durch Einstellungen im Formular «Einstellungen» und durch Vorgaben im Formular 2 sichtbar geschaltet und teilweise mit anderen Titeln versehen werden.

Formular 2.1.3 zeigt die Summen aus der Rechnung an. Es ist gelb hinterlegt damit klar wird, dass dieses Formular nicht zur Bearbeitung dient. Natürlich ist das Formular auch anderweitig schreibgeschützt. Wird über die Navigationsleiste im unteren Bereich des Formulars die Filterung für dieses Formular ausgeschaltet, so werden alle Felder aus der Ansicht "viw_Rechnung_Summe" sichtbar, die mit der in Formular 2.1 ausgesuchten Rechnung verknüpft werden können.

nen. Es könnte also auch durch diese Ansicht gescrollt werden, um mehr Informationen zu erhalten.

Formular 2.1.4 schließlich enthält nur eine Schaltfläche und zwei Markierfelder. Das Formular greift auf die "viw_Rechnung" zu, die für den Ausdruck wichtig ist. Durch die Verbindung zu Formular 2.1 wird aus "viw_Rechnung" nur der Datensatz geladen, der mit der entsprechenden "Rechnung_ID" zusammenhängt. Auf diesen Datensatz kann dann mit Makros zugegriffen werden.

Formular 2.1.5 dient zur Auswahl von eventuell in die XRechnung einzubindende binäre Dateien. Die möglichen Dateien sind über den «mimeCode» festgelegt. Sie werden von einer Stelle im Dateisystem in das Archiv (mit anderer Namensbezeichnung) kopiert und dort auch in das Base64-Format umgewandelt. Diese Base64-Dateien werden bei der Rechnungserstellung in die XRechnung eingefügt.

Formular 2.1.2.1 enthält die Einträge, die der darüber stehenden Rechnungzeile untergeordnet werden. Dieser Formularteil setzt für die Auswertung innerhalb der XRechnung die XRechnung-Extension voraus. Die Summe der Beträge aus dieser Untergliederung entspricht dem Eintrag in dem darüberliegenden Formular 2.1.2. Existiert hier ein Eintrag, so wird der Eintrag in 2.1.2 direkt auf die Anzahl '1' gesetzt und die Summe entsprechend bestimmt.

Das Formular frm_Kunde ist über das Makro *FilterStart* als Startformular festgelegt.

frm_Zusatz

zus. Einträge auf Rechnungsebene

ID Rechnungsnummer

Zustellzeitraum

StartDatum EndDatum

1 Referenzen

ProjektNr

VertragsNr

BestellNr

AuftragsNr

AusschreibungsNr

ObjektNr

Kontoreferenz Kunde

abweichender Dateiname

Abschlagszahlungen

Rechnungen, auf die Bezug genommen werden soll

+	

Datensatz 1 von 1

1 frmZusatz (Abfrage: <i>qry_frmZusatz-Rechnung</i>)	1.1 frmRelRechnung (Tabelle: <i>tbl_rel_Rechnung</i>)
--	---

Makros in Formulareigenschaften		
1.1	Nach dem Datensatzwechsel	Filter. <i>ListenfeldfilterRechnungsreferenz</i>

Makros in Feldeigenschaften		
1.1	<input type="button" value="Eingabe beenden"/> (Aktion ausführen)	Formularwechsel. <i>Rechnungszusatz-Beenden</i>

SQL-Code Listenfelder	
1.1	<p>Listenfeld «IboStart», auch Listenfeld im Tabellenkontrollfeld - wird durch Makro jeweils aktualisiert</p> <pre> 001 SELECT "RechnungsdatumD" ' → ' "RechnungsnummerMitZusatz" ' → ' COALESCE ("Rechnung_Anmerkung" ' → ', '') REPLACE(ROUND("Brutto", 2), '.', ',') ' €' AS "Field", "Rechnung_ID", "Rechnungsdatum" AS "Sort" 002 FROM "viw_Rechnung" ORDER BY "Sort" ASC </pre>

Dieses Zusatzformular wird von Rechnungsformularen oder Lieferformularen aus gestartet. Es ergänzt den Datensatz, der in dem ursprünglichen Formular abgespeichert wurde, um zusätzliche Daten. Solch ein Formular ist nur deswegen notwendig, weil sonst der Bildschirm des Ursprungsformulars irgendwann hoffnungslos überfrachtet wäre.

frm_Zusatz_Kunde

1	frmZusatz (Abfrage: <i>qry_frmZusatz-Kunde</i>)
----------	--

Makros in Feldeigenschaften		
1	IBAN (bei Fokusverlust)	Inputcontrol. <i>IBANValid</i>
1	<input type="button" value="Eingabe beenden"/> (Aktion ausführen)	Formularwechsel. <i>KundezusatzBeenden</i>

frm_Lieferadresse

1	frmFilter (Tabelle: tbl_Filter)	1.1	frmKundeLieferadresse (Tabelle: tbl_Kunde_Lieferadresse)
----------	--	------------	---

Makros in Formulareigenschaften

1.1	Beim Laden	Inputcontrol. FormVars
1.1	Vor der Datensatzaktion	Inputcontrol. SaveRequired
1.1	Nach dem Datensatzwechsel	Inputcontrol. FormChange

Makros in Feldeigenschaften

1	Listenfeld Filter «Kunde» (Status geändert)	Filter. ListboxFilter
1.1	Feld «Firma» (Bei Fokusverlust) ebenso «Plz», «Ort», «Land»	Inputcontrol. FieldRequired
1.1	Eingabe beenden (Aktion ausführen)	Formularwechsel. LieferadresseBeenden

SQL-Code Listenfelder

1	Listenfeld Filter «Kunde» <pre>001 SELECT "Firma" ' → E-Mail: ' "E-Mail" AS "Field", "ID" AS "Key" 002 FROM "tbl_Kunde" 003 ORDER BY "Field"</pre>
1.1	Listenfeld «Land», Datenquelle: qry_Land_Listbox

Es können in den Formularen für die Rechnung und die Bearbeitung von Lieferungen abweichende Lieferadressen eingegeben werden. Die Adressen hängen dabei direkt mit dem jeweiligen Kunden zusammen. Soll so eine abweichende Lieferadresse neu eingegeben oder bearbeitet werden, so wird dieses Formular über  von dem Ursprungsformular aus gestartet, die Adresse neu hinzu gefügt oder eine bestehende Adresse bearbeitet, abgespeichert und das Formular über den entsprechenden Button geschlossen.

Hinweis

Wird das Formular nicht über [Eingabe beenden](#) geschlossen sondern stattdessen das Fenster durch den Fensterschließer X geschlossen, so wird zwar die bearbeitete Lieferadresse gespeichert, das Listenfeld zur Auswahl dieser Lieferadresse im ursprünglichen Formular aber nicht aktualisiert. In dem Fall kann die Aktualisierung über die Navigationsleiste im ursprünglichen Formular erfolgen.

frm_Lieferung

The screenshot shows the 'frm_Lieferung' application interface. Red circles highlight the following elements:

- 1**: Filter button at the top.
- 2**: 'Lieferant' field in the contact information section.
- 2.1**: 'Lieferungsaufnahme' section header.
- 2.1.1**: 'Zusatz*' table with columns for Betrag* and Steuersatz*.
- 2.1.2**: 'Rechnungsposten' table with columns for XRech-ZeilenID, Anzahl*, Einheit, Ware*, Preis*, Rabatt, Steuersatz*, Steuercode*, StartDatum, and EndDatum.
- 2.1.3**: 'Summen berechnen' table with columns for Netto, Steuer, Brutto, and bereits gezahlt.
- 2.1.4**: 'Anhang' section with a table for Anhang*.

1	frmFilter (Tabelle: <i>tbl_Filter</i>)				
2	frmLieferant (Tabelle: <i>tbl_Lieferant</i>)	2.1	frmLieferung (SQL, siehe unten)	2.1.1 frmLieferungZusatz (Tabelle: <i>tbl_Lieferung_Zusatz</i>)	
				2.1.2 frmEingang (Abfrage: <i>qry_frmEingang</i>)	2.1.2.1 frmEingangSub (Abfrage: <i>qry_frmEingang</i>)
				2.1.3 frmSumme (Ansicht: <i>viw_Lieferung_Summe</i>)	
				2.1.4 frmAnhang (Tabelle: <i>tbl_Lieferung_Anhang</i>)	

Makros in Formulareigenschaften		
1	Beim Laden	Formularwechsel. <i>FormStart</i>
2, 2.1, 2.1.4	Beim Laden	Inputcontrol. <i>FormVars</i>
2, 2.1, 2.1.4	Vor der Datensatzaktion	Inputcontrol. <i>SaveRequired</i>
2, 2.1, 2.1.4	Nach dem Datensatzwechsel	Inputcontrol. <i>FormChange</i>
2.1.1, 2.1.2	Nach der Datensatzaktion	Filter. <i>FormParallelReload</i>

Makros in Feldeigenschaften		
1	Listenfeld Formularwechsel (Status geändert)	Formularwechsel. <i>Navigation</i>
1	Listenfeld Filter «Lieferant» (Status geändert)	Filter. <i>ListboxFilter</i>
2	Feld «Firma» (Bei Fokusverlust) ebenso «StraßeNr», «Plz», «Ort», «Land», «Kontaktperson → Name», «Kontaktperson → Telefon»	Inputcontrol. <i>FieldRequired</i>
2	Feld «E-Mail» (bei Fokusverlust) ebenso «Kontaktperson → E-Mail»	Inputcontrol. <i>MailValid_FieldRequired</i>
2	Feld «IBAN» (bei Fokusverlust)	Inputcontrol. <i>IBANValid_FieldRequired</i>
2	+ (weitere Angaben zum Lieferanten, Aktion ausführen)	Formularwechsel. <i>Lieferantzusatz</i>
2	Einlesen (Aktion ausführen)	Import. <i>Import</i>
2.1	Feld «Datum» (bei Fokusverlust) ebenso «LieferantZeichen»	Inputcontrol. <i>FieldRequired</i>
2.1	 (Lieferadresse bearbeiten, Aktion ausfüh- ren)	Formularwechsel. <i>KundeLieferadresse</i>
2.1	+ (weitere Rechnungsangaben, Aktion aus- führen)	Formularwechsel. <i>KundeRechnungs- zusatz</i>
2.1.4	Anhang anzeigen (Aktion ausführen)	Design. <i>ShowAttachment</i>

Formularfilter	
1	001 ("tbl_Filter"."ID" = (SELECT "ID" FROM "viw_Filter"))
2	001 ("ID" = COALESCE((SELECT "Lieferant_ID" FROM "viw_Filter"),"ID"))
2.1.3	001 ("viw_Lieferung_Summe"."Sort" = '4 - Gesamt')

SQL-Code Formular	
2.1	001 SELECT * 002 FROM "tbl_Lieferung" 003 WHERE "ID" NOT IN (SELECT "tbl_Lieferung_ID" FROM "tbl_Lieferung_Aenderung")

SQL-Code Listenfelder	
1	Listenfeld Filter «Lieferant» 001 SELECT "Firma" ' → E-Mail: ' "E-Mail" AS "Field", "ID" AS "Key" 002 FROM "tbl_Lieferant" 003 ORDER BY "Field"
2, 2.1.5	Listenfeld «Land*», Datenquelle: qry_Land_Listbox
2	Listenfeld «Geschlecht», Werteliste: 'd';'m';'w', Listeneinträge: 'divers';'männlich';'weiblich'
2.1	Listenfeld «abw. Lieferadr» 001 SELECT "Firma" COALESCE (' , ' "StraßeNr", '') COALESCE (' , ' "Plz", '') COALESCE ("Ort", '') AS "Field", "ID" 002 FROM "tbl_Kunde_Lieferadresse" 003 WHERE "Kunde_ID" = (SELECT "Listbox_ID" FROM "viw_Filter") 004 ORDER BY "Field" ASC
2.1	Listenfeld «Rechnungstyp*», Datenquelle: qry_Rechnungstyp_Listbox
2.1.2	Listenfeld «Einheit», Datenquelle: qry_Maßeinheit_Listbox

2.1.2,
2.1.2.1

Listenfeld «Steuercode*», Datenquelle: [qry_Steuercode_Listbox](#)

Im Formular 1 liegt ein Listenfeld, mit dem von einem Formular zum anderen navigiert werden kann. Darüber hinaus ist ein Listenfeld enthalten, mit dem der Datenstand für bestimmte Lieferanten gesucht werden kann.

Formular 2 ist gegenüber dem Kundenformular umfangreicher, da hier auch die Kontaktperson sowie die Bankverbindung mit aufgenommen ist. Wieder sind viele Felder als notwendige Eingaben mit «*» gekennzeichnet. Das Feld mit der Schaltfläche **Durchsuchen...** ist nicht mit einem Formularfeld verbunden. Hier wird das Dateisystem nach der zu importierenden XRechnung durchsucht. Über **Einlesen** wird das entsprechende Makro gestartet und die eingelesenen Daten anschließend angezeigt. Da nicht alle Felder in diesem Formular Pflichtfelder sind kann es sein, dass auch nicht alle Felder in der XRechnung auftauchen. Entsprechend können hier die Telefonnummer oder der Zahlungsempfänger ergänzt werden.

Formular 2.1 zeigt nur die Lieferungen an, die zum aktuell ausgewählten Lieferanten passen. Hier wird dann neben den Informationen im Formular für die Kunden auch die Zahlungsfrist (sofern angegeben und auslesbar) angegeben. Zusätzlich wird angezeigt, welche Summen in der Rechnung angegeben wurden.

Formular 2.1.1 zeigt wieder die Beträge an, die für die gesamte Lieferung gelten. In Formular 2.1.2 sind dann die einzelnen Lieferungsposten aufgeführt. Hier wird auch die ZeilenID aus der XRechnung mit erwähnt, die nur aus der XRechnung auslesbar ist. Der Rabatt pro Rechnungsposten wird als Betrag ausgegeben, da dieser Eintrag ein Pflichteintrag bei der Gewährung von Rabatt ist, während die prozentuale Angabe nur freiwillig gemacht werden kann.

Formular 2.1.3 dient schließlich dazu, die rechnerische Summierung mit der für die Lieferung angegebenen Summe in Formular 2 zu vergleichen. Hier können vor allem dadurch Differenzen auftauchen, dass die Angabe der Steuersätze für die einzelnen Lieferposten in XRechnungen nicht Pflicht sind. Das kann gegebenenfalls durch die nachträgliche Eingabe der Steuersätze korrigiert werden. Auch werden in die Tabellen dieser Datenbank nur Währungsbeträge aufgenommen, die maximal 4 Stellen hinter dem Komma haben. Das führt im obigen Screenshot zu der Differenz zwischen Berechnung und tatsächlichen Rechnungsbetrag.

Formular 2.1.4 zeigt schließlich in die XRechnung eingebettete Dateien an.

frm_Zusatz_Lieferant

zus. Einträge zum Lieferanten

ID 8

UST-IdNr DE42424242

ZahlungTage

RegisterID

Gläubigernummer DE98ZZZ0999999999

UST-Ausnahme

SteuerNr

Eingabe beenden

1 frmZusatz (Abfrage: [qry_frmZusatz-Lieferant](#))

Makros in Feldeigenschaften

1	Eingabe beenden (Aktion ausführen)	Formularwechsel. LieferantzusatzBeenden
---	---	---

frm_Lieferung_Aenderung

Wird die Datenbank dazu benötigt, eingehende Rechnungen auf ihre Richtigkeit zu überprüfen und anschließend an den ursprünglichen Adressaten weiter zu leiten, so sollte das dieses Formular erfüllen. Die über das Lieferantenformular eingelesenen Daten stehen hier zur Nachbearbeitung zur Verfügung. Dabei bleiben die ursprünglichen Daten unberührt. Die Daten einer Lieferung werden über einen Button in 2.2 in das Formular übertragen und können dann ggf. korrigiert weiter gegeben werden.

1	frmFilter (Tabelle: <i>tbl_Filter</i>)						
2	frmLieferant (Tabelle: <i>tbl_Lieferant</i>)	2.1	frmLieferung (Tabelle: <i>tbl_Lieferung_Aenderung</i>)	2.1.1	frmLieferungZusatz (Tabelle: <i>tbl_Lieferung_Zusatz_Aenderung</i>)		
				2.1.2	frmEingang (Abfrage: <i>qry_frmEingangAenderung</i>)	2.1.2.1	frmEingangSub (Abfrage: <i>qry_frmEingangAenderung</i>)
				2.1.3	frmSumme (Ansicht: <i>viw_Lieferung_Summe_Aenderung</i>)		
				2.1.4	frmAnhang (Tabelle: <i>tbl_Lieferung_Anhang_Aenderung</i>)		
				2.1.5	frmKunde (Tabelle: <i>tbl_Kunde</i>)		
				2.1.6	frmDruck (Ansicht: <i>viw_Lieferung_Aenderung</i>)		
		2.2	frmFilterLieferung (Tabelle: <i>tbl_Filter</i>)				

Makros in Formulareigenschaften		
1	Beim Laden	Formularwechsel. <i>FormStart</i>
2, 2.1, 2.1.4	Beim Laden	Inputcontrol. <i>FormVars</i>
2, 2.1	Vor der Datensatzaktion	Inputcontrol. <i>SaveRequired</i>
2, 2.1, 2.1.4	Nach dem Datensatzwechsel	Inputcontrol. <i>FormChange</i>
2.1.1, 2.1.2	Nach der Datensatzaktion	Filter. <i>FormParallelReload</i>
2.1.4	Vor der Datensatzaktion	Anhang. <i>AnhangEinlesen</i>
2.1.4	Nach der Datensatzaktion	Anhang. <i>AnhangSpeichern</i>
2.2	Beim Laden	Filter. <i>Listenfeldfilter</i>
2.2	Beim erneuten Laden	Filter. <i>Listenfeldfilter</i>

Makros in Feldeigenschaften		
1	Listenfeld Formularwechsel (Status geändert)	Formularwechsel. <i>Navigation</i>
1	Listenfeld Filter «Lieferant» (Status geändert)	Filter. <i>ListboxFilter</i>
2	Feld «Firma» (Bei Fokusverlust) ebenso «StraßeNr», «Plz», «Ort», «Land», «Kontaktperson → Name», «Kontaktperson → Telefon»	Inputcontrol. <i>FieldRequired</i>
2	Feld «E-Mail» (bei Fokusverlust) ebenso «Kontaktperson → E-Mail»	Inputcontrol. <i>MailValid_FieldRequired</i>
2	Feld «IBAN» (bei Fokusverlust)	Inputcontrol. <i>IBANValid_FieldRequired</i>
2	 (weitere Angaben zum Lieferanten, Aktion ausführen)	Formularwechsel. <i>Lieferantzusatz</i>
2.1	Feld «Datum» (bei Fokusverlust) ebenso «LieferantZeichen»	Inputcontrol. <i>FieldRequired</i>
2.1	 (Lieferadresse bearbeiten, Aktion ausführen)	Formularwechsel. <i>KundeLieferadresse</i>
2.1	 (weitere Rechnungsangaben, Aktion ausführen)	Formularwechsel. <i>KundeRechnungszusatz</i>
2.1.4	 Anhang anzeigen (Aktion ausführen)	Design. <i>ShowAttachment</i>
2.1.6	 XRechnung erstellen (Aktion ausführen, Zusatzinformation: '')	Druck. <i>FillTableCarryOver</i>
2.1.6	 Rech. ZUGfERD erstellen (Aktion ausführen, Zusatzinformation: 'ZUGfERD')	Druck. <i>FillTableCarryOver</i>
2.2	 Lieferung übertragen (Aktion ausführen)	EingangBearbeiten. <i>Import_to_Aenderung</i>
2.1.1, 2.1.2, 2.1.2.1	Alle numerischen Felder in Tabellenkontrollfeldern (bei Fokuserhalt) (Die Inhalte werden beim Fokuserhalt markiert wie bei Textfeldern)	Design. <i>SelectContent</i>

Formularfilter	
1	001 ("tbl_Filter"."ID" = (SELECT "ID" FROM "viw_Filter"))
2	001 ("ID" = COALESCE((SELECT "Lieferant_ID" FROM "viw_Filter"),"ID"))
2.1.3	001 ("viw_Lieferung_Summe"."Sort" = '4 - Gesamt')

SQL-Code Listenfelder	
1	Listenfeld Filter «Lieferant» 001 SELECT "Firma" ' → E-Mail: ' "E-Mail" AS "Field", "ID" AS "Key" 002 FROM "tbl_Lieferant" 003 ORDER BY "Field"
2, 2.1.5	Listenfeld «Land», Datenquelle: qry_Land_Listbox
2	Listenfeld «Geschl.», Werteliste: 'd';'m';'w', Listeneinträge: 'divers';'männlich';'weiblich'
2.1	Listenfeld «Mitarbeiter» 001 SELECT "Name", "ID" 002 FROM "tbl_Mitarbeiter" 003 ORDER BY "Name" ASC
2.1	Listenfeld «abw. Lieferadr» 001 SELECT "Firma" COALESCE (' , ' "StraßeNr", ' ') COALESCE (' , ' "Plz", ' ') COALESCE ("Ort", ' ') AS "Field", "ID" 002 FROM "tbl_Kunde_Lieferadresse" 003 WHERE "Kunde_ID" = (SELECT "Listbox_ID" FROM "viw_Filter") 004 ORDER BY "Field" ASC
2.1, 2.1.2, 2.1.2.1	Listenfeld «Änderung», Datenquelle: qry_Aenderung_Listbox
2.1	Listenfeld «Rechnungstyp*», Datenquelle: qry_Rechnungstyp_Listbox
2.1.2	Listenfeld «Einheit», Datenquelle: qry_Maßeinheit_Listbox
2.1.2, 2.1.2.1	Listenfeld «Steuercode*», Datenquelle: qry_Steuercode_Listbox

Dieses Formular ist eine Kombination aus dem Lieferformular und dem Formular, mit dem sonst die Rechnungen erstellt werden können. So kann der Kunde in diesem Formular (2.1.5) auch über eins der Rechnungsformulare ausführlicher bearbeitet werden. Dieses Formular bietet einfach nicht für alle Felder den notwendigen Platz.

frm_Konto

Konto Filter
Rechnungsnummer Kunde/Lieferant

Offene Zahlungen von Lieferungen 1

Lieferant	Datum	LieferantZeichen	UnserZeichen	Anmerkung	Brutto	NochOffen
[Seller name] → E-Mail: seller@em	11.12.17	0000123456	11002002-98765		10.686,20 €	10.686,20 €
Dolmetscher → E-Mail: info@dolm	03.07.24	RF1993668	04011000-900X9		1.666,00 €	1.666,00 €
Base XRechnung Extended → E-M.	25.03.24	R-2023-0001	-		737,80 €	737,80 €

Datensatz 1 von 3

Datum	Betrag	lfd. Summe	lfd. off. Betrag

Datensatz 1 von 1

Offene Zahlungen von geänderten Lieferungen 3

Lieferant	Datum	LieferantZeichen	UnserZeichen	Anmerkung	Brutto	NochOffen
[Seller name] → E-Mail: xxx@schu	24.04.15	PP31582112	99000000-18188		10.555,30 €	10.555,30 €

Datensatz 1 von 1

Datum	Betrag	lfd. Summe	lfd. off. Betrag

Datensatz 1 von 1

Offene Eingänge von Rechnungen 4

Kunde	Datum	RechnungsNr.	Teilrechnungen	Anmerkung	Brutto	NochOffen
Dreaming & Co. → E-Mail: hierhin@	08.12.23	R-2023-0002		Arbeitsleistung	1.606,32 €	1.106,32 €
Dreaming & Co. → E-Mail: hierhin@	25.03.24	R-2024-0001			737,80 €	737,80 €

Datensatz 1 von 2

Datum	Betrag	lfd. Summe	lfd. off. Betrag
10.12.23	500,00 €	500,00 €	1.106,32 €

Datensatz 1 von 1

Kontostand 5

Zahldatum	Titel	Firma	Datum	Lief-Zeichen/R-Nummer	Anmerkung	Brutto	Wert	lfdSumme
01.09.19	Lieferung	[Seller trading name]	20.08.19	1234567		13,03 €	-13,03 €	-13,03 €
30.04.21	Lieferung	Testverkäufer	23.04.21	112233		1.804,00 €	1.804,00 €	1.790,97 €
15.11.23	Rechnung	Base XRechnung Extended	11.11.23	R-2023-0001		5.795,46 €	2.000,00 €	3.790,97 €
20.11.23	Rechnung	Base XRechnung Extended	11.11.23	R-2023-0001		5.795,46 €	3.000,00 €	6.790,97 €
01.12.23	Rechnung	Base XRechnung Extended	11.11.23	R-2023-0001		5.795,46 €	795,46 €	7.586,43 €

Datensatz 1 von 6

1	frmFilter (Tabelle: <i>tbl_Filter</i>)		
2	frmLieferungen (Abfrage: <i>qry_Lieferung_Zahlung</i>)	2.1	frmLieferungZahlung (Abfrage: <i>qry_frmLieferungRechnung</i>)
3	frmLieferungenAenderung (Abfrage: <i>qry_Lieferung_Zahlung_Aenderung</i>)	3.1	frmLieferungZahlungAenderung (Abfrage: <i>qry_frmLieferungRechnung_Aenderung</i>)
4	frmRechnungen (Abfrage: <i>qry_Rechnung_Zahlung</i>)	4.1	frmRechnungZahlung (Abfrage: <i>qry_frmKontoRechnung</i>)
5	frmSumme (Abfrage: <i>qry_Konto</i>)		

Makros in Formulareigenschaften		
1	Beim Laden	Formularwechsel. <i>FormStart</i>
2, 3	Nach der Datensatzaktion	Filter. <i>FormParallelReload</i>
2.1, 3.1, 4.1	Nach der Datensatzaktion	Filter. <i>FormsReload</i>

Makros in Feldeigenschaften		
1	Listenfeld Formularwechsel (Status geändert)	Formularwechsel. <i>Navigation</i>
1	Listenfeld Rechnungsnummer (Status geändert)	Filter. <i>Rechnungsfilter</i>
1	Listenfeld Kunde/Lieferant (Status geändert)	Filter. <i>Rechnungsfilter</i>

Formularfilter	
1	001 ("tbl_Filter"."ID" = (SELECT "ID" FROM "viw_Filter"))
2	001 ("a"."ZahlDatum" IS NULL)
3	001 ("a"."ZahlDatum" IS NULL)

SQL-Code Listenfelder	
1	Listenfeld Filter «Rechnungsnummer» 001 SELECT 'R-' EXTRACT(YEAR FROM "Datum") '-' RIGHT('0000000000' "ReNummer_lfdJahr", COALESCE ((SELECT "ReStellen" FROM "tbl_Firma" WHERE "ID" = TRUE), 6)) AS "Rechnungsnummer" FROM "tbl_Rechnung" 002 UNION 003 SELECT "LieferantZeichen" FROM "tbl_Lieferung"
1	Listenfeld Filter «Kunde/Lieferant» 001 SELECT 'Kunde: ' "Firma", 'K-' "ID" AS "Nummer" FROM "tbl_Kunde" 002 UNION 003 SELECT 'Lieferant: ' "Firma", 'L-' "ID" AS "Nummer" FROM "tbl_Lieferant"

Dieses Formular dient lediglich dazu, einen Überblick über den Kontostand zu behalten. Für die Erzeugung von XRechnungen oder das Einlesen von XRechnungen ist es ohne Belang.

In den Formularen 2, 3 und 4 erscheinen die Daten für Lieferungen bzw. Rechnungen, die noch nicht komplett durch einen Zahlungsbetrag ausgeglichen sind. In den Formularen 2.1, 3.1 und 4.1 kann der jeweiligen Zeile des übergeordneten Formulars eine Zahlung zugewiesen werden. Hier ist also auch die Zuweisung von Teilzahlungen möglich.

Diese Writer-Datei ist so eingestellt, dass sie über **Ansicht → Tabellenbegrenzungen** die Tabellen anzeigt. Auch die Formatierungszeichen sind sichtbar. So ist auch deutlich, dass die eigentlichen Rechnungszeilen in Tabellenform gelistet werden.

The image shows a Writer document with a table-based invoice layout. The following sections are highlighted with red circles:

- Adresse:** A table containing customer address information, with the header `<ANSCHRIFT_KUNDE>`.
- Logo:** A table in the top right corner.
- Absender:** A table containing sender information, with headers like `<FIRMA>`, `<STRASSENR>`, `<PLZ>`, `<ORT>`, `<TELEFON>`, and `<E-MAIL>`.
- Rechnungsinhalt:** A large table containing the main invoice details, including `<RECHNUNGSTYP>`, `Rechnung-Nummer:`, `Datum:`, `<RECHNUNGSDATUMD>`, `Kundennummer:`, and various date and reference fields.
- Rechnungsinfo:** A table containing payment information, with headers like `UMSATZLIEFERNR`, `ZAHLUNGSART`, `<USt-IDNR>`, and `<ZAHLUNGSNOTIZ>`.
- FirmaFußzeile1:** A table at the bottom containing company footer information, with headers like `FIRMA (REG.-NAME)`, `TELEFON/FAX/WEB`, `BANK/BLZ/BIC`, and `<FIRMA REGISTRIERT>`.

Die Vorlagendatei besteht aus drei besonders benannten Tabellen, so dass die Tabellen über Makros auch entsprechend angesteuert werden können. Die Tabelle «Rechnungsinfo» wird zur Zeit nicht mit zusätzlichen Informationen gefüllt. Sie dient nur dazu, Steuernummer und Zahlungsweise sauber zu formatieren. Dabei kann als Steuernummer alternativ zur Umsatzsteuer-

nummer auch die einfache Steuernummer (bei Steuerbefreiung für Kleinunternehmen) angegeben werden.

Darunter befindet sich die Tabelle «Rechnungsinhalt», die lediglich aus 2 Zeilen besteht. Die erste Zeile enthält die Vorlage für die Tabellenüberschriften, die zweite Zeile die für den Inhalt. Die Spalten für die Tabelle werden über die "tbl_Druck" sowie ein entsprechendes Makro erstellt und für die entsprechenden Formate vorformatiert.

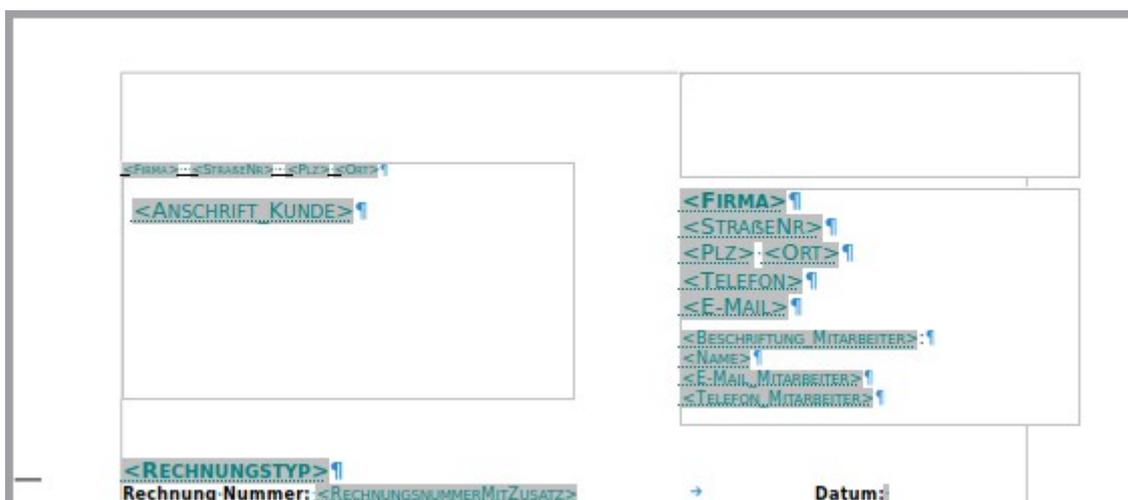
Am Seitenschluss befindet sich die Tabelle «FirmaFußzeile1». Dies ist die Fußzeilentabelle für die erste Seite. Auf der Folgeseite muss stehen dann für alle weiteren Seiten die Tabelle «FirmaFußzeile2». Die Fußzeilentabellen werden gegebenenfalls um eine weitere Spalte per Makro ergänzt. Dies ist abhängig davon, ob entsprechende Registereinträge in der Tabelle "tbl_Firma" vorgenommen wurden.

Rechts oben befindet sich ein einfacher Textrahmen, der mit dem Namen «Logo» versehen ist. Ein im Formular «frm_Einstellungen» eingelesenes Logo wird hier im richtigen Seitenverhältnis von der Größe her an den Rahmen angepasst wieder gegeben.

In Cyanblau mit grauem Hintergrund sind auf der Vorlage Platzhalter ersichtlich. Diese Platzhalter sind vom Namen her gleich den Feldern in "viw_Rechnung". Wird über das Makro ein Platzhalter gefunden, der eine Namensgleichheit mit einem eingelesenen Feld aufweist, so wird der entsprechende Wert des Feldes aus "viw_Rechnung" statt des Platzhalter dargestellt.

Der gesamte Text ist durch Vorlagen definiert. Soll also eine andere Schriftart, Schriftgröße oder Schriftfarbe verwendet werden, so sollte dies am besten durch die Einstellung der Vorlagen geschehen, die durchgängig mit dem Namen «LibO» beginnen.

Für den Ausdruck existieren 2 Vorlagen, die sich in der Positionierung des Adressfeldes und der Falzmarken unterscheiden:



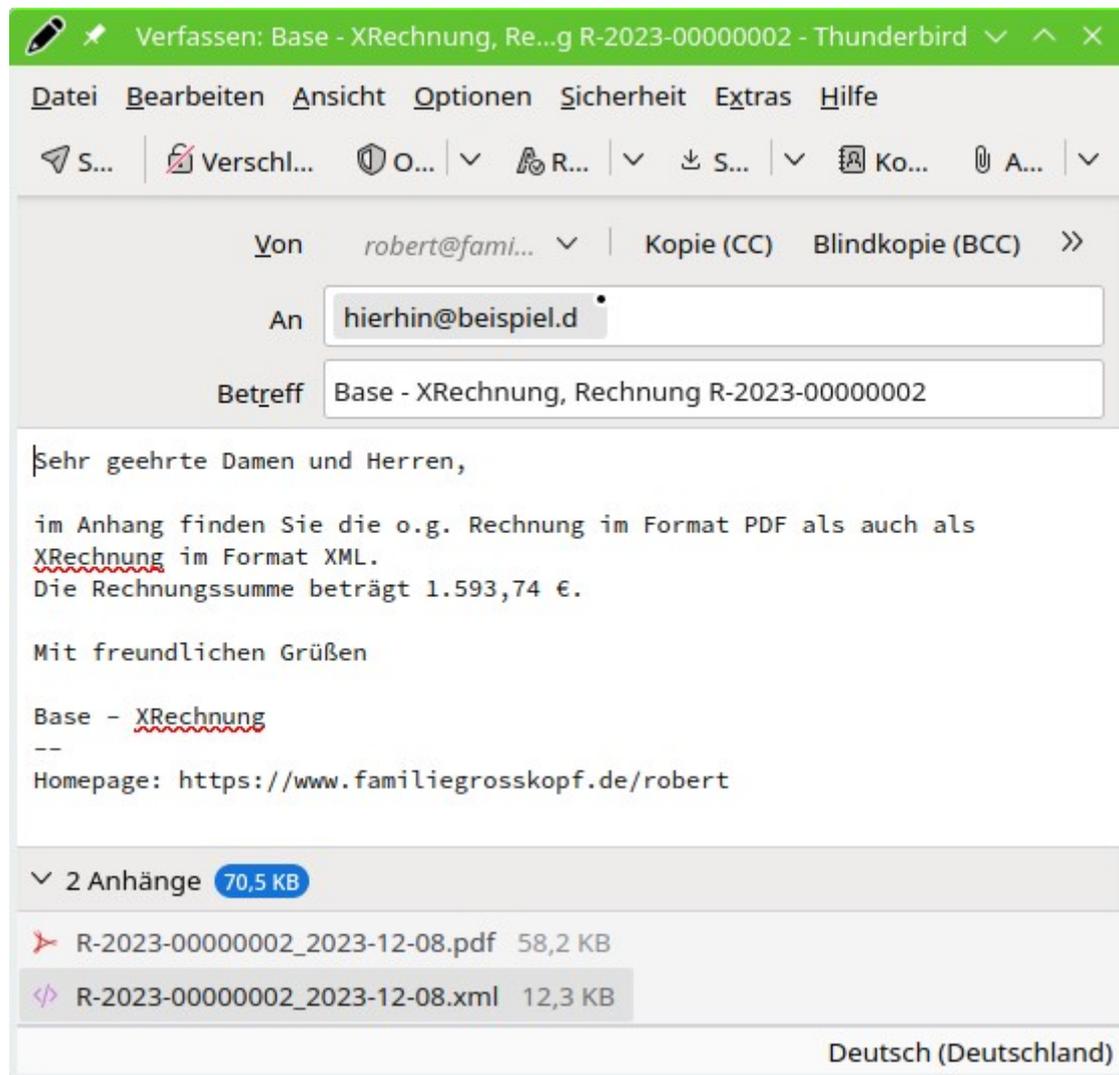
Briefvorlage nach DIN 5008, Typ A. Hier ist das Feld für die Anschrift weit oben positioniert. So eine Vorlage ist bei großen Briefumschlägen üblich. Da die XRechnung sowieso nicht als Brief versendet wird ist dies auch die Vorlage, die als "Vorlage_Rechnung_Extension.ott" als Standardvorlage vorgesehen ist. So bleibt mehr Platz für den eigentlichen Rechnungsinhalt.



Briefvorlage nach DIN 5008, Typ B. Hier ist das Feld für die Anschrift weiter unten positioniert als im Typ A. So eine Vorlage ist bei normalen Briefumschlägen mit 2-facher Falzung üblich. Soll diese Vorlage genutzt werden, so muss sie mit dem Namen der Standardvorlage «Vorlage_Rechnung_Extension.ott» versehen werden.

Beim Aufruf der Druckfunktion wird die Vorlagendatei entsprechend gefüllt und die Tabelle «Rechnungsinhalt» beständig erweitert. Schließlich entsteht die Writer-Datei.

Anschließend werden eine Writer-Datei, eine PDF-Datei und die über Makro erzeugte XRechnung-Datei im Archiv gelagert. Die Writer-Datei bleibt noch geöffnet. Im Hintergrund meldet sich schon das Mailprogramm:



Eine Mail wird komplett mit Inhalt und Anhang erstellt. Die PDF-Datei und die XRechnungs-Datei sind im obigen Screenshot unten erkennbar. Der Text der Mail wurde so gestaltet, dass auch der Rechnungsbetrag gleich sichtbar ist. Im Betreff ist außerdem die Rechnungsnummer enthalten.

Makros

Die Makros werden hier nicht vom Code her beschrieben. Der Code kann bei geöffneter Datenbankdatei über **Extras → Makros → Makros bearbeiten** eingesehen werden. Dort sind auch teilweise weitere Kommentare hinzugefügt. In diesem Kapitel werden lediglich die Prozeduren und ihre Verbindung zu Formularen und eventuell anderen Prozeduren aufgeführt. Zusätzlich erfolgt eine kurze Erklärung, was die entsprechende Prozedur bewirkt.

Die Gliederung dieses Kapitel entspricht der Gliederung im Makroeditor. Die Prozeduren sind also in der Reihenfolge aufgeführt, wie sie in den jeweiligen Modulen auftauchen. Die Module werden im Editor nach dem Alphabet geordnet angezeigt.

✓ Hinweis

Die Bezeichnung der Prozeduren sowie der Variablen ist teilweise deutschsprachig, dann wieder der englischen Sprache angelehnt. Dies liegt auch daran, dass sie zu unterschiedlichen Zeiten und unterschiedlichen Zwecken erstellt wurden und nicht in dieser Datenbankdatei zum ersten Mal genutzt werden.

Anhang

AnhangEinlesen

Aufruf aus
Formular: <i>frm_Rechnung, frm_Lieferung_Aenderung</i>

Benötigt
Makro: <i>AnhangPfad</i>

Mit dieser Prozedur wird entschieden, ob eine Datei eingelesen werden kann (Dateityp und Datei müssen übereinstimmen mit dem Mimecode). Beim Löschen einer Datei wird nach dem Pfad gesucht, in dem die bisherige Datei sowie die Datei im Base64-Format gespeichert werden. So wird nicht nur der Pfad zu den Dateien sondern auch die Dateien selbst gelöscht.

AnhangSpeichern

Aufruf aus
Formular: <i>frm_Rechnung, frm_Lieferung_Aenderung</i>

Benötigt
Makro: <i>ByteStreamToBase64Stream, AnhangPfad</i>

Soll in die XRechnung eine Datei eingefügt werden, so geschieht dies im Base64-Format. Mit dieser Prozedur wird der ausgewählte «Anhang» (eingebundene Datei) zuerst in das Archiv kopiert und mit der Rechnungsnummer verknüpft. Diese Datei wird anschließend angezeigt. Darüber hinaus wird eine Datei im Base64-Format erstellt, die in die XRechnung eingebunden werden kann.

AnhangPfad

Aufruf aus
Makro: <i>AnhangEinlesen, AnhangSpeichern</i>

BenötigtTabelle: [tbl_Rechnung](#)

Der Pfad zum Anhang wird anhand der Rechnungsnummer und des Rechnungsdatums erstellt.

Backup

DatabaseStart

Aufruf aus

Datenbankdatei → Extras → Anpassen → Ereignisse → Dokument öffnen

BenötigtMakro: [DatabaseBackup](#), [FilterStart](#)

Hier werden lediglich zwei Prozeduren hintereinander gestartet und mit dem gleichen Ereignis verbunden. Dies ist zuerst die Prozedur «DatabaseBackup», die nach 10 Backups jeweils die älteste Backup-Datei löscht. Danach folgt die Prozedur «FilterStart».

DatabaseClose

Aufruf aus

Datenbankdatei → Extras → Anpassen → Ereignisse → Unterdokument wurde geschlossen

BenötigtMakro: [DataWrite](#)

Schließt automatisch die Datenbankdatei, wenn das letzte Unterdokument geschlossen wurde. Dies betrifft vor allem die Formularbedienung des Normalusers. Wird das geöffnete Formular geschlossen, so wird damit auch die Base-Datei geschlossen – es sei denn, es ist z. B. noch der Makroeditor mit Bezug zur Base-Datei geöffnet.

In der Version für LO 24.8 ist das Schließen der Datenbankdatei noch ausgeschlossen, wenn nur die Datenbankdatei geöffnet ist. Aus bisher unbekanntem Gründen führt dies, abhängig vom Betriebssystem, zu einem Hängen oder einem Absturz von LO.

DatabaseClose1

Dummy, falls an Formularen der Datenbank oder sonst wie am Code gearbeitet werden soll und ein automatisches Schließen unterbunden werden soll. Dann einfach «DatabaseClose1» ändern zu «DatabaseClose» und das bisherige «DatabaseClose» zu «DatabaseClose1». Dann läuft statt der eigentlichen Prozedur eine leere Prozedur ab.

DatabaseBackup

Aufruf ausMakro: [DatabaseStart](#)**Benötigt**Makro: [BackupTarget](#)

Von der Datenbankdatei sowie einer eventuell extern liegenden FIREBIRD-Datei wird eine Kopie beim Start der ODB-Datei angelegt. Die Anzahl der Kopien wird beim Aufruf festgelegt und ist zum Start auf maximal 10 Kopien eingestellt. Ist die Maximalanzahl erreicht, dann wird die älteste Kopie durch eine neue Kopie ersetzt.

BackupTarget

Aufruf aus
Makro: <i>DatabaseBackup</i>

Diese Funktion ermittelt den korrekten Pfad und die entsprechende Benennung der Backup-Datei. Hier wird das Verzeichnis genutzt, das unter **Extras → Optionen → LibreOffice → Pfade → Sicherungskopien** als Backupverzeichnis hinterlegt ist. Existiert das Verzeichnis, wie bei einer Neuinstallation, noch nicht, so wird es angelegt.

FilterStart

Aufruf aus
Makro: <i>DatabaseStart</i>

Benötigt
Tabelle: <i>tbl_Filter</i>
Ansicht: <i>viw_Filter</i>
Formular: <i>frm_Rechnung</i>

Zum Start der Datenbankdatei muss in Mehrbenutzersystemen die aktuelle Verbindungs-ID in die Filtertabelle geschrieben werden. Erst dann ist es möglich, die Filterung, die z. B. für den Formularwechsel benötigt wird, korrekt ablaufen zu lassen.

Neben der korrekten Filterung wird in dieser Prozedur auch das Formular festgelegt, das beim Start der Datenbankdatei automatisch geöffnet wird. Standardmäßig ist dies das Formular, mit dem erweiterte XRechnungen erstellt werden können.

DataWrite

Aufruf aus
Datenbankdatei → Extras → Anpassen → Ereignisse → Ansicht wird geschlossen
Makro: <i>DatabaseClose</i>

Soll die Datenbankdatei geschlossen werden, so ist bei FIREBIRD je nach Version und genutzter Datenverbindung noch notwendig, die Datenbankdatei abzuspeichern. Diese Makro macht das automatisch, so dass keine Nachfrage beim Schließen der Datenbankdatei kommt. Ausnahme ist natürlich, dass irgendein Inhalt geändert wurde, der das Design der Datenbankdatei betrifft und nicht nur Daten in die Datenbank eingegeben wurden.

Die interne HSQLDB macht diese Speicherung seit Bestehen von LibreOffice automatisch.

Base64Binaer

Anhänge werden in die XRechnungen eingebunden. Die Einbindung erfolgt im lesbaren Base64-Code. Dieser Code muss umgewandelt werden in Binärcode, damit daraus wieder entsprechend darstellbare Dateien entstehen. Diese Prozeduren sind teilweise Kopien aus einem Dokument¹²,

¹² <https://ask.libreoffice.org/t/base64-encode-decode/25761>

das Regina Henschel erstellt hat. Sie sind leider recht langsam, erfüllen aber ihren Zweck, da wohl nicht häufig mit Anhängen in den XRechnungen gearbeitet wird.

Base64StringToByteStream

Aufruf aus

Makro: [Import](#)

Benötigt

Makro: [GetBitsLeft](#), [GetBitsRight](#), [EncodedToSixBit](#)

Diese Prozedur macht aus dem eventuell in einer XRechnung enthaltenen Anhang im Base64-Format wieder eine Datei im Binärformat, die anschließend z. B. mit einem Bildbetrachtungsprogramm, einem PDF-Betrachter oder Calc (Tabellenkalkulationsdateien und CSV-Dateien) geöffnet werden können.

GetBitsLeft

Aufruf aus

Makro: [Base64StringToByteStream](#)

GetBitsRight

Aufruf aus

Makro: [Base64StringToByteStream](#)

EncodedToSixBit

Aufruf aus

Makro: [Base64StringToByteStream](#)

SixBitToEncoded

Aufruf aus

Makro: [ByteStreamToBase64Stream](#)

ByteStreamToBase64Stream

Aufruf aus

Makro: [AnhangSpeichern](#)

Benötigt

Makro: [GetBitsLeft](#), [GetBitsRight](#), [SixBitToEncoded](#)

Erstellt aus einer binären Datei eine Datei in Format Base64, wie sie beim Einfügen der Datei in eine XRechnung notwendig ist.

Design

Das Modul «Design» hat zum Start erst einmal einige globale Variablen notiert, die für das Einblenden und Ausblenden von Feldern im Formular «Kunde» bei der Eingabe von Rechnungszeilen notwendig sind.

ToolbarsHide1

Diese Prozedur ist als Dummy gedacht. Formulare werden durch die Prozedur «ToolbarsHide» ohne eine Menüleiste ausgegeben. Dies kann beim bearbeiten von Formularen (nicht bei der Dateneingabe) zu Problemen führen. Bevor ein Formular bearbeitet wird kann hier schlicht der Name getauscht werden, so dass eben das Makro ohne Inhalt abläuft und die Menüleiste sichtbar wird.

ToolbarsHide

Aufruf aus
Formular: <i>frm_Einstellungen, frm_Rechnung, frm_Lieferung, frm_Zusatz, frm_Lieferung_Aenderung, frm_Konto</i>
Benötigt
Tabellen: <i>tbl_Firma</i>
Makro: <i>ShowColumns</i>

Hier werden Formulare in kompletter Bildschirmgröße als Fenster dargestellt. Außerdem wird die Größe auf den Maximalwert gezoomt. Dazu wird in der Prozedur angegeben, wie viele Pixel das größte Formular in der Originalgröße (Zoom 100 %) hat.

Ist das Formular aufgezogen, so werden alle bisher genutzten Werkzeugleisten, auch die Seitenleiste und die Menüleiste, ausgeblendet.

ToolbarsShow

Aufruf aus
Formular: <i>frm_Einstellungen, frm_Rechnung, frm_Lieferung, frm_Zusatz, frm_Lieferung_Aenderung, frm_Konto</i>

Über diese Prozedur werden zumindest einige Elemente wie die Menüleiste und die Seitenleiste wieder beim Schließen eines Formulars auf sichtbar geschaltet. Gegebenenfalls können hier weitere Elemente hinzugefügt werden, falls sie nicht automatisch auftauchen.

ShowAttachment

Aufruf aus
Formular: <i>frm_Rechnung, frm_Lieferung, frm_Lieferung_Aenderung</i>

Anzeige von Anhängen, die sonst nur in einem grafischen Kontrollfeld verkleinert und mit lediglich einer Seite präsentiert werden.

ChangeColumnName

Aufruf aus
Makro: <i>FormChange</i>
Formular: <i>frm_Rechnung</i>

Benötigt
Makro: <i>MainNurTitel</i>

Beim Rechnungsausdruck werden die Spaltennamen «Beschreibung», «Warennummer» und «Warenzusatz» angeboten. Alternativ hierzu können andere Benennungen erfolgen. Diese Benennungen finden sich dann auch in dem Formular zur Eingabe der Rechnungszeilen bei den Kunden wieder.

ShowColumns

Aufruf aus
Makro: <i>ToolbarsHide</i>

Benötigt
Formular: <i>frm_Rechnung</i>

Zeige die Spalten, die in dem Einstellungsformular bzw. der Tabelle "tbl_Firma" als anzuzeigende Spalten gekennzeichnet sind. Verberge die Spalten, die dort abgewählt sind.

MainNurTitel

Aufruf aus
Makro: <i>ChangeColumnName</i>

Benötigt
Formular: <i>frm_Rechnung</i>

Zeige im Kundenformular im Bereich «Rechnungsposten» nur eine Spalte für den Titel (und ggf. eine Spalte für die Steuer) an. Dies dient zur Untergliederung der später ausgedruckten Rechnung.

TitelDruckSetzen

Aufruf aus
Makro: <i>FormChange</i>

Benötigt
Formular: <i>frm_Rechnung</i>

Wenn nur der Titel in der vorhergehenden Prozedur angeschaltet wurde, dann soll der Titel auch für den Druck vorausgewählt werden. Es werden also nicht komplette Zeilen ausgegeben.

DefaultSteuercodeErstellen

Aufruf aus
Formular: <i>frm_Einstellungen</i>

Stellt, wenn in dem Einstellungsformular die Anzeige der Steuer ausgeschaltet werden soll, die Eingabe für den Standardsteuersatz ein und setze ihn auf '19%'.

SelectContent

Aufruf aus
Formular: <i>frm_Rechnung, frm_Lieferung_Aenderung</i>

Bei numerischen Feldern und Datumsfeldern wird der Inhalt beim Erreichen mit dem Cursor innerhalb eines Tabellenkontrollfeldes nicht markiert. Den entsprechenden Feldern wird deshalb zur besseren Bearbeitung beim Fokuserhalt diese Prozedur zugewiesen.

Druck

FillTableCarryOver

Aufruf aus
Formular: <i>frm_Rechnung, frm_Lieferung_Aenderung</i>

Benötigt
Tabelle: <i>frm_Rechnung: tbl_Rechnung_Zusatz, tbl_Rechnung_Anhang, tbl_Rechnung_Zahlung</i> <i>frm_Lieferung_Aenderung: tbl_Lieferung_Zusatz_Aenderung, tbl_Lieferung_Anhang_Aenderung, tbl_Lieferung_Zahlung_Aenderung</i>
Ansicht: <i>frm_Rechnung: viw_Rechnung_DruckSpalten, viw_Rechnung_Spalten, viw_Rechnung_Ausgang, viw_Rechnung_Summe, viw_Rechnung</i> <i>frm_Lieferung_Aenderung: viw_Lieferung_DruckSpalten, viw_Lieferung_Spalten_Aenderung, viw_Lieferung_Eingang_Aenderung, viw_Lieferung_Summe_Aenderung, viw_Lieferung_Aenderung</i>
Formular: <i>frm_Rechnung, frm_Lieferung_Aenderung</i>
Makro: <i>ExportImageIntern, FillTextfields, ExportPDF, SQLDateToDateYY, CellSetNumberFormat</i>

Diese Prozedur startet den *Ausdruck* und erweitert in der Vorlagedatei vor allem die Tabelle, die die einzelnen Rechnungsposten aufnimmt. Es existiert in der Vorlage eine leere Zeile in der entsprechenden Tabelle. Zuerst wird eine weitere leere Zeile angehängt, so dass die Formatierungen übernommen werden. Dann wird in die vorherige leere Zeile der aktuell ausgelesene Inhalt, also eine Rechnungszeile, eingefügt.

Springt die Tabelle mit einer Zeile auf die nächste Seite, so wird auf der vorhergehenden Seite eine leere Zeile für den Übertrag eingefügt. Dann werden so viele zusätzlichen Zeilen eingefügt bis die erste leere Zeile auf der aktuellen (neuen) Seite liegt und auch dort wird der Übertrag in die Zeile geschrieben.

Zum Schluss wird unter die Rechnungszeilen noch die Summierung (Netto, Steuerbeträge für verschiedene Steuersätze, sonstige Beträge und der Bruttopreis) geschrieben.

Existieren in der Ansicht "viw_Rechnung" Einträge in den Feldern "RegisterE1", "RegisterE2" oder "RegisterE3", so wird für diese Felder in der Fußzeile eine zusätzliche Spalte eingefügt und die entsprechenden Werte eingefügt.

Nachdem die Tabelle fertig gestellt ist wird zuerst die Prozedur «Textfelder_fuellen_ID» und danach die Prozedur «PDF_Export» ausgeführt.

SQLDateToDate

Wandelt ein Datum in SQL-Schreibweise in ein Datum im Format DD.MM.YYYY um. Wird hier allerdings nicht genutzt.

SQLDateToDateYY

Aufruf aus
Makro: <i>FillTableCarryOver</i>

Wandelt ein Datum in SQL-Schreibweise in ein Datum im Format DD.MM.YY um.

CellSetNumberFormat

Aufruf aus
Makro: <i>FillTableCarryOver</i>

Diese Funktion ermittelt zu einem Zahlenformat, wie es in Calc oder bei den Formateinstellungen in Base eingestellt werden kann, die interne Zahl. Die ist notwendig, um einer Zelle ein entsprechendes Format zuweisen zu können.

FillTextfields

Aufruf aus
Makro: <i>FillTableCarryOver</i>

In der Vorlagendatei existieren viele Textfelder, auch als Platzhalter bezeichnet. Diese Felder haben die gleichen Namen wie die Felder in "viw_Rechnung". Entspricht der Name eines Textfeldes dem Namen einer Spalte in "viw_Rechnung", dann wird durch diese Prozedur das Textfeld mit dem Inhalt aus "viw_Rechnung" gefüllt.

ExportPDF

Aufruf aus
Makro: <i>FillTableCarryOver</i>

Benötigt
Makro: <i>SaveXRechnung, StartMail, SaveZUGFeRD</i>

Dieses Modul sorgt für die Speicherung der Daten in einem Archivverzeichnis. Dazu wird ein Archiv erstellt, dass in dem Ordner «Archiv» jährlich neue Ordner anlegt und in diesen Ordnern dann noch Monatsordner erstellt. Der Dateiname für eine Datei wird aus der Rechnungsnummer und dem Datum der Rechnung zusammengesetzt. So ein Name sieht beispielsweise so aus:

«**R-2023-00000042_2023-06-17.odt**».

Die bereits erstellte Writer-Datei wird abgespeichert und geöffnet gelassen. Mit gleicher Bezeichnung, aber anderer Dateiendung wird mittels "writer_pdf_Export" eine PDF-Datei erstellt.

Zum Schluss werden die Prozeduren für die Erstellung und Speicherung der XRechnung sowie den Mailaufruf gestartet.

ExportImageIntern

Aufruf aus

Makro: *FillTableCarryOver*

Benötigt

Makro: *ImportImage*

In die Vorlagendatei soll auch ein Logo übertragen werden. Diese Prozedur liest das Logo erst einmal aus der Datenbank aus und speichert es vorübergehend in einer temporären Datei ab. Dann wird anschließend die Prozedur «ImportImage» aufgerufen.

ImportImage

Aufruf aus

Makro: *ExportImageIntern*

Hiermit wird das einmal ausgelesene Bild aus der temporären Datei in die dafür vorgesehene Position der Vorlagendatei eingelesen. Das Bild wird **als Zeichen** verankert, da so kein zusätzlicher Platz in der Tabelle für eine weitere Textzeile benötigt wird.

EingangBearbeiten

Import_to_Aenderung

Aufruf aus

Formular: *frm_Lieferung_Aenderung*

Benötigt

Tabelle: *tbl_Lieferung_Aenderung, tbl_Lieferung, tbl_Eingang_Aenderung, tbl_Eingang, tbl_Lieferung_Anhang_Aenderung, tbl_Lieferung_Anhang, tbl_Lieferung_Zusatz_Aenderung, tbl_Lieferung_Zusatz*

Überträgt die Inhalte aus einer eingegangenen Lieferung in die entsprechenden Änderungstabelle. So bleibt die alte Lieferung bestehen und die Änderung kann separat abgespeichert werden.

ListboxSave

Sichert den Inhalt einer Listbox und führt anschließend ein Update der Tabelle durch. Wird zur Zeit nicht genutzt.

Filter

FormParallelReload

Aufruf aus

Formular: *frm_Rechnung, frm_Lieferung, frm_Konto*

Makro: *MainRowSaved*

Hier wird lediglich ein Formular, das in beiden auslösenden Formulardokumenten den gleichen Namen hat, neu geladen. Es handelt sich dabei um das Formular, das die Summierung der

Rechnungsbeträge bzw. Lieferbeträge anzeigt. So bleibt das Formular auch bei laufenden Datenänderungen immer auf dem neuesten Stand.

ListboxFilter

Aufruf aus
Formular: <i>frm_Rechnung, frm_Lieferung, frm_Lieferadresse</i>

Über diese Funktion erfolgt eine Filterung der jeweiligen Eingabeformulare mittels eines Listenfeldes. In den Eigenschaften des Listenfeldes ist in den Zusatzinformationen der Name des Formulars vermerkt, dass nach dem Abspeichern des Datensatzes aus dem Listenfeld aktualisiert werden soll.

Listenfeldfilter_allround

Ein Filter für Listenfelder, der mehrere Listenfelder auf entsprechende Filterwerte einstellen kann. Wird zur Zeit nur in abgespeckter Version als «Listenfeldfilter» genutzt.

Listenfeldfilter

Aufruf aus
Formular: <i>frm_Lieferung_Aenderung</i>

Wenn eine Lieferung zum Ändern eingelesen werden soll, sollen in dem Listenfeld nur die Lieferungen zur Verfügung stehen, die bisher nicht zum Ändern eingelesen wurden. Das Listenfeld muss also nach jedem Einlesen neu eingestellt werden.

ListenfeldfilterRechnungsreferenz

Aufruf aus
Formular: <i>frm_Zusatz</i>

Benötigt
Tabelle: <i>tbl_rel_Rechnung</i>
Ansicht: <i>viw_Rechnung</i>

Für die Abschlagszahlungen darf eine Rechnung nur einmal einer anderen Rechnung zugewiesen werden. Zusätzlich ist es auch möglich, eine Rechnung sich selbst zuzuweisen (Abschlagszahlungen ohne vorausgehende Rechnungen).

FormsReload

Aufruf aus
Formular: <i>frm_Konto</i>

Nach dem Abspeichern werden die entsprechenden Formulare neu geladen, damit das Ergebnis der Speicherung der Zahlung im Unterformular sowohl im darüber liegenden Formular als auch im Kontostand sichtbar wird.

Rechnungsfilter

Aufruf aus
Formular: <i>frm_Konto</i>

Rechnungen sollen nach Rechnungsnummer und Kunden gefiltert werden. Dafür werden in Listenfeldern Werte ausgesucht, in "tbl_Filter" geschrieben und anschließend alle Formulare der obersten Ebene im Formular frm_Konto neu geladen.

Formularwechsel

Navigation

Aufruf aus
Formular: <i>frm_Einstellungen, frm_Rechnung, frm_Lieferung, frm_Konto, frm_Lieferung_Aenderung</i>

Der Wert aus der auslösenden Listbox, das heißt der Zielname für das nächste Formular, wird abgespeichert. Anschließend wird das bestehende Formular geschlossen und das neue Formular geöffnet.

FormStart

Aufruf aus
Formular: <i>frm_Einstellungen, frm_Rechnung, frm_Lieferung, frm_Konto, frm_Lieferung_Aenderung</i>

Wird ein Formular von der Datenbankoberfläche aus geladen, so stimmt die Anzeige des Formularnamens in dem Listenfeld nicht unbedingt. Deswegen wird mit dieser Prozedur das Formular für das Listenfeld, das zur Navigation dient, mit dem aktuellen Fenster in Übereinstimmung gebracht, indem der Fenstername abgespeichert und das Formular neu geladen wird.

KundeLieferadresse

Aufruf aus
Formular: <i>frm_Rechnung, frm_Lieferung, frm_Lieferung_Aenderung</i>

Benötigt
Tabelle: <i>tbl_Filter</i>

Die Lieferung muss nicht unbedingt an die Adresse erfolgen, die auch die Rechnungsadresse ist. Daher hier die Eingabemöglichkeit einer alternativen Lieferadresse.

LieferadresseBeenden

Aufruf aus
Formular: <i>frm_Lieferadresse</i>

Schließt das Formular und aktualisiert das Listenfeld für die Lieferadresse, so dass die neue (alternative) Lieferadresse direkt zur Auswahl verfügbar ist.

KundeRechnungszusatz

Aufruf aus
Formular: <i>frm_Rechnung, frm_Lieferung, frm_Lieferung_Aenderung</i>

Benötigt
Tabelle: <i>tbl_Filter</i>
Abfrage: <i>qry_frmZusatz-Rechnung, qry_frmZusatz-Lieferung, qry_frmZusatz-Lieferung_Aenderung</i>

Die Formulare können manchmal nicht alle Felder aufnehmen. Hiermit wird ein Unterformular gestartet, das nur den aktuellen (bereits gespeicherten) Datensatz mit zusätzlichen Daten ergänzt. Da das Formular sowohl aus dem Rechnungsformular als auch aus dem Lieferformular gestartet werden kann, werden unterschiedliche Felder sichtbar geschaltet.

RechnungszusatzBeenden

Aufruf aus
Formular: <i>frm_Zusatz</i>

Mit den zusätzlichen Eingaben können auch andere Rechnungen die Summe der Rechnung beeinflussen. Deswegen wird über einen Button das Formular beendet und das Formular «frm-Summe» in dem Formular neu eingelesen, das das Formular «frm_Zusatz» gestartet hat.

RechnungenGesperrt

Aufruf aus
Formular: <i>frm_Rechnung</i>

Alle Rechnungen, die bereits bezahlt sind, sollen nicht mehr bearbeitet werden. Das ist durch unterschiedliche Formulare gelöst. Endet der Name des Formulars mit «_all», so ist dies das entsprechenden Formular, das grundsätzlich schreibgeschützt ist und sämtliche Rechnungen anzeigen kann.

Kundezusatz

Aufruf aus
Formular: <i>frm_Rechnung</i>

Benötigt
Tabelle: <i>tbl_Filter</i>
Ansicht: <i>viw_Filter</i>

Die Formulare können manchmal nicht alle Felder aufnehmen. Hiermit wird ein Unterformular gestartet, das nur den aktuellen (bereits gespeicherten) Datensatz mit zusätzlichen Daten ergänzt. Es handelt sich hier um Daten, die in der *tbl_Kunde* gespeichert werden.

KundezusatzBeenden

Aufruf aus
Formular: <i>frm_Zusatz_Kunde</i>

Das Formular wird durch einen Button lediglich geschlossen. Ein Update des vorher aufrufen- den Formulars mit den neu eingegebenen Daten ist nicht erforderlich. Wird stattdessen das Fenster direkt geschlossen, so erfolgt bei Datenänderung eine Nachfrage, ob die Daten gespei- chert werden sollen.

Lieferanzusatz

Aufruf aus
Formular: frm_Lieferung , frm_Lieferung_Aenderung

Benötigt
Tabelle: tbl_Filter
Ansicht: viw_Filter

Die Formulare können manchmal nicht alle Felder aufnehmen. Hiermit wird ein Unterformular gestartet, das nur den aktuellen (bereits gespeicherten) Datensatz mit zusätzlichen Daten ergänzt. Es handelt sich hier um Daten, die in der `tbl_Lieferant` gespeichert werden.

LieferanzusatzBeenden

Aufruf aus
Formular: frm_Zusatz_Lieferant

Das Formular wird durch einen Button lediglich geschlossen. Ein Update des vorher aufrufen- den Formulars mit den neu eingegebenen Daten ist nicht erforderlich. Wird stattdessen das Fenster direkt geschlossen, so erfolgt bei Datenänderung eine Nachfrage, ob die Daten gespei- chert werden sollen.

Einstellungenzusatz

Aufruf aus
Formular: frm_Rechnung , frm_Einstellungen

Benötigt
Tabelle: tbl_Filter
Ansicht: viw_Filter
Abfrage: qry_frmZusatz-Einstellungen

Die Formulare können manchmal nicht alle Felder aufnehmen. Hiermit wird ein Unterformular gestartet, das zur Zeit zusätzliche Einstellungen für das Erstellen der Tabelle in der PDF-Datei enthält.

EinstellungenzusatzBeenden

Aufruf aus
Formular: frm_Rechnung , frm_Einstellungen

Das Formular wird durch einen Button lediglich geschlossen. Ein Update des vorher aufrufen- den Formulars mit den neu eingegebenen Daten ist nicht erforderlich. Wird stattdessen das Fenster direkt geschlossen, so erfolgt bei Datenänderung eine Nachfrage, ob die Daten gespei- chert werden sollen, sofern dies nicht sowieso schon durch das Verlassen des Datensatzes erfolgt ist.

Import

InsertSQL

Aufruf aus
Makro: <i>Import</i>

Dies ist eine Allroundfunktion zur Vorbereitung von Daten für den Insert-Befehl. Das Makro sorgt bei entsprechendem Inhalt dafür, dass für leere Felder NULL eingefügt wird und dass Texte mit einfachen Anführungszeichen «'» so maskiert werden, dass das einfache Anführungszeichen anschließend nicht als Textende interpretiert wird.

XMLCharToChar

Aufruf aus
Makro: <i>Import</i>

XML-Dateien dürfen bestimmte Sonderzeichen nicht enthalten. Beim Export wurde diese Zeichen in HTML-Schreibweise umgewandelt. Diese Prozedur nimmt die entsprechende Umwandlung wieder zurück.

DecimalPoint

Aufruf aus
Makro: <i>Import</i>

Wenn Double-Variablen in eine Datenbank eingefügt werden sollen, dann führt das bei der Umwandlung zu Text dazu, dass das in Deutschland übliche Dezimaltrennzeichen genutzt wird. Damit lassen sich keine Werte in die Datenbank schreiben. Der Dezimaltrenner wird als Punkt in einem Text festgeschrieben.

Import

Aufruf aus
Formular: <i>frm_Lieferung</i>

Benötigt
Tabelle: <i>tbl_Lieferant, tbl_Lieferung, tbl_Lieferung_Zusatz, tbl_Eingang</i>
Makro: <i>InsertSQL, XMLCharToChar, DecimalPoint</i>

Diese Prozedur dient dazu, XRechnungsdateien der Formate UBL und CII in die Datenbank als Lieferantendaten ein zu lesen. Die Prozedur beschränkt sich auf das Auslesen der Daten, die in der Datenbank auch ein entsprechendes Feld vorfinden.

Es wird in einem ersten Zugriff auf die Dateien grundsätzlich der Inhalt in ein Array eingelesen, der sich auf die Rechnungszeilen bezieht. Der Inhalt kommt mehrmals vor und muss auch in einer separaten Tabelle abgespeichert werden.

Anschließend werden die Grundlagendaten ausgelesen und in die entsprechenden Tabellenfelder übertragen. Passiert dabei ein Fehler, weil z. B. ein benötigtes Feld nicht vorhanden ist, dann wird das Einlesen abgebrochen und eine Fehlermeldung erzeugt.

Auch der Dateninhalt für den Empfänger wird zumindest ausgelesen. Ihn abzuspeichern macht aber keinen Sinn, da ja der Empfänger (eigentlich) die Firma ist, die die Rechnung erhalten hat.

Hier könnte beispielsweise der Inhalt in einer Messagebox angezeigt werden, so dass klar ist: Die Datei passt zu dem besagten Empfänger und ist nicht irrtümlich zugestellt.

Inputcontrol

FormVars

Aufruf aus
Formular: <i>frm_Einstellungen, frm_Rechnung, frm_Lieferung, frm_Lieferadresse</i>

Benötigt
Makro: <i>FormChange</i>

Mit dieser Prozedur werden beim Start eines Formulars alle Felder in ein Array übernommen, bei denen die Eingabe notwendig ist. Auch die Farbe für die Umrandung dieser Felder bei fehlendem Inhalt wird festgelegt. Zum Schluss wird nach die Prozedur «FormChange» aufgerufen, um die Felder entsprechend zu markieren.

FormChange

Aufruf aus
Formular: <i>frm_Einstellungen, frm_Rechnung, frm_Lieferung, frm_Lieferadresse</i>
Makro: <i>FormVars</i>

Benötigt
Makro: <i>ChangeColumnName, TitelDruckSetzen</i>

Nach dem Datensatzwechsel wird geschaut, ob die benötigten Felder einen Inhalt aufweisen. Enthalten sie, wie bei neuen Datensätze, keinen Inhalt, dann erscheinen die Felder rot umrandet.

FieldRequired

Aufruf aus
Formular: <i>frm_Einstellungen, frm_Rechnung, frm_Lieferung, frm_Lieferadresse</i>
Makro: <i>MailValid_FieldRequired, IBANValid_FieldRequired</i>

Wird beim Fokusverlust von Feldern ausgelöst, die einen Eintrag erfordern. Ist ein Eintrag notwendig und nicht vorhanden, so erscheinen die Felder rot umrandet.

SaveRequired

Aufruf aus
Formular: <i>frm_Einstellungen, frm_Rechnung, frm_Lieferung, frm_Lieferadresse</i>

Mit dieser Funktion wird das Abspeichern abgefangen, wenn Felder mit einer notwendigen Eingabe noch rot umrandet erscheinen. Es erscheint eine Fehlermeldung, dass noch Einträge notwendig sind.

MailValid

Aufruf aus
Makro: <i>MailValid_FieldRequired</i>

Hiermit wird eine E-Mail-Adresse auf ihre Gültigkeit überprüft. Es wird bei einem auftretenden Fehler mitgeteilt, welches Element nicht korrekt ist. Dabei handelt es sich nicht um eine Überprüfung, ob die Mail tatsächlich existiert.

MailValid_FieldRequired

Aufruf aus
Formular: <i>frm_Einstellungen, frm_Rechnung, frm_Lieferung</i>

Benötigt
Makro: <i>MailValid, FieldRequired</i>

Zwei Prozeduren werden direkt nacheinander gestartet: Zuerst die Überprüfung der Mail-Adresse und dann noch, ob das Feld eventuell mit Inhalt gefüllt ist oder komplett leer geblieben ist.

IBANValid

Aufruf aus
Formular: <i>frm_Zusatz_Kunde</i>
Makro: <i>IBANValid_FieldRequired</i>

Hiermit wird eine IBAN auf ihre Gültigkeit überprüft. Funktioniert auch mit IBAN-Angaben, die nicht mit «DE» beginnen.

IBANValid_FieldRequired

Aufruf aus
Formular: <i>frm_Einstellungen, frm_Lieferung</i>

Benötigt
Makro: <i>IBANValid, FieldRequired</i>

Zwei Prozeduren werden direkt nacheinander gestartet: Zuerst die Überprüfung der IBAN und dann noch, ob das Feld eventuell mit Inhalt gefüllt ist oder komplett leer geblieben ist.

SaveSubRow

Aufruf aus
Formular: <i>frm_Rechnung</i>

Von einem Unterformular aus soll eine Änderung in einem Wert des Hauptformulars erzeugen. Dazu wird das Unterformular gespeichert, das Hauptformular neu eingelesen und das Unterformular auf den zuletzt bearbeiteten Datensatz gesetzt.

BeforeSaveSubRow

Aufruf aus
Formular: <i>frm_Rechnung</i>

Benötigt
Makro: <i>ValueUnique</i>

Von einem Unterformular aus soll eine Änderung in einem Wert des Hauptformulars erzeugen. Dazu wird vor dem Speichern des Unterformulars der Primärschlüssel des Hauptformulars in eine Variable übernommen. Das ist notwendig, falls die Speicherung durch das Verlassen des Tabellenkontrollfeldes und erstellen eines neuen Datensatzes in dem Hauptformular erfolgt.

SaveMainRow

Aufruf aus
Formular: <i>frm_Rechnung</i>

Benötigt
Makro: <i>ValueUnique</i>

Dient nur dazu, bei einer Speicherung den Primärschlüssel des Datensatzes in einer globalen Variablen zu speichern, damit mit diesem Schlüsselwert der richtige Datensatz im Hauptformular wieder gefunden werden kann.

MainRowSaved

Aufruf aus
Formular: <i>frm_Rechnung</i>

Benötigt
Makro: <i>ChangeSubToMain, FormParallelReload, ChangeMainToSub</i>

Wird nach der Sicherung aus dem Hauptformular aufgerufen. Startet lediglich zwei Prozeduren nacheinander.

ChangeSubToMain

Aufruf aus
Makro: <i>MainRowSaved</i>

Benötigt
Tabelle: <i>tbl_Ausgang</i>

Übertragung von Werten aus dem Unterformular in das Feld des Hauptformulars.

ChangeMainToSub

Aufruf aus
Makro: <i>MainRowSaved</i>

Benötigt

Tabelle: [tbl_Ausgang](#)

Bei Änderung von Werten im übergeordneten Formular müssen bei den verlinkten Feldern die Werte angepasst werden. Sonst verlieren die Daten des Unterformulars den Kontakt zum übergeordneten Formular.

ValueUnique

Aufruf aus

Formular:

Makro: [BeforeSaveSubRow](#), [SaveMainRow](#)

Benötigt

Tabelle: [tbl_Ausgang](#)

Hiermit wird sicher gestellt, dass bei einer selbst erstellten Zeilennummerierung nur unterschiedliche Texte eingegeben werden. Die Zeilen müssen für die XRechnung eindeutig sein.

Kopieren

RechnungCopyInsert

Aufruf aus

Formular: [frm_Rechnung](#)

Benötigt

Tabelle: [tbl_Rechnung](#), [tbl_Rechnung_Zusatz](#), [tbl_Rechnung_Anhang](#), [tbl_Ausgang](#)

Von einer bestehenden Rechnung wird eine Kopie erstellt und als neue Rechnung (mit neuer ID) eingefügt. Wenn oft Rechnungen mit identischem Inhalt erstellt werden ist dies die Möglichkeit, einiges an Tipparbeit zu sparen.

Update

SelectSQL

Die Prozedur wird zur Zeit nicht benötigt. Sie würde einen Wert so darstellen, dass er in einer Abfrage keine Probleme bereiten kann.

Fields

Aufruf aus

Makro: [Import_Values](#)

Aus der zu importierenden Datenbank und der aktuellen Datenbank werden die Felder der verfügbaren Tabellen ausgelesen. Die Felder werden miteinander verglichen und nur gleichnamige Felder in gleichnamigen Tabellen werden von der abgebenden Datenbank in die aufnehmende Datenbank aufgenommen.

Import_all

Aufruf aus
Datenbankdatei → Extras → Makros → Makro ausführen...

Benötigt
Tabelle: <i>tbl_Rechnung_Zahlung, tbl_Lieferung_Zahlung_Aenderung, tbl_Lieferung_Zahlung, tbl_Rechnung, tbl_Lieferung_Aenderung, tbl_Lieferung, tbl_Kunde, tbl_Lieferant, tbl_Mitarbeiter, tbl_Firma</i>
Makro: <i>Import_Values, TableRestart</i>

Das Makro wird nur direkt gestartet, nicht über die grafische Oberfläche. Zuerst werden alle Tabellen geleert, anschließend mit dem neuen Inhalt gefüllt und auf den nächsten Wert für die automatisch erstellten Primärschlüssel gesetzt. Zum Leeren werden nur die Tabellen benötigt, die nicht mit einem Fremdschlüsse und **On Delete** → **Cascade** mit einer Ursprungstabelle verbunden sind.

Das Makro setzt voraus, dass die Datenbank, die bisher genutzt wurde, als "Import_XR" in LibreOffice als Datenbank angemeldet ist.

Import_Values

Aufruf aus
Formular: <i>Import_all</i>

Benötigt
Tabelle: <i>tbl_Kunde, tbl_Mitarbeiter, tbl_Kunde_Lieferadresse, tbl_Rechnung, tbl_Rechnung_Anhang, tbl_Rechnung_Zahlung, tbl_Rechnung_Zusatz, tbl_rel_Rechnung, tbl_Ausgang, tbl_Lieferant, tbl_Lieferung, tbl_Lieferung_Anhang, tbl_Lieferung_Zahlung, tbl_Lieferung_Zusatz, tbl_Eingang, tbl_Lieferung_Aenderung, tbl_Lieferung_Anhang_Aenderung, tbl_Lieferung_Zahlung_Aenderung, tbl_Lieferung_Zusatz_Aenderung, tbl_Eingang_Aenderung, tbl_Firma</i>
Makro: <i>Fields</i>

Die Daten aus allen abgebenden Tabellen werden ausgelesen und, sofern die Felder in den aufnehmenden Tabellen vorhanden sind, in diese Felder eingefügt.

Wartung

TableRestart

Aufruf aus
Datenbankdatei → Extras → Makros → Makro ausführen...
Makro: <i>Import_all</i>

Benötigt
Tabelle: <i>tbl_Ausgang, tbl_Eingang, tbl_Eingang_Aenderung, tbl_Kunde, tbl_Lieferant, tbl_Lieferung, tbl_Lieferung_Aenderung, tbl_Lieferung_Anhang, tbl_Lieferung_Anhang_Aenderung, tbl_Lieferung_Zusatz, tbl_Lieferung_Zusatz_Aenderung, tbl_Mitarbeiter, tbl_Rechnung, tbl_Rechnung_Anhang, tbl_Rechnung_Zusatz, tbl_Kunde_Lieferadresse, tbl_Lieferung_Zahlung, tbl_Lieferung_Zahlung_Aenderung, tbl_Rechnung_Zahlung</i>

Wird viel mit einer Datenbank zum Testen gearbeitet, so werden die automatisch erzeugten Primärschlüssel beständig hoch geschrieben. Mit dieser Prozedur werden die Tabellen zurück auf den Wert gestellt, der als nächster Maximalwert gesetzt werden kann. Dies ist bei einer leeren Tabelle dann die '1'.

ViewsErstellen

Aufruf aus
Datenbankdatei → Extras → Makros → Makro ausführen...
Benötigt
Ansicht: <i>viw_Rechnung, viw_Rechnung_Summe, viw_Rechnung_Steuer, viw_Rechnung_Ausgang, viw_Rechnung_Spalten, viw_Lieferung_Summe, Diese Ansicht dient zum Druck der erstellten geänderten Lieferung. Sie ist vom Prinzip her genauso aufgebaut wie die viw_Rechnung_Spalten., viw_Lieferung_Aenderung, viw_Lieferung_Eingang_Aenderung, viw_Lieferung_Spalten_Aenderung, viw_Lieferung_Summe_Aenderung, viw_Lieferung_Steuer_Aenderung, viw_Filter</i>
Abfrage: Code wie bei den Ansichten, werden zum Editieren der Views erstellt

Die Ansichten bauen teilweise aufeinander auf. So benötigt die Ansicht "viw_Rechnung" die Ansicht "viw_Rechnung_Summe". "viw_Rechnung_Summe" kann so nicht bearbeitet werden. Stattdessen muss erst "viw_Rechnung" gelöscht werden, dann "viw_Rechnung_Summe" bearbeitet werden und schließlich wieder "viw_Rechnung" erstellt werden. Aus diesem Grunde sind alle Ansichten auch als Abfragen in dieser Datenbank gespeichert. Es werden die Abfragen editiert und anschließend die Ansichten über diese Prozedur komplett gelöscht und aus den Abfragen neu erstellt,

Nach so einer Aktion sollte im Tabellenordner **Ansicht → Tabellen aktualisieren** ausgeführt werden. Alternativ kann auch die Datenbankdatei geschlossen und wieder geöffnet werden.

TableContentDeleteAll

Aufruf aus
Datenbankdatei → Extras → Makros → Makro ausführen...
Benötigt
Tabelle: <i>tbl_Rechnung_Zahlung, tbl_Lieferung_Zahlung_Aenderung, tbl_Lieferung_Zahlung, tbl_Rechnung, tbl_Lieferung_Aenderung, tbl_Lieferung, tbl_Kunde, tbl_Lieferant, tbl_Mitarbeiter</i>

Sämtliche selbst eingegebenen Daten werden aus den Tabellen entfernt. Nur die Daten, die für XRechnungen in Listen vorgegeben wurden, sowie die Einträge in «tbl_Firma» erstellt wurden, bleiben bestehen.

XRechnung

SaveXRechnung

Aufruf aus
Makro: <i>ExportPDF</i>

Benötigt
Tabelle: tbl_Rechnung_Zusatz , tbl_Rechnung_Anhang , tbl_mimeCode
Ansicht: viw_Rechnung , viw_Rechnung_Ausgang , viw_Rechnung_Summe
Makro: Round2Decimalplaces , CharToXMLChar

Mit dieser Prozedur wird die XRechnungsdatei erstellt und abgespeichert. Die entsprechenden Daten werden aus den verschiedenen Ansichten und Tabellen ausgelesen und in die entsprechende XML-Struktur eingefügt. Dadurch wird eine XML-Datei erzeugt, die sauber formatiert ist und nicht alle Inhalte aneinander reiht. So ist die Datei besser für den Normaluser lesbar. Für das Auslesen der Inhalte durch ein entsprechendes Programm (oder die Importfunktion dieser Datenbankdatei) spielt die Formatierung keine Rolle.

Sind keine Daten verfügbar, so wird die Erstellung abgebrochen und eine Fehlermeldung ausgegeben. Ebenso wird die Erstellung bei einem Bruttobetrag von 0 abgebrochen, da so eine Rechnung sinnlos erscheint.

Die gesamte Prozedur ist entsprechend kommentiert und besonders mit den Hinweisen versehen, welche Felder notwendig sind. Steht dort eine **1...1**, dann ist das Feld genau einmal notwendig. Taucht eine **1...n** auf, dann kann das Feld beliebig oft vorkommen.

Round2Decimalplaces

Aufruf aus
Makro: SaveXRechnung , SaveZUGFeRD

Für die Aufsummierung bei den Rechnungsdaten müssen abschließend die Werte auf 2 Dezimalstellen gerundet werden. StarBasic liefert hier keine gesonderte Funktion. Diese Funktion erfüllt den gewollten Zweck.

CharToXMLChar

Aufruf aus
Makro: SaveXRechnung , SaveZUGFeRD

Zeichen, die im XML-Code nicht erlaubt sind, werden hier in entsprechende alternative Zeichen nach XML-Standard umgewandelt. So ist z.B. das '&' nicht direkt möglich, dürfte aber in Firmennamen wohl häufiger vorkommen.

StartMail

Aufruf aus
Makro: ExportPDF

Über den Uno-Service SimpleSystemMail oder SimpleCommadMail wird, je nach Betriebssystem, eine Mail zusammengestellt und anschließend an das Mailprogramm weiter gereicht. Die Mail enthält im Anhang die Rechnung als PDF-Datei und XRechnung-Datei.

Je nach benutzter LibreOffice-Version und Betriebssystem muss eventuell unter **Extras → Optionen → Internet → E-Mail** das passende Mailprogramm angegeben werden, das das Standardmailprogramm des Systems ist.

ZUGFeRD

SaveZUGFeRD

Aufruf aus
Makro: <i>ExportPDF</i>
Benötigt
Tabelle: <i>tbl_Rechnung_Zusatz, tbl_Rechnung_Anhang, tbl_mimeCode</i>
Ansicht: <i>viw_Rechnung, viw_Rechnung_Ausgang, viw_Rechnung_Summe</i>
Makro: <i>Round2Decimalplaces, CharToXMLChar</i>

Mit dieser Prozedur wird die XML-Datei für die ZUGFeRD-PDF-Datei erstellt und abgespeichert. Im Gegensatz zur XML-Datei für die XRechnung muss hier die Erstellung im CII - Modus (Cross-IndustrieInvoice) erfolgen. Ansonsten ähnelt das grundsätzliche Vorgehen der vorhergehenden Prozedur.