sql Finance

Handbuch für Anwender

Dr. Peter Halbherr

sqlFinance - prozess-orientierte Buchführung

sqlFinance
Handbuch für Anwender

I. EINLEITUNG 4

I. GRUNDLAGEN DER BUCHFÜHRUNG 7

1. Die "doppelte" Buchhaltung 7
2. Der Beitrag von Luca Pacioli (1494) 8
3. Normalisierung der Daten 11
4. Das Datenmodell von sqlFinance 14
5. Prozessorientierte Buchführung 15

II. SCHNELLER EINSTIEG 18

1. Installation 18
2. Einrichten der Anwendung und erste Buchung 20
3. Ein- und zweiseitige Buchungen 24
4. Kontierung nach Prozessen 29
5. Anwendungsübersicht 32
6. Datenorganisation 42

III. ANWENDUNGS-MODULE 59

1. Arbeiten mit der Finanzbuchhaltung 59
2. Ein einfaches Verkaufsbeispiel 62
3. Debitoren-/Kreditoren 65
4. Kurze Einführung in die Lohnabrechnung 71
5. Vermögensverwaltung 77

IV. SCHLÜSSELFUNKTIONEN 78

1. Formulare und Kaskaden 78
2. Felder 81
3. Menüs 83
4. Ziehen mit der Maus (drag and drop) 87
5. Berichte 91
6. Tastatur-Kombinationen 93

V. SUPPORT & KONTAKT 94

VI. ANHANG 95

1. Bilanz und Erfolgsrechnung 95
2. Bericht Prozesse und Kontierung 99

sqlFinance

Handbuch für Anwender

Vorwort

Prof. Dr. Paul Weilenmann, em.

Unternehmungen (und viele NGO's) sind **komplexe Organisationen**, die bestimmte **Ziele** verfolgen. Diese Ziele sind konkret zu beschreiben und aufeinander abzustimmen in einem **Zielsystem**. Um **Massnahmen**, die der Zielerreichung dienen, **planen** und nachträglich den **Zielerreichungsgrad messen zu können**, werden **Informationen unterschiedlichster Art** benötigt.

Einen besonders wichtigen Platz nehmen dabei **geldwertmässige Informationen** ein, für deren Beschaffung und Aufbereitung das **Rechnungswesen** der betreffenden Organisation zuständig ist. Zentraler Teil des Rechnungswesens ist die **doppelte Buchhaltung**, die einerseits die **Vermögensbestände** (Aktiven und Passiven) sowie andererseits die **Ursachen für diese Veränderungen** (Aufwand und Ertrag) in **systematischer Art und Weise** festhält.

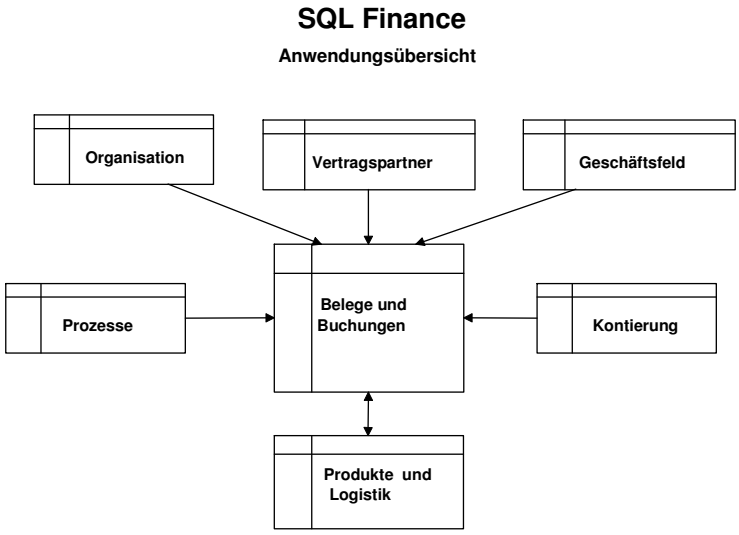
Selbstverständlich **genügt es nicht**, ein Rechnungssystem zu führen, das **lediglich** die **Erstellung von Bilanzen und Erfolgsrechnungen** ermöglicht. Vielmehr muss es die Rolle eines **valablen Führungsinstruments** übernehmen. Ausgangspunkt für die Gestaltung dieses Informationssystems ist die **Abklärung und Beschreibung** der sich aus den **strategischen und operativen Zielsetzungen** ergebenden **Informationsbedürfnisse**. Dieses **Feststellen** der **Art und des Umfanges der Informationen** ist eines, das **Generieren** unterschiedlicher konkreter **Informationen** aus dem **intelligent gestalteten Buchhaltungssystem** ein anderes.

Mit seinem **innovativen Lösungsansatz** knüpft **sqlFinance** an diesem Punkt an. Einerseits stützt es sich auf die von der modernen Wissenschaft beschriebenen **theoretischen Grundlagen der doppelten Buchhaltung** und andererseits auf die **Möglichkeiten der modernen Computertechnologie**. Besondere Aufmerksamkeit wird den wichtigen Ansprüchen bezüglich **Effektivität und Effizienz** gewidmet. Der **Aufbau der Finanzbuchhaltung** nach **Prozessen** ergibt einen hohen Grad an **Flexibilität** im Hinblick auf die **Auswertungen**, z.B. nach **Vertragspartnern, Geschäftsfeldern oder Organisationseinheiten**.

I. Einleitung

Beim Einsatz der Informatik im Personalwesen ist es entscheidend, **Kunden- und Kandidatenkontakte richtig abzubilden**. Dies betrifft sowohl die Personalrekrutierung mit ihren komplexen Kontaktmustern wie auch die Personaladministration. sqlFinance ist in diesem Umfeld entstanden und in verschiedenen Schritten von einer UNIX-basierten Datenbank zu einer prozessorientierten Client-Server-Anwendung weiterentwickelt worden.

Von Kernfunktionen wie "**Customer Relationship Management**" (CRM) und Belegverarbeitung ausgehend wurde **sqlFinance** schrittweise ausgebaut zu einer umfassenden Informatik-Anwendung, die allgemein für die **Abbildung und Steuerung von Geschäftsprozessen** eingesetzt werden kann. Die objekt-orientierte Oberfläche erlaubt den gleichzeitigen **Zugriff auf lokale und Server Datenbanken**. Mehrere **Mandate** pro **Datenbank** können eröffnet werden mit verschiedenen **Kontenplänen**, aber doch **gemeinsamen Kundendaten**. Auf verschiedenen Servern **dezentral erfasste Daten** können zudem für Abschlussberichte über Replikationsinstrumente jederzeit zusammengeführt oder ausgetauscht werden.



Als einzigartige **Innovation** bietet **sqlFinance** den Aufbau der **Finanzbuchhaltung** nach **Prozessen**. Gleichartige, in **Belegen** gruppierte Buchungen werden **Prozessen** zugeordnet. Dies führt zu bedeutenden **Rationalisierungseffekten**, nicht nur bei der Datenerfassung, sondern auch bei der **Auswertung** und insbesondere bei der **Revision**.

In der Datenerfassung lassen sich die gleichartigen Buchungsmuster leicht **kopieren** und **wiederholt verwenden**. Mittels **Drag-and-Drop** können aus alten Vorlagen **neue erstellt** und anschliessend **angepasst** werden. Nach diesen Prozessen erstellte **Berichte** zeigen zusammenfassend die **Buchungsmuster** und können leicht auf **Konsistenz** und **Fehler** überprüft werden.

Die **Revision** der Buchungen kann **prozessweise** erfolgen. Dies macht nicht nur die **Fehler** leichter ersichtlich. Es ist auch möglich, in der

Prozessdokumentation direkt Informationen über die **Verbuchungs-richtlinien** abzulegen und diese dann mit der **Buchungspraxis** zu vergleichen (zum Beispiel bei Bewertungs- und Abschreibungsfragen).

sqlFinance strukturiert ergänzend Buchungsdaten nach **Vertragspartnern**, **Geschäftsfeldern** und **Organisationseinheiten**. Als Vertragspartner können **sämtliche vertraglichen Beziehungen** erfasst werden, wie zum Beispiel zu Kunden, Lieferanten, Mitarbeitern, aber auch Bankkonten, Mietverträge, Versicherungspolicen et cetera. Das Geschäftsfeld dient der Gliederung nach **Geschäftstätigkeit**, also zum Beispiel Grossprojekte oder Marktsegmente. Die **Organisationseinheit** dient demgegenüber zur Darstellung **interner Strukturen** wie Profit-Center oder Kostenstellen.

Von der Finanzbuchhaltung ausgehend kann so ein **wirkungsvolles Controlling** aufgebaut werden. Je nach Bedarf können Sie die Auswertung mehr auf die Überwachung der **vertraglichen Beziehungen**, den **Erfolg der Geschäftstätigkeit** oder die **Leistungsfähigkeit einzelner Organisationseinheiten** ausrichten.

sqlFinance unterstützt Berichte für noch **offene** oder bereits **abgeschlossene Perioden**, je nach Wunsch des Anwenders. Auf der Buchungszeile können **Fremdwährungen**, **Rundungen**, **Abrechnungsspesen** sowie **Mehrwert-steuern** und andere Abgaben abgerechnet werden. Ein Modul für die **Gehaltsabrechnung** unterstützt wiederkehrende Zahlungen an die Mitarbeiter sowie die automatische Erzeugung von Buchungen für die **Abrechung der Soziallasten**. Die **Stundenabrechnung** ermöglicht ergänzend die Verwaltung des **Arbeitsaufwands** pro Geschäft und Organisationseinheit.

In den **Verkaufs-** und **Einkaufsmodulen** können **Produkte** mit **Mengenangaben** verwaltet und zu **Stücklisten** zusammengefasst werden. Die **Fakturierung** im Verkaufsmodul ermöglicht die Berechnung verschiedener **Steuersätze** in derselben Rechnung. Verschiedene Adressen können für die **Rechnungsstellung** und **Lieferung** verwendet werden. In der Server-Version wird auch die Führung von **Lagerbeständen** und die **Abwicklung** von Lieferungen unterstützt.

Der **Übersetzung in verschiedene Sprachen** wurde von Anfang an besondere Beachtung geschenkt. Sämtliche Sprachanzeigen werden aus einer **Datenbank** generiert. Eine Software für die Sprachverwaltung (**Language Manager**) vereinfacht die Portierung in eine weitere Sprache.

Eine **Gratisversion** kann auf unserer Homepage (**www.sqlfinance.com**) herunterladen werden. Diese bietet Ihnen einen **einfachen Einstieg** in die Software. Ein **IFRS Kontenplan** ist bereits enthalten. Auf der Buchungszeile ist die Erfassung von Fremdwährungen, Organisationseinheiten (Kostenstellen), Geschäftsfeldern und Vertragspartnern möglich. Die Gratisversion erlaubt eine **unbeschränkte Anzahl Buchungen**, ist aber auf den **Einzelplatz** beschränkt.

sqlFinance wurde bereits erfolgreich eingesetzt für die Buchführung von Schweizer Unternehmen mit

PRICEWATERHOUSECOOPERS

als **Revisionsstelle**. Revisionen sind ebenso von der staatlichen Schweizer Altersvorsorge (AHV) durchgeführt worden. Bilanz und Erfolgsrechnungen wurden von den kantonalen Steuerbehörden akzeptiert.

In einer fachtechnischen Beurteilung von **Prof. Dr. Beatrice Meyer**, Dozentin an der **Hochschule für Betriebswirtschaft in Winterthur**, erhielt **sqlFinance**

uneingeschränkte Anerkennung, sowohl für den innovativen Ansatz der Buchführung wie auch für die objektorientierte Gestaltung der Software.

Die **Datenmodellierung** mit Beleg und Buchungen sowie die Handhabung **einseitiger** und **zweiseitiger** Buchungen wurde auch mit **Prof. Dr. Paul Weilenmann**, emeritierter Dozent für Betriebswirtschaft an der **Universität Zürich,** überprüft und von ihm als fachlich absolut korrekt beurteilt.

sqlFinance hat bisher ausnahmslos **alle Anforderungen und Prüfungen** von Fachexperten, Behörden und Revisoren **erfüllt**.

Wir möchten **Herrn Prof. Weilenmann** an dieser Stelle unseren besonderen Dank aussprechen. Sein Beitrag im theoretischen Bereich brachte Klarheit in vielen Schlüsselfragen und bestärkte uns, den eingeschlagenen Weg weiter zu beschreiten und das Produkt zur vollen Marktreife zu bringen.

Besonders danken möchte ich aber auch meiner Frau **Silvie Halbherr-Zehnder**, die mich über die lange Zeit immer wieder motiviert hatte, an der Innovation festzuhalten und die Arbeit bis zum Abschluss weiterzuführen.

Für die Unterstützung in fachtechnischen Fragen danke ich meinem Informatik-Partner **Urs Bretscher** sowie **Dario Scheuber**, der als Finanzchef wertvolle Erfahrung aus der „Linie" einbrachte.

Danken möchte ich auch den Kunden und Mitarbeitern von **IPAG Inter Personal AG**, in deren Umfeld diese „Diversifikation" entstehen konnte. Unterstützt haben mich Saskia Wolf bei der Redaktion der Manuale und den abschliessenden Einsatztests sowie **Thomas Ruckstuhl** bei der Lösung vieler systemtechnischer Probleme im Umfeld von Microsoft SQL Server und ODBC.

Nicht unerwähnt bleiben sollten an dieser Stelle ferner auch die zahlreichen **Interview-Partner** aus den Bereichen **Wirtschaftsprüfung** und **Informatik**, die in vielen Fällen bereit waren zu einem spannenden fachtechnischen Austausch und mir wertvolle Informationen aus der Praxis gaben.

sqlFinance verfolgt ein **innovatives Konzept der Buchführung**. Mein Ziel war aber immer, nicht nur die **Theorie** zu liefern, sondern auch den praktischen Nutzen mit einem **einsatzbereiten Software-Produkt** zu belegen**.** So kann sich der Anwender anhand **konkreter Beispiele** zuerst mit der besonderen **Funktionalität** und den vielen **Auswertungsmöglichkeiten** vertraut machen. Dann ist die Türe offen für den **produktiven Einsatz** und eine **neu zu entdeckenden Freude** an der **Effizienz** und **Aussagekraft** der **Buchführung**.

Dr. Peter Halbherr, Bern Mai 2009

I. Grundlagen der Buchführung

1. Die "doppelte" Buchhaltung

Buchhaltung ist **Mathematik**. Einfachste Mathematik, könnte man denken. Die häufigste Operation ist die Addition: **Aufwand und Ertrag** werden separat zusammengezählt, die Differenz ergibt den **Gewinn**. Vielleicht gerade weil dies so einfach erscheint, findet man in der Fachliteratur wenige Angaben, welchen **mathematischen Grundanforderungen** ein Buchführungssystem genügen muss.

Ferner gibt es in der Praxis verschiedene **Buchführungsstile**, die sich insbesondere in der **Art der Datenerfassung** unterscheiden. In Europa wird auf einer Buchungszeile meistens nicht nur ein **Konto** und ein Betrag, sondern auch ein **Gegenkonto** erfasst. In den USA ist dagegen die **einfache Buchungszeile** mit nur einem **Konto** und einem **Betrag** verbreitet. In gewissen Fällen werden die Buchungszeilen mit einem gemeinsamen Gegenkonto (zum Beispiel Debitoren-Sammelkonto) verbunden, was eine **Mischform** zwischen den zwei Erfassungsarten ergibt.

Wir möchten die erstgenannte Variante als „**doppelseitige Buchung**" („double sided booking" beziehungsweise „DSB") bezeichnen, die Zweitgenannte als „**einseitige Buchung**" („single sided booking" beziehungsweise „SSB"). **sqlFinance** unterstützt **beide Erfassungsarten**. Damit ist es möglich, in **demselben Mandat** in einer ersten Phase noch den doppelseitigen Buchungsstil weiterzuführen und sich gleichzeitig mit den **Vorteilen** einseitiger Buchungen vertraut zu machen.

Die **„doppelte Buchhaltung"** („double entry book-keeping" oder „double entry accounting") ist im **Mittelalter** entstanden. Zu dieser Zeit wurden alle Daten noch von Hand auf **Papier** erfasst. Es ist offensichtlich, dass sich mit dem Einsatz der **Informatik** und insbesondere der **relationalen Datenbanken** heute andere Möglichkeiten ergeben.

Um den Ansprüchen der Regulatoren und der in dieser Frage vielfach unsicheren Kunden zu genügen, haben viele Hersteller die **alten Gepflogenheiten** in ihre Softwarelösungen **übernommen**. Möchte man aber das **Potential der Informatik** besser nutzen, ist es nötig, die **mathematischen** Anforderungen an ein Buchführungssystem klarer zu formulieren und das überlieferte **Regelwerk** in dieser Hinsicht zu **überprüfen**. Dabei lässt sich feststellen, dass dem Buchführer **verschiedene Lösungsmöglichkeiten** offen stehen, die es in Bezug auf **Vor- und Nachteile** in der Praxis sorgfältig **abzuwägen** gilt.

Es lohnt sich deshalb, sich die **historische Entstehung der doppelten Buchhaltung** ebenso wie die mathematischen Anforderungen an ein **Buchführungssystem** zu vergegenwärtigen.

2. Der Beitrag von Luca Pacioli (1494)

Die Historiker sind sich einig, dass die doppelte Buchhaltung auf den italienischen Franziskanerpater **Luca Pacioli** (1445-1517) und seine Publikation aus dem Jahre 1494 zurückgeht. In seinem Werk **„Summa"** lieferte der in der Zeit Leonardo da Vincis ebenfalls bekannte **Mathematiker** eine Zusammenfassung der mathematischen Methoden seiner Zeit. Im berühmten Tractatus XI mit dem Titel „Particularis de computis et scripturis" beschreibt Pacioli die **Regeln der Buchführung**, wie sie damals von den norditalienischen **Handelshäusern** eingesetzt wurde.

Typisch für die Buchführung dieser Zeit ist eine grosse **Redundanz** in der **Datenerfassung**. Der Geschäftsvorfall wurde zuerst notizenartig in einem **Memorial** erfasst. Von diesem Protokoll ausgehend wurde später von einer Fachperson eine Buchung mit Angabe von **Gläubiger** und **Schuldner** („Soll Geben" und „Soll Haben") erstellt und in ein **Journal** eingetragen. In einem dritten Schritt wurde diese Buchung dann in das **Hauptbuch** übertragen, das für jeden Schuldner und Gläubiger ein eigenes **Kontoblatt** enthielt.

Luca Pacioli: Summa de arithmetica, geometria, proportioni et proportionalità. Toscolano, 1523 (2. Ausg, Signatur: 72520)

Eigene Kontoblätter waren nicht nur für die Abrechnung mit den **Vertragspartnern** (Kunden und Lieferanten) vorgesehen, sondern auch für die **Lagerführung** der **Handelsbestände**. Ergänzend wurde der **Betriebsaufwand** auf einigen Konten zusammengefasst, ebenso die **Barbestände** auf der Schuldnerseite und das eingesetzte **Kapital** des Unternehmens auf der Gläubigerseite.

Durch die **Verwendung der Buchungszeile** versuchte man sicherzustellen, dass zu jeder Veränderung auf **Gläubigerseite** immer auch eine gleiche Veränderung auf **Schuldnerseite** gehörte. Entstanden bei der Verbuchung **Abweichungen**, so waren diese sofort als **dem Geschäftsvorfall zuzuordnender Gewinn oder Verlust** auszuweisen und auf das entsprechende Sammelkonto **„Gewinne und Verluste"** zu verschieben, das beim Abschluss mit dem **Eigenkapital** verrechnet wurde. Solche Abweichungen ergaben sich typischerweise im **Lager**, wenn der Verkaufspreis höher oder tiefer war als der

Einkaufspreis und diese Abweichung sofort dem **Betriebsgewinn zuzuordnen** war.

Das **Ziel der Buchführung** war demzufolge nicht einfach, Schuldner und Gläubiger in ausgeglichener Position zu halten, sondern den **Ertrag** für **jeden einzelnen Geschäftsvorfall** festzustellen und sofort mit dem **Gewinn- und Verlustkonto** beziehungsweise dem Eigenkapital zu **verrechnen**. Interessanterweise liefert Pacioli aber **keine Vorschläge für die Zusammenfassung** der Gewinne oder Verluste pro Kunde, was die Grundlage für ein einfaches **Controlling** hätte ergeben können.

Gemäss Pacioli „stimmt" eine Buchhaltung dann, wenn die **Summen auf Gläubiger- und Schuldnerseite** getrennt addiert den gleichen (positiven) Betrag ergeben. Werden die Beträge auf der Gläubigerseite als negative Werte verrechnet, „stimmt" die Buchhaltung, wenn die **Summe aller Schuldner- und Gläubigerwerte** gleich **null** ist.

Die **doppelte Buchhaltung** des Mittelalters kann deshalb definiert werden als eine **Liste von Buchungswerten**, deren **Summe null** ergibt. Stellt man die positiven Beträge unter der Sammelbezeichnung **„Aktiven"** dar und die negativen unter der Sammelbezeichnung **„Passiven"**, erhält man eine **Bilanz** mit der **Summe null**.

Bereits bei **Pacioli** ist gut ersichtlich, wie beim Buchen in der Bilanz **nicht weiter erklärbare Wertveränderungen** besonderen **Aufwand-** und **Ertragskonten** belastet oder gutgeschrieben werden und sich durch deren Summierung der **Gewinn** oder **Verlust** beim **Abschluss** einer Periode errechnen lässt.

Wie **Paul Weilenmann** (Grundlagen des Betriebswirtschaftlichen Rechnungswesens, Sauerländer/SKV 1988, Seite 21) treffend ausführt, ist es nahe liegend, den **nicht weiter erklärbaren Vermögenszugang oder –Abgang** von der Bilanz abzugrenzen und als **„Aufwand"** und **„Ertrag"** in einer separaten **„Erfolgsrechnung"** aufzuführen. So erhält man die heutige typische Aufteilung in **Aktiven, Passiven, Aufwand und Ertrag**.

Mit der zunehmenden Komplexität der kaufmännischen Aktivitäten nahm die Zahl sowohl der **Buchungsobjekte** (man begnügte sich nicht mehr mit Konten für Schuldner und Gläubiger) als auch der **Buchungstatbestände** zu. Das drückte sich in der Buchhaltung darin aus, dass sich immer mehr Buchungstatbestände ergaben, bei denen sich nur ein Aktivum oder nur ein Passivum veränderte und keine Gegenwirkung auf einem anderen dieser Bestandkonten vorhanden war bzw. nicht festgestellt wurde.

Es war nun das grosse Verdienst **Paciolis**, dass er die diese nicht vorhandenen oder nicht festgestellten Gegenwirkungen nicht einfach als Summe ermittelte, sondern sie nach Ursachen für die jeweilige einseitige Bestandesveränderung sammelte, als **Aufwand** (einseitige Verminderung eines Aktivums oder Zunahme eines Passivums) oder als **Ertrag** (einseitige Vermehrung eines Aktivums oder Veränderung eines Passivums).

Die moderne **Buchhaltungstheorie**, die der heutigen Praxis zugrunde liegt (s. dazu Paul Weilenmann, ebenda) erklärt:

- **Aufwand** ist Geld-, Sachgüter- oder Dienstleistungsabgang (oder Passivzunahme) ohne oder ohne festgestellten Zugang.
- **Ertrag** ist Geld-, Sachgüter- oder Dienstleistungszugang (oder Passivabnahme) ohne oder ohne festgestellten Abgang.

Nicht feststellbar auf einem zweiten Bestandeskonto sind z.B. Aufwände wie Lohnzahlungen, Abschreibungen auf Anlagevermögen oder freiwillige Zuwendungen (NGO, zusätzliche Einzahlungen an Pensionskassen etc.) bzw.

Erträge wie Verkauf von Dienstleistungen oder Wertvermehrungen von Finanzanlagen.

Das vorliegende Handbuch stützt sich auf diese Grundlage und erlaubt somit sowohl Buchungen, bei denen konsequent jeder Soll-Buchung eine zahlenmässig gleiche Haben-Buchung gegenübersteht, als auch ein- oder beidseitige Summierungen, wobei die zu verbuchenden Beträge mit den **Rechenzeichen (+) und minus (-)** bezeichnet werden. Die **Summe** der Eintragungen ergibt somit **immer null**.

(persönliche Mitteilung, Beitrag von **Paul Weilenmann**, Zürich, den 29. April 2009)

Erlaubt man nun Buchungen **zwischen** Bilanz und Erfolgsrechnung, entsteht eine **Differenz** zwischen **Aktiven und Passiven** und die **umgekehrte Differenz** zwischen **Aufwand und Ertrag**. Diese Differenz bezeichnen wir als **Gewinn** beziehungsweise **Verlust**. Er bezeichnet die Summe der **nicht weiter erklärbaren Differenz** zwischen **Aufwand** und **Ertrag** (s. dazu Paul Weilenmann, ebenda).

Bilanz	**Erfolgsrechnung**
erklärbare Zu- und Abgänge	nicht erklärbare Zu- und Abgänge
(1) + Aktiven	(3) + Aufwand
(2) - Passiven	(4) - Ertrag
= **+ Gewinn(Bilanz)**	**= - Gewinn (Erfolgsrechnung)**

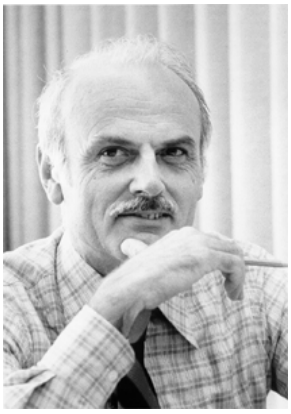
Bilanz und Erfolgsrechnung mit erklärbaren und nicht erklärbaren Zu- und Abgängen

Ein **Buchführungssystem** ist damit minimal **definiert als eine Liste von Buchungen**, bestehend aus mindestens einer Angabe von **Konto** und **Betrag**, deren **Summe null** ergibt, wobei die Konten den vier verschiedenen Typen (**Aktiven, Passiven, Aufwand und Ertrag**) zugeordnet sind, und der **Gewinn** in der **Bilanz** als **Summe** von **Aktiven + Passiven** beziehungsweise der **negative Gewinnbetrag** in der **Erfolgsrechnung** als **Summe** von **Aufwand + Ertrag** berechnet wird. Buchungen können in beliebiger Art zugefügt werden, vorausgesetzt dass die **Summe** der zugefügten Buchungen ebenfalls **null** beträgt.

3. Normalisierung der Daten

Ein Buchführungssystem ist so gut wie das zugrunde liegende Datenmodell. Eine gute Datenorganisation vereinfacht nicht nur die Eingabe, sondern erhöht die **Flexibilität** und **Aussagekraft der Berichte**. Es ist deshalb erstaunlich, wie wenig die **Datenmodellierung** als **Innovationsmöglichkeit** auch in der Finanzbuchhaltung aufgegriffen worden ist.

Die nach mathematischen Kriterien **optimierte Organisation der Daten** ist eine Grundvoraussetzung für den effizienten Einsatz von relationalen Datenbanken und wird als **Normalisierung** bezeichnet. Die grundlegenden Konzepte wurden von **Edgar F. Codd** formuliert (E.F.Codd, „A Relational Model of Data for Large Shared Data Banks", Comm.ACM 13 (6), June 1970, pp. 377-387). In der Arbeit mit grossen Datenbanken entwickelte er seine **Normalisierungsregeln**, die bis heute breite Akzeptanz finden.



Codd, E.F. (1970). "A Relational Model of Data for Large Shared Data Banks". *Communications of the ACM* 13 (6): 377–387

Codd erkannte früh, dass sich durch die **Normalisierung** der Daten eine **universelle Datenbanksprache** entwickeln liesse. Die Anwendung konnte so eine **maximale Unabhängigkeit** von den Daten erreichen. Ferner wurde die **Redundanz** weitgehend **eliminiert**. So konnten **Mehrfacheinträge**, die wiederum **komplizierte Verwaltungsprogramme** nach sich ziehen würden, **vermieden** werden.

In der **Praxis** wird die maximale Eliminierung von Redundanz allerdings **nicht konsequent** umgesetzt. Ein wichtiger Grund ist der **Zeitfaktor**. Bei einer Faktur zum Beispiel ist es wichtig, die in der Rechnung verwendete **Anschrift des Kunden** vollumfänglich **dokumentiert** zu haben. Es wäre datentechnisch **nicht richtig**, wenn sich eine **Adressänderung** noch auf **bereits verschickte Rechnungen** auswirken würde.

Ein weiterer einschränkender Faktor ist der **Applikationsaufwand**. Konsequent **normalisierte Daten** führen zu **sehr vielen kleinen Tabellen**, die für die Anwendungen über sog. **„Views"** (Sichten) zusammengefasst werden. Der **Einsatz von Views** verlangt aber wiederum eine entsprechende **technische Grundlage** bei der eingesetzten **Datenbank** und ist relativ **kompliziert** in der Realisierung.

Als **drittes Element** ist ferner festzuhalten, dass **Codd**'s Überlegungen sich vor allem auf die **Auslagerung der Stammdaten** beziehen, die optimalerweise aus einem **Index** und **beschreibenden Daten** bestehen (zum Beispiel Adress-Nr. und Adressdaten). Die im Kontext der **doppelseitigen Buchung** wichtige Frage

nach der Handhabung **mehrfacher Indexe** in **derselben Transaktionsdatei** bleibt hier offen.

Folgt man der Argumentation von Codd, bietet sich eine weitere **Normalisierung** der **doppelseitigen** Buchungen in **einseitige** Buchungen an. Bereits Codd's erste **Normalisierungsstufe** verlangt, dass Wiederholungen gleichartiger Daten in der Datenstruktur zu vermeiden sind:

Statt einer Liste mit der **Struktur A** (doppelseitig):

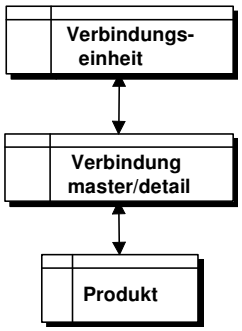
- Beleg, Konto1, Konto2, Betrag

Erstellt man besser eine Liste mit **Struktur B** (einseitig):

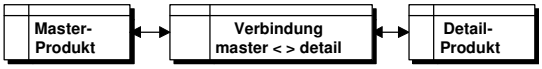
- Beleg, Konto1, Betrag
- Beleg, Konto2, Betrag

Bei der Buchhaltung ist dies besonders nahe liegend, da die **Buchungen** einem **Beleg** zugeordnet sind. Folgt man der Liste mit **Struktur A**, ergeben sich sofort **Probleme** beim Aufruf von **Druckprogrammen**. Wie soll man mit einer solchen **Datenstruktur** einen **Bericht** mit Bilanz oder Erfolgsrechnung, oder auch nur einen einfachen **Kontoauszug** erstellen? Bei jeder Abfrage muss überprüft werden, ob Konto1 oder Konto2 dem Ablagekonto entsprechen, und zudem im ersten Fall der **positive Betrag** und im zweiten der **negative Betrag** für die Berechnung des Totals eingesetzt werden.

Dieser Aspekt der Datenmodellierung ist aus der Programmierung von **Stücklisten**, sog. „**Bills of Material**" gut bekannt. Der Datensatz einer Stückliste enthält typischerweise **zwei Produkte-Nummern**, eine für das **übergeordnete** Produkt und eine für das **untergeordnete**. Dies ist nötig, da jedes Produkt mehrfach einem anderen Produkt untergeordnet werden kann (vgl. zu diesem Thema auch: Carl August Zehnder, Informationssysteme und Datenbanken, B.G.Teubner, Stuttgart 1989, insbesondere Seiten 60 f.).



vollständig normalisierte Stückliste



unvollständig normalisierte Stückliste mit Master- und Detail-Index

Wird diese sogenannte **„Many-to-Many"-Verbindung** vollständig **normalisiert**, muss in Ergänzung zu den **Verbindungselementen** ein übergeordnetes **Sammelgefäss** für die **Verbindungseinheit** in der **Datenbank erfasst** werden. Dies ermöglicht dann die Erfassung **einseitiger** Verbindungen zu den Produkten. Erst durch diese **vollständige Normalisierung** kann auch die Produktion von verschiedenen **Varianten** korrekt dargestellt werden. Für die Verbindung zwischen Master und Detail stehen dann **verschiedene Verbindungseinheiten beziehungsweise –Möglichkeiten** zur Verfügung.

In der Praxis wird vielfach versucht, diesen etwas aufwändigen Stil zu **umgehen**, da er nicht geeignet ist für Anwendungen mit der Computermaus, bei der zum Beispiel **Detailprodukte** einfach über ein **Masterprodukt „gezogen"** werden können. Man kann also die Anwendung **vereinfachen**, erhält dann aber einen **Mehraufwand** bei den **Auswertungen** (hier zum Beispiel bei der Berechnung der **Lagerbewegungen**).

In Anbetracht der heutigen technischen Möglichkeiten hat man in der **Buchhaltung** aber wenig Interesse, den **„Beleg"** wegzulassen. Im Gegenteil, für die Revision und das Verständnis des Verbuchens ist es äusserst **wertvoll**, eine **Anzahl Buchungen einem Buchungsbeleg zuzuordnen**. Dies entspricht auch der Darstellungspraxis auf Papier. Dabei ist nicht wesentlich, ob dem Beleg **einseitige oder zweiseitige** Buchungszeilen zugeordnet werden. Es gilt lediglich sicherzustellen, dass die **Summe der übergebenen Buchungen** ein **Total von null** ergibt. Bei **zweiseitigen** Buchungen ist diese Anforderung prinzipiell **immer** erfüllt, bei **einseitigen** Buchungen kann man den Beleg **erst abschicken**, wenn dieses Kriterium **erfüllt** ist.

Besonders interessant ist auch die **Analyse** der Kontierung unter dem Gesichtspunkt der **Normalisierung**. Tatsächlich beinhalten fast alle Kontenpläne mehrere **Dimensionen**, die zu einem schwierig zu handhabenden Register **zusammengefasst** werden. Ein Kontenplan enthält vielfach **verschiedene Elemente** der internen Organisation, der Geschäftstätigkeit und der Vertragspartner. Dadurch wird die Buchführung jedoch schnell **unübersichtlich** und es muss viel **Aufwand** betrieben werden, um die **Kontenvielfalt** in der Erstellung der **Berichte** wieder geeignet **zusammenfassen** zu können.

Die **Komplexität** der Kontenpläne ist schon in den Ausführungen **Paciolis** ersichtlich. War die **ursprüngliche Idee** des Kontos die einer **Kundenabrechnung** (also der vertraglichen Beziehung zum Schuldner oder Gläubiger), so werden alsbald **virtuelle** Schuldner/Gläubiger für Warenbeschaffungen, Warenlager, bald aber auch allgemeiner Betriebsaufwand und Gewinne/Verluste eingeführt.

Es ist deshalb wesentlich effizienter, auch in der **Finanzbuchhaltung** ein kohärentes **Datenmodell** einzusetzen. Damit kann der **Kontenplan** schlank gehalten und auf das **finanzielle Berichtwesen** ausgerichtet werden. Details zu **Vertragspartnern**, **Geschäftsfeldern** oder **Organisationseinheiten** können je nach Bedarf **ergänzend abgefragt** werden.

4. Das Datenmodell von sqlFinance

Die **konsequente Datenmodellierung** bietet eine gute **Grundlage** für spätere Auswertungen und ein **wirkungsvolles Controlling**. Normalisierte Buchungen eignen sich besonders für die Bearbeitung mit statistischen Methoden und können als ein **multivariates System** betrachtet werden. Eine Anzahl **Parameter** (Konto, Organisation, Vertrag, et cetera) beeinflusst als **resultierende Variable** den **Betrag**. Wir erhalten so auf Anhieb einen mehrdimensionalen Datenwürfel (**Data-Cube)**, wie er in der **„Business Intelligence"** verwendet wird. Der Data-Cube enthält die nachfolgenden wichtigsten **Parameter**:

- **Vertrag** (contract): der **Vertragspartner** (Kunde, Lieferant, Mitarbeiter)
- **Geschäft** (business): das einzelne **Kundenprojekt** oder eine Zusammenfassung von kundenorientierten Aktivitäten (in diesem Fall Geschäftsfeld)
- **Organisation** (organisation): die interne Organisation des Unternehmens, vielfach auch als **Kostenstellenstruktur** bezeichnet
- **Produkt** (product): das materielle **Produkt** oder die **Dienstleistung**, mit zugehöriger **Produktelinie** sowie **Masseinheiten** und **Preisen**

Die Möglichkeit, bei allen Buchungen einen **Vertragspartner** anzugeben, bringt eine wesentliche **Entlastung des Kontenplans**. Da die Vertragspartner in eine eigene Referenzdatei **ausgelagert** sind, enthält der Kontenplan nur noch das **stabile Gestaltungsgerüst** für die Darstellung von **Bilanz und Erfolgsrechnung**. Die Vertragsnummer kann **ebenso** für die Verwaltung von **Bankkonten, Versicherungspolicen, Telecom-Nummern, Mieten** et cetera verwendet werden. Ferner ist zu beachten, dass die **Kundenprojekte** ausserhalb der Vertragspartner in einer weiteren **Referenzdatei** verwaltet werden. Dadurch wird es möglich, mit einseitigen Buchungen Beschaffungen **direkt auf dem Kundenprojekt** zu verbuchen (zum Beispiel den Einkauf von Material direkt auf ein Installationsprojekt bei einem Kunden).

Während die Vertragspartner die **externen** Beziehungen des **Unternehmens** abbilden, dient die **Organisation** der Strukturierung **interner** Beziehungen und Verantwortlichkeiten. Die Organisation kann eine **Kostenstelle** ebenso wie ein **Profit-Center** repräsentieren. Organisationseinheiten können mit **übergeordneten** Organisationseinheiten (zum Beispiel Abteilungen oder grösseren Verkaufsgebieten) verbunden werden. Ebenso können die **Mitarbeiter** einzeln den Organisationseinheiten zugeordnet werden. Das **Produktefeld** ist nur in den Modulen Einkauf und Verkauf verfügbar und bietet nicht nur die Möglichkeit **Material und Dienstleistungen**, sondern auch ganze **Stücklisten** zu verwalten.

sqlFinance enthält in der **Vollversion** auch **Lohn- und Stundenabrechnung**. Die **Löhne** können **Organisationseinheiten** zugeordnet werden. Der **Zwischenabschluss** der Teilperiode bringt dann die effektiven **Lohnkosten pro Organisationseinheit**, die in die **Finanzbuchhaltung** übernommen werden. Ebenso können in der **Stundenabrechnung** geleistete Stunden pro Mitarbeiter einem **Projekt** oder **Geschäftsfeld** zugeordnet und allenfalls von einer Organisationseinheit auf eine andere umgelegt werden. Auch diese Daten lassen sich als Abschluss der Teilperiode in der Form von Totalen in die **Finanzbuchhaltung** übernehmen.

5. Prozessorientierte Buchführung

Vergleicht man die erfassten **Belege** einer Buchhaltung, so wird ersichtlich, dass sich die **Buchungsmuster** wiederholen. Für den **Buchführenden** wie auch den **Revisor** ist es deshalb interessant, nicht jeden einzelnen Beleg, sondern **in Prozessen zusammengefasste Belege** zu analysieren und zu prüfen.

Felder | Liste

process	table_name	type	header
audit	PROCESS	P	external audit
booking	PROCESS	P	master booking process
business	PROCESS	P	master business process
cash	PROCESS	P	cash account in banks
clear	PROCESS	P	clearing
closing	PROCESS	P	closing in/to gen.ledger
consulting	PROCESS	P	consulting services
customer	PROCESS	P	customers (sales)
default	PROCESS	P	default process
document	PROCESS	P	documents
email	PROCESS	P	internet mail
employee	PROCESS	P	company employees

Beispiel einer Prozessstruktur

Typische Prozesse, denen Belege zugeordnet werden, sind zum Beispiel **Einkauf und Verkauf, Zahlungen, Zinsrechnungen, Kapitalerhöhungen**, aber auch nicht mit einem Geldfluss verbundene Prozesse wie zum Beispiel **Abschreibungen und Rückstellungen.** Für **jeden einzelnen Prozess** kann eine kleine **Bilanz- und Erfolgsrechnung** erstellt werden. Da der Prozess eine **homogene Struktur** hat, wird die Ausgabe nur **wenige Konten** enthalten. Damit sind **Fehlbuchungen** leicht zu identifizieren.

Prozess/Konten (Master Prozess)

Process: booking 01.07.09

Prozess	booking	master booking process	
Klasse	booking	includes as detail all booking processes (entered in the batch windows)	
Zähler	20		
Prozess	interest	bank interests and expenses	
Klasse	booking		
Zähler	3		
Konto	118050	SBC Bank Account	Bank Account Swiss Bank Corporation
Gruppe	1180	liquid assets	
Konto	341010	Finance Expenses (Bank)	\Various Finance Expenses (Bank)
Gruppe	3410	finance expenses	
Konto	441010	Finance Revenues (Bank)	Finance Revenues Bank (interest payments)
Gruppe	4410	finance revenues	
Prozess	payable	accounts payable	
Klasse	booking	payroll and infrastructure payables	
Zähler	5		
Konto	118050	SBC Bank Account	Bank Account Swiss Bank Corporation
Gruppe	1180	liquid assets	
Konto	231010	Payroll Clearing	Payroll Clearing Account
Gruppe	2310	payroll	
Konto	312011	Purchase of Goods	
Gruppe	3120	Purchase Goods	
Konto	323010	Employee Expenses	\Various Employee Expenses
Gruppe	3230	infrastructure	
Konto	323011	Infrastructure	
Gruppe	3230	Infrastructure	
Prozess	receivable	accounts receivable	
Klasse	booking		
Zähler	2		

Seite 1 01.07.09

Prozess/Konten (Master Prozess)

Process: booking 01.07.09

Prozess	booking	master booking process	
Klasse	booking	includes as detail all booking processes (entered in the batch windows)	
Zähler	20		
Konto	112010	Receivables (main)	Main Receivables Clearing Account
Gruppe	1120	accounts receivable	
Konto	118050	SBC Bank Account	Bank Account Swiss Bank Corporation
Gruppe	1180	liquid assets	
Prozess	sales	sales orders	
Klasse	business		
Zähler	4		
Konto	112010	Receivables (main)	Main Receivables Clearing Account
Gruppe	1120	accounts receivable	
Konto	223010	VAT Payables	VAT Value Added Tax Clearing Account (payables and payments)
Gruppe	2230	tax VAT	
Konto	412020	Hardware	Computer Hardware (PC)
Gruppe	4120	sales goods	
Konto	413010	Sales of Services (goods)	Installation and Maintenance
Gruppe	4130	sales of services	

Seite 2 01.07.09

Master Prozess „booking" mit den Prozessen „payable", „receivable", „interest" und „sales" sowie der zugehörigen Kontierung (s. vergrösserte Darstellung im Anhang)

Durch **Ziehen mit der Maus** können die **Konten** mit den **Prozessen** verbunden werden. Damit erhält der Anwender **prozessbezogene Buchungsmuster**, die Ordnung in die **Kontierung** bringen. Die **Auswahl** des richtigen **Kontos** bei der Eingabe von Buchungen erfolgt dann durch Auswahl aus einer **Liste von prozessbezogenen Konten**. **Fehler in der Kontierung** werden so **wesentlich verringert** und die **Datenerfassung vereinfacht** und **beschleunigt**.

Prozessbezogen kann ein Gewinn oder Verlust berechnet werden und so die **Entstehung des Reingewinns** über die einzelnen **Prozesse** verfolgt werden. Dabei kann es wertvoll sein zwischen zwei **Prozessklassen** zu unterscheiden und Buchungen, die den **Bargeldfluss** betreffen (zum Beispiel Zinsen und Spesen) von **kalkulatorischen Buchungen** (wie zum Beispiel Abschreibungen) zu trennen. **Pro Prozess** muss die **Summe** aller Buchungen **null** ergeben. Entsprechend ist bei dieser Auswertung auch der **Gewinn** aus der **Bilanz** gleich dem **negativen Gewinn** aus der **Erfolgsrechnung**.

Mandat: MyCompany **Totale pro Buchungsprozess/Profite** 01.07.09
Period: 2009.1 Basis Datei period (1/22) 01.01.09 - 31.12.09

Prozess	Titel	Total	Gewinn Bilanz	Gewinn ER
interest	bank interests and...	0.00	150.00	150.00
payable	accounts payable	0.00	-5,050.00	-5,050.00
receivable	accounts receivable	0.00	0.00	0.00
sales	sales orders	0.00	13,000.00	13,000.00
Gesamt-Total		0.00	8,100.00	8,100.00

Entstehung des **Gesamtgewinns** als Summe
der **Einzelgewinne** pro **Buchungsprozess**
(für Periode 2009.1)

Gewisse **Prozesse** wie beispielsweise die Lohnabrechnung werden aus der Finanzbuchhaltung **ausgelagert**. So kann das Datenvolumen reduziert, oft aber auch die **Funktionalität der Anwendung** ohne Rücksicht auf die Anforderungen der Finanzbuchhaltung **optimiert** werden. Auch in diesen Fällen bietet aber die prozessorientierte Datenstruktur von **sqlFinance** eine optimale **Schnittstelle** für die Übernahme der Daten, die in **Belegen** mit einer beliebigen Anzahl **einseitiger Buchungen** erfolgen kann. Voraussetzung ist auch hier, dass die Summe der übergebenen Buchungen null ergibt.

Vor allem bei der **Revision** zeigt sich die **Effizienz der prozessorientierten Finanzbuchhaltung**. Der Revisor muss nicht mehr den Beleg aus dem Journal rekonstruieren, sondern kann im Gegenteil ganze Beleggruppen **prozessweise** überprüfen. Aus der Sicht des Revisors **besonders prüfenswerte Prozesse** wie Abschreibungen und andere Wertberichtigungen lassen sich leicht identifizieren. In den Notizen des **Belegs** ist allenfalls auch die **Argumentation** des Buchführenden erfasst. Diese kann dann in Bezug auf ihren **Inhalt** sowie die **Konsistenz der Verbuchung** wirkungsvoll geprüft werden.

In einer **Variante** kann der Prozess auch dem **Vertrag** zugeordnet werden. Im Gegensatz zum **Buchungsprozess** („booking process") kann man hier von einem **Geschäftsprozess** („business process") sprechen. Dabei werden die **Verträge** einem **Prozess** zugeordnet, typischerweise Kunden, Lieferanten, Banken, Vermieter, Mitarbeiter et cetera. Auch in diesem Fall ergibt sich eine aussagekräftige **Auswertung**, die den **Beitrag** der einzelnen **Vertragspartner** zum **Gesamtergebnis** darstellt (Mitarbeiter – ohne erfolgswirksame Buchungen- und Lieferanten bezahlt 10'487.40; Bankzins und Kundenzahlung incl. MWST erhalten 14'138 ergibt ein Total für den aktuellen Cash-Bestand von 3'650.60):

Mandat: MyCompany **Totale pro Geschäftsprozess/Gewinne** 01.07.09
Period: 2009.1 Basis Datei period (1/22) 01.01.09 - 31.12.09

Prozess	Titel	Total	Gewinn Bilanz	Gewinn ER
customer	customers (sales)	-13,988.00	-988.00	13,000.00
employee	company employees	5,437.40	5,437.40	0.00
finance	financial services providers	3,500.60	3,650.60	150.00
supplier	suppliers (purchase)	5,050.00	0.00	-5,050.00
Gesamt-Total		0.00	8,100.00	8,100.00

Entstehung des **Gesamtgewinns** als Summe der **Einzelgewinne** pro **Geschäftsprozess**

Bei dieser Auswertung zeigt sich auch, dass die **Summe der Gewinne** in der **Bilanz** minus Summe der Gewinne in der **Erfolgsrechnung** über alle Buchungen eines oder mehrerer Belege immer **null** ergibt (vergleiche dazu Kapitel über Luca Pacioli). Ungleichheiten im Gewinn zwischen Bilanz und Erfolgsrechnung ergeben sich, sobald eine **Zahlung** erfolgt. Zu diesem Zeitpunkt wird die **Bilanzposition** des Vertragspartners ausgeglichen, der **erfolgswirksame Aufwand oder Ertrag** desselben bleibt aber weiterhin stehen.

Der **Geschäftsprozess** wird immer über die ganze **Periode** und **alle** darin enthaltenen **Belege** ermittelt. Beim **Buchungsprozess** hingegen wird der Gewinn nur für die dem **Prozess zugeordneten Belege** berechnet. Pro Buchungsprozess müssen sowohl **Bilanz** wie auch **Erfolgsrechnung** deshalb immer den **gleichen Gewinnbetrag** ergeben, da der **Buchungsprozess** eine Anzahl **Belege** mit **Buchungen** umfasst, deren Total **null** ergibt.

II. Schneller Einstieg

1. Installation

Instruktionen für die Installation

Auf der **Homepage** (http://www.sqlfinance.com) kann unsere Software sqlFinance direkt herunter geladen werden. Im Downloadbereich (**Download Center**) befindet sich nicht nur unsere Software, sondern auch Instruktionen und ein Handbuch zu sqlFinance.

Sie haben die **Wahl** zwischen der **sqlFinance "Public Version"** (Gratisversion) oder der **sqlFinance "Full Version"** (Vollversion). Sobald Sie eine Version gewählt haben, können Sie ein **Bestellformular ausfüllen**. Sie geben dort Ihre **persönlichen Daten** ein und erhalten dann **per Email** einen **Link** für das **Herunterladen**. Gemäss den **Instruktionen** können Sie nun **sqlFinance installieren**. Das Vorgabe-Verzeichnis (c:\programs\sqlFinance") wird anhand Ihrer Systemvorgaben bestimmt und kann auch anders gewählt werden.

Anmerkungen zur Installation der Vollversion von sqlFinance

Zwischen der **sqlFinance "Public Version"** und der **sqlFinance "Full Version"** gibt es wichtige Unterschiede. Es folgt nun eine kurze Liste über die Funktionen der beiden Versionen von **sqlFinance**:

Die **Gratisversion** („Public Version") ist limitiert auf den Einsatz im **Einzelplatz** („single user version") und bietet:

- Finanzbuchhaltung mit unbeschränkter Anzahl von Mandaten und Teilperioden

Die **Vollversion** („Full Version") von sqlFinance ist **mehrplatzfähig**, unterstützt den Zugriff auf **PC-Datenbanken** sowie **sqlServer** und enthält ergänzend folgende **Module**:

- Einkauf und Verkauf
- Lagerführung mit Lieferung
- Debitoren und Kreditoren
- Human Ressources mit Lohnabrechnung
- Zeitabrechnung
- Vermögensverwaltung

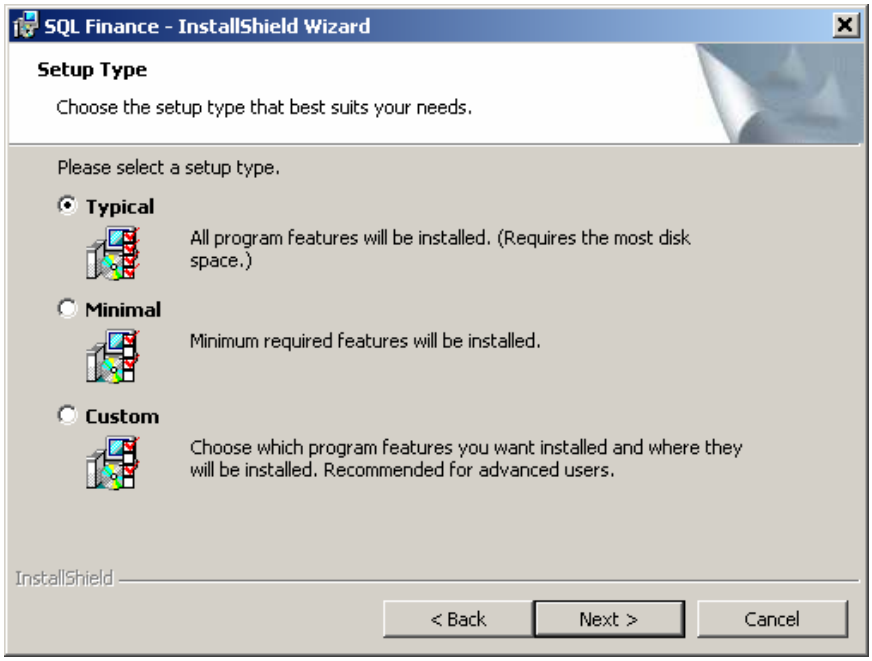
Anmerkungen zur Benutzung von Fremdsprachen

Wenn Sie sqlFinance in Verbindung mit Fremdsprachen einsetzen möchten, können Sie den "**BT Language Manager**" verwenden. Eine aktuelle Version des "BT Language Manager" befindet sich auf der sqlFinance **Homepage** im **Download Center**. Der **Language Manager** sollte im Verzeichnis **' c:\'** installiert werden. Er enthält Werkzeuge für die Beschleunigung von Übersetzung und zur Sicherung eines konsistenten Sprachgebrauchs. **sqlFinance** stellt eine **Verbindung** zum **Language Manager** her anhand des Werts der **„lmroot"-Variable** in der **„l.ini"-Datei** (zu finden im Stammverzeichnis von sqlFinance).

Ist der **Language Manager** korrekt installiert, wird **sqlFinance** automatisch in der gewünschten **Fremdsprache** zu arbeiten beginnen (je nach Wert der lmroot-Variable und der Einstellung der Fremdsprache im **Language Manager**). Bitte beachten Sie, dass Sie **sqlFinance zweimal starten** müssen, um eine korrekte Umstellung in die gewünschte Fremdsprache zu erhalten (einmal um die Einstellung zu ändern und ein zweites Mal, um von Anfang an eine Übersetzung zu erhalten).

Zusätzliche Informationen

Auf unserer Homepage **http://www.sqlfinance.com** gibt es immer die neuesten Informationen über unser Unternehmen und unsere Software sqlFinance.



Das sqlFinance Installationswerkzeug

2. Einrichten der Anwendung und erste Buchung

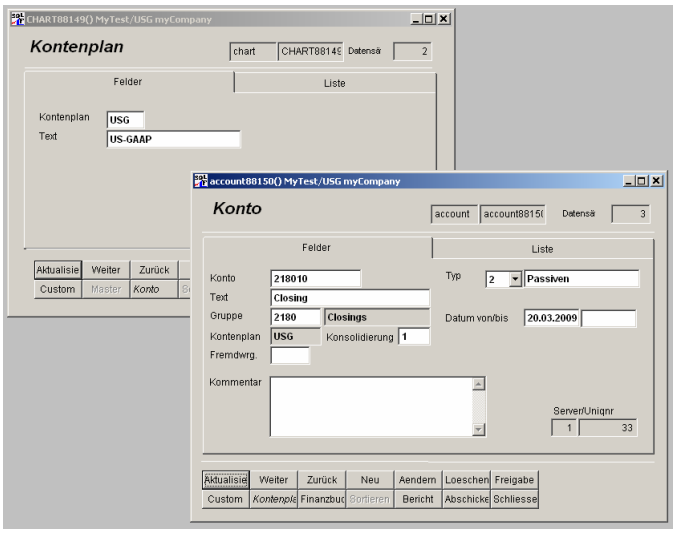
Nach der Installation und dem Start der Software sqlFinance stellt das Programm sofort eine Verbindung zum Anwendungsbeispiel **„myCompany"** her, wählt als **aktives Mandat** „myCompany" und präsentiert das **Selektionswerkzeug**. Der Anwender hat nun die Wahl, **das Beispiel nach seinen Anforderungen umzubauen** oder ein **neues Mandat** mit allenfalls auch einem **neuen Kontenplan**, gegebenenfalls in einer **neuen Datenbank** einzurichten.

Soll nur das bestehende Mandat **„myCompany"** geändert werden, kann man einfach die **vorhandenen Belege löschen** und dann sofort mit der **Eingabe neuer Belege** beginnen. Beim Löschen von Belegen werden (nach Rückfrage an den Anwender und dessen Bestätigung) die mit dem Beleg verbundenen Buchungen ebenfalls gelöscht.

Möchte man dagegen ein **neues Mandat** mit einem **neuen Kontenplan** eröffnen und eine **erste Buchung** eingeben, ist wie folgt vorzugehen:

a. **Zuerst** den **neuen Kontenplan** erstellen. Im Beispielmandat „myCompany", vom **Selektionsfenster** aus den **Kontenplan** selektionieren. Mit Taste **„hinzufügen"** einen **neuen Kontenplan** (zum Beispiel **„USG"** für „US-GAAP") eingeben und mit der Taste **„ändern"** bestätigen.

b. Als **Detail** zum Kontenplan ein Formular für die **Konten** selektionieren (Klick auf die dritte Taste von links in der zweiten Tastaturzeile) und eine Anzahl **Konten eingeben**. Die Konten müssen **Gruppen** zugeordnet werden. Allenfalls können vom Selektionswerkzeug aus alle Gruppen selektioniert und vorgängig weitere Gruppen erfasst werden. Für jedes Konto ist auch der **Grundtyp** (Aktiven, Passiven, Aufwand und Ertrag) anzugeben.

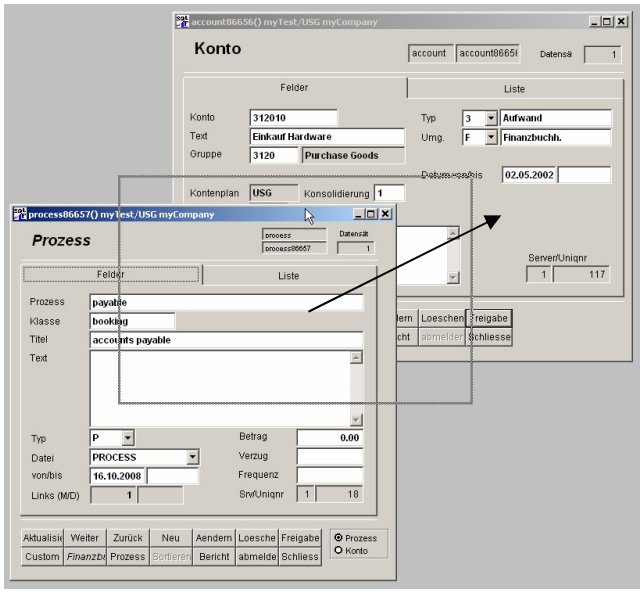


Der neue Kontenplan „USG" und das Abschlusskonto „218010"

c. Der neue Kontenplan sollte die folgenden **Spezialkonten** (vorzugsweise in den Passiven Typ „2") enthalten:

- Abschlusskonto
- Vorgabekonto Beleg

d. Die **Konten** mit **Prozessen** in Verbindung bringen. Mit dem Selektionswerkzeug je ein **Formular** mit allen **Prozessen** und ein weiteres mit allen Konten auswählen. Durch Ziehen mit der Maus (Seite **„Felder"**) vom Prozess-Formular zum Konten-Formular die Prozesse den Konten **zuordnen**. Mit Hilfe der **Optiongroup** (rechts unten beim Prozessformular, Doppelclick auf die gewünschte Option) können die zugeordneten Konten **selektioniert** – und mit der Taste **„abmelden"** auch wieder **getrennt** werden.



Zuordnen Prozess „payable" zu
Konto 312010 (Einkauf Hardware)

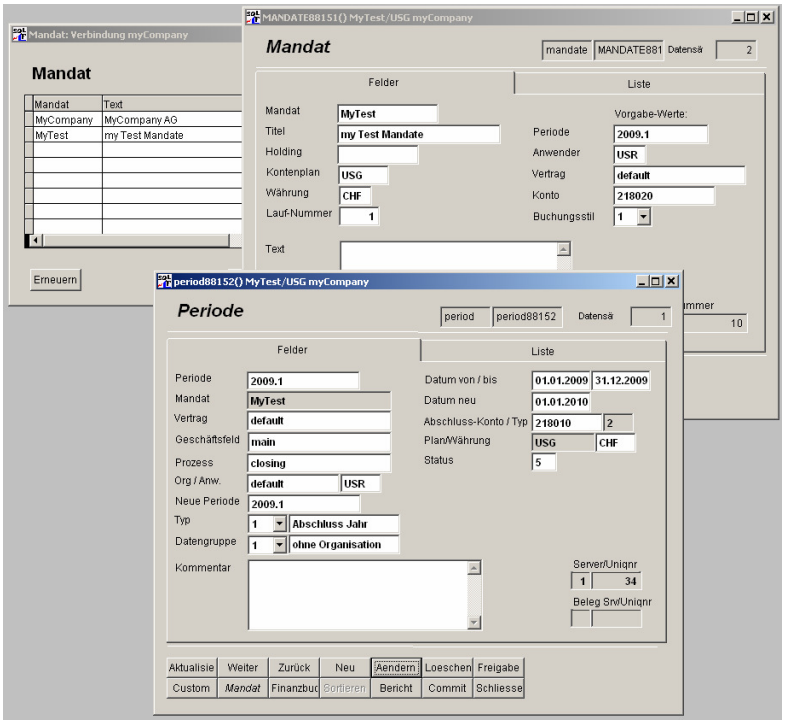
e. Vom **Selektionswerkzeug** aus das alte Mandat **(„myCompany")** selektionieren. Ein **neues Mandat** eingeben (zum Beispiel **„myTest"**). Im Mandat den neuen Kontenplan **„USG"** angeben, ebenso das **Abschlusskonto**, aber **noch keinen Wert für die Periode**.

f. Nach der Eingabe des neuen Mandats im **Dialog mit der Software** das neue Mandat als **aktives Mandat** akzeptieren und vom (automatisch präsentierten Mandatsfenster) mit der Taste **„weiter"**/"continue" wieder zum **Selektionsfenster** gehen. Das neue **Mandatsfenster** löschen und **nochmals selektionieren**, allenfalls mit den Tasten **„weiter"** und **„zurück"** vom alten Mandat zum neuen Mandat blättern.

g. Mit einem Doppelklick auf dem Feld **„Periode"** des Mandatsfensters (mit Wildcard **„%"**) das **Periodenformular** aufrufen und eine **neue Periode** (zum Beispiel **„2009.1"**) eingeben. Das Feld für die **Abschlussperiode** („neue Periode") vorerst noch **leer lassen**. Zuerst eine **Anzahl Perioden** eingeben, nachher die **Abschlussperiode anpassen**.

h. Durch Blättern auf die gewünschte Periode gehen, und mit der Taste **„schliessen"**/"quit" das Periodenformular verlassen. Der **Wert der**

gewählten Periode wird dabei auf das Mandatsformular **übertragen**. Auch diese Änderung mit der Taste **„ändern"**/"update" bestätigen. Im Mandat sind damit die **Vorgaben für die aktuelle Periode** und **das Vorgabekonto des Belegs** erfasst.



Formulare „Mandat" und „Periode" mit Mandatswerkzeug (oben) für die Aktivierung des neuen Mandats

i. Ein **Belegformular** selektionieren. Im leeren Formular einen **ersten Beleg** eingeben. Das Formular verlangt einen Wert für die Felder **„Prozess"** und **„Aktivität"**. Für die Auswahl eines Prozesses im Feld **„Prozess"** eine Wildcard (Prozentzeichen **„%"**) einsetzen und **doppelklicken**. Jetzt im Prozessformular einen **Prozess auswählen** oder einen geeigneten Eintrag erstellen. Zurück mit der Taste **„Schliessen"** oder der Tastaturkombination **„ctrl-F9"**: die Werte für **„Prozess"** und **„Aktivität"** werden übertragen.

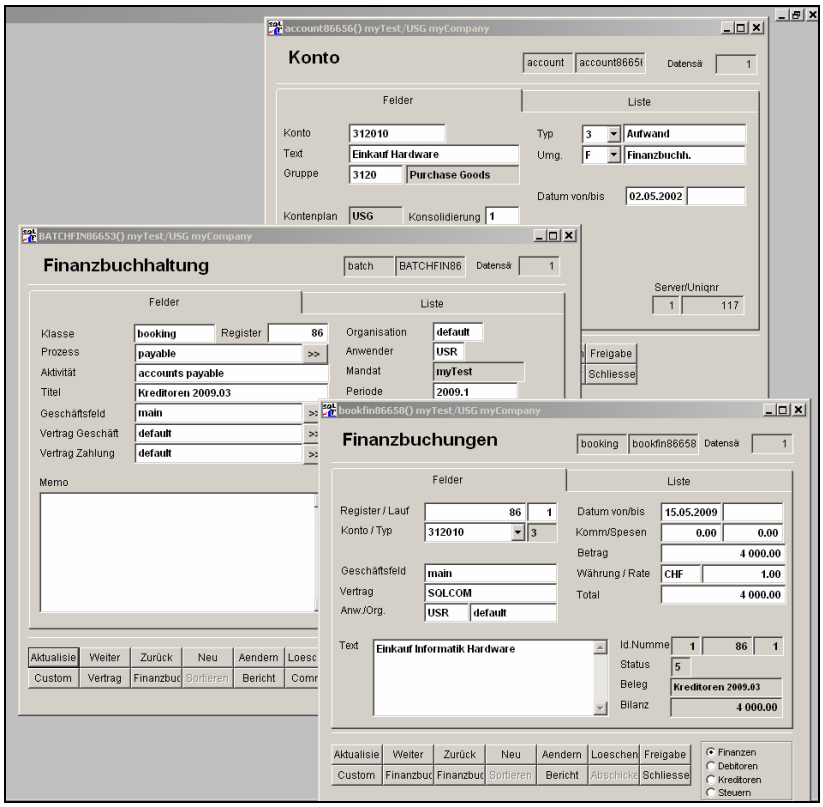
j. Mit der Taste **„Detail"** beziehungsweise **„Finanzbuchungen"** (Taste drei von links in der zweiten Tastenkolonne) die zum Beleg gehörenden **Buchungen** selektionieren. Im leeren Formular eine **erste Buchung** mit dem richtigen Wert für das **Konto** und den **Betrag** erstellen.

k. Falls **prozessbezogen** Konten hinterlegt wurden, können diese mit der **Combobox** des **Kontenfelds** angezeigt werden (mit Klick auf die Pfeiltaste des Kontenfelds). Es können aber auch **andere Konten** angegeben werden, soweit sie im Kontenplan enthalten sind. Eingabe „%" im **Kontenfeld** und **Doppelklick** zeigen die **vollständige Liste** der **verfügbaren Konten**.

l. Bei der Eingabe von **„SQLCOM"** im Feld **„Vertrag"** beachten, wie mit der **„Whichpen"**-Funktion nach wenigen Buchstaben die **Vorgabewerte für Geschäftsfeld, Konto, Organisation und Betrag** vom **Vertrag** in die Buchung **übernommen werden** (gegebenenfalls **Vorgabewerte** im

Vertrag noch auf den neuen Kontenplan **anpassen** oder neue Verträge eingeben).

m. Nach den ersten Buchungen kann die Erfassung der Daten in der **Bilanz** überprüft werden. Die **aktuelle Periode** aufrufen, mit der Taste **„Berichte"**/"reports" zum Berichtswerkzeug gehen und dort einen Bericht auswählen (zum Beispiel **„Details pro Konto"** oder **„Totale Bilanz"**). Den Bericht mit **Doppelklick** auf die Auswahl **„Vorschau"** starten.



Formulare für Finanzbeleg mit Finanzbuchungen, sowie das Konto „Einkauf Hardware"

n. Vor dem Abschluss der **Periode** die **Belege abschliessen** (Taste **„abschicken"/"commit"**). Dabei wird die **Summe null** über alle Buchungen **überprüft**.

o. Während der ersten Schritte den Prozess **„Test"** und den Vertrag **„Default** verwenden". Im Vertrag **„Default"** die **Soll- und Haben-Konten** gemäss **neuem Kontenplan** ändern.

3. Ein- und zweiseitige Buchungen

Durch die konsequente Umsetzung des **Datenmodells** ergibt sich bei sqlFinance die Möglichkeit, die **Verwendung der ein- und doppelseitige Buchungserfassung innerhalb derselben Buchhaltung** zu erlauben. Nachfolgend sind die zwei Erfassungsmethoden kurz vorgestellt:

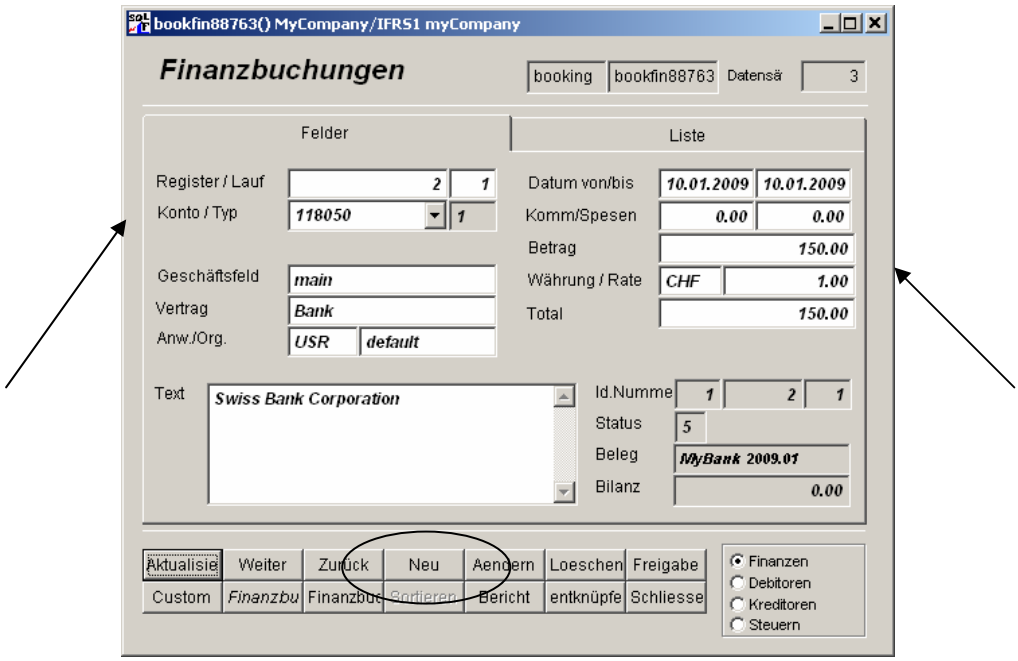
Methode 1: Einseitiger Buchungsstil
Belege mit mehreren Buchungen, wie:

> Konto, Datum, Text, Betrag

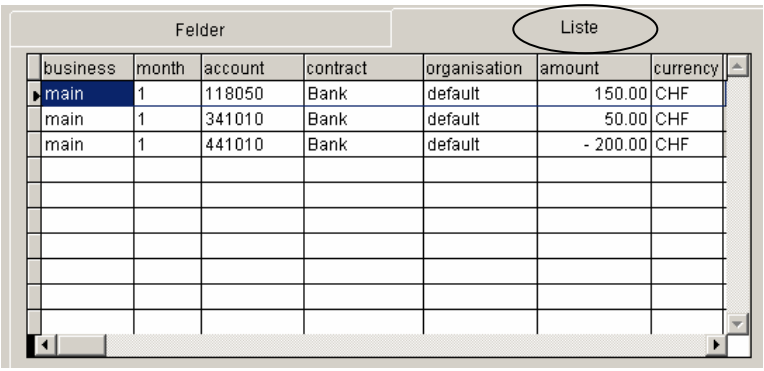
wobei gilt**:** **Summe** (Betrag1, Betrag2, Betrag3 …) **gleich null**.

Im **Bericht** werden in diesem Fall die Buchungszeilen **unverändert** dargestellt:

> Konto1, Datum, Text, Betrag1
> Konto2, Datum, Text, Betrag2
> Konto3, Datum, Text, Betrag3
et cetera



Struktur der Buchungszeile in der Finanzbuchhaltung von sqlFinance (Methode 1)



Beispiel mit drei einseitigen Buchungen, die zu einem Beleg mit einem Total von Null zusammengefasst werden

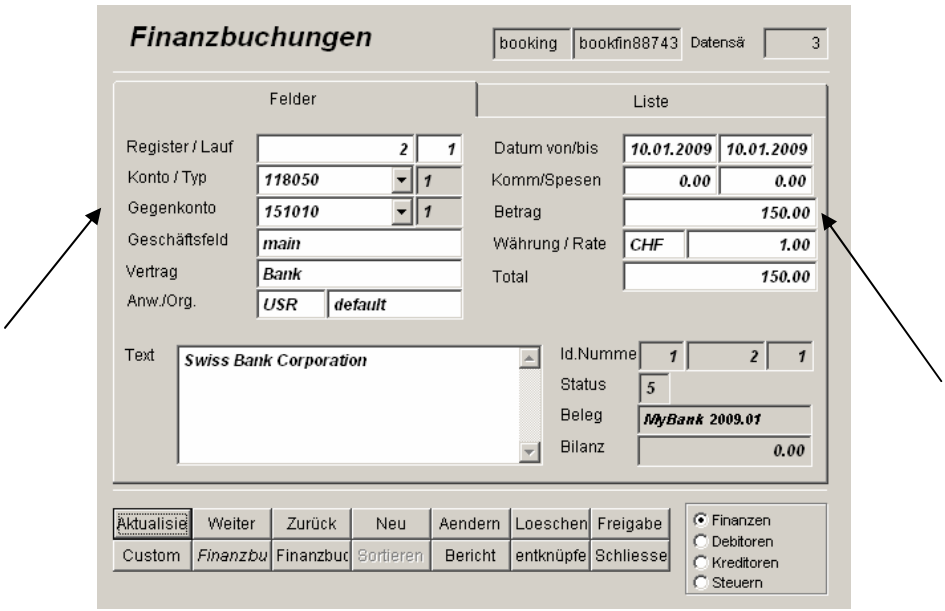
Methode 2: Zweiseitiger Buchungsstil
Einzelne Soll-/Haben-Buchungen (Buchungszeilen) wie:

> Sollkonto, Habenkonto, Datum, Text, Betrag

Im **Bericht** wird die Buchungszeile **aufgelöst** in **zwei Buchungen**, die immer ein **Total von null** ergeben, wie zum Beispiel:

> Sollkonto, Datum, Text, + Betrag
> Habenkonto, Datum, Text, - Betrag

Bei **Methode (2)** ist die **Null-Summen-Regel** auf **jeder Buchungszeile** (mit Sollkonto, Habenkonto, Betrag) erfüllt. Diese **Buchungszeile** wird im **Bericht** aufgelöst in **zwei Buchungen** mit **demselben** positiven beziehungsweise negativen **Betrag** (Sollkonto, + Betrag und Habenkonto, - Betrag).



Struktur der Buchungszeile in der Finanzbuchhaltung von sqlFinance (Methode 2)

Felder | Liste

business	month	account	c_account	contract	organisation	amount
main	1	118050	151010	Bank	default	150.00
main	1	341010	151010	Bank	default	50.00
main	1	441010	151010	Bank	default	- 200.00

Beispiel mit drei zweiseitigen Buchungen, die im Bericht alle auf je zwei Buchungen aufgelöst werden

Dabei ist zu beachten, dass **nur** Berichte mit **allen Buchungen pro Beleg** (typischerweise Bilanzen und Erfolgsrechnungen) eine **Summe von Null** anzeigen werden. Berichte zu ausgewählten Konten werden demgegenüber die **Summe der Bewegungen** auf **diesen Konten** anzeigen.

Im Bestreben die Buchhaltung möglichst verständlich zu halten, **ändert** die Software in der Regel die **Darstellung** von (2) **Passiven** und (4) **Ertrag**. Durch **Umkehr des Vorzeichens** wie in Methode 1 werden die meist **negativen** Zahlen **positiv** dargestellt. Diese Funktion wird vor allem bei den **Berichten** verwendet. Wenn **Listen** nach über **alle vier Typen** gruppiert sind (1 Aktiven, 2 Passiven, 3 Aufwand, 4 Ertrag) **ändert** das Programm die Darstellung von (2) Passiven und (4) Ertrag von **negativen** zu **positiven** Werten.

Die Aufführung des Gewinns in der **Bilanz** (+Aktiven-Passiven, in der Regel ein **positiver** Wert) kann gegebenenfalls auch weggelassen werden, und der **Gewinn** aus der **Erfolgsrechnung** (-Ertrag +Aufwand, in der Regel ein **negativer Wert**) wird mit **umgekehrtem Vorzeichen** als **positive Zahl** dargestellt. **Einfache Buchungslisten** hingegen werden die Buchungen weiterhin mit ihrem **mathematisch korrekten Wert** anzeigen: Aktiven und Aufwand mit meist positiven Werten, Passiven und Ertrag mit meist negativen Werten.

Einnahmen sind wie folgt zu verbuchen:

(1) + Aktiven
(4) - Ertrag
= 0

Für die Aktivseite kann zum Beispiel ein Bank-, Cash- oder Debitorenkonto verwendet werden.

Die Verbuchung von Ausgaben erfolgt dementsprechend:

(1) - Aktiven
(3) + Aufwand
= 0

Statt den Aufwand bei den Aktiven (zum Beispiel einem Bankkonto) zu verbuchen, kann man hier auch ein Passivkonto verwenden (zum Beispiel Kreditoren). Als Aufwand oder Ertrag werden dabei vielfach **mehrere Buchungszeilen** verwendet werden, also zum Beispiel eine **Anzahl Rechnungen**, die über eine **Sammelzahlung** bezahlt werden, oder

verschiedene **Fakturapositionen**, die einem **Debitoren-Sammelbetrag** gegenüberstehen.

Einige **Formulare** (wie Einkauf, Verkauf, Debitoren oder Kreditoren) verwenden Darstellungen mit **umgekehrtem** Vorzeichen. Während Einkaufs- und Debitoren-buchungen ohnehin meist positive Zahlen enthalten, werden Buchungen im **Verkauf** und bei den **Kreditoren** meist als **negative** Zahlen gespeichert, im **Formular** aber als **positive** Werte dargestellt. Um ein **Null-Total** zu erhalten, kommen bei solchen Belegen meist **verschiedene Buchungsformulare kombiniert** zum Einsatz. Bei einem **Verkaufsbeleg** sind dies **Verkaufsbuchungen** (umgekehrt) und **Debitoren** (ohne Umkehr), bei einem **Einkaufsbeleg** sind dies **Einkaufsbuchungen** (ohne Umkehr) und **Kreditoren** (mit Umkehr). Zum Einkaufs- und Verkaufsbeleg gehört immer auch ein **Buchungsformular für die Steuern** (zum Beispiel Mehrwertsteuer). Der Steuerwert ist in der Regel eine Verpflichtung, das Steuerformular verwendet deshalb immer umgekehrte Werte.

Buchungsformulare im Einsatz bei Verkaufsbelegen:

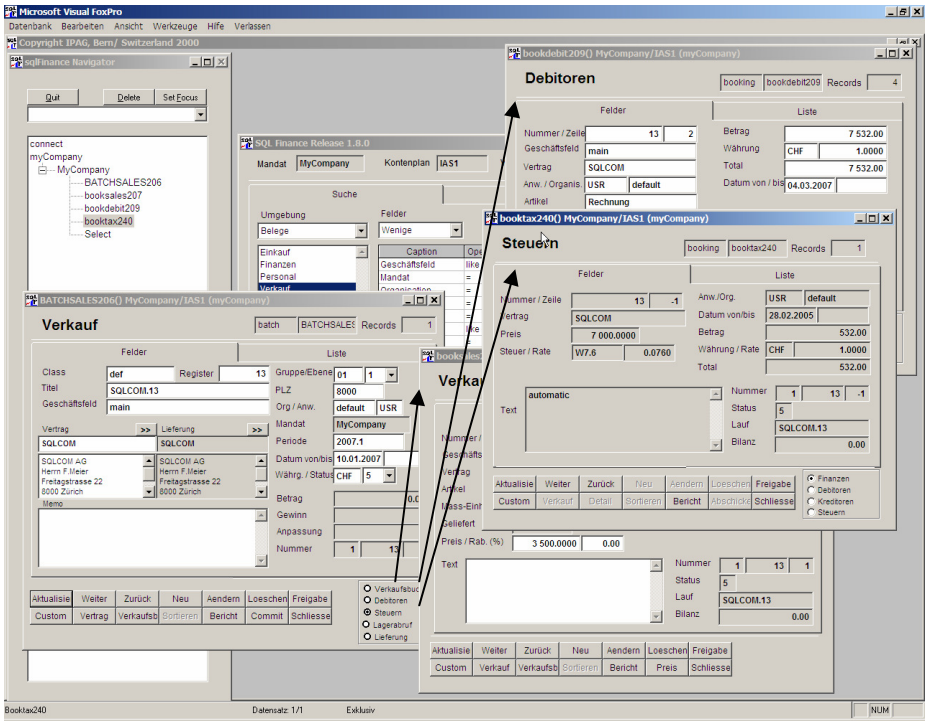
(1)	+ Aktiven:	Debitoren-Formular	nicht umgekehrt
(2)	- Passiven:	Steuer-Formular	umgekehrt
(4)	- Ertrag:	Verkaufs-Formular	umgekehrt

Buchungsformulare im Einsatz bei Einkaufsbelegen:

(2)	- Passiven:	Kreditoren-Formular	umgekehrt
(2)	- Passiven:	Steuer-Formular	umgekehrt
(4)	+ Aufwand:	Einkaufs-Formular	nicht umgekehrt

Für **jede Linie** in diesen Beispielen könnten auch mehrere Zeilen eingegeben werden. Ein einzelner **Beleg** kann **mehrere Aufwand- oder Ertragsbuchungen** enthalten und so ganze Einkaufs- oder **Verkaufsgeschäfte** darstellen. Auch Debitoren oder **Kreditoren** können in einem einzigen **Beleg** mehrere **Buchungen** umfassen, zum Beispiel mehrere Teilzahlungen oder Korrekturen in Ergänzung zu einer Zahlung.

Das System wird Steuerverpflichtungen auf Belegstufe automatisch nachführen. **Mehr als eine Steuerart pro Beleg ist möglich**, was zu mehreren Buchungszeilen im Steuerformular führt. In Ländern mit **Mehrwertsteuer** können die im Einkauf und Verkauf berechneten positiven und negativen Steuerwerte in einem **Clearing-Konto** verbucht werden. Der Kontoauszug dieses Kontos lässt sich dann für die **Steuererklärung** verwenden.

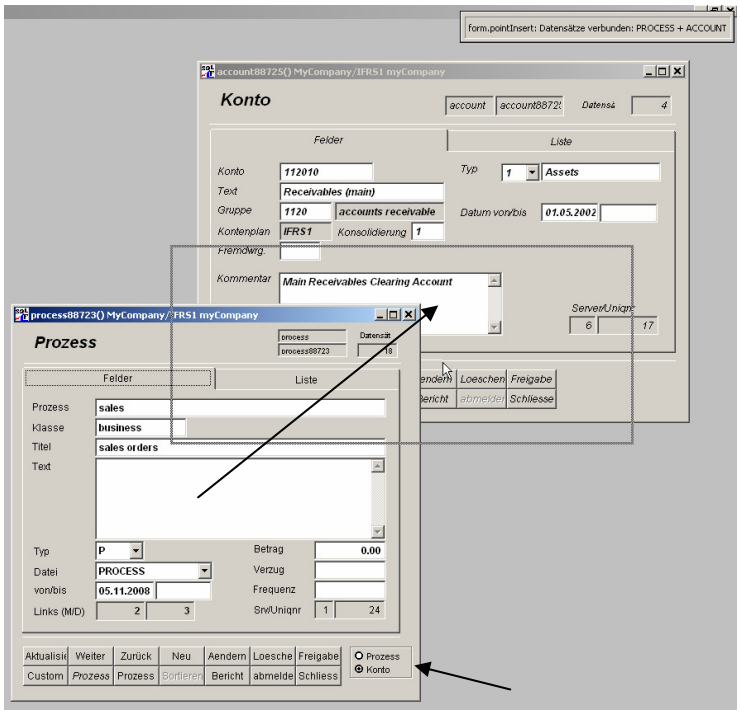


Beispiel eines Verkaufsbelegs (links) mit Buchungsformularen für Debitoren-, Steuer- und Verkaufsbuchungen (Vollversion)

Die verschiedenen **Buchungsformulare** (zum Beispiel Debitoren, Verkauf und Steuern) können vom **Beleg** aus aufgerufen werden, indem man die entsprechende **Auswahl** neben den Funktionstasten mittels **Doppelklick** aktiviert. Man kann aber auch nur die richtige **Option** mit einem einfachen Klick wählen und dann über die Funktionstaste **„Detail"** beziehungsweise **„Finanzbuchungen"** das gewünschte **Buchungsformular** in der Kaskade aktivieren.

4. Kontierung nach Prozessen

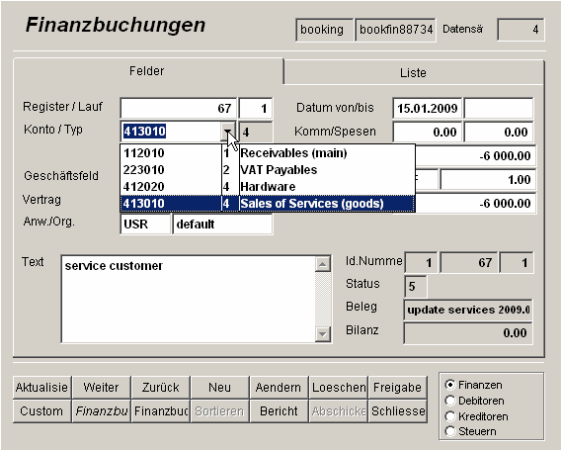
sqlFinance ermöglicht die **Zuordnung** von **Konten** zu **Prozessen**. Die Zuordnung erfolgt durch **Ziehen mit der Maus** von einem **Prozessfenster** zu einem **Kontofenster**. Betätigen Sie im Seitenrahmen **„Felder"** des Prozessfensters die **rechte Maustaste** und ziehen Sie die ganze Seite (mit gedrückter Maustaste) über das gewünschte **Kontofenster**. Zur **Bestätigung** erscheint die **Meldung**: „Datensätze verbunden: PROZESS + KONTO".



Zuordnung der Konten zu Prozessen durch Ziehen mit der Maus (von links nach rechts)

Die **neu verbundenen Kontoeinträge** können vom **Prozessfenster** aus **selektioniert** werden. Dazu muss die Option **„Konto"** (im Prozessfenster rechts unten) aktiviert werden. Durch **Doppelklick** auf die gewählte Option oder auf die Taste **„Detail"** (dritte Taste von Links in der zweiten Tastenzeile) wird das **Kontenfenster** mit den entsprechenden Konten angezeigt. Ist das Kontenfenster auf diese Art und Weise selektioniert worden, kann die **Verbindung** (Prozess-Konto) durch Anklicken der Taste **„abmelden"** (sechste Taste von links in der zweiten Tastenzeile) allenfalls auch wieder **getrennt** werden.

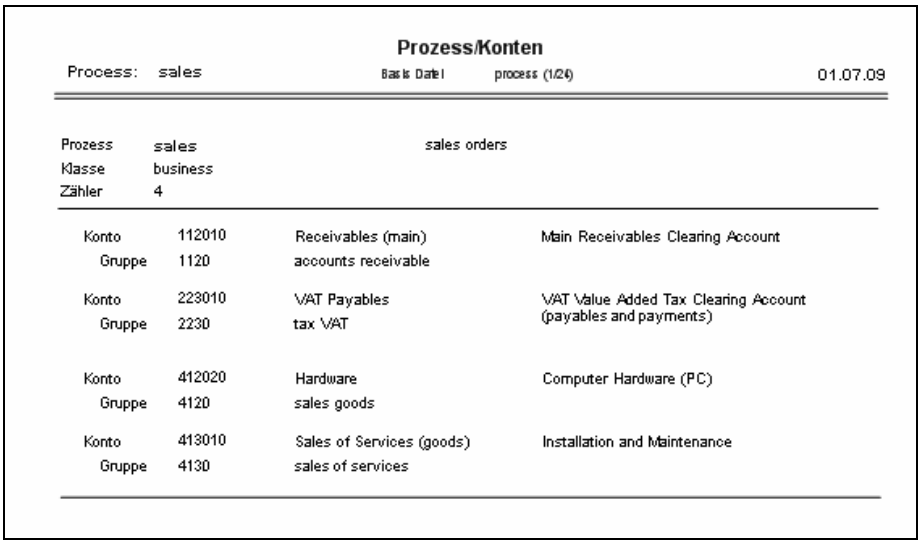
Durch Ziehen mit der Maus von einem Prozessfenster zu einem anderen können auch **Master-Prozesse** mit **Detail-Prozessen** in Verbindung gebracht werden. Die Prozesse lassen sich so in eine beliebige **Hierarchie** bringen. Verschiedene **Druckprogramme** stehen zur Verfügung zur Darstellung dieser hierarchischen Strukturen. Vom **Master-Prozess** ausgehend lassen sich **Detail-Prozesse** und **zugeordnete Buchungen** übersichtlich auflisten. Für den Master-Prozess **„booking"** kann man so einen Bericht zu den Teilprozesse **„credit", debit", „interest"** und **„sales"** sowie der entsprechenden **Kontierung** erstellen.



Anzeige der prozessbezogenen Kontenauswahl im Kontofeld

Bei der **Erfassung von Buchungen** liefert das Kontofeld eine **Auswahl von Konten**, die **dem Prozess entsprechen**. Der für die Buchungen relevante Prozess wird vom **übergeordneten Beleg** vorgegeben. Die Kontenfelder sind als **ComboBox** ausgeführt. Durch Klick auf den Pfeil am rechten Feldende öffnet sich die **Liste** der dem Prozess entsprechenden **Konten**.

Zur Liste **zugefügt** sind auch allfällige **andere Konten**, die der Anwender im aktuellen Beleg **verwendet** hat oder über die Doppelklick-Funktion **abgefragt** hat. Die ComboBox liefert nur **Vorschläge** für die Kontierung, die Eingabe **anderer Konten** ist weiterhin **möglich**.



Prozess/Konten

Process: sales Basis Datei process (1/24) 01.07.09

Prozess	sales	sales orders	
Klasse	business		
Zähler	4		
Konto	112010	Receivables (main)	Main Receivables Clearing Account
Gruppe	1120	accounts receivable	
Konto	223010	VAT Payables	VAT Value Added Tax Clearing Account (payables and payments)
Gruppe	2230	tax VAT	
Konto	412020	Hardware	Computer Hardware (PC)
Gruppe	4120	sales goods	
Konto	413010	Sales of Services (goods)	Installation and Maintenance
Gruppe	4130	sales of services	

Prozess „sales" und dazugehörige Kontierung (Bericht aufgerufen vom Formular „Prozess" aus)

Um **Prozesse** und die dazugehörigen **Konten** aufzulisten, können die Berichte im Formular **"Prozess"** verwendet werden. Das Berichtsfenster kann im Formular **„Prozess"** mit der Taste **„Berichte"** (fünfte Taste von links in der zweiten Tastenzeile) aufgerufen werden. Um hingegen **Prozesse** mit **Buchungen** und deren **Totalen** erstellen zu können, muss die ergänzend **Periode** berücksichtigt werden. Diese Berichte sind demzufolge unter den Berichten des Formulars **„Periode"** aufzufinden.

Mandat: MyCompany
Period: 2009.1

Totale pro Buchungsprozess

Basis Date: period (1/22)
Process process: like '%sales%'

01.07.09
01.01.09 - 31.12.09

Prozess sales

1 Aktiven

Konto	Soll	Haben	Total
1120 accounts receivable			
112010 Receivables (main) (CHF)	13,988.00		13,988.00
	13,988.00		13,988.00
Total Typ 1 (Aktiven)			13,988.00

2 Passiven

Konto	Soll	Haben	Total
2230 tax VAT			
223010 VAT Payables (CHF)		-988.00	-988.00
		-988.00	-988.00
Total Typ 2 (Passiven)			-988.00

4 Ertrag

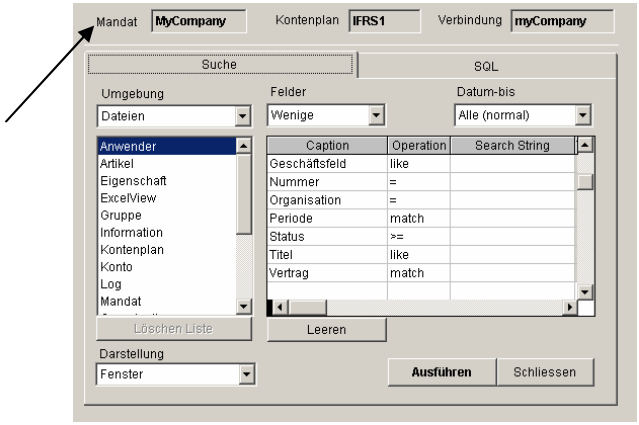
Konto	Soll	Haben	Total
4120 sales goods			
412020 Hardware (CHF)		-7,000.00	-7,000.00
		-7,000.00	-7,000.00
4130 sales of services			
413010 Sales of Services (goods)...		-6,000.00	-6,000.00
		-6,000.00	-6,000.00
Total Typ 4 (Ertrag)			-13,000.00

Totale Prozess (sales)	0.00
Gewinn Prozess (sales)	13,000.00
Gesamt-Total	0.00
Gewinn Total	13,000.00

Buchungsprozess "sales" mit Kontogruppen, Konten und dazugehörigen Buchungstotalen, pro Periode "2009.1" (Bericht aufgerufen vom Formular "Periode" aus)

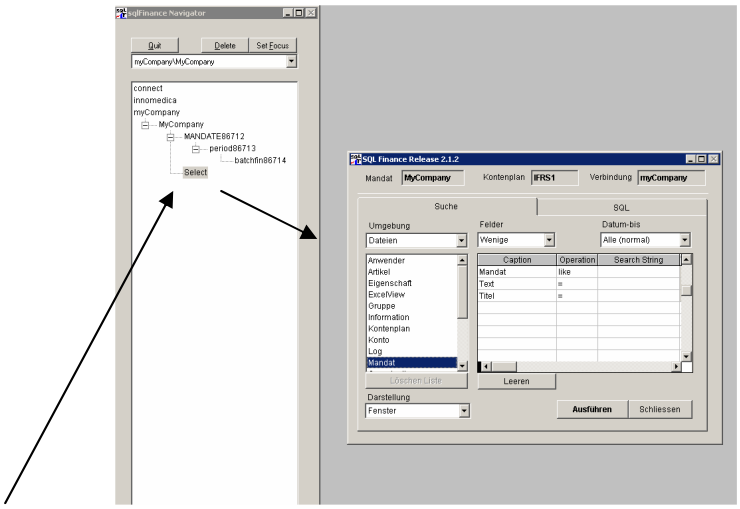
5. Anwendungsübersicht

Die Anwendung startet mit einem **Anmeldungsverfahren**, das zu einem **Selektionsfenster** führt. Von diesem Selektionsfenster aus kann man eine unbeschränkte Anzahl **Formulare** öffnen. Jedes Formular gibt Zugriff auf eine **Tabelle** aus der Datenbank und bietet eine Darstellung in **Listenform** oder mit den einzelnen **Feldern**. Über **Master-** und **Detail-Tasten** können – vom aktuellen Datensatz ausgehend – **weitere** Formulare geöffnet werden.

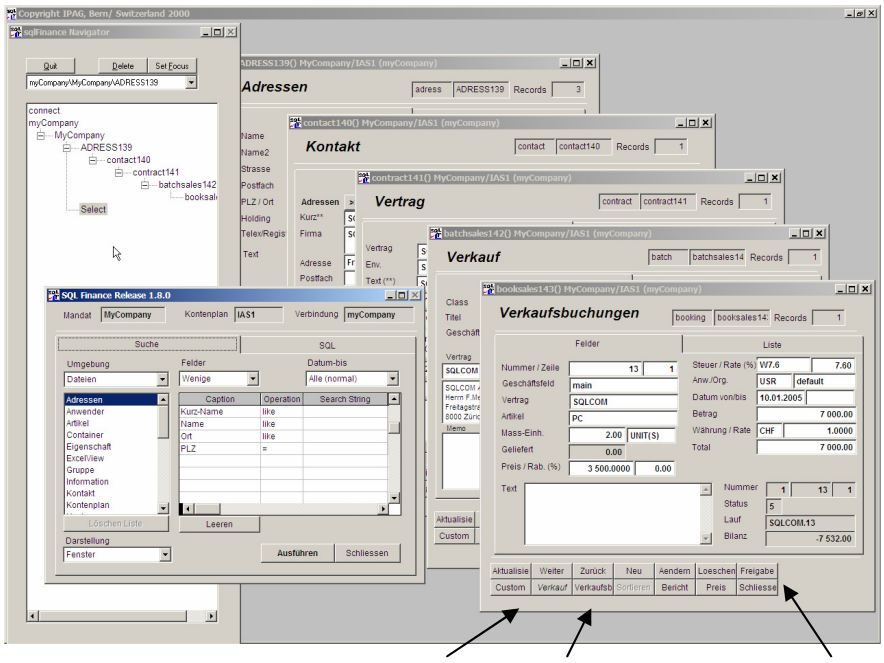


Selektionsfenster im Mandat "myCompany"

Mehrere derart verbundene **Formulare** bilden eine **Kaskade**. Durch einfache Klicks auf die **Master-** und **Detail-Tasten** kann der Anwender sich in der Kaskade nach oben und unten bewegen. In der Kaskade ist jeweils **nur** das zurzeit **aktive Formular sichtbar**. Die Anwendung bleibt damit auch bei mehreren offenen Kaskaden (jede wieder mit mehreren Formularen) **übersichtlich** in der Handhabung. Auf Wunsch kann man ein Formular auch aus der Kaskade herauslösen ("Freigabe"/"free", rechts oben) und danach über die Master- beziehungsweise Detailtasten weitere Formulare in einer neuen Kaskade öffnen.



Selektionsfenster mit Baumstruktur

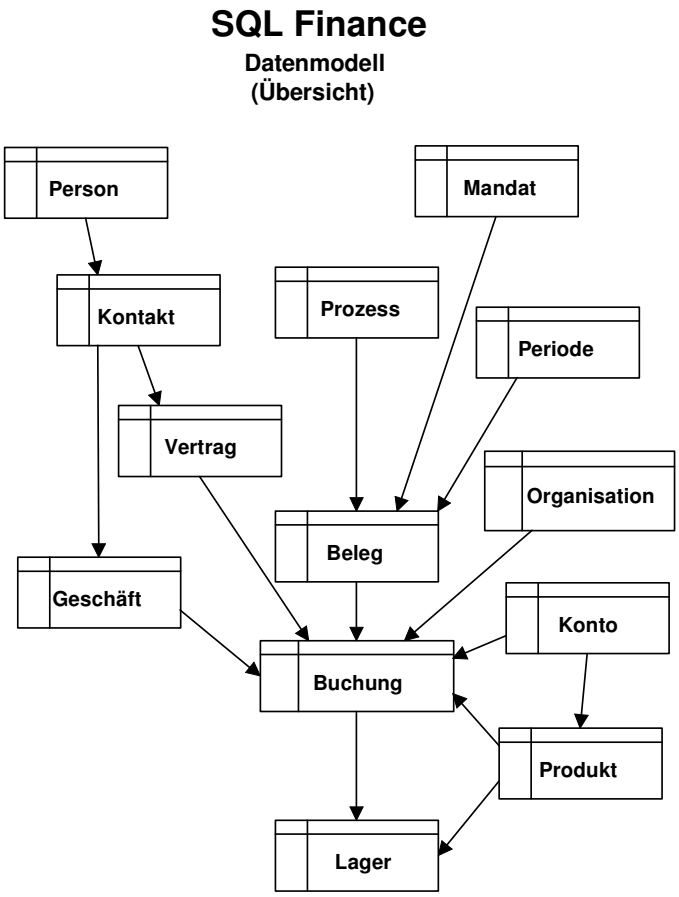


Kaskade von Formularen mit Master-, Detail- und Freigabe-Taste

Das Fenster mit der **Baumstruktur** in der **linken Hälfte** des Bildschirms (siehe Abbildung oben) unterstützt die **Handhabung** mehrerer **Formulargruppen** (Kaskaden). Jedes Formular einer Kaskade kann durch Drücken der **rechten Maustaste** in der Baumstruktur sichtbar gemacht werden. Formulare können auch über ein **Ablagewerkzeug** („Record Tool") für die Verwaltung einzelner Datensätze gezogen werden, um sich die Selektion in einem Eintrag zu merken. Zum Eintrag kann ein **Kommentar** beigefügt werden. Die gespeicherten Formulare können mit einem Klick auf die Taste **„Suchen Formular"** wieder aktiviert und weiter bearbeitet werden.

Durch **"Ziehen und Fallen lassen"** mit der Maus (**Drag and Drop**) können Datensätze von einem Formular zu einem anderen **kopiert** werden. Mit dieser Technik lassen sich zum Beispiel **Buchungen** oder ganze **Belege mit Buchungen** von einer Periode oder einem Mandat zu einer anderen Periode oder einem anderen Mandat kopieren. Eine **Repetierfunktion** ermöglicht das effiziente Bearbeiten ganzer **Belegs- und Buchungslisten**.

Die Formulare in **Kaskadenordnung** folgen im Prinzip der **hierarchischen Struktur des Datenmodells**. Um dem Anwender entgegenzukommen, wurde das Datenmodell möglichst einfach gehalten.



Dieses **Datenmodell** findet sich in der **sqlFinance** Software als **Kaskadenordnung** wieder oder kann über die **Master-Detail Funktionen** oder **Doppelklicks** in den Datenfeldern beim **Aufruf der Formulare** genutzt werden. So kann zum Beispiel die Datengruppe **Beleg – Buchung** direkt in einer entsprechenden Kaskade bearbeitet werden, indem man einen **Beleg eröffnet** und dazu als **Detail die Buchung selektioniert**. Durch einen **Doppelklick** auf das **Vertragsfeld** kann die Kaskade mit dem Vertrag als Detail erweitert werden.

Kernelement des Datenmodells ist die **Buchungszeile**, auf die sich verschiedene **Referenztabellen** (zum Beispiel Währungen, Konten, Kostenstellen et cetera) beziehen. In mathematischer Sicht kann ein Buchführungssystem auch als **multidimensionale Struktur** betrachtet werden, bei dem verschiedene **Parameter** die Buchungszeile mit dem Betrag als **Resultante** ansteuern.

a) Buchungszeile

Die Buchungszeile enthält Basiswerte und Parameter. **Basiswerte** sind einfache Einträge in einem Feld, deren Wert vom System nicht weiter überprüft wird. **Parameter** beziehen sich hingegen auf einen Eintrag in der entsprechenden Referenztabelle. Damit zum Beispiel der Eintrag eines Kontowertes im Buchungssatz akzeptiert werden kann, muss in der Referenztabelle „Konten" ein Datensatz mit demselben Kontowert vorhanden sein.

Basiswerte
- Text
- Betrag
- Linien-Nummer
- Datum

Parameter
- Periode
- Konto
- Vertrag
- Geschäft
- Organisation
- Anwender
- Währung

Referenztabellen können in der Regel durch Doppelklick auf dem einzelnen Feld aktiviert werden. Ein Doppelklick auf dem Kontofeld der Buchungszeile öffnet als Detail ein Formular mit Datensätzen aus der Kontotabelle, die dem Wert in diesem Feld entsprechen. Dieser Wert kann **auch Jokerzeichen** (sog. „**Wildcards**") enthalten. Ein Wert wie "MyCompany%" würde eine Liste von Datensätzen aus der Vertragstabelle bringen, deren Vertragsbezeichnung mit "MyCompany……" beginnt. **Durch Doppelklick können** so immer auch mehrere **Datensätze** in der Referenztabelle **gesucht werden**. Beim Abschluss der Neueingabe oder Änderung darf die Buchungszeile aber in allen Parameterfeldern nur noch Werte enthalten, die zu einem einzigen Referenzdatensatz führen.

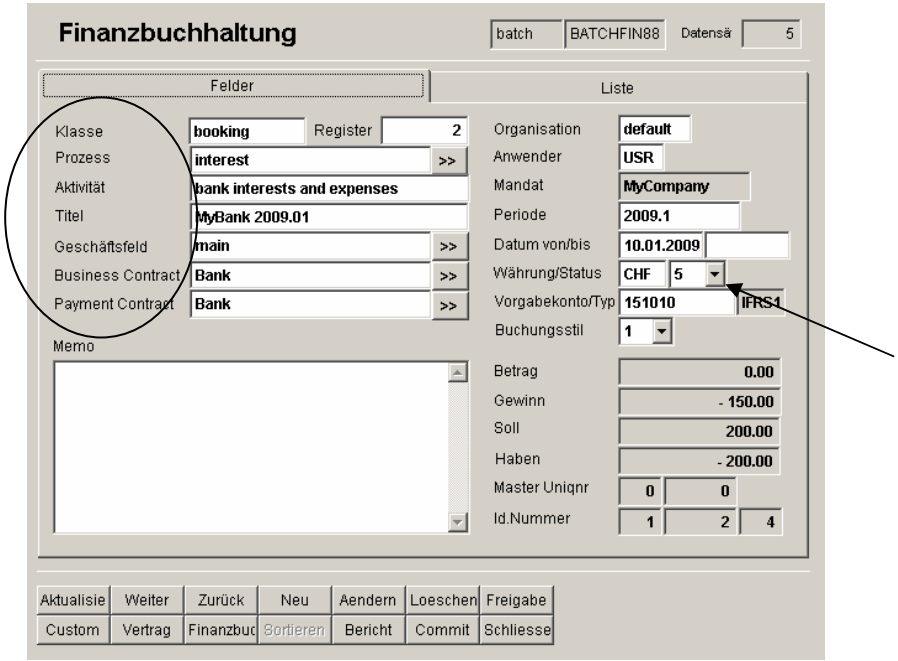
In der **Vertragstabelle** kann sowohl für das Konto wie auch für die Organisation ein Vorgabewert erfasst werden. Wird im Buchungssatz der Wert für den Vertrag eingetippt, prüft das System laufend die vorhandenen Einträge in der Vertragstabelle. Sobald nur noch ein einziger Eintrag gefunden wird, **ergänzt das System** (mit einer "**Whichpen"-Funktion**) **automatisch den Wert** für den Vertrag und übernimmt auch die Vorgabewerte für Konto und Organisation in die entsprechenden Felder der Buchungszeile. Der Zeiger wandert gleichzeitig zum Betragsfeld, sodass der Anwender mit der Eingabe des Zahlenwertes fortfahren kann.

Mit dieser Hilfe ist die Eingabe von Kreditoren und Debitoren zeitsparend und einfach. Durch die Vorgabewerte des Vertrages ist die **konsistente Zuordnung der Buchungszeilen zu Konten und Organisationseinheiten** sichergestellt und der Anwender vom häufigen Nachschlagen entlastet. Die Eingabe beschränkt sich auf wenige Zeichen für die Identifikation der Vertragsbezeichnung und das Ausschreiben des Betrags.

b) Belege und Prozesse

Basiswerte	Parameter
- Referenz - Titel - Text - Datum - Status	- Periode - Prozess - Vertrag - Geschäft - Organisation - Anwender

Eine **Anzahl Buchungssätze** werden jeweils zu einem **Beleg** zusammengefasst. Wenn Buchungssätze neu eingegeben werden, übernimmt das System in einem ersten Schritt die **Vorgaben** aus dem **Beleg** (Vertrag, Organisation, Anwender, Datum). Wird später ein Wert für den Vertrag eingegeben, **überschreibt** die oben dargestellte Hilfsfunktion die Vorgabewerte. Um neue Buchungssätze eingeben zu können, sollte ein **Buchungsfenster** folglich möglichst in einer **Kaskade** mit dem entsprechenden **Beleg** eröffnet werden.



Beleg mit Status auf Wert 5 (offen)

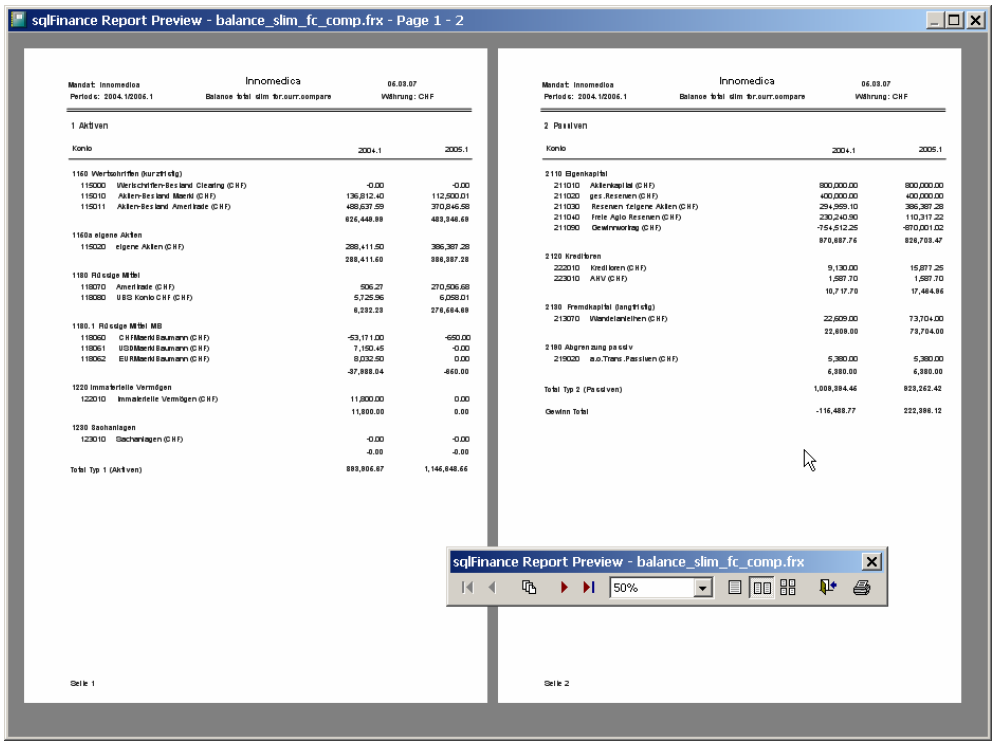
Mit den Feldern **„Klasse"**/"class", **„Prozess"**/"process", **„Aktivität"**/"activity" und **„Titel"**/"header" lassen sich die Belege wirkungsvoll **ordnen**. Der wichtigste Bezug ist das Feld **„Prozess"**. Belege mit **gleichartigen Buchungsmustern** werden zu **Prozessen** zusammengefasst. Während der **„Prozess"** als **übergeordneter** Begriff zu verstehen ist, bezeichnet die **„Aktivität"** jeweils den im Beleg erfassten **Einzelfall**. Nehmen wir zum Beispiel den Prozess **„Abschreibungen Hardware"**, dann wäre die Aktivität zum Beispiel

„Abschreibungen Computer 2009" oder **„Abschreibungen Server 2009"**. Ebenso könnten zu einem Prozess **„Bankzinsen und –spesen"** die entsprechenden Aktivitäten der einzelnen **Bankkonten** gehören.

Die **„Klasse"** ist eine Gruppierung für die Prozesse. Sind einige Buchstaben im Feld „Prozess" eingegeben, kann durch **Doppelklick** das **Prozessfenster** geöffnet werden. Die eingegebenen Buchstaben werden als **Suchwerte** übernommen. Hat der Anwender den gewünschten Prozess **identifiziert**, kann er das Fenster über die Taste **„Schliessen"**/"quit" oder mit den Funktionstasten **ctrl-F9** verlassen. Dabei werden die Werte für die Felder **„Prozess"**, **„Klasse"** und **„Aktivität"** übertragen. Der Anwender muss dann nur noch die Vorgabe für das Feld „Aktivität" leicht anpassen und im Feld **„Titel"**/"header" eine geeignete Überschrift für den Beleg eingeben.

c) Periode, Status und Abschluss

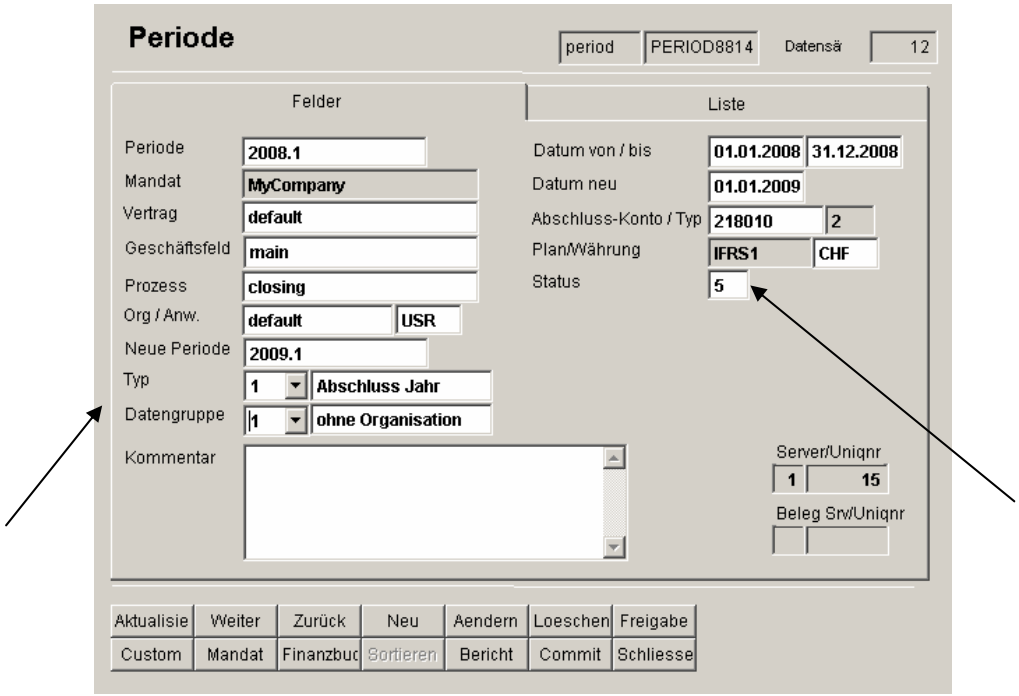
Belege haben ein **Statusfeld**. Über die Funktion „Abschliessen"/"close" kann ein Beleg für die Bilanz- und Erfolgsrechnung freigegeben werden. Zu diesem Zeitpunkt muss die **Summe aller Buchungen im Beleg** den Wert null ergeben. Ist diese Anforderung erfüllt, zeigt das Statusfeld den Wert „1"(abgeschlossen). **Bilanz und Erfolgsrechnung** dürfen in der abschliessenden Form nur Belege mit dem **Wert „1"** enthalten. Vorher können jedoch auch Berichte mit offenen und erst teilweise abgeschlossenen Belegen erstellt werden. Mit Hilfe des Statusfelds können Sie **ausgewählte Belege in den Abschluss einbeziehen** und so verschiedene Darstellungen erhalten. Dies bringt viel Flexibilität in der Abschlussgestaltung. Für die Revision ist allerdings eine Endform mit ausschliesslich abgeschlossenen Belegen erforderlich.



Beispiel für einen Jahresabschluss mit Fremdwährungen und Jahresvergleich (s. auch Anhang, mit Erlaubnis von Innomedica Holding AG, Zug)

Ebenso wie den Status können Sie auch die **Periode für die Steuerung des Abschlussprozesses** nutzen. Im Periodenfenster erlaubt eine Programmfunktion alle Belege einer Periode abzuschliessen. Erfolgt der Abschluss nur von einer **Teilperiode** in eine Hauptperiode (zum Beispiel Monatsabschlüsse im laufenden Jahr), werden sowohl Werte aus der Bilanz wie auch der Erfolgsrechnung übernommen. Beim Abschluss der Hauptperiode in eine neue **Hauptperiode** (in der Regel vom Vorjahr zum neuen Jahr) werden dagegen nur die Bilanzwerte und ein Eintrag für den Gewinn beziehungsweise Verlust der Periode übernommen. Die Summe dieser Abschlussbuchungen ergibt wiederum ein **Total von null**.

Alle **Abschlussfunktionen sind reversibel**. Abgeschlossene Perioden können wieder geöffnet und abgeschlossene Belege wieder aktiviert werden. Dadurch lassen sich komplizierte Prozeduren für die **Datensicherung** und Wiederherstellung vermeiden. Die Konsistenz für die Revision wird nicht über komplizierte Programmfunktionen erreicht, sondern durch einen nicht-redundanten Datenbankentwurf, der im Prinzip auf der Kaskade (Periode – Beleg – Buchung) aufbaut.



Abschlussperiode für das Jahr 2008 mit Status auf Wert 5 (offen)

d) Vertrag und Geschäft

Der **Vertrag** (Contract) und das **Geschäft** (Business) können auch als Schnittstelle für die **Verwaltung der Kontakte** (so genannte **"Customer Relationsship Management – CRM"**) betrachtet werden. Das **CRM** liefert Detailinformationen zur **Vertragsperson**, wie zum Beispiel Adresse, Telefonnummer, Zivilstand et cetera. Ein **Vertrag** oder ein **Geschäft** wird immer als **Detail** zu einer aktuellen **Kontaktadresse** (privat, geschäftlich) eröffnet. Der **Vertrag** bezeichnet dabei immer den direkten **Vertragspartner**, den die **Buchungszeile** betrifft. Bei **Fakturen** ist dies der **Kunde**, beim **Einkauf** der **Lieferant**, bei der **Bank** das **Bankkonto**, bei einer **Versicherung** die **Police**,

bei einem **Vermieter** der **Mietvertrag**, bei einem **Mitarbeiter** der **Arbeitsvertrag**.

Das **Geschäft** (Business) ist datentechnisch **ähnlich** aufgebaut, wird aber für die **Zusammenfassung** verschiedener Detailaktivitäten zu einem grösseren **Projekt**, meist einem **Kundenauftrag**, verwendet. Da man auf der Buchungszeile sowohl den **Vertragspartner** wie auch das **Geschäftsfeld** erfassen kann, lassen sich so zum Beispiel **Beschaffungsaufwände** direkt einem **Kundenprojekt** zuordnen. Dies erspart die mühsame **Abrechnung** über ein **Projektkonto**. **Gemeinkosten** können ebenso **internen Projekten** oder **Geschäftsfeldern** zugeordnet werden.

e) Produkt, Produktelinie und Kontierung

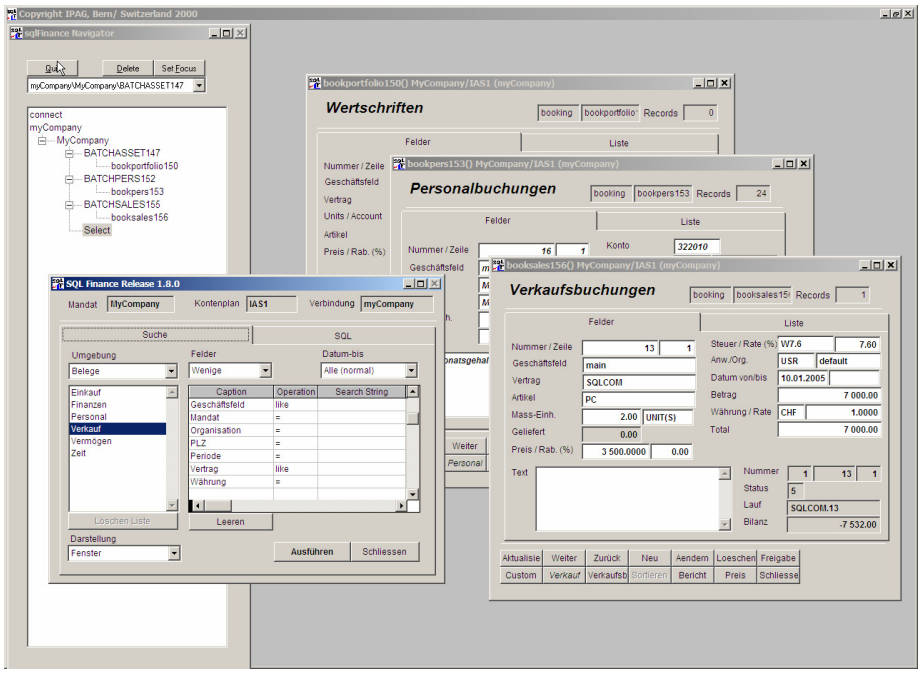
In den Anwendungsmodulen **Einkauf**, **Verkauf** und **Portfolio-Management** ist es auch möglich, Information zum **Produkt** einzugeben. Produkte können in **Stücklisten** organisiert werden. Mit der Produkteinformation ergibt sich folgende Darstellung für die Buchungszeile:

Produktbezogene Daten der Buchungszeile (Vollversion)

Basiswerte	Parameter
- Anzahl - Preis - Rabatt	- Produkt (zu einer Produktelinie gehörend) - Mass - Steuersatz

Einkauf, **Verkauf**, **Portfolio-Management** und **Gehaltsabrechnung** verwenden eine ähnliche Struktur für den Buchungssatz. Sobald das System einen Eintrag im **Produktefeld** vorfindet, wird die **Kontierung** nicht über Vorgaben im Vertrag, sondern über das Kontofeld und die **Kontierungsangaben** der **Produktelinie** ("Line") gesteuert. Im Moment der Bestätigung überprüft das Programm die **Kalkulation** der Buchungszeile (Anzahl * Preis – Rabatt = Betrag).

Spezielle Felder ermöglichen im **Portfolio-Management** und in der Finanzbuchhaltung auch **die Erfassung von Kommissionen** und **Spesen**. Enthält der Beleg Buchungen mit **Steuersätzen**, wird vom System pro Beleg und Steuersatz eine **automatische Buchung** erstellt und darin das **Steuertotal** nachgeführt. Verschiedene **Steuersätze** pro Beleg sind erlaubt. Die **Steuerverpflichtungen** können so **belegspezifisch** überprüft und in einer Abrechnung für die laufende Periode zusammengefasst werden.



Beispiel für Buchungssätze im Portfolio Management, beim Verkauf und bei der Gehaltsabrechnung (Vollversion)

Die Anwendungsstruktur in der **Gehaltsabrechnung** ist ähnlich wie im Modul „Verkauf". Der Bezug auf ein **(Dienstleistungs-)Produkt** und die dazugehörige **Produkte-Linie** (product line) liefert hier die nötigen Angaben für die **Kontierung** Spezielle Tabellen sind für die Verwaltung der monatlichen oder jährlichen Zahlungen an die Mitarbeiter und die Umlagesätze für die Alters-, Krankheits-, Unfall- und Arbeitslosenversicherung vorgesehen. Bei der Lohnabrechnung können ausgewählte Zahlungen einfach mittels "Ziehen und Fallen lassen" (Drag and Drop) über ein Buchungsfenster gezogen werden. Das Programm erstellt dann in einem ersten Schritt die **Buchungen für die Zahlungsaufträge**, in einem zweiten die **Buchungen für die Anteile der Sozialleistungen**.

Die **Umlagefunktion** kann auch in der **Finanzbuchhaltung** oder im **Verkaufsmodul** für die Erzeugung von **Zusatzbuchungen** verwendet werden. **Gehaltsbuchungen** werden vorzugsweise in einer monatlichen **Zwischenperiode** verbucht, sodass in der **Finanzbuchhaltung** nur noch die **Abschlussbuchungen** mit **Totalen** pro Konto (auf Wunsch auch pro **Kostenstelle/Organisations-einheit**) erscheinen. Damit bleibt die Buchhaltung übersichtlich und die Details der Gehaltszahlungen sind nicht ohne weiteres einsehbar.

f) Fremdwährungen

Die Vorteile der einseitigen Buchungen zeigen sich besonders klar beim Buchen mit Fremdwährungen. Da die Buchungszeile nur ein Kontofeld enthält, entsteht kein Problem in der Zuordnung der Fremdwährung zum Soll- oder Haben-Konto. Durch die **Konsistenz des Designs** wird die Arbeit mit Fremdwährungen wesentlich einfacher.

Buchungen auf **Fremdwährungskonten** lassen sich in der **Fremdwährung** wie **auch in der Hauptwährung** auflisten. So können die **Währungsgewinne** oder –**verluste** beim Abschluss wirkungsvoll in der **Erfolgsrechnung** ausgewiesen werden. Die Fremdwährung kann im **Kontofenster** eingegeben werden.

Nachfolgend ein **Datenbeispiel**, erstellt von der **Periode** ausgehend und dem Druckprogramm **„Fremdwährung (Totale und Kurse)"**/"foreign currency (totals and rates)":

Mandat: Innomedica
Period: 2008.1

Fremdwährung (Totale und Kurse)
Basis Date1 period (6/29)
Booking account like '%11805%'

04.03.09
01.01.08 - 31.12.08

1 Aktiven

Konto			Betrag FW	Kurs		Total
1180 Flüssige Mittel						
118051	CreditSuisse CHF	CHF	111'419.68	1.00	111'419.68	
						111'419.68
118052	CreditSuisse USD	CHF	-3'745.19	1.00	-3'745.19	
118052	CreditSuisse USD	USD	34'428.18	1.17	40'388.79	
		USD	34'428.18	1.06	36'643.60	36'643.60
118053	CreditSuisse EUR	CHF	409.93	1.00	409.93	
118053	CreditSuisse EUR	EUR	355.56	0.33	117.57	
		EUR	355.56	1.48	527.50	527.50
						148'590.78
Total Typ 1 (Aktiven)						148'590.78
Gesamt-Total						148'590.78

Berichtsbeispiel zur Bereinigung von Fremdwährungs-gewinnen und –verlusten (Innomedica 2008)

6. Datenorganisation

Haben Sie **sqlFinance** in der „free" oder „public version" erworben und installiert, beginnen Sie mit der **Organisation** Ihrer **Stammdaten**. Wichtige Stammdaten sind insbesondere das **Mandat**, der **Kontenplan** mit den dazugehörigen **Konten**, die diversen **Verträge**, **Gruppen**, **Perioden** und die **Organisation**.

Die Organisation der **Daten** ist auf die **Geschäftsprozesse** abzustimmen. Neben dem **Kontenplan** benötigt **sqlFinance** eine **Organisationsstruktur** („Kostenstellen") sowie eine Zuordnung von **Kunden, Lieferanten und Mitarbeitern** zu **Verträgen**. **Waren und Dienstleistungen** sind als **Produkte** zu erfassen. Die **Kontierung** kann entweder (in der Finanzbuchhaltung) über eine Vorgabe im **Vertrag** oder (im Einkauf, Verkauf, bei der Wertschriftenverwaltung und der Gehaltsabrechnung) über die **Produktelinie** gesteuert werden.

Es folgt nun eine kurze Einführung in das **Datenmanagement** von **sqlFinance.**

a. Verbindung

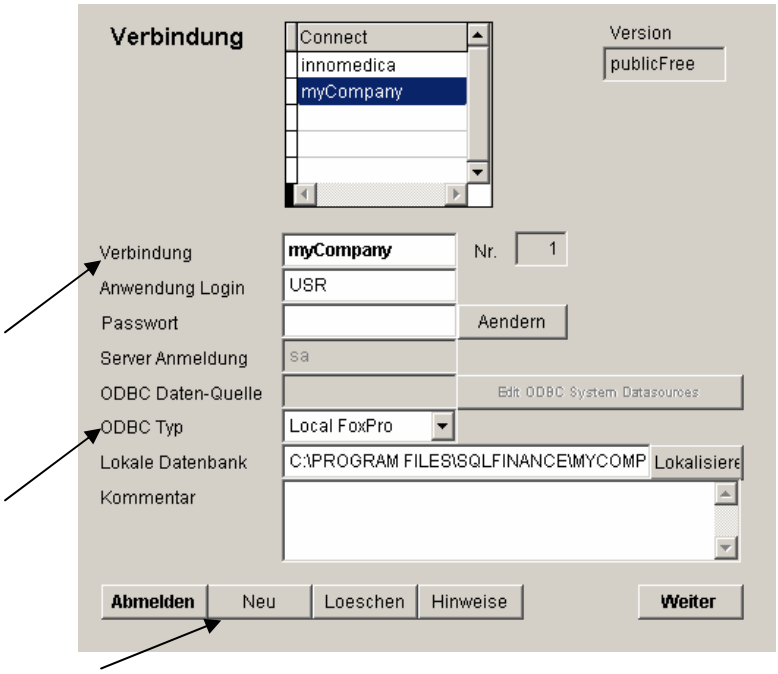
Sobald Sie das Programm **sqlFinance** öffnen, müssen Sie zuerst eine **Verbindung zur Zieldatenbank** herstellen (connection tool). Dabei können Sie entweder eine Verbindung zu einer **lokalen Datenbank** (Local FoxPro) wie "myCompany" herstellen **oder die Serverdatenbank** benutzen (Im connection tool - Feld des ODBC Typs: SQLServer). **Im Verbindungsfenster** sind sqlFinance-spezifische Informationen über die **physische Datenbank** und die **Verbindung** dazu abgelegt.

Wenn Sie **sqlFinance** das erste Mal öffnen, ist die bereits bestehende lokale Datenbank **"myCompany"** aktiviert. Indem Sie im **sqlFinance Menü** oben links auf dem Bildschirm das **Verbindungswerkzeug"/**"connection tool" aktivieren und in der Liste die gewünschte **Datenbank** anwählen, können Sie nun eine **Verbindung** herstellen. Sie klicken auf die Taste **"verbinden"/"connect"** und auf den anschliessend auf **"weiter"**/"continue". Es öffnet sich ein weiteres Fenster mit dem Namen **"Mandat"**. Hier wählen Sie Ihr gewünschtes **Mandat** und kommen durch Klicken der Taste **"weiter"** zum Ausgangsfenster, dem **Selektionswerkzeug** von **sqlFinance.**

Erstellung einer neuen Datenbank

Um eine Verbindung zu einer **neuen lokalen Datenbank** herzustellen, kopieren Sie die Datenbank **"myCompany"** (befindet sich auf ihrem Computer zum Beispiel im Verzeichnis C:\Program Files\sqlFinance) und benennen diese mit dem gewünschten **neuen Namen** (zum Beispiel **MusterFirma**). Im **Verbindungsfenster** von **sqlFinance** klicken Sie auf **"Neu"**. Sie können nun die neue **Verbindung** zu dieser Datenbank eingeben und den **ODBC-Typ** auf **Local FoxPro** setzten. Die Anzahl der **Datenbanken** ist **nicht** beschränkt. Ebenso ist die Anzahl der enthaltenen **Mandate** in jeder Datenbank **nicht limitiert**. Bitte beachten Sie, dass bei der Datenbank **"myCompany"** der **Vorgabewert** für das Verbindungsfeld **"Login Anwender"**/"user login" mit **"USR"** angegeben ist. Der Eintrag für den Anwender muss in der Datei für **Zugriffsrechte** (Privilegien/"Privileges") vorhanden sein. In dieser Datei können auch **weitere Anwender** mit allenfalls **reduzierten Berechtigungen** definiert werden.

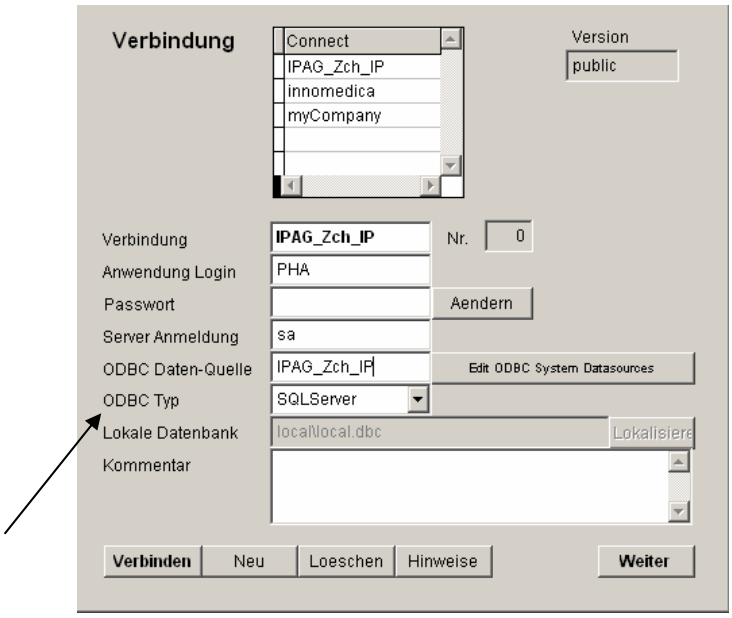
Datenbanken wie **myCompany** können auch später noch **umbenannt** werden durch Ändern der Dateinamen myCompany.dbc, myCompany.dct und myCompany.dcx. Das **Verzeichnis** kann ebenso **umbenannt** oder die **Datenbank** in ein anderes Verzeichnis an einen beliebigen Ort auf der Arbeitsstation oder im Netzwerk **kopiert** werden. Nach dem Umbenennen müssen Sie auch die in **sqlFinance** erfasste **Verbindung** ändern oder eine **neue Verbindung** eingegeben. Für die Herstellung einer Verbindung zu einer lokalen Datenbank wird kein weiteres Werkzeug benötigt.



" Verbindungswerkzeug" / "Connection tool"
der lokalen Datenbank "myCompany"

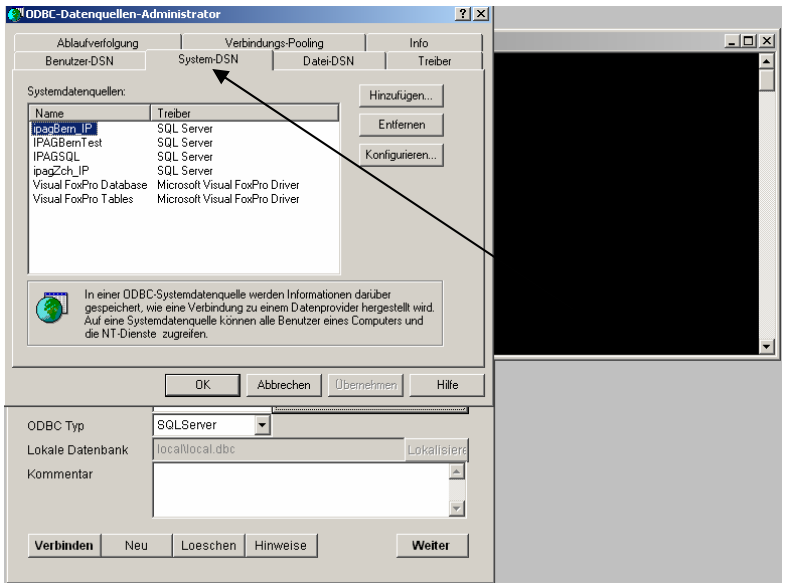
Für den **Zugang zur Server-Datenbank** (sqlServer) hingegen muss in Ergänzung zum sqlFinance Verbindungswerkzeug ein Eintrag auf der Ebene des **Betriebssystems** eingerichtet werden. Der Server kann im **lokalen Netz** oder aber auch **„remote"** über das **Internet** angeschlossen sein.

Die **System-Parameter** und **Meta-Daten** werden in der **"sqlf_db"** Datenbank im **"data"** Verzeichnis gespeichert. Durch Änderungen oder Eingaben in dieser Datenbank können Sie **eigene Tabellen** und Datenstrukturen definieren. sqlFinance hat **Werkzeuge** entwickelt für die Verwaltung dieser Metadaten.



Verbindungsfenster zur Serverdatenbank "IPAG_Zürich"

Um eine Verbindung zu einer **Serverdatenbank** herzustellen, setzen Sie den Wert des **ODBC Typ - Feldes** auf **"SQLServer"**. Klicken Sie auf die Taste "Edit ODBC System Datasources" und wählen Sie im ODBC Manager **"User DSN"** oder **"System DSN"**. Geben Sie durch **"add"** eine neue **Verbindung** ein mit dem Driver **"SQL Server"** und klicken sie auf **"Fertig stellen"**. Nun können Sie den frei wählbaren **Namen** der Verbindung eingeben. Bitte beachten Sie, dass der **Name im Verbindungsfenster** mit diesem Namen **übereinstimmen** muss. **Konfigurieren** Sie nun diese Verbindung und identifizieren den richtigen **Server** und die richtige **Datenbank**. Bitte beachten Sie, dass die Datenbank zuerst auf dem Server erstellt werden muss. Um die Verbindung zu testen, verwenden Sie **"Verbindung testen"** am Ende des Dialogs. Ist eine **Verbindung** möglich, erhalten Sie die Rückmeldung **„erfolgreich"**.



"Edit ODBC System Datasources"
(gestartet vom Selektionsfenster aus)

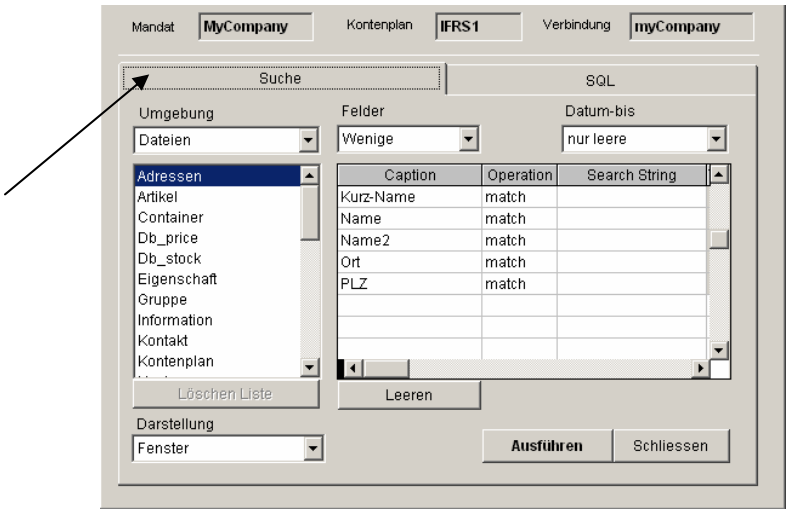
Nachdem nun eine gültige **ODBC Verbindung** vorhanden ist, geben Sie im Verbindungsfenster den Namen der Verbindung im Feld **"ODBC Datenquelle"**/"ODBC connection" ein. Setzen Sie den Namen des Anwenders für das Login im **Betriebssystem** im Feld **"Server Anmeldung**"/"Server login" (zum Beispiel **"sa"** für den **System Administrator**) ein, um in der **Zugriffskontrolle** der **Datenbank** akzeptiert zu werden. Im **Feld "Anwendung Login"** können Sie sich dann bezüglich Ihrer **Zugriffsrechte** innerhalb der **Anwendungsfunktionen** von **sqlFinance** identifizieren, beispielsweise durch die Abkürzung Ihres Namens oder Firmennamens. Durch die Eingabe eines **Passwortes** können Sie den Zutritt **exklusiv** sichern. Im Feld 'Verbindung' können Sie den Namen Ihres Unternehmens eingeben (zum Beispiel FirmaMuster) eingeben.

Es ist möglich, **gleichzeitig mehrere** Verbindungen offen zu haben und beispielsweise Formulare von verschiedenen Verbindungen von einer Datenbank zu einer anderen zu **übertragen**. Sobald Sie auf ein **Formular** klicken, wird die dazu gehörende **Verbindung** aktiviert. Durch **Ziehen der Maus** (drag and drop) können Sie Daten von einem Formular zum anderen **kopieren**, auch wenn die Formulare zu zwei verschiedenen Verbindungen gehören. Dies kann ein wirkungsvolles Vorgehen sein, um **Daten** von einer Datenbank zur anderen zu **übertragen**.

Falls Sie aus Versehen einen bestehenden **Datenbankzugriff** im Verbindungsfenster **löschen**, wählen Sie die Taste **"Neu"**/"New" und geben den **Namen** der gelöschten **Verbindung** ein. Sobald Sie auf **"Anmelden"**/"Connect" klicken, **übernimmt** das System die **'alten'** Daten.

b. Selektionsfenster

Nach dem Öffnen einer **Verbindung** und der Auswahl des aktuellen **Mandats** gelangen Sie zum **Selektionsfenster**, dem wichtigsten Hilfsmittel für die **Suche von Daten** in einer Datenbank wie zum Beispiel "myCompany". Das Selektionsfenster kann auch vom **Menu** aus aufgerufen werden (data/search) oder durch einen Klick der rechten **Maustaste** auf der Position **„select"** in der **Baumstruktur** (Treeview) aktiviert werden.



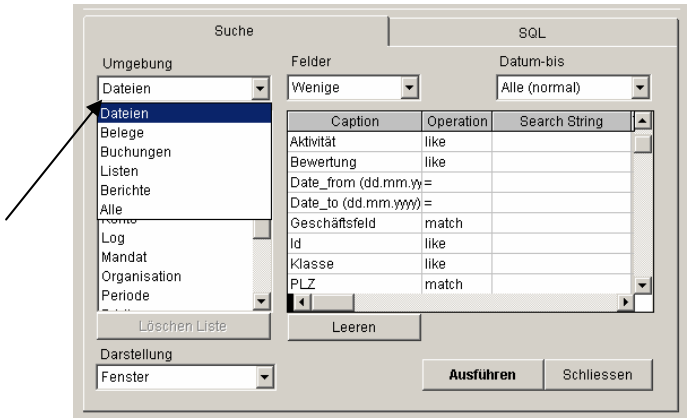
Das Selektionswerkzeug der Vollversion

Das **Selektionswerkzeug** bietet einen ersten **Zugang** zu einer aktiven **Datenbank** und unterstützt die **Suche** von Daten**.** Ausgehend von den **Suchkriterien** des Anwenders wird das Selektionswerkzeug ein **SQL Statement** (Standard Query Language Abfrageaufruf) generieren und ausführen. Die gefundenen **Daten** werden in eine **Liste** (Cursor) gelesen, in der **Memory** der Arbeitsstation gespeichert und auf dem Bildschirm in einem **Standardformular** angezeigt.

Ein solches erstes **Formular** bildet den Ausgangspunkt einer **Kaskade** von weiteren **Formularen**, die der Anwender auf Wunsch **generieren** kann. Durch Verwenden von **Master-** und **Detail-Tasten** auf dem Formular oder durch **Doppelklicks** in bestimmten **Feldern** kann die Kaskade ausgedehnt werden. Auf der **zweiten Seite** der Formularseiten (Pageframes) im **Selektionswerkzeug** (Titel: **„SQL"**) können Sie das zu verwendende **SQL Statement** selbst ändern. Ebenso können Sie auf dieser Seite die **Reihenfolge** (ordering) der angezeigten **Datensätze** beeinflussen.

a) Auswahl Dateien

Die **Auswahl** der **abzufragenden Daten** kann im **"Selektionsfenster"** /"Select" eingestellt werden. Die **obere Listbox** (Umgebung/"Environment") bietet verschiedene **Rubriken** zur Auswahl. Je nach Einstellung der Rubrik in der oberen Listbox liefert die **untere Listbox** eine entsprechende **Auswahl**:



Selektionsfenster mit der Auswahl der Dateien

- **Dateien (tables):** Wird die obere Listbox auf die Rubrik "Dateien" / "Tables" gesetzt, erhält man in der unteren Listbox zur Auswahl die Kontaktverwaltung (mit "Adressen", "Personen", "Kontakte", "Positionen"), sowie (Lager-)"Container", "Eigenschaften", "Artikel", "Lager"(-positionen), "Mass", "Prozess", "Preis", "Steuern", "Transfer" (beziehungsweise Umlagen) und "Zahlung". Zu den „Dateien" gehören also vor allem Parametertabellen und ihre Ergänzungen (zum Beispiel Produkt und Preis).

- **Belege (batches):** Unter der Rubrik "Belege" / "batches" befinden sich zusätzlich - neben der Finanzbuchhaltung - die Module: Einkauf, Personal, Verkauf, Vermögen und die Zeitverwaltung.

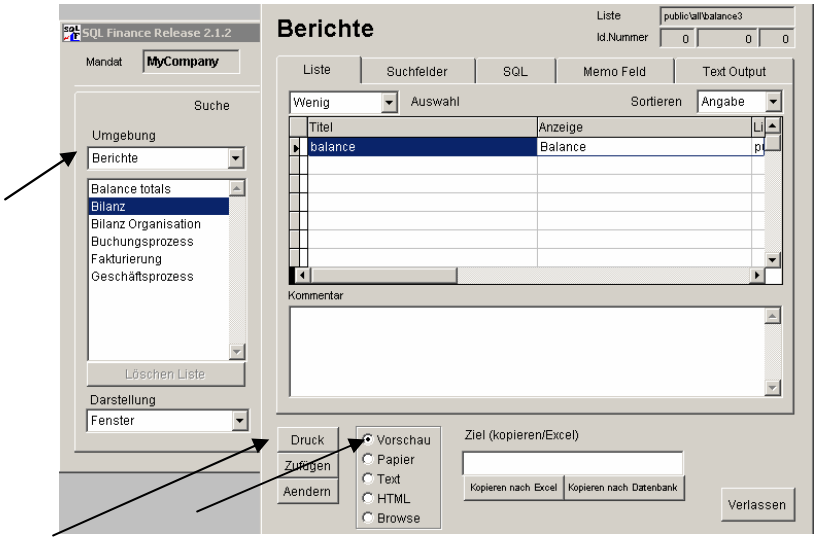
- **Buchungen (bookings):** Über die Rubrik "Buchungen" / "bookings" kann man ohne den Umweg über die Belege direkt Buchungen suchen und ändern. Neben den Finanzbuchungen finden Sie zusätzlich

Buchungen aus den Modulen: Debitoren, Einkauf, Kreditoren, Lager, Lieferung, Personal/Gehalt, Produktion, Steuern, Verkauf, Wertschriften, Zeiterfassung. Über die Master-Taste kann vom Buchungsfenster bei Bedarf das zugehörende Beleg-Fenster geöffnet werden.

- **Selektionslisten (cursors):** Selektionslisten enthalten Daten zu ausgeführten und angezeigten Selektionen, angezeigt in Formularen (in aktueller Version nicht unterstützt).

- **Berichte (reports):** Die Rubrik "Berichte" / "reports" ermöglicht den **direkten Zugang** zu Berichten. Normalerweise ist es einfacher, zuerst einen Datensatz (zum Beispiel die aktuelle Periode) zu selektionieren, und dann von diesem aus über die Taste "Berichte" "reports" das Berichtswerkzeug aufzurufen. In diesem Fall bezieht sich der Bericht automatisch auf diesen Datensatz, zum Beispiel die selektionierte Periode, liefert dann also **nur Buchungen aus dieser Periode** (beziehungsweise beim Periodenvergleich aus der aktuellen Periode und der nachfolgenden Abschlussperiode). Ebenso kann ein Bericht von einer Organisationseinheit, einem Vertrag oder Geschäftsfeld ausgehend aufgerufen werden. In diesen Fallen ist es aber nötig, ergänzend noch eine **aktuelle Periode** als Auswahlkriterium anzugeben.

  Unter gewissen Umständen ist es aber wünschenswert, Berichte **ohne vorgängige Einschränkungen** aufrufen zu können. Dies ist insbesondere dann der Fall, wenn man Daten **über mehrere Perioden** auswerten möchte, also beispielsweise bei einem **Mehrjahresvergleich** oder einer **Abschreibung** über **mehrere Jahre**. Ansonsten aber ist es dem Anwender überlassen, ob er einen Bericht eher von einem Formular aus oder direkt vom Selektionsfenster aus selektionieren möchte.

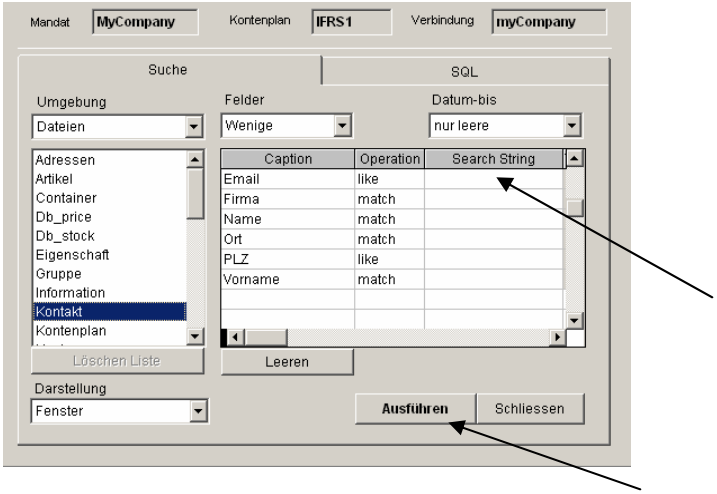


Links: Selektionsfenster, Rechts: Berichtsfenster

Buchungen werden normalerweise mit einem Klick auf die Taste "Detail" eines Belegformulars selektioniert. Sie können aber auch direkt vom **Selektionswerkzeug** aus abgerufen werden, indem Sie den Wert für **„Gruppe"** auf **„Buchungen"** setzen. Für die Dateneingabe ist es aber praktisch, wenn Belege und Buchungen in einer **Kaskade** verbunden sind.

b) Suchbegriffe

Für ein oder mehrere Felder können in der Kolumne **"Suchwert"** (Search String) **Suchkriterien** eingegeben werden. Die **Anzahl** der sichtbaren Felder lässt sich durch Setzen des Werts "Felder"/"operation" auf die Werte **"wenige"**/"few", **"viele"**/"many", **"alle"**/"all" (über der Tabelle) verändern.



"Suchmaske" / "Search String"

c) Abfrage ausführen

Mit einem Klick auf die Taste **"Ausführen"**/"" wird das **SQL Statement** generiert (Beispiel: Suche in der Tabelle „Kontakt" nach einem Unternehmen mit dem Namen „myCompany"). Abhängig vom Vorgabewert **"Anzeige"**/"Presentation" werden die Daten in einem **Formular** oder einer einfachen **Browser-Liste** angezeigt.

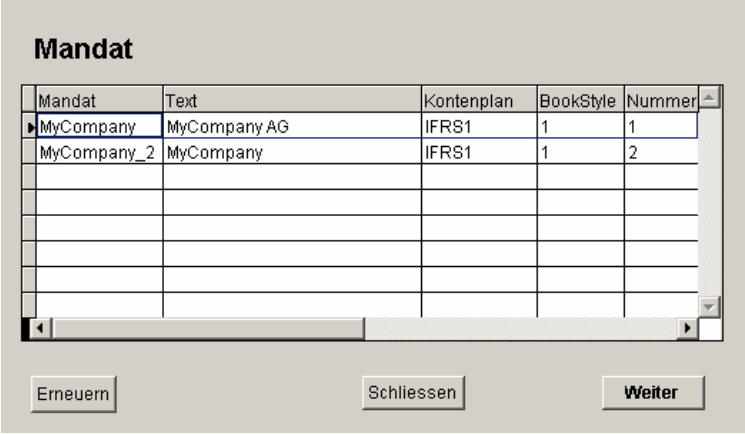
Geht die Suche zu lang, kann sie mit der **ESC-Taste** abgebrochen werden. Ist die Suche erfolgreich, erscheint das **Anwendungsformular** und **verbirgt** das Selektionsfenster in der Kaskade. Der nächste Suchschritt wird oft die Abfrage von **weiteren Details** vom **neuen Formular aus** sein. Das Selektionsfenster kann mit Hilfe des Menus oder über die Baumstruktur **wieder sichtbar** gemacht werden.

c. Mandat

Sobald eine Verbindung hergestellt ist, aktiviert das System das **Mandatswerkzeug**. Eine Datenbank kann eine beliebige Anzahl von Mandaten enthalten, wobei Mandate typischerweise verschiedene Unternehmen einer Holdingstruktur oder eines privaten Unternehmens darstellen. Vor der Selektion von Daten müssen Sie also das aktuelle **Mandat** in der Gitterliste **auswählen**. Danach können Sie auf "**weiter**"/"**continue**" klicken und befinden sich im Hauptfeld von **sqlFinance**.

Hinweis: Nicht alle Tabellen sind mandats-spezifisch aufgebaut. Daten aus der **Kontaktverwaltung** (Customer Relationsship Management CRM) beziehen sich nicht auf ein einzelnes Mandat, sondern auf **alle Mandate** einer Datenbank.

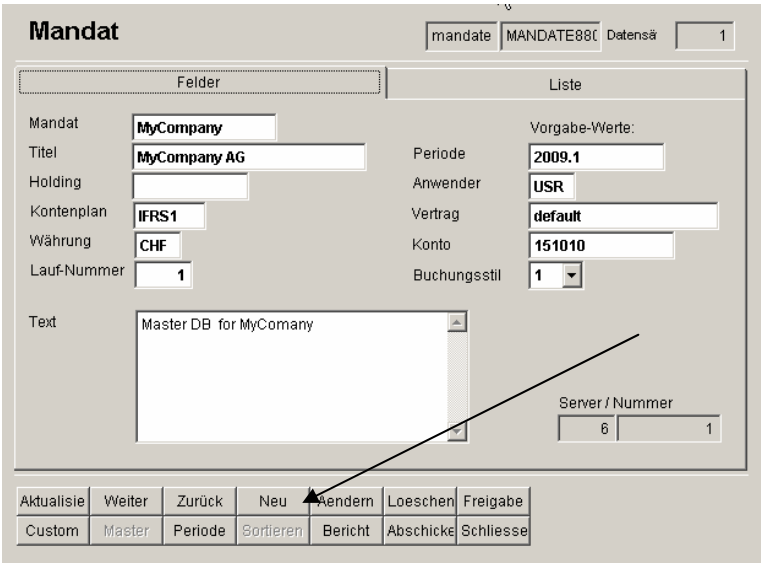
Dementsprechend kann eine physische Datenbank nur ein **einziges CRM** enthalten.



Mandat "myCompany" im Werkzeug "Mandat" / "Mandate"

Während für jedes Mandat ein eigener **Kontenplan** verwendet werden kann (oder aber auch derselbe Kontenplan bei mehreren Mandaten), bleiben **Kundendaten** wie Adressen, Kontakte, Verträge et cetera gültig für alle Mandate einer Datenbank. Wenn Sie jedem Mandat eigene Kundendaten zuordnen wollen, ist es besser, die Mandate in verschiedenen Datenbanken abzulegen.

Gleichzeitiger Zugang zu **mehreren Mandaten** ist auch in diesem Fall gewährleistet. Bei Bedarf können Sie die Daten immer noch mit der Maus (Drag and Drop) von einem Mandat zum anderen ziehen. Bei der Arbeit mit mehreren Verbindungen und Mandaten ist es am einfachsten, in der **Baumstruktur** (treeview, linker Bereich auf dem Bildschirm) durch einen **Klick** auf die **rechte Maustaste** das **Suchwerkzeug** oder die gewünschten **Formulare** zu aktivieren.



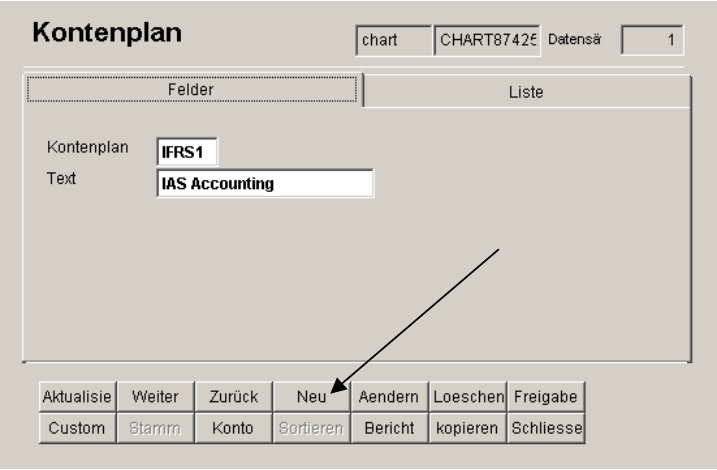
Das Mandat/mandate "myCompany"

Angaben oder **Ergänzungen zum Mandat** können in der **sqlFinance** Umgebung (Dateien, Mandat) eingegeben werden. Durch Doppelklicken auf "Mandat"/"Mandate" öffnet sich ein Fenster. Wenn Sie darin auf "Neu"/"New" klicken und Ihre Daten eingeben, können Sie ein neues Mandat speichern. Die Software wird Sie dann fragen, ob Sie das neue Mandat als **aktives Mandat** einsetzen möchten. Wenn Sie die Frage bejahen, kommen Sie zur **Mandatsauswahl** und von dort zum **Suchwerkzeug** (aktiviert auf das **neue Mandat**).

d. Kontenplan

In der Datenbank **"myCompany"** ist ein **Kontenplan** gemäss Richtlinien des IFRS (in der Schweiz Swiss-GAAP-FER) vorhanden, angepasst an die Bedürfnisse der kleinen und mittelgrossen Unternehmen (KMU) . Andere Kontenpläne und Einträge für Konten können leicht hinzugefügt werden. Bei einem neuen Kontenplan setzen Sie das Umgebungsfeld des Selektionswerkzeuges auf **"Dateien"/"Tables"**. Durch Doppelklicken auf **"Kontenplan"/"Chart"** erscheint ein Beispielfenster. Indem Sie **"Neu"/"New"** klicken, können Sie einen **neuen Kontenplan** erstellen (zum Beispiel Kontenplan "Käfer"). Parallel zum Kontenplan können Sie die **Konten** einrichten, indem Sie im Fenster das Feld "Konto"/"Account" anklicken (zweite Zeile, drittes Feld).

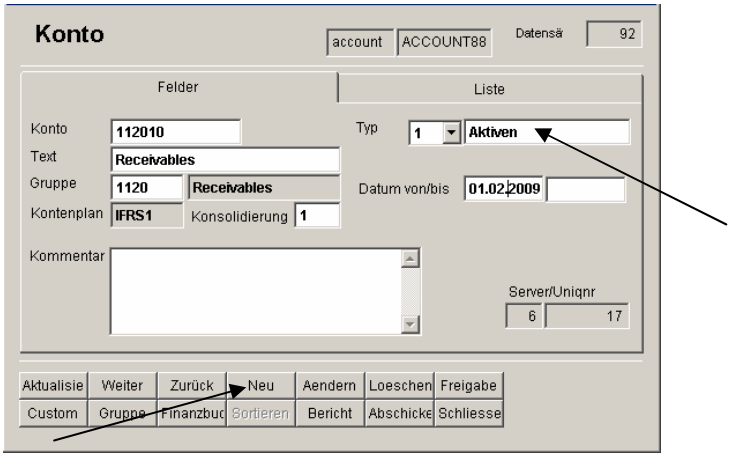
Das **Konten-Formular** kann auch über das **Suchwerkzeug** aufgerufen werden indem Sie die **Umgebung** wiederum auf **„Dateien"** setzen, mit Doppelklick auf **„Konto"** das Formular erzeugen, auf die Taste **"Neu"**/"New" klicken und dann die **Kontodaten** eingeben. Den **Beispiel-Kontenplan** können Sie abrufen, wenn Sie im Kontenplanfenster des Kontenplans **"IFRS1"** auf **"Bericht"**/"Report" und dann auf **"Vorschau"**/"Preview" doppelklicken. Im **Kontofenster** sehen Sie auf der Seite **"Liste"**/"List" die bereits erfassten **Kontonummern** und die dazugehörigen Bezeichnungen. Bitte beachten Sie, dass Sie nach der Eingabe eines neuen **Kontenplans** den entsprechenden Kontenplan im **Mandatsfenster** eingeben müssen, damit das Mandat die neue Kontenstruktur **übernimmt.**



Fenster "Kontenplan" /"Chart of accounts"

Konten sind in **„Gruppen"** und **„Typen"** organisiert. Nur die mathematisch minimal nötigen **vier Kontotypen** (1 Aktiven, 2 Passiven, 3 Aufwand, 4 Ertrag) sind vorgesehen. Dadurch bleibt die Anwendung schlank und inhaltlich

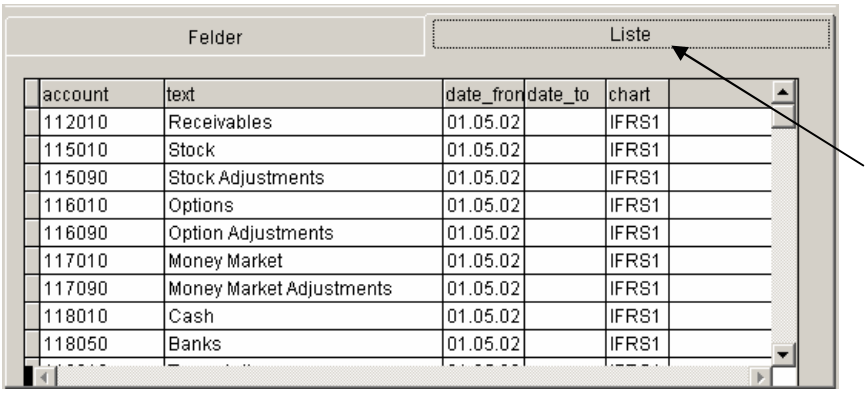
konsistent. Für andere Kontotypen wie **„Steuern"** oder „**Finanzertrag"** sollte man eher die Einteilung in „Gruppen" als die Klassifikation nach „Typ" verwenden.



Fenster "Konto"/ "Account"

Die **Gruppierung** der Konten beeinflusst die Datenstellung der Daten in den **Berichten**. Aus dem IFRS Kontenplan ist ersichtlich, wie die Konten gruppiert werden können. Jeder **Kontenplan** sollte ein **Abschlusskonto** (zum Beispiel **Konto 8000 für Bilanz**) enthalten. Dieses Konto wird im **Periodenfenster** bestimmt und ist nötig für den **Jahres- oder Zwischenabschluss** einer Periode. Ist ein Unternehmen profitabel, sollte das Abschlusskonto in den **Passiven** abgelegt werden. Bei Verlusten kann das Abschlusskonto auch in den Aktiven angelegt werden.

Über die „Gruppen" des Kontos kann auch eine **mehrstufige Deckungsbeitragsrechnung** erstellt werden. Die Konten lassen sich verschiedenen **„Teilen"**/"part" zuordnen und innerhalb dieser „Teile" nach einer Nummer **sortieren**. So können beispielsweise im ersten Teil **Warenaufwand und –ertrag** gegenübergestellt, in einem zweiten Teil **weitere projektbezogene Betriebsaufwände** verbucht, und in einem dritten Teil die **administrativen Kosten** ausgewiesen werden. In einem vierten und fünften Teil können dann zum Beispiel noch **Finanzertrag** und **Steuern** abgerechnet werden. Entsprechende **Berichte** finden sich im Formular der **„Periode"**.

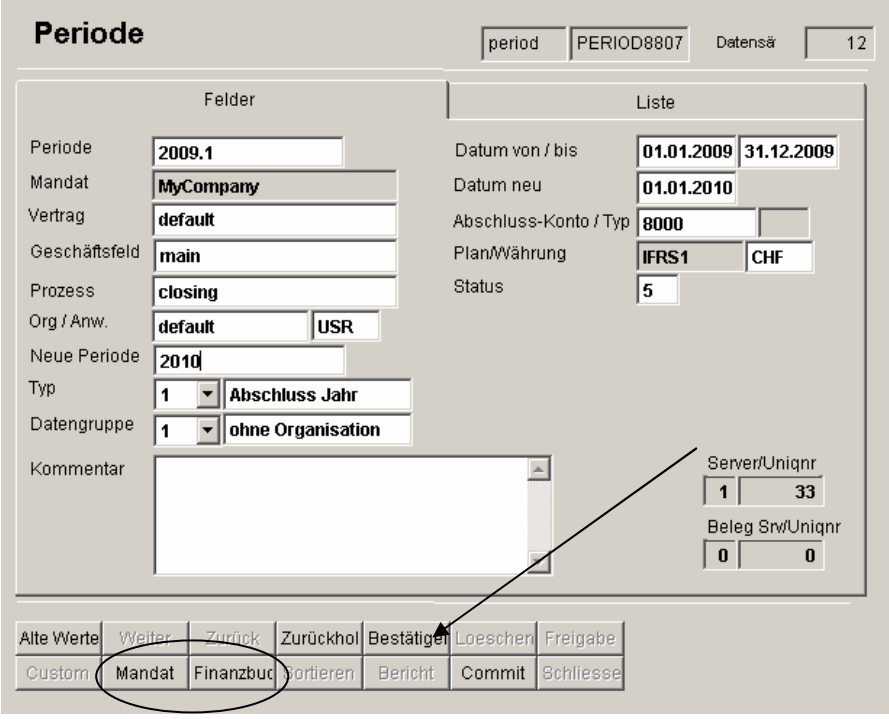


Liste mit Einträgen aus dem IFRS Kontenplan

e. Periode

Belege und deren Buchungen werden in Perioden zusammengefasst. Eine typische Periode ist die **Jahresperiode**, mit Dauer vom 1. Januar bis zum 31. Dezember (andere Daten sind möglich). Bei jeder Periode muss das Abschlusskonto angegeben werden. Beim Abschluss der Periode berechnet die Software Totale für jedes Konto und überträgt diese als Eröffnungsbuchungen in die neue Periode.

Statt Perioden können **auch Teilperioden** verwendet werden. Jahresperioden erstellen nur Totale für **Aktiven (1)** und **Passiven (2)**, sowie eine einzige **Ausgleichsbuchung** mit dem **Gewinntotal** (um so eine Summe von Null im Beleg zu erreichen). **Zwischenperioden** enthalten dagegen Buchungen für **alle vier Typen** (1 Aktiven, 2 Passiven, 3 Aufwand, 4 Ertrag) und **keine Buchung** für den **Gewinn-** beziehungsweise **Verlustvortrag**. Die Verwendung von **Zwischenperioden** ist nicht unbedingt nötig, bei der **Gehaltsabrechnung** jedoch empfohlen, ebenso bei der Handhabung grösserer Volumen von **Debitoren** und **Kreditoren**. Perioden können zum Beispiel 2009.1 genannt werden, Zwischenperioden 2009.01, 2009.02 et cetera.

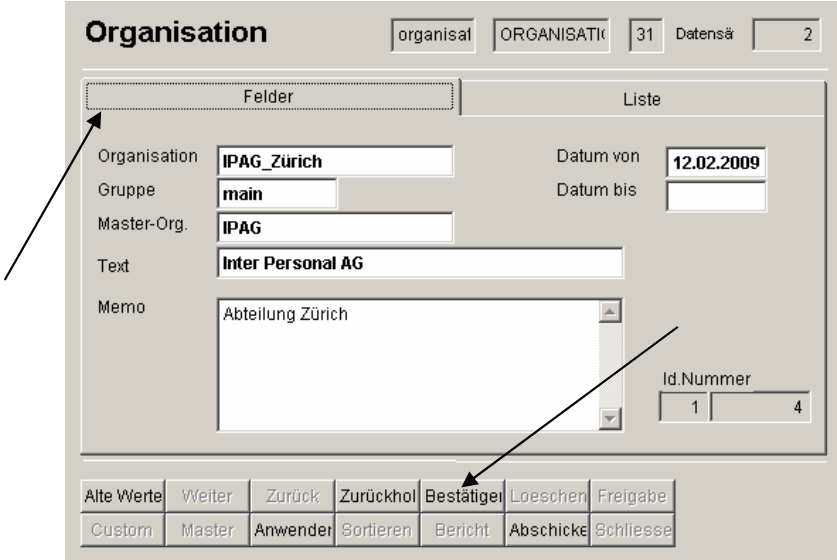


Fenster "Periode"/"Period"

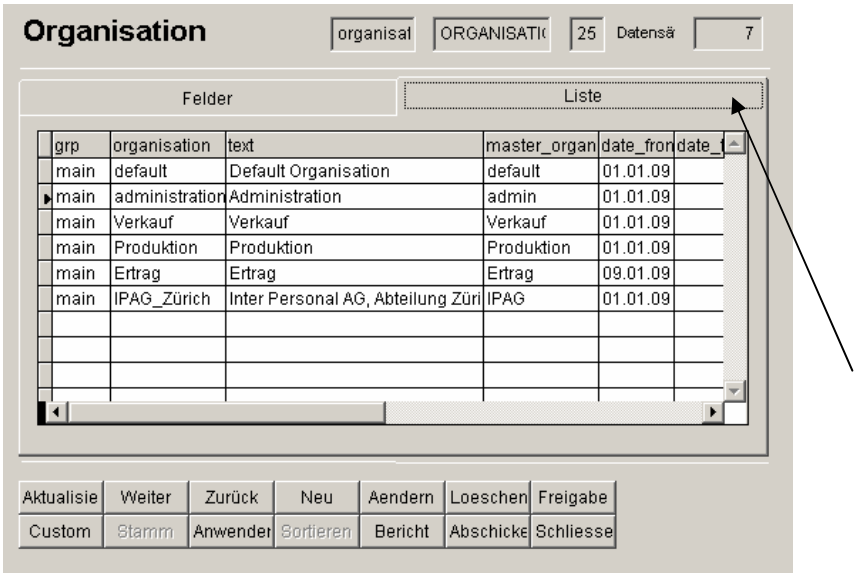
Sie kommen **direkt zum Periodenfenster** und somit zu einer neuen Periode, indem Sie im Suchfenster die Umgebung auf **"Dateien"**/"Tables" setzen und doppelklicken auf **"Periode"**/"Period". Wenn Sie nun auf die Taste **"Neu"** / "New" klicken, eröffnen Sie direkt eine neue Periode. Um die neue Periode zu speichern, müssen Sie auf die Taste **"bestätigen"**/"confirm" klicken. In der Master-Detail-Funktion steht die Periode zwischen dem Mandat und dem Finanzbeleg.

f. Organisation

In allen Anwendungsmodulen wird bei der Eingabe einer Buchung ein **Wert für die Organisationseinheit** verlangt. Mit der Organisation (zum Beispiel "Abteilung Zürich") stellen Sie die **interne Struktur eines Unternehmens** (beziehungsweise Mandats) dar. Elemente einer Organisation können zum Beispiel Geschäftseinheiten, Teams oder Abteilungen sein, die Verantwortung für Aufwand und/oder Ertrag übernehmen. In der Organisation können auch Kostenstellen oder Profit Centers dargestellt werden. Die Organisation kann eine baumförmige Struktur haben. Es ist zudem möglich, Anwender verschiedenen Organisationseinheiten (beispielsweise auch "Administration", "Verkauf", "Produktion" und "Ertrag") zuzuordnen.



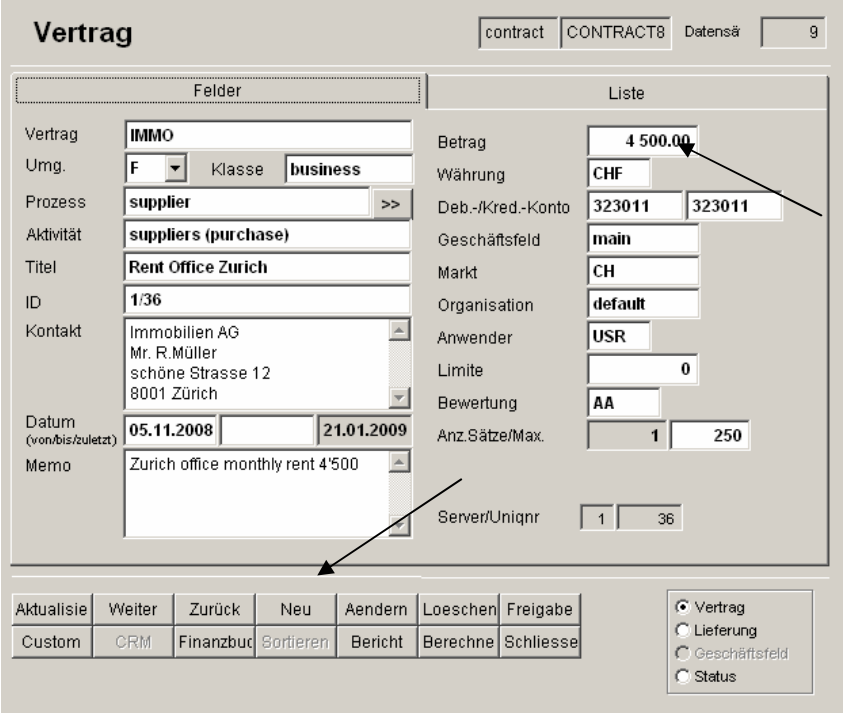
Fenster "Organisation"/"organisation"
(Seiten „Felder"/"fields" und „Liste"/"list")



Liste mit Einträgen aus dem Organisationsfenster

g. Vertrag

Einen **"Vertrag"**/"Contract" verwenden Sie für die **Darstellung einer Geschäftsbeziehung**. Dazu gehört typischerweise die **Kunden-** und **Lieferantenbeziehung** (also zum Beispiel Vertrag mit Kunde beziehungsweise Firma „Muster"). Bevor Sie mit Buchen beginnen, empfiehlt es sich zuerst für **jeden Kunden** oder **Lieferanten** einen **Vertrag** mit dem dazugehörigen **Konto** („Debitoren-/Kreditoren-Konto") einzugeben. Die **Vorgabe** für **(Geschäfts-) Vertrag** und **(Zahlungs-)Vertrag** im **Beleg** wird dann beim Buchen für **jeden neuen Datensatz** übernommen. Beim Überschreiben der Vertragsnummer ist auch ein **„Whichpen"** aktiv. Sobald das System den **Vertrag** anhand einiger eingegebener Buchstaben identifizieren kann, werden **Organisation**, **Konto** und **Betrag** vom Vertragsformular übernommen und springt der Zeiger zum **Datumsfeld**. Dies kann zum Beispiel die Erfassung von **Kreditoren** wesentlich **beschleunigen** und eine **einheitliche Kontierung** sicherstellen.



Feldstruktur des Vertrages mit monatlicher Zahlungsvorgabe von CHF 4'500

Zusammen mit dem Vertrag werden auch Informationen über die kommerzielle **Leistung** des Vertragspartners während der aktuellen Periode gespeichert. Dazu gehören **Totale** von Einkauf und Verkauf, Debitoren und Kreditoren sowie Netto-Bilanz und Gewinn. Während der Vertrag nur Informationen zur laufenden Periode enthält, lassen sich in der **Statustabelle** auch Daten zu **früheren Perioden** finden.

Im **Gehaltsmodul** wird der **"Vertrag"**/"Contract" verwendet, um **Mitarbeiter** zu erfassen. Verträge sind identifizierbar über einen einzigartigen Ausdruck (zum Beispiel „Kunde.1234"). Mit dem **„Vertrag"**/"Contract" ist die **Kundenverwaltung** verbunden (**Customer Relationsship Management**). Das **CRM** verbindet die Informationen von Personen, Adressen und Positionen in einer Tabelle für **„Kontakte"**/"Contact". Der **Kontakt** wird also mit dem

Vertrag verbunden, wobei der Kontakt die Detailinformationen enthält. Im Vertrag wird jedoch nur die Adresse als **Textfeld** wiederholt. Im Kontakt können auch **Daten von externen Quellen** (in der Schweiz zum Beispiel TelSearch oder TwixTel) mit dem **Clipboard** übernommen werden. Der **Vertrag** ist mit dem **Verkauf** (Finanzbeleg) in der **Master-/Detail Funktion** verbunden. Wenn Sie diesen jedoch direkt öffnen möchten, können Sie im **Selektionswerkzeug** die Umgebung wiederum auf „Dateien" und dann auf „Vertrag" ("Neu"/"New") setzen.

Das Modul **Customer Relationsship Management (CRM)** ist in der Gratisversion („Public Version") von **sqlFinance** nicht enthalten. Die Gratisversion verwendet der Einfachheit halber nur die Tabelle **„Vertrag"** mit Feldern für einen Vertragsnamen und eine Kunden- beziehungsweise Lieferantenadresse.

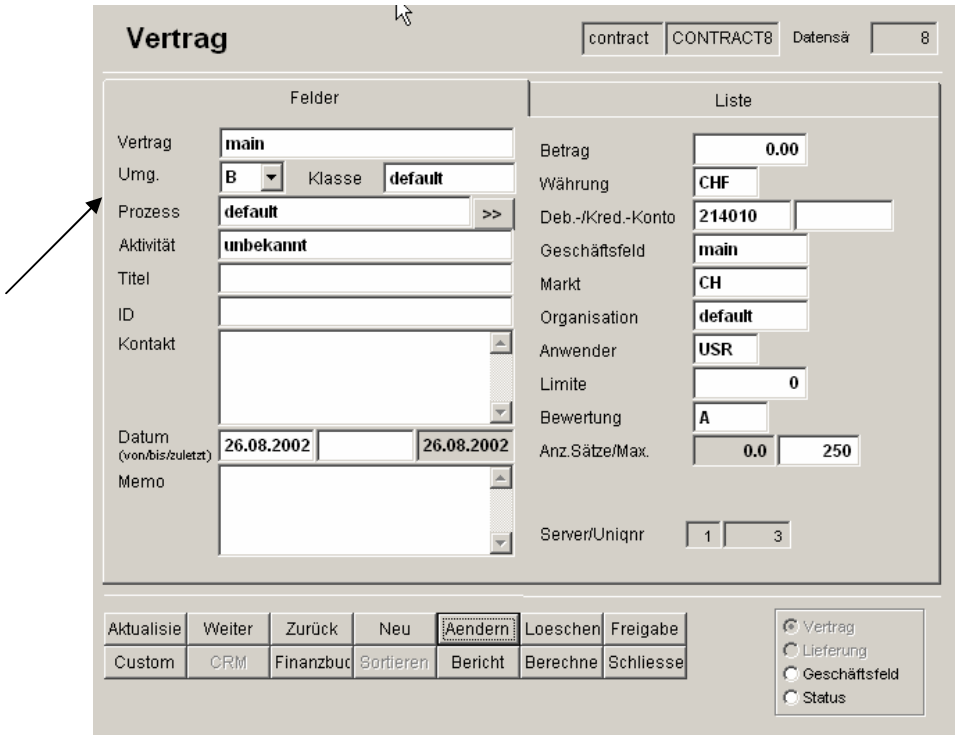
In der Vollversion unterstützt **sqlFinance** zwei Arten von Verträgen: Im Beleg können Sie einen Vertrag sowohl unter **"Contract"** wie auch **"Shipping"** erfassen. Im "Shipping" wird beispielsweise eine **Lieferadresse** hinterlegt. Unter "Contract" erfassen Sie dagegen den **Vertragspartner** für die **Rechnungsstellung**. Bei der Eingabe von Buchungen wird dann als Vorlage in der Regel der Vertrag von **"Contract"** verwendet. Nur in der **Finanz-buchhaltung** verwendet die Funktion **„Bezahlen"**/"pay" die Vorlage des **(Zahlungs-)** beziehungsweise **Liefervertrags**.

h. Geschäftsfeld

Das **"Geschäftsfeld"/"business"** wird als Parameter ähnlich wie der "Vertrag"/"Contract" eingesetzt. Für Geschäftsfelder wird dieselbe Tabelle wie beim Vertrag benützt. Im Formular für Verträge muss der Wert für das Feld **"Umgebung"**/"Environment"(Env.) auf **„B"** wie „business"("Geschäftsfeld") gesetzt werden. **Verträge** dagegen werden in demselben Feld je nach ihrer Zughörigkeit zu einer Anwendung klassifiziert**: "F"** für **"Finanzen"**/"Finance", **"S"** für **"Verkauf"**/"Sales", **"P"** für **"Einkauf"**/"Purchase", **"E"** für **"Personal"**/"Employees", **"A"** für **"Anlagen"**/"Assets" und **"T"** für **"Zeit"**/"Time".

Das **Geschäftsfeld** ist wie der Vertrag ein **Detail** zum **Kontakt**. In einer einfachen Variante können die Kunden wie alle anderen Vertragspartner und "Vertrag" erfasst werden. In diesem Fall wird "Geschäft"/"business" nur ergänzend für die Erfassung ganzer Geschäftsbereiche verwendet (zum Beispiel „Partner", „Beratung" et cetera).

Unter gewissen **Umständen** kann es allerdings wertvoll sein, wenn man Lieferungen direkt auf **Kundenprojekte** buchen kann. In diesen Fällen verwendet man **"Geschäft"**/"business" für die Erfassung der **einzelnen Projekte** (oder jeweils einer Gruppe von Projekten). Ohne Zwischenkonto können dann die Lieferungen auf dem Projekt **verbucht** und zum Beispiel in der **Kundenabrechnung** aufgeführt werden. In ähnlicher Art ist es auch möglich, **"Geschäft"**/"business" für die Profit-Center-Rechnung zu verwenden. In vielen Fällen ist es aber einfacher, dazu die **"Organisations-Einheit"**/"Organisation" (beziehungsweise **Kostenstelle**) zu verwenden.



Fenster "Geschäft" / Business" mit Umgebungswert „B"

i. Belege und Buchungen

Eine Anzahl **Buchungen** können Sie in einem **"Beleg"**/"Batch" zusammenfassen. Im **sqlFinance** Selektionswerkzeug setzen Sie die Umgebung auf **"Belege"**/"Batches". Die Master-Detail Funktion erlaubt es, zwischen dem **"Vertrag"**/"Contract", dem **"Beleg"**/"Batch" und den dazugehörigen **"Buchungen"**/"Gen. Ledger" hin und her zu springen (zum Beispiel Finanzbeleg, Finanzbuchungen). **Belege** entsprechen in etwa einer Seite **Papier** mit einer **Überschrift** und einer Anzahl gleich strukturierter **Zeilen** oder eben **Buchungen**.

In vielen Fällen wird das **Textfeld** des **Belegs** auch für die Speicherung grösserer **Textelemente** verwendet, wie zum Beispiel Briefe, Berichte, Notizen oder Kommentare zu den Buchungen. Der Text kann im ASCII. oder HTML-Format abgelegt werden. Das Textfeld kann zum Beispiel eine Kopie der anhand der Buchungszeilen erstellten Faktur im **ASCII-Format** enthalten. Dies ist besonders praktisch für die **Übernahme eines Texts in die Datenbank**. Vor allem der "Beleg"/"Batch" und das dazugehörende Feld **„Kommentar"** anerbieten sich für die Aufnahme von solchen Texten, die zum Beispiel aus der **sqlFinance Fakturierung** stammen können, allenfalls aber auch über **Belegleser** und **OCR** übernommen werden können.

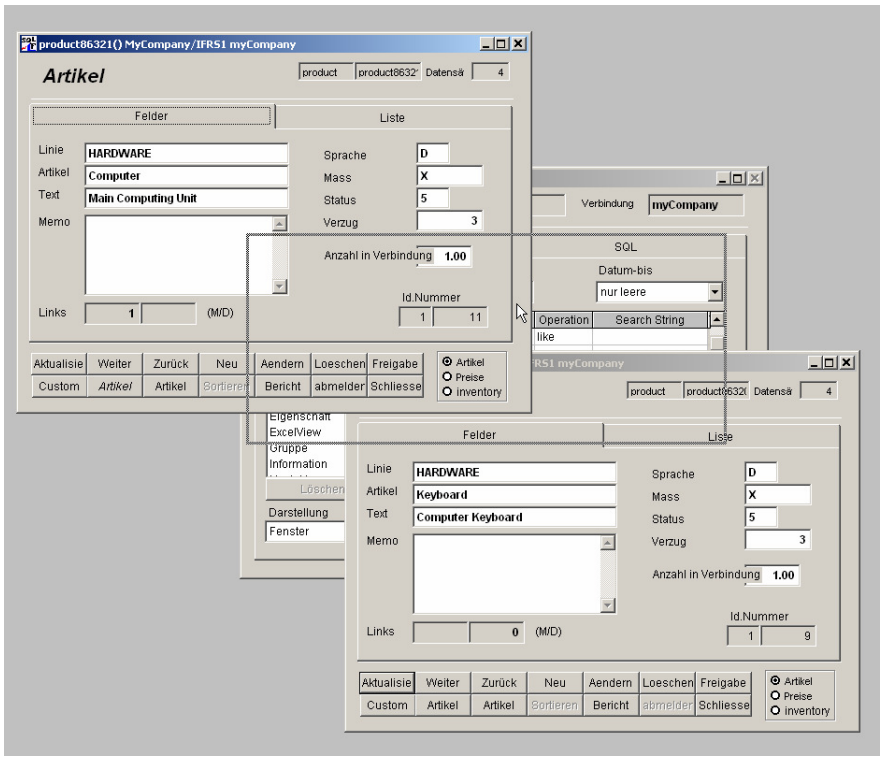
Ein Beleg ist immer **einer Periode zugeordnet**. Mit einem **Doppelklick** auf dem Textfeld kann ein **Textformular** aktiviert werden, welches Funktionen für das **Schreiben** von Texten bietet. Mit "Kopieren" und "Einfügen" (copy/paste) kann ein Text in andere Anwendungen **exportiert** oder von diesen in die Anwendungsdatenbank **importiert** werden.

Finanzbeleg / General Ledger Batch mit Periode 2009.1

Belegformulare unterscheiden sich nach der **Anwendung**, zu der sie gehören (Finanzbuchhaltung, Einkauf, Verkauf, Debitoren, Kreditoren, Vermögens-verwaltung, Gehalt, Zeiterfassung). Je nach Beleg ändert sich auch die **Darstellung des Buchungsformulars**. Einige Belege bieten Zugang zu verschiedenen Arten von Buchungen (zum Beispiel der Verkaufsbeleg bietet Zugang zu Buchungen für Debitoren, Steuern und Verkauf). Verwenden Sie die **Auswahlliste** rechts unten auf dem Formular (und einen Doppelklick), um zum gewünschten Buchungsformular zu kommen. Belege können Buchungen aus **verschiedenen Darstellungen** zusammenfassen, um ein **Null-Total** über alle Buchungen zu erreichen. Das Total aller Buchungen eines Belegs wird sowohl im Belegs- wie auch im Buchungsformular aufgeführt (rechts unten).

j. Logistik

Produkte können in der Anwendung für Verkauf und Einkauf angegeben werden. Mittels Ziehen der Maus (drag and drop) lassen sich Produkte zu Stücklisten zusammenfassen. Zieht man Produkte über eine Buchung, wird ein neuer Buchungseintrag gestartet. In der Tabelle für **"Lager"/"inventory"** (Vollversion) können Sie **Lagerbestände** sowie durchschnittliche Verkaufspreise verwalten. Die **Preistabelle** enthält Informationen zur Preisstruktur, abhängig vom Markt, mit der Möglichkeit für die Angabe von Rabatten pro Markt oder Kundengruppe. Produkte gehören zu Produktelinien, und die **Tabelle „Linien"/"Line"** enthält Angaben, wie der Verkauf oder Einkauf bestimmter **Produktegruppen** zu verbuchen ist. In der Produktelinie sind auch Informationen über den anzusetzenden **Steuersatz** enthalten.



Master- und Detail-Artikel werden mittels Ziehen mit der Maus (rechte Maustaste, grauer Rahmen als gezogener Schatten) in der Stückliste verbunden (Tastatur als Detail eines PC's)

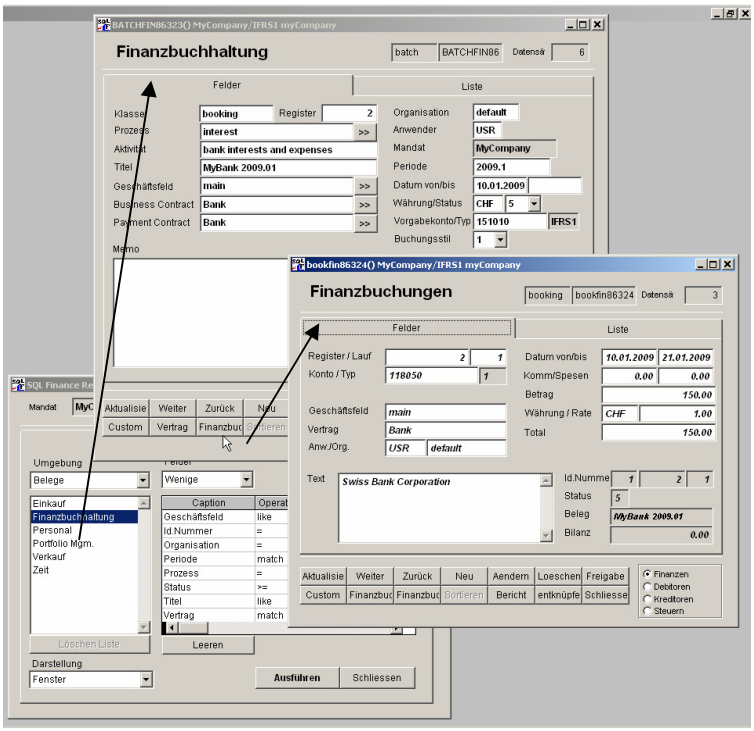
III. Anwendungs-Module

1. Arbeiten mit der Finanzbuchhaltung

Ein neuer Anwender möchte in vielen Fällen so schnell wie möglich mit der **Eingabe von Buchungen** beginnen können. Zu diesem Zweck wird die Software mit der Beispieldatenbank und dem Mandat **„myCompany"** ausgeliefert. Beim ersten Start des Programms kommt der Anwender sofort zu dieser Datenbank und diesem Mandat und kann mit dem **Selektionswerkzeug** Daten abrufen.

Zwei **Beispielbelege** mit Buchungen zeigen, wie Daten einzugeben sind. Mit Hilfe des **Selektionswerkzeugs** lässt sich ein **"Beleg"**/"Batch" Fenster öffnen. Setzen Sie den Wert **"Umgebung"**/"Environment" auf **"Belege"**/"Batch" und führen Sie dann einen Doppelklick auf der Position **"Finanzen"**/"Gen. Ledger**"** aus. Mit einem Klick auf die Taste **"Neu"**/"New" lässt sich nun ein **neuer Beleg** erfassen.

Der dazugehörige **Vertrag** kann durch die **Master-Detail Funktion** abgefragt werden. Sie können passende Verträge mittels Eingabe einer **Wildcard** in einem **Vertragsfeld** und anschliessendem **Doppelklick** suchen. Verwenden Sie für die Wildcard das Zeichen **„%"**. Es öffnet sich dann ein neues Fenster. In diesem speichern Sie allenfalls Ihre neuen **Vertragsdaten** und übernehmen diese mit **Klick** auf die Taste **"Schliessen"**/ "Close" (oder Tastenkombination ctrl-F9). Sobald das **Buchungsformular** erscheint, verschwindet das **Vertragsformular**.



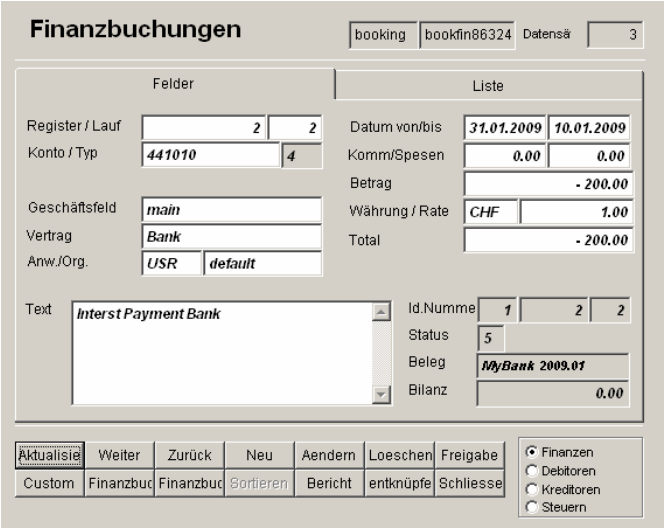
Finanzbeleg und Finanzbuchungen: Schritt 1: Klick auf die Zeile Finanzbuchhaltung" in der Umgebung des Selektionsfensters; Schritt2: Klick auf die Taste „Finanzbuchungen" im Finanzbeleg

Verwenden Sie wiederum die **„Master"-** oder **„Detail"-Tasten** (zweite und dritte Taste in der zweiten Tastenlinie, direkt unter den Tasten „Weiter"/"next", beziehungsweise „Zurück"/"previous" - diese Taste ist ggf. mit dem Namen des Formulars angeschrieben (zum Beispiel „Finanzbuchhaltung" oder „Finanzbuchungen"), um von einem Formular zum anderen zu gelangen. Mit der Taste **„Freigabe"**/"free" können Sie die beiden Formulare in der Kaskade voneinander **trennen** und beide **sichtbar** machen.

Finanzbeleg mit Bankvertrag

Jetzt kann mit der **Eingabe von Buchungen** begonnen werden. **Vorgabewerte** wie „2009.1" für die aktuelle Periode, „default" für Organisation und Vertrag, und „USR" für Anwender werden automatisch eingesetzt und können jederzeit geändert werden.

Ein **IFRS Kontenplan** steht zur Verfügung und kann je nach Bedarf angepasst werden. Eine **Liste der verfügbaren Konten** lässt sich abrufen, indem man die **Wildcard „%"** in ein Kontofeld setzt und dann mit einem Doppelklick zur Parameter-Tabelle springt. Nach der Wahl des gewünschten **Kontos** kann man mit der Taste **"Schliessen"**/"**drop**" oder der Tastenkombination **ctrl-F9** zum **Buchungsformular** zurückkehren, wobei der **Kontowert** übertragen wird.



Fenster für die Eingabe von Finanzbuchungen (Zinszahlung Bank CHF 200.--)

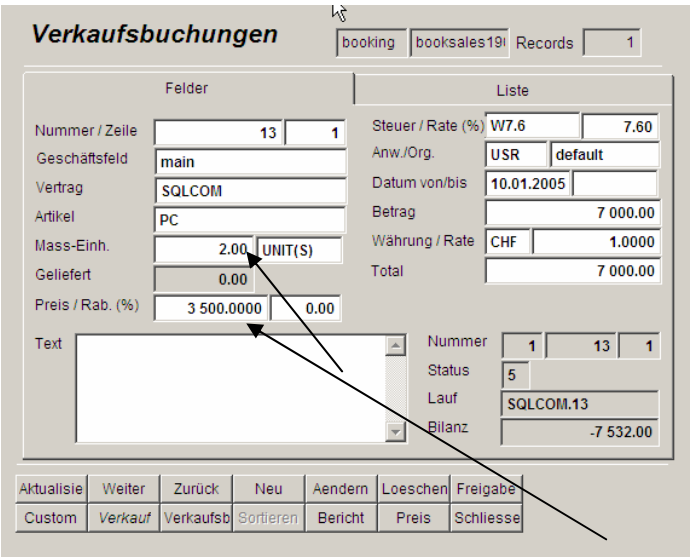
Wird ein **neuer Datensatz** eingegeben und möchte man die Werte des letzten Datensatzes übernehmen, kann man die Taste **"alte Werte"**/"old values" benützen (oder die Tastenkombination **ctrl-q**). Die Wirkung neuer Buchungssätze auf die Bilanz- und Erfolgsrechnung kann jederzeit durch **Drucken** entsprechender **Berichte** und Ansicht auf dem Bildschirm verfolgt werden. Entsprechende Berichte findet man am einfachsten im Formular der **aktuellen Periode**, die mit dem Suchwerkzeug abgerufen wird. Ein Klick auf die Taste **"Bericht"**/"Report" bringt eine Auswahl von Druckprogrammen, zum Beispiel die häufig verwendete Variante **„Totale (Bilanz)"/**„Total (Balance)".

2. Ein einfaches Verkaufsbeispiel

Die **Vollversion** von sqlFinance enthält auch **Module** für **Einkauf und Verkauf**. Einkaufs- und Verkaufsbuchungen werden ähnlich wie Finanzbuchungen erfasst. Man sollte dabei möglichst immer von einer Kaskade **Beleg-Buchung** ausgehen. Dazu öffnen Sie zuerst einen **Beleg** mit Hilfe des Selektionswerkzeugs, um dann das Formular für die **Buchungen** mit der Detail-Taste (dritte Taste von links in der zweiten Kolonne) zu öffnen. Gehen Sie für die **Erfassung eines einfachen Verkaufs** wie folgt vor:

a. Öffnen Sie mit dem Selektionswerkzeug einen **Verkaufsbeleg** (Umgebung: Belege), erfassen Sie einen neuen Datensatz (Taste **"Neu"**/"New"). Die automatisch eingegebenen **Vorgabewerte** sollten die Bestätigung der Transaktion erlauben. Beachten Sie, dass das **Total** noch mit dem Wert **null** angezeigt wird.

b. Gehen Sie mit Hilfe der **Detail-Taste** in das **Buchungsformular**. Klicken Sie auf die Taste **"Neu"**/"New" um einen Neueintrag zu erhalten. Setzen Sie zum Beispiel den Wert **"Artikel"**/"product" auf **„PC"** (personal computer „PC" aus der Produktelinie Hardware „HW") und geben Sie einen **Preis** von CHF 3'500 ein. Das Feld **"Einheiten"**/"Units" setzen Sie auf den Wert 2 (2 PC zu CHF 3'500), im Rabattfeld ergänzen Sie allenfalls einen Wert für den Rabatt.

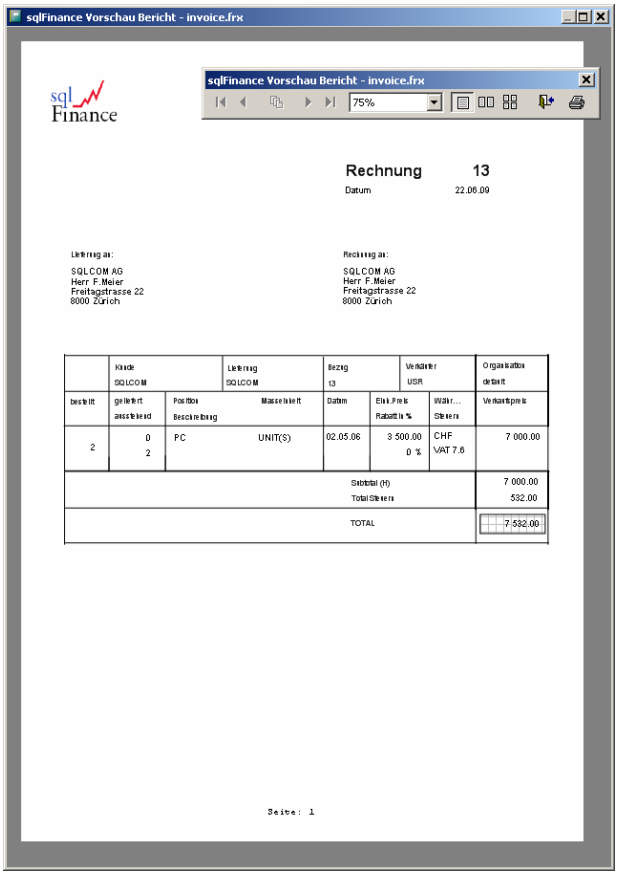


Eingabe der Verkaufsbuchung: 2 PC's zum Preis von CHF 3'500

c. Nach dem **Abschluss** der Eingabe beträgt das **Total des Belegs** (rechts unten) 2 x CHF 3'500.-- + CHF 532.-- (MWST 7.5%): CHF – 7'532.--. Das **Total** ist **negativ**, da Verkäufe gemäss **Kontenplan** unter **Typ 4** (Erträge) zu erfassen sind. Der Wert für das Kontenfeld („412011": „Sales Goods") wird **automatisch** eingesetzt anhand von Angaben in der **Produktelinie** (die Produktelinie **„HW"** ist eine Mastertabelle zum Produkt **„PC"**). Um die **Kontierung** zu ändern, selektionieren Sie die Tabelle **„Linie"** mit dem Suchwert **„HW"**, oder gelangen Sie mit Doppelklicks auf dem Produktefeld und dann wiederum auf dem Linien-Feld in das Linien-Formular.

d. Mit dem **Berichtswerkzeug** kann eine **Rechnung** an den Kunden gedruckt werden. Die Rechnung kann je eine **Faktura-** und **Lieferadresse** enthalten,

sowie Angaben über **Nachlieferungen** („backorders"). Verschiedene **Steuersätze** können in derselben Rechnung verwendet werden, und ebenso Berechnungen in **Fremdwährungen**. Die Produkteinformation wird vervollständigt durch **Liefereinheiten** und **Mass**, sowie **Rabatte** pro Produkt, Markt oder Kunde. Das Verkaufsmodul gehört zur Vollversion (**„full version"**) von **sqlFinance** und enthält auch eine umfassende **Kundenverwaltung** (CRM Customer Relationship Management), eine **Stückliste** (BOM Bill of Materials) und eine Expertensystem für **Preise** (price expert).



Bericht mit Faktura (aus dem Verkaufsmodul)

e. Schliessen Sie das Verkaufsformular und holen Sie (vom Verkaufsbeleg aus) ein **Debitorenformular** (mittels einem Klick auf die Position **"Debitoren"**/"Receivables" auf der **Liste** rechts unten im Formular).

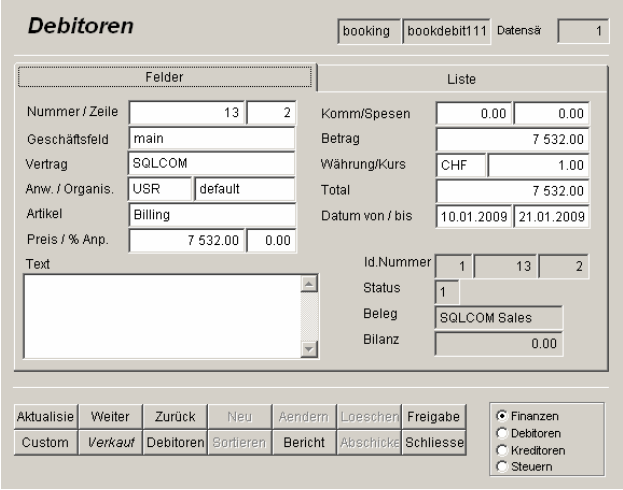
Die **Optionen** dieser Liste umfassen:

- **Verkauf**
- **Debitoren**
- **Steuern**
- **Lagerabruf**
- **Lieferung**

Lagerabruf und Lieferung werden nur in der Server-Version unterstützt.

f. Erfassen Sie eine **Debitorenbuchung**. Wählen Sie als Produkt zum Beispiel „Rechnung" („invoicel") und setzen Sie im Betragsfeld CHF 7'532 ein. Der Betrag kann auch vom Feld **„Bilanztotal"** mit der rechten Maustaste über das Zielfeld gezogen werden. Beachten Sie, wie der Betrag automatisch von -7'532 zum Gegenwert von 7'532 wechselt. Ein derartiges Wechseln des Werts kann auch durch **Doppelklick** auf dem **Zahlenfeld** erreicht werden.

g. Nach dem Erfassen des Debitors beträgt das **Belegtotal** wiederum **null**. Jetzt kann der Beleg auf **zwei verschiedene Arten** an die **Finanzbuchhaltung** übergeben werden: **entweder** durch **Abschicken** der **unveränderten Buchungen** als Teil des aktuellen Belegs, **oder** durch **Berechnen** der **Totale des Belegs pro Konto** (beinhaltend Verkauf, Debitoren und Steuern). In **letzerem Fall** wird ein **neuer Beleg** in der Finanzbuchhaltung mit den entsprechenden Buchungen erstellt. Dieser Beleg verwendet die **Abschlussperiode** der im **Verkaufsbeleg** eingesetzten Periode. Durch Ändern des **Statuswerts** auf **„1"** und betätigen der Taste **„ändern"**/"update" kann man den Beleg **unverändert** abschliessen. Durch verwenden der Taste **„abschicken"/"commit"** wird ein **neuer Beleg** mit **Totalen pro Konto** in der **Finanzbuchhaltung** erstellt. In **beiden Fällen** kann mit **derselben Taste** (angeschrieben **„recall"**/"rückgängig") die Transaktion wieder **rückgängig** gemacht werden.



Buchungsformular "Debitoren" mit Totalbetrag von CHF 7'532.--

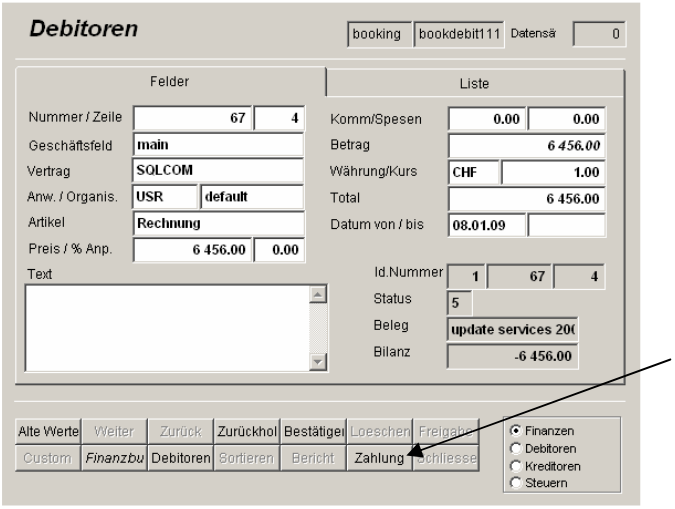
3. Debitoren-/Kreditoren

Debitoren (bzw. analog auch die Kreditoren) können in **Finanz- und Verkaufs- bzw. Einkaufsbelegen** wirkungsvoll organisiert werden. Man kann zum Beispiel pro **Schlüsselkunde** einen Beleg offen halten und darin die **Debitoren-positionen** sammeln. So lässt sich leicht überwachen, welche **Kundenforderungen** noch **ausstehend** sind.

Zum Jahresende kann der **Beleg** mit einer einzigen Position auf **null abgeschlossen** werden. Durch **Ziehen mit der Maus** kann die **Abschluss-buchung** dann im neuen Jahr als **Saldovortrag** übernommen werden. Zieht man mit der Maus statt der Buchung den **Beleg**, erfolgt diese Funktion **automatisch** und gegebenenfalls über eine ganze **Belegsliste**.

Pro **Beleg**, **Periode** oder **perioden-übergreifend** kann ein **Bericht** der **offenen und abgeschlossenen** (mit einer anderen Buchung verbundenen) **Buchungen** erstellt werden. Diese lassen sich nach **Kunden** ordnen und ergeben so einen **Kontokorrent-Auszug**, der sich für die Überwachung von **Debitoren**, aber auch anderen **Zahlungen** eignet.

Zahlt ein Kunde nur einen **Teil** der Rechnung, kann entweder die **Rechnung geändert** (mit automatischen Anpassungen bei den Steuerwerten) oder die **Differenz** in einem **Spezialkonto** abgerechnet werden (zum Beispiel Verkaufs-verluste). Die Rückvergütung von Mehrwertsteuern kann dann am Jahresende dem Total der Verluste entsprechend berücksichtigt werden.



Formular „Debitoren" vom Finanzbeleg aus aufgerufen, mit Taste für „Zahlung"/"payment" (nach Betätigen der Taste „zufügen"/"add")

Die **Kontowahl** erfolgt bei den Debitoren ähnlich wie im Verkauf über das **Produkt** bzw. die **Produktelinie**. Mit der Angabe des Produkts kann der **Zahlungsmodus** (z.B. Frist 30 Tage) **frei definiert** werden. Je nach **Kontierungsangabe** der **Produktelinie** werden die Einträge dann **im betreffenden Konto erfasst** und können dort nach **Zahlungseingang** auch **abgebucht** werden.

In das **Debitorenformular** gelangt man meist von einem **Finanz- oder Verkaufsbeleg** aus. Durch **Doppelklick** auf die **Auswahlliste** (rechts unten)

kann das Debitorenformular als **Detail des Belegs** aufgerufen werden. Ein Klick auf die Taste **"Neu"**/"New" startet dann die **Eingabe einer neuen Buchung**. Es ist auch möglich, diese Formulare direkt vom **Suchwerkzeug** aus aufzurufen. Dazu muss der Wert für die Umgebung auf **"Buchungen"** gesetzt werden.

Für die **Bereinigung** von Debitoren ist es wichtig, dass Sie Buchungen miteinander **verbinden** können (offene Kreditoren werden ggf. ebenso überwacht). In allen Buchungsformularen ist es möglich, Buchungen miteinander zu verbinden, indem Sie mit der **rechten Maustaste** eine Buchung über die andere ziehen. Anhand der **Verbindung** können Sie die anderen Buchungen jederzeit auffinden, indem Sie die **Detailtaste** des **Buchungsformulars** anklicken.

Sobald eine Buchung eine Verbindung enthält, erscheint sie in **Kursivschrift**. Wird die Verbindung aufgehoben, wechselt die Darstellung wieder auf **Normalschrift**. Wird eine Buchung über eine andere gezogen, fragt die Software den Anwender, ob er einen **neuen Eintrag mit Verbindung**, **ohne Verbindung**, oder **nur eine Verbindung** (ohne Neueintrag) erhalten möchte.

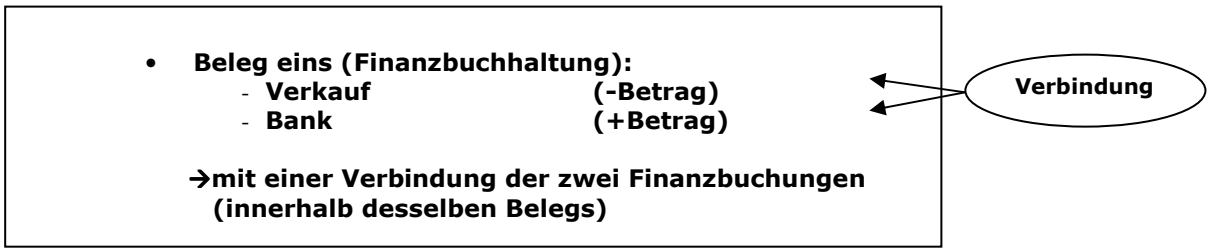
Beim **Abschluss des Beleg** muss nicht nur das **Total** aller Buchungen **null** betragen, sondern auch das **Total** jeder **Verbindung**, die Buchungen **innerhalb des Belegs** oder über **mehrere Belege** erfasst. Für die **Verwaltung** dieser Verbindungen bietet **sqlFinance** ein **praktisches Verbindungswerkzeug („Clearing Tool"**).

Will man die **Bezahlung eines Verkaufs** verbuchen, gibt es grundsätzlich **zwei Vorgehensweisen**:

- **Direkte Verbuchung**: Die **erfolgswirksame Buchung des Verkaufs** kann direkt mit dem **Zahlungseingang der Bank** in Verbindung gebracht werden. In diesem Fall werden **beide Buchungen** meist in **demselben Beleg** verbucht und miteinander **verbunden**.

- **Verbuchung über Debitoren-Clearing**: Der **Verkaufsbeleg** enthält die **Verkaufsbuchungen** und eine oder mehrere **Debitoren-Buchungen**. Der **Zahlungs-Beleg** enthält die **Gegenbuchung** zu dem oder den **Debitoren** und die **Buchung** zum **Zahlungseingang auf der Bank**.

Bei der **direkten Verbuchung** kann die **zweite Buchung** über die Taste **"Zahlung"**/"Payment" mit der **vorangehenden Buchung** in Verbindung gebracht werden. Diese Funktion ist **während der Eingabe** verfügbar, also nachdem man die Taste **„zufügen"**/"add" betätigt hat. Dabei wird der **negative Betrag** eingesetzt und der Vertrag gemäss **Vorlage im Beleg** auf den **„Vertrag Zahlung"** geändert. So können zum Beispiel die **Debitoren** einer **Woche** oder eines **Monats**, deren Zahlungsmodus über **dasselbe Bankkonto** erfolgt, in einem **Beleg** zusammengefasst werden.



Direkte Verbuchung von Verkauf und Zahlungseingang

- **Beleg eins (Verkauf oder Finanzbuchhaltung):**
  - **Verkauf (-Betrag)**
  - **Debitorenbuchung (+Betrag)**
- **Beleg zwei (Finanzbuchhaltung):**
  - **Debitoreneingang (-Betrag)**
  - **Zahlungseingang Bank (+Betrag)**

→mit einer Verbindung der zwei Debitorenbuchungen (über die beiden Belege)

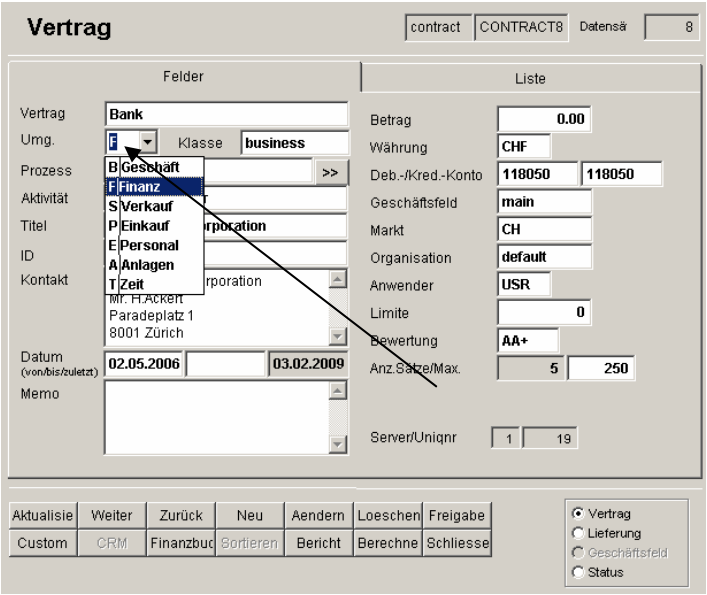
Verbindung

Verbuchung mit Debitoren

Nachfolgend finden Sie eine **ausführliche Anleitung** zur **Verbuchung** über ein **Debitoren-Clearing**. Für die **Kreditorenverwaltung** ist **analog** vorzugehen.

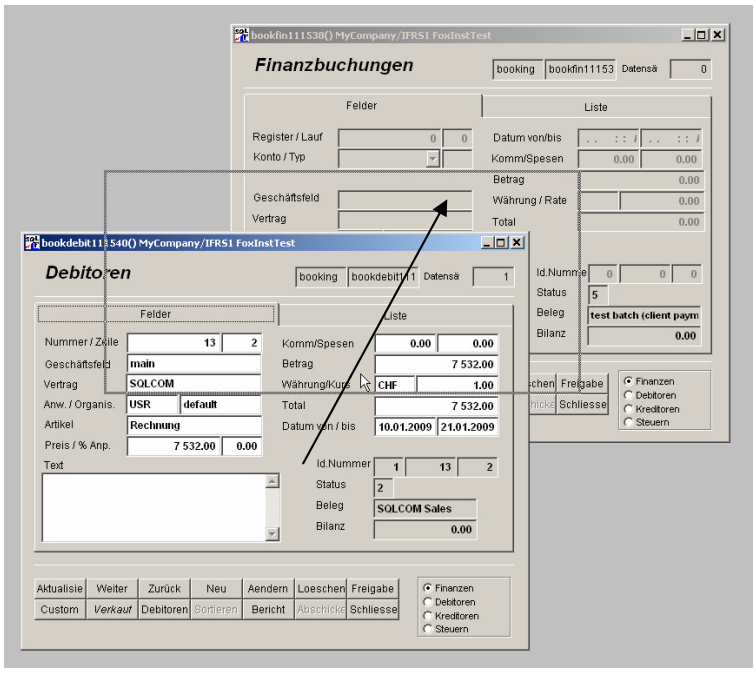
a. Öffnen Sie eine Liste offener **Debitoren-Positionen** (wählen Sie in der Umgebung des Selektionswerkzeuges "Buchungen"/"Gen.Ledger", dann Debitoren).

b. Falls noch nicht geschehen, erfassen Sie einen **Vertrag** für Ihr **Bankkonto** (mit Wert **„F" Finanzbuchhaltung** für das Feld **„Umgebung"**). Als ein Detail zum Bankenvertrag, öffnen Sie einen **Beleg** der Finanzbuchhaltung und anschliessend mit Doppelklick in der Wahlliste ein Formular für **Debitorenbuchungen**.



Vertrag verwendet in der Finanzbuchhaltung
(mit Wert für Umgebung auf „F"/Finanzbuchhaltung gesetzt)

c. Bei Erhalt der **Zahlung** wird die Debitoren-Position (CHF +7’532) über ein anderes leeres Debitoren- oder Finanzbuchhaltungsformular **gezogen** und damit ein Eintrag mit dem **Gegenbetrag** von CHF -7’532 **ausgelöst**. Beachten Sie, dass die **neue Buchung** in **Kursivschrift** erscheint, um Ihnen die **Verbindung** mit der **Debitoren-Position** anzuzeigen. Die verbundene Buchung kann mit der **„Detail“-Taste** in einem **eigenen Formular** angezeigt werden.

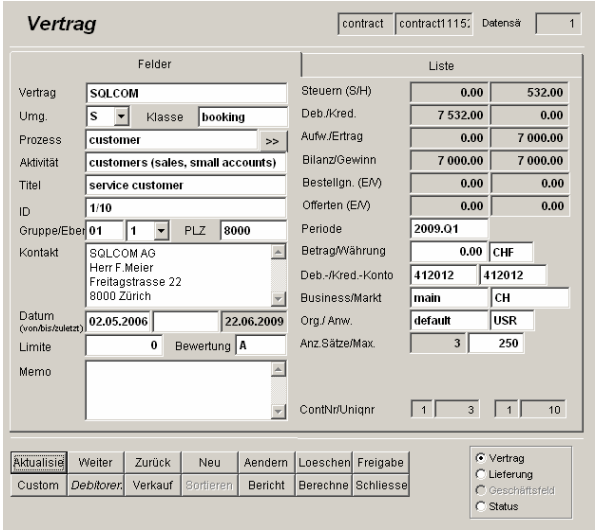


Ziehen der Debitorenbuchung über eine Finanzbuchung zur Eingabe der Zahlung

d. Um die **Verbindung** wieder **aufzuheben**, können Sie die Taste **"trennen"**/"unlink" verwenden. Getrennte Datensätze erscheinen dann wiederum in **Normalschrift**. Um von einem verbundenen Datensatz zum anderen zu kommen, verwenden Sie einfach die Detail-Taste beziehungsweise Alt-F3.

e. Der **neue Beleg** (mit der Gegenbuchung zur Debitoren-Position) wird jetzt durch Verbuchen des **Zahlungseingangs** bei der Bank (CHF +7'532) auf die **Summe von null** gesetzt und **abgeschlossen**. Mit der Taste **"Zahlung"**/"Payment" oder der Tastatur-Kombination **ctrl-F10** kann man den **Vertragswert** für die Zahlung vom **Beleg** holen und eine **Vorlage** mit dem **richtigen Betragswert** erstellen.

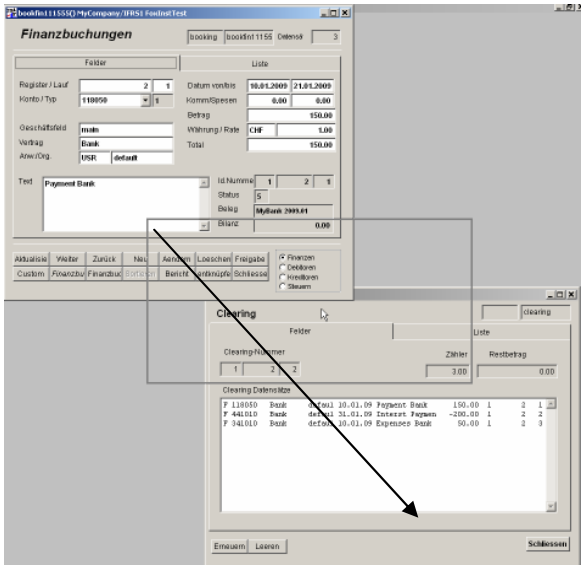
f. Sehen Sie im **Vertrag** nach, wie die Verkaufsbuchungen und der Abschluss der Belege die **Totale** in den Berichtsfeldern beeinflusst haben. Vor dem **“abschicken”**/”commit” wird nur das **Bestelltotal** CHF 7'000.—anzeigt. Nach dem **“abschicken”**/**“commit”** zeigen die Felder für den **Gewinn** CHF 7’000.--, für die **Steuern** CHF 532.--, und für die **Debitoren** CHF 7’532.--.



Vertragsformular mit Statistikdaten aus dem Verkaufsbeispiel (Vollversion)

g. Beim Abschluss des Belegs wird überprüft, ob die **Summe** aller verbunden Buchungen **null** beträgt. Enthält der Beleg verbundene Datensätze, deren Verbindung nicht null beträgt, wird eine **Warnung** ausgegeben. Falls nötig, kann die Verbindung mit der Taste **"trennen"**/"unlink" wieder aufgehoben werden.

h. Das **Werkzeug "Bereinigen"/"Clearing"** (im Menü unter "Ansicht") unterstützt Sie im Handhaben **verbundener Buchungen**. Sobald eine verbundene Buchung über das Werkzeug gezogen wird, erscheint eine **Liste aller verbunden Buchungen**. Diese können anhand ihrer Server/Uniqnr-Identifikation leicht im entsprechenden Beleg gefunden und geändert werden.



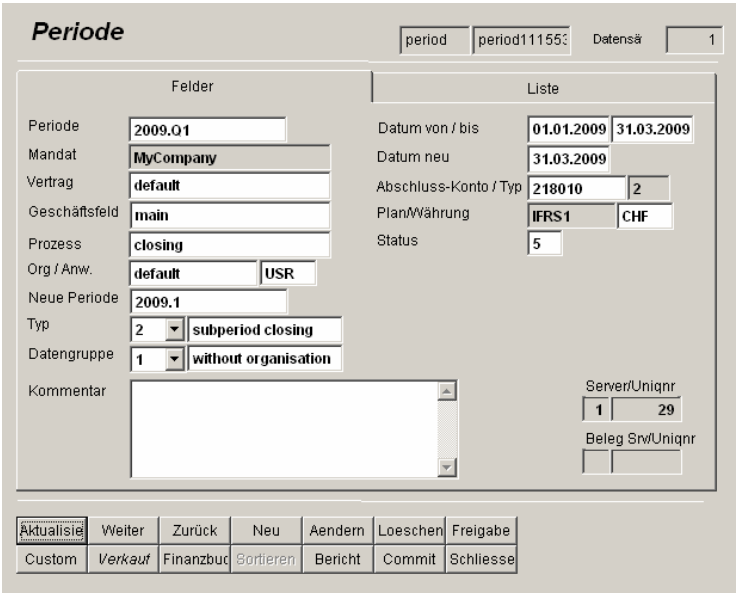
Ziehen der Buchung über das Werkzeug "Bereinigen"/"Clearing" liefert eine Liste der verbundenen Datensätze

i. Ebenso kann das **Werkzeug „Clearing"** über ein **Buchungsfenster** gezogen werden, um eine Verbindung zu markieren. Die eingesetzte Verbindung muss noch mit der Taste **"Ändern"**/"Update" bestätigt werden.

j. Zieht man einen Beleg der **alten** Periode über ein leeres **Belegformular**, erscheint ein interaktives Formular, das diese Funktion für den **aktuellen Beleg** oder eine ganze **Liste von Belegen** ausführt. Der Anwender kann dabei entscheiden, ob er nur **Belege**, oder **Belege und Buchungen** kopieren will, oder ob er eine **Abschlussbuchung** im **alten** Beleg erstellen und einen **neuen Beleg** mit einer **Eröffnungsbuchung** generieren will (siehe dazu ausführlich im Kapitel IV/4 über „Drag and Drop").

k. Die **Periode** (zum Beispiel 2009.1) kann jetzt ebenfalls mit der Taste **"abschliessen"**/"commit" abgeschlossen werden. Verwenden Sie dazu das **Periodenformular**, das am einfachsten mit einem **Doppelklick** auf dem **Periodenfeld** des Belegs aufgerufen wird.

l. Beachten Sie, dass Perioden einen **Zwischenabschluss** oder einen **Jahresabschluss** enthalten können. Beim **Zwischenabschluss** werden zum Beispiel die **Totale** der **monatlichen Perioden 01 – 12** in die Jahresperiode übertragen, beim **Jahresabschluss** dagegen die Totale des **alten Jahres** auf das **neue Jahr** übertragen. **Zwischenabschlüsse** enthalten **Totale für alle vier Kontotypen** (Aktiven, Passiven, Aufwand und Ertrag), **Jahresabschlüsse** dagegen nur Totale für **Aktiven und Passiven** (und einen Eintrag für das **Gewinn- beziehungsweise Verlusttotal**).



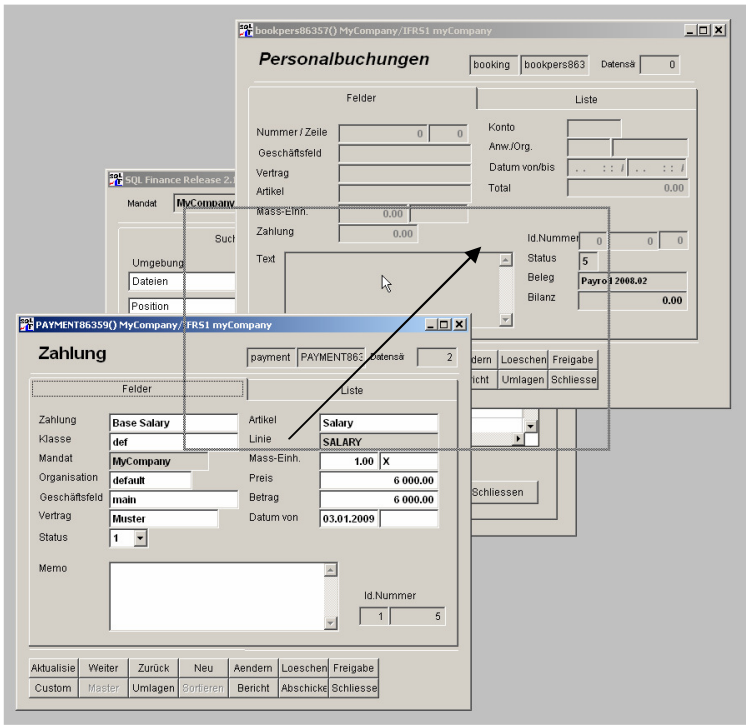
Zwischenperiode (Quartal) 2009.Q1
mit Abschluss zum Jahr 2009.1

4. Kurze Einführung in die Lohnabrechnung

Das Gehaltsmodul von **sqlFinance** verwendet in der Vollversion die Tabelle **"Vertrag"**/"Contract", um **Mitarbeiter und deren Anstellungsverträge** zu modellieren. So kann bei Bedarf ein Mitarbeiter auch in mehr als einem Vertrag angestellt sein. Monatliche und jährliche **Gehaltszahlungen** werden mit dem Vertrag verbunden. Zahlungen sind **Klassen** zugeordnet, und in Abhängigkeit von der Klasse werden **Umlagen** auf die Zahlungen automatisch berechnet. Umlagen sind typischerweise Kosten für die **Sozialversicherungen**, die teilweise von den Lohnzahlungen abgezogen werden.

Um eine Lohnabrechnung durchzuführen, werden die Zahlungen über ein **leeres Formular von Gehaltsbuchungen** gezogen. Nach einem Klick auf die Taste **„Umlagen"**/"Transfer" (im Buchungsfenster) berechnet das System die Umlagen und erstellt entsprechende Buchungen. Vom **Beleg** aus, der zu den Gehaltsbuchungen gehört, können **Berichte** mit Totalen und **Lohnabrechnungen** erstellt werden.

Ist der Lauf fehlerhaft, kann der Beleg **gelöscht** werden. Das System wird dann auch **alle dazugehörigen Buchungen** löschen. Der **Beleg** lässt sich anschliessend **zurückrufen**, und der Lauf kann von einem leeren Buchungsformular aus **nochmals** durchgeführt werden. Am Ende wird der Beleg **abgeschickt** und die zum Beleg gehörende **Zwischenperiode** abgeschlossen. Erst dann tauchen die Totale des Zwischenabschlusses (je nach Wunsch aufgeschlüsselt nach Kostenstellen oder nicht) in der **Hauptperiode** auf.



Durch Ziehen der Gehaltszahlungen über die Personalbuchungen (rechte Maustaste, Drag & Drop) wird die Lohnabrechnung gestartet

a. Grundsätzliches Vorgehen in der Lohnabrechnung:

- Zahlungen im **Zahlungsformular** eingeben.
- Eine **Auswahl** von auszuführenden **Zahlungen** selektionieren (zum Beispiel Status = "1").
- Einen **Personalbeleg** öffnen (normalerweise ein neuer Beleg, der noch keine Buchungen enthält). Möglichst eine neue **Teilperiode** im Beleg verwenden (zum Beispiel 2009.01).
- **Zahlungen** über das leere **Buchungsformular** ziehen.
- Personalbuchungen vom Beleg aus neu **selektionieren**.
- Durch Klick auf die Taste **"Umlagen"**/"Transfer" im Buchungsfenster Umlagebuchungen automatisch generieren.
- Bei Browse-Fenstern mit **'yes'** oder der Eingabe von **ctrl-w** antworten. Diese Fenster erlauben die Prüfung des Datenflusses.
- Nochmals Buchungen **neu selektionieren**, um Zahlungen und Umlagen zu **überprüfen**.
- Zurückkehren zum **Personalbeleg** und **Gehaltsabrechnung drucken** (Start des Berichtswerkzeugs über die Taste **"Bericht"**/"Report").
- Bei einem **Fehler** den **Beleg löschen**, das System wird alle dazugehörigen Buchungen ebenfalls löschen, verbesserte Prozedur wiederholen.
- Bei erfolgreichem Abschluss Beleg **abschliessen**.
- **Teilperiode abschliessen** (überträgt Totale in die Hauptperiode).

b. Perioden-Management in der Lohnabrechnung

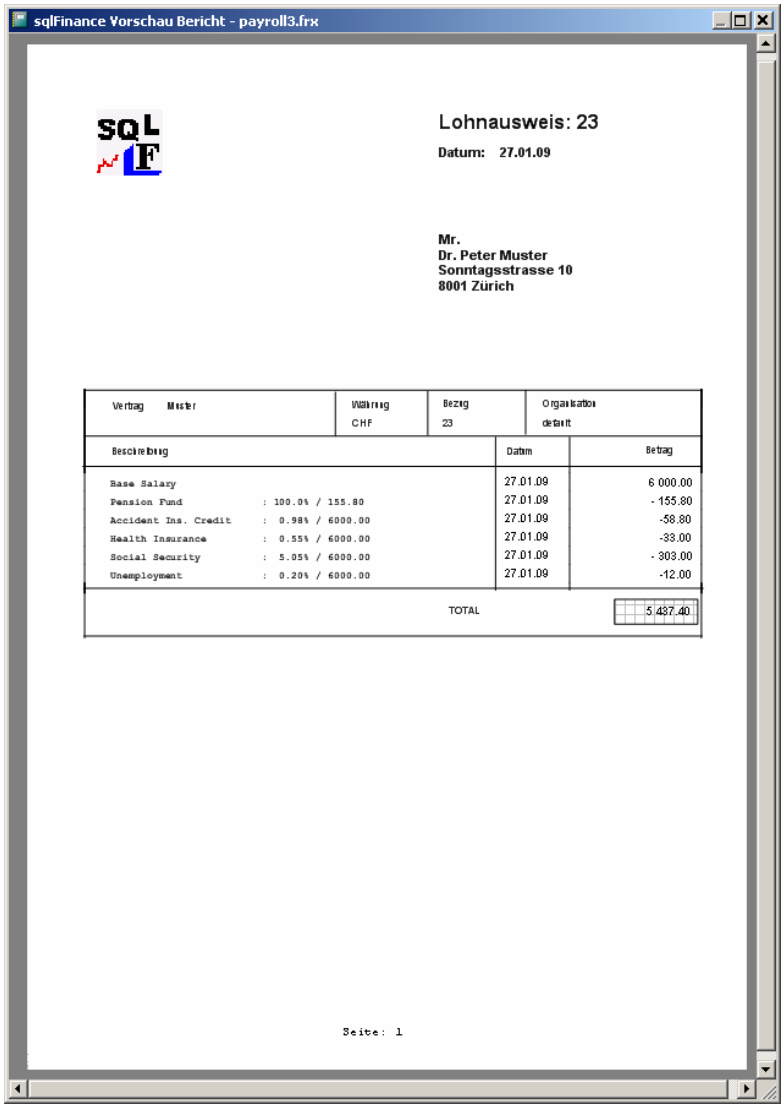
- Eine monatliche **Lohnabrechnung** wird meistens in einer **Zwischenperiode** abgelegt (zum Beispiel Zwischenperiode 2009.01, mit Abschluss in die Jahresperiode 2009.1).
- Nach dem Abschluss des **Belegs** muss auch die **Zwischenperiode** abgeschlossen werden. Der Zwischenabschluss produziert **Buchungen mit Totalen** in der **Jahresperiode**. Die Buchungen können **Totale pro Kostenstelle** oder nur **Gesamttotale** enthalten (siehe entsprechende Vorgaben in der Teilperiode).
- Die **Zwischenperioden** der Lohnabrechnung sollten möglichst nicht auch für andere Buchungen verwendet werden. Mehrere Zwischenperioden pro Monat sind aber möglich (zum Beispiel 2009.01a).

c. Druck der Lohnabrechnung

- Der Druck der **monatlichen Gehaltsabrechnung** erfolgt vom Personalbeleg aus (verbunden mit der monatlichen Teilperiode).

- Drucken Sie den **jährlichen Gehaltsbericht** von der Jahresperiode aus und verwenden Sie den **Bericht "Payroll Period"**. Stellen Sie sicher, dass dieser Bericht von der richtigen Periode aus gestartet wird, normalerweise der Jahresperiode, in die alle Abschlussbuchungen der Zwischenperioden fliessen.

- Ein spezieller Bericht unterstützt die Erstellung von Lohnabrechnungen am Jahresende zuhanden der Behörden (neuer Lohnausweis 2009).



Beispiel einer Gehaltsabrechnung in der Berichtsanzeige

Datenstruktur der Lohnabrechnung

- Verwenden Sie das Feld **"Vertrag"**/"Contract" für die Handhabung der **Mitarbeiterverträge**. Auch diese Verträge können weiter zu Geschäftsfeldern („business") zusammengefasst und in Organisationseinheiten oder Profitzentren („organisation") gruppiert werden.

- Einträge in der **Tabelle "Zahlungen"**/"Payments" verlangen Angaben zu **Produkten** und **Produktelinien** (wie zum Beispiel das Monatsgehalt: Produkt „monthlySale" aus der Produktelinie „salary"). In der Tabelle **„Linie"** können Debitoren- und Kreditoren-Konten angegeben oder geändert werden. Diese Werte steuern anschliessend die **Kontierung** der Gehaltsbuchungen. Verwenden Sie denselben Ansatz für die Tabelle „Umlagen/transfers", die ebenfalls Angaben zum Produkt verlangt. In den Umlagen kann auch ein Wert für das **Geschäftsfeld** („Geschäft"/„business") eingesetzt werden. Die **Verbindung** zwischen **Zahlungen und Umlagen** erfolgt aber in der Lohnabrechnung **nur** aufgrund des Werts für das Feld **„Produktelinie"**

- Zahlungen und Umlagen sind über das Feld **"Klasse"**/"Class" miteinander verbunden.

- **Bildung von drei Kontotypen**:

  - Lohnaufwand (Aufwand)
  - Lohnabrechnung Mitarbeiter (Passiven)
  - Abrechnung Versicherungen (Passiven)

  Lohnzahlungen werden verbucht:
  + Lohnaufwand, - Lohnabrechnung

  Umlagen Mitarbeiter werden verbucht:
  + Lohnabrechnung, - Abrechnung Vers.

  Umlagen Unternehmen werden verbucht:
  + Lohnaufwand, - Abrechnung Vers.

  Die Umlagen Mitarbeiter enthalten den vom **Mitarbeiter** getragenen Anteil einer Versicherungsleistung, die Umlagen Untenehmen den vom **Unternehmen** übernommenen Restanteil.

  Beachten Sie, dass die **Bruttolöhne** in der Lohnabrechnung (ein Passivkonto) als negative, die dem Mitarbeiter belasteten **Abzüge** entsprechend als positive Beträge erscheinen. Die **Druckprogramme** für den Lohnausweis ändern die Vorzeichen, in den **Buchungsauszügen** werden die Beträge jedoch unverändert aufgeführt.

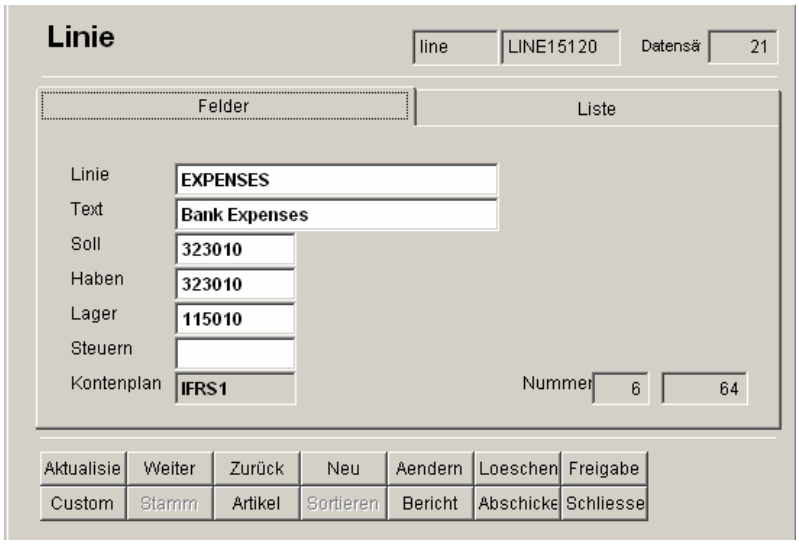
- Die **Kontierung** von Produkten für Zahlungen und Umlagen in den Feldern 'Debit', 'Credit' und 'Lager' der Tabelle **'Linie'** ist wie folgt auszuführen:

  a. 'Debit': Lohnaufwand Unternehmen
  b. 'Credit': Lohnabrechnung Mitarbeiter
  c. 'Lager': Abrechnung Versicherungen

  Beispiel :

  3/322010: Lohnaufwand Unternehmen
  2/231010: Lohnabrechnung Mitarbeiter (Passiven)
  2/211050: Abrechnung Versicherungen (Passiven)

- Datensätze der Tabelle **"Linie"**/"Line" akzeptieren auch **leere Werte** in den Kontenfeldern. In diesem Fall wird **keine Umlage** verrechnet. **Debit zu Lager** belastet die Sozialleistung dem Unternehmen, **Credit zu Lager** dagegen dem Mitarbeiter.



Beispiel für die Kontierung von AHV Leistungen in einer Schweizer Lohnabrechnung

d. Beispiel für die Lohnabrechnung mit IFRS Kontierung (aus der Datenbank "myCompany")

Debit Konten:

322010	Gehälter
322050	Sozialversicherung
322060	Krankheit
322070	Unfall
322080	Pensionskasse
322090	andere Personalversicherungen

Credit Konten:

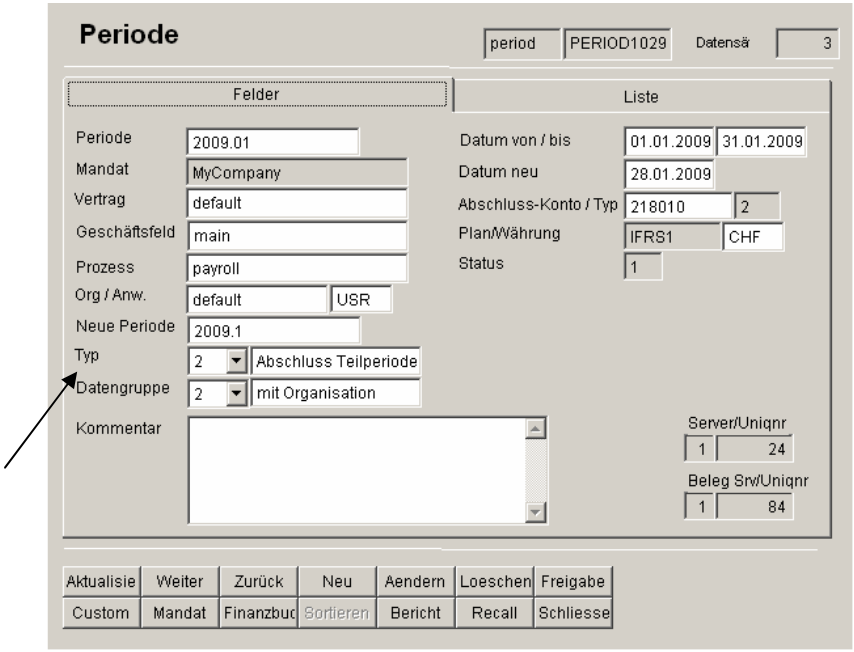
231010	Lohnabrechnung Mitarbeiter (Kontokorrent)

Lager Konten:

211050	Sozialversicherung
211060	Krankheit
211070	Unfall
211080	Pensionskasse
211090	andere Personalversicherungen

Vergleichen Sie die Beträge in den **Lagerkonten** mit den **effektiven Zahlungen** zugunsten der Sozialversicherungen, die Sie zum Beispiel in Konten mit Nummern 211050 - 211090 führen. So können Sie recht gut erkennen, mit welchen Forderungen in diesem Bereich noch zu rechnen ist und ob ihr **Bericht an die Versicherungen** und deren **Rechnungsstellung** übereinstimmen.

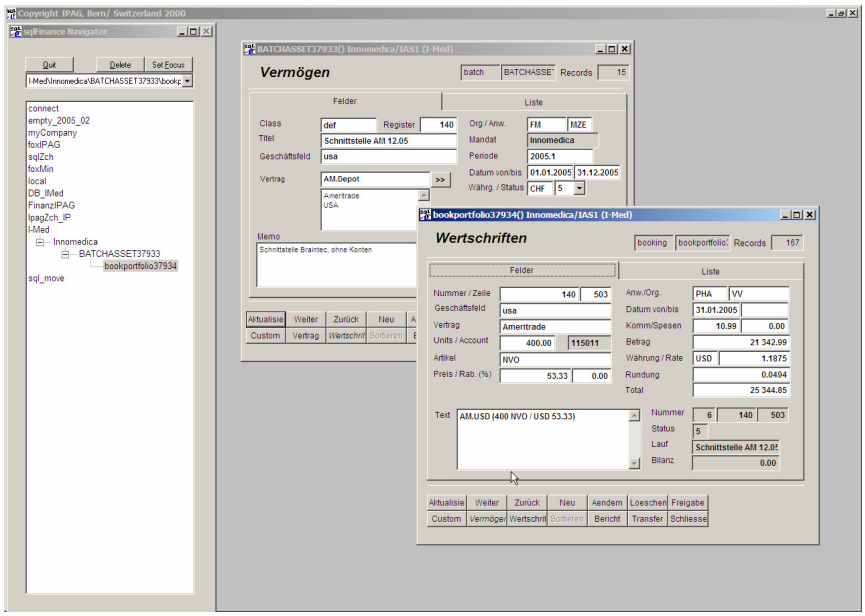
Das Konto für die **Lohnabrechnung** der Mitarbeiter enthält alle Buchungen, die den Mitarbeiter direkt betreffen. Mit einem **monatlichen Auszug** dieses Kontos erhält man folglich die **Lohnabrechnung pro Mitarbeiter**, wie sie für den Lohnausweis verwendet wird. Nach **Auszahlung** der Löhne (-Bank, +Lohnabrechnung Mitarbeiter) erreicht das **Kontokorrent** für die Lohnabrechnung der Mitarbeiter wieder den Wert **null**. Die Lohnzahlung kann als **Total** für alle Mitarbeiter erfasst werden. Es ist aber auch möglich, im Kontokorrent einen Eintrag für **jede Lohnzahlung** pro Mitarbeiter zu erfassen, und den Beleg dann mit dem Total der Bankzahlung wieder auf null abzuschliessen.



Beispiel für eine offene Teilperiode aus der Lohnbuchhaltung (mit Abschluss nach Kostenstellen/Organisation)

5. Vermögensverwaltung

sqlFinance beinhaltet in der **Vollversion** ein **Modul** für die **Vermögensverwaltung**. Das Modul enthält eine Schnittstelle für den **Import** von Daten aus dem Handel. Wie alle anderen Module werden auch hier **Fremdwährungen** unterstützt. Felder sind vorgesehen für Kommissionen und Spesen der Banken, Steuern und andere Spesen. Von der **Periode** ausgehend können Berichte für die **Bereinigung** der Fremdwährungskonten in der Hauptwährung erzeugt werden. Die **Berichte** machen es auch möglich, die **Performance** einzelner Titel nachzuvollziehen und **Titelgewinne** gegenüber **Währungsgewinnen** abzugrenzen.



Kaskade in der Vermögensverwaltung mit Beleg und Buchungen (mit Beispieldaten des Mandats „Innomedica")

Für die **Vermögensverwaltung** liegen **Beispieldaten** über drei Jahre vor. Diese Daten wurden von der Finanzgesellschaft **Innomedica Holding** freundlicherweise für Ausbildungs- und Marketing-Zwecke zur Verfügung gestellt und sind in der Gratis- und Vollversion **enthalten**. Die Daten zeigen die Verbuchung von:

- Wertschriftenhandel mit kotierten Aktien (Gesundheitssektor)
- Handel mit eigenen Aktien
- Ausgabe Wandelanleihe
- Abrechnung von Aufwänden für die Vermögensverwaltung
- Abgrenzung von Währungsgewinnen und Kapitalgewinnen aus dem Aktienhandel

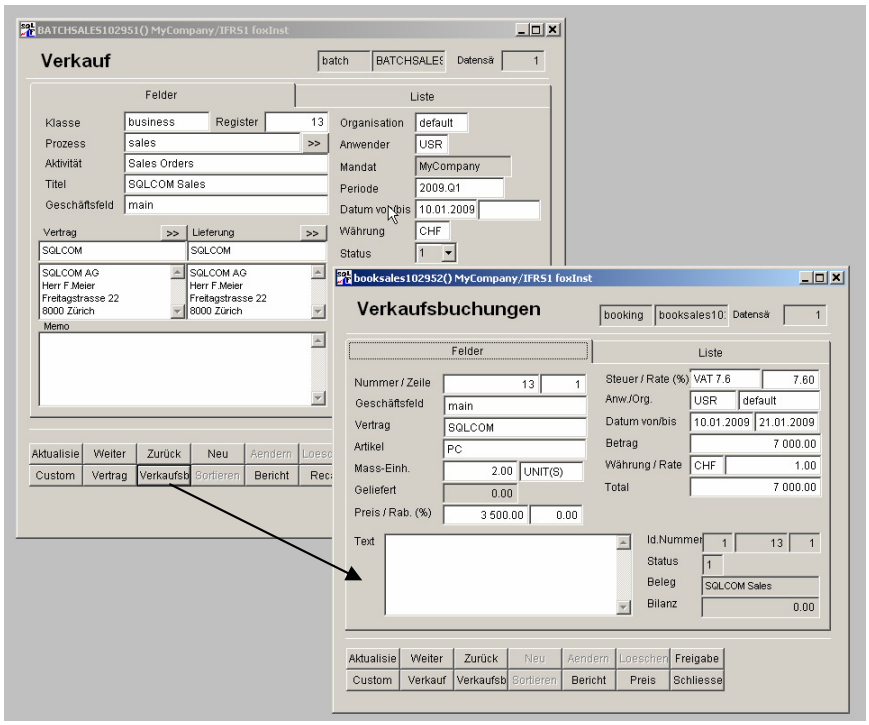
Die Beispieldaten sind auch gut geeignet, um **Buchungsbeispiele** zu studieren und sich mit den **Druckprogrammen** vertraut zu machen. Die Beispieldaten finden Sie, indem Sie die Verbindung „Innomedica" im Verbindungswerkzeug aktivieren und anschliessend das Mandat „Innomedica" auswählen. Sehen Sie sich dann die drei **Perioden** 2003.1, 2004.1 und 2005.1 an. Vom Periodenfenster kann zum Beispiel der Abschluss über das **Druckprogramm** „Totale (Bilanz)"/"Totals (Balance)" ausgedruckt werden.

IV. Schlüsselfunktionen

1. Formulare und Kaskaden

sqlFinance setzt **standardisierte Formulare** ein für die Anzeige von Selektionsdaten. Alle Formulare haben einen **Seitenrahmen** mit zwei Seiten. Auf der ersten Seite werden die einzelnen **Felder** des aktiven Datensatzes angezeigt, auf der zweiten erscheint eine **Liste** der ausgewählten Daten in Tabellenform. Jedes Formular enthält auch den gleichen Satz **Tasten zum Ändern der Daten** und die Aktivierung anderer Formulare.

Eine Anzahl verbundener Formulare ist in einer **Kaskade** organisiert. Die Formulare sind normalerweise über eine **Master-Detail-Beziehung** miteinander verbunden. Die Suche von Master zu Detail kann über **mehrere Stufen** weitergeführt werden und ist nur durch die logischen Aspekte des Datenmodells oder die Ressourcen Ihres Systems beschränkt. Master-Detail-Beziehungen folgen der **Hauptstruktur des Datenmodells** (zum Beispiel Adresse – Kontakt – Vertrag – Beleg – Buchung). Studieren Sie deshalb das **Datenmodell**, um ein gutes Verständnis der **Master-Detail-Beziehungen** zu erreichen.



Kaskade Vertrag-Verkaufsbeleg (Klick auf die Taste „Verkauf" im Verkaufsbeleg bringt neues Fenster mit Verkaufsbuchungen)

In der Kaskade ist nur das **aktuelle Formular sichtbar**, die übrigen bleiben **versteckt**. Durch Drücken der Taste "**Freigabe**"/"Free" können die Formulare von der Kaskade **losgelöst** werden und wiederum den Ausgangspunkt für **neue Kaskaden** bilden. Dies kann in vielen Situation praktisch sein, zum Beispiel wenn der Anwender einen Datensatz, den er über einen Doppelklick gefunden hat, für später in Erinnerung behalten möchte.

a. Tastatur-Kombinationen

Für die meisten Tasten gibt es eine Tastaturkom**bina**tion. Zum Beispiel kann durch die Eingabe von **ctrl-s** die **Suche** von Daten im Selektionswerkzeug oder im Report gestartet werden. **Ctrl-a** („add") fügt im Datenformular einen **neuen Datensatz** ein, **ctrl-u** bestätigt die **Änderungen**. Eine **Liste** der Tastatur-Kombinationen findet sich am Ende des Manuals. Die Tastatur-Kombination kann aber auch direkt durch **Klicken** mit der **rechten Maustaste** auf die **Taste des Formulars** angezeigt werden.

b. Standardtasten auf dem Formular:

Obere Reihe:

- Aktualisieren (requery, ctrl-q): selektioniert die aktuellen Daten erneut von der Festplatte
- Weiter (next, ctrl-n): zum nächsten Datensatz
- Zurück (previous, ctrl-p): zum vorherigen Datensatz
- Neu (add, ctrl-a): einen neuen Datensatz generieren
- Ändern (update, ctrl-u): Änderungen auf die Festplatte speichern
- Löschen (delete, ctrl-d): aktuellen Datensatz löschen
- Freigabe (free, ctrl-k): Verbindung der Formulare trennen

Untere Reihe:

- Einrichten (custom): Vorgabewerte ändern in der Meta-Datenbank
- Master (ctrl-F1): Selektion von Master-Daten (bezogen auf aktuellen Datensatz)
- Detail (ctrl-F2): Selektion von Detail-Daten (bezogen auf aktuellen Datensatz)
- Sortieren (order): die Reihenfolge der Datenliste ändern
- Bericht (report, ctrl-p): Zugang zum Berichtswerkzeug (mit Bezug auf den aktuellen Datensatz des Formulars)
- Bestätigen (commit, ctrl-F10): Starten einer Formular-spezifischen Aktion zum Abschliessen eines Belegs, einer Periode et cetera
- Schliessen (drop, Taste ESC): Formular schliessen, Daten im Zwischenspeicher entfernen

c. Die **Master-** und **Detailtasten** suchen **neue Daten** unter Bezugnahme auf den **aktuellen Datensatz**. Ein Detail zu **"Adresse"**/"Adress" ist zum Beispiel der **"Kontakt"**/"Contact". Das neue Kontaktformular zeigt dann Kontaktdaten, die zur aktuellen Adresse passen.

d. Gewisse Tasten ändern ihre Aktion je nach Formular:

- Beleg und Periode: Bestätigen (commit) schliesst den Beleg oder die Periode (oder öffnet sie wieder)
- Personalbuchungen: Erstellen von Umlagebuchungen für die erfassten Zahlungen, gemäss den Vorgaben der Tabelle "Umlage"/"transfer"
- Kontaktformular: Über den Zwischenspeicher können Daten von TwixTel (mit ctrl-c in den Zwischenspeicher von Windows kopieren) oder anderen Adressverzeichnissen geholt und direkt in die einzelnen Felder eingetragen werden; eine praktische Funktion für das Kopieren von privaten oder Unternehmensadressen

e. Ist bereits ein Master- oder Detailformular geöffnet worden, erscheinen die entsprechenden Tasten mit **Kursivschrift** (italic). Ein **Klick** auf die Taste **aktiviert** das zugehörige Formular in der **Kaskade**.

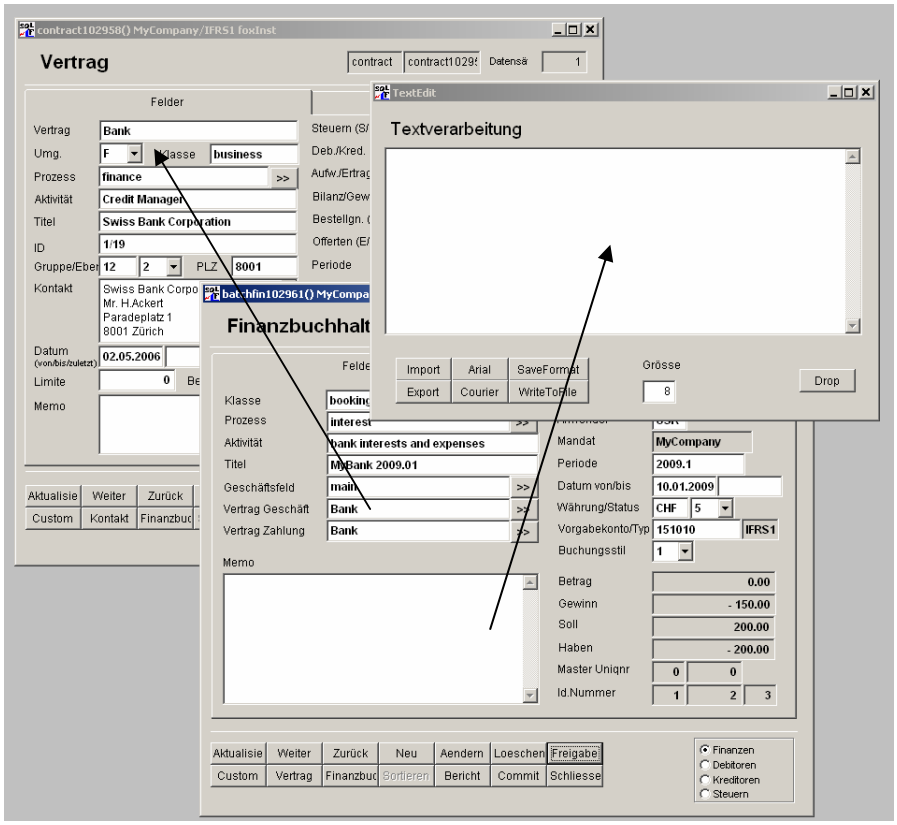
f. Mit Hilfe von **Master- und Detail-Tasten** ist es möglich, **schnell von einem Formular zu einem anderen** zu gelangen. Anstatt auf die Master- und Detail-Tasten zu klicken, kann man auch die Tastatur-Kombinationen **ctrl-F2** (master) und **ctrl-F3** (detail) verwenden.

2. Felder

Felder erscheinen hauptsächlich auf der **ersten Seite** des Seitenrahmens im Formular. Felder gehören zu **Feldtypen**, die bestimmte Werte darstellen wie Zahlen, Buchstaben, Datum et cetera. Felder haben ihrem Typ entsprechend ein **standardisiertes Verhalten** und reagieren gleichartig auf Aktionen des Anwenders wie zum Beispiel **Doppelklicks** oder **Ziehen mit der Maus**.

a. Textfelder

Textfelder haben oft einen **Bezug** zu einer **übergeordneten Tabelle**. Ein **Vertragsname** zum Beispiel, eingegeben im **Vertragsfeld** eines Buchungsformulars, bezieht sich auf einen entsprechenden Eintrag in der Parameter-Tabelle **"Vertrag"**/"Contract". Mit einem **Doppelklick** auf dem **Vertragsfeld** (des Buchungsformulars) kommt der Anwender direkt zum richtigen Datensatz in der **Vertragstabelle**. Enthält der Feldeintrag so genannte „**Wildcards**" (zum Beispiel * oder %), zeigt die neue Liste eine **Auswahl** von Datensätzen gemäss **Suchkriterium**. Die Eingabe von **„*myCompany*"** oder **„%myCompany%"** wird zum Beispiel alle Datensätze auswählen, die einen Vertragsnamen haben, der die **Zeichenkette** „myCompany" **enthält** (zum Beispiel „myCompany.001", et cetera). Nach **Verlassen** des Parameterformulars wird der **Wert** des ausgewählten Datensatzes in das **Feld** übertragen, von dem der **Doppelklick** ausgeführt worden war.



Vertrag "Bank" (aktiviert mit Doppelklick im Feld „Vertrag Geschäft" des Belegs) und Textverarbeitung, aktiviert mit Doppelklick auf das Memofeld des Belegs)

b. Numerische Felder

- Einfache Zahlen gehören zum Typ **"integer"**, Dezimalzahlen zum Typ **"Geld"**/"money". Nur Felder vom Typ **„Geld"** (zum Beispiel Betrag und Total bei der Buchung) verfügen über besondere Möglichkeiten.

- Ein **Doppelklick** auf Felder des Typs **"Geld"** wechselt das **Vorzeichen** des Betrags (von einem negativen auf einen positiven Wert oder umgekehrt).

- Viele Felder vom Typ **„Geld"** können mit der **Maus** über ein **anderes Feld** desselben Typs gezogen werden. Verwenden Sie die **rechte Maustaste** beim Ziehen (die linke Maustaste ist stark belegt durch Standardfunktionen des Microsoft-Betriebssystems) und bringen Sie das Feld über ein anderes. Der **Wert** wird übertragen, sobald Sie die Maustaste **loslassen**. Im Buchungsformular kann dies zu **weiteren Wertänderungen** führen (ändern Sie zum Beispiel den Preis oder den Währungskurs, ändert sich auch das Total). Am häufigsten wird das **offene Total** eines Belegs über das Total einer **abschliessenden Buchung** gezogen, um so den Beleg auf ein **Gesamttotal** über alle Buchungen von **null** zu bringen.

  Geldfelder im Buchungsformular sind so eingerichtet, dass die Kalkulation **konsistent** bleibt (Anzahl * Preis = Betrag, Betrag * Währungskurs = Total). Es ist besser, einen Wert über das Preis- oder Betragsfeld als über das Totalfeld zu ziehen, da Änderungen im Total durch Neu-Berechnung des Währungskurses (und nicht des Preises) kompensiert werden. Die Anpassung des Währungskurses kommt vor allem zum Einsatz in der Vermögensverwaltung oder beim Abschluss von Fremdwährungskonten.

c. Datumfelder

- Datumfelder enthalten immer Werte für **Datum** und **Zeit**. Ist die Erfassung der Zeit nicht wichtig, kann der Wert weggelassen werden.

- **Doppellklicks** auf dem Datumfeld bringen beim ersten Mal ein leeres Feld, beim zweiten Mal das **aktuelle Datum**.

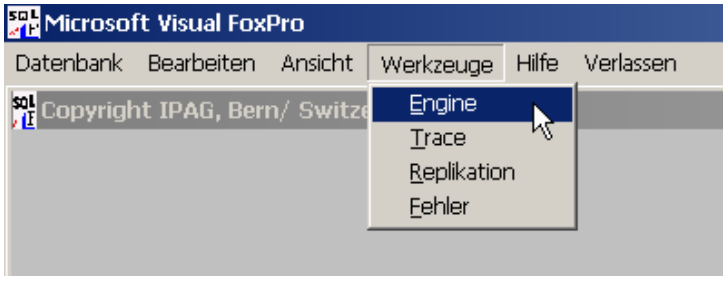
d. Memofelder

- Memofelder werden zum Abspeichern von **mehrzeiligen Textelementen** verwendet. Eine **Kundenadresse** in einem Brief ist ein Beispiel für ein mehrzeiliges Textelement. In Memofeldern gespeicherter Text kann bis zu **mehreren Seiten lang** sein, begrenzt nur durch die Möglichkeiten des Computersystems. Die einzelnen **Linien** im Memofeld (bis zu einem **Zeilenumbruch**) dürfen aber nicht länger als 8'192 Zeichen sein).

- **Doppelklicks** auf dem Memofeld aktivieren in einem ersten Schritt den **sqlFinance Texteditor**. Ein zweiter Klick auf dem Textfeld des Editors aktiviert die Shareware Textverarbeitung **„Notepad"**. Voraussetzung dazu ist allerdings, dass Sie eine Version des Notepads **herunterladen** und im sqlFinance **Stammverzeichnis** (zum Beispiel c:\programme\sqlFinance) **installiert** haben. Der **sqlFinance Editor** enthält nur ein einfaches Textfenster, das vergrössert werden kann und bei dem die **Schrift** ausgewählt und grösser oder kleiner dargestellt werden kann.

3. Menüs

sqlFinance verwendet Menüs nur in geringem Ausmass. Nach dem Start der Anwendung lässt sich das **Hauptmenü** am **oberen Bildschirmrand** finden. Nur bei Aktivierung des **Berichtwerkzeugs** wird das Menu zu einem **Systemmenü** geändert. Das Menü wird vor allem für den Start von **Werkzeugen** verwendet. Wenn Sie ein Formular **sichtbar** machen wollen, ist es meistens einfacher, die **Baumstruktur** (treeview) **links auf dem Bildschirm** zu verwenden. Nachfolgend eine Übersicht über die Möglichkeiten der Menüs:



sqlFinance Hauptmenü / main menu

Datenbank

a. **Verbindung**
Eine Verbindung mit einer **physischen Datenbank** auf der Arbeitsstation oder dem Server aufbauen. In einer Arbeitssitzung können mehrere Verbindungen gleichzeitig geöffnet werden. Mit dem Verbindungswerkzeug oder der Baumstruktur können Sie die gewünschte Verbindung **aktivieren**.

b. **Mandat**
Einmal mit der physischen Datenbank verbunden, muss ein Mandat **ausgewählt** werden. Das Feld **"Listennummer"**/"list number" bestimmt die **Reihenfolge**, in der die Mandate aufgelistet werden und welches Mandat zuoberst erscheint. Ein Mandat stellt in der Regel ein einzelnes Unternehmen dar, oder die konsolidierten Zahlen einer Unternehmensgruppe. Um **Eigenschaften** des Mandats zu ändern, selektioniert man am besten das Mandat vom **Selektionswerkzeug** aus.

c. Selektion
Hauptwerkzeug für die Selektion von Daten aus der Anwendungsdatenbank. Oben auf dem Selektionswerkzeug sind die **aktive Verbindung**, das **aktive Mandat** und der **aktive Kontenplan** aufgeführt.

Bearbeiten

a. **Fenster**
Eine **Liste** der zurzeit **geöffneten Formulare**. Die Liste ist in **Tabellenform** dargestellt. Ein **Klick** auf eine Zeile **aktiviert** das entsprechende Formular. Ein praktisches Werkzeug bei der Arbeit mit **vielen offenen Formularen** oder eine gute Hilfe beim **Auffinden** von versteckten oder anderweitig verborgenen Formularen.

b. **Datensätze**
Datensätze sind ein **lokales Register** für Datensätze des Anwenders. Die

Liste wird in Form einer Tabelle dargestellt. Durch **Ziehen** einer Formularseite mit Feldern über die Tabelle können Sie einen **Eintrag** mit entsprechender Notiz von Referenznummer und Art des Datensatzes erstellen. In **späteren Arbeitssitzungen** können diese Datensätze durch Betätigen der **Selektionstaste** wieder abgerufen werden. Die Einträge in der Liste können nach **Prioritäten** gruppiert werden (1 bis x, jeweils der erste Eintrag der Liste dient auch als „Überschrift"). Jeder Datensatz kann auch mit einem **Titel** und einem **Kommentar** versehen werden. Auf Wunsch können diese Einträge **exportiert** und in einer anderen Installation wieder **importiert** werden.

c. **Notizen**
Notizen sind ein einfaches Werkzeug um kleinere Textelemente während der Arbeitssitzung **ablegen** zu können. Die Grösse des Notizwerkzeugs kann geändert werden. Im **Textfeld** des Notizen-Werkzeug können Daten aus einem **Adress-Verzeichnis** wie **TelSearch** oder **Twixtel** eingesetzt werden. Wird das Notizwerkzeug dann mit der **Maus** (rechte Maustaste) über ein **Kontaktfenster** gezogen, analysiert das System den Text und **füllt die einzelnen Felder des Kontaktformulars sinngemäss aus**.

d. **Optionen**
Eine Liste von **Parameter-Werten**. Einige können vom Anwender geändert werden. Die Parameter-Liste wird in einer Tabelle dargestellt. Zu einem Parameter gehören immer ein Name und ein Wert, wobei der Wert Text- oder Zahlen-Charakter haben kann.

- **ODBC_mgr**
  Eine Variable zum Abspeichern des Namens für den Microsoft ODBC Managers, normalerweise ODBCAD32X.EXE. Der Microsoft ODBC Manager kann auch vom Betriebssystem aus gestartet werden, indem man "ODBCAD32X.EXE" in das Befehlsfenster schreibt. Es gibt verschiedene Versionen für den ODBC Manager im Markt. Der Einsatz gewisser Versionen kann die Antworten des SQL Servers sehr langsam machen. Alle Lieferungen von sqlFinance enthalten eine Kopie des geeigneten ODBC Managers im Verzeichnis "drivers". ODBCAD32X.EXE wird normalerweise im "windows\system32" Verzeichnis installiert.

- **Datum-Check**
  Ein Systemwert, der die **Selektion von Datensätzen** beeinflusst. Ist der Wert für die Variable „Datum-Check" auf **1** gesetzt (Datum-Check auf „ein"), bringt der Selektionsprozess nur Daten mit einem **leeren „Datum-bis" Feld**. Ist der Wert auf **0** gesetzt (Datum-Check auf „aus"), dann werden auch Daten mit einem **nicht-leeren „Datum-bis" Feld** selektioniert. Während im „Datum-von" Feld meistens von Anfang an das aktuelle Datum erfasst wird, kann das "Datum-bis" Feld gut für diese Funktion verwendet werden (zum Beispiel bei der Buchungs-Tabelle). Setzen Sie den Datum-Check auf „ein" und selektionieren Sie Datensätze zur Bearbeitung. Erledigte Datensätze können dann durch einen Doppelklick auf dem Feld „Datum-bis" (mit Eintrag des aktuellen Datums, Änderung noch mit der Taste „Ändern" bestätigen) abgebucht werden. In einer erneuten Selektion werden sie in der Liste nicht mehr erscheinen (solange der Datum-Check auf „ein" gesetzt bleibt). Unabhängig vom Wert der Variable Datum-Check werden immer alle Datensätze selektioniert, wenn das Feld „Datum-bis" einen Suchwert enthält. Der Wert für die Variable Datum-Check kann im **Selektionswerkzeug** auf einfache Weise geändert werden.

- **Dezimale**
  Die Anzahl Dezimalstellen für Felder des Typs "Geld" (normalerweise 2).

- **Autoboot**
  Mit Autoboot auf "ein" (1=default) wird das System beim Starten die Anmeldeprozeduren automatisch durchlaufen, die letzte Verbindung und das letzte Mandat aktivieren und das Selektionswerkzeug anzeigen.

- **Mandat**
  Der Namen des zuletzt verwendeten Mandats.

- **Translate Check**
  Parameter zum Aktivieren der Übersetzung in eine Fremdsprache (Übersetzen von englischsprachigen Feldbezeichnungen und Meldungen).

- **Textedit_size**
  Vorgabewert für die Zeichengrösse im Text Editor.

Ansicht

a. **Tastenleiste**
Ein Werkzeug mit einer Anzahl Tasten, die Menupositionen entsprechen. Das Werkzeug kann mit dem Menu sichtbar oder unsichtbar gemacht werden. Eine ältere Darstellungsmöglichkeit, die einige Anwender bevorzugen.

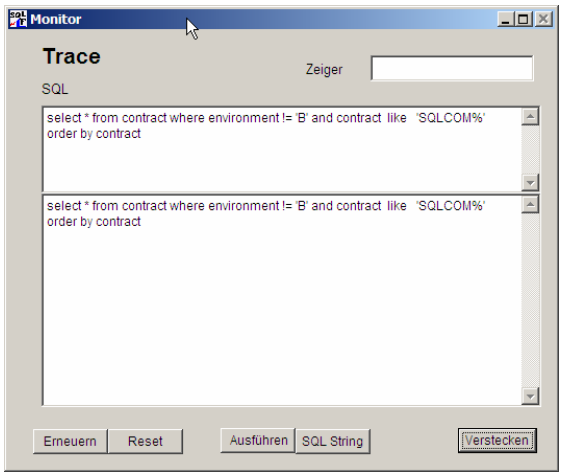
b. **Baumstruktur** (treeview)
Macht die Baumstruktur auf der **linken Bildschirmseite** sichtbar oder unsichtbar. Die Grösse der Baumstruktur kann mit der Maus geändert werden.

c. **Bereinigen** (clearing)
Ein Werkzeug um Verbindungen zwischen Buchungen zu überprüfen. Durch Ziehen der Seite „Felder" eines Buchungsformulars über das Werkzeug "Bereinigen"/"clearing" erhält man eine Liste aller verbundenen Buchungen. Mit Hilfe dieses Werkzeugs können Debitoren und Kreditoren verwaltet werden (s. Details im entsprechenden Kapitel über Debitoren- und Kreditorenwerkzeug). Um Buchungen miteinander zu verbinden, wird die Seite „Felder" eines Buchungsformulars über ein anderes Buchungsformular gezogen. Um die Verbindung wieder zu trennen kann die Taste "Trennen"/"unlink" des Buchungsformulars verwendet werden.

Werkzeuge

a. **Engine**
Ein Werkzeug für die Verwaltung des Datenbankzugangs und der Werte aktueller Variablen. Die „Engine" ist immer **aktiv** im **Hintergrund** und kann für die Lösung von Problemen sichtbar gemacht werden.

b. **Trace**
Ein Werkzeug für die Analyse und Prüfung von **SQL Statements**. Das zuletzt ausgeführte SQL Statement wird angezeigt, ebenso wie alle SQL Statements der letzten Transaktion (in einem Log-File). Das Werkzeug „Trace" kann ein leistungsfähiges Instrument sein für Anwender mit SQL-Kenntnissen. Benützen Sie zum Beispiel das Werkzeug „Trace" um herauszufinden, warum ein **Bericht** nicht die gewünschten Datensätze selektioniert.



Ansicht des Werkzeuges „Trace"

c. **Replikation**
Das Replikationswerkzeug **kopiert** Eingaben und Änderungen von Daten von einem Computer zu einem anderen. In einem ersten Schritt werden die Daten von der **Quelldatenbank** in die Datenbank **„empty_DB"** kopiert. In einem zweiten Schritt können die Daten über das Internet oder mit Hilfe einer abnehmbaren Festplatte **transportiert** werden. Im dritten Schritt werden die Daten von der Datenbank **„empty_DB"** in die **Zieldatenbank** übertragen. Die Datenbank „empty_DB" ist nur bei der Lieferung der Vollversion von sqlFinance enthalten.

d. **Fehler**
Werkzeug für die **Anzeige von Fehlermeldungen** (normalerweise von den gespeicherten Prozeduren des Servers kommend). Alle Meldungen des Servers werden im **Logfile** aufgeschrieben.

Hilfe

a. **Hilfe**
Manualseiten, gespeichert in der Tabelle **"info"** der Anwendungsdatenbank. Die Manualseiten können auch vom Selektionswerkzeug aus abgerufen werden, indem man die Stammtabelle "info" mit dem Suchwert **"tips"** für das Feld **"reference"** selektioniert.

b. **Information**
Copyright und Release Information. Dessen Inhalt gilt auch für dieses Manual:

> **sqlFinance** ist ein **registriertes Design**. Die Software ist **Copyright geschützt**. Obwohl ein Teil der Software gratis verteilt wird, dürfen Sie sqlFinance nicht weitergeben, verkaufen oder in ein Produkt integrieren **ohne schriftliche Einwilligung** des Besitzers. Die Software ist **exklusives Eigentum** von sqlFinance, Dr. Peter Halbherr. Mit dem Download werden Sie **nicht Eigentümer** des Produkts, sondern Sie erhalten eine **Lizenz**, die **Sie berechtigt** die Software **auf unbeschränkte Zeit zu verwenden**.

c. **Verlassen**
Anwendung verlassen.

4. Ziehen mit der Maus (drag and drop)

Um Daten von einem Formular zum anderen zu übertragen, verwenden Sie am besten das **Ziehen mit der Maus**. Sie klicken die **rechte** Maustaste an und ziehen das Formular über das Andere. Die Formulare sind nun miteinander verknüpft. In den meisten Fällen müssen die Formulare dabei **kompatibel** sein und dieselben Felder aufweisen. Das können zwei **gleiche** Formulare **aus zwei verschiedenen Datenbanken** sein, oder aber auch zwei Buchungsformulare mit Buchungen aus **zwei verschiedenen Perioden** (selektioniert als Detail zu zwei verschiedenen Belegen). Auf diese Art können Daten allenfalls auch von einem entfernten Computer oder Server **importiert** werden. Um ganze Datengruppen zu übertragen empfiehlt sich aber eher der Einsatz des **Replikationswerkzeugs**.

Durch Ziehen der **ersten Seite** (mit Feldern) des Seitenrahmens im Formular über das andere Formular wird die **Eingabe** eines **neuen Datensatzes** mit den übertragenen Werten ausgelöst. Danach werden Sie **gefragt**, ob Sie die Eingabe **bestätigen** möchten (mit einem Klick auf die Taste **„Ändern"**).

- **Konto**
  Zum **Kopieren von Konten** von einem Kontenplan zum anderen gehen Sie am Besten wie folgt vor:

  - Öffnen Sie mit dem Selektionswerkzeug ein Formular mit **Konten** aus dem Kontenplan.
  - Nochmals mit dem Selektionswerkzeug öffnen Sie einen **Kontenplan** und geben einen neuen Datensatz ein.
  - Vom neuen Kontenplan ausgehend, gehen Sie mit der Taste **„Detail"** in ein (noch leeres) Formular für die Eingabe der Konten.
  - Durch Ziehen der Seite **"Felder"** des ersten Konten-Formulars über das leere **Konten-Formular** starten Sie die Übertragung.
  - Die Software wird einen neuen Datensatz generieren, die Werte übertragen, und dann fragen, ob in dieser Art die **ganze Liste** durchgearbeitet werden soll.

- **Produkt**
  Das **Ziehen von Produktedaten** von einem Formular über das andere führt zu einem besonderen Effekt. Produkte können in **Stücklisten** (Bill of Materials, BOM) organisiert werden. Die Stückliste ist eine **hierarchische Organisation** mit **Master-Produkten**, zu denen jeweils eine Gruppe von **Detail-Produkten** gehört. Stücklisten können gut in **Kaskaden** dargestellt und bearbeitet werden. Durch Drücken der **Detail-Taste** lassen sich die zu einem Produkt gehörenden Elemente selektionieren. Mit der **Master-Taste** kommt man leicht wieder zum übergeordneten Produkte eines Elements.

  Um eine neue **Verbindung** zwischen einem übergeordneten Produkte und seinem Element herzustellen, können Sie einfach die Seite **„Felder"** eines Produkteformulars über die entsprechende Seite eines anderen Produkteformulars **ziehen**. Zum **Trennen** eines Elements aus der Liste verwenden Sie die Taste **"entknüpfen"**/"disconnect", wobei das Element aber vorher als **Detail des übergeordneten Produkts** selektioniert worden sein muss.

- **Prozess**
  Prozess-Formulare lassen sich ebenso in **Kaskaden** organisieren wie Produkteformulare. Ziehen von "Prozess" über "Vertrag": Verwenden Sie diese Funktion um die Verträge in Gruppen (Pools) zusammenzufassen (zum Beispiel Schlüsselkunden oder kleine Kunden in Ihrem CRM).

- **Beleg**
  Besonders effizient ist das **Kopieren von Belegen** durch **Ziehen mit der Maus**. Wird ein **Beleg** von einem Belegfenster in ein anderes Belegfenster gezogen, startet das System ein **Dialog-Formular**, bei dem zwischen folgenden drei **Optionen** ausgewählt werden kann:

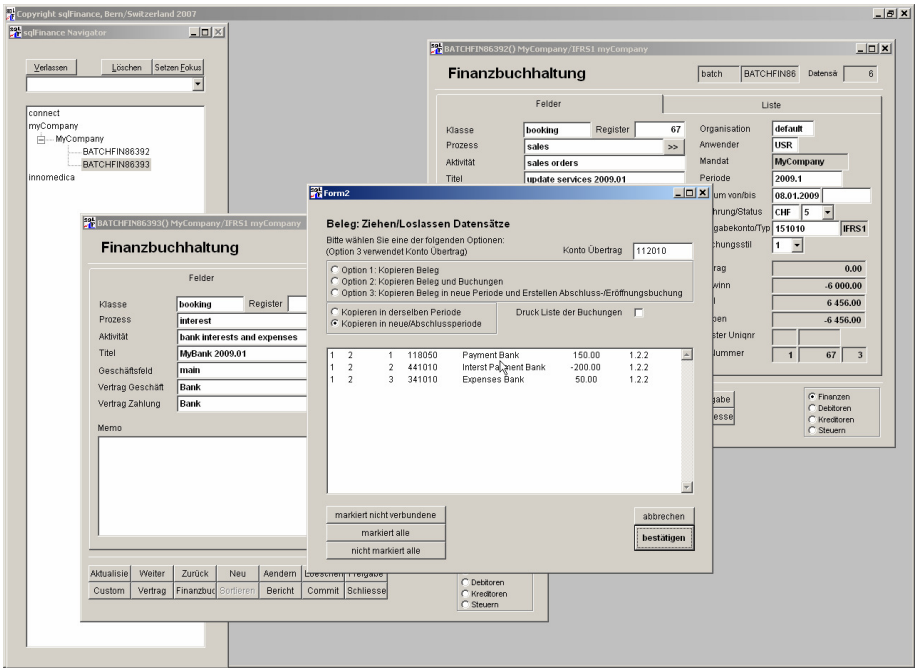
  1. **Kopieren Beleg**

  2. **Kopieren Beleg und Buchungen**

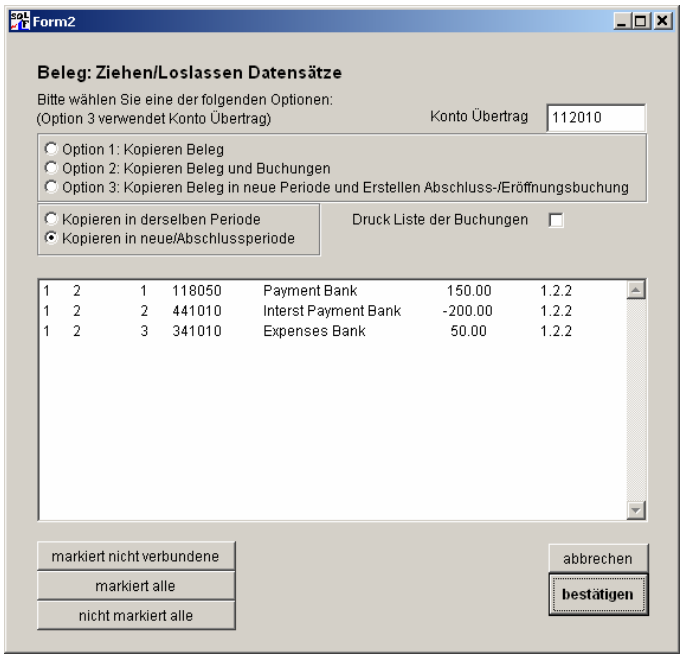
  3. **Kopieren Beleg in neue Periode und Erstellen Abschluss-/Eröffnungsbuchung**

  Option (3) ist vor allem geeignet für den **Abschluss offener Debitoren-Belege** in eine **neue Periode** (zum Beispiel am Jahresende). Für diese Funktion muss ein **Konto** für den **Übertrag** angegeben werden. Ferner kann der Anwender entscheiden, ob die Abschlussbuchung im Textfeld eine **detaillierte Liste** der Buchungen enthalten soll.

  Bei allen drei Optionen kann der Anwender wählen, ob der **Beleg** in der **bisherigen Periode** kopiert oder in die **neue Periode** übernommen werden soll. Beim **Übertrag** in die neue Periode wird das **Datum** der einzelnen Buchungen **angepasst**, also zum Beispiel eine Buchung vom **30.6.2008** auf den **30.6.2009** übertragen.



Übersicht zum Ziehen mit der Maus (Drag & Drop) von einem Belegformular zum anderen



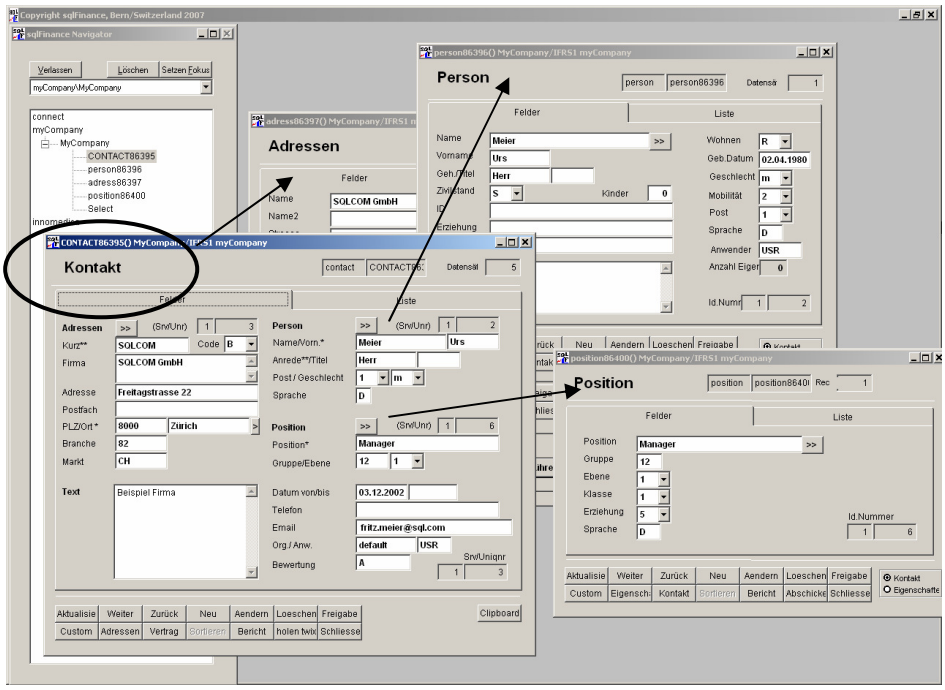
Detailansicht des Dialogformulars

Ziehen mit der Maus: verwenden Sie die rechte Maustaste auf der Seite "Felder" um den Seitenrahmen in Bewegung zu bringen

- **Task Manager**
  Ziehen einer Tabelle über den Task-Manager (Werkzeug "Datensätze"/"Records"): Durch Ziehen der Seite **"Felder"** eines Datenformulars über die Tabellenlliste des Task-Managers führt sogleich zu einem Eintrag. Mit der Taste **"holen"**/"get" kann der Datensatz wieder im Formular angezeigt werden.

- **Kontakt**
  In der **Kontaktverwaltung** (Customer Relationsship Management CRM) führt das Ziehen von **Adressen-**, **Personen-** oder **Positionsformulare** über das Kontaktformular nur zu einer Übertragung der **Werte** – ohne jedes Mal die Eingabe eines Datensatzes auszulösen. Dabei werden nicht nur Anwendungsdaten, sondern auch **Referenznummern** übertragen (server/uniqnr), die eine eindeutige Verbindung darstellen können.

  Die Kontakttabelle kann auch als **Detail** einer Adresse geöffnet werden. Durch Drücken der Taste **"Detail"**/"Detail" beziehungsweise **"Kontakt"**/"Contact" im Adressformular kann das verbundene Kontaktformular geöffnet werden. Mit der Taste **"Neu"**/"New" wird ein neuer Datensatz erstellt, der **Adressdaten** mitsamt **Referenznummer** sogleich übernimmt. Personen- und Positionsdaten können dann durch **Ziehen** der entsprechenden Seitenrahmen über das Kontaktformular ergänzt werden.



Das Kontaktformular mit Bezug zu Adresse, Person und Position

Das Master-Formular kann durch Doppelklick im Textfeld (Feld „Kurzname" für Fenster „Adresse", Feld „Name" für Fenster „Person" und Feld „Position" für Fenster „Position") oder durch einfachen Klick auf den Tastenknopf daneben aktiviert werden.

5. Berichte

Berichte starten Sie am einfachsten von einem **Anwendungsformular** aus (klicken Sie auf die Taste **"Bericht"**/"Report"). Der beste Bericht lässt sich meistens in einem **Master-Formular** finden und liefert dann Daten zu diesem **Master-Datensatz**, ergänzt mit **Listen** von passenden **Detaildaten**. Ein Bericht zu einer **Periode** wird zum Beispiel alle **Belege** und deren **Buchungen** auflisten, die zu dieser Periode gehören.

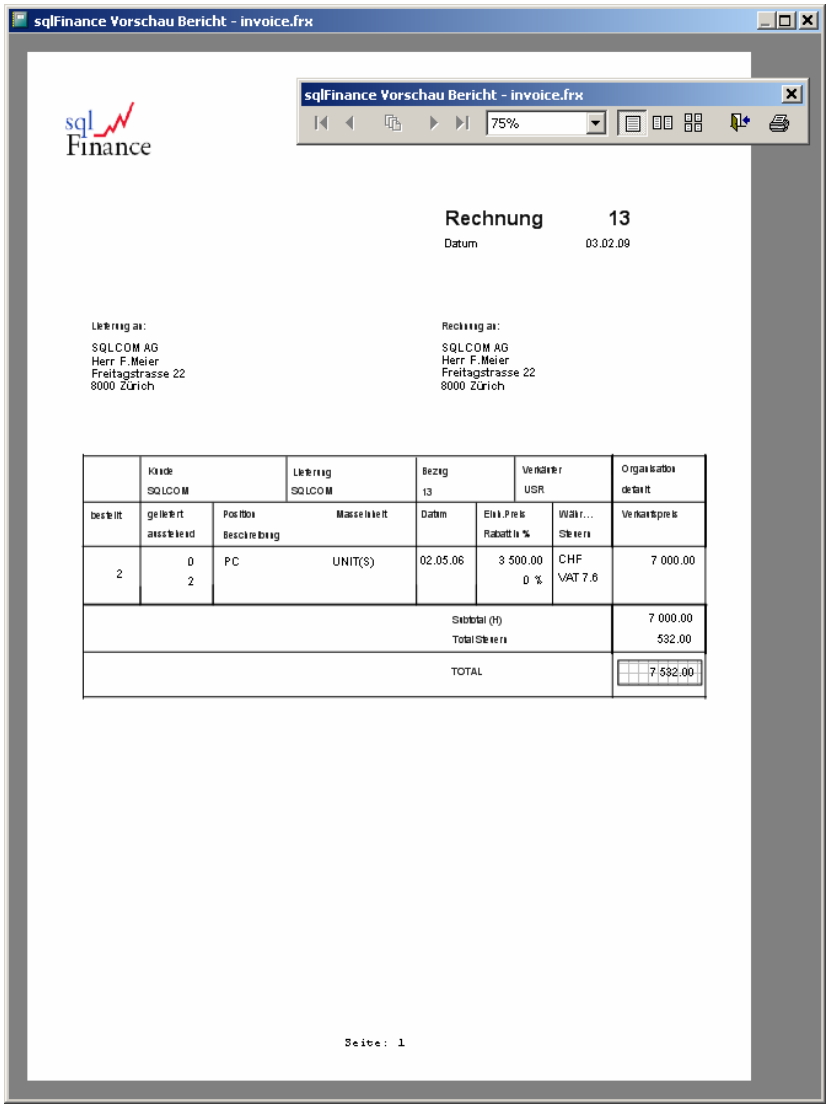
Mit **sqlFinance** können Sie einige Berichte auch vom **Selektionswerkzeug** aus starten. In den meisten Fällen ist es aber **einfacher**, den Bericht von einem **Anwendungsformular** aus abzurufen. In diesem Fall kann sich die Datenabfrage entweder auf den **aktuellen Datensatz** des Formulars (zum Beispiel den aktuellen Beleg, die aktuelle Periode) oder dessen **gesamte Datenliste** beziehen.

Das Werkzeug Bericht, von einem Periodenformular aus aufgerufen, mit den Berichten "Balance total slim", "Detail Balance" und "Payroll"

Die **Datenliste** wird verwendet, wenn im Bericht **keine** weiteren Angaben für einen **SQL Ausdruck** vorhanden sind (s. Seite "Felder" des Berichtwerkzeugs). Sobald ein SQL Ausdruck **gegeben** ist, geht die Suche vom **aktuellen Datensatz** aus und selektioniert **verbundene Details** gemäss Formulierung der Abfrage. Durch **Änderung** des SQL-Ausdrucks kann der Anwender **weitere Felder** in die Suche einschliessen oder die Suche in der **"Where-Clause"** weiter begrenzen.

Ohne Erwerb einer weiteren Software-Lizenz ist es möglich, den **Bericht** zu **bearbeiten** und die Darstellung zu **ändern**. Um zum Beispiel einen Bericht zu einer Periode zu erhalten, klickt man im Perioden-Formular auf die Taste **"Bericht"**/"Report" und aktiviert das entsprechende **Werkzeug**.

In den Berichtslisten lässt sich der bevorzugte **Bericht „Totale (Bilanz)"/"totals (Balance)"** leicht auffinden. Klicken Sie auf die Taste **"Ändern"**/"Change" um den Bericht zu ändern. **Bestätigen** Sie die Änderungen mit **ctrl-w**. Durch entsprechende **Doppelklicks** auf der Liste oder einfachen Klick auf die Taster **„Druck"** kann der Bericht an verschiedene Orte geschickt werden: Bildschirm, Drucker oder Textfeld auf Seite drei im Berichtswerkzeug (in der Form HTML oder ASCII). Vom Textfeld aus kann der Inhalt mit **Doppelklick** im **sqlFinance Editor** weiter bearbeitet werden.



Beispiel für einen einfachen Bericht
(Rechnung aus dem Verkaufsmodul)

6. Tastatur-Kombinationen

Die folgenden Tastatur-Kombinationen können in den Anwendungsformularen verwendet werden:

*CTRL-e:	Textverwaltung
*CTRL-s:	Selektion
*CTRL-n:	Weiter
*CTRL-p:	Zurück
*CTRL-a:	Neu
*CTRL-u:	Ändern, in Textformular verlassen mit Speichern
*CTRL-d:	Löschen
*CTRL-q:	Erneuern
*CTRL-k:	Behalten
*CTRL-q:	Bericht
*CTRL-g:	Seite Tabelle (grid)
*CTRL-l:	Seite Felder (list)
*CTRL-z:	Verlassen
*ctrl-F1:	Selektionswerkzeug aktivieren
*ctrl-F2:	Master
*ctrl-F3:	Detail
*ctrl-F9:	Master-Formular schliessen und Wert ändern
*ctrl-F10:	Abschliessen (im Änderungsmodus: Bezahlen)
*ESC:	Verlassen

V. Support & Kontakt

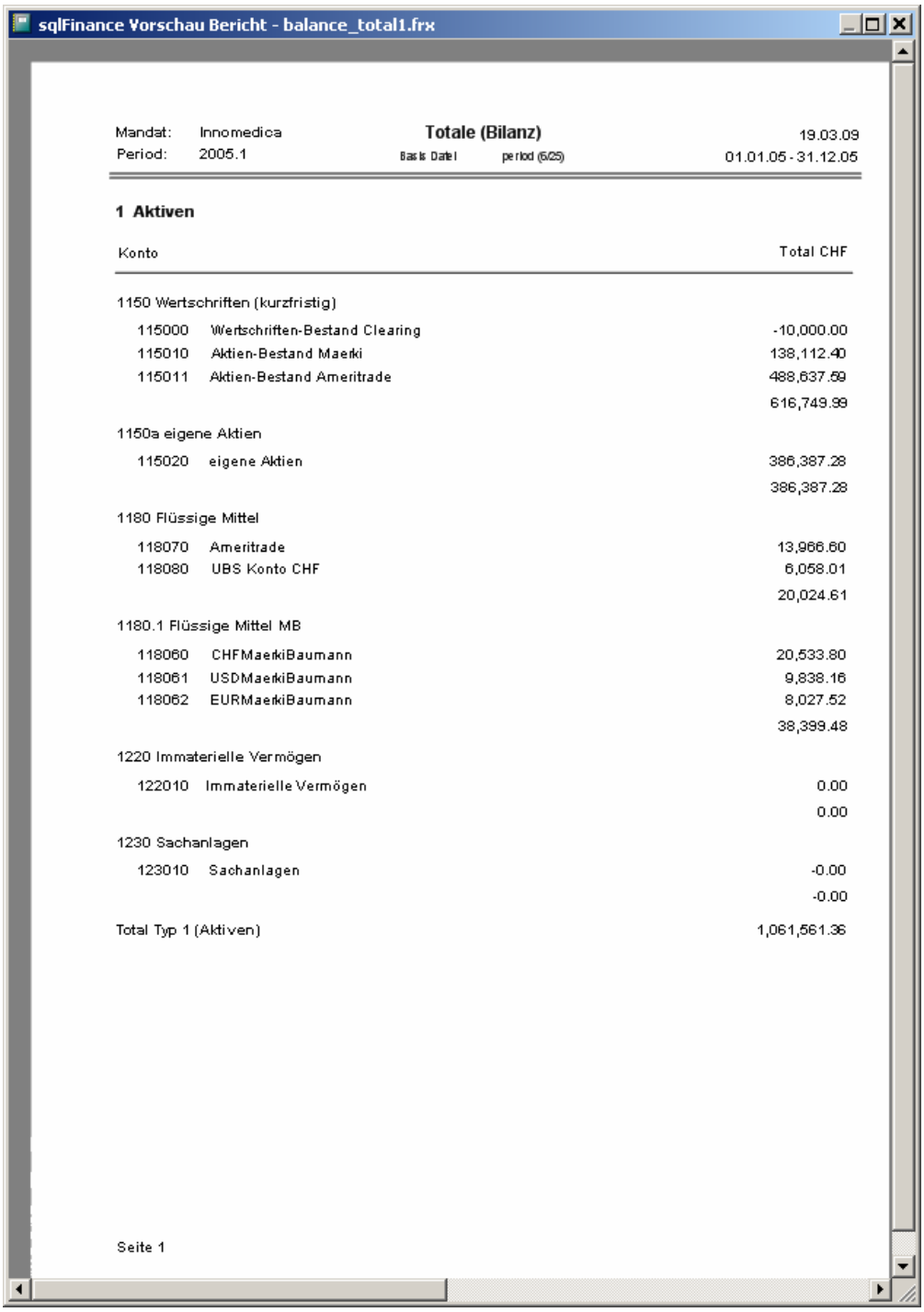
Brauchen Sie **Unterstützung** bei Installation oder dem Einsatz von **sqlFinance**? **Zögern Sie nicht**, uns zu **kontaktieren**. Gerne werden wir Ihre **Fragen beantworten** und Ihnen einen **Lösungsvorschlag** unterbreiten.

Homepage:	**www.sqlfinance.com**, www.ipag.ch
Email:	peter.halbherr@sqlfinance.com
Adresse:	Dr. Peter Halbherr Gesellschaftsstrasse 16 Postfach 3012 Bern
Telefon:	+41 31 311 04 27
Geschäftsstelle Zürich:	Zollikerstrasse 144 Postfach 8034 Zürich
Telefon:	+41 44 383 88 22
Fax:	+41 44 383 14 25

VI. Anhang

1. Bilanz und Erfolgsrechnung

Jahresabschluss der Innomedica Holding AG, Zug aus dem Jahr 2005 (Bericht „Totale Bilanz" von der Periode 2005.1 ausgehend)



sqlFinance Vorschau Bericht - balance_total1.frx

Mandat: Innomedica — **Totale (Bilanz)** — 19.03.09
Period: 2005.1 — Basis Datei — period (6/25) — 01.01.05 - 31.12.05

1 Aktiven

Konto		Total CHF
1150 Wertschriften (kurzfristig)		
115000	Wertschriften-Bestand Clearing	-10,000.00
115010	Aktien-Bestand Maerki	138,112.40
115011	Aktien-Bestand Ameritrade	488,637.59
		616,749.99
1150a eigene Aktien		
115020	eigene Aktien	386,387.28
		386,387.28
1180 Flüssige Mittel		
118070	Ameritrade	13,966.60
118080	UBS Konto CHF	6,058.01
		20,024.61
1180.1 Flüssige Mittel MB		
118060	CHFMaerkiBaumann	20,533.80
118061	USDMaerkiBaumann	9,838.16
118062	EURMaerkiBaumann	8,027.52
		38,399.48
1220 Immaterielle Vermögen		
122010	Immaterielle Vermögen	0.00
		0.00
1230 Sachanlagen		
123010	Sachanlagen	-0.00
		-0.00
Total Typ 1 (Aktiven)		1,061,561.36

Seite 1

sqlFinance Vorschau Bericht - balance_total1.frx - Page 2

Mandat: Innomedica
Period: 2005.1

Totale (Bilanz)

19.03.09
01.01.05 - 31.12.05

2 Passiven

Konto		Total CHF
2110 Eigenkapital		
211010	Aktienkapital	800,000.00
211020	ges.Reserven	400,000.00
211030	Reserven f.eigene Aktien	445,734.28
211040	Freie Agio Reserven	110,317.22
211090	Gewinnvortrag	-870,001.02
		886,050.47
2120 Kreditoren		
222010	Kreditoren	15,877.25
223010	AHV	1,587.70
		17,464.95
2130 Fremdkapital (langfristig)		
213070	Wandelanleihen	73,704.00
		73,704.00
2190 Abgrenzung passiv		
219020	a.o.Trans.Passiven	5,380.00
		5,380.00
Total Typ 2 (Passiven)		982,599.42
Gewinn Total		78,961.94

Seite 2

sqlFinance Vorschau Bericht - balance_total1.frx - Page 3

Mandat: Innomedica **Totale (Bilanz)** 19.03.09
Period: 2005.1 01.01.05 - 31.12.05

3 Aufwand

Konto		Total CHF
3110 Einkauf		
311050	Rundungen	124.73
		124.73
3230 Infrastruktur		
323011	Infrastruktur	1,118.50
		1,118.50
3250 Abschreibungen		
325010	Abschreibungen	11,800.00
325020	Abschreibungen VST	2.05
		11,802.05
3260 Dienstleistungen		
326040	Vermögensverwaltung	15,838.25
326045	Fremdleistungen Normal Fix	5,380.00
326051	Internet	35.00
326054	Werbung	1,396.90
		22,650.15
3410 Finanzaufwand		
341020	Bankspesen	93.40
341040	Akt.Darlehen Zinsen Aufwand	439.50
		532.90
3510 Steuern		
351010	Steuern	234.50
		234.50
Total Typ 3 (Aufwand)		36,462.83

Seite 3

sqlFinance Vorschau Bericht - balance_total1.frx - Page 4

Mandat: Innomedica **Totale (Bilanz)** 19.03.09
Period: 2005.1 01.01.05 - 31.12.05

4 Ertrag

Konto		Total CHF
4210 Währungsgewinne		
421010	Währungsgewinne	16,143.06
		16,143.06
4410 Finanzertrag		
441010	Finanzertrag Zinsen	5.93
		5.93
4420 Handelsertrag		
442020	Ertrag Wertschriftenhandel	1,300.00
442021	Bewertungsgewinn eigene Aktien	92,915.10
		94,215.10
4421 Handel eigene Aktien		
442022	Handel eigene Aktien	5,060.68
		5,060.68
Total Typ 4 (Ertrag)		115,424.77
Gewinn Total		78,961.94
Gesamt-Total		0.00

Seite 4

2. Bericht Prozesse und Kontierung

Bericht „Prozess/Konten (Master Prozess)“ von Master-Prozess „booking“ ausgehend, mit Detailprozessen und deren Kontierung

sqlFinance Vorschau Bericht - process_accounts02.frx - Page 1

Prozess/Konten (Master Prozess)

Process: booking Basis Datel process (1/26) 01.07.09

Prozess	booking	master booking process
Klasse	booking	includes as detail all booking processes (entered in the batch windows)
Zähler	20	

Prozess	interest	bank interests and expenses
Klasse	booking	
Zähler	3	

Konto	118050	SBC Bank Account	Bank Account Swiss Bank Corporation
Gruppe	1180	liquid assets	
Konto	341010	Finance Expenses (Bank)	Various Finance Expenses (Bank)
Gruppe	3410	finance expenses	
Konto	441010	Finance Revenues (Bank)	Finance Revenues Bank (interest payments)
Gruppe	4410	finance revenues	

Prozess	payable	accounts payable
Klasse	booking	payroll and infrastructure payables
Zähler	5	

Konto	118050	SBC Bank Account	Bank Account Swiss Bank Corporation
Gruppe	1180	liquid assets	
Konto	231010	Payroll Clearing	Payroll Clearing Account
Gruppe	2310	payroll	
Konto	312011	Purchase of Goods	
Gruppe	3120	Purchase Goods	
Konto	323010	Employee Expenses	Various Employee Expenses
Gruppe	3230	infrastructure	
Konto	323011	Infrastructure	
Gruppe	3230	Infrastructure	

Prozess	receivable	accounts receivable
Klasse	booking	
Zähler	2	

Seite 1 01.07.09

sqlFinance Vorschau Bericht - process_accounts02.frx - Page 2

Prozess/Konten (Master Prozess)

Process: booking 01.07.09

Prozess	booking		master booking process
Klasse	booking		includes as detail all booking processes (entered in the batch windows)
Zähler	20		
Konto	112010	Receivables (main)	Main Receivables Clearing Account
Gruppe	1120	accounts receivable	
Konto	118050	SBC Bank Account	Bank Account Swiss Bank Corporation
Gruppe	1180	liquid assets	
Prozess	sales		sales orders
Klasse	business		
Zähler	4		
Konto	112010	Receivables (main)	Main Receivables Clearing Account
Gruppe	1120	accounts receivable	
Konto	223010	VAT Payables	VAT Value Added Tax Clearing Account (payables and payments)
Gruppe	2230	tax VAT	
Konto	412020	Hardware	Computer Hardware (PC)
Gruppe	4120	sales goods	
Konto	413010	Sales of Services (goods)	Installation and Maintenance
Gruppe	4130	sales of services	

Seite 2 01.07.09