



**LibreOffice**



# Base Beispiele

## ***XRechnung mit Base auslesen und erstellen***

*LibreOffice 24.2*

# 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 können Sie in deutscher Sprache an die Adresse [discuss@de.libreoffice.org](mailto:discuss@de.libreoffice.org) 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 01.02.2024. Basierend auf der Version LibreOffice 24.2.

# Inhalt

---

<b>Einführung</b> .....	<b>6</b>
<b>Vorbereitende Schritte</b> .....	<b>7</b>
<b>Formulare in der Übersicht</b> .....	<b>10</b>
Das Kunden-Formular .....	10
Das Lieferanten-Formular .....	12
Das Einstellungs-Formular .....	12
Das Konto-Formular .....	13
Navigation durch die Formulare .....	13
Neueingabe von Daten .....	14
<b>Rechnung erstellen</b> .....	<b>15</b>
<b>Technische Grundlagen</b> .....	<b>18</b>
Erforderliche Einträge in der XRechnung .....	18
<b>Tabellen</b> .....	<b>22</b>
Rechnungserstellung im Kundenbereich .....	22
tbl_Kunde .....	23
tbl_Land .....	24
tbl_Rechnung .....	24
tbl_Mitarbeiter .....	25
tbl_Rechnung_Zusatz .....	25
tbl_Rechnung_Anhang .....	26
tbl_Ausgang .....	26
tbl_Anzahl_Code .....	27
tbl_mimeCode .....	28
Eingang von Waren über Lieferanten .....	29
tbl_Lieferant .....	29
tbl_Lieferung .....	31
tbl_Lieferung_Zusatz .....	32
tbl_Lieferung_Anhang .....	32
tbl_Eingang .....	33
Einzeltabelle ohne definierte Verbindung .....	33
tbl_Firma .....	34
tbl_Filter .....	35
Verknüpfung von Tabellen und Löschen von Datensätzen .....	35
<b>Ansichten</b> .....	<b>38</b>
viw_Filter .....	38
viw_Lieferung_Steuer .....	38
viw_Lieferung_Summe .....	39
viw_Rechnung .....	40
viw_Rechnung_Ausgang .....	42
viw_Rechnung_Steuer .....	43
viw_Rechnung_Summe .....	44
<b>Abfragen</b> .....	<b>46</b>
qry_Forms .....	46
qry_Konto .....	47
qry_Lieferung_Zahlung .....	48
qry_Rechnung_Zahlung .....	48

<b>Formulare .....</b>	<b>50</b>
frm_Einstellungen .....	50
frm_Kunde .....	52
frm_Lieferant .....	55
frm_Konto .....	58
<b>Ausdruck .....</b>	<b>60</b>
<b>Makros .....</b>	<b>63</b>
Anhang .....	63
AnhangEinlesen .....	63
AnhangLoeschen .....	63
AnhangPfad .....	63
Backup .....	64
DatabaseStart .....	64
DatabaseBackup .....	64
BackupTarget .....	64
FilterStart .....	64
DataWrite .....	65
Base64Binaer .....	65
Base64StringToByteStream .....	65
GetBitsLeft .....	65
GetBitsRight .....	65
EncodedToSixBit .....	66
SixBitToEncoded .....	66
ByteStreamToBase64Stream .....	66
Design .....	66
ToolbarsHide1 .....	66
ToolbarsHide .....	66
ToolbarsShow .....	66
ShowAttachment .....	67
Druck .....	67
FillTableCarryOver .....	67
FillTextfields .....	67
ExportPDF .....	68
ExportImageIntern .....	68
ImportImage .....	68
Filter .....	68
FormParallelReload .....	68
ListboxFilter .....	69
Formularwechsel .....	69
Navigation .....	69
FormStart .....	69
Import .....	69
InsertSQL .....	69
DecimalPoint .....	69
Import .....	70
Inputcontrol .....	70
FormVars .....	70
FormChange .....	70
FieldRequired .....	71
SaveRequired .....	71
MailValid .....	71
MailValid_FieldRequired .....	71
IBANValid .....	71

IBANValid_FieldRequired .....	71
Wartung .....	72
TableRestart .....	72
ViewsErstellen .....	72
XRechnung .....	72
SaveXRechnung .....	72
Round2Decimalplaces .....	73
StartMail .....	73

## Einführung

---

Ab Januar 2025 ist die Rechnungsstellung in elektronischer Form für Rechnungen von Firmen untereinander Pflicht. Im Kontakt mit Behörden ist eine solche Rechnungsstellung schon seit längerer Zeit Voraussetzung<sup>1</sup>.

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.

Diese Dokumentation sowie die entsprechende Datenbank mit Vorlagendatei behandelt nur die XRechnung. Es geht um die Erstellung einer Rechnung (Kundeneingabe, Rechnungseingabe), die Abspeicherung der Rechnung und den Versand der PDF- und XML-Rechnung.

Das Einlesen einer XRechnung ist in zwei zugelassenen Formaten möglich. Dazu muss bei ZUGFeRD-Rechnungen die enthaltene XML-Datei separat abgespeichert werden.

Zum Einlesen werden aber 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.

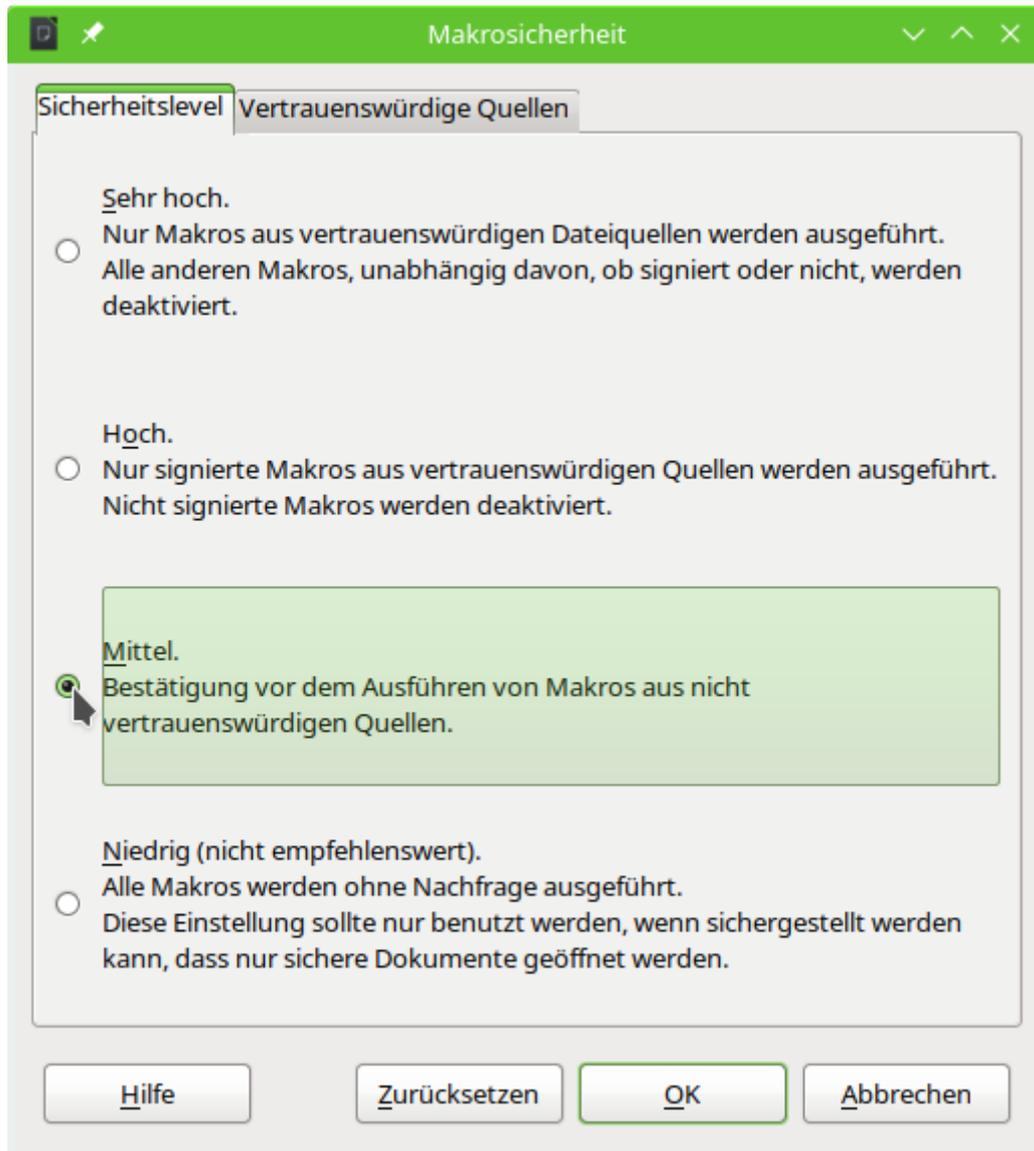
---

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

## Vorbereitende Schritte

Die Datei «Xrechnung.odt» sowie die Datei «Vorlage\_Rechnung.odt» werden in ein Verzeichnis kopiert. In diesem Verzeichnis wird beim Druck 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.

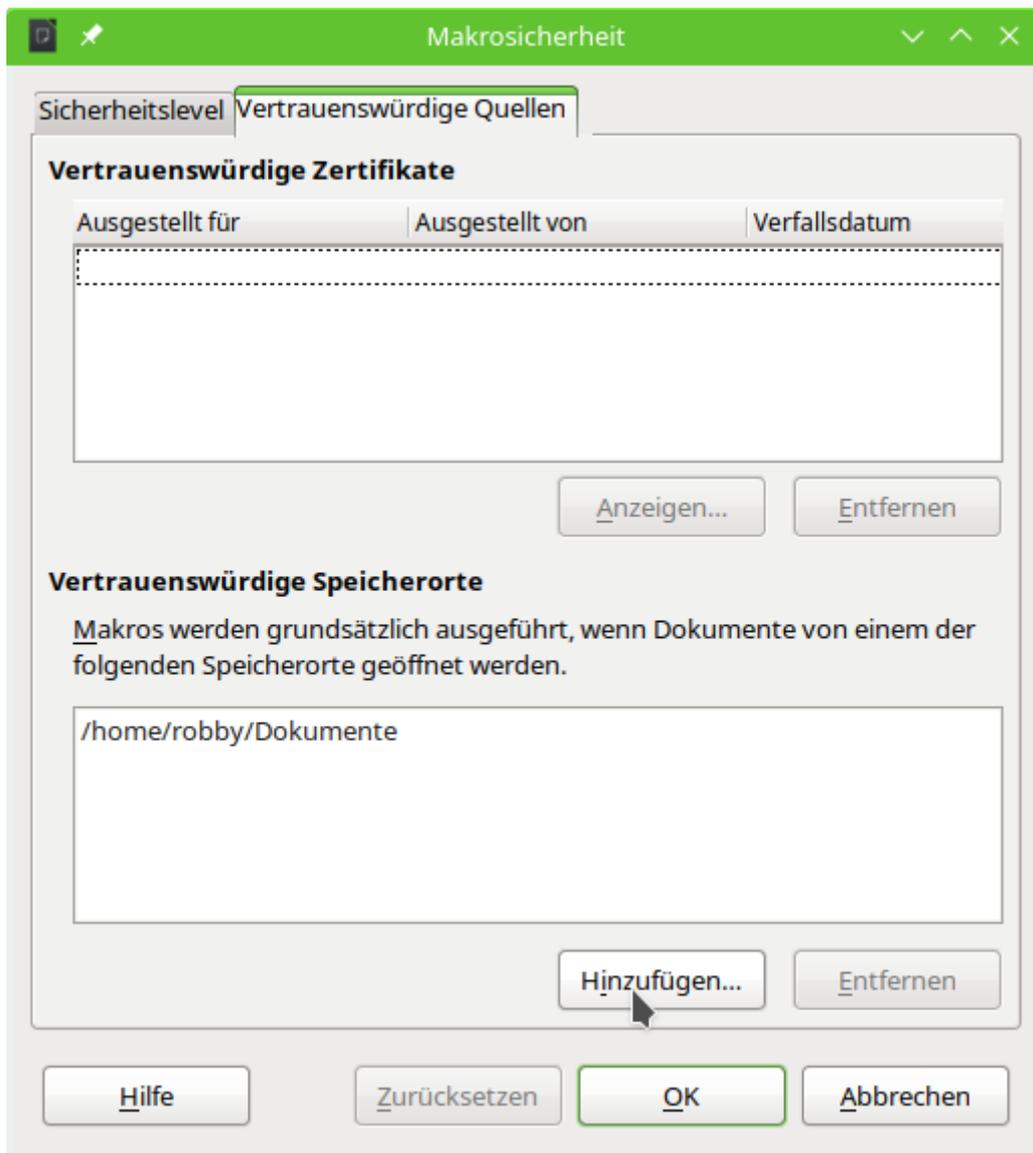
Sämtliche Formulare und auch die Vorlagendatei sind mit der Schriftart «DejaVu Sans» erstellt worden. Sollte diese frei verfügbare Schriftart nicht vorhanden sein, so muss gegebenenfalls die Vorlagendatei angepasst werden.



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



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 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 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. Im Anschluss daran sollte über **Extras → Makros → Makro ausführen** in der Base-Datei das Modul **Wartung → TableRestart** ausgeführt werden. Dann beginnen alle automatisch hoch zählenden Felder wieder mit der niedrigst möglichen Zahl. Bei leeren Tabellen ist dies auf '1' gesetzt.

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

---

## Formulare in der Übersicht

Die Datenbank enthält zur Zeit vier Formulare. Zur eigenen Rechnungserstellung wird das Formular "Kunde" 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 XRechnungs-Datei an eine E-Mail angehängt werden.

### Das Kunden-Formular

**Kunde**

Filter: Kunde

ID: 1

Firma\*: Dreaming    AnsprechP: Lisa Ichwillwissen    AnsprechPGeschlecht: weiblich

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

E-Mail\*: hierhin@beispiel.de    Telefon:    UST-IdNr:

Datensatz 1 von 1

---

**Rechnungserstellung**

ID: 1

Datum\*: 11.11.23    KundeZeichen\*: -    Mitarbeiter der Firma\*: Elly Workoholic

Zahlungsnotiz\_1:    Zahlungsnotiz\_2:

Datensatz 1 von 2    Rechnung erstellen

Zusatz*	Betrag*	Steuersatz*	Steuercode*	Sort	Netto	Steuer	Brutto
Portokosten	7,50 €	19 %	Normalsatz	6 - Gesamt	4.095,93 €	768,23 €	4.864,16 €
Normal	-25,00 €	19 %	Normalsatz				

Datensatz 1 von 2

Anzahl*	Einheit*	Bestellnummer	Ware*	Preis*	Rabatt	Steuersatz*	Steuercode*
2	Stück		Open-Source DVD	12,45 €	10 %	19 %	Normalsatz
3	Stück		Handbuch Base 7.6, gedruckt	21,90 €		19 %	Normalsatz
45	Stück		Stunden Arbeit am XRechnung-Formular	25,00 €		19 %	Normalsatz
3	Stück		Tafeln Schokolade mit Mandeln	1,75 €		7 %	Normalsatz
1	Stück		Dose Brustkaramellen für den	4,50 €		7 %	Normalsatz
1	Stück		Laufschuhe	142,00 €	5 %	19 %	Normalsatz

Datensatz 1 von 17

Startformular "Kunde" - hier werden neue Kunden eingegeben und Rechnungen für Kunden erstellt.

Über ein Listenfeld wird oben links von einem Formular zum anderen gewechselt.

Das danebenliegende Listenfeld im Kasten **Filter** dient dazu, bestimmte Kunden aus der Liste auszuwählen, die anschließend im darunterliegenden Formulareil 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 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.

Im Bereich über der zweiten Navigationsleiste befinden sich die Grunddaten für die Rechnung. Das Feld "ID" kann nicht bearbeitet werden (gelber Hintergrund). Die anderen 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 den zwei Tabellen mit weißem Hintergrund können die Rechnungsdaten eingegeben werden. Dies sind Eingaben auf Rechnungsebene (wie in dem Beispiel die Portokosten) und Eingaben der Rechnungsposten. Die Aufführung von Rechnungsposten ist bei der XRechnung Pflicht.

Bis auf die Felder "Bestellnummer" und "Rabatt" in dem unteren Tabellenfeld 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 Eingabevorschlügen 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.

Das Eingeben der Anhänge ist in dem vorherigen Screenshot nicht angezeigt. Hier eine separate Abbildung des Elements, das sich sowohl im Kundenformular als auch im Lieferantenformular befindet.

The screenshot shows a form titled "Anhänge" (Attachments). It includes a dropdown menu for "mimeCode\*" set to "image/png". Below this is a large area labeled "Anhang" which displays a preview of a file named "logo-sc.png". The preview shows the LibreOffice Community logo. To the left of the preview is a button labeled "Anhang anzeigen". Below the preview, there are input fields for "Dateiname" (filename) containing "logo-sc.png" and "Beschreibung" (description) containing "Nur das Logo von LibreOffice". At the bottom of the form, there is a control bar showing "Datensatz 1 von 2" and several navigation icons.

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

Manche Anhänge wie hier das Logo von LibreOffice 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 vorgesehenen Programm geschickt und damit geöffnet.

## Das Lieferanten-Formular

**Lieferant**

Filter  
Lieferant

ID

Firma\*  Kontaktperson  
Name\*  Geschlecht  E-Mail\*  Telefon\*

StraßeNr\*  Plz\*  Ort\*  Land\*

E-Mail\*  Telefon  IBAN\*  Zahlungsempfänger

Datensatz 10 von 10

**Lieferungsaufnahme**

ID

Datum\*  LieferantZeichen\*  UnserZeichen  Zahlungsfrist

Zahlungsnotiz

Datensatz 1 von 1

Zusatz*	Betrag*	Steuersatz*
Portokosten	7,50 €	19 %
Rabatt	-25,00 €	19 %

Datensatz 1 von 2

Sort	Netto	Steuer	Brutto
4 - Gesamt	1.340,26 €	253,48 €	1.593,74 €

XRech-ZeilenID	Anzahl*	Einheit	Ware*	Preis*	Rabatt	Steuersatz*
B-24-X-18	2	Stück	Open-Source DVD	12,45 €	2,49 €	19 %
HB 7.6	3	Stück	Handbuch Base 7.6, gedruckt	21,90 €	0,00 €	19 %
Schaffe-Schaffe	45	Stück	Stunden Arbeit am XRechnung-Formular	25,00 €	0,00 €	19 %
E-Schoggi	3	Stück	Tafeln Schokolade mit Mandeln	1,75 €	0,00 €	7 %
E-BonBon	1	Stück	Dose Brustkaramellen für den schmutzigen	4,50 €	0,00 €	7 %
S 42,5	1	Stück	Laufschuhe	142,00 €	7,10 €	19 %

Datensatz 1 von 6

Das Lieferanten-Formular bietet neben den Feldern aus dem Kundenformular u.a. eine Möglichkeit zum Einlesen von XRechnung-Dateien

Der Aufbau des Lieferanten-Formulars entspricht dem Prinzip des Kunden-Formulars. 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](#).

## Das Einstellungs-Formular

**Einstellungen**

Firma\*

StraßeNr\*  Plz\*  Ort\*

E-Mail\*  Website  Telefon\*

Firma\_Registriert\*  USt-IdNr

Bank\*  IBAN\*  BIC\*

RegisterE1  RegisterE2  RegisterE3

Zahlungsnotiz  Logo

ZahlungTage  Zahlungsempfänger

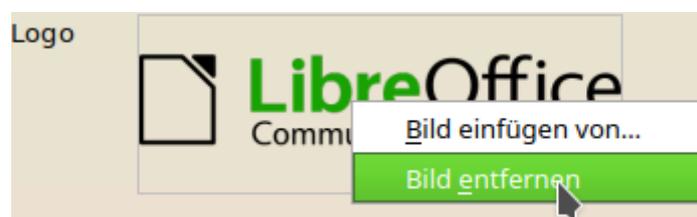
**MitarbeiterInnen**

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

Datensatz 1 von 2

Rechnung Stellenanzahl

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  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.

## Das Konto-Formular

**Konto**

**Offene Zahlungen von Lieferungen**

Lieferant	Datum	LieferantZeichen	NochOffen	ZahlDatum
▶ Mustermann GmbH → E-Mail: mail@muster.de	27.11.20	18383	233,00 €	
[Seller name] → E-Mail: rechnungsausgang@test	20.08.19	1234567	2.576,41 €	
[Seller name] → E-Mail: kundencenter@sellder.d	28.02.19	123456789	-225,14 €	
[Seller name] → E-Mail: kundencenter@sellder.d	05.06.18	123456789	53.171,59 €	

Datensatz 1 von 7

**Offene Eingänge von Rechnungen**

Kunde	Datum	Rechnungsnummer	Brutto	ZahlDatum
▶ Dreaming → E-Mail: hierhin@beispiel.de	08.12.23	R-2023-00000002	1.593,74 €	

Datensatz 1 von 1

**Kontostand**

Titel	Firma	Datum	Lief-Zeichen/R-Nummer	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 - XRechnung	11.11.23	R-2023-000000001	4.864,16 €	20.11.23	4.864,16 €	-19.008,73 €

Datensatz 1 von 4

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 lediglich Eingaben im "ZahlDatum" zu.

## 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 Kundenformular und im Lieferantenformular ist ein Filter für die Auswahl der Kunden bzw. Lieferanten eingebaut:

Datensatz 1 von 7 

Navigationleiste 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.

## Neueingabe von Daten

The screenshot shows a web-based form for entering customer data. At the top, there's a 'Filter' dropdown set to 'Kunde'. The main form has an 'ID' field with '<AutoFeld>' and several required fields marked with an asterisk: 'Firma\*', 'StraÙeNr\*', 'E-Mail\*', 'Plz\*', and 'Ort\*'. 'Land\*' is set to 'Deutschland'. Below the form is a navigation bar for 'Datensatz 2 von 2'. Underneath is a sub-form 'Rechnungserstellung' with fields for 'ID', 'Datum\*', 'KundeZeichen\*', and 'Mitarbeiter der Firma\*'. Its navigation bar shows 'Datensatz 0 von 0' and a 'Rechnung erstellen' button.

Im Formulardokument für die Kundeneingabe ist ein neuer Datensatz ausgewählt worden, weil ein neuer Kunde aufgenommen werden soll. In der Navigationsleiste steht «Datensatz 2 von 2». 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 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.

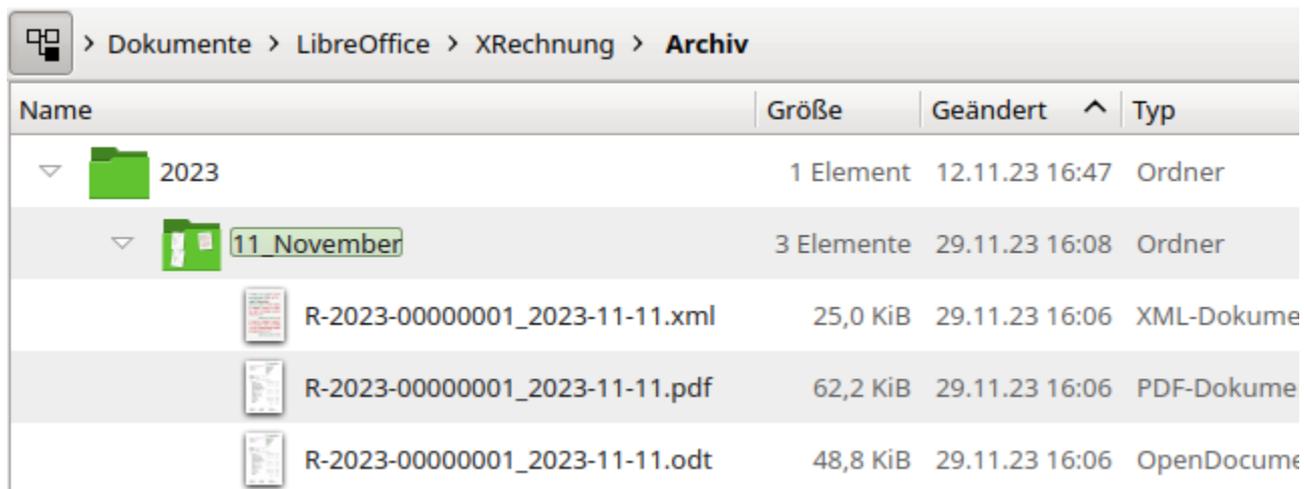
### ✓ Hinweis

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** ... .

## 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.

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.

### 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 Mailprogramm 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 Vorlagendatei, 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 - Open Source Allee 42 - 05040 Überall

Dreaming  
Hamburger Damm 12a  
90919 Irgendwo

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

MitarbeiterIn:  
Ely Workoholic  
work@beispiel.de  
013457902

**Rechnung**

Rechnung Nummer: R-2023-00000001      Datum: 11.11.2023  
Kundennummer: K-0000001

STEUERNUMMER      ZAHLUNGSART  
42-42-42      Zahlbar innerhalb von 14 Tagen ab Rechnungsdatum

MENGE	BESCHREIBUNG	EINZELPREIS	RABATT	STEUER	SUMME
2 Stück	Open-Source DVD (LibreOffice, Gimp, Scribus usw.)	12,45 €	10%	19%	22,41 €
3 Stück	Handbuch Base 7.6, gedruckt (alles über Base auf über 700 Seiten)	21,90 €		19%	65,70 €
45 Stück	Stunden Arbeit am XRechnung-Formular Erstellung eines einfachen Formulars, anschließend übertragen des Formulars nach Base hin (besser bedienbar), anschließend auch Import von XRechnungen nach Base	25,00 €		19%	1.125,00 €
3 Stück	Tafeln Schokolade mit Mandeln	1,75 €		7%	5,25 €
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 60 l, 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 Stück	A4-Papier Weißegrad 90% blauer Engel	7,15 €		19%	14,30 €
<b>Übertrag:</b>					<b>1.754,86 €</b>

FIRMA (REG. NAME)  
LibreOffice Testcenter  
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.

Seite 2 von 2		Base - XRechnung · Open Source Allee 42 · 05040 Überall		
		Übertrag:		1.754,86 €
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 €
1	Stück	Ratgeber digitale Bildverwaltung Von der Fotoerstellung bis zur Formatierung der SSD	15,20 €	7% 15,20 €
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 €
		Netto:	4.113,43 €	
		zugl. Portokosten:	7,50 €	
		abzgl. Rabatt:	25,00 €	
		MWSt 7%:	5,83 €	
		MWSt 19%:	762,40 €	
		Gesamt:	4.864,16 €	

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.

# Technische Grundlagen

## Erforderliche Einträge in der XRechnung

Die folgende Tabelle listet alle Felder auf, die für eine XRechnung<sup>3</sup> 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.

<b>XRechnung</b>	<b>XML-Version UBL</b>	<b>XML-Version CII</b>	<b>Datenbank</b>
INVOICE	Invoice	CrossIndustryInvoice	
Invoice number	ID	ExchangedDocument → ID	"viw_Rechnung"."Rechnungsnummer"
Invoice issue date	IssueDate	ExchangedDocument → IssueDateTime	"tbl_Rechnung"."Datum"
Invoice type code	InvoiceTypeCode	ExchangedDocument → TypeCode	Macro Xrechnung → SaveXrechnung ('380': Handelsrechnung)
Invoice currency code	DocumentCurrencyCode	ApplicableHeaderTradeSettlement → InvoiceCurrencyCode	Macro Xrechnung → SaveXrechnung ('EUR': Euro)
Buyer reference	BuyerReference	BuyerReference	"tbl_Rechnung"."KundeZeichen"
PROCESS CONTROL	<i>Direkt auf Invoice-Ebene</i>	ExchangedDocument Context	
Business process type	ProfileID	BusinessProcessSpecifiedDocumentContextParameter → ID	Macro Xrechnung → SaveXrechnung 'urn:fdc:peppol.eu:2017:poacc:billing:01:1.0'
Specification identifier	CustomizationID	GuidelineSpecifiedDocumentContextParameter → ID	Macro Xrechnung → SaveXrechnung 'urn:cen.eu:en16931:2017#compliant#urn:xeinkauf.de:kosit:xrechnung_3.0'
SELLER	AccountingSupplierParty	SellerTradeParty	
Seller name	PartyLegalEntity → RegistrationName	Name	"tbl_Firma"."Firma_Registriert"
Seller electronic address	EndpointID schemeID="EM"	URIUniversalCommunication → URIID schemeID="EM"	"tbl_Firma"."E-Mail"

<sup>3</sup> Siehe <https://xeinkauf.de/dokumente/> → dort nach XRechnung suchen...

<b>XRechnung</b>	<b>XML-Version UBL</b>	<b>XML-Version CII</b>	<b>Datenbank</b>
SELLER POSTAL ADDRESS	PostalAddress	PostalTradeAddress	
Seller city	CityName	CityName	"tbl_Firma"."Ort"
Seller postcode	PostalZone	PostcodeCode	"tbl_Firma"."Plz"
Seller country code	IdentificationCode	CountryID	Macro Xrechnung → SaveXrechnung ('DE': Deutschland)
SELLER CONTACT	Contact	DefinedTradeContact	
Seller contact point	Name	PersonName	"tbl_Mitarbeiter"."Name"
Seller contact telephone number	Telephone	TelephoneUniversalCommunication → CompleteNumber	"tbl_Mitarbeiter"."Telefon"
Seller contact email address	ElectronicMail	EmailURIUniversalCommunication → URIID	"tbl_Mitarbeiter"."E-Mail"
BUYER	AccountingCustomerParty	BuyerTradeParty	
Buyer name	PartyLegalEntity → RegistrationName	Name	"tbl_Kunde"."Firma"
Buyer electronic address	EndpointID schemeID="EM"	URIUniversalCommunication → URIID schemeID="EM"	"tbl_Kunde"."E-Mail"
BUYER POSTAL ADDRESS	PostalAddress	PostalTradeAddress	
Buyer city	CityName	CityName	"tbl_Kunde"."Ort"
Buyer postcode	PostalZone	PostcodeCode	"tbl_Kunde"."Plz"
Buyer country code	IdentificationCode	CountryID	"tbl_Kunde"."Land"
PAYMENT INSTRUCTIONS <sup>4</sup>	PaymentMeans	SpecifiedTradeSettlementPaymentMeans	
Payment means type code	PaymentMeansCode	TypeCode	Macro Xrechnung → SaveXrechnung ('58': nur IBAN (innerhalb des SEPA-Raumes))
DOCUMENT TOTALS	LegalMonetaryTotal	SpecifiedTradeSettlementHeaderMonetarySummation	
Sum of Invoice line net amount	LineExtensionAmount	LineTotalAmount	"viw_Rechnung_Summe"."Netto" (Summe Rechnungszeilen)
Invoice total amount without VAT	TaxExclusiveAmount	TaxBasisTotalAmount	"viw_Rechnung_Summe"."Netto" (Gesamt)

4 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.

<b>XRechnung</b>	<b>XML-Version UBL</b>	<b>XML-Version CII</b>	<b>Datenbank</b>
Invoice total amount with VAT	TaxInclusiveAmount	GrandTotalAmount	"viw_Rechnung_Summe"."Brutto" (Gesamt)
Amount due for payment	PayableAmount	DuePayableAmount	"viw_Rechnung_Summe"."Brutto" (Gesamt)
VAT <sup>5</sup> BREAKDOWN (1...n)	TaxTotal	ApplicableTradeTax	"viw_Rechnung_Summe"."Steuer" (Gesamt)
VAT category taxable amount	TaxSubtotal → TaxableAmount	BasisAmount	"viw_Rechnung_Summe"."Netto" (Summe Steuersatz)
VAT category tax amount	TaxSubtotal → TaxAmount	CalculatedAmount	"viw_Rechnung_Summe"."Steuer" (Summe Steuersatz)
VAT category code	TaxSubtotal → TaxCategory → ID	CategoryCode	"viw_Rechnung_Summe"."Steuer_Code_ID"
VAT category rate	TaxSubtotal → TaxCategory → Percent	RateApplicablePercent	"viw_Rechnung_Summe"."Steuersatz"
INVOICE LINE (1...n)	InvoiceLine	IncludedSupplyChainTradeLineItem	
Invoice line identifier	ID	AssociatedDocumentLineDocument → LineID	"viw_Rechnung_Ausgang"."RechnungAusgang_ID"
Invoiced quantity	InvoicedQuantity	BilledQuantity	"viw_Rechnung_Ausgang"."Anzahl"
Invoiced quantity unit of measure code	unitCode="XPP"	unitCode="XPP"	"viw_Rechnung_Ausgang"."Anzahl_Code_ID" (Auch die deutschsprachige Bezeichnung in "Name_DE")
Invoice line net amount	LineExtensionAmount	SpecifiedTradeSettlementLineMonetarySummation → LineTotalAmount	"viw_Rechnung_Ausgang"."AnzahlPreis"
PRICE DETAILS	Price	NetPriceProductTradePrice	
Item net price	PriceAmount	ChargeAmount	"viw_Rechnung_Ausgang"."Preis"
LINE VAT INFORMATION	ClassifiedTaxCategory	ApplicableTradeTax	
Invoiced item VAT category code <sup>6</sup>	ID	CategoryCode	"viw_Rechnung_Ausgang"."Steuer_Code_ID"

5 VAT: Value Added Tax → Mehrwertsteuer

<b>XRechnung</b>	<b>XML-Version UBL</b>	<b>XML-Version CII</b>	<b>Datenbank</b>
ITEM INFORMATION	Item	SpecifiedTradeProduct	
Item name	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.

- 
- 6 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 keine Zuweisung zum normalen Mehrwertsteuersatz oder zum ermäßigten Steuersatz ermitteln.

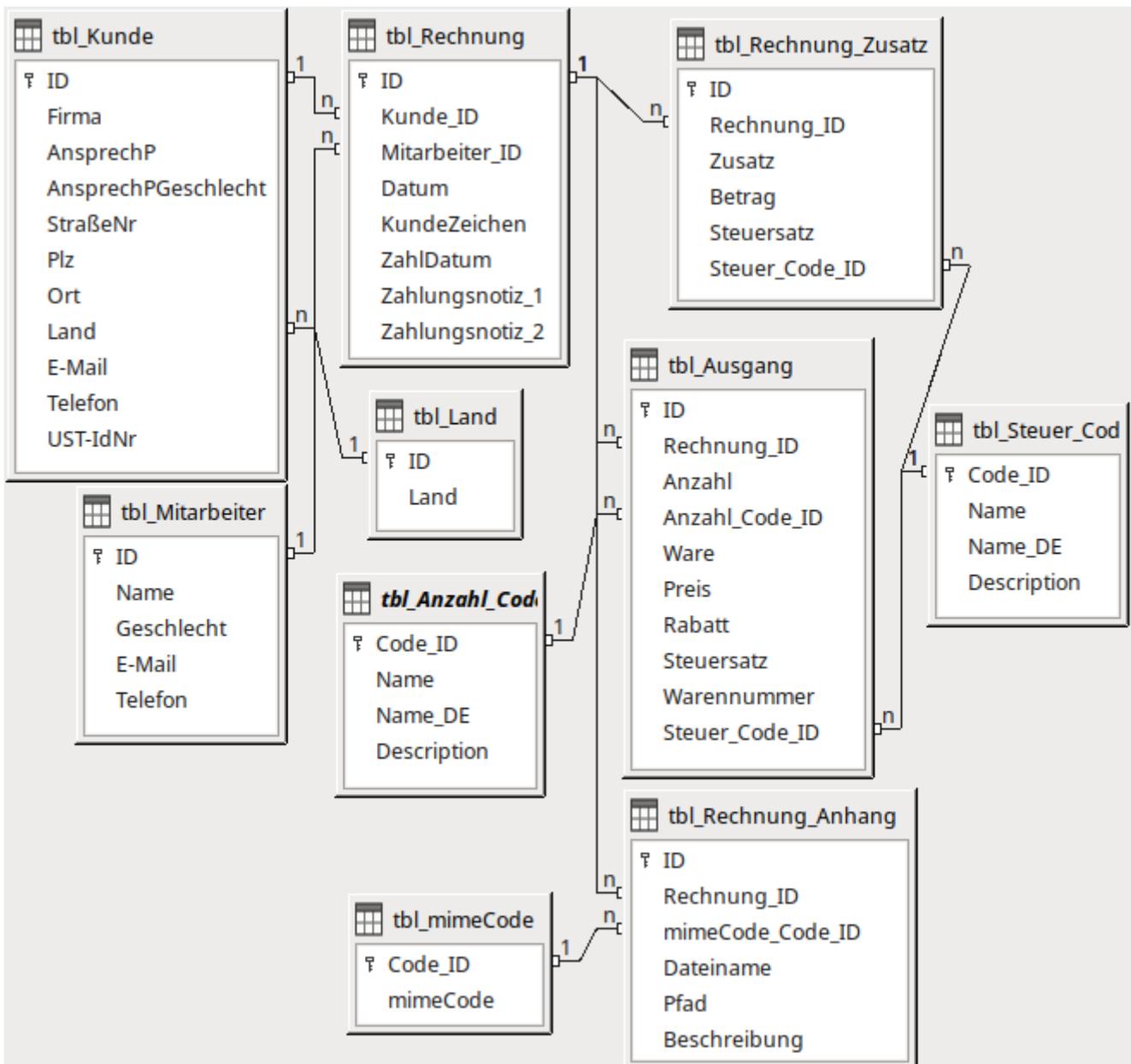
## Tabellen

---

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 Kasselführung 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

<b>Datenziel</b>
Ansicht: <a href="#">viw_Rechnung</a>
Formular: <a href="#">frm_Kunde</a>

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
AnsprechP	Text	Der Name der Ansprechperson in der Firma Eingabe erforderlich: Nein

<b>Feldname</b>	<b>Feldtyp</b>	<b>Beschreibung</b>
AnsprechPGe- schlecht	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)
Plz	Text	Postleitzahl für Waren an die Firma/Person Eingabe erforderlich: Ja
Ort	Text	Lieferort für Waren an die Firma/Person Eingabe erforderlich: Ja
Land	Text	Landesangabe für Waren an die Firma/Person – 2 Buchstaben Fremdschlüssel aus Tabelle "tbl_Land"."ID" Eingabe erforderlich: Ja
E-Mail	Text	E-Mail-Adresse der Firma zur Zusendung der XRechnung Eingabe erforderlich: Ja
Telefon	Text	Telefonkontakt zu der Firma Eingabe erforderlich: Nein

### **tbl\_Land**

<b>Datenziel</b>
Ansicht: <a href="#">viw_Rechnung</a>

<b>Feldname</b>	<b>Feldtyp</b>	<b>Beschreibung</b>
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

### **tbl\_Rechnung**

<b>Datenziel</b>
Ansicht: <a href="#">viw_Rechnung</a>
Abfrage: <a href="#">qry_Rechnung_Zahlung</a>
Formular: <a href="#">frm_Kunde</a>
Makro: <a href="#">AnhangPfad</a>

<b>Feldname</b>	<b>Feldtyp</b>	<b>Beschreibung</b>
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

<b>Feldname</b>	<b>Feldtyp</b>	<b>Beschreibung</b>
Datum	Datum	Rechnungsdatum Eingabe erforderlich: Ja
KundeZeichen	Text	Zeichen des Kunden (Bestellung?) - bei Behörden die Leitweg-ID; kann auf «-» gesetzt werden, falls kein Zeichen vorhanden ist Eingabe erforderlich: Ja
ZahlDatum	Datum	Datum für den Tag, an dem die Zahlung der Rechnung erfolgt ist Eingabe erforderlich: Nein
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
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

### **tbl\_Mitarbeiter**

<b>Datenziel</b>
Ansicht: <a href="#">viw_Rechnung</a>
Formular: <a href="#">frm_Einstellungen</a>

<b>Feldname</b>	<b>Feldtyp</b>	<b>Beschreibung</b>
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
Geschlecht	Text	Geschlecht der Person Eingabe erforderlich: Nein
E-Mail	Text	E-Mail-Kontakt zu der Person Eingabe erforderlich: Ja
Telefon	Text	Telefonkontakt zu der Person Eingabe erforderlich: Ja

### **tbl\_Rechnung\_Zusatz**

<b>Datenziel</b>
Ansicht: <a href="#">viw_Rechnung_Steuer</a>
Formular: <a href="#">frm_Kunde</a>
Makro: <a href="#">FillTableCarryOver</a> , <a href="#">SaveXRechnung</a>

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

<b>Feldname</b>	<b>Feldtyp</b>	<b>Beschreibung</b>
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
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 zugewiesen Fremdschlüssel aus "tbl_Steuer_Code"."Code_ID" Eingabe erforderlich: Ja Standard auf 'S' (normaler Steuersatz) gesetzt.

### **tbl\_Rechnung\_Anhang**

<b>Datenziel</b>
Ansicht:
Formular: <i>frm_Kunde</i>
Makro: <i>SaveXRechnung</i>

<b>Feldname</b>	<b>Feldtyp</b>	<b>Beschreibung</b>
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
Dateiname	Text	Dateiname der eingelesenen Datei – wird automatisch im Formular eingetragen Eingabe erforderlich: Nein
Pfad	Text	Pfad zu der eingelesenen Datei im Binärformat. Die Datei wird in einem grafischen Kontrollfeld angezeigt. Eingabe erforderlich: Nein
Beschreibung	Text	Kurze Beschreibung des Anhangs Eingabe erforderlich: Nein

### **tbl\_Ausgang**

<b>Datenziel</b>
Ansicht: <i>viw_Rechnung_Ausgang, viw_Rechnung_Steuer</i>
Formular: <i>frm_Kunde</i>

<b>Feldname</b>	<b>Feldtyp</b>	<b>Beschreibung</b>
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
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: Ja
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 Nachkommastellen möglich, (bei Energierechnungen üblich) 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
Warennummer	Text	ID für die Ware, Bestellnummer... Eingabe erforderlich: Nein
Steuer_Code_ID	Text (3)	Hiermit wird der Steuercode aus XRechnung der Rechnung zugewiesen Fremdschlüssel aus "tbl_Steuer_Code"."Code_ID" Eingabe erforderlich: Ja Standard auf 'S' (normaler Steuersatz) gesetzt.

Beim Ausgang wird bisher davon ausgegangen, dass es nur um ganze Stückzahlen geht. XRechnung lässt hier aber viele Maßeinheiten und eine Zahl mit beliebig viel Nachkommastellen zu. Gegebenenfalls müsste diese Tabelle um einen Eintrag mit Maßeinheiten erweitert werden.

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

### **tbl\_Anzahl\_Code**

<b>Datenziel</b>
Ansicht: <a href="#">viw_Rechnung_Ausgang</a>
Formular: <a href="#">frm_Kunde</a>

<b>Feldname</b>	<b>Feldtyp</b>	<b>Beschreibung</b>
Code_ID	Text (3 Zeichen)	Primärschlüssel der Tabelle. Der Primärschlüssel muss eindeutig sein. Das Feld ist durch den Code für die Maßeinheit der Anzahl <sup>7</sup> bei XRechnungen vorgegeben.

<sup>7</sup> [https://www.xrepository.de/details/urn:xoev-de:kosit:codeliste:rec21\\_3](https://www.xrepository.de/details/urn:xoev-de:kosit:codeliste:rec21_3)

<b>Feldname</b>	<b>Feldtyp</b>	<b>Beschreibung</b>
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

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

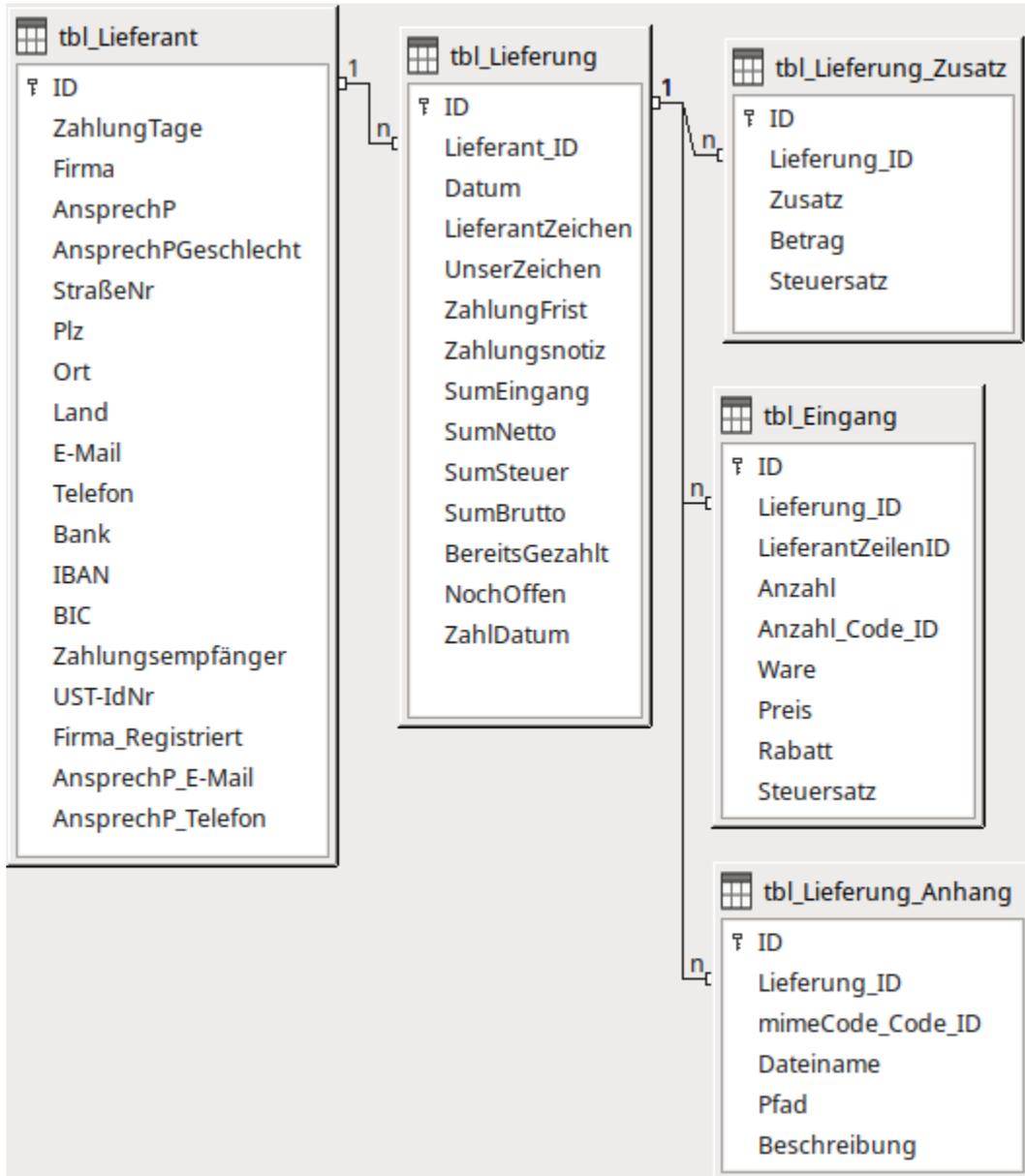
Den gleichen Aufbau haben die Tabellen "tbl\_Rechnungstyp\_Code", "tbl\_Steuer\_Code" und "tbl\_Zusatz\_Code" (Ermäßigungen und Zuschläge).

### **tbl\_mimeCode**

<b>Datenziel</b>
Formular: <i>frm_Kunde, frm_Lieferant</i>
Makro: <i>SaveXRechnung</i>

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

## Eingang von Waren über Lieferanten



Hier ist nur der Block für die Lieferanten abgebildet. Die Beziehung von anderen Tabellen zur Tabelle "tbl\_Eingang"."Anzahl\_Code\_ID" oder "tbl\_Lieferung\_Anhang"."mime-Code\_Code\_ID" ist zwar definiert, aber hier nicht dargestellt.

### tbl\_Lieferant

<b>Datenziel</b>
Formular: <i>frm_Lieferant</i>
Abfrage: <i>qry_Konto, qry_Lieferung_Zahlung</i>
Makro: <i>Import</i>

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

<b>Feldname</b>	<b>Feldtyp</b>	<b>Beschreibung</b>
Firma	Text	Der Name der Firma/Person Eingabe erforderlich: Ja
AnsprechP	Text	Der Name der Ansprechperson in der Firma Eingabe erforderlich: Ja
AnsprechPGe- schlecht	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
AnsprechP_Tel efon	Text	Telefonnummer der Ansprechperson Eingabe erforderlich: Ja
StraßeNr	Text	Straße und Hausnummer der Firma/Person Eingabe erforderlich: Ja (obwohl für XRechnung nicht erfor- derlich)
Plz	Text	Postleitzahl für Waren von der Firma/Person Eingabe erforderlich: Ja
Ort	Text	Lieferort für Waren von der Firma/Person Eingabe erforderlich: Ja
Land	Text	Landesangabe für Waren von der Firma/Person – 2 Buchsta- ben Fremdschlüssel aus Tabelle "tbl_Land" Eingabe erforderlich: Ja
E-Mail	Text	E-Mail-Adresse der Firma zur Zusendung der XRechnung Eingabe erforderlich: Ja
Telefon	Text	Telefonkontakt zu der Firma Eingabe erforderlich: Ja
Bank	Text	Name der Bank Eingabe erforderlich: Nein
IBAN	Text	IBAN Eingabe erforderlich: Ja
BIC	Text	BIC Eingabe erforderlich: Nein
ZahlungTage	Integer	Zahlungsfrist in Tagen nach Rechnungsstellung Eingabe erforderlich: Nein
Zahlungs- empfänger	Text	Empfänger der Zahlung bei der Bank Eingabe erforderlich: Nein
UST-IdNr	Text	Umsatzsteuernummer Eingabe erforderlich: Ja
Firma-Regis- triert	Text	Registrierter Name der Firma – kann anders als der Rufname sein Eingabe erforderlich: Ja

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

<b>Datenziel</b>
Formular: <i>frm_Lieferant</i>
Abfrage: <i>qry_Konto, qry_Lieferung_Zahlung</i>
Makro: <i>Import</i>

<b>Feldname</b>	<b>Feldtyp</b>	<b>Beschreibung</b>
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
ZahlDatum	Datum	Datum für den Tag, an dem die Überweisung von der eigenen Firma für die Lieferung erfolgt ist 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\_Lieferung\_Zusatz**

<b>Datenziel</b>
Ansicht: <i>viw_Lieferung_Steuer</i>
Formular: <i>frm_Lieferant</i>
Makro: <i>Import</i>

<b>Feldname</b>	<b>Feldtyp</b>	<b>Beschreibung</b>
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
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

## **tbl\_Lieferung\_Anhang**

<b>Datenziel</b>
Ansicht:
Formular: <i>frm_Lieferant</i>
Makro: <i>SaveXRechnung</i>

<b>Feldname</b>	<b>Feldtyp</b>	<b>Beschreibung</b>
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
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\_Eingang

<b>Datenziel</b>
Ansicht: <i>viw_Lieferung_Steuer</i>
Formular: <i>frm_Lieferant</i>
Makro: <i>Import</i>

<b>Feldname</b>	<b>Feldtyp</b>	<b>Beschreibung</b>
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

### ✓ 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.

## 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

<b>Datenziel</b>
Ansicht: <i>viw_Rechnung</i>
Formular: <i>frm_Einstellungen</i>

<b>Feldname</b>	<b>Feldtyp</b>	<b>Beschreibung</b>
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
StraßeNr	Text	Straße und Hausnummer der Firma Eingabe erforderlich: Ja (obwohl für XRechnung nicht erforderlich)
Plz	Text	Postleitzahl für die eigene Firma Eingabe erforderlich: Ja
Ort	Text	Ortsangabe für die eigene Firma Eingabe erforderlich: Ja
E-Mail	Text	E-Mail-Adresse der eigenen Firma Eingabe erforderlich: Ja
Website	Text	Website der eigenen Firma Eingabe erforderlich: Nein
Telefon	Text	Telefonkontakt zur eigenen Firma Eingabe erforderlich: Ja
UST-IdNr	Text	Umsatzsteuernummer Eingabe erforderlich: Ja
Firma-Registriert	Text	Registrierter Name der Firma - kann anders als der Rufname sein. Muss für die XRechnung ausgefüllt sein. Eingabe erforderlich: Ja
Bank	Text	Name der Bank Eingabe erforderlich: Ja (für die Information in der Fußzeile der ausdrückbaren Rechnung)
IBAN	Text	IBAN Eingabe erforderlich: Ja
BIC	Text	BIC Eingabe erforderlich: Ja (für die Information in der Fußzeile der ausdrückbaren Rechnung)
RegisterE1	Text	Amtsregistereintrag 1, der auf der Rechnung erscheinen soll Eingabe erforderlich: Nein
RegisterE2	Text	Amtsregistereintrag 2, der auf der Rechnung erscheinen soll Eingabe erforderlich: Nein
RegisterE3	Text	Amtsregistereintrag 3, der auf der Rechnung erscheinen soll Eingabe erforderlich: Nein
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

<b>Feldname</b>	<b>Feldtyp</b>	<b>Beschreibung</b>
ZahlungTage	Integer	Zahlungsfrist in Tagen nach Rechnungsstellung Eingabe erforderlich: Nein
Zahlungs-empfänger	Text	Empfänger der Zahlung bei der Bank Eingabe erforderlich: Nein
ReStellen	SmallInteger	Anzahl der Stellen in der Rechnungsnummer Eingabe erforderlich: Nein

## **tbl\_Filter**

<b>Datenziel</b>
Ansicht: <i>viw_Filter</i>
Formular: <i>frm_Einstellungen, frm_Kunde, frm_Lieferant</i>
Makro: <i>FilterStart</i>

<b>Feldname</b>	<b>Feldtyp</b>	<b>Beschreibung</b>
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.

## **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
9	frm_Kunde		

#### Datenquelle

Tabelle: *tbl\_Filter*

#### Datenziel

Abfrage: *qry\_Forms*

Formular: *frm\_Einstellungen*

Makro: *FilterStart*

```
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.

### viw\_Lieferung\_Steuer

Lieferung_ID	Sort	Steuersatz	Netto	Steuer	Brutto
7	1 - Rechnungszeilen	0,07	83,32	5,8324	89,1524
7	1 - Rechnungszeilen	0,19	4030,11	765,7209	4795,8309
7	2 - Lieferung pauschal	0,19	-17,5	-3,325	-20,825

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\_Eingang, tbl\_Lieferung\_Zusatz*

#### Datenziel

Ansicht: *viw\_Lieferung\_Summe*

```
001 SELECT "Lieferung_ID", '1 - Rechnungszeilen' AS "Sort", "Steuersatz",
002     SUM( "AnzahlPreis" ) AS "Netto",
003     SUM( "AnzahlPreis" ) * "Steuersatz" AS "Steuer",
004     SUM( "AnzahlPreis" ) * ( 1 + "Steuersatz" ) AS "Brutto"
005 FROM
006     ( SELECT "Anzahl" * "Preis" - COALESCE ( "Rabatt", 0 ) AS "AnzahlPreis",
007         "Steuersatz", "Lieferung_ID"
008     FROM "tbl_Eingang" )
```

```

009 AS "a"
010 GROUP BY "Lieferung_ID", "Steuersatz"
011 UNION
012 SELECT "Lieferung_ID", '2 - Lieferung pauschal' AS "Sort", "Steuersatz",
013 SUM( "Betrag" ) AS "Netto",
014 SUM( "Betrag" ) * "Steuersatz" AS "Steuer",
015 SUM( "Betrag" ) * ( 1 + "Steuersatz" ) 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.

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\_Summe

	Lieferung_ID	Sort	Steuersatz	Netto	Steuer	Brutto
▶	7	1 - Rechnungszeilen	0,07	83,32	5,8324	89,1524
	7	1 - Rechnungszeilen	0,19	4030,11	765,7209	4795,8309
	7	2 - Lieferung pauschal	0,19	-17,5	-3,325	-20,825
	7	3 - Summe Steuersatz	0,07	83,32	5,8324	89,1524
	7	3 - Summe Steuersatz	0,19	4012,61	762,3959	4775,0059
	7	4 - Gesamt		4095,93	768,2283	4864,1583

### Datenquelle

Ansicht: [viw\\_Lieferung\\_Steuer](#)

## Datenziel

Formular: *frm\_Lieferant*

```
001 SELECT * FROM "viw_Lieferung_Steuer"
002 UNION
003 SELECT "Lieferung_ID", '3 - Summe Steuersatz' AS "Sort", "Steuersatz",
004     SUM( "Netto" ) AS "Netto",
005     SUM( "Steuer" ) AS "Steuer",
006     SUM( "Brutto" ) AS "Brutto"
007 FROM "viw_Lieferung_Steuer"
008 GROUP BY "Lieferung_ID", "Steuersatz"
009 UNION
010 SELECT "Lieferung_ID", '4 - Gesamt' AS "Sort", NULL AS "Steuersatz",
011     SUM( "Netto" ) AS "Netto",
012     SUM( "Steuer" ) AS "Steuer",
013     SUM( "Brutto" ) AS "Brutto"
014 FROM "viw_Lieferung_Steuer"
015 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\_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 die letzten Felder mit u.a. dem berechneten Datum "ZahlungBis" und der zusammengestellten "Rechnungsnummer".

## Datenquelle

Tabelle: *tbl\_Kunde, tbl\_Rechnung, tbl\_Mitarbeiter, tbl\_Land, tbl\_Firma*

Ansicht: *viw\_Rechnung\_Summe*

## Datenziel

Formular: *frm\_Kunde*

Abfrage: *qry\_Konto, qry\_Rechnung\_Zahlung*

Makro: *FillTableCarryOver, SaveXRechnung*

```
001 SELECT "tbl_Firma"."Firma", "tbl_Firma"."StraßeNr", "tbl_Firma"."Plz",
         "tbl_Firma"."Ort", "tbl_Firma"."E-Mail", "tbl_Firma"."Website",
         "tbl_Firma"."Telefon", "tbl_Firma"."UST-IdNr", "tbl_Firma"."Firma_Registriert",
         "tbl_Firma"."Bank", "tbl_Firma"."IBAN", "tbl_Firma"."BIC",
         "tbl_Firma"."RegisterE1", "tbl_Firma"."RegisterE2", "tbl_Firma"."RegisterE3",
002 COALESCE("a"."Zahlungsnotiz_1", "tbl_Firma"."Zahlungsnotiz")||
         COALESCE(ASCII_CHAR(13)||"a"."Zahlungsnotiz_2", '') AS "Zahlungsnotiz",
         "tbl_Firma"."ZahlungTage",
003 COALESCE("tbl_Firma"."Zahlungsempfänger", "tbl_Firma"."Firma_Registriert")
         AS "Zahlungsempfänger",
004 "tbl_Firma"."Logo", "tbl_Mitarbeiter"."Name", "tbl_Mitarbeiter"."Geschlecht",
005 "tbl_Mitarbeiter"."E-Mail" AS "E-Mail_Mitarbeiter",
006 "tbl_Mitarbeiter"."Telefon" AS "Telefon_Mitarbeiter",
```

```

007 'K-'||RIGHT('000000'||"tbl_Kunde"."ID",6) AS "Kundenummer",
008 "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",
009 UPPER("tbl_Land"."Land") AS "Land",
"tbl_Kunde"."E-Mail" AS "E-Mail_Kunde",
"tbl_Kunde"."Telefon" AS "Telefon_Kunde",
010 "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",
011 "a"."ID" AS "Rechnung_ID",
012 RIGHT('0'||EXTRACT(DAY FROM "a"."Datum"),2)||'.'||RIGHT('0'||EXTRACT(MONTH FROM
"a"."Datum"),2)||'.'||EXTRACT(YEAR FROM "a"."Datum") AS "RechnungsdatumD",
013 "a"."Datum" AS "Rechnungsdatum",
014 "a"."Datum" + "tbl_Firma"."ZahlungTage" AS "ZahlungBis",
015 "a"."KundeZeichen",
016 'R-'||"c"."Jahr"||'-'||RIGHT('0000000000'||"c"."lfdJahr",
COALESCE("tbl_Firma"."ReStellen",6)) AS "Rechnungsnummer",
017 "viw_Rechnung_Summe"."Netto", "viw_Rechnung_Summe"."Steuer",
"viw_Rechnung_Summe"."Brutto", "a"."ZahlDatum"
018 FROM "tbl_Kunde"
019 LEFT JOIN "tbl_Rechnung" AS "a" ON "a"."Kunde_ID" = "tbl_Kunde"."ID"
020 LEFT JOIN "tbl_Mitarbeiter" ON "a"."Mitarbeiter_ID" = "tbl_Mitarbeiter"."ID"
021 LEFT JOIN "tbl_Land" ON "tbl_Kunde"."Land" = "tbl_Land"."ID"
022 LEFT JOIN "viw_Rechnung_Summe" ON "a"."ID" = "viw_Rechnung_Summe"."Rechnung_ID"
023 LEFT JOIN
(SELECT "ID", EXTRACT(YEAR FROM "Datum") AS "Jahr", RANK() OVER
(PARTITION BY EXTRACT(YEAR FROM "Datum") ORDER BY "ID") AS "lfdJahr"
FROM "tbl_Rechnung")
AS "c" ON "a"."ID" = "c"."ID",
024 "tbl_Firma"
025 WHERE "tbl_Firma"."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: 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 5 und 6: 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 7 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 9 wird deswegen das Feld "tbl\_Land"."Land" mit **UPPER** in Großschreibweise umgewandelt.

In Zeile 10 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 11: 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 14 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 16 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 Nullen, 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 23).

Ab Zeile 18 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.

In Zeile 23 wird eine Funktion von Firebird genutzt, die eine laufende Nummer nach einer bestimmten Sortierung erstellt: **RANK() OVER (PARTITION BY EXTRACT(YEAR FROM "Datum") ORDER BY "ID")**. Abhängig von dem Jahr wird eine Zahlenreihenfolge, beginnen mit 1, erstellt. Die Sortierung für diese Reihenfolge erfolgt nach Feld "ID", das ja ein eindeutiges Feld ist. Die gesamte Abfrage bekommt anschließend den Alias "c" zugewiesen, um auch weiter oben ansprechbar zu sein. Dabei muss das Feld "ID" dieser Abfrage gleich sein dem Feld "ID" aus der Tabelle "tbl\_Rechnung", die hier mit dem Alias "a" angesprochen wird.

Die "tbl\_Firma" in Zeile 23 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 25 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

	Rec...	Rechn...	Anzahl	Anzahl_C...	Name_DE	Ware	Preis	Rabatt	AnzahlPreis	Steuersatz
▶	1	1	2	XPP	Stück	Open-Source DVD(LibreOffic	12,45 €	10 %	22,41 €	19 %
	1	2	3	XPP	Stück	Handbuch Base 7.6, gedrucl	21,90 €		65,70 €	19 %
	1	3	45	XPP	Stück	Stunden Arbeit am XRechnu	25,00 €		1.125,00 €	19 %
	1	4	3	XPP	Stück	Tafeln Schokolade mit Mande	1,75 €		5,25 €	7 %
	1	5	1	XPP	Stück	Dose Brustkaramellen für de	4,50 €		4,50 €	7 %
	1	6	1	XPP	Stück	Laufschuhevor allem für Stra	142,00 €	5 %	134,90 €	19 %

Die Darstellung dieser Ansicht zeigt einmal die korrekt formatierten Felder für "Preis", "Rabatt", "AnzahlPreis" 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.

### Datenquelle

Tabelle: *tbl\_Ausgang, tbl\_Anzahl\_Code*

### Datenziel

Makro: *FillTableCarryOver, SaveXRechnung*

```
001 SELECT "Rechnung_ID",
002     COALESCE("Warennummer", RANK() OVER (PARTITION BY "Rechnung_ID" ORDER BY "ID"))
      AS "RechnungAusgang_ID",
003     "Anzahl", "Anzahl_Code_ID", "Name_DE", "Ware", "Preis", "Rabatt",
004     "Anzahl" * "Preis" * (1 - COALESCE("Rabatt",0)) AS "AnzahlPreis",
005     "Steuersatz", "Warennummer", "Steuer_Code_ID"
006 FROM "tbl_Ausgang", "tbl_Anzahl_Code"
007 WHERE "tbl_Ausgang"."Anzahl_Code_ID" = "tbl_Anzahl_Code"."Code_ID"
```

Bei dieser Ansicht wird lediglich die "tbl\_Ausgang" abgefragt und dabei mit zusätzlichen Informationen versehen.

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 "RechnungAusgang\_ID".

In Zeile 4 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.

## viw\_Rechnung\_Steuer

Rechnung_ID	Sort	Steuersatz	Steuer_Code_ID	Netto	Steuer	Brutto
1	1 - Rechnungszeilen	0,07	S	83,32	5,8324	89,1524
1	1 - Rechnungszeilen	0,19	S	4030,11	765,7209	4795,8309
1	2 - Rechnung pauschal	0,19	S	-17,5	-3,325	-20,825

### Datenquelle

Tabelle: *tbl\_Ausgang, tbl\_Rechnung\_Zusatz*

### Datenziel

Ansicht: *viw\_Rechnung\_Summe*

```
001 SELECT "Rechnung_ID", '1 - Rechnungszeilen' AS "Sort", "Steuersatz",
      "Steuer_Code_ID",
002     SUM( "AnzahlPreis" ) AS "Netto",
003     SUM( "AnzahlPreis" ) * "Steuersatz" AS "Steuer",
004     SUM( "AnzahlPreis" ) * ( 1 + "Steuersatz" ) AS "Brutto"
005 FROM
006     ( SELECT "Anzahl" * "Preis" * (1 - COALESCE ( "Rabatt", 0 )) AS "AnzahlPreis",
007         "Steuersatz", "Steuer_Code_ID", "Rechnung_ID"
008     FROM "tbl_Ausgang" )
009     AS "a"
010 GROUP BY "Rechnung_ID", "Steuersatz", "Steuer_Code_ID"
011 UNION
```

```

012 SELECT "Rechnung_ID", '2 - Rechnung pauschal' AS "Sort", "Steuersatz",
      "Steuer_Code_ID",
013     SUM( "Betrag" ) AS "Netto",
014     SUM( "Betrag" ) * "Steuersatz" AS "Steuer",
015     SUM( "Betrag" ) * ( 1 + "Steuersatz" ) AS "Brutto"
016 FROM "tbl_Rechnung_Zusatz"
017 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 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** bei einem leeren Feld "Rabatt" stattdessen '0' genutzt.

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

	Rechnung_ID	Sort	Steuersatz	Steuer_Code_ID	Netto	Steuer	Brutto
▶	1	1 - Rechnungszeilen	0,07	S	83,32	5,8324	89,1524
	1	1 - Rechnungszeilen	0,19	S	4030,11	765,7209	4795,8309
	1	2 - Rechnung pauschal	0,19	S	-17,5	-3,325	-20,825
	1	3 - Summe Rechnungszeilen			4113,43	771,5533	4884,9833
	1	4 - Summe Rechnung pauschal			-17,5	-3,325	-20,825
	1	5 - Summe Steuersatz	0,07	S	83,32	5,8324	89,1524
	1	5 - Summe Steuersatz	0,19	S	4012,61	762,3959	4775,0059
	1	6 - Gesamt			4095,93	768,2283	4864,1583

### Datenquelle

Ansicht: [viw\\_Rechnung\\_Steuer](#)

### Datenziel

Ansicht: [viw\\_Rechnung](#)

Formular: [frm\\_Kunde](#)

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

```

001 SELECT * FROM "viw_Rechnung_Steuer"
002 UNION

```

```

003 SELECT "Rechnung_ID", '3 - Summe Rechnungszeilen' AS "Sort", NULL AS "Steuersatz",
      NULL AS "Steuer_Code_ID",
004     SUM( "Netto" ) AS "Netto",
005     SUM( "Steuer" ) AS "Steuer",
006     SUM( "Brutto" ) AS "Brutto"
007 FROM "viw_Rechnung_Steuer"
008 WHERE LEFT("Sort",1) = '1'
009 GROUP BY "Rechnung_ID"
010 UNION
011 SELECT "Rechnung_ID", '4 - Summe Rechnung pauschal' AS "Sort",
      NULL AS "Steuersatz", NULL AS "Steuer_Code_ID",
012     SUM( "Netto" ) AS "Netto",
013     SUM( "Steuer" ) AS "Steuer",
014     SUM( "Brutto" ) AS "Brutto"
015 FROM "viw_Rechnung_Steuer"
016 WHERE LEFT("Sort",1) = '2'
017 GROUP BY "Rechnung_ID"
018 UNION
019 SELECT "Rechnung_ID", '5 - Summe Steuersatz' AS "Sort", "Steuersatz",
      "Steuer_Code_ID",
020     SUM( "Netto" ) AS "Netto",
021     SUM( "Steuer" ) AS "Steuer",
022     SUM( "Brutto" ) AS "Brutto"
023 FROM "viw_Rechnung_Steuer"
024 GROUP BY "Rechnung_ID", "Steuersatz", "Steuer_Code_ID"
025 UNION
026 SELECT "Rechnung_ID", '6 - Gesamt' AS "Sort", NULL AS "Steuersatz",
      NULL AS "Steuer_Code_ID",
027     SUM( "Netto" ) AS "Netto",
028     SUM( "Steuer" ) AS "Steuer",
029     SUM( "Brutto" ) AS "Brutto"
030 FROM "viw_Rechnung_Steuer" 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\_gry\_...» 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\_Forms

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

Datenquelle
Ansicht: <i>viw_Filter</i>

Datenziel
Formular: <i>frm_Einstellungen, frm_Kunde, frm_Lieferant, frm_Konto</i>

```
001 SELECT
002 TRIM(BOTH FROM "Anzeige") AS "Anzeige",
003 TRIM(BOTH FROM "Formular") AS "Formular", "Sort"
004 FROM
005 (SELECT ' Kunde' AS "Anzeige", 'frm_Kunde' AS "Formular". 'Kunde' AS "Sort"
      FROM "viw_Filter"
006 UNION
007 SELECT ' Einstellungen' AS "Anzeige", 'frm_Einstellungen' AS "Formular",
      'Einstellungen' AS "Sort" FROM "viw_Filter"
008 UNION
009 SELECT ' Lieferant' AS "Anzeige", 'frm_Lieferant' AS "Formular",
      'Lieferant' AS "Sort" FROM "viw_Filter"
010 UNION
011 SELECT ' Konto' AS "Anzeige", 'frm_Konto' AS "Formular", 'Konto' AS "Sort"
      FROM "viw_Filter")
012 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\_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 €

<b>Datenquelle</b>	
Tabelle:	<i>tbl_Lieferant, tbl_Lieferung</i>
Ansicht:	<i>viw_Rechnung</i>

<b>Datenziel</b>	
Formular:	<i>frm_Konto</i>

```

001 SELECT "a".*,
002     SUM("Wert") OVER (ORDER BY "ZahlDatum") AS "lfdSumme"
003 FROM
004     (SELECT 'Lieferung' AS "Titel", "tbl_Lieferant"."Firma",
005           "tbl_Lieferung"."Datum", "tbl_Lieferung"."LieferantZeichen",
006           "tbl_Lieferung"."NochOffen" AS "Brutto", "tbl_Lieferung"."ZahlDatum",
007           "tbl_Lieferung"."ID", "tbl_Lieferung"."NochOffen" * -1 AS "Wert"
008     FROM "tbl_Lieferung", "tbl_Lieferant"
009     WHERE "tbl_Lieferung"."Lieferant_ID" = "tbl_Lieferant"."ID"
010           AND "tbl_Lieferung"."ZahlDatum" IS NOT NULL
011     UNION
012     SELECT 'Rechnung' AS "Titel", "Firma", "Rechnungsdatum", "Rechnungsnummer",
013           ROUND("Brutto",2) AS "Brutto", "ZahlDatum", "Rechnung_ID",
014           ROUND("Brutto",2) AS "Wert"
015     FROM "viw_Rechnung"
016     WHERE "ZahlDatum" IS NOT NULL
017     ORDER BY 6, 3, 7) AS "a"

```

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 gleichen "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\_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 FROM "tbl_Lieferung" AS "a"
```

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 3). 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.

Grundsätzlich wäre das Ganze auch im Formular möglich, indem einfach ein Listenfeld für die Darstellung des Inhaltes genutzt würde. Nur kostet das Verfahren mit dem Listenfeld unnötig Speicher, da die gesamten Inhalte aus der "tbl\_Lieferant" für jedes Listenfeld in dem Tabellenkontrollfeld geladen werden müssen.

## 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".*,
002     ( SELECT "Rechnungsnummer"
        FROM "viw_Rechnung" WHERE "Rechnung_ID" = "a"."ID" ) AS "Rechnungsnummer",
```

```
003 ( SELECT "Firma_Kunde" || ' → E-Mail: ' || "E-Mail_Kunde"
    FROM "vw_Rechnung" WHERE "Rechnung_ID" = "a"."ID" ) AS "Kunde",
004 ( SELECT "Brutto"
    FROM "vw_Rechnung" WHERE "Rechnung_ID" = "a"."ID" ) AS "Brutto"
005 FROM "tbl_Rechnung" AS "a"
```

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.



<b>Makros in Feldeigenschaften</b>		
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>

<b>Formularfilter</b>	
1	001 ( "tbl_Filter"."ID" = (SELECT "ID" FROM "viw_Filter") )
2	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 **Firmeneinstellungen speichern** ist direkt über **Steuerelement-Eigenschaften → Allgemein → Aktion → Datensatz speichern** mit der Speicherfunktion verbunden.

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.

# frm\_Kunde

**Kunde**

Filter  
Kunde

ID

Firma\*  AnsprechP  AnsprechPGeschlecht

StraßeNr\*  PL\*  Ort\*  Land\*

E-Mail\*  Telefon  UST-IdNr

Datensatz 1 von 1

**Rechnungserstellung**

ID

Datum\*  KundeZeichen\*  Mitarbeiter der Firma\*

Zahlungsnotiz\_1  Zahlungsnotiz\_2

Datensatz 1 von 2

Rechnung

Ermäßigungen und Aufschläge auf Rechnungsebene

Zusatz*	Beitrag*	Steuersatz*	Steuercode*	Netto	Steuer	Brutto
Portokosten	1,00 €	19 %	Normalsatz	0,95,93 €	768,23 €	4.864,16 €
Haar	-2,00 €	19 %	Normalsatz			

Datensatz 1 von 2

**Rechnungsposten**

Anzahl*	Einheit*	Bestellnummer	Ware*	Preis*	Rabatt	Steuersatz*	Steuercode*
2	Stück		Open-Source DVD	12,45 €	10 %	19 %	Normalsatz
3	Stück		Handbuch Base 7.6, gedruckt	21,90 €		19 %	Normalsatz
45	Stück		Stunden Arbeit am XRechnung-For	25,00 €		19 %	Normalsatz
3	Stück		Tafeln Schokolade mit Mandeln	1,75 €		7 %	Normalsatz
1	Stück		Dose Brustkaramellen für den	4,50 €		7 %	Normalsatz
1	Stück		Laufschuhe	142,00 €	5 %	19 %	Normalsatz

Datensatz 1 von 17

**Anhänge**

mimeType\*

Anhang

Dateiname

Beschreibung

Datensatz 1 von 2

<b>1</b>	frmFilter (Tabelle: <i>tbl_Filter</i> )				
<b>2</b>	frmKunde (Tabelle: <i>tbl_Kunde</i> )	<b>2.1</b>	frmRechnung (Tabelle: <i>tbl_Rechnung</i> )	<b>2.1.1</b>	frmRechnungZusatz (Tabelle: <i>tbl_Rechnung_Zusatz</i> )
				<b>2.1.2</b>	frmAusgang (Tabelle: <i>tbl_Ausgang</i> )
				<b>2.1.3</b>	frmSumme (Ansicht: <i>viw_Rechnung_Summe</i> )
				<b>2.1.4</b>	frmDruck (Ansicht: <i>viw_Rechnung</i> )
				<b>2.1.5</b>	frmAnhang (Tabelle: <i>tbl_Rechnung_Anhang</i> )

<b>Makros in Formulareigenschaften</b>		
1	Beim Laden	Formularwechsel. <i>FormStart</i>
2, 2.1, 2.1.5	Beim Laden	Inputcontrol. <i>FormVars</i>
2, 2.1, 2.1.5	Vor der Datensatzaktion	Inputcontrol. <i>SaveRequired</i>
2, 2.1, 2.1.5	Nach dem Datensatzwechsel	Inputcontrol. <i>FormChange</i>
2.1.1, 2.1.2	Nach der Datensatzaktion	Filter. <i>FormParallelReload</i>
2.1.5	Löschen bestätigen	Anhang. <i>AnhangLoeschen</i>

<b>Makros in Feldeigenschaften</b>		
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>
2.1	Feld «Datum» (bei Fokusverlust) ebenso «KundeZeichen», «MitarbeiterID»	Inputcontrol. <i>FieldRequired</i>
2.1.4	<b>Rechnung drucken</b> (Aktion ausführen)	Druck. <i>FillTableCarryOver</i>
2.1.5	Feld «Anhang» (bei Fokusverlust)	Anhang. <i>AnhangEinlesen</i>
2.1.5	<b>Anhang anzeigen</b> (Aktion ausführen)	Design. <i>ShowAttachment</i>

<b>Formularfilter</b>	
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' )

<b>SQL-Code Listenfelder</b>	
1	Listenfeld Filter «Kunde» 001 SELECT "Firma"  ' → E-Mail: '  "E-Mail" AS "Field", "ID" AS "Key" 002 FROM "tbl_Kunde" 003 ORDER BY "Field"

2.1.2	Listenfeld «Einheit», Datenquelle: <i>tbl_Anzahl_Code</i> <pre> 001 SELECT * FROM 002     (SELECT "Name_DE", "Code_ID", 1 AS "Sort" 003     FROM "tbl_Anzahl_Code" WHERE "Code_ID" = 'XPP' 004     UNION 005     SELECT "Name_DE", "Code_ID", 2 AS "Sort" 006     FROM "tbl_Anzahl_Code") 007 ORDER BY 3, 1 ASC </pre>
-------	---

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.

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

Formular 2.1.4 schließlich enthält nur eine Schaltfläche. 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.

Das Formular frm\_Kunde ist über das Makro *FilterStart* als Startformular festgelegt.

# frm\_Lieferant

**Lieferant**

Filter Lieferant

ID: 25

Firma\*: Base - XRechnung

Kontaktperson: Name\* Elly Workoholic, Geschlecht, E-Mail\* work@beispiel.de, Telefon\* 0134

StraßeNr\*: Open Source Allee 42, Plz\*: 05000, Ort\*: Überall, Land\*: Deutschland

E-Mail\*: info@beispiel.de, Telefon: 13457002, IBAN\*: DE75512108001245126199, Zahlungsempfänger: LibreOffice

Datensatz 10 von 10

---

**Lieferungsaufnahme**

ID: 12

Datum\*: 08.12.23, LieferantZeichen\*: R-2010000002, UnserZeichen, Zahlungsfrist

Zahlungsnotiz: Zahlbar innerhalb von 14 Tagen ab Rechnungsdatum

Summen lt. Rechnung

Netto	1.340,26 €	Steuer	253,48 €	Brutto	1.593,74 €
bereits gezahlt	0,00 €	noch offen	1.593,74 €		

Summen berechnet

Sort	Steuer	Brutto
4 - Gesamt	1.340,26 €	1.593,74 €

Zusatz*	Betrag*	Steuersatz*
Portokosten	7,50 €	19 %
Rabatt	-25,00 €	19 %

XRech-ZeilenID	Anzahl*	Einheit	Ware*	Preis*	Rabatt	Steuersatz*
B-24-X-18	2	Stück	Open-Source DVD	12,45 €	2,49 €	19 %
HB 7.6	3	Stück	Handbuch Base 7.6, gedruckt	21,90 €	0,00 €	19 %
Schaffe-Schaffe	45	Stück	Stunden Arbeit am Rechnungsformular	25,00 €	0,00 €	19 %
E-Schoggi	3	Stück	Tafeln Schokolade mit Mandeln	1,75 €	0,00 €	7 %
E-BonBon	1	Stück	Dose Brustkaramellen für den schmutzigen	4,50 €	0,00 €	7 %
S 42,5	1	Stück	Laufschuhe	142,00 €	7,10 €	19 %

Datensatz 1 von 2

**Anhänge**

mim eCode\*: image/png

Anhang



Anhang anzeigen

Dateiname: logo-sc.png

Beschreibung: Nur das Logo von LibreOffice

Datensatz 1 von 2

<b>1</b>	frmFilter (Tabelle: <i>tbl_Filter</i> )				
<b>2</b>	frmLieferant (Tabelle: <i>tbl_Lieferant</i> )	<b>2.1</b>	frmLieferung (Tabelle: <i>tbl_Lieferung</i> )	<b>2.1.1</b>	frmLieferungZusatz (Tabelle: <i>tbl_Lieferung_Zusatz</i> )
				<b>2.1.2</b>	frmEingang (Tabelle: <i>tbl_Eingang</i> )
				<b>2.1.3</b>	frmSumme (Ansicht: <i>viw_Lieferung_Summe</i> )
				<b>2.1.4</b>	frmAnhang (Tabelle: <i>tbl_Lieferung_Anhang</i> )

<b>Makros in Formulareigenschaften</b>		
1	Beim Laden	Formularwechsel. <i>FormStart</i>
1	Listenfeld Filter «Kunde» (Status geändert)	Filter. <i>ListboxFilter</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>

<b>Makros in Feldeigenschaften</b>		
1	Listenfeld Formularwechsel (Status geändert)	Formularwechsel. <i>Navigation</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	Einlesen (Aktion ausführen)	Import. <i>Import</i>
2.1	Feld «Datum» (bei Fokusverlust) ebenso «LieferantZeichen»	Inputcontrol. <i>FieldRequired</i>
2.1.4	Anhang anzeigen (Aktion ausführen)	Design. <i>ShowAttachment</i>

<b>Formularfilter</b>	
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' )

<b>SQL-Code Listenfelder</b>	
1	Listenfeld Filter «Lieferant» 001 SELECT "Firma"  ' → E-Mail: '  "E-Mail" AS "Field", "ID" AS "Key" 002 FROM "tbl_Lieferant" 003 ORDER BY "Field"

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.

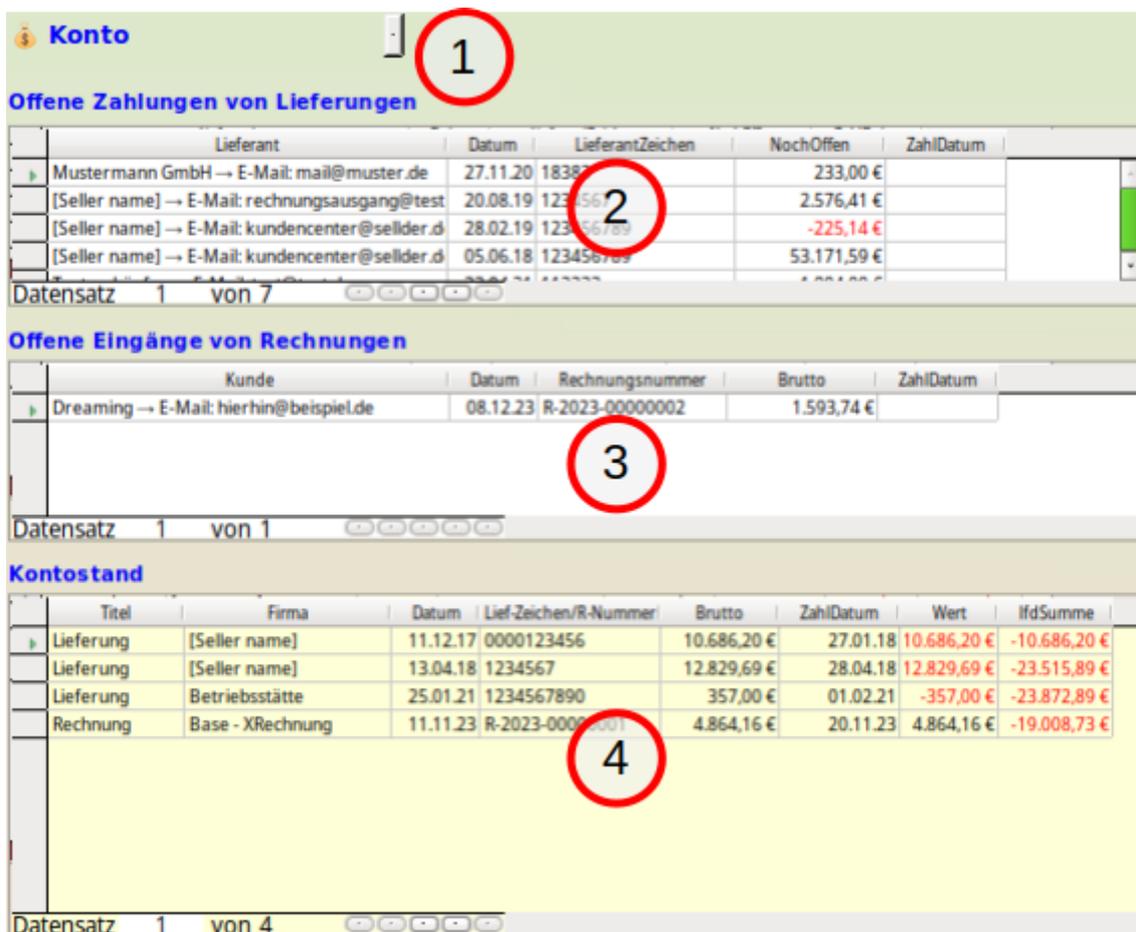
Formular 2.1 zeigt nur die Lieferungen an, die zum aktuell ausgewählten Lieferanten passen. Hier wird dann neben den Informationen im Formular für die Kunden auch die Zahlungsfrist (sofern angegeben und auslesbar) angegeben. Zusätzlich wird angezeigt, welche Summen in der Rechnung angegeben wurden.

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\_Konto



1	frmFilter (Tabelle: <i>tbl_Filter</i> )
2	frmLieferungen (Abfrage: <i>qry_Lieferung_Zahlung</i> )
3	frmRechnungen (Abfrage: <i>qry_Rechnung_Zahlung</i> )
4	frmSumme (Abfrage: <i>qry_Konto</i> )

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

<b>Makros in Feldeigenschaften</b>		
1	Listenfeld Formularwechsel (Status geändert)	Formularwechsel. <i>Navigation</i>

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

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.

Formular 2 und 3 sind nur für das Ändern von Datensätzen freigeschaltet. Datensätze lassen sich nicht löschen, nur das letzte Feld in den beiden Tabellenkontrollfeldern, das Feld "ZahlDatum", kann mit Inhalt gefüllt und der Datensatz dann abgespeichert werden.

Beide Formulare sind außerdem so eingestellt, dass sie beim Einlesen nur die Datensätze anzeigen, bei denen das "ZahlDatum" keinen Eintrag hat. Nach der Abspeicherung wird in Formular 4 dann der entsprechende Datensatz übernommen und die Summierung in dem Konto stand sichtbar.

# Ausdruck

## Datenquelle

Makro: *FillTableCarryOver*

Die Datenbank enthält keine Berichte. Der gesamte Ausdruck erfolgt über eine Vorlagendatei.

Base--XRechnung--Open-Source-Allee-42--05040-Überall

Dreaming  
Hamburger-Damm-12a  
90919-Irgendwo



**Base--XRechnung**  
Open-Source-Allee-42  
05040-Überall  
0203040512345  
info@beispiel.de  
MitarbeiterIn:  
Elly Workoholic  
work@beispiel.de  
013457902

**Rechnung**  
Rechnung-Nummer: R-2023-00000002 → Datum: 08.12.2023  
Kundennummer: K-000001

STEUERNUMMER	ZAHLUNGSART
42-42-42	Zahlbar innerhalb von 14 Tagen ab Rechnungsdatum

MENGE	BESCHREIBUNG	EINZELPREIS	RABATT	STEUER	SUMME
2 Stück	Open-Source DVD (LibreOffice, Gimp, Scribus usw.)	12,45 €	10%	19%	22,41 €
3 Stück	Handbuch Base-7.6, gedruckt (alles über Base auf über 700 Seiten)	21,90 €		19%	65,70 €
45 Stück	Stunden Arbeit am XRechnung-Formular Erstellung eines einfachen Formulars, anschließend übertragen des Formulars nach Base hin (besser bedienbar), anschließend auch Import von XRechnungen nach Base	25,00 €		19%	1.125,00 €
3 Stück	Tafeln Schokolade mit Mandeln	1,75 €		7%	5,25 €
1 Stück	Dose Brustkaramellen für den schmuddeligen 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 €
		<b>Netto:</b>			1.357,76 €
		zuzgl. Portokosten:			7,50 €
		abzgl. Rabatt:			25,00 €
		MwSt-7%:			0,68 €
		MwSt-19%:			252,80 €
		<b>Gesamt:</b>			1.593,74 €

FIRMA (REG. NAME)
LibreOffice-Testcenter
Open-Source-Allee-42
05040-Überall

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

BANK/IBAN/BIC
Spasskass e Überall
DE75512108001245126199
SPKAUEBL42

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. Um in der Spalte für den Rabatt nicht

laufend 0 % stehen zu haben muss mittels **Tabelle** → **Zahlenformat** der Formatcode auf [=0]#;0" "% eingestellt werden.

<FIRMA> <STRASSE NR> <PLZ> <ORT> <ANSCHRIFT_KUNDE>		<FIRMA> <STRASSE NR> <PLZ> <ORT> <TELEFON> <E-MAIL> MitarbeiterIn: <NAME> <E-MAIL MITARBEITER> <TELEFON MITARBEITER>													
<b>Rechnung</b> Rechnung Nummer: <RECHNUNGSNUMMER> → Datum: <RECHNUNGSDATUM D> Kundennummer: <KUNDENNUMMER>															
STEUERNUMMER <UST-IdNo>		ZAHLUNGSART <ZAHLUNGSNOTIZ>													
<table border="1"> <thead> <tr> <th>MENGE</th> <th>BESCHREIBUNG</th> <th>EINZELPREIS</th> <th>RABATT</th> <th>STEUER</th> <th>SUMME</th> </tr> </thead> <tbody> <tr> <td> </td> <td> </td> <td> </td> <td> </td> <td> </td> <td> </td> </tr> </tbody> </table>		MENGE	BESCHREIBUNG	EINZELPREIS	RABATT	STEUER	SUMME								
MENGE	BESCHREIBUNG	EINZELPREIS	RABATT	STEUER	SUMME										
FIRMA (REG. NAME) <FIRMA REGISTRIERT> <STRASSE NR> <PLZ> <ORT>		TELEFON/MAIL/WEB <TELEFON> <E-MAIL> <WEBSEITE>													
BANK/IBAN/BIC <BANK> <IBAN> <BIC>															

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. Darunter befindet sich die Tabelle «Rechnungsinhalt», die lediglich mit den Tabellenköpfen versehen ist und eine Zeile enthält, in der die Felder schon vorformatiert sind, so dass Währungen und Prozentangaben entsprechend formatiert erscheinen. 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 «Fir-

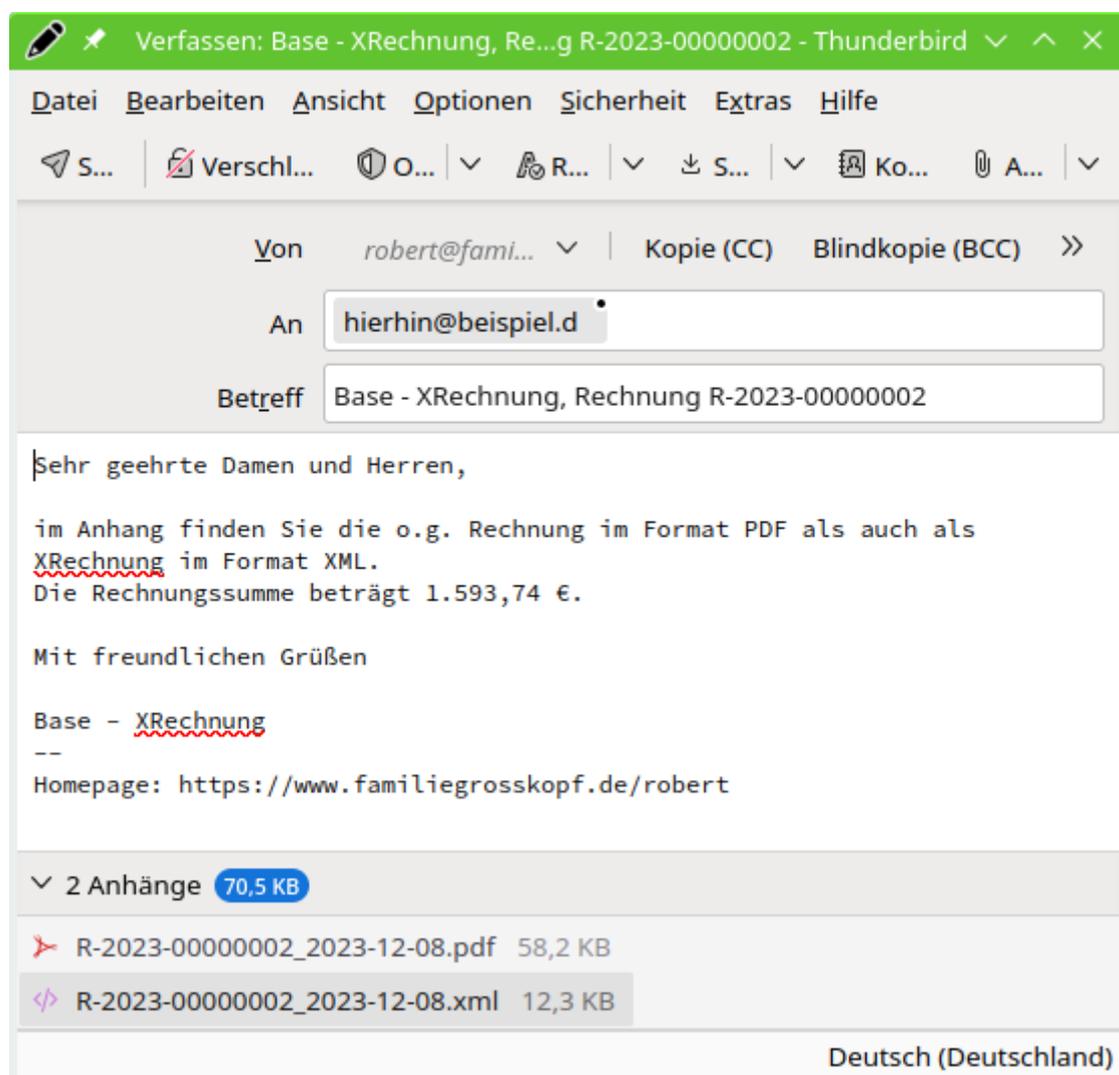
maFußzeile2». Die Fußzeilentabellen werden gegebenenfalls um eine weitere Spalte per Makro ergänzt. Dies ist abhängig davon, ob entsprechende Registerinträge in der Tabelle "tbl\_Firma" vorgenommen wurden.

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 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.

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

<b>Aufruf aus</b>
Formular: <i>frm_Kunde</i>

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

### AnhangLoeschen

<b>Aufruf aus</b>
Makro: <i>frm_Kunde</i>

<b>Benötigt</b>
Makro: <i>AnhangPfad</i>

Wird ein Anhang in der Tabelle «tbl\_Rechnung\_Anhang» gelöscht, so existiert er weiterhin als eingeleseene Datei im Binärformat und im Base64-Format. Beide verbleibenden Dateien müssen auch entfernt werden.

### AnhangPfad

<b>Aufruf aus</b>
Makro: <i>AnhangEinlesen, AnhangLoeschen</i>

**Benötigt**Tabelle: [tbl\\_Rechnung](#)

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

## Backup

### DatabaseStart

**Aufruf aus**

Datenbankdatei → Extras → Anpassen → Ereignisse → Dokument öffnen

**Benötigt**Makro: [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».

### DatabaseBackup

**Aufruf aus**Makro: [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. Ist die Maximalanzahl erreicht, dann wird die älteste Kopie durch eine neue Kopie ersetzt.

### BackupTarget

**Aufruf aus**Makro: [DatabaseBackup](#)

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: [DatabaseStart](#)**Benötigt**Tabelle: [tbl\\_Filter](#)Ansicht: [viw\\_Filter](#)Formular: [frm\\_Kunde](#)

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.

## DataWrite

### Aufruf aus

Datenbankdatei → Extras → Anpassen → Ereignisse → Ansicht wird geschlossen

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 Dokument<sup>8</sup>, 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*

<sup>8</sup> <https://ask.libreoffice.org/t/base64-encode-decode/25761>

## EncodedToSixBit

### Aufruf aus

Makro: [Base64StringToByteStream](#)

## SixBitToEncoded

### Aufruf aus

Makro: [ByteStreamToBase64Stream](#)

## ByteStreamToBase64Stream

### Aufruf aus

Makro: [AnhangEinlesen](#)

### 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

### 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: [frm\\_Einstellungen](#), [frm\\_Kunde](#), [frm\\_Lieferant](#)

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: [frm\\_Einstellungen](#), [frm\\_Kunde](#), [frm\\_Lieferant](#)

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

<b>Aufruf aus</b>
Formular: <i>frm_Kunde, frm_Lieferant</i>

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

## Druck

### FillTableCarryOver

<b>Aufruf aus</b>
Formular: <i>frm_Kunde</i>

<b>Benötigt</b>
Tabelle: <i>tbl_Rechnung_Zusatz</i>
Ansicht: <i>viw_Rechnung_Ausgang, viw_Rechnung_Summe, viw_Rechnung</i>
Formular: <i>frm_Kunde</i>
Makro: <i>ExportImageIntern, FillTextfields, ExportPDF</i>

Diese Prozedur startet den *Ausdruck* und erweitert in der Vorlagendatei 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.

### FillTextfields

<b>Aufruf aus</b>
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: [FillTableCarryOver](#)

### Benötigt

Makro: [SaveXRechnung](#), [StartMail](#)

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.

## Filter

### FormParallelReload

### Aufruf aus

Formular: [frm\\_Kunde](#), [frm\\_Lieferant](#), [frm\\_Konto](#)

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

<b>Aufruf aus</b>
Formular: <i>frm_Kunde, frm_Lieferant</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.

## Formularwechsel

### Navigation

<b>Aufruf aus</b>
Formular: <i>frm_Einstellungen, frm_Kunde, frm_Lieferant, frm_Konto</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

<b>Aufruf aus</b>
Formular: <i>frm_Einstellungen, frm_Kunde, frm_Lieferant, frm_Konto</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 Fensternamen abgespeichert und das Formular neu geladen wird.

## Import

### InsertSQL

<b>Aufruf aus</b>
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.

### DecimalPoint

<b>Aufruf aus</b>
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 Dezimatrenner wird als Punkt in einem Text festgeschrieben.

## Import

<b>Aufruf aus</b>
Formular: <i>frm_Lieferant</i>

<b>Benötigt</b>
Tabelle: <i>tbl_Lieferant, tbl_Lieferung, tbl_Lieferung_Zusatz, tbl_Eingang</i>
Makro: <i>InsertSQL, 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

<b>Aufruf aus</b>
Formular: <i>frm_Einstellungen, frm_Kunde, frm_Lieferant</i>

<b>Benötigt</b>
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

<b>Aufruf aus</b>
Formular: <i>frm_Einstellungen, frm_Kunde, frm_Lieferant</i>
Makro: <i>FormVars</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

<b>Aufruf aus</b>
Formular: <i>frm_Einstellungen, frm_Kunde, frm_Lieferant</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

<b>Aufruf aus</b>
Formular: <i>frm_Einstellungen, frm_Kunde, frm_Lieferant</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

<b>Aufruf aus</b>
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

<b>Aufruf aus</b>
Formular: <i>frm_Einstellungen, frm_Kunde, frm_Lieferant</i>

<b>Benötigt</b>
Makro: <i>MailValid, FieldRequired</i>

Zwei Prozeduren werden direkt nacheinander gestartet: Zuerst die Überprüfung der Mailadresse und dann noch, ob das Feld eventuell mit Inhalt gefüllt ist oder komplett leer geblieben ist.

## IBANValid

<b>Aufruf aus</b>
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

<b>Aufruf aus</b>
Formular: <i>frm_Einstellungen, frm_Lieferant</i>

### **Benötigt**

Makro: *IBANValid, FieldRequired*

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.

## Wartung

### **TableRestart**

#### **Aufruf aus**

Datenbankdatei → Extras → Makros → Makro ausführen...

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...

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.

## XRechnung

### **SaveXRechnung**

#### **Aufruf aus**

Makro: *ExportPDF*

#### **Benötigt**

Tabelle: *tbl\_Rechnung\_Zusatz, tbl\_Rechnung\_Anhang, tbl\_mimeCode*

Ansicht: *viw\_Rechnung, viw\_Rechnung\_Ausgang, viw\_Rechnung\_Summe*

Makro: *Round2Decimalplaces*

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

<b>Aufruf aus</b>
Makro: <a href="#">SaveXRechnung</a>

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.

## StartMail

<b>Aufruf aus</b>
Makro: <a href="#">ExportPDF</a>

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 unter **Extras → Optionen → Internet → E-Mail** das passende Mailprogramm angegeben werden, das das Standardmailprogramm des Systems ist.