



LibreOffice
The Document Foundation

Base

Kapitel 6
Berichte

Copyright

Dieses Dokument unterliegt dem Copyright © 2015. 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

Jost Lange

Jochen Schiffers

Michael Niedermair

Rückmeldung (Feedback)

Kommentare oder Vorschläge zu diesem Dokument können Sie in deutscher Sprache an die Adresse 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 01.02.2023. Basierend auf der LibreOffice Version 7.5.

Inhalt

Kapitel 6 Berichte	1
Berichte mit dem Report-Designer	4
Die Benutzeroberfläche des Report-Designers	5
Gliederung eines Berichtes	13
Allgemeine Eigenschaften von Feldern	13
Besondere Eigenschaften des grafischen Kontrollfeldes	17
Diagramme im Bericht einbinden	18
Dateneigenschaften von Feldern	23
Funktionen im Report-Designer	24
Formeleingaben	24
Benutzerdefinierte Funktionen	31
Formeleingabe für ein Feld	33
Bedingte Anzeige	33
Bedingte Formatierung	34
Beispiele für Berichte mit dem Report-Designer	35
Rechnungserstellung	35
Ausdruck von Berichten zum aktuellen Datensatz des Formulars	42
Aufbau der Filtertabelle	43
Aufbau des Makros zum Start des gefilterten Berichtes	43
Wechselnde Einfärbung von Zeilen	44
Zweispaltige Berichte	46
Bugs und Workarounds beim Report-Designer	50
Der Inhalt eines Feldes aus einer Abfrage erscheint nicht	50
Ein Bericht lässt sich nicht ausführen	50
Datums- und Zeitwerte werden in Diagrammen nicht angezeigt	51
Gruppierungen mit wiederholendem Bereich zeigen nur den ersten Wert	52
Ein Neustart der Seitenzählung mit der Gruppe ist nicht möglich	53
Nachträgliche Bearbeitung des Berichtsdokuments	56
Andere Formen der Berichtserstellung	56
Das alte Berichtsmodul	57
Die BaseReportExtension	57
Weitere Berichtsmöglichkeiten	57

Berichte mit dem Report-Designer

Mit Hilfe von Berichten werden Daten so dargestellt, dass sie auch für Personen ohne Datenbankkenntnisse gut lesbar sind. Berichte können

- Daten tabellarisch gut lesbar darstellen,
- zu Daten Diagramme erstellen,
- mit Hilfe von Daten Etikettendruck ermöglichen,
- Serienbriefe wie z. B. Rechnungen, Mahnungen oder auch nur Bestätigungen über einen Vereinsbeitritt oder -austritt erstellen.

Um einen Bericht zu erstellen, muss die Datenbankgrundlage des Berichtes gut vorbereitet sein. Ein Bericht kann nicht, wie ein Formular, Unterberichte und damit zusätzliche Datenquellen aufnehmen. Ein Bericht kann auch nicht, wie im Formular, über Listenfelder andere Daten darstellen als in der zugrundeliegenden Datenquelle vorhanden sind.

Am besten werden Berichte mit Abfragen vorbereitet. Dort sollten alle variablen Inhalte festgeschrieben werden. Es sollte aber, wenn in dem Bericht noch sortiert werden soll, auf jeden Fall eine Abfrage erstellt werden, die das Sortieren zulässt. Dies bedeutet, dass Abfragen im direkten SQL-Modus unter diesen Bedingungen vermieden werden müssen. Muss der Datenbestand über so eine Abfrage zur Verfügung gestellt werden, so lässt sich eine Sortierung erreichen, indem aus der Abfrage eine «Tabellenansicht» erstellt wird. Diese Ansicht ist in der grafischen Benutzeroberfläche von Base immer sortierbar und filterbar.

Vorsicht



Der Report-Designer ist beim Editieren eines Berichtes mit laufendem Abspeichern zu begleiten. Dazu zählt nicht nur das Abspeichern im Report-Designer selbst nach jedem wichtigen Arbeitsschritt, sondern auch das Abspeichern der gesamten Datenbank.

Je nach Version von LibreOffice kann es beim Editieren auch zu plötzlichen Abstürzen des Report-Designers kommen.

Die Funktionsweise fertiger Berichte ist davon nicht betroffen – auch wenn diese Berichte unter einer anderen Version erstellt wurden, wo eben z. B. das oben genannte Verhalten nicht auftaucht.

Hinweis

Der Report-Designer ist seit LO 4.1 bereits so in LO integriert, dass er nicht mehr sichtbar in den Erweiterungen («Extensions») erscheint. Es sollte auf jeden Fall vermieden werden, neben dem integrierten Report-Designer noch eine nicht zu der LO-Version passende Report-Designer-Erweiterung zu installieren. Bei der Nutzung von LO-Versionen, die von Ubuntu angeboten werden, ist darauf zu achten, dass der Report-Designer auch wirklich mit installiert wird.

Neben dem Report-Designer existiert auch in LibreOffice Base weiterhin ein Assistent zur Berichtserstellung mittels eines Serienbriefverfahrens. Dieser Assistent ist allerdings nicht zugänglich, sobald der Report-Designer installiert wurde.

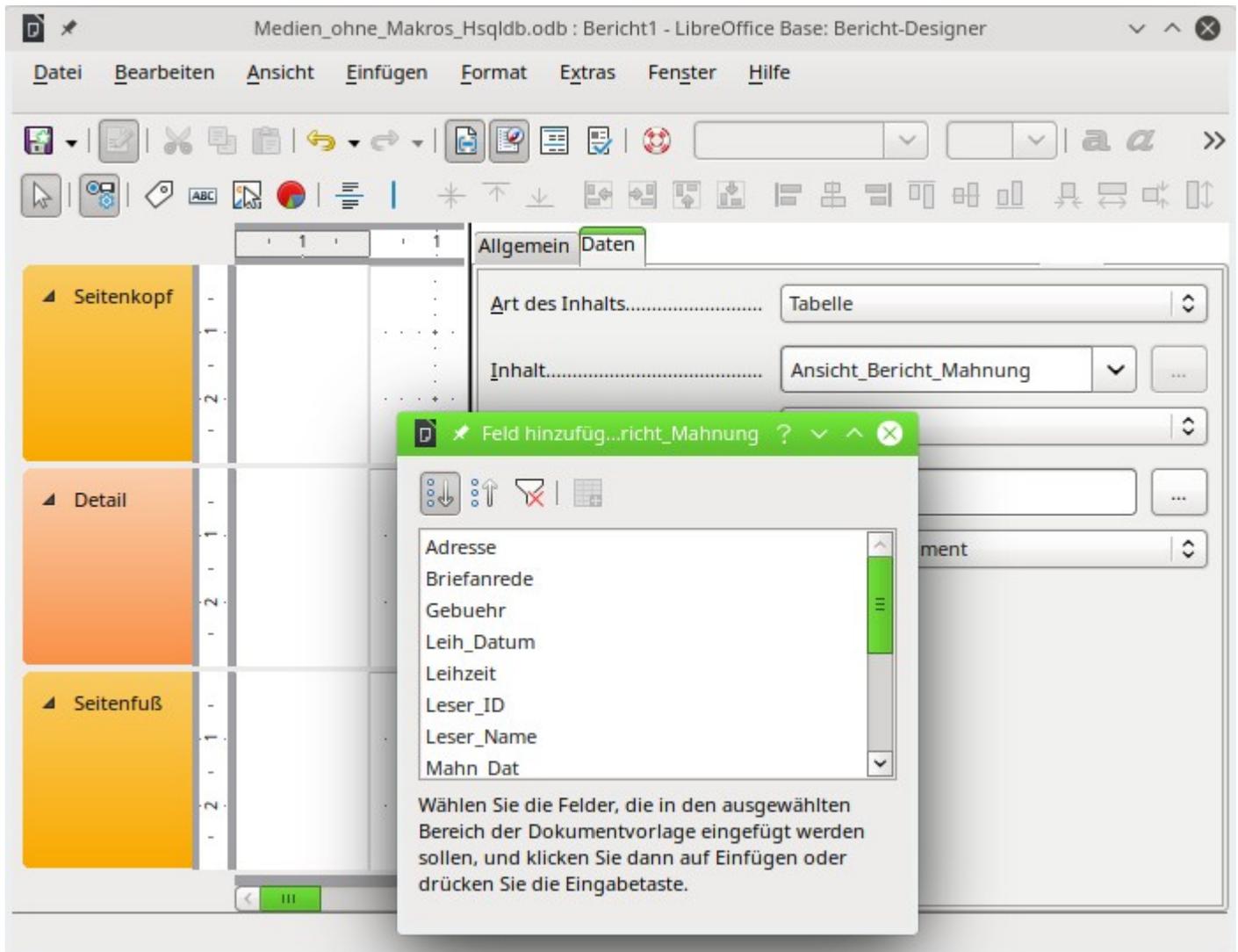
Der Assistent ist weitgehend selbsterklärend. Er produziert schnell übersichtliche Berichte in vorgefertigten Designs. Seine Berichte sind rein tabellarisch angelegt.

Da mit dem Report-Designer Berichte weiter ausgestaltet werden können, behandelt dieses Kapitel nur diese Berichtsvariante. Auch der Report-Designer kann über einen Assistenten gestartet werden. Hier wird allerdings die Bearbeitung in der Entwurfsansicht erläutert.

Leider tauchen im Report-Designer immer wieder ärgerliche Bugs auf. Am besten dafür einfach den Bugtracker <https://bugs.documentfoundation.org/buglist.cgi?quicksearch=ReportBuilder> aufsuchen. Dort stehen teilweise auch Tipps zum Umgehen von Bugs. So wird z. B. Bei einer Gruppierung des Berichts der Inhalt des äußersten Gruppe laufend wiederholt (Bug 82097). Der Tipp dazu: Eine weitere Gruppierung als erste Gruppierung benennen und diese einfach leer lassen.

Die Benutzeroberfläche des Report-Designers

Über **Berichte** → **Bericht in der Entwurfsansicht erstellen ...** wird der Report-Designer aufgerufen.



Der Report-Designer startet in einer dreiteiligen Ansicht. Links ist die vorläufige Einteilung des Berichts in Seitenkopf, Detail und Seitenfuß zu sehen, in der Mitte befinden sich die entsprechenden Bereiche, die Inhalte des Berichtes aufnehmen, und rechts werden die Eigenschaften des Berichtes angezeigt. Die oberen und unteren Seitenränder sind nicht sichtbar. Die Größe dieser Ränder kann über **Format** → **Seite** eingesehen und geändert werden.

Gleichzeitig wird bereits der Dialog **Feld hinzufügen** angezeigt. Dieser Dialog entspricht dem Dialog aus der Formularerstellung. Er erzeugt Felder mit der entsprechenden dazugehörigen Feldbezeichnung.

Ohne einen Inhalt aus der Datenbank lässt sich ein Bericht nicht sinnvoll nutzen. Deshalb wird zu Beginn direkt der Reiter **Daten** angezeigt. Hier kann der Inhalt des Berichtes eingestellt werden, im obigen Beispiel **Art des Inhalts** → **Tabelle** und **Inhalt** → **Ansicht_Bericht_Mahnung**. Solange **SQL-Befehl analysieren** → **Ja** eingestellt ist, kann der Bericht auch Sortierungen, Gruppierungen und Filterungen vornehmen. Da bereits als Grundlage eine Ansicht (*View*) gewählt wurde, kommt eine Filterung nicht zum Einsatz. Sie wurde bereits in der der Ansicht zugrundeliegenden Abfrage vorgenommen.

Hinweis

Als **Art des Inhaltes** kann eine Tabelle, eine Ansicht (View), eine Abfrage oder direkt SQL-Code angegeben werden. Am besten zu handhaben ist der Report-Designer, wenn er die Daten so weit wie möglich aufbereitet erhält. So können z.B. in Abfragen Berechnungen vorher durchgeführt werden und auch der Bereich der Daten, die im Bericht erscheinen sollen, begrenzt werden.

Vor allem bei früheren LO-Versionen kommt es manchmal zu Problemen, wenn Abfragen aus mehreren Tabellen später gruppiert werden sollen. Solche Probleme können vermieden werden, indem auf eine Ansicht zugegriffen wird. Eine Ansicht erscheint für Programmteile wie den Report-Designer wie eine Tabelle der Datenbank. Feldnamen sind dann unverrückbar vorgegeben. Der anschließende Zugriff mit Sortier- und Gruppierbefehlen ist fehlerfrei möglich.

Hinweis

Der Report-Designer kennt nicht alle Datentypen aller möglichen Datenbanken. Enthält ein Datentyp nur Text, so sollte er deshalb durch eine Abfrage in einen bekannten Textdatentyp umgewandelt werden. Dieses Problem hat vor allem **FIREBIRD** mit dem Datentyp **CLOB**:

```
001 SELECT CAST( ( "CLOB-Feld" ) AS VARCHAR ( 8000 ) ) AS "Text_lang"  
002 FROM "Tabelle"
```

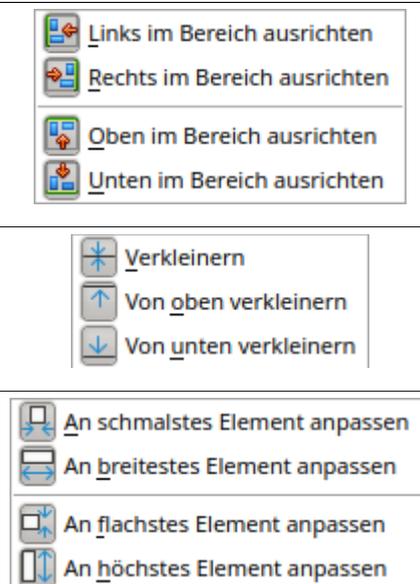
Zwei Ausgabeformate für Berichte sind über **Datei** → **Ausgabeformat** wählbar: **Textdokument**, also ein Writer-Dokument oder **Tabellendokument**, also ein Calc-Dokument. Soll einfach nur eine tabellarische Übersicht ausgegeben werden, so ist das Calc-Dokument für die Berichtserstellung eindeutig vorzuziehen. Es ist wesentlich schneller erstellt und kann anschließend auch besser nachformatiert werden, da weniger Formatvorgaben berücksichtigt werden und Spalten einfach nach der erforderlichen Breite anschließend entsprechend gezogen werden können.

Standardmäßig sucht der Report-Designer als Datenquelle die erste Tabelle der Datenbank aus. So ist auf jeden Fall gewährleistet, dass zumindest ein Ausprobieren der Funktionen möglich ist. Erst nach Auswahl der Datenquelle kann der Bericht mit Feldern beschriftet werden.

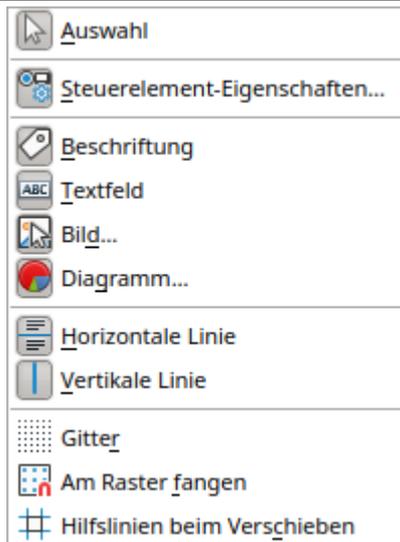
Schaltflächen inhaltliche Bearbeitung



Schaltflächen Elementausrichtung



Schaltflächen inhaltliche Bearbeitung

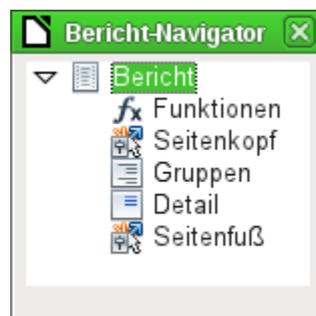


Schaltflächen Elementausrichtung



Der Report-Designer stellt einige zusätzliche Schaltflächen zur Verfügung, so dass in der vorstehenden Tabelle noch einmal die Schaltflächen mit einer entsprechenden Beschriftung abgebildet sind. Die Schaltflächen zur Elementausrichtung werden in diesem Kapitel nicht weiter beschrieben. Sie sind hilfreich beim schnellen Anpassen von Feldern in einem Bereich des Report-Designers. Prinzipiell geht dies alles aber auch über die direkte Bearbeitung der Eigenschaften des jeweiligen Feldes. Zusätzlich gibt es noch die Symbolleiste «Zeichnungsobjekte». Diese Leiste wird standardmäßig nicht angezeigt. Die Zeichnungsobjekte sind auch über **Einfügen → Form** verfügbar.

Wie schon bei den Formularen ist es hilfreich, den entsprechenden Navigator über **Ansicht → Berichtsnavigator** oder die entsprechende Schaltfläche bei Problemen aufzurufen. So kann es zum Beispiel sein, dass durch einen unvorsichtigen Klick beim Start des Report-Designers die Eigenschaften zu den Daten des Berichts verzweifelt gesucht werden. Diese Daten können nur über den Bericht-Navigator erreicht werden:



Ein Klick mit der linken Maustaste auf **Bericht** und die Eigenschaften des Berichtes sind wieder erreichbar.

Der Navigator zeigt zu Beginn neben den sichtbaren Unterteilungen des Dokumentes (Seitenkopf, Gruppen, Detail und Seitenfuß) noch die möglichen Inhalte von Funktionen an. Gruppen ordnen z. B. alle anzumahnenenden Medien einer Person zu, so dass nicht viele Einzelmahnungen erstellt werden müssen. Detailbereiche zeigen die zu den Gruppen passenden Datensätze an. Funktionen dienen z. B. zur Berechnung einer Summe einer Rechnung.

Um bei dem oben geöffneten Beispiel sinnvolle Ausgaben zu erhalten, muss der Inhalt der Ansicht gruppiert wiedergegeben werden. Ein Leser soll gebündelt die Anmahnungen für alle seine entliehenen und überzogenen Medien erhalten.

Über **Ansicht → Sortierung und Gruppierung** bzw. den entsprechenden Button startet die Gruppierungsfunktion. Hier können neben Gruppen, die angezeigt werden sollen, auch einfach Sor-

tierungen eingestellt werden. Soll nur sortiert werden, so wird zwar eine Gruppierung ausgewählt, der Gruppenkopf und der Gruppenfuß aber auf «Nicht vorhanden» eingestellt.

Zuerst erfolgt die Sortierung nach dem obersten eingetragenen Feld. Gegebenenfalls müssen also die Einträge entsprechend verschoben werden, wenn die gewünschte Sortierung nicht eintritt.



Abbildung 1: Sortierung und Gruppierung

Hier wurde nach dem Feld "Leser_Name" gruppiert und sortiert. In die obere Tabelle können untereinander mehrere Felder eingetragen werden. Soll z. B. zusätzlich nach dem "Leih_Datum" gruppiert und sortiert werden, so ist dies als zweite Zeile anzuwählen.

Direkt unter der Tabelle erscheinen verschiedene Gruppenaktionen zur Auswahl: Eine Verschiebung der Gruppe nach oben, eine Verschiebung nach unten oder die komplette Löschung der Gruppe. Da für den geplanten Bericht nur eine Gruppierung notwendig ist, steht in [Abbildung 1](#) nur mit dem Symbol ganz rechts die **Gruppenaktion → Löschung** zur Verfügung.

Die **Eigenschaft → Sortierung** ist selbsterklärend. Bei der Erstellung des Eintrags hat sich im Report-Designer auf der linken Seite sofort eine neue Einteilung gebildet. Neben der Feldbezeichnung "Leser_Name" steht dort noch «Kopf». Diese Einteilung nimmt also die Kopfzeile des Berichtes auf. In der Kopfzeile steht nachher z. B. der Name der Person, die eine Mahnung erhalten soll. Ein Gruppenfuß ist hingegen bisher nicht vorhanden. Er könnte z. B. den zu zahlenden Betrag oder den Ort und das aktuelle Datum sowie einen Bereich für die Unterschrift der anmahrenden Person enthalten.

Standardmäßig wird nach jedem Wert gruppiert. Ändert sich also "Leser_Name", so entsteht eine neue Gruppe. Alternativ kann hier nach dem Anfangsbuchstaben gruppiert werden. Das würde aber bei einem Mahnverfahren alle Leser-Nachnamen mit gleichem Anfangsbuchstaben zusammenfassen in einer Gruppe. 'Schmidt', 'Schulze' und 'Schulte' erhielten so eine gemeinschaftliche Mahnung. Eine wohl recht sinnlose Aktion an dieser Stelle für dieses Beispiel.

Nur wenn nach Anfangsbuchstaben gruppiert wird, kann noch zusätzlich eingegeben werden, nach wie vielen Werten die nächste Gruppe beginnen soll. Denkbar wäre hier z. B. eine Gruppierung für ein kleines Telefonbüchlein. Je nach Bekanntenkreis reicht da vielleicht eine Gruppierung nach jedem 2. Wert, also A und B in einer Gruppe, dann C und D usw.

Je nach Einstellung kann eine Gruppe entweder mit dem ersten Detail zusammen gehalten werden, oder, sofern möglich, als ganze Gruppe. Standardmäßig ist **Zusammenhalten** → **Keines** eingestellt. Für ein Mahnverfahren wird vermutlich sowieso die Gruppe so angeordnet, dass für jede Person, die eine Mahnung erhalten soll, eine Seite ausgedruckt wird. Daher ist stattdessen an anderer Stelle zu wählen, dass nach der Gruppe jeweils ein Seitenumbruch erfolgt, bevor der nächste Wert abzuarbeiten ist. Ein Zusammenhalten der ganzen Gruppe bewirkt gegebenenfalls, dass bei zu viel Inhalt auf der ersten Seite der gesamte Inhalt auf die Folgeseite rutscht!

The screenshot shows a LibreOffice Writer document titled "Libre Office Bibliothek". The document is a mail merge template for a library reminder ("Mahnung"). The layout is as follows:

- Seitenkopf:** Title "Libre Office Bibliothek".
- Leser_Name Kopf:** A text box containing the formula `=Adresse`.
- Detail:** A table with the following structure:

Mahndatu...	Mahn-Nr	Medium	Leihdatu...	Überzog...	Gebühr
<code>=Mahn_D</code>	<code>=Mahn_Nr</code>	<code>=Medium</code>	<code>=Leih_Dat</code>	<code>=ResTage</code>	<code>=Gebuehr</code>
- Leser_Name Fuß:** A text box containing the formula `=SUMME`.
- Seitenfuß:** A text box containing the text "Mit freundlichen Grüßen" and "(Bibliotheksverwaltung)".

Sind Gruppenkopf und gegebenenfalls Gruppenfuß ausgewählt, so erscheinen diese Elemente als Teile des Berichtsnavigators unter dem entsprechenden Feldnamen "Leser_Name". Zusätzlich wird auch da wieder die Möglichkeit für Funktionen geboten, die sich nur auf diese Gruppe beschränken.

Das Hinzufügen der Felder läuft über **Ansicht** → **Feld hinzufügen** wie im Formular. Allerdings sind hier die Beschreibungen und die Inhaltsfelder nicht miteinander gruppiert. Beide können

also unabhängig voneinander verschoben, in der Größe beeinflusst und auf unterschiedliche Einteilungsebenen gezogen werden. Auch das direkte Hinzufügen von Textfeldern (für den Inhalt aus der Datenbank) und Beschriftungsfeldern ist natürlich möglich.

Das obige Bild zeigt den Berichtsentwurf für die Mahnung an. Im Seitenkopf ist fest die Überschrift «Libre Office Bibliothek» als Beschriftungsfeld eingesetzt. Hier könnte auch ein Briefkopf mit Logo stehen, da auch die Einbindung von Grafiken möglich ist. Wenn die Ebene «*Seitenkopf*» heißt, so bedeutet das nicht, dass darüber kein Rand existiert. Dieser wurde in den Seiteneinstellungen bereits festgelegt und liegt oberhalb des Seitenkopfes.

«*Leser_Name Kopf*» ist die Gruppierung und Sortierung, nach der die Daten zusammengefasst werden sollen. In den Feldern, die später Daten aufnehmen, steht hellgrau die Bezeichnung der Datenfelder, die hier ausgelesen werden. So hat die dem Bericht zugrundeliegende Ansicht z. B. ein Feld mit der Bezeichnung Adresse, in dem die komplette Adresse mit Straße und Ort für die anzumahnende Person steht. Um dies in ein Feld zu setzen, sind Absatzumbrüche in der Abfrage notwendig. Mittels **CHAR(13) || CHAR(10)** (**FIREBIRD**: Statt **CHAR()** **ASCII_CHAR()** verwenden) wird in einer Abfrage ein Absatz erzeugt. Beispiel:

```
001 SELECT
002 "Anrede" || CHAR(13) || CHAR(10) || "Vorname" || ' ' || "Nachname" || CHAR(13) ||
    CHAR(10) || "Strasse" || ' ' || "Nr" || CHAR(13) || CHAR(10) || "Postleitzahl" || '
    ' || "Ort" AS "Adresse"
003 FROM "Leser"
```

Bei dem Feld «=TODAY()» handelt es sich um eine eingebaute Funktion, die das aktuelle Datum an dieser Stelle einliest. Diese Funktion ist über **Einfügen → Datum und Uhrzeit** direkt zu erreichen.

In «*Leser_Name Kopf*» sind außer der Anrede und weiteren Informationen auch die Spaltenköpfe für die anschließende Tabellenansicht untergebracht. Sie sollen ja nur einmal auftauchen, auch wenn mehrere Medien aufgelistet werden.

In den Hintergrund dieser Spaltenköpfe wurde je ein graues Rechteck gelegt. Dieses Rechteck sorgt gleichzeitig für eine entsprechende Umrandung.

Der Detailbereich wird so oft wiederholt, wie unterschiedliche Datensätze mit den gleichen Daten existieren, die in "*Leser_Name*" stehen. Hier werden also alle Medien aufgelistet, die nicht rechtzeitig zurückgegeben wurden. Auch hier liegt im Hintergrund ein Rechteck, um die Inhalte zu umranden. Das Rechteck selbst hat die Füllfarbe «weiß».

Hinweis

Grundsätzlich gibt es in LO auch die Möglichkeit, horizontale und vertikale Linien hinzu zu fügen. Im Design-Modus werden diese auch angezeigt.

Diese Linien haben den Nachteil, dass sie nur als Haarlinien ausgelegt sind. Sie lassen sich besser nachbilden, indem Rechtecke benutzt werden. Der Hintergrund der Rechtecke wird auf die Farbe Schwarz eingestellt, die Größe wird z. B. mit einer Breite von 17 cm und einer Höhe von 0,03 cm festgelegt. Dann erscheint eine horizontale Linie mit einer Dicke von 0,03 cm mit einer Länge von 17 cm.

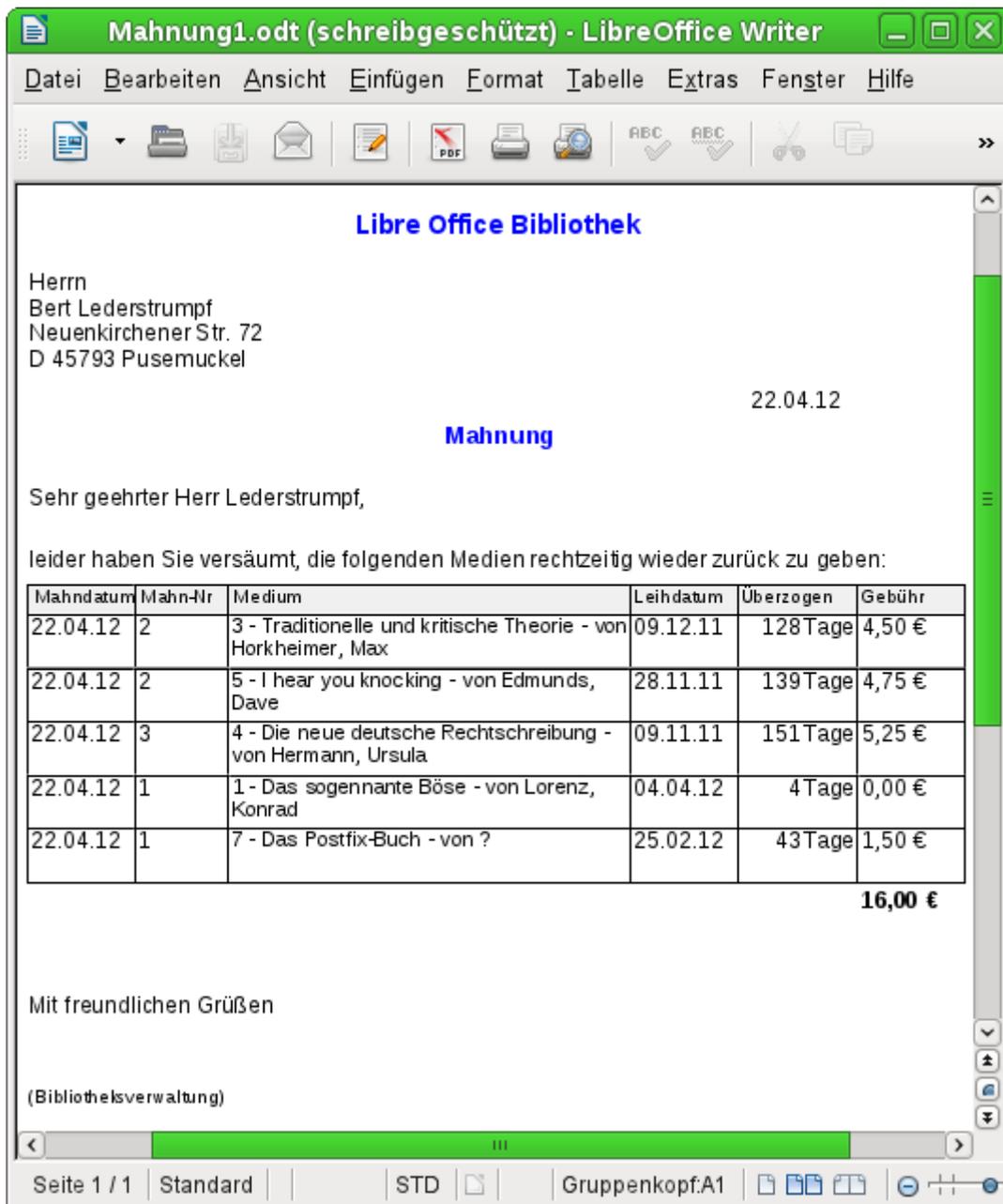
Leider hat auch diese Variante einen Nachteil: grafische Elemente lassen sich nicht richtig positionieren, wenn der Bereich über eine Seite hinweg geht.

Allgemein	
Name.....	Gruppenfuß
Seitenumbruch erzwingen.....	Nach Bereich
Zusammenhalten.....	Ja
Bereich wiederholen.....	Nein
Sichtbar.....	Ja
Höhe.....	4,87cm
Ausdruck für bedingte Anzeige....	
Hintergrund transparent.....	Ja
Hintergrundfarbe.....	

Der «*Leser_Name Fuß*» schließt schließlich das Briefformat mit einer Grußformel und einem Unterschriftsbereich ab. Der Fuß ist so definiert, dass anschließend ein Seitenumbruch nach dem Bereich erfolgt: **Seitenumbruch erzwingen → Nach Bereich**. Außerdem ist er gegenüber den Standardeinstellungen verändert worden und so eingestellt, dass der Bereich auf jeden Fall zusammen gehalten werden soll. Schließlich würde es reichlich merkwürdig aussehen, wenn bei vielen Mahnungen allein z. B. das Unterschriftsfeld auf die nächste Seite verschoben, die Gebührensumme und der Gruß aber auf der Vorseite präsentiert würde.

Hinweis

Zusammenhalten bezieht sich hier immer auf den Seitenumbruch. Soll der Inhalt eines Datensatzes unabhängig vom Seitenumbruch zusammengehalten werden, so ist dies zur Zeit nur möglich, indem der Datensatz nicht als «Detail» eingelesen wird, sondern als Grundlage für eine Gruppierung genommen wird. Der Bereich «Detail» wird leider auch dann aufgetrennt, wenn **Zusammenhalten → Ja** ausgewählt wird.



Für die Aufsummierung der Gebühren wurde eine interne Funktion benutzt. Hierfür wurde zuerst ein Textfeld ohne Inhalt aufgezogen. Anschließend wurde in den **Eigenschaften** → **Daten** → **Datenfeld-Typ** → **Funktion** gewählt. Anschließend wurde **Datenfeld** → **Gebuehr** eingestellt. Unter **Funktion** stehen Summe, Minimum und Maximum zur Verfügung. Folgerichtig wurde hier die **Summe** ausgewählt. Schließlich ist im **Geltungsbereich** noch wählbar, ob die Funktion für eine bestimmte Gruppe oder für den ganzen Bericht gelten soll. Die Summe soll bezogen auf den Leser erstellt werden. Also kam hier nur die **Gruppe: Leser_Name** in Frage. Intern weist der Report-Designer dieser Funktion den Namen *SummeGebuehrLeser_Name* zu. Aus dem Namen ist ersichtlich, dass es sich um eine Summierung der «Gebuehr» handelt. Diese Summierung wird für jede Gruppe «Leser_Name» durchgeführt.

So könnte dann eine entsprechende Mahnung aussehen. Im Detailbereich sind hier 5 Medien angegeben, die der Leser entliehen hat. Im Gruppenfuß wird die Summe für die Anmahnung ausgegeben.

Hinweis

Berichte können auch für einzelne Datensätze über mehrere Seiten gehen. Die Bereichsgröße sagt nichts über die Seitengröße aus. Allerdings kann die Ausdehnung des Bereiches Detail über mehr als eine Seite dazu führen, dass die Umbrüche nicht einwandfrei sind. Hier ist der Report-Designer in der Abstandsberechnung noch fehlerhaft. Kommen Gruppierungsbereiche und grafische Elemente hinzu, so entstehen teilweise nicht mehr durchschaubare Bereichsgrößen.

Tipp

Soll sich ein Bereich über mehrere Seiten erstrecken, so muss der Beginn des Bereiches klar auf der ersten Seite definiert sein. Um vertikale Abstände zwischen Textfeldern des Berichtes zu konstruieren, setzt der Report-Designer Tabellenzeilen ein. Passt eine Tabellenzeile nicht mehr auf eine Seite, so wird sie komplett auf die nächste Seite verschoben – mit dem gesamten folgenden Inhalt. Hier hilft es, ein kleines leeres Beschriftungsfeld genau dort zu positionieren, wo der Beginn der neuen Seite sein soll. Der Report-Designer erstellt dort dann die erste Tabellenzeile der neuen Seite. So ist die Positionierung von Textfeldern und Beschriftungsfeldern auch bei größeren Bereichen möglich.

Gliederung eines Berichtes

Die Gliederung eines Berichtes erfolgt über die Zuweisung von Gruppierungen. Zu jeder Gruppierung können Gruppenköpfe und Gruppenfüße angezeigt werden, die dann das zusammenfassende Element der Gliederung aufnehmen. Die Sortierreihenfolge wird über **Ansicht → Sortierung und Gruppierung** eingestellt. Sie ist im Editor dadurch sichtbar, dass die führende Sortierung durch den zuerst angezeigten Gruppenkopf dargestellt wird. Ist allerdings **Gruppenkopf → Nicht vorhanden** eingestellt, so muss für die Übersicht zur Sortierung weiter der Sortierungsdialog aufgesucht werden.

Soll ein Bericht mit einer besonderen Startseite versehen werden, z.B. dem Briefkopf einer Firma, so ist dies über **Bearbeiten → Berichtskopf/-fuß einfügen** möglich. Es ist nicht möglich, den Seitenkopf für eine einmalige Anzeige zu nutzen, da die bedingte Anzeige des Seitenkopfes daran scheitert, dass für den Seitenkopf zur Zeit keine unterschiedlichen Bedingungen definierbar sind. Wird der Berichtskopf genutzt, so sollte der Seitenkopf grundsätzlich in den Eigenschaften auf **Sichtbar → Nein** eingestellt werden, da sonst der Berichtskopf beim Ausdruck nicht an erster Position steht.

Grundsätzlich sollte bei Seitenkopf und Seitenfuß klar sein, dass es sich hier um eine Kopf- bzw. Fußzeile handelt, die zusätzlich zu den Randeinstellungen von **Format → Seite** Platz beansprucht. Über **Bearbeiten → Seitenkopf/-fuß löschen** kann der anfänglich automatische eingefügte Seitenkopf bzw. -fuß entfernt werden.

Hinweis

Die Möglichkeit, über das Menü **Bearbeiten** einen **Spaltenkopf/-fuß** einzustellen, ist beständig inaktiv. Die Funktion, einen Bericht mit Spalten zu versehen, wurde anfangs für den Report-Designer geplant, dann aber leider nicht realisiert. Wie so etwas dennoch teilweise nachgebildet werden kann, steht im Kapitel [Zweispaltige Berichte](#).

Allgemeine Eigenschaften von Feldern

Zur Darstellung von Daten gibt es lediglich drei unterschiedliche Felder. Neben dem Textfeld, das im Gegensatz zu seiner Namensgebung auch Zahlen und Formatierungen beherbergen kann, gibt es noch ein Feld, das Bilder aus der Datenbank aufnehmen kann. Das Diagrammfeld stellt eine Zusammenfassung von Daten dar.

The image shows a configuration window for a data field. The 'Allgemein' tab is active, and the 'Daten' sub-tab is selected. The field is named 'Gebuehr'. The 'Sichtbar' (Visible) property is set to 'Ja'. The 'Position X' is 15,55 cm, 'Position Y' is 0,00 cm, 'Breite' is 1,85 cm, and 'Höhe' is 0,50 cm. The 'Automatisches Anwachsen' (Automatic growth) property is set to 'Nein'. The 'Ausdruck für bedingte Anzeige' (Expression for conditional display) is empty. The 'Wiederholende Werte anzeigen' (Show repeating values) property is set to 'Ja'. The 'Bei Gruppenwechsel anzeigen' (Show on group change) property is set to 'Ja'. The 'Hintergrund transparent' (Background transparent) property is set to 'Ja'. The 'Hintergrundfarbe' (Background color) is set to 'Standard'. The 'Schrift' (Font) is set to 'Liberation Sans, Standard, 12'. The 'Horizontale Ausrichtung' (Horizontal alignment) is set to 'Links' (Left), and the 'Vertikale Ausrichtung' (Vertical alignment) is set to 'Oben' (Top). The 'Formatierung' (Formatting) is set to '1.234,57 €'.

Felder werden wie bei den Formularen mit Namen bezeichnet. Standardmäßig ist hier der Name gleich der Feldbezeichnung der zugrundeliegenden Datenquelle.

Ein Feld kann unsichtbar geschaltet werden. Bei Feldern macht dies vielleicht wenig Sinn, bei Gruppenkopf oder Gruppenfuß hingegen schon eher, da hier auch andere Funktionen der Gruppierung erfüllt sein sollen, der Gruppenkopf oder Gruppenfuß aber nicht unbedingt mit Inhalt versehen ist.

Automatisches Anwachsen ist in LO 6.4 neu hinzugekommen. Jetzt können Felder so definiert werden, dass sich die Höhe automatisch an den Inhalt anpasst, wenn mehr Inhalt vorhanden ist, als das vorgesehene Feld fassen kann. Allerdings ist diese Funktion nicht auf die einzelne Zelle sondern auf die gesamte Zeile bezogen. Bei Auswahl des Anwachsens *für das erste Feld* werden alle Felder in der Zeile automatisch auf den entsprechenden Inhalt angepasst.

In dem Report-Designer kann die Ansicht bestimmter Inhalte durch einen **Ausdruck für bedingte Anzeige** unterdrückt werden oder der Wert des Feldes als Grundlage für eine Formatierung von Schrift und Hintergrund genommen werden. Mehr zu diesen Ausdrücken unter [Bedingte Anzeige](#).

Wenn **Wiederholende Werte anzeigen** deaktiviert wird, so wird die Anzeige ausgesetzt, wenn direkt davor das Feld mit einem gleichen Dateninhalt bestückt wurde. Dies funktioniert einwandfrei nur bei Datenfeldern, die einen Text beinhalten. Zahlenfelder oder Datumsfelder ignorieren die Deaktivierung, Beschriftungsfelder werden bei einer Deaktivierung komplett ausgeblendet, auch wenn sie nur einmal vorkommen.

Die Funktion **Bei Gruppenwechsel anzeigen** konnte im Bericht nicht nachvollzogen werden.

Ist der Hintergrund nicht als transparent definiert, so kann für das jeweilige Feld eine Hintergrundfarbe definiert werden.

Die weiteren Einträge beziehen sich auf den Inhalt innerhalb der gezogenen Felder. Dies sind im Einzelnen die Schriftart (mit Schriftfarbe, Schriftdicke etc., siehe [Abbildung 2](#)), die Ausrichtung des Schriftzugs in dem Feld und die Formatierung mit dem entsprechenden Dialog **Zahlenformat** (siehe [Abbildung 3](#)).

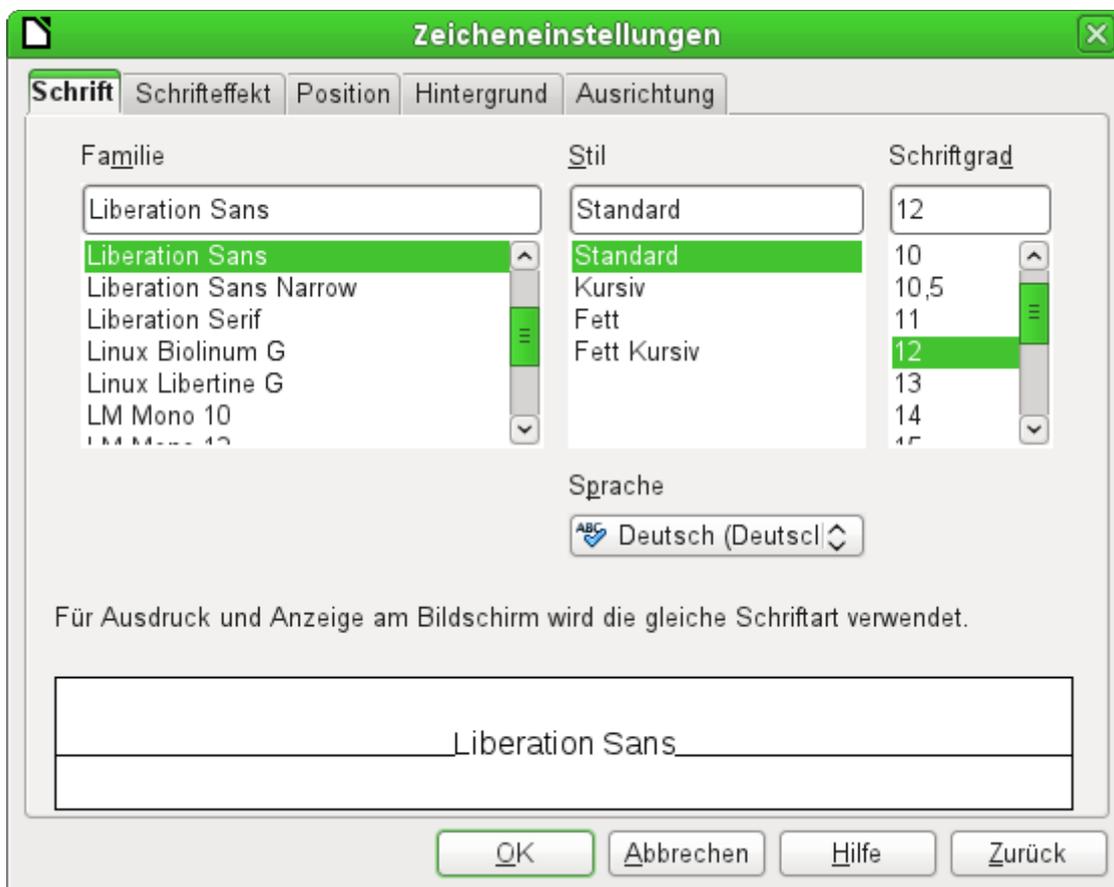


Abbildung 2: Schrift - Zeicheneinstellungen

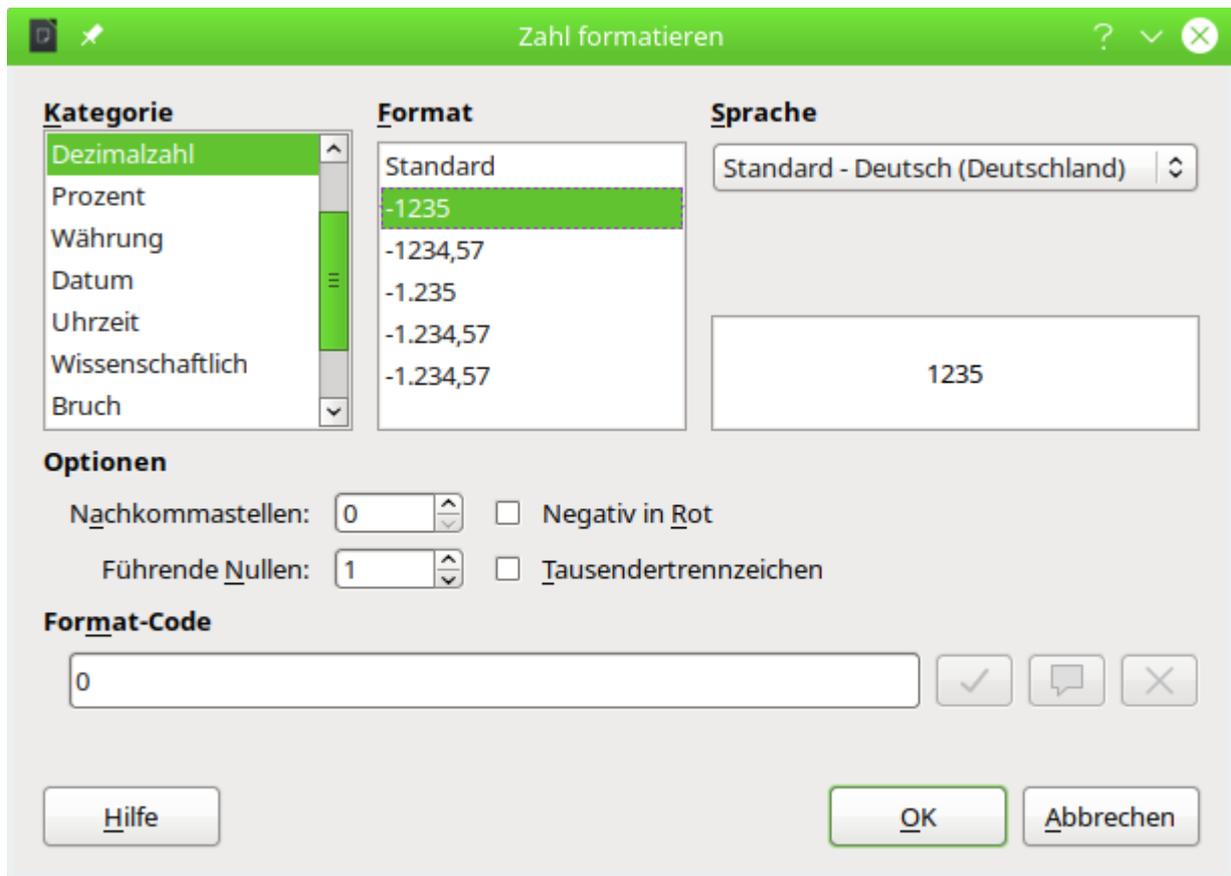


Abbildung 3: Formatierung von Zellen

Ist eine Zelle mit dem Standardformat für Dezimalzahlen belegt, so werden 0-Werte nicht angezeigt. Hier muss der **Format-Code** → **0** gesetzt werden.

Tipp

Die Positionierung von Feldern im Report-Designer ist manchmal von (scheinbar) unerklärlichen Fehlermeldungen begleitet, weil ein Element das andere überschneiden würde. Ein Blick auf die eingegebenen Werte in der Maßeinheit Zentimeter widerspricht dem. Ein Blick auf einen ausgeführten Bericht, zum Bearbeiten geöffnet, zeigt allerdings, dass bei der erzeugten Tabellenstruktur etwas nicht stimmt. Teilweise werden zum Ausgleich Tabellenspalten mit 0,04 cm Breite eingesetzt und somit viel zu viele Spalten erzeugt.

Der Report-Designer arbeitet intern alle anderen Maßeinheiten in DTP-Punkte um. Ein solcher Punkt entspricht 1/72 Zoll. Dadurch lässt sich in einem Bericht nur genau positionieren, was sich in diese Punkte umrechnen lässt. Alles andere ergibt sich durch eine Rundung. Ein Bericht, testweise über **Extras** → **Optionen** → **Spracheinstellung** → **Gebietsschema** → **Englisch (USA)** mit der Maßeinheit Zoll in einer dem Punktschema entsprechenden Seitengröße erstellt, zeigt bei einer Einhaltung der Schritte von 1/4 Zoll, dass daraus exakt positionierte Felder mit einer klaren Tabellenstruktur bei der Berichtsausführung werden.

Leider lässt sich beim Report-Designer bisher nicht, wie z. B. dem Writer, die Maßeinheit «Punkt» einstellen. Das würde sicher so einige Positionierungsprobleme lösen.

Besondere Eigenschaften des grafischen Kontrollfeldes

Allgemein		Daten
Name.....	Grafik	
Als Verknüpfung vorbereiten.....	Ja	
Position X.....	4,00cm	
Sichtbar.....	Ja	
Position Y.....	0,50cm	
Breite.....	4,00cm	
Höhe.....	4,00cm	
Wiederholende Werte anzeigen....	Ja	
Ausdruck für bedingte Anzeige.....		...
Bei Gruppenwechsel anzeigen.....	Nein	
Hintergrund transparent.....	Ja	
Hintergrundfarbe.....		
Vertikale Ausrichtung.....	Oben	
Grafik.....		...
Skalieren.....	Seitenverhältnis	

Das grafische Kontrollfeld kann sowohl Grafiken von außerhalb der Datenbank darstellen als auch Grafiken aus der Datenbank auslesen. Leider ist es zur Zeit nicht möglich, eine Grafik dauerhaft in Base zu speichern, um z. B. ein Brieflogo einzulesen. Hierzu muss zwingend die Grafik in dem gesuchten Pfad vorhanden sein, auch wenn die Auswahl anbietet, Bilder ohne Verknüpfung aufzunehmen und das erste Feld mit **Als Verknüpfung vorbereiten** auf eine entsprechende geplante Funktionalität schließen lässt.

Hinweis

Die Dateinamen dürfen keine Sonderzeichen wie [] { } \ < > % " und Leerzeichen enthalten. Ein Bericht mit diesen Bildern wird nicht erstellt.

Alternativ dazu kann natürlich eine Grafik innerhalb der Datenbank selbst gespeichert werden und so auch intern verfügbar bleiben. Sie muss dann aber über die dem Bericht zugrundeliegende Abfrage in einem der Felder zugänglich sein.

Um eine «*externe Grafik*» aufzunehmen, ist über **Grafik → Button** ... die Grafik zu laden. Um ein Datenbankfeld auszulesen, muss im Register **Daten** das Feld angegeben werden. Dieses Feld kann auch den Pfad zu einer externen Grafik beinhalten. Ist also in der Tabelle ein relativer Pfad zu einer Grafik angegeben, so wird diese Grafik entsprechend auch dargestellt.

Die Einstellung der vertikalen Ausrichtung scheint im Entwurf nichts zu bewirken. Wird allerdings der Bericht aufgerufen, so erscheint die Grafik entsprechend positioniert.

Beim **Skalieren** kann 'Nein', 'Seitenverhältnis beibehalten' und 'Autom. Größe' gewählt werden. Dies entspricht den Einstellungen im Formular:

- 'Nein': Das Bild wird nicht an das Kontrollfeld angepasst. Ist das Bild zu groß, so wird ein Ausschnitt des Bildes gezeigt. Das Bild wird nicht verzerrt.
- 'Seitenverhältnis beibehalten': Das Bild wird in das Kontrollfeld eingepasst, aber nicht verzerrt dargestellt.

- 'Automatische Größe': Das Bild wird der Größe des Kontrollfeldes angepasst und gegebenenfalls verzerrt dargestellt.

Nach der Bearbeitung von Berichten mit Bildern fällt auf, dass die Datenbankdatei deutlich größer wird. Innerhalb der *.odb-Datei legt Base aus nicht nachvollziehbaren Gründen im Berichtsverzeichnis einen Ordner «ObjectReplacements» an. Dieser Ordner enthält dann eine Datei «report», die für eine entsprechende Vergrößerung der *.odb-Datei sorgt.



Wenn die Datenbankdatei in einem Packprogramm geöffnet wird, ist dieser Ordner mit Inhalt im Verzeichnis «reports» im Unterverzeichnis zu dem jeweiligen Bericht sichtbar. Dieses Verzeichnis kann über das Packprogramm gefahrlos gelöscht werden.

Hinweis

Wenn Berichte nicht wiederholt editiert werden, reicht ein einmaliges Löschen des Verzeichnisses «ObjectReplacements» aus. Der Umfang des Verzeichnisses kann sehr rasch anschwellen. Dies hängt von der Menge und Größe der Dateien ab. Allein eine 2,8 MB *.jpg-Datei vergrößerte so in einem Test eine *.odb-Datei um 11 MB!

Der Bug ist hier gemeldet: https://bugs.freedesktop.org/show_bug.cgi?id=80320

Diagramme im Bericht einbinden

Über das entsprechende Kontrollfeld oder **Einfügen → Bericht-Steuerelemente → Diagramm** lässt sich ein Diagramm dem Bericht hinzufügen. Ein Diagramm ist die einzige Möglichkeit, in einem Bericht Daten wiederzugeben, die nicht aus der Quelle stammen, die im Bericht als Datenquelle angegeben ist. Ein Diagramm ist insofern ein Unterbericht des Berichtes, kann aber auch als eigenständiger Berichtsteil gesehen werden.

Hinweis

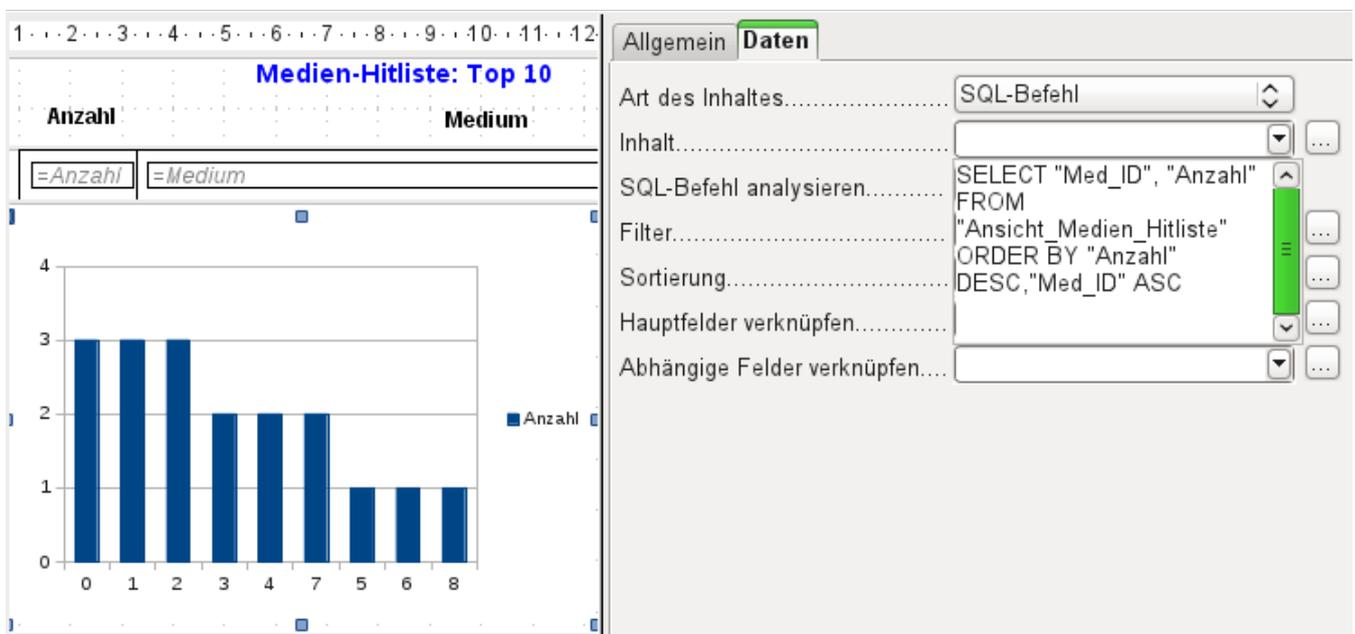
Die Funktionsfähigkeit von Diagrammen ist in Berichten nicht beständig gewährleistet. Von LO 5.3 bis LO 7.3.2 funktionierte kein Diagramm im Report-Designer. Noch schlimmer: Alte Diagramme ließen sich nicht einmal entfernen, da die Auswahl der Diagramme dazu führte, dass LO direkt abstürzt.

Seit der Version LO 7.3.3 werden Diagramme wieder angezeigt. Auch die Absturzgefahr beim Bearbeiten von Diagrammen, die eventuell nicht korrekt funktionieren, ist behoben.



Das Diagramm wird mit der Maus aufgezogen. Bei den allgemeinen Eigenschaften zeigt sich neben bekannten Feldern die Möglichkeit, einen **Diagrammtyp** über den **Button ...** auszuwählen (siehe entsprechende Typen in Calc). Außerdem wird eine maximal für die Vorschau benutzte Anzahl an Datensätzen (**Vorschau für Zeile(n)**) eingestellt, die schon einmal eine Vorstellung davon geben kann, wie das Diagramm letztlich aussieht.

Diagramme können, ebenfalls wie in Calc, entsprechend formatiert werden (Doppelklick auf das Diagramm). Hierzu siehe die Beschreibungen im Handbuch Calc.



Das Diagramm wird im Reiter **Daten** mit den erforderlichen Datenfeldern verbunden. Hier, im Beispielbericht «Medien-Hitliste», soll das Diagramm die Häufigkeit deutlich machen, mit der bestimmte Medien entliehen wurden. Dazu wurde ein SQL-Befehl über den Abfrageeditor wie bei Listefeldern erstellt und eingebunden. Die erste Spalte wird als die angesehen, mit der die Säulen beschriftet werden sollen, die zweite Spalte liefert dann die Anzahl der Entleihvorgänge, die sich in den Höhen der Säulen widerspiegelt.

In dem obigen Beispiel zeigt das Diagramm erst einmal nur sehr wenig an, da die Test-Entleihvorgänge sich zum Zeitpunkt der SQL-Eingabe in Grenzen hielten.

The screenshot shows the 'Übersicht Teilnehmer' report in the Report Designer. The report includes a bar chart with six bars representing age groups: m 12, m 14, m 16, m 20, m 30, and m 50. The y-axis represents the number of participants, ranging from 0 to 60. The chart is titled 'Anzahl'. The 'Daten' tab is active, showing the following settings:

- Art des Inhaltes: Abfrage
- Inhalt: Anzahl_Altersgruppe_Geschl
- SQL-Befehl analysieren: Ja
- Filter: (empty)
- Sortierung: (empty)
- Hauptfelder verknüpfen: "Geschlecht"
- Abhängige Felder verknüpfen: "Geschlecht"

In den Dateneigenschaften des Diagramms ist hier eine **Abfrage** eingebunden. Dieses Diagramm aus der Datenbank «Beispiel_Sport.odt»¹ zeigt neben den Grundlagen für die Erstellung von Diagrammen in Berichten eine Besonderheit auf: Die Vorschau des Diagramms zeigt mehr Säulen an, als an sich für das Diagramm vorgesehen sind. Dies liegt an der Zusammensetzung der Abfrage, die gleichzeitig mehrere Spalten darstellt, die eben nicht alle im Diagramm selbst dargestellt werden sollen.

Eine Filterung und Sortierung mit den internen Werkzeugen des Report-Designers ist nicht notwendig, weil dies eben schon so weit wie möglich über die Abfrage erledigt wurde.

Tipp

Grundsätzlich sollte die Berichtserstellung von so viel Aufgaben wie möglich entlastet werden. Was vorher durch Abfragen bewerkstelligt werden kann, das muss nachher nicht mehr die im Verhältnis zur Datenbank doch recht träge Ausführung des Berichtes belasten.

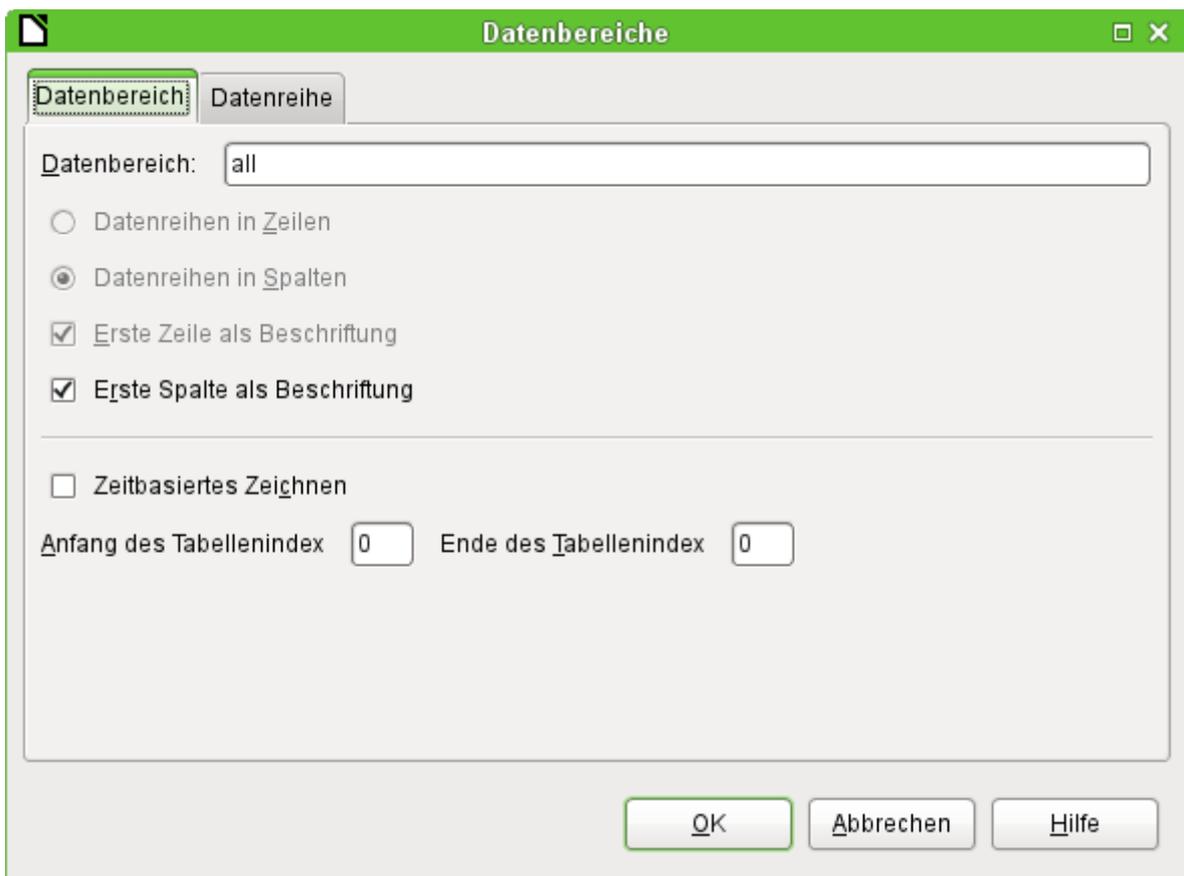
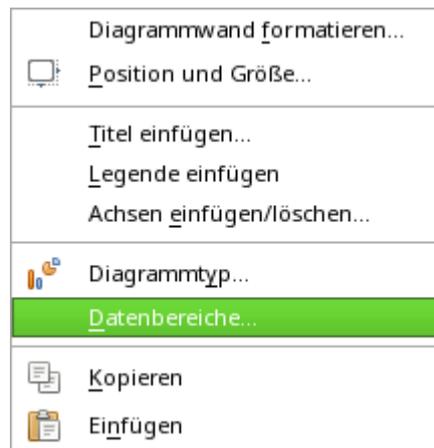
Wie bei Hauptformularen und Unterformularen werden jetzt Felder verknüpft. Im eigentlichen Bericht werden tabellarisch die Altersgruppen für männliche und weibliche Teilnehmer des Sportwettkampfs gelistet. Dabei ist nach den Geschlechtern gruppiert worden. In jeder Gruppe erscheint jetzt ein gesondertes Diagramm. Damit das Diagramm nur die zu dem Geschlecht gehörigen Daten übermittelt, sind die Felder "Geschlecht" aus dem Bericht und "Geschlecht" aus dem Diagramm miteinander verbunden.

Hinweis

Haben Diagramme die gleiche Datenquelle (Tabelle, Abfrage) wie der Bericht selbst, so funktioniert die Verknüpfung von Hauptfeldern und abhängigen Feldern nicht. Hier muss für eine funktionierende Verknüpfung eine zweite Abfrage gleichen Inhaltes erstellt oder der SQL-Code direkt als Datenquelle für das Diagramm eingegeben werden.

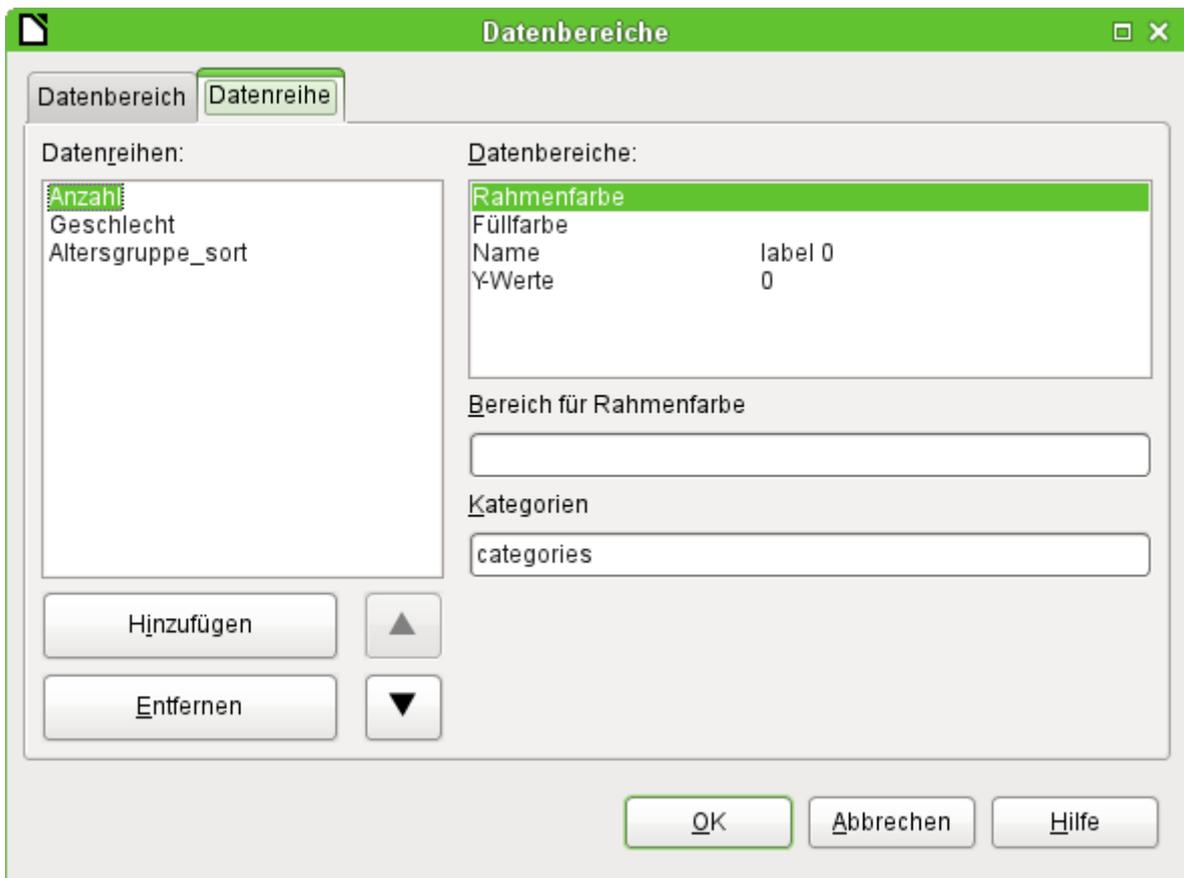
Die x-Achse des Diagramms wird automatisch erst einmal mit der ersten Tabellenspalte der Datenquelle verbunden. Bei mehr als zwei Tabellenspalten stellt die Automatik gleich mehrere Säulen in dem Diagramm dar. Weitere Einstellungen zu dem Diagramm sind zu erreichen, wenn das gesamte Diagramm mit einem Doppelklick markiert wird. Mit einem Klick der rechten Maustaste öffnet sich über dem Diagramm, je nach markiertem Element, ein Kontextmenü. Hierin enthalten ist immer die Einstellungsmöglichkeit für die Datenbereiche:

¹ Die Datenbank «Beispiel_Sport.odt» ist den Beispieldatenbanken für dieses Handbuch beigelegt.



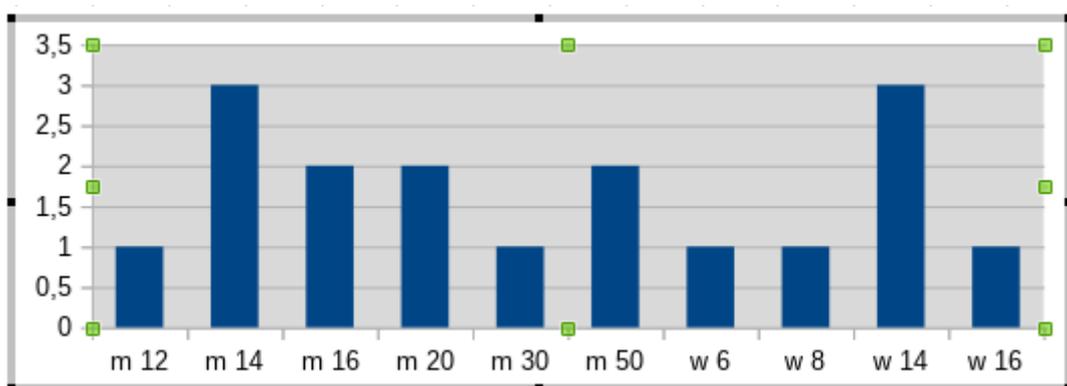
Datenreihen in Spalten ist ausgegraut, also nicht änderbar. Auch eine Änderung des Markierfeldes **Erste Spalte als Beschriftung** ist nicht möglich. An den weiteren Einstellungen des Datenbereichs sollten tunlichst keine Änderungen vorgenommen werden, da dieser Dialog mehr Möglichkeiten offeriert, als tatsächlich für den Report-Designer sinnvoll machbar sind.

Der Reiter **Datenreihen** hingegen verbirgt ein paar Einstellungsmöglichkeiten, die die Standarddarstellung deutlich verändern können. Es werden dort alle Datenreihen angeboten, die neben der ersten Spalte der Abfrage noch verfügbar sind. Datenreihen, die nicht angezeigt werden sollen, können hier entfernt werden.



In dem obigen Beispiel waren zu viele Säulen in dem Diagramm sichtbar. Hier muss also nachgebessert werden. Weder die Datenreihe «Geschlecht» noch die Datenreihe «Altersgruppe_sort», deren Bezeichnungen jeweils mit denen in der zugrundeliegenden Abfrage überein stimmen, ergeben hier einen Sinn. Die Datenreihe «Geschlecht» dient schließlich zur Verbindung des Diagramms mit der Datengrundlage des Berichts und lässt sich numerisch überhaupt nicht darstellen. Die Datenreihe «Altersgruppe_sort» dient nur zur Sortierung der Werte der Abfrage, da sonst Bezeichnungen wie «m 8» nicht am Anfang des Diagramms sondern direkt vor «m 80» einsortiert würden – die Sortierung eines Textes führt eben manchmal zu unerwünschten Ergebnissen.

Nachdem bis auf die Datenreihe «Anzahl» alle Datenreihen entfernt wurden, sieht die Vorschau des Diagramms so aus:



Die Vorschau zeigt hier 10 Säulen auf – die ersten 10 Säulen aus der Abfrage. In der Ausführung werden allerdings anschließend nur die Säulen dargestellt, die zu der entsprechenden Gruppe passen: 'm' für «männlich» und 'w' für «weiblich».

Die y-Achse weist noch eine ungünstige Achsenteilung auf. Schließlich kann es nicht halbe Personen geben. Hier könnte zwar nachgebessert werden. Allerdings wird bei der automatischen Fassung dieser Einstellungen schnell eine Anpassung zu Ganzzahlen erfolgen, wenn die Werte eben nicht, wie in obigen Beispiel, bereits bei einer Anzahl von '3' enden. Dann wäre erneut ein Nachbessern notwendig, wenn vorher die Automatik ausgeschaltet wurde.

Hinweis

Aufgrund verschiedener Bugs hat die Anzeige eines Diagramms Probleme mit Datumsfeldern und Zeitfeldern. Hier empfiehlt es sich, die Datums- bzw. Zeitangaben in einer Abfrage in Zahlen umzuwandeln. Eine mögliche Abfrage für ein Datumsfeld in der **HSQLDB** wäre:

```
001 SELECT
002 DATEDIFF( 'dd', '1899-12-30', "Datum" ) AS "DatumInteger",
003 "Betrag"
004 FROM "Verkauf"
```

Anschließend muss allerdings die entsprechende Achse des Diagramms bearbeitet werden. Die automatische Erkennung aus der Datenquelle erstellt hieraus eine Zahl. Diese automatische Erkennung muss abgewählt und stattdessen das Datum angewählt werden.

Alle weiteren Einstellungen sind ähnlich den Möglichkeiten, die LO-Calc für die Einstellung von Diagrammen bietet.

Dateneigenschaften von Feldern



In den Eigenschaften zeigt sich hinter dem Reiter **Daten** standardmäßig nur das Feld der Datenbank, aus dem die Daten für die Felder des Berichtes ausgelesen werden. Allerdings stehen neben dem **Datenfeld-Typ → Feld oder Formel** noch die Typen '*Funktion*', '*Zähler*' und '*Benutzerdefinierte Funktion*' zur Verfügung.

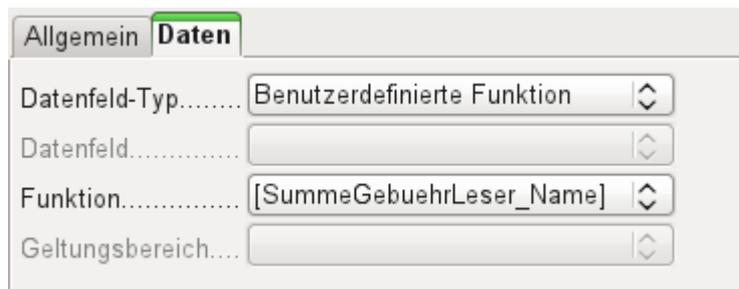
Vorwählbar sind die Funktionen '*Summe*', '*Minimum*' und '*Maximum*'. Ihr Geltungsbereich bezieht sich entweder auf die aktuelle Gruppe oder auf den gesamten Bericht. Bereits diese Funktionen können zu Problemen führen, wenn ein Feld leer, also **NULL**, ist. In solchen Feldern, sofern sie als Zahl formatiert sind, erscheint «N,aN», was wohl so viel wie «kein Zahlenwert» bedeuten soll. Mit leeren Feldern wird keine Rechnung durchgeführt; das Ergebnis ist '0'.

Solche Felder können zwar in der Ansicht durch die folgende Formel im Bereich «Daten» zur Anzeige von 0-Werten umformatiert werden.

```
001 IF([Zahlenfeld];[Zahlenfeld];0)
```

Die Funktion rechnet aber mit dem tatsächlichen Wert, also mit dem Feld, das eben keinen Wert besitzt. Da ist es sinnvoller, wenn gewünscht, die dem Bericht zugrundeliegende Abfrage so zu formulieren, dass statt NULL in Zahlenfeldern '0' wiedergegeben wird.

Der '*Zähler*' zählt einfach nur die Datensätze, die entweder in einer Gruppe oder im ganzen Bericht enthalten sind. Wird der Zähler in den Bereich Detail eingesetzt, so wird durch ihn jeder Datensatz mit einer fortlaufenden Nummer versehen. Dabei kann die Nummerierung auch hier nur die Gruppe oder sämtliche Datensätze des Berichtes fortlaufend nummerieren.



Schließlich stehen noch die detaillierten *Benutzerdefinierte Funktionen* zur Verfügung. Es kann passieren, dass der Report-Designer diese Variante selbst wählt, wenn er zwar den Rechenschritt vorgegeben bekommt, aber die Datenquelle aus irgendeinem Grunde nicht richtig interpretieren kann. Dann können allerdings dort auch weitere Einstellungen vorgenommen werden.

Alle Funktionen sind anschließend auch über den *Bericht-Navigator* zur weiteren Bearbeitung verfügbar. Dort kann dann z.B. auch eine Vorausberechnung von Werten eingestellt werden. Über den Bericht-Navigator können auch eigene benutzerdefinierte Funktionen für jede Gruppe oder den gesamten Bericht erstellt werden.

Funktionen im Report-Designer

Der Report-Designer bietet sowohl bei den Daten als auch bei den Bedingungen für eine Anzeige die Nutzung verschiedenen Funktionen an. Reichen die Funktionen nicht, so lassen sich mit einfachen Rechenschritten auch benutzerdefinierte Funktionen erstellen, die vor allem in Gruppenfüßen als Zusammenfassungen Sinn machen.

Formeleingaben

Dem Report-Designer liegt der «Pentaho Report Designer» zugrunde. Ein kleiner Teil der Dokumentation ist unter <http://wiki.pentaho.com/display/Reporting/9.+Report+Designer+Formula+Expressions> zu finden.

Eine weitere Quelle sind die Spezifikationen nach «OpenFormular-Standard»:

<http://www.oasis-open.org/committees/download.php/16826/openformula-spec-20060221.html>

Teilweise ist auch ein einfacher Blick in Richtung Calc sinnvoll, da dort eine deutsche Beschreibung zumindest einiger Formeln aus dem Report-Designer enthalten ist.

Grundlegend gilt:

Formeleingaben starten mit einem Gleichheitszeichen	=
Bezüge zu Datenfeldern werden in eckige Klammern gesetzt	[Feldbezeichnung]
Enthalten die Datenfelder Sonderzeichen, zu denen auch Leertasten gehören, so ist die Feldbezeichnung außerdem noch in doppelte Anführungszeichen zu setzen	["Diese Feldbezeichnung wäre in doppelte Anführungszeichen zu setzen"]
Texteingaben haben immer in doppelten Anführungszeichen zu stehen	"Texteingabe"
Der Dezimaltrenner ist wie in SQL der Punkt, nicht das Komma	1.27
Folgende Operatoren sind möglich	+ , - , * (Multiplikation), / (Division), % (dividiert vorher stehende Zahl durch 100), ^ (Potenzierung mit der nachfolgenden Zahl), & (verbindet Texte miteinander),
Folgende Beziehungsdefinitionen sind möglich	= , <> , < , <= , > , >=
Runde Klammerungen sind ebenfalls möglich	()
Standardfehlermeldung	NA (vermutlich not available, in Calc in NV übersetzt – nicht verfügbar)
Fehlermeldung beim Bericht, wenn ein leeres Feld auftaucht und als Zahl definiert ist.	N, aN (erscheint, wenn kein Wert enthalten ist – vielleicht not a number?)

Sämtliche Formeleingaben wirken sich nur auf den aktuell eingelesenen Datensatz aus. Beziehungen zu vorhergehenden oder nachfolgenden Datensätzen sind also nicht möglich, wohl aber z. B. die Berechnung einer Summe zwischen Feldern eines Datensatzes. Enthält z. B. ein Datensatz die Felder "Nettobetrag" und "MWSt", so ergibt die folgende Formel den Bruttobetrag:

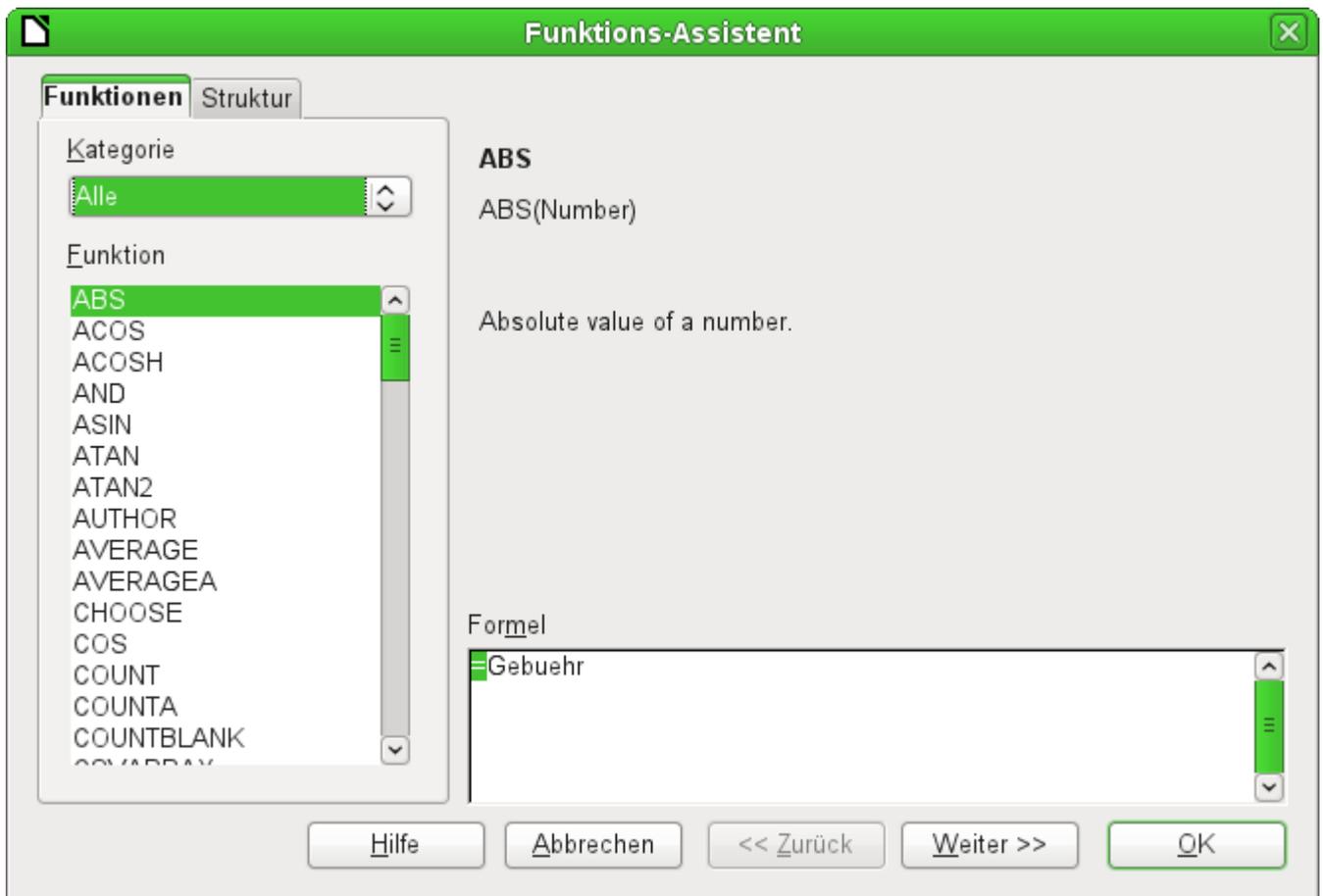
`001` =[Nettobetrag]+[MWSt]

Möglich ist aber auch:

`001` =SUM([Nettobetrag];[MWSt])



Neben dem Datenfeld erscheint der Button `...`, sofern eine Formeleingabe möglich ist. Dieser Button startet den Funktionsassistenten.



Im Gegensatz zu den Funktionen in Calc sind hier die Funktionen nicht ins Deutsche übertragen. Die Funktionen sind längst nicht so umfangreich wie die, die in Calc üblich sind. Viele Funktionen haben allerdings ihre Entsprechung in Calc. Dort gibt der Assistent dann auch direkt das Ergebnis der Funktion wieder.

Der Funktions-Assistent funktioniert nicht immer einwandfrei. So werden z. B. Texteingaben nicht mit doppelten Anführungszeichen übernommen. Nur Eingaben mit doppelten Anführungszeichen werden allerdings verarbeitet, so dass die Anführungszeichen später wieder ergänzt werden müssen.

Die folgenden Funktionen stehen zur Verfügung: [**Funktion im Report-Designer** / (Funktion in Calc)]

Funktion	Beschreibung
Datums - und Zeitfunktionen	
DATE (DATUM)	Erzeugt aus einer Zahlenangabe für das Jahr, den Monat und den Tag ein gültiges Datum.
DATEDIF (TAGE MONATE JAHRE)	Gibt die Anzahl an Jahren, Monaten oder Tagen zwischen zwei Datumswerten wieder.
DATEVALUE (DATWERT)	Wandelt eine amerikanische Datumseingabe in Textform (Anführungsstriche) in eine Datumsbeschreibung um. Die erzeugte amerikanische Variante kann durch Formatierungseinstellung umgewandelt werden.
DAY (TAG)	Gibt den Tag des Monats eines angegebenen Datums wieder. DAY([Datumsfeld])

DAYS (TAGE)	Gibt die Zahl der Tage zwischen zwei Datumseingaben wieder.
HOURL (STUNDE)	Gibt die Stunde einer angegebenen Zeit im Rahmen von 0 bis 23 wieder. HOUR([DatumZeitFeld]) gibt die Stunde des Feldes wieder.
MINUTE (MINUTE)	Gibt die Minuten einer internen Zahl wieder. MINUTE([Zeitfeld]) gibt den Minutenanteil der Zeit wieder.
MONTH (MONAT)	Gibt den Monat einer Datumseingabe als Zahl wieder. MONTH([Datumsfeld])
NOW (JETZT)	Gibt das aktuelle Datum und die aktuelle Zeit wieder.
SECOND (SEKUNDE)	Gibt die Sekunden einer internen Zahl wieder. SECOND(NOW()) gibt den Sekundenanteil der Zeit beim Ausführen des Berichts wieder.
TIME (ZEIT)	Zeigt die aktuelle Zeit an.
TIMEVALUE (ZEITWERT)	Wandelt die Texteingabe einer Zeit in eine Zeit für Berechnungen um
TODAY (HEUTE)	Gibt das aktuelle Datum wieder
WEEKDAY (WOCHENTAG)	Gibt den Wochentag einer Datumseingabe als Zahl wieder. Der Tag mit der Nummer 1 ist der Sonntag.
YEAR (JAHR)	Gibt das Jahr einer Datumseingabe wieder.
Logische Funktionen	
AND (UND)	Gibt TRUE wieder, wenn alle Argumente TRUE sind
FALSE (FALSCH)	Definiert den logischen Wert als FALSE
IF (WENN)	Wenn eine Bedingung TRUE, dann diesen Wert, ansonsten einen anderen Wert.
IFNA	Wenn der Wert nicht verfügbar ist, dann setze diesen Wert. Ist ähnlich der Anweisung IFNULL in SQL. Vor allem bei Funktionen sinnvoll, die Feldwerte addieren sollen, bei denen aber nicht klar ist, ob jedes Feld einen Wert hat. IFNA([Feldwert];0) rechnet dann trotzdem z.B. die Summe aus. (ab LO 3.5)
NOT (NICHT)	Kehrt den logischen Wert eines Argumentes um
OR (ODER)	Gibt TRUE zurück, wenn eine der Bedingungen TRUE ist.
TRUE (WAHR)	Definiert den logischen Wert als TRUE
XOR	Nur wenn ein einziger Wert der Verknüpfung TRUE ist, ist auch der logische Wert TRUE.
Rundungsfunktionen	
INT	Rundet zum nächsten Ganzzahlwert ab

Mathematische Funktionen	
ABS (ABS)	Gibt den absoluten, nicht negativen Wert einer Zahl wieder.
ACOS (ARCCOS)	Berechnet den Arcuscosinus einer Zahl. - Werteingabe zwischen -1 und 1 (ab LO 3.5)
ACOSH (ARCCOSHYP)	Berechnet den Areacosinus (inverser hyperbolischer Cosinus) - Werteingabe ≥ 1 (ab LO 3.5)
ASIN (ARCSIN)	Berechnet den Arcussinus einer Zahl. - Werteingabe zwischen -1 und 1 (ab LO 3.5)
ATAN (ARCTAN)	Berechnet den Arcustangens einer Zahl. (ab LO 3.5)
ATAN2 (ARCTAN2)	Berechnet den Arcustangens einer x-Koordinate und einer y-Koordinate. (ab LO 3.5)
AVERAGE (MITTELWERT)	Ermittelt den Mittelwert der angegebenen Werte (taucht im Formularassistenten LO 3.3.4 doppelt auf).
AVERAGEA (MITTELWERTA)	Ermittelt den Mittelwert der angegebenen Werte. Text wird als 0 gewertet. (ab LO 3.5)
COS (COS)	Winkel im Bogenmaß, dessen Cosinus berechnet werden soll. (ab LO 3.5)
EVEN (GERADE)	Rundet eine positive Zahl zum nächsten geraden Integer auf, eine negative Zahl zum nächsten geraden Integer ab.
EXP (EXP)	Berechnet die Exponentialfunktion zur Basis 'e' (ab LO 3.5)
LN (LN)	Berechnet den natürlichen Logarithmus einer Zahl. (ab LO 3.5)
LOG10 (LOG10)	Berechnet den Logarithmus einer Zahl zur Basis '10'. (ab LO 3.5)
MAX (MAX)	Gibt den Maximalwert aus einer Reihe von Werten wieder.
MAXA (MAXA)	Gibt den Maximalwert aus einer Reihe wieder. Eventuell vorhandener Text wird als '0' gesetzt.
MIN (MIN)	Gibt den Minimalwert aus einer Reihe von Werten wieder.
MINA (MINA)	Gibt den Minimalwert aus einer Reihe wieder. Eventuell vorhandener Text wird als '0' gesetzt.
MOD (REST)	Gibt den Rest einer Division nach Eingabe von Dividend und Divisor wieder
ODD (UNGERADE)	Rundet eine positive Zahl zum nächsten ungeraden Integer auf, eine negative Zahl zum nächsten ungeraden Integer ab.
PI (PI)	Liefert den Wert der Zahl ' π ' (ab LO 3.5)

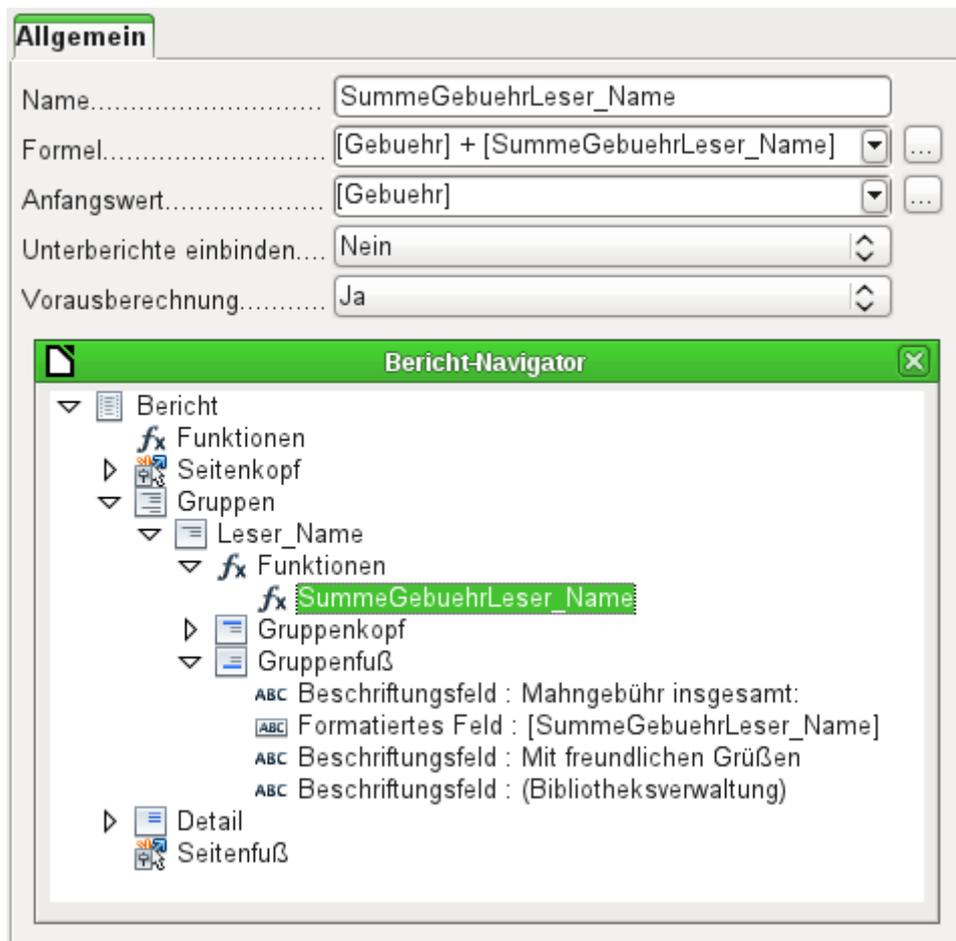
POWER (<i>POTENZ</i>)	Liefert aus Basis und Exponent die Potenz. (ab LO 3.5)
SIN (<i>SIN</i>)	Berechnet den Sinus einer Zahl. (ab LO 3.5)
SQRT (<i>WURZEL</i>)	Berechnet die Quadratwurzel einer Zahl. (ab LO 3.5)
SUM (<i>SUMME</i>)	Summiert eine Liste von Zahlenwerten
SUMA	Summiert eine Liste von Zahlenwerten. Text und Ja/Nein-Felder sind erlaubt. Leider endet diese Funktion (noch) mit Fehlermeldung. (ab LO 3.5)
VAR (<i>VARIANZ</i>)	Berechnet die Varianz, ausgehend von einer Stichprobe. (ab LO 3.5)
Textfunktionen	
EXACT (<i>IDENTISCH</i>)	Zeigt an, ob zwei Texte völlig gleich sind.
FIND (<i>FINDEN</i>)	Gibt die Position eines zu suchenden Textes in einem anderen Text an.
LEFT (<i>LINKS</i>)	Der Text wird von links aus in der angegebenen Zeichenzahl wiedergegeben.
LEN (<i>LÄNGE</i>)	Gibt die Anzahl an Zeichen wieder, die ein Text hat.
LOWER (<i>KLEIN</i>)	Gibt den Text in Kleinbuchstaben wieder.
MESSAGE	Formatiert die Werte in dem angegebenen Ausgabeformat. (ab LO 3.5)
MID (<i>TEIL</i>)	Gibt Text ab einer Anfangsposition mit einer gegebenen Zeichenzahl wieder.
REPLACE (<i>ERSETZEN</i>)	In dem Text wird ein Teil durch einen anderen Text ersetzt. Die Startposition und die Länge des zu ersetzenden Textes wird eingegeben.
REPT (<i>WIEDERHOLEN</i>)	Wiederholt einen Text die angegebenen Male.
RIGHT (<i>RECHTS</i>)	Der Text wird von rechts aus in der angegebenen Zeichenzahl wiedergegeben.
SUBSTITUTE (<i>WECHSELN</i>)	Ersetzt in einem vorgegebenen Text bestimmte Textteile durch andere Textteile. Zusätzlich kann angegeben werden, der wievielte Textteil ersetzt werden soll.
T (<i>T</i>)	Gibt den Text wieder oder einen leeren Textwert, wenn es sich z. B. um eine Zahl handelt.
TEXT (<i>TEXT</i>)	Konvertierung von Zahlen oder Uhrzeiten in Text.
TRIM (<i>GLÄTTEN</i>)	Entfernt führende Leerzeichen, Leerzeichen am Textende und reduziert mehrere Leerzeichen hintereinander im Text auf ein Leerzeichen.

UNICHAR (UNIZEICHEN)	Wandelt eine Unicode-Nummer in Dezimalschreibweise in ein Unicode-Zeichen um 196 wird zu 'Ä' ('Ä' hat den Hexadezimalwert 00C4, daraus wird bei der Dezimalschreibweise 196 ohne führende Nullen)
UNICODE (UNICODE)	Wandelt eine Unicode-Zeichen in eine Unicode-Nummer in Dezimalschreibweise um 'Ä' wird zu 196
UPPER (GROSS)	Gibt den Text in Großbuchstaben wieder.
URLENCODE	Wandelt einen vorgegebenen Text in einen Text um, der URL-Konform ist. Wenn keine besonderer Code vorgegeben wurde, pas-siert dies nach ISO-8859-1.
Informationsfunktionen	
CHOOSE (Wahl)	Zuerst wird ein Index angegeben, danach eine Liste von Werten, aus der über den Index der entsprechende Wert ausgesucht wird. CHOOSE(2;"Apfel";"Birne";"Banane") gibt Birne wieder. CHOOSE([Altersstufenfeld];"Milch";"Cola";"Bier") gibt je nach "Altersstufenfeld" aus der Datenquelle wieder, welches Getränk möglich ist.
COUNT (Anzahl)	Nur die Felder werden gezählt, die eine Zahl oder einen Datums- bzw. Zeitwert enthalten. COUNT([Zeit];[Nummer]) gibt 2 aus, wenn beide Felder einen Wert enthalten, also nicht NULL sind, ansonsten 1 oder schließlich 0.
COUNTA (Anzahl2)	Berücksichtigt auch die Felder, die einen Text beinhalten. Auch NULL wird gezählt, ebenso boolsche Felder.
COUNTBLANK (ANZAHLLEEREZELLEN)	Zählt die leeren Felder in einem Bereich.
HASCHANGED	Überprüft, ob die mit ihrem Namen angegebene Spalte sich geändert hat. Allerdings werden keine Spaltenangaben übernommen.
INDEX (INDEX)	Arbeitet mit Bereichen (ab LO 3.5)
ISBLANK (ISTLEER)	Prüft, ob das Feld NULL (leer) ist.
ISERR (ISTFEHL)	Gibt TRUE zurück, wenn die Eingabe fehlerhaft ist, aber nicht vom Typ NA ISERR(1/0) ergibt TRUE
ISERROR (ISTFEHLER)	Wie ISERR, nur dass auch NA als Meldung TRUE ergibt.
ISEVEN (ISTGERADE)	Prüft, ob es sich um eine gerade Zahl handelt.
ISLOGICAL (ISTLOG)	Prüft, ob es sich um einen Ja/Nein-Wert handelt. ISLOGICAL(TRUE()) oder ISLOGICAL(FALSE()) ergeben TRUE , Textwerte wie ISLOGICAL("TRUE") ergeben FALSE .
ISNA (ISTNV)	Prüft, ob der Ausdruck vom Fehlertyp NA ist.
ISNONTEXT (ISTKTEXT)	Prüft, ob es sich bei dem Wert nicht um einen Text handelt.

ISNUMBER (ISTZAHL)	Prüft, ob es sich um eine Zahl handelt. ISNUMBER(1) ergibt TRUE , ISNUMBER("1") ergibt FALSE
ISODD (ISTUNGERADE)	Prüft, ob es sich um eine ungerade Zahl handelt.
ISREF (ISTBEZUG)	Prüft, ob es sich um einen Bezug handelt. ISREF([Feldname]) ergibt TRUE , ISREF(1) ergibt FALSE .
ISTEXT (ISTTEXT)	Prüft, ob es sich bei dem Wert um einen Text handelt.
NA (NV)	Gibt den Fehlercode NA wieder.
VALUE	(ab LO 3.5)
Benutzerdefiniert	
CSVARRAY	Wandelt einen CSV-Text in ein Array um. (ab LO 3.5)
CSVTEXT	Wandelt ein Array in einen CSV-Text um. (ab LO 3.5)
NORMALIZEARRAY	(ab LO 3.5)
NULL	Gibt NULL wieder
PARSEDATE	Wandelt Text in ein Datum um. Nutzt das SimpleDateFormat. Benötigt ein Datum als Text sowie die Beschreibung des Datumformates. Beispiel: PARSEDATE("9.10.2012";"dd.MM.yyyy") ergibt die intern verwertbare Zahl für das Datum. (ab LO 3.5)
Dokumentsinformationen	
AUTHOR	Autor, wird ausgelesen aus Extras → Optionen → LibreOffice → Benutzerdaten . Der eigentliche Autor wird hier also nicht wiedergegeben, sondern der aktuelle Nutzer der Datenbank.
TITLE	Gibt den Titel des Berichts wieder.

Benutzerdefinierte Funktionen

Benutzerdefinierte Funktionen können z. B. dazu dienen, bestimmte Zwischenergebnisse nach einer Gruppe von Datensätzen wieder zu geben. Im obigen Beispiel ist so etwas über die vordefinierte Funktion der Summe bei der Berechnung der Gebühr im Bereich «Leser_Name_Fuß» automatisch erstellt worden.



Im Bericht-Navigator ist für die Gruppe «*Leser_Name*» eine Funktion verzeichnet. Durch Rechtsklick auf Funktionen können zusätzliche Funktionen hier mit Namen definiert werden.

Die Funktion «*SummeGebuehrLeser_Name*» wird in den Eigenschaften angezeigt. Die **Formel** führt eine Addition des Feldes «*Gebuehr*» mit dem in der Funktion selbst bereits gespeicherten Wert durch. Der **Anfangswert** ergibt sich aus dem Wert des Feldes «*Gebuehr*» beim ersten Durchgang durch die Gruppe. Dieser Wert wird in der Funktion unter dem Funktionsnamen zwischengespeichert und in der Formel wieder benutzt, bis die Schleife beendet ist und der Gruppenfuß geschrieben wird.

Unterberichte einbinden scheint momentan keine Funktion zu haben, es sei denn, dass Diagramme als Unterberichte verstanden werden.

Ist für die Funktion **Vorausberechnung** aktiviert, so kann das Ergebnis auch im Gruppenkopf stehen. Ohne die Aktivierung steht im Gruppenkopf nur der entsprechende Wert des abgefragten ersten Feldes der Gruppe.

Selbstdefinierte Funktionen können auch auf selbstdefinierte Funktionen zurückgreifen. Dabei ist dann aber zu beachten, dass die genutzten Funktionen vorher ausgeführt werden müssen. Wird die Vorausberechnung in dieser Funktion, die auf andere Funktionen zurückgreift, ausgeschaltet, so lässt sich mit der folgenden Kombination im Detail-Bereich der Zensurenschnitt nach jedem Datensatz berechnen.

001 [SummeZensurKlasse] / [ZählerKlasse]

Wird die Kombination in den Gruppenfuß der Gruppe «*Klasse*» gesetzt, so erscheint dort der Durchschnitt für die gesamte Klasse. Im Gruppenkopf erscheint hingegen als Durchschnitt nur die Zensur des ersten Schülers, sofern die Vorausberechnung für **SummeZensurKlasse** ausgeschaltet wurde. Mit eingeschalteter Vorausberechnung ist auch im Gruppenkopf der korrekte Wert zu erreichen.

Funktionen wie «ZählerKlasse» zeigen ihren Inhalt nur in der Gruppe an. Werden sie im Seitenkopf oder Seitenfuß genutzt, so bleibt der Inhalt leer.

Formeleingabe für ein Feld

Über den Weg **Daten** → **Datenfeld** können Formeln eingegeben werden, die nur ein einziges Feld im Bereich «Detail» betreffen.

```
001 IF([boolschesFeld];"ja";"nein")
```

schreibt, dort eingegeben, statt WAHR und FALSCH einfach "ja" und "nein".

Ein boolsches Feld hat ja auch die Möglichkeit, neben "ja" und "nein" ein leeres Feld (**NULL**) zu präsentieren. Hier ein Code, der das Ganze gleich in entsprechende Feldform bringt:

```
001 IF([boolschesFeld];"☒";IF(ISBLANK([boolschesFeld]);"-";"☐"))
```

Es kann passieren, dass in einem Feld mit einer Formeleingabe grundsätzlich eine Zahl erscheint. Bei Text ist das dann eine «0». Hier muss nachgebessert werden, indem für das Textfeld vom Standardformat «Zahl» zum Format «Text» gewechselt wird.

Bedingte Anzeige



Gruppenköpfe, Gruppenfüße, Felder – in sämtlichen Untergliederungen befindet sich unter den allgemeinen Eigenschaften das Feld **Ausdruck für bedingte Anzeige**. Formeln, die in dieses Feld geschrieben werden, beeinflussen den Inhalt eines Feldes oder gleich die Anzeige eines ganzen Bereiches. Auch hier steht der Funktions-Assistent zur Verfügung.

```
001 [Feldbezeichnung]="true"
```

sorgt dafür, dass der Inhalt nur dann angezeigt wird, wenn der Inhalt des Feldes «Feldbezeichnung» wahr ist.² Dies funktioniert auch bei grafischen Steuerelementen.

Grafische Formen reagieren zur Zeit nicht auf Einträge in der bedingten Anzeige. Soll z. B. eine farbige Trennlinie nach dem 10. Platz einer Wettkampfliste eingezogen werden, so geht dies nicht, indem der grafischen Form über die bedingte Anzeige mitgegeben wird

```
001 [Platz]=10
```

Dieser Befehl wirkt nicht auf die Grafik. Sie erscheint in dem Abschnitt «Detail» dann weiter nach jedem Datensatz.

Dieser Bug ist als [Bug 73707](#) gemeldet.

Soll lediglich eine rechteckige Form bedingt an dieser Stelle eingeblendet werden, so geht dies aber über ein grafische Steuerelement, das mit einer entsprechenden (eventuell einfarbigen) Grafikdatei angesprochen wird. Bei den allgemeinen Eigenschaften wird **Skalieren** → **Autom. Größe** gewählt. Dann passt sich die Grafik der Form an und die Bedingung greift.

Sicherer ist es, die bedingte Anzeige an einen *Gruppenfuß* statt an die Grafik zu binden, sofern dieser nicht anderweitig benötigt wird. Die Linie wird im *Gruppenfuß* positioniert. Der *Gruppenfuß* wird mit der Bedingung versehen. Dann erscheint die Linie auch tatsächlich nach dem 10. Platz, wenn sie wie oben formuliert wird. Dadurch, dass der Gruppenfuß mit der Bedingung versehen wird, nimmt die Linie auch nur dann den entsprechenden Platz ein, wenn sie wirklich benötigt wird. Sonst erfolgt durch die bedingte Anzeige die Ausgabe eines Leerraumes statt der Linie.

Hier lassen sich dann auch neben den Formen aus **Einfügen** → **Formen** → **Standardformen** z. B. die horizontalen Linien mit dem Gruppenfuß ausblenden.

² Siehe hierzu die Datenbank «Beispiel_Bericht_bedingte_Einblendung_von_Grafiken.odb», die diesem Handbuch beiliegt.

Hinweis

Manchmal scheint es wünschenswert, eine bedingte Anzeige in Abhängigkeit von der Seitenzahl zu erstellen. Die Funktionen PageNumber() und PageCount() können für irgendwelche Formatierungshinweise zur bedingten Anzeige allerdings nicht genutzt werden, da die Werte erst innerhalb des Writer-Dokumentes erstellt werden.

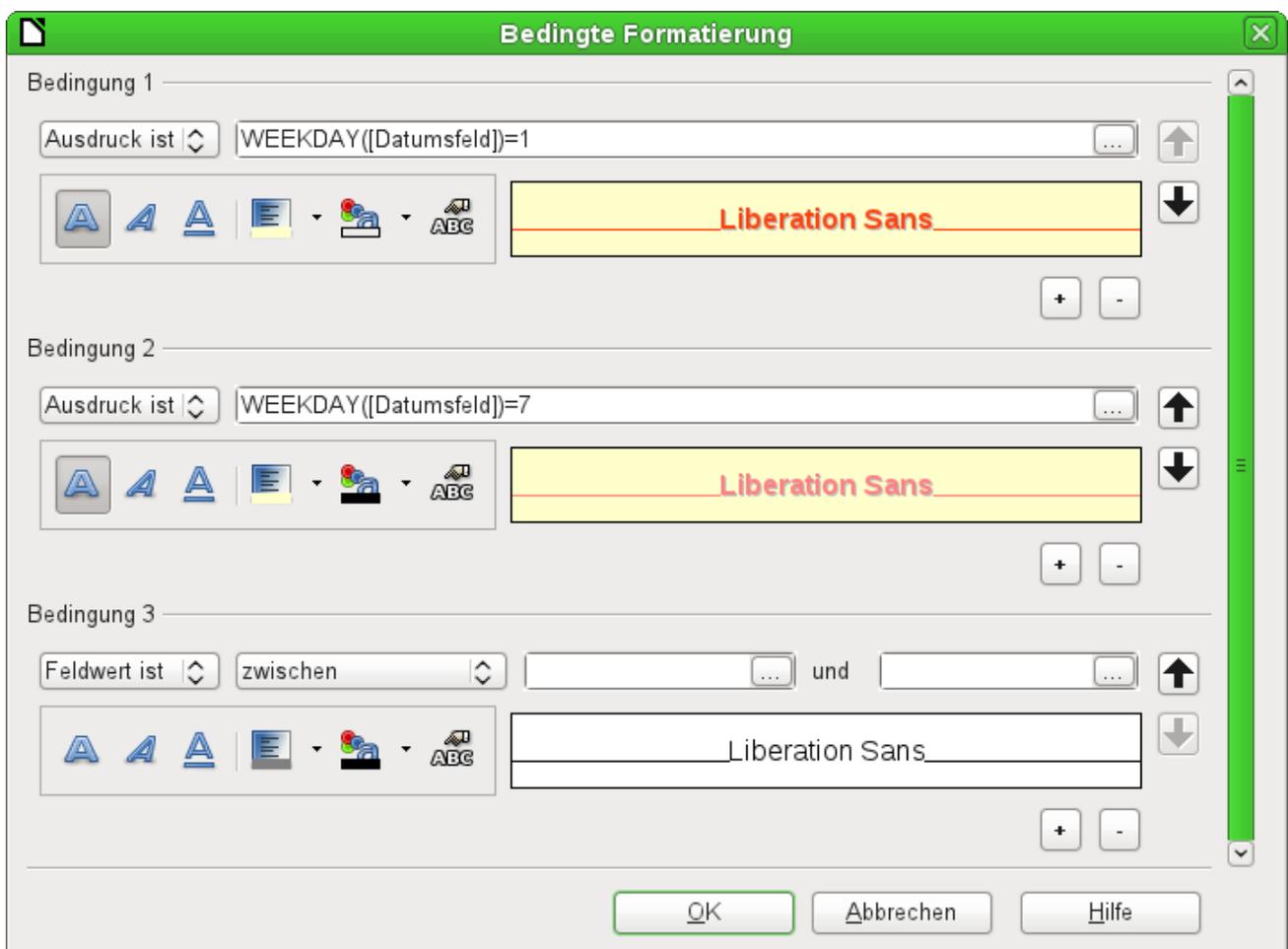
Es scheint momentan überhaupt keine Möglichkeit zu geben, das Erscheinen des Seitenkopfes oder Seitenfußes über die dort mögliche bedingte Anzeige zu steuern.

Bedingte Formatierung

Bei der bedingten Formatierung kann z. B. ein Kalender so formatiert werden, dass die Wochenenden besonders gekennzeichnet werden. Unter **Format → bedingte Formatierung** ist dann einzutragen

```
001 WEEKDAY([Datumsfeld])=1
```

sowie die entsprechende Formatierung für den Sonntag.



Wird in der bedingten Formatierung «Ausdruck ist» gewählt, so kann eine Formel eingegeben werden. Wie auch in Calc üblich können mehrere Bedingungen formuliert werden, die nacheinander abgearbeitet werden. Im obigen Beispiel wird so zuerst der Sonntag abgefragt und dann der Samstag. Zum Schluss könnte dann noch eine Abfrage nach dem Inhalt des Feldes kommen. So könnte z. B. der Inhalt 'Urlaub' mit einer entsprechend anderen Formatierung angezeigt werden.

Hinweis

Der Report-Designer ist ein Addon. Tauchen anscheinend nicht behebbare Fehler auf (Formel wird nicht umgesetzt, zu langer Text wird als ein leeres Feld angezeigt ...), so empfiehlt es sich manchmal, Teile des Berichts zu löschen oder einfach den Bericht neu zu erstellen.

Beispiele für Berichte mit dem Report-Designer

Der Report-Designer birgt einige Tücken in der Anwendung, da bestimmte Funktionen zwar grundsätzlich vorgesehen sind, aber zur Zeit nicht richtig funktionieren. Auch gibt es eine nur sehr geringe Hilfestellung innerhalb der Hilfefunktion von LibreOffice. Deshalb hier einige recht komplexe Beispiele, wie der Report-Designer für verschiedene Berichtstypen genutzt werden kann. Zum Einstieg in den Report-Designer sei hier noch auf das Einführungskapitel dieses Handbuches verwiesen. Dort wird auf mehreren Seiten Schritt für Schritt ein Bericht erstellt.

Rechnungserstellung

Folgende Vorgaben sollen für die Erstellung der Rechnung³ erfüllt werden:

- Die einzelnen Rechnungspositionen sollen durchnummeriert werden.
- Rechnungen, die mehr als eine Seite umfassen, sollen mit einer Seitennummerierung versehen werden.
- Rechnungen, die mehr als eine Seite umfassen, sollen auf jeder Seite eine Zwischensumme bilden und auf der darauffolgenden Seite diese Zwischensumme als Übertrag darstellen.

Mehrere aktuelle existierende Bugs scheinen das Verfahren unmöglich zu machen:

- *Bug 51452*: Wird eine Gruppierung auf «Bereich wiederholen» eingestellt, so wird automatisch vor und hinter der Gruppierung ein Seitenumbruch eingefügt.
- *Bug 51453*: Gruppierungen mit neu startender Seitenzählung sind zwar eigentlich vorgesehen, funktionieren aber nicht.
- *Bug 51959*: Ein Gruppenfuß lässt sich nicht wiederholen. Er kann nur am Ende einer Gruppe auftauchen, nicht z. B. am Ende einer jeden Seite. Wird er auf «Bereich wiederholen» gesetzt, so verschwindet er komplett.

Außerdem gibt es noch Schwierigkeiten, Linien in den Bericht einzufügen. Die vorgesehenen horizontalen und vertikalen Linien werden erst ab den Versionen LO 4.0.5 bzw. 4.1.1 angezeigt. Als Ersatz dafür können Rechtecke genutzt werden. Die lassen sich aber nicht richtig positionieren, wenn in einem Bereich ein Seitenumbruch stattfindet.

Der Bericht in einer einfachen Form sollte also so aussehen:

³ Die Datenbank «Beispiel_Bericht_Rechnung.odt» ist den Beispieldatenbanken für dieses Handbuch beigefügt. Eine komplette Beschreibung der Datenbank ist in «Base_Beispiele.pdf» verfügbar.

Büroshop

Norderstr. 17, 43219 Phantastica

Rechnungsnummer: 2013-0
Datum: 11.03.2013

Anzahl	Ware	Preis	Anzahl*Preis
2	Papier, 500 Blatt	4,23€	8,46 €
1	Radiergummi	0,75€	0,75 €
4	Ansitzer	1,27€	5,08 €
2	Bleistif HB	0,23€	0,46 €
1	Collegblock	0,98€	0,98 €
1	Briefumschläge, 25 Stck.	1,25€	1,25 €
4	CD-Hüllen, 100 Stck.	1,89€	7,56 €
1	Wachsmalkreiden, 6 Stck.	3,85€	3,85 €
1	Ordner, 5cm Rückenbreite	1,89€	1,89 €
2	Klembrett	4,45€	8,90 €
1	Ringbuch A4	5,76€	5,76 €
1	CD-Beschriftungsstifte, 4 Stck.	4,56€	4,56 €
2	CD-Rohlinge, 50 Stck.	12,34€	24,68 €
1	Papier, Recycling, 500 Blatt	3,76€	3,76 €
4	Ordnungsmappe mit Register	6,87€	27,48 €
2	Outtingblock, 50 Blatt	1,35€	2,70 €
10	Rechnungsbuch, 50 Blatt	3,15€	31,50 €
1	Datumsstempel, einfach	18,50€	18,50 €
1	Geldkassette, klein	12,00€	12,00 €
12	Hängemappen, 25 Stck.	14,75€	177,00 €
2	Universalketten 70x32, 2700 Stck.	20,50€	41,00 €
1	Universalketten Klempack 12x30, 700 Stck.	4,15€	4,15 €
2	Druckerköpfe, schwarz	24,00€	48,00 €
1	Druckerköpfe, color	26,30€	26,30 €
3	Druckertintenpatronen, schwarz, 32ml	5,40€	16,20 €
5	Druckertintenpatronen, color, 17ml	5,20€	26,00 €
		Übertrag:	508,77 €

Seite 1

Kontoverbindung: Stadtparkasse Phantastica - BLZ 123 456 01 - Konto 1020384756

Büroshop

Norderstr. 17, 43219 Phantastica

Seite 2

Anzahl	Ware	Preis	Anzahl*Preis
		Übertrag:	508,77 €
2	Toner Laserdrucker schwarz	65,89 €	131,78 €
1	Toner, Laserdrucker, color	78,89 €	78,89 €
2	Register für Ordner, A-Z	1,25 €	2,50 €
1	Heftstreifen, 25 Stck.	0,85 €	0,85 €
2	Schnellhefter Recyclingkarton	0,65 €	1,30 €
1	Papier, 500 Blatt, Recycling	4,15 €	4,15 €
1	Bleistifte, 10 Stck., versch. Stärken	4,85 €	4,85 €
2	Kugelschreiber	1,35 €	2,70 €
1	Wasserfarbkasten, 12 Farben	8,75 €	8,75 €
1	Aquarellkasten, 24 Farben	17,15 €	17,15 €
2	Aquarellpinsel, 3 Stck., versch. Stärken	8,34 €	16,68 €
1	Zeichenblock, A3, 20 Blatt	3,85 €	3,85 €
4	Schreibunterlage 50*70 cm	15,67 €	62,68 €
2	Tonpapier, div. Farben, 50*70 cm	0,45 €	0,90 €
10	Tonkarton, div. Farben, 50*70 cm	0,89 €	8,90 €
1	Datumsstempel, einfach	18,50 €	18,50 €
1	Geldkassette, klein	12,00 €	12,00 €
12	Hängemappen, 25 Stck.	14,75 €	177,00 €
2	Universalketten 70x32, 2700 Stck.	20,50 €	41,00 €
1	Universalketten Klempack 12x30, 700 Stck.	4,15 €	4,15 €
2	Druckerköpfe, schwarz	24,00 €	48,00 €
1	Druckerköpfe, color	26,30 €	26,30 €
3	Druckertintenpatronen, schwarz, 32ml	5,40 €	16,20 €
5	Druckertintenpatronen, color, 17ml	5,20 €	26,00 €
2	Toner Laserdrucker schwarz	65,89 €	131,78 €
1	Toner, Laserdrucker, color	78,89 €	78,89 €
		Übertrag:	1.434,52 €

Seite 2

Kontoverbindung: Stadtparkasse Phantastica - BLZ 123 456 01 - Konto 1020384756

Um trotz der oben genannten Einschränkungen eine den Anforderungen entsprechende Rechnungserstellung durchführen zu können, muss genau mit den Seitenmaßen des auszudruckenden Dokumentes gearbeitet werden. In diesem Beispiel wird von einem DIN-A4-Ausgabeformat ausgegangen. Die Gesamthöhe eines Blattes beträgt also 29,7 cm.

Der Bericht muss dazu in mehrere Gruppen unterteilt werden. Zwei Gruppen beziehen sich auf das gleiche Tabellenfeld und enthalten jeweils nur einen entsprechenden Tabellenwert.

1		1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17														
Seitenkopf		Büroshop Norderstr. 17, 43219 Phantastica														
Rechnungskopf		Rechnungsnummer: =Rechnungsnu Datum: =Datum														
ID Kopf		Anzahl Ware Preis Anzahl*Preis =Anzahl =Ware =Pr =Anzahl*Pre														
ID Kopf		Übertrag: =SummeAnza ählerRechnunasnum hlerRechnunasnum														
Detail		Anzahl Ware Preis Anzahl*Preis Übertrag: =SummeAnza														
Rechnung...		Summe: =Summ														
Seite...		Kontoverbindung: Stadtparkasse Phantastica - BLZ 123 456 01 - Konto 1029384756														

Die folgende Tabelle zeigt die Einteilung der Seite in unterschiedliche Bereich des Berichtes:

A	Seitenrand oben (nicht im Screenshot zu sehen, über Format → Seite einstellbar)	2,00 cm
B	Seitenkopf (erscheint auf jeder Seite, enthält keine Eingaben der Datenbank, sondern z. B. nur das Firmenlogo sowie die Adresse des Absenders)	3,00 cm
C	Gruppenkopf für die Rechnungsnummer (Nur die Beträge, die zu einer Rechnungsnummer gehören, sollen später auch addiert werden. Der Gruppenkopf erscheint nur zum Beginn der Rechnung.)	2,50 cm
D	Gruppenkopf für die Rechnungsposten (Der Bereich «Detail» wird für andere Inhalte benötigt. Deshalb erfolgt hier eine Gruppierung, die gleichzeitig die Sortierung der Rechnungsinhalte z. B. nach der Eingabe vornimmt. Diese Gruppe besteht immer nur aus einem Wert.)	0,70 cm
E	Gruppenkopf, ebenfalls an Rechnungsposten gebunden (Dieser Bereich wird nur dann angezeigt, wenn so viele Rechnungsposten vorkommen, dass auf jeden Fall eine weitere Rechnungsseite erforderlich ist. Er enthält die Zwischensumme und die Seitennummerierung unten auf der Seite. Nach diesem Bereich erfolgt ein Seitenumbruch.)	2,00 cm
F	Detail-Bereich (Dieser Bereich wird nur dann angezeigt, wenn so viele Rechnungsposten vorkommen, dass auf jeden Fall eine weitere Rechnungsseite erforderlich ist. Er enthält den Übertrag, die Seitennummer oben auf der Seite)	2,50 cm
G	Gruppenfuß für die Rechnungsnummer (Hier erfolgt die Ausgabe des Rechnungsbetrages, ggf. auch mit Angabe der Mehrwertsteuer. Der Gruppenfuß erscheint nur zum Schluss der Rechnung.)	1,60 cm
H	Seitenfuß (z. B. Angabe der Kontoverbindung)	1,00 cm
I	Seitenrand (wie «Seitenrand oben» nicht im Screenshot zu sehen, über Format → Seite einstellbar)	1,00 cm

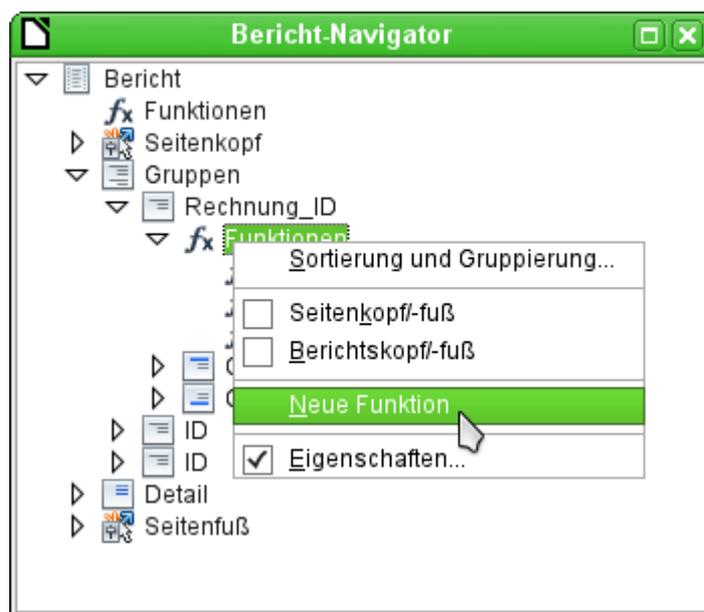
Ein Seitenumbruch soll nur dann erfolgen, wenn zu viele Rechnungsposten vorhanden sind. Für die Rechnungsposten ergibt sich der folgende freie Platz:

	29,70 cm	(DIN A 4)
-	2,00 cm	(Pos. A)
-	3,00 cm	(Pos. B)
-	2,50 cm	(Pos. C)
-	1,60 cm	(Pos. G)
-	1,00 cm	(Pos. H)
-	1,00 cm	(Pos. I)
=	18,60 cm	

Der verbleibende freie Platz kann also maximal $18,60 \text{ cm} / 0,70 \text{ cm} = 26,57\dots$, abgerundet also 26 Rechnungspositionen erfassen.

Sobald die 27. Rechnungsposition vorkommt, muss auf jeden Fall ein Seitenumbruch erfolgen. Das bedeutet, dass dann auf jeden Fall Gruppenkopf E und der Detail-Bereich angezeigt werden müssen. Es muss also ein Zähler für die Rechnungsnummer (Bereich C) eingerichtet werden. Erreicht dieser Zähler die Nummer 27, so wird der Detail-Bereich (F) angezeigt.

Der Zähler der Rechnungsnummer wird wie folgt definiert:



Über den Berichtsnavigator wird die Gruppe «Rechnung_ID» aufgesucht. Die neu zu gründende Funktion heißt einfach «ZählerRechnungsnummer». Die Formel ist «[ZählerRechnungsnummer] + 1». Der Startwert ist 1. Es werden keine Unterberichte eingebunden (die Funktion gibt es gar nicht ...). Es wird auch nicht vorausberechnet. Für die Vorausberechnung wird ein gesonderter Zähler «ZählerKomplett» erstellt.

Der Gruppenkopf E und der Detailbereich F werden angezeigt, wenn insgesamt mehr als 26 Positionen in der Rechnung stehen und die momentane Rechnungsposition 26 erreicht hat. Der Ausdruck für die bedingte Anzeige ist also in beiden Bereichen

`001 AND ([ZählerRechnungsnummer]=26; [ZählerKomplett]>26)`

Der Inhalt dieser Bereiche erscheint also nur, wenn mindestens ein 27. Rechnungsposten zu erwarten ist. Gruppenkopf E erscheint auf der ersten Seite. Er ist mit einem Seitenumbruch

nach dem Bereich versehen. Der Inhalt des Bereiches «Detail» wird auf der folgenden Seite angezeigt.

Jetzt ist zu berechnen, wie groß der Anteil des Bereiches ist, der noch auf der ersten Seite angezeigt wird:

	29,70 cm	(DIN A 4)
-	2,00 cm	(Pos. A)
-	3,00 cm	(Pos. B)
-	2,50 cm	(Pos. C)
-	18,20 cm	(Pos. D * 26)
-	1,00 cm	(Pos. H)
-	1,00 cm	(Pos. I)
=	2,00 cm	

Der Gruppenfuß fällt aus der ersten Seite raus, insgesamt 26 Rechnungsposten sind enthalten. Der Gruppenkopf E kann also maximal 2 cm auf der ersten Seite anzeigen. In diesen 2 cm muss auf jeden Fall die Zwischensumme und die Seitenzahl untergebracht werden. Um auf jeden Fall einen korrekten Seitenumbruch zu erzeugen, sollte der Bereich also etwas kleiner sein. In dem Beispiel wurde er auf 1,90 cm eingestellt.

Der Bereich «Detail» wird auf der Folgeseite oben angezeigt. Da auf der Folgeseite der Gruppenkopf der Rechnungsnummer (C) nicht mehr erscheint, kann hier der Bereich «Detail» so viel Platz einnehmen wie der Gruppenkopf, nämlich 2,50 cm. Dann starten die darauffolgenden Rechnungsposten wieder mit der gleichen Einstellung wie auf der Vorseite.

Der Übertrag wird jeweils als einfache Summierung der vorhergehenden Posten erreicht.

Über den Berichtsnavigator wird die Gruppe «Rechnungsnummer» aufgesucht. Die neu zu gründende Funktion heißt einfach «SummePreis». Die Formel ist «[Preis] + [SummePreis]». Der Startwert ist [Preis]. Es werden keine Unterberichte eingebunden. Es wird auch nicht vorausberechnet.

Der Übertrag wird in dem Gruppenkopf E und dem Detailbereich F angezeigt. Im Gruppenkopf E ist der Übertrag ganz oben positioniert. Er erscheint auf der ersten Seite unten. Im Detailbereich F ist der Übertrag ganz unten positioniert. Er erscheint auf der zweiten Seite direkt unterhalb der Tabellenköpfe.

Die Seitennummerierung wird ähnlich abgefragt wie die Anzeige des Gruppenkopfes E des Bereiches «Detail».

```
001 IF ([ZählerRechnungsnummer]=26;"Seite 1";"
```

Hiermit wird die Seitennummer auf der ersten Seite erstellt. Weitere IF-Abfragen können dann für die weiteren Seitennummern eingebunden werden.

Die Seitennummer für die Folgeseite wird bei der gleichen Bedingung einfach auf "Seite 2" gesetzt.

Werden die Formeln entsprechend weitgehender formuliert, so können beliebig viele Seiten des Berichtes abgedeckt werden.

Der Ausdruck für die **bedingte Anzeige** wechselt also von

```
001 AND ([ZählerRechnungsnummer]=26; [ZählerKomplett]>26)
```

zu

```
001 AND (MOD ([ZählerRechnungsnummer]; 26)=0;  
[ZählerKomplett]>[ZählerRechnungsnummer])
```

Der Gruppenkopf E und der Detailbereich F erscheinen also nur, wenn sich aus der Division des Zählers der Rechnungsnummer mit 26 kein Rest ergibt und die Komplettzahl der Rechnungs-
posten größer ist als der Zähler der Rechnungsnummer.

Der Ausdruck für die **Seitenzahl** wechselt von

001 IF([ZählerRechnungsnummer]=26;"Seite 1";")

zu

001 "Seite "&[ZählerRechnungsnummer]/26

für die aktuelle Seite bzw.

001 "Seite "&([ZählerRechnungsnummer]/26)+1

als Anzeige für die Folgeseite.

Der folgende Berichtsausdruck ist mit diesen Einstellungen noch nicht zu bewerkstelligen:

Büroshop		Büroshop	
Norderstr. 17, 43219 Phantastica		Norderstr. 17, 43219 Phantastica	
Rechnungsnummer: 2013-0 Datum: 11.03.2013		Seite 2 Rechnungsnummer: 2013-0 Datum: 11.03.2013	
Anzahl	Ware	Preis	Anzahl*Preis
2	Papier, 500 Blatt	4,23 €	8,46 €
1	Radlergummi	0,75 €	0,75 €
4	Anspitzer	1,27 €	5,08 €
2	Bleistift HB	0,23 €	0,46 €
1	Collegoblock	0,98 €	0,98 €
1	Briefumschläge, 25 Stck.	1,25 €	1,25 €
4	CD-Hüllen, 100 Stck.	1,89 €	7,56 €
1	Wachsmalkreiden, 6 Stck.	3,85 €	3,85 €
1	Ordner, 5cm Rückenbreite	1,89 €	1,89 €
2	Kleimbrett	4,45 €	8,90 €
1	Ringbuch A4	5,76 €	5,76 €
1	CD-Beschäftigungstifte, 4 Stck.	4,56 €	4,56 €
2	CD-Rohlinge, 50 Stck.	12,34 €	24,68 €
1	Papier, Recycling, 500 Blatt	3,76 €	3,76 €
4	Ordnermappe mit Register	6,87 €	27,48 €
2	Outtingblock, 50 Blatt	1,35 €	2,70 €
10	Rechnungsbuch, 50 Blatt	3,15 €	31,50 €
1	Datumsstempel, einfach	18,50 €	18,50 €
1	Geldkassette, Klein	12,00 €	12,00 €
12	Hängemappen, 25 Stck.	14,75 €	177,00 €
	Übertrag:		347,12 €
Seite 1			
Kontoverbindung: Stadtparkasse Phantastica - BLZ 123 456 01 - Konto 102 034 756			

Anzahl	Ware	Preis	Anzahl*Preis
	Übertrag:		347,12 €
2	Universaletiketten 70x32, 2700 Stck.	20,50 €	41,00 €
1	Universaletiketten Kleinpack 12x30, 700 Stck.	4,15 €	4,15 €
2	Druckerköpfe, schwarz	24,00 €	48,00 €
1	Druckerköpfe, color	26,30 €	26,30 €
3	Druckerpatronen, schwarz, 32ml	5,40 €	16,20 €
5	Druckerpatronen, color, 17ml	5,20 €	26,00 €
2	Toner Laserdrucker schwarz	65,89 €	131,78 €
1	Toner, Laserdrucker, color	78,89 €	78,89 €
2	Register für Ordner, A-Z	1,25 €	2,50 €
1	Heftstreifen, 25 Stck.	0,85 €	0,85 €
2	Schnellhefter Recyclingkarton	0,65 €	1,30 €
1	Papier, 500 Blatt, Recycling	4,15 €	4,15 €
1	Bleistifte, 10 Stck., versch. Stärken	4,85 €	4,85 €
2	Kugelschreiber	1,35 €	2,70 €
1	Wasserfarbkasten, 12 Farben	8,75 €	8,75 €
1	Aquarellkasten, 24 Farben	17,15 €	17,15 €
2	Aquarellpinsel, 3 Stck., versch. Stärken	8,34 €	16,68 €
1	Zeichenblock, A3, 20 Blatt	3,85 €	3,85 €
4	Schreibunterlage 50*70 cm	15,67 €	62,68 €
2	Tonpapier, div. Farben, 50*70 cm	0,45 €	0,90 €
10	Tonkarton, div. Farben, 50*70 cm	0,89 €	8,90 €
1	Datumsstempel, einfach	18,50 €	18,50 €
1	Geldkassette, Klein	12,00 €	12,00 €
12	Hängemappen, 25 Stck.	14,75 €	177,00 €
	Übertrag:		1.052,20 €
Seite 2			
Kontoverbindung: Stadtparkasse Phantastica - BLZ 123 456 01 - Konto 102 034 756			

Durch das Adressfeld sind auf der ersten Seite der Rechnung weniger Rechnungsposten enthalten als auf der zweiten Seite. Der Bereich «Detail», der auf der zweiten Seite ganz oben abgebildet ist, ist also deutlich kleiner als der Gruppenkopf für die Rechnungsnummer (C).

Um unterschiedliche Mengen an Rechnungsposten auf der ersten Seite und den Folgeseiten zuzulassen, müssen also die Formeln noch einmal angepasst werden.

Die folgende Berechnung sichert ab, dass die entsprechenden Bereiche korrekt angezeigt werden.

001 AND
002 (MOD([ZählerRechnungsnummer] - 20; 24)=0;
003 [ZählerRechnungKomplett]>[ZählerRechnungsnummer])

Vom Zähler für die Rechnungsnummer wird die Anzahl der Rechnungsposten auf der ersten Seite subtrahiert. Diese Differenz wird durch die mögliche Anzahl der Rechnungsposten auf der zweiten Seite dividiert. Wenn die Division glatt aufgeht (kein Rest, $\text{MOD} = 0$), dann ist die erste Bedingung für die Anzeige von Gruppenkopf E und der Detailbereich F gegeben. Außerdem muss, wie bereits vorher definiert, der Zähler für die Rechnungsnummer kleiner sein als die insgesamt zu erwartenden Rechnungsposten. Sonst würde ja die Rechnungssumme noch genug Platz auf der entsprechenden Seite haben.

Die mögliche Anzahl der Rechnungsposten auf der zweiten Seite wurde dadurch geringer, dass jetzt auf der zweiten Seite zusätzlich die Rechnungsnummer und das Datum enthalten sind.

Die Seitenzahl wird jetzt etwas einfacher berechnet:

```
001 "Seite "&INT([ZählerRechnungsnummer]/24)+1
```

INT rundet auf die nächste Ganzzahl ab. Die erste Seite enthält maximal 20 Rechnungsposten. Die Division ergibt für die erste Seite also ein Ergebnis < 1 . Abgerundet wird das zu 0. Also muss zu der ermittelten Seitenzahl 1 addiert werden, damit auf der ersten Seite auch 1 erscheint. Entsprechend muss für die zweite Seite 2 addiert werden.

Der Bericht weist in der obigen Form noch einen Schönheitsfehler auf. Ein genauer Blick auf die Rechnungsposten zeigt, dass sich die unteren drei Rechnungsposten gleichen. Dies wurde hier durch einfaches Kopieren der Datensätze erzeugt. Es handelt sich also nicht um die gleichen Datensätze, sondern unterschiedliche Rechnungspositionen, die nacheinander über die Software abgearbeitet wurden. Besser wäre hier, die Waren entsprechend gruppiert anzuzeigen, damit die gleiche Ware nicht mehrmals, sondern nur einmal mit entsprechender Anzahl aufgelistet wird.

Grundsätzlich sollten möglichst viele Berechnungen, Gruppierungen usw. aus dem Report-Designer ausgelagert werden. Deshalb wird für eine entsprechende Gruppierung nicht mit den Gruppen des Report-Designers gearbeitet, sondern mit den Gruppierungsfunktionen des Abfrageeditors. Damit der Report-Designer die Abfrage einwandfrei verarbeiten kann, wird schließlich daraus eine Ansicht gemacht. Der Report-Designer versucht ansonsten, die Abfrage weiter durch seine Gruppierungsvorgaben und Sortierungsvorgaben zu ergänzen, was sehr schnell zu unbrauchbarem Code führt.

Dann ergibt sich die folgende Rechnungsübersicht:

Büroshop
Norderstr. 17, 43219 Phantastica

Büroshop - Norderstr. 17 - 43219 Phantastica
Herrn
Marko Mustermann
Schloßallee 42
05741 Hirtensberg

Rechnungsnummer: 2013-0
Datum: 11.03.2013

Anzahl	Ware	Preis	Anzahl*Preis
2	Papier, 500 Blatt	4,23 €	8,46 €
1	Radiergummi	0,75 €	0,75 €
4	Anspritze	1,27 €	5,08 €
2	Bleistift HB	0,23 €	0,46 €
1	Collegblock	0,98 €	0,98 €
1	Brotumschläge, 25 Stck.	1,25 €	1,25 €
4	CD-Hüllen, 100 Stck.	1,89 €	7,56 €
1	Wachsmalkreiden, 6 Stck.	3,85 €	3,85 €
1	Ordner, 5cm Rückenbreite	1,89 €	1,89 €
2	Klembrett	4,45 €	8,90 €
1	Ringbuch A4	5,76 €	5,76 €
1	CD-Beschriftungsstifte, 4 Stck.	4,56 €	4,56 €
2	CD-Rohlinge, 50 Stck.	12,34 €	24,68 €
1	Papier, Recycling, 500 Blatt	3,76 €	3,76 €
4	Ordnungsmappe mit Register	6,87 €	27,48 €
2	Outtingblock, 50 Blatt	1,35 €	2,70 €
10	Rechnungsbuch, 50 Blatt	3,15 €	31,50 €
2	Datumsstempel, einfach	18,50 €	37,00 €
2	Geldkassette, klein	12,00 €	24,00 €
24	Hängemappen, 25 Stck.	14,75 €	354,00 €
	Übertrag:		554,62 €

Seite 1

Kontoverbindung: Stadtparkasse Phantastica - BLZ 123 456 01 - Konto 1029384756

Büroshop
Norderstr. 17, 43219 Phantastica

Seite 2

Rechnungsnummer: 2013-0
Datum: 11.03.2013

Anzahl	Ware	Preis	Anzahl*Preis
	Übertrag:		554,62 €
4	Universaletketten 70x32, 2700 Stck.	20,50 €	82,00 €
2	Universaletketten Kleinpäck 12x30, 700 Stck.	4,15 €	8,30 €
4	Druckerköpfe, schwarz	24,00 €	96,00 €
2	Druckerköpfe, color	26,30 €	52,60 €
6	Druckerpatronen, schwarz, 32ml	5,40 €	32,40 €
10	Druckerpatronen, color, 17ml	5,20 €	52,00 €
4	Toner Laserdrucker schwarz	65,89 €	263,56 €
2	Toner, Laserdrucker, color	78,89 €	157,78 €
4	Register für Ordner, A-Z	1,25 €	5,00 €
2	Heftstreifen, 25 Stck.	0,85 €	1,70 €
4	Schnellhefter Recyclingkarton	0,65 €	2,60 €
1	Papier, 500 Blatt, Recycling	4,15 €	4,15 €
1	Bleistifte, 10 Stck., versch. Stärken	4,85 €	4,85 €
2	Kugelschreiber	1,35 €	2,70 €
1	Wasserfarbkasten, 12 Farben	8,75 €	8,75 €
1	Aquarellkasten, 24 Farben	17,15 €	17,15 €
2	Aquarellpinsel, 3 Stck., versch. Stärken	8,34 €	16,68 €
1	Zeichenblock, A3, 20 Blatt	3,85 €	3,85 €
4	Schreibunterlage 50*70 cm	15,67 €	62,68 €
2	Toppapier, div. Farben, 50*70 cm	0,45 €	0,90 €
10	Tonerkarton, div. Farben, 50*70 cm	0,89 €	8,90 €
	Summe:		1.439,17 €

Kontoverbindung: Stadtparkasse Phantastica - BLZ 123 456 01 - Konto 1029384756

Alle Rechnungsposten sind nur einmal enthalten. Aus den ursprünglich auf Seite 1 unten stehenden 12 Hängemappen wurden in der Rechnung jetzt 24 Hängemappen. "Anzahl*Preis" wurde entsprechend angepasst.

Ausdruck von Berichten zum aktuellen Datensatz des Formulars

Gerade bei der Rechnungserstellung wie in dem vorhergehenden Beispiel kann es sinnvoll sein, nach jeder Eingabe der Rechnungsposten einen entsprechenden Ausdruck anzufertigen. In einem Formular soll der Inhalt für die Rechnung bestimmt werden und schließlich der Ausdruck des Einzeldokumentes erfolgen.

Berichte lassen sich nicht über Makros mit einem Filter starten. Allerdings kann die Abfrage, die als Grundlage für den Bericht genutzt wird, vorher gefiltert werden. Dies geht entweder über die Form einer Parameterabfrage

```
001 SELECT * FROM "Rechnung"
002 WHERE "ID" = :ID
```

oder über eine Abfrage, die über eine einzeilige Filtertabelle mit Daten versorgt wird:

```
001 SELECT * FROM "Rechnung"
002 WHERE "ID" = (SELECT "Integer" FROM "Filter" WHERE "ID" = TRUE)
```

Bei der Parameterabfrage muss der Inhalt in ein entsprechendes Dialogfeld nach dem Start des Berichtes eingegeben werden.

Bei der Steuerung über eine Filtertabelle wird der Inhalt der Filtertabelle per Makro geschrieben. Eine separate Eingabe ist also nicht mehr nötig. Diese Möglichkeit ist also für den Nutzer einfacher zu handhaben und soll deshalb im Folgenden beschrieben werden.

Aufbau der Filtertabelle

Die Filtertabelle soll lediglich einen Datensatz enthalten. Deshalb kann das Primärschlüsselfeld ein «Ja/Nein»-Feld sein. Andere Felder der Tabelle werden von der Benennung so gewählt, dass bereits eindeutig ist, was für einen Inhalt sie speichern können. In diesem Beispiel heißt das Feld, das den Primärschlüssel der Tabelle "Rechnung" filtern soll, "Integer", da der Primärschlüssel der Tabelle "Rechnung" eben vom Feldtyp «Integer» ist. Für andere Filterungen können andere Felder zusätzlich eingebaut werden. Der Filter "Integer" kann auch für mehrere Tabellen genutzt werden, da ja vor dem Druck der alte Wert einfach mit dem aktuellen Wert überschrieben wird. Diese Doppelnutzung funktioniert so allerdings nur in einer Einzeldatenbank (Base zusammen mit der internen **HSQldb** oder der internen **FIREBIRD**). In einer Mehrbenutzerdatenbank könnte in dem Moment, in dem die Abfrage des Filters erfolgt, von einem anderen Nutzer der Filterwert in einer normalen Tabelle schon wieder geändert werden. Deswegen wird in Mehrbenutzerdatenbanken die VerbindungsID als Primärschlüssel "ID" genutzt.

Feldname	Feldtyp
ID	Ja/Nein [BOOLEAN]
Integer	Integer [INTEGER]

Diese Tabelle wird zum Start mit einem Datensatz gefüllt. Hierzu muss lediglich das Feld "ID" einmal markiert werden, so dass es den Wert «Ja» (oder in SQL «TRUE») hat.

Aufbau des Makros zum Start des gefilterten Berichtes

Das Formular muss für das Aufrufen eines einzelnen Berichtes an irgendeiner Stelle den Primärschlüssel der Tabelle "Rechnung" enthalten. Dieser Primärschlüsselwert wird ausgelesen und über ein Makro in die Tabelle "Filter" übertragen. Anschließend wird in dem Makro noch der gewünschte Bericht gestartet.

```
001 SUB Filtern_und_Drucken
002   DIM oDoc AS OBJECT
003   DIM oDrawpage AS OBJECT
004   DIM oForm AS OBJECT
005   DIM oFeld AS OBJECT
006   DIM oDatenquelle AS OBJECT
007   DIM oVerbindung AS OBJECT
008   DIM oSQL_Anweisung AS OBJECT
009   DIM stSQL AS STRING
010   oDoc = thisComponent
011   oDrawpage = oDoc.Drawpage
012   oForm = oDrawpage.Forms.getByName("MainForm")
013   oFeld = oForm.getByName("fmtID")
014   oDatenquelle = ThisComponent.Parent.CurrentController
015   If NOT (oDatenquelle.isConnected()) THEN
016     oDatenquelle.connect()
017   END IF
018   oVerbindung = oDatenquelle.ActiveConnection()
019   oSQL_Anweisung = oVerbindung.createStatement()
020   stSql = "UPDATE ""Filter"" SET ""Integer"" = '"+oFeld.GetCurrentValue()+
021     "' WHERE ""ID"" = TRUE"
022   oSQL_Anweisung.executeUpdate(stSql)
023   ThisDatabaseDocument.ReportDocuments.getByName("Rechnung").open
024 END SUB
```

Das Formular hat in diesem Beispiel den Namen "MainForm". Das Primärschlüsselfeld heißt "fmtID". Dieses Schlüsselfeld muss nicht sichtbar sein, um darauf mit dem Makro zugreifen zu können. Der Wert dieses Feldes wird ausgelesen und mit dem UPDATE-Befehl in die Tabelle "Filter" geschrieben. Anschließend wird der Bericht gestartet. Die Ansicht, auf der der Bericht beruht, wurde entsprechend um eine Bedingung erweitert:

```
001 ... WHERE "Rechnung_ID" = COALESCE((SELECT "Integer" FROM "Filter" WHERE
    "ID" = TRUE), "Rechnung_ID") ...
```

Es wird das Feld "Integer" ausgelesen. Falls dieses Feld keinen Wert enthält, wird stattdessen "Rechnung_ID" = "Rechnung_ID" gesetzt. Das bedeutet, dass alle Datensätze angezeigt werden – nicht nur der Filterdatensatz. So können also gegebenenfalls mit der gleichen Ansicht auch alle gespeicherten Rechnungen ausgedruckt werden.

Wechselnde Einfärbung von Zeilen

Tabellenzeilen in einem Bericht können manchmal beim Lesen dazu führen, dass der Blick um eine Zeile nach unten oder oben rutscht. Dagegen kann eine Einfärbung von mindestens einer Zeile helfen. Im folgenden Beispiel⁴ werden die Zeilen einfach wechselseitig mit unterschiedlichen Farben eingefärbt. Der Bericht sieht dann wie folgt aus:

Kathrin	17.01.84
Sally	15.02.91
Mick	03.03.53
Hanne	13.04.70
Meike	13.04.71
Lara	23.04.85

Grundlage des Berichtes ist eine Abfrage mit Namen und Datumseinträgen. Es wurde aus der ursprünglichen Tabelle eine Abfrage erstellt, damit die Daten nach Monaten und Tagen sortiert sind, also die Geburtsstagsreihenfolge im Jahr anzeigen. Dies geht über

```
001 ... ORDER BY
002 MONTH("Geburtstag") ASC,
003 DAY("Geburtstag") ASC (HSQLDB, FIREBIRD)
```

Damit der Code sowohl in der HSQLDB als auch in FIREBIRD läuft, muss er folgendermaßen angepasst werden:

```
001 ... ORDER BY
002 EXTRACT(MONTH FROM "Geburtstag") ASC,
003 EXTRACT(DAY FROM "Geburtstag") ASC (HSQLDB, FIREBIRD)
```

Um eine wechselseitige Einfärbung zu erreichen muss irgendeine Funktion erstellt werden, aus der heraus über einen Wert später die Bedingung für die Einfärbung abgeleitet werden kann. Hier wird ein Textfeld im Bericht aufgezogen und über die **Eigenschaften → Daten → Datenfeld-Typ** ein Zähler definiert.



Der Name der Funktion wird für die bedingte Formatierung benötigt. Der Zähler selbst braucht beim Ausdruck nicht zu erscheinen. Der Name kann direkt aus dem aufgezogenen Feld abgelesen werden. Ist das Feld wieder gelöscht, so ist der Funktionsname weiterhin über **Ansicht → Berichts-Navigator** nachschlagbar.

⁴ Die Datenbank «Beispiel_Bericht_Zeilen_Farbwechsel_Spalten.odt» zu diesem Bericht ist den Beispieldatenbanken für dieses Handbuch beigelegt.

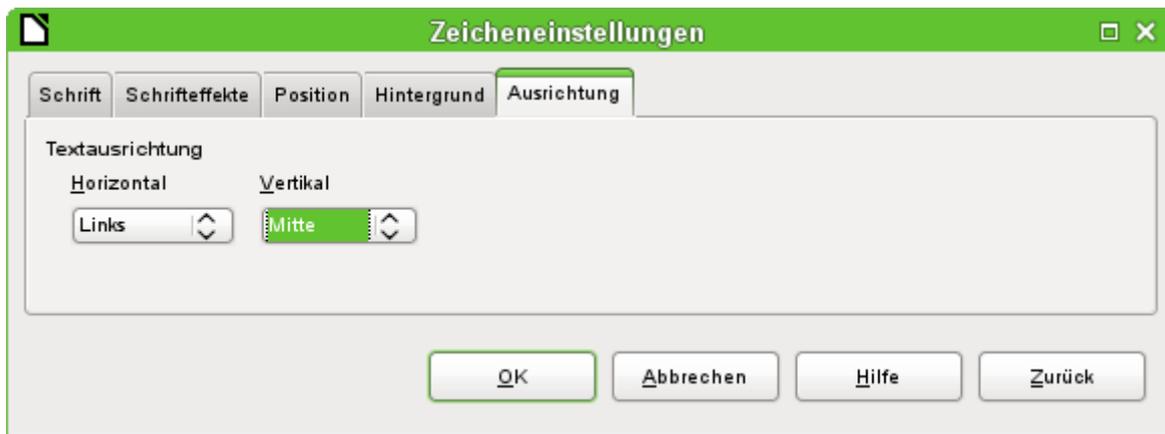


Jetzt muss jedem der Textfelder über den Zähler ein entsprechendes Format zugewiesen werden. Die Bedingung ist jedes Mal ein Ausdruck, der nicht direkt mit dem Feld zusammen hängt. Deshalb wird als **Bedingung 1** → **Ausdruck ist** → **MOD([ZählerBericht];2)>0** eingestellt. MOD berechnet den Rest einer Division. Bei allen ungeraden Zahlen ist der Rest größer als 0, bei allen Geraden Zahlen ist der Rest 0. Zeile 1, 3, 5 usw. wird also mit dem entsprechenden Format versehen.



Als zweite Bedingung wird jetzt das Gegenteil von der ersten Bedingung formuliert und eine entsprechende Formatierung zugewiesen. Diese zweite Bedingung könnte auch ausgelassen werden und stattdessen in den Eigenschaften eines jeden Feldes ein entsprechendes Format eingestellt werden. Die bedingte Formatierung der ersten Bedingung würde dann nur die Standardformatierung ersetzen, wenn die erste Bedingung zutrifft.

Da die bedingte Formatierung alle Formate der Standardformatierung überschreibt, muss z. B. eine Ausrichtung der Schrift in den Zeicheneinstellungen der bedingten Formatierung erfolgen:



Hier wird die Schrift vertikal mittig in dem eingefärbten Textfeld dargestellt. Horizontal lassen sich die üblichen Ausrichtungen einstellen, nicht aber eine Einrückung, so dass die Buchstaben nicht am linken Rand des Textfeldes stehen bleiben.

Versuche mit der Verbindung von Leerzeichen mit dem Inhalt in einer Abfrage oder einer Formel führen hier nicht dazu, dass das Textfeld tatsächlich den Text eingerückt darstellt. Die Leerzeichen werden vermutlich einfach abgeschnitten.



Zielführender ist es dagegen, einfach ein Textfeld vor dem eigentlichen Text zu positionieren, das aber nicht mit irgendeinem Datenfeld-Typ verbunden wird. Das Textfeld wird genau wie die anderen enthaltenen Felder mit der entsprechenden bedingten Formatierung versehen, so dass eine einheitlich breite (scheinbare) Einrückung beim Ausdruck erfolgt.

Zweispaltige Berichte

Mit geschickter Abfragetechnik ist es möglich, einen Bericht mit mehreren Spalten zu erstellen, bei dem die folgenden Spalten auch die folgenden Datensätze darstellen:

Geburtstagsliste

Kathrin	17.01.84	Sally	15.02.91
Mick	03.03.53	Hanne	13.04.70
Meike	13.04.71	Lara	23.04.85
Monika	05.06.86	Karl	01.07.67
Paul	11.07.89	Egon	23.07.67
Susanne	02.08.65	Georg	28.08.95
Ysabelle	17.09.89	Maik	28.10.93
John	19.11.97	Erkan	17.12.75
Johann	23.12.91		

Der erste Datensatz wird in der linken Spalte aufgeführt, der zweite Datensatz in der rechten Spalte. Die Daten sind nach der Lage des Geburtstages im Jahr sortiert.

Gerade die Sortierung nach Geburtstagen führt dazu, dass die Abfrage für diesen Bericht recht lang ausfällt. Könnte stattdessen eine Sortierung nur nach dem Primärschlüssel der zugrundeliegenden Tabelle erfolgen, so würde der Inhalt der Abfrage stark abnehmen. Bei dem Sortierkriterium handelt es sich um einen immer wiederkehrenden Textblock, der weiter unten erklärt wird.

Grundlage für diesen Bericht ist die folgende Abfrage: (HSQLDB, FIREBIRD (Codeanpassung für Firebird siehe zum Schluss dieses Kapitels))

```

001 SELECT "T1"."Name" AS "Name1", "T1"."Geburtsdatum" AS "Geburtsdatum1",
      "T2"."Name" AS "Name2", "T2"."Geburtsdatum" AS "Geburtsdatum2"
002 FROM
003     (SELECT "Name", "Geburtsdatum", "ZeilenNr" AS "Zeile" FROM
004         (SELECT "a".*,
005             ( SELECT COUNT( "ID" ) FROM "Geburtstage" WHERE
006                 RIGHT( '0' || MONTH( "Geburtsdatum" ), 2 ) ||
                 RIGHT( '0' || DAY( "Geburtsdatum" ), 2 ) || "ID"
                 <= RIGHT( '0' || MONTH( "a"."Geburtsdatum" ), 2 ) ||
                 RIGHT( '0' || DAY( "a"."Geburtsdatum" ), 2 ) || "a"."ID" )
             AS "ZeilenNr"
007         FROM "Geburtstage" AS "a"
008     WHERE MOD( "ZeilenNr", 2 ) > 0 )
009 AS "T1"
010 LEFT JOIN
011     (SELECT "Name", "Geburtsdatum", "ZeilenNr"-1 AS "Zeile" FROM
012         (SELECT "a".*,
013             ( SELECT COUNT( "ID" ) FROM "Geburtstage" WHERE
014                 RIGHT( '0' || MONTH( "Geburtsdatum" ), 2 ) ||
                 RIGHT( '0' || DAY( "Geburtsdatum" ), 2 ) || "ID"
                 <= RIGHT( '0' || MONTH( "a"."Geburtsdatum" ), 2 ) ||
                 RIGHT( '0' || DAY( "a"."Geburtsdatum" ), 2 ) || "a"."ID" )
             AS "ZeilenNr"
015         FROM "Geburtstage" AS "a"
016     WHERE MOD( "ZeilenNr", 2 ) = 0 )
017 AS "T2"
018 ON "T1"."Zeile" = "T2"."Zeile"
019 ORDER BY "T1"."Zeile"

```

Es werden zwei identische Unterabfragen in der Abfrage dargestellt. Die ersten beiden Spalten beziehen sich auf die Unterabfrage mit dem Alias "T1", die letzten beiden Spalten auf die Unterabfrage mit dem Alias "T2".

Die Unterabfragen stellen neben den Feldern der Tabelle "Geburtstage" noch Felder zur Verfügung, die eine Unterscheidung in Zeilen und damit eine Sortierung ermöglichen. Dazu wird im Kern die Abfragemöglichkeit zur «Zeilennummerierung» genutzt.

```

006 RIGHT( '0' || MONTH( "Geburtsdatum" ), 2 ) || RIGHT( '0' ||
      DAY( "Geburtsdatum" ), 2 ) || "ID"

```

Diese Formulierung dient dazu, eine eindeutige Reihenfolge der Datensätze zu gewährleisten. Da die Beispieldatensätze nach dem Datum sortiert werden sollen, ließe sich ja leicht sagen, dass lediglich das Datum zum Vergleich herangezogen werden sollte. Etwas schwieriger ist es allerdings schon dadurch, dass nicht das Geburtsdatum, sondern die Lage des Datums im Jahr maßgebend sein soll. Zusätzliche Probleme bereiten schließlich gleiche Datumswerte, die eine eindeutige Reihenfolge verhindern. Deshalb wird zur Sortierung neben dem Monat und dem Tag auch noch der Primärschlüssel der Tabelle herangezogen, der schließlich eindeutig ist. Damit nicht der Monat '10' vor den Monat '2' gesetzt wird, wird bei der Zusammenführung des Sortierkriteriums in einen Text über || vor jede Monatszahl eine führende '0' positioniert, die anschließend bei zweistelligen Monatszahlen über **RIGHT**(... , 2) wieder entfernt wird.

Mit **SELECT COUNT("ID")** wird die Anzahl der Datensätze ermittelt, deren Kombination aus Monat, Tag und Primärschlüssel kleiner oder gleich der entsprechenden Kombination des aktuellen Datensatzes der Tabelle "Geburtstage" ist. Hier handelt es sich um eine «Korrelierte Unterabfrage».

Über **MOD("ZeilenNr", 2)** wird aus den so ermittelten Zeilennummerierungen ermittelt, ob es sich um eine gerade oder ungerade Zahl handelt. **MOD** ermittelt den Rest der Division, in dem

genannten Beispiel der Division durch 2. Dadurch gibt die **"ZeilenNr"** wechselweise die Nummern '1' und '0' aus. Hierdurch werden die Abfragen für "T1" und "T2" unterschieden.

In der nächsthöheren Abfrageebene von "T2" wird als "Zeile" "ZeilenNr"-1 definiert. Dadurch werden "T1" und "T2" direkt vergleichbar.

"T1" wird mit "T2" über **LEFT JOIN** verbunden, damit auch bei ungerader Datensatzzahl der Tabelle "Geburtstage" alle Daten dargestellt werden. In der Zusammenführung von "T1" und "T2" kann jetzt auf die Spalten direkt Bezug genommen werden: **"T1"."Zeile" = "T2"."Zeile"**.

Zum Schluss wird der ganze Inhalt noch nach dem Wert aus "Zeile" sortiert, der für "T1" und "T2" gleich ist. Dies könnte auch der Bericht über die Gruppierung direkt übernehmen.

Geburtstagsliste

Januar		Februar	
Kathrin	17.01.84	Sally	15.02.91
März		April	
Mick	03.03.53	Hanne	13.04.70
		Meike	13.04.71
		Lara	23.04.85
Mai		Juni	
		Monika	05.06.86
Juli		August	
Karl	01.07.67	Susanne	02.08.65
Paul	11.07.89	Georg	28.08.95
Egon	23.07.67		
September		Oktober	
Ysabelle	17.09.89	Maik	28.10.93
November		Dezember	
John	19.11.97	Erkan	17.12.75
		Johann	23.12.91

Wesentlich komplizierter sind von der Zusammensetzung Abfragetechniken, die neben der zweiseitigen Darstellung auch noch zusätzliche Unterteilungen ermöglichen. Hier fallen dann nämlich Leerzeilen mitten im Bericht an, die bei der vorhergehenden zweiseitigen Darstellung höchstens am Schluss vorkommen. Mit einer so erstellten Abfrage kommt der Report-Designer erst einmal nicht zurecht. Deswegen wird stattdessen auf zwei miteinander verbundene Ansichten zurückgegriffen.

Die folgende Ansicht "MonatZ" wird zuerst erstellt:

```

001 SELECT "a"."ID", "a"."Name", "a"."Geburtsdatum",
002     MONTH( "a"."Geburtsdatum" ) AS "MonatZahl",
003     ( SELECT COUNT( "ID" ) FROM "Geburtstage"
004     WHERE MONTH( "Geburtsdatum" ) = MONTH( "a"."Geburtsdatum" )
005     AND RIGHT( '0' || DAY( "Geburtsdatum" ), 2 ) || "ID" <=
006     RIGHT( '0' || DAY( "a"."Geburtsdatum" ), 2 ) || "a"."ID" )
007     AS "MonatZaehler"
007 FROM "Geburtstage" AS "a"

```

Alle Felder der Tabelle "Geburtstage" werden übernommen. Zusätzlich wird der Monat als Zahl dargestellt. Der Tabelle "Geburtstage" wird ein Alias "a" zugewiesen, damit auf die Tabelle mit einer korrelierenden Unterabfrage zugegriffen werden kann.

Die Unterabfrage zählt innerhalb eines Monats alle Datensätze, deren Datumswert einen kleiner oder gleichen Tag vorweist. Bei gleichen Tagen unterscheidet schließlich der Primärschlüssel, welcher Datensatz die niedrigere Nummer erhält. Die Technik ist hier gleich wie bei dem vorhergehenden Beispiel.

Auf die Ansicht "MonatZ" greift die Ansicht "Bericht_Monat_zweispaltig" zu. Diese Ansicht wird hier nur in Ausschnitten wieder gegeben:

```
001 SELECT
002     "Tab1"."Name" AS "Name1",
003     "Tab1"."Geburtsdatum" AS "Geburtsdatum1",
004     1 AS "MonatZahl1",
005     IFNULL("Tab1"."MonatZaehler",999) AS "MonatZaehler1",
006     'Januar' AS "Monat1",
007     "Tab2"."Name" AS "Name2",
008     "Tab2"."Geburtsdatum" AS "Geburtsdatum2",
009     2 AS "MonatZahl2",
010     IFNULL("Tab2"."MonatZaehler",999) AS "MonatZaehler2",
011     'Februar' AS "Monat2"
012 FROM
013     (SELECT * FROM "MonatZ" WHERE "MonatZahl" = 1) AS "Tab1"
014     RIGHT JOIN (SELECT * FROM "MonatZ" WHERE "MonatZahl" = 2) AS "Tab2"
015     ON "Tab1"."MonatZaehler" = "Tab2"."MonatZaehler"
016 UNION
017 SELECT
018     "Tab1"."Name" AS "Name1",
019     "Tab1"."Geburtsdatum" AS "Geburtsdatum1",
020     3 AS "MonatZahl1",
021     IFNULL("Tab1"."MonatZaehler",999) AS "MonatZaehler1",
022     'März' AS "Monat1",
023     "Tab2"."Name" AS "Name2",
024     "Tab2"."Geburtsdatum" AS "Geburtsdatum2",
025     4 AS "MonatZahl2",
026     IFNULL("Tab2"."MonatZaehler",999) AS "MonatZaehler2",
027     'April' AS "Monat2"
028 FROM
029     (SELECT * FROM "MonatZ" WHERE "MonatZahl" = 3) AS "Tab1"
030     LEFT JOIN (SELECT * FROM "MonatZ" WHERE "MonatZahl" = 4) AS "Tab2"
031     ON "Tab1"."MonatZaehler" = "Tab2"."MonatZaehler"
032 UNION
    ...
    ORDER BY "MonatZahl1", "MonatZahl2", "MonatZaehler1", "MonatZaehler2"
```

Zuerst werden in der Unterabfrage aus "MonatZ" alle Daten ausgelesen, bei denen die "MonatZahl" **1** ist. Diese Auswahl wird mit dem Alias "Tab1" versehen. Gleichzeitig werden mit dem Alias "Tab2" alle Daten ausgelesen, bei denen die "MonatZahl" **2** ist. Beide Tabellen werden durch einen **RIGHT JOIN** verbunden, so dass alle Datensätze aus "Tab2" und nur die Datensätze aus "Tab1" angezeigt werden, bei denen der "MonatZaehler" mit dem von "Tab2" übereinstimmt.

Die Spalten der Ansicht müssen unterschiedliche Bezeichnungen ausweisen, so dass jede Spalte mit einem Alias versehen ist. Außerdem wird direkt als Spaltenwert für "Tab1" eine **1** als "MonatZahl" sowie **'Januar'** als "Monat1" eingegeben. Diese Einträge erscheinen auch, wenn es keinen Datensatz aus "Tab1", wohl aber noch Datensätze aus "Tab2" gibt. Ist kein "MonatZaehler" vorhanden, so soll dort der Wert **999** eingetragen werden. Da "Tab1" mit "Tab2" über einen **RIGHT JOIN** verbunden ist, kann es ja vorkommen, dass bei weniger Datensätzen in "Tab1" stattdessen leere Felder angezeigt werden. Leere Felder würden aber bei einer späteren Sortierung vor allen Feldern mit Inhalt sortiert, so dass stattdessen ein besonders hoher Wert gewählt wurde.

Bei der Darstellung der Spalten für "Tab2" wird entsprechend vorgegangen. Hier könnte allerdings **IFNULL("Tab2"."MonatZaehler",999)** entfallen, da bei einem **RIGHT JOIN** zugunsten von "Tab2" zwar alle Zeilen aus "Tab2", nicht aber mehr Zeilen aus "Tab1" als aus "Tab2" dargestellt werden.

Genau dieses Problem wird mit der Verbindung zweier Abfragen gelöst. Mit **UNION** werden alle Datensätze aus der ersten Abfrage und alle Datensätze aus der zweiten Abfrage dargestellt. Die Datensätze aus der zweiten Abfrage erscheinen aber nur dann, wenn sie nicht identisch mit einem vorhergehenden Datensatz sind. **UNION** wirkt also wie **DISTINCT**.

Über **UNION** wird die gleiche Abfrage noch einmal gestellt, nur sind "Tab1" und "Tab2" jetzt mit einem **LEFT JOIN** verbunden. Damit erscheinen auf jeden Fall alle Datensätze aus "Tab1" auch wenn in "Tab2" weniger Datensätze vorhanden sind als in "Tab1".

Für die Monate 3 und 4, 5 und 6 usw. werden entsprechend angepasste Abfragen verwendet und wiederum mit **UNION** an die vorhergehenden Abfragen angehängt.

Das Ergebnis der Ansicht wird schließlich nach "MonatZahl1", "MonatZaehler1" und "MonatZaehler2" sortiert. Nach "MonatZahl2" muss nicht sortiert werden, da "MonatZahl1" ja bereits die entsprechende Reihenfolge wiedergibt.

Sollen die Abfragen zur Geburtstagsliste an **FIREBIRD** angepasst werden, so sind Tage und Monate durch **EXTRACT(DAY FROM "Datum")** bzw. **EXTRACT(MONTH FROM "Datum")** zu ermitteln. Außerdem ist statt **IFNULL()** die Funktion **COALESCE()** erforderlich. Hier braucht allerdings nur der Name ausgetauscht zu werden.

Bugs und Workarounds beim Report-Designer

Der Report-Designer birgt manchmal Fehler, die nicht so ohne weiteres nachvollzogen werden können. Hier einige Fehlerquellen und eventuell nützliche Gegenmaßnahmen.

Der Inhalt eines Feldes aus einer Abfrage erscheint nicht

Eine Datenbank soll den Verkauf von Waren simulieren. In einer Abfrage wird aus der Anzahl einer gekauften Ware und dem Einzelpreis der Gesamtpreis ermittelt.

```
001 SELECT "Verkauf"."Anzahl", "Ware"."Ware", "Ware"."Preis",
002 "Verkauf"."Anzahl"*"Ware"."Preis"
003 FROM "Verkauf", "Ware" WHERE "Verkauf"."Ware_ID" = "Ware"."ID"
```

Diese Abfrage dient als Grundlage des Berichtes. Wird aber das Feld

"Verkauf"."Anzahl"*"Ware"."Preis" in dem Bericht aufgerufen, so bleibt es ohne Inhalt.

Wird dem Feld in der Abfrage hingegen ein Alias zugewiesen, so kann der Report-Designer darauf einwandfrei zugreifen:

```
001 SELECT "Verkauf"."Anzahl", "Ware"."Ware", "Ware"."Preis",
002 "Verkauf"."Anzahl"*"Ware"."Preis" AS "GPreis"
003 FROM "Verkauf", "Ware" WHERE "Verkauf"."Ware_ID" = "Ware"."ID"
```

Das Feld greift jetzt auf **"Gpreis"** zu und stellt den entsprechenden Wert dar.

Ein Bericht lässt sich nicht ausführen

Manchmal kommt es aber vor, dass sich ein Bericht zwar erstellen, aber anschließend nicht ausführen oder nicht einmal speichern lässt. Es erscheint eine Fehlermeldung, die erst einmal wenig aussagekräftig ist:

«Bericht konnte nicht ausgeführt werden. Eine Ausnahme vom Typ com.sun.star.lang.WrappedTargetException wurde entdeckt.»

Hier kann es schon einmal hilfreich sein, sich die entsprechend angebotenen zusätzlichen Informationen anzeigen zu lassen. Taucht dort irgendwie ein Bezug zu «SQL» auf, so ist vermutlich der Report-Designer nicht in der Lage, den SQL-Code der Datenquelle korrekt zu interpretieren.

Hier hilft es vielleicht, über den Berichtsnavigator die Eigenschaften des Berichtes aufzusuchen: **Daten → SQL-Befehl analysieren → Nein**. Leider hat diese Problemlösung zur Folge, dass eine einmal eingestellte Gruppierung nicht mehr funktioniert.

Hinweis

Ein besser geeigneter Weg, Problemen mit dem SQL-Code einer Abfrage aus dem Weg zu gehen, ist, statt eines Abfrage als Grundlage für den Bericht eine **Ansicht** zu nutzen. Die wird von der Datenbank so erstellt, dass sie für den Report-Designer wie eine Tabelle erscheint und ohne Probleme zu verarbeiten ist. Hier funktioniert sogar die Sortiervorgabe der Ansicht reibungslos.

Auch Abfragen, die sonst nur in direktem SQL-Code ausführbar und damit für den Report-Designer teilweise nicht interpretierbar sind, können über die Ansicht ihre Inhalte problemlos an den Report-Designer weitergeben.

Datums- und Zeitwerte werden in Diagrammen nicht angezeigt

Leider kam es wiederholt vor, dass Berichte mit Diagrammen im Report-Designer überhaupt nicht angezeigt wurden. Lässt sich ein Bericht mit Diagrammen anzeigen, so kann es vorkommen, dass Achsen mit einer Datums- oder Zeitangabe die Anzeige der Werte für das Diagramm unmöglich machen: [Bug 87012](#). Hier hilft die Umwandlung der Datums- und Zeitwerte in einer Abfrage. Aus Datumsangaben werden dadurch Integer-Zahlen, aus Zeitwerten werden dadurch Dezimalzahlen, die die Zeit als Bruchteil eines Tages darstellen.

HSQLDB:

```
001 SELECT
002     DATEDIFF( 'dd', '1899-12-30', "Datum" ) AS "DatumInteger",
003     "Temperatur"
004 FROM "Tabelle"
```

FIREBIRD:

```
001 SELECT
002     "Datum" - DATE '1899-12-30' AS "DatumInteger",
003     "Temperatur"
004 FROM "Tabelle"
```

rechnet den Zeitunterschied in Tagen zu dem internen 0-Wert um. Die Datumszählung beginnt für LO beim 30.12.1899.

HSQLDB:

```
001 SELECT
002     HOUR( "Zeit" ) / 24.000000 + MINUTE( "Zeit" ) / 1440.000000
003     AS "ZeitDezimal",
004     "Temperatur"
005 FROM "Tabelle"
```

FIREBIRD:

```
001 SELECT
002     ("Zeit" - TIME '00:00')/86400.000000 AS "ZeitDezimal",
003     "Temperatur"
004 FROM "Tabelle"
```

wandelt die Zeitangabe in Bruchteile eines Tages um. Dabei muss darauf geachtet werden, dass die Division durch Dezimalzahlen mit Nachkommastellen erfolgt. Sonst werden nur ganze Tage wiedergegeben.

Die Integer-Angaben für das Datum bzw. die Dezimalzahlen für die Zeit können für die Achsen des Diagramms als Datum bzw. Zeit formatiert werden. Die dargestellten Werte sind korrekt. Das Diagramm wird auch gezeichnet.

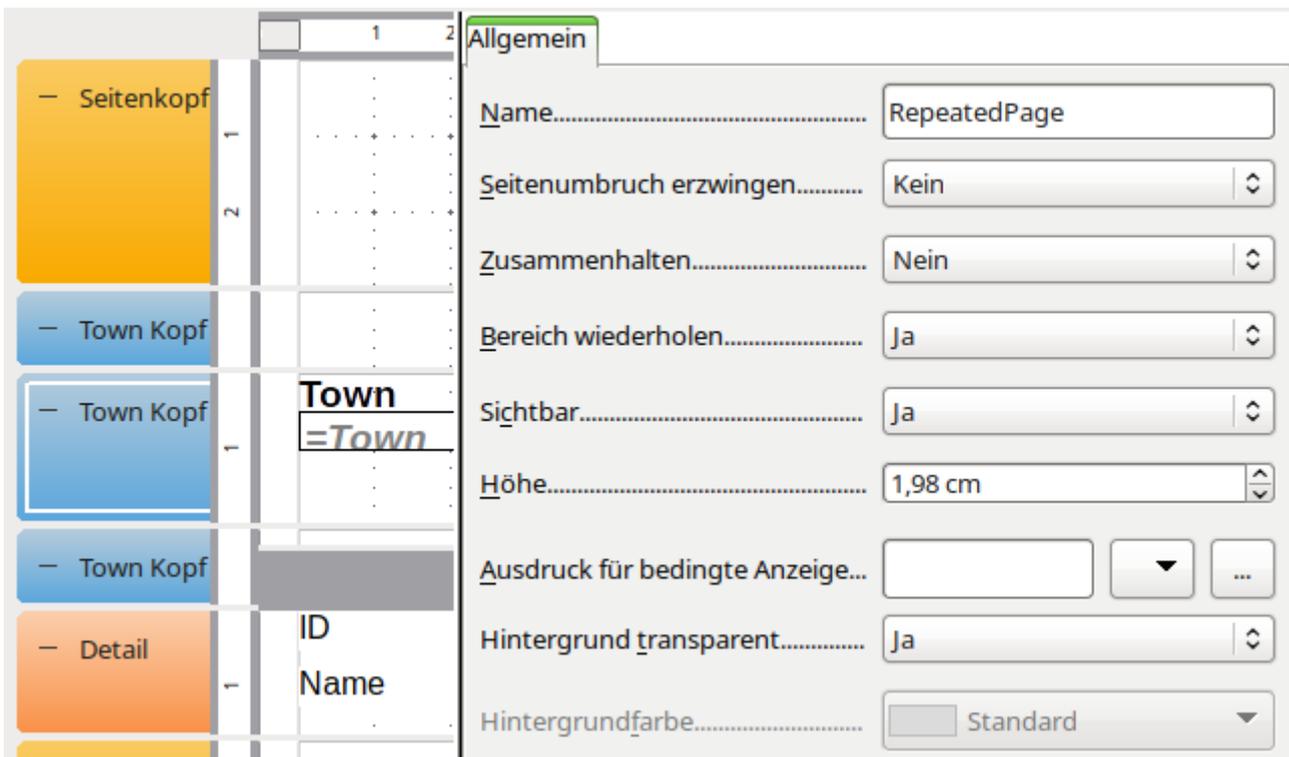
Gruppierungen mit wiederholendem Bereich zeigen nur den ersten Wert

Wird ein Bericht mit einer Gruppierung erstellt, die auf jeder Seite angezeigt werden soll (**Bereich wiederholen** → 'Ja'), dann zeigt anschließend der Bericht den Inhalt des ersten Gruppenbereichs bei jeder Gruppe an: [Bug 82097](#). Dies liegt daran, dass der wiederholende Bereich bei der Erstellung des Dokumentes in die Kopfzeile der Seite geschrieben wird. Leider wird aber der Zeitpunkt nicht konkret belegt, wann denn eine neue Seitenvorlage erstellt werden soll, so dass die Kopfzeile jetzt die nächste Gruppe anzeigen könnte.

Wird zu dem gleichen Element eine weitere Gruppe erzeugt, die vor der wiederholenden Gruppe steht, nicht sichtbar ist und den Bereich nicht wiederholen lässt, so lässt sich darüber der Seitenumbruch erzeugen, der auch zu einem Wechsel in der Kopfzeile führen kann.

The screenshot shows a report design tool interface. On the left, a table is displayed with two columns, '1' and '2'. The rows are grouped into four categories: 'Seitenkopf' (orange), 'Town Kopf' (blue), 'Town Kopf' (blue), and 'Detail' (orange). The 'Town Kopf' groups are further divided into sub-groups: the first is 'Town Kopf', the second is 'Town Kopf', and the third is 'Town Kopf'. The 'Detail' group is 'ID' and 'Name'. The 'Town Kopf' group is currently selected, and its properties are shown in the 'Allgemein' (General) panel on the right. The properties are: Name: Invisible; Seitenumbruch erzwingen: Nein; Zusammenhalten: Nein; Bereich wiederholen: Nein; Sichtbar: Nein; Höhe: 1,00 cm; Ausdruck für bedingte Anzeige: (empty); Hintergrund transparent: Ja; Hintergrundfarbe: Standard.

Hier existieren 3 Gruppen zu dem Feld «Town». Die erste Gruppe ist unsichtbar geschaltet. Alle Eigenschaften sind auf 'Nein' gesetzt. Die Höhe spielt keine Rolle.

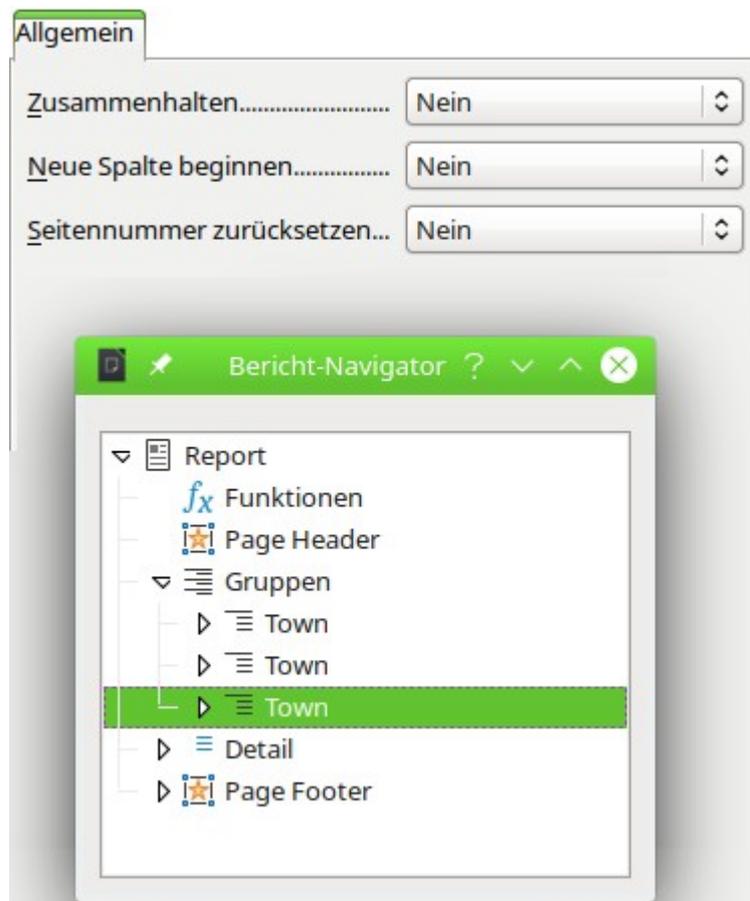


In der zweiten Gruppe steht der Wiederholungsbereich. Die Gruppe ist sichtbar. Dieser Inhalt wird beim Ausführen des Berichtes in der Kopfzeile des Dokumentes angezeigt.

Es ist nicht erforderlich, **Seitenumbruch erzwingen** → 'Ja' einzustellen. Das geschieht automatisch mit der Wiederholung des Bereichs im Hintergrund: [Bug 51452](#). Dieser Bug hat auch zur Folge, dass eine Gruppe, die vor der Wiederholungsgruppe im Editor auftaucht, immer alleine auf der vorhergehenden Seite erscheint. Soll dies nicht geschehen, so dürfen vor der Gruppe, die wiederholt werden soll, nur Gruppen stehen, die gar nicht erst angezeigt werden (wie oben die unsichtbare Gruppe, die dem Report-Designer nur hilft, den Änderung des Inhaltes bei der folgenden Gruppe zu erstellen).

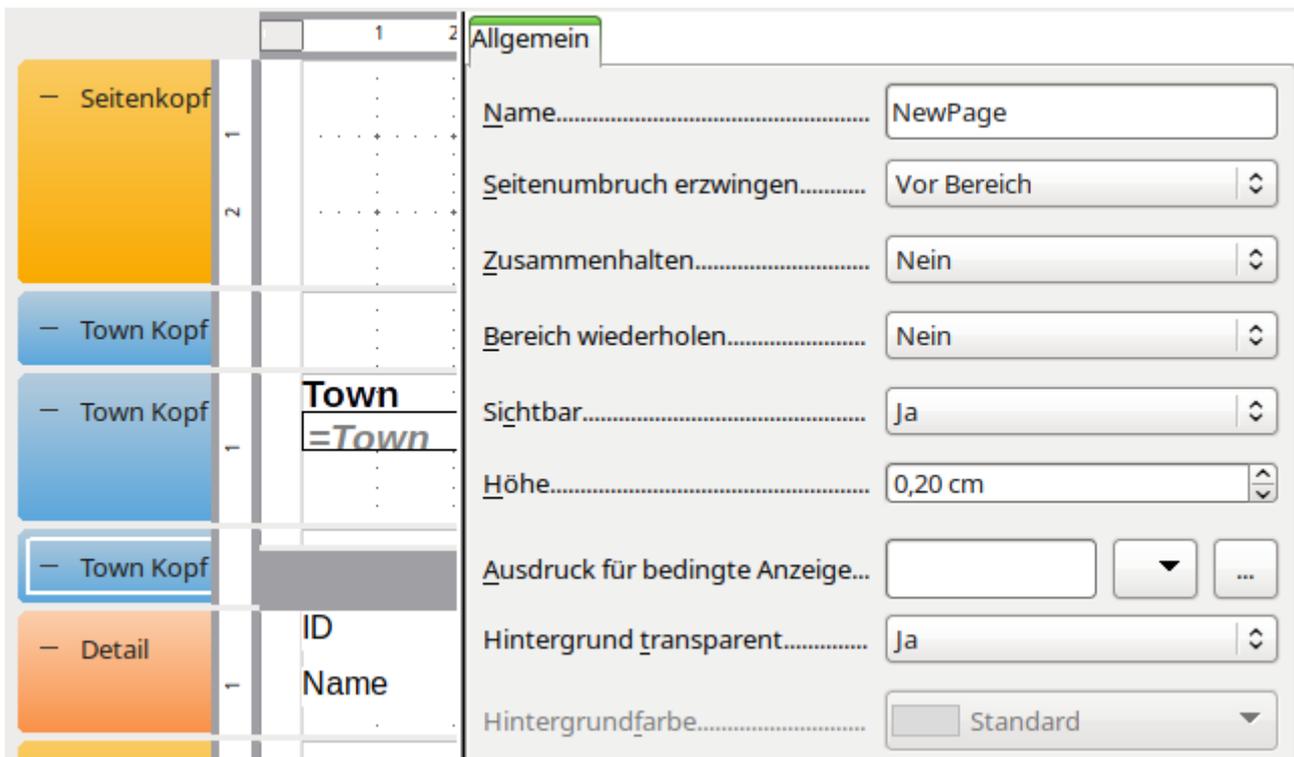
Ein Neustart der Seitenzählung mit der Gruppe ist nicht möglich

Manche Berichte werden ausgedruckt und anschließend an verschiedene Gruppe verteilt, die jeweils nur einen Teil des Berichtes – eben den für ihre Gruppe – erhalten. Hier wäre es gut, wenn der Report-Designer einen Wechsel der Seitenzahlen anbietet: [Bug 51453](#). Offiziell tut er das auch – nur funktioniert die Einstellung leider nicht:



Über den Berichtsnavigator ist jeweils eine zusätzliche Einstellung für die Gruppe erreichbar. Weder **Neue Spalte beginnen** ([Bug 52944](#)) noch **Seitennummer zurücksetzen** bewirken hier irgend etwas.

Für das Setzen der neuen Seitennummer wurde in dem obigen Beispiel eine dritte Gruppe eingebaut:



Diese Gruppe muss sichtbar sein, sollte aber in diesem Fall nur wenig Platz beanspruchen. Sie wird nach dem wiederholten Bereich eingebaut, da sie sonst alleine auf der vorhergehenden Seite stehen würde. Auch wenn hier **Seitenumbruch erzwingen** → **Vor Bereich** steht, so heißt das nicht, dass der wiederholte Bereich nicht auf der gleichen Seite liegt. Reichlich verwirrend, aber dann etwas klarer, wenn der Aufbau des Berichtes klar wird: Der wiederholte Bereich liegt in der Kopfzeile, der erzwungene Seitenumbruch in der ersten Tabelle ganz oben auf der Seite (aber unterhalb der Kopfzeile).

Der Bereich **Name** → **'NewPage'** wird bei der Ausführung des Berichts im Writer zu Tabellen mit der Bezeichnung 'NewPage', 'NewPage0', 'NewPage1' usw. Für jede Gruppe wird also durchnummeriert ein entsprechender Tabellename erstellt. Hier greift das Makro an, das anhand des Tabellennamens die Tabelle ansteuert und dort dann zu dem Umbruch noch die Änderung der Seitennummer hinzu fügt.

```

001 SUB ReportStart
002     oReport = ThisDatabaseDocument.ReportDocuments.getByName("Report1").open
003     GroupNewPage(oReport)
004 END SUB

001 SUB GroupNewPage(oReport AS OBJECT)
002     DIM oTables AS OBJECT
003     DIM oTable AS OBJECT
004     DIM inT AS INTEGER
005     oTables = oReport.getTextTables()
006     FOR inT = 0 TO oTables.count() - 1
007         oTable = oTables.getByIndex(inT)
008         IF Left$(oTable.name, 7) = "NewPage" THEN
009             oTable.PageNumberOffset = 1
010         ENDIF
011     NEXT inT
012 END SUB

```

Der Bericht wird aus einem Formular heraus gestartet. Name des Berichts ist hier schlicht «Report1». Beim Start des Berichts wird das entsprechende Dokument direkt als Variable **oReport** zur Weitergabe an die entsprechende Prozedur **GroupNewPage** eingelesen.

Diese Prozedur holt sich aus dem erstellten Writer-Dokument alle Tabellendokumente (Zeile 5). In diesen Tabellendokumenten wird dann nach denen gesucht, die mit dem Namen «NewPage»

beginnen (Zeile 8). Für diese Tabellen wird dann die Seitenzahl für den definierten Seitenumbruch auf 1 gestellt (Zeile 9). So startet jede Gruppe mit einer neuen Seitenzahl 1.

Allerdings ist die Gesamtzahl der Seiten hiervon nicht berührt. Die Gesamtzahl der Seiten für eine Gruppe lässt sich auch in einem Writer-Dokument nicht so ohne weiteres ermitteln. Hier steht immer nur die Gesamtzahl der Seitenzahlen für das ganze Dokument.

Nachträgliche Bearbeitung des Berichtsdokuments

Berichte, die über den Writer erstellt werden, sind Dokumente mit vielen Texttabellen. Die Bearbeitung dieser Dokumente ist zwar möglich, aber doch recht mühselig. Einfacher ist es oft, den Bericht mit Arbeitsanweisungen für die nachträgliche Bearbeitung über Makros zu starten. Damit lassen sich auch Fehler, die der Report-Designer macht, ausgleichen.

Das folgende Makro zeigt einen solchen Zugriff auf die Tabelle eines Berichtes. Es ändert in der Tabelle mit der Bezeichnung «Detail» die Rahmen um die Tabelle und die Zellen. Hierfür gibt es im Bericht sonst keine Einstellung. Es ist lediglich möglich, mit viel Mühe horizontale und vertikale Linien einzufügen. Aber Achtung: Um alle Zellen werden mit dem Makro Rahmen gezogen. Wenn der Detail-Bereich z.B. höher ist als die darin enthaltenen Felder für den Text, dann gibt es entsprechend viele Linien.

```
001 SUB TableBorder
002     DIM oReport AS OBJECT
003     DIM oTables AS OBJECT
004     DIM oTable AS OBJECT
005     DIM inT AS INTEGER
006     DIM inI AS INTEGER
007     oReport = ThisDatabaseDocument.ReportDocuments.getByName("NameDesBerichts").open
008     oTables = oReport.getTextTables()
009     FOR inT = 0 TO oTables.count() - 1
010         oTable = oTables.getByIndex(inT)
011         IF Left$(oTable.Name, 6) = "Detail" THEN
012             oBorder = oTable.Tableborder
013             oBorderline = oBorder.TopLine
014             oBorderline.outerlinewidth = 1
015             oBorderline.innerlinewidth = 0
016             oBorderline.linedistance = 0
017             oBorderline.color = 0
018             oBorder.Topline = oBorderline
019             oBorder.Bottomline = oBorderline
020             oBorder.Leftline = oBorderline
021             oBorder.Rightline = oBorderline
022             oBorder.Horizontalline = oBorderline
023             oBorder.Verticalline = oBorderline
024             oBorder.Distance = 1
025             oTable.Tableborder = oBorder
026         ENDIF
027     NEXT inT
028 END SUB
```

Die Tabellen haben dabei die gleichen Namen wie die Bereiche im Report-Designer. Gegebenenfalls kann einfach ein Bericht erstellt werden und der Tabellename daraus ausgelesen werden. An dieser Stelle stehen alle Möglichkeiten für **oTables** zur Verfügung, die unter https://api.libreoffice.org/docs/idl/ref/servicecom_1_1sun_1_1star_1_1text_1_1TextTable.html verzeichnet sind.

Andere Formen der Berichtserstellung

Der Report-Designer ist zur Zeit die einzige Möglichkeit, über die grafische Benutzeroberfläche direkt Berichte zu gestalten. Nur wenn der Report-Designer nicht installiert ist, kommt eine ältere Möglichkeit der Berichtserstellung zutage.

Das alte Berichtsmodul

Bevor der Report-Designer entwickelt wurde, ist ein anderes Modul für die Erstellung von Berichten entwickelt worden. Dieses Modul kann Berichte gruppiert darstellen, ist aber nur zu tabellarischen Berichten in der Lage. Ist der Report-Designer nicht mit LO installiert worden, so steht dieses Berichtsmodul weiterhin für die Erstellung von Berichten zur Verfügung. Ein Aufruf der so erstellten Berichte ist auch mit installiertem Report-Designer weiterhin möglich.

Die BaseReportExtension

Unter <https://extensions.libreoffice.org/extensions/basereportextension> stellt Georg Mößlacher eine Berichtsmöglichkeit zur Verfügung, die er seit Jahren selbst nutzt. Diese Berichtsmöglichkeit benötigt eine Writer-Vorlagendatei (Endung: .ott). Diese Vorlagendatei wird mit mindestens einer zweizeiligen Tabelle an beliebiger Stelle versehen, die über den Tabellennamen aus der Extension heraus gefunden und mit Inhalten gefüllt wird. Die erste Zeile der Tabelle enthält die Spaltenbeschriftungen, in den nachfolgenden Zeilen wird der Inhalt eingefügt. Fehlende Tabellenzeilen werden über die Extension hinzugefügt.

Mit der BaseReportExtension kann auf den Inhalt geöffneter Formulare Bezug genommen werden. Auch lässt sich der Inhalt der Ausgabe über eine Input-Box filtern. Die Extension ist damit vor allem für so etwas wie die Erstellung von Rechnungen gut geeignet. Die Extension ist allerdings nicht in der Lage, Berichte wie im Report-Designer zu gruppieren.

Weitere Berichtsmöglichkeiten

Weitere Möglichkeiten, einen Bericht zu erstellen, sind im Kapitel «Makros» > «Datenbankaufgaben mit Makros erweitert» > «Drucken aus Base heraus» beschrieben. Auch die diesem Handbuch beigefügten Base-Beispiele enthalten Möglichkeiten des Seriendrucks und des Rechnungsdrucks in Tabellenform mit dem Writer incl. der Erstellung eines Feldes, das beim Seitenwechsel einen Übertrag ermittelt .