



GDI-Business-Line 3.x

SQL-System ohne BDE (TSQLSystem)

Kurz-Dokumentation

Stand: 12. Juli 2012
3.2.1.215

Einführung	3
Aufruf des neuen SQL-Systems.....	3
Neuheiten in Stichpunkten	3
Welche "alten" Funktionen sind entfallen?.....	4
Aufbau des SQL-Systems	4
Shortcuts.....	5
Selektionen speichern:	5
Kontextmenüs	6
Mehrschrittige SQLs / SQLBatch	8
Der Memo-Editor	8
Berechtigungen innerhalb des SQL-Systems.....	9
Optionen für den Aufruf des SQL-Systems	10
Beispiel-Aufruf des SQL-Systems über das Menü	10
Beispiel-Aufruf des SQL-Systems per Startdialog	10
Export- und Importfunktionen des SQL-Systems.....	11
Export.....	11
Import.....	12
Formatierung der Ergebnisspalten	13

Informationen zum neuen SQL-System

Stand 3.2.1.215 12.07.2012

Einführung

In der GDI-Business-Line 3.x (Bline 3.x) wurde das bisherige, BDE-behaftete SQL-System ab Version 3.2.1.208 durch ein neues SQL-System abgelöst, welches auf BDE-freien Datenzugriffskomponenten beruht. Ein Hauptaspekt bei der Umsetzung lag in der Unterstützung der bekannten Funktionalitäten, so dass bestehende abgespeicherte SQL-Auswertungen übernommen werden können. Gleichzeitig kamen neue Funktionen hinzu. Diese Dokumentation fasst die wichtigsten Informationen zusammen.

Aufruf des neuen SQL-Systems

Das neue SQL-System wird über den Menüpunkt "TSQSystem" aufgerufen. Bei einem Versionswechsel werden bestehende Menues durch die Versionsreorganisation automatisch aktualisiert, d.h. der alte Aufruf "TFSSQL" wird durch "TSQSystem" ersetzt.

Das alte SQL-System wurde als separate Anwendung aus der Bline herausgelöst und liegt nun als SQLSystem.exe vor. Falls erforderlich, kann diese Anwendung über einen Menüpunkt mit Verknüpfung zu dem neu geschaffenen Programmaufruf "SQLSystem" eingebunden werden. Somit steht – für eine Übergangsphase – die alte Funktionalität noch zur Verfügung. Da dadurch jedoch keine BDE-Freiheit erreicht werden kann, sollte der Rückgriff auf diese Option die Ausnahme bleiben!

Neuheiten in Stichpunkten

- Anzeige der gespeicherten SQLs in einer Baumansicht (TreeView analog zu Reportmanager/TReporter und Pivot-Auswertungen). Dabei Zugriff auf SQLs auf Programmebene und Mandantenebene, beliebige Unterordner zum jeweiligen Hauptordner "SQL" sind möglich.
- Über das Menü lässt sich voreinstellen, welches Unterverzeichnis (zum Verzeichnis SQL) beim Start des SQL-Systems im TreeView (Baum) angezeigt wird, ob nur bestimmte SQLs angezeigt werden sollen und ob eine bestimmte SQL markiert/angewählt sein soll. So können den Bediener(gruppen) gezielt die SQLs bereitgestellt werden, die für sie bestimmt sind.
- Sofern ein Bediener selbst SQLs verfassen darf, merkt sich das SQL-System zuletzt ausgeführte SQLs auch ohne explizites Speichern als SQL-Datei. Solche "freien SQLs" werden mandantenübergreifend pro Bediener in einer Datei [BedName].SQLs im SQL-Ordner unterhalb des Programmpfades gespeichert. Über Navigatorbuttons kann durch diese "freien SQLs" geblättert werden. Dies stellt insbesondere für Systembetreuer eine nützliche Funktion dar.
- Speichern von Selektionen: Werden SQLs abgespeichert verwaltet das SQL-System neben der eigentlichen SQL-Datei (*.SQL) zugehörige Selektionseinstellungen in einer gleichnamigen SQLSel-Datei (*.SQLSel). Zu jeder Selektionseinstellung gehören die Werte der Eingabe-Parameter, bei Bedarf ein Beschreibungstext und der Name einer Ausgabedatei. Zusätzlich wird in dieser Datei auch das Griddesign der Ergebnistabelle (Datengrid) festgehalten.
- Der SQL-Editor besitzt eine Syntaxhervorhebung.
- Zur Kennzeichnung von Parametern in SQL-Statements genügt ein Doppelpunkt, z.B. ...
`where Artikelnr = :Artikelnr` (im alten SQL-System musste man immer zwei

Doppelpunkte setzen, z.B. ... where Artikelnr = ::Artikelnr). Evtl. vorhandene zweifache Doppelpunkte werden beim Laden/Speichern einer SQL in einfache gewandelt. Somit sind SQLs mit Parametern nach dem Speichern mit dem neuen SQL-System nicht mehr konform mit dem alten SQL-System.

Welche "alten" Funktionen sind entfallen?

- Im alten SQL-System konnte die aktuelle Formatierung der Ergebnistabelle - z.B. nachdem Spaltenbreiten eingestellt oder Veränderungen über das Designen des Grids (rechte Maustaste auf Gridüberschrift) vorgenommen wurden - über einen Button in Form eines speziellen Kommentares in das SQL-Statement übernommen werden. Somit war dieselbe Formatierung bei späterem Abruf wieder verfügbar. Diese Funktion konnte nicht übernommen werden, da im neuen SQL-System eine andere Grid-Komponente zum Einsatz kommt. Im neuen SQL-System wird der Gridaufbau in einer zusätzlichen Datei (SQLSel-Datei) gespeichert.
- Ein Zugriff auf Paradox-Tabellen (*.db, *.mb) ist nicht möglich. Somit entfällt der Daten-Export nach Paradox bzw. das Öffnen und Importieren von Paradox-Tabellen.

Aufbau des SQL-Systems

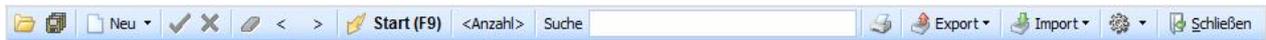
Folgende Hardcopy zeigt den Aufbau des neuen SQL-Systems:

The screenshot shows the SQL-System interface with several callout boxes:

- Drei Buttons zum Zugriff auf "freie SQLs"**: Points to three buttons in the top-left toolbar.
- SQL speichern**: Points to a button in the top-left toolbar.
- Kartei "Parameter" mit Parameter-Grid (oben) und Datengrid (unten)**: Points to the top-right panel containing a parameter grid.
- Kartei "SQL" zum Editieren/Schreiben der SQL**: Points to the central SQL editor area.
- Kartei "Statistik" (Informationen zur Ausführungszeit etc.)**: Points to the bottom-right panel.
- Neuanlage/Löschen/Speichern einer Selektion bei gespeicherten SQLs**: Points to a set of icons (+, -, x) in the middle-left area.
- Datengrid**: Points to the data table at the bottom of the interface.
- TreeView zur Anzeige/Auswahl der gespeicherten SQLs**: Points to the left-hand sidebar.
- Symbol mit Schloss a SQL aus Proppfad, Symb. ohne Schloss a aus MandPfad**: Points to a lock icon in the sidebar.
- Umschaltung zur direkten Anzeige der Tabellen der Datenbank. Dort kann per Doppelklick auf eine Tabelle diese direkt geöffnet werden.**: Points to the bottom-most section of the interface.

KUNDENNR	ADRESSGRP	NAME1	NAME2	STRASSE	PLZ	ORT
11000	100	Weiss Computer		Lindenstr. 11	67059	Ludwigs
18000	100	Edgar Kranz oHG	Hardware	Klausewitzstr. 1	78570	Mühlhei
19000	100	Anton Winkler	Fleisch	chlachthof 8	76829	Landau
21000	100	Joseph Kaiser	Kfz-Re	den Neugärten 121	67483	Edeshei
50000	100	Computerstudio 3000	Filiale Greffen	Gartnischer Weg 33	47929	Grefrath
50200	100	Computerstudio 3000	Filiale Berlin	Treskowallee 75	10318	Berlin

Die Symbolleiste (ganz oben in der Hardcopy) enthält von links nach rechts:



- Zwei Buttons zum Laden und Speichern von SQLs von/auf der Festplatte
- Button "Neu" zum Leeren des SQL-Editors. Der Button beinhaltet ein Pulldown-Menü mit verschiedenen SQL-Syntaxvorschlägen als eine Art "Nachschlagewerk"
- Zwei Buttons zum Speichern/Verwerfen von Änderungen an einer gespeicherten SQL
- Drei Buttons zum Zugriff auf die sog. "freien SQLs". Über "<" und ">" kann durch die zuletzt ausgeführten SQLs geblättert werden, über das links danebenstehende "Radierer"-Symbol kann eine SQL aus der Historie der "freien SQLs" gelöscht werden.
- Start-Button zum Ausführen der aktuellen SQL (F9)
- "Anzahl"-Button zur Ermittlung der Datensatzanzahl der Ergebnistabelle (Anzahl Datensätze im Datengrid)
- Sucheingabe-Feld zur Volltextsuche im Datengrid
- Drucker-Button zum Ausdruck des SQL-Editors, des Datengrids oder des Statistik-Grid, je nachdem wo sich der Fokus befindet
- Export- und Import-Buttons
- Designer-Button
- Button "Schließen" zum Beenden des SQL-Systems

Shortcuts

Folgende Shortcut-Funktionen sind im SQL-System hinterlegt:

F4: Memo-Editor öffnen (nur bei Blob-Feldern der Ergebnistabelle)

F5: Refresh-Funktion für SQL-TreeView und Tabellenübersicht

F9: Ausführen der SQL

F11: Vollbildanzeige der Ergebnistabelle "ein/aus" in der rechten Bildschirmhälfte

Selektionen speichern

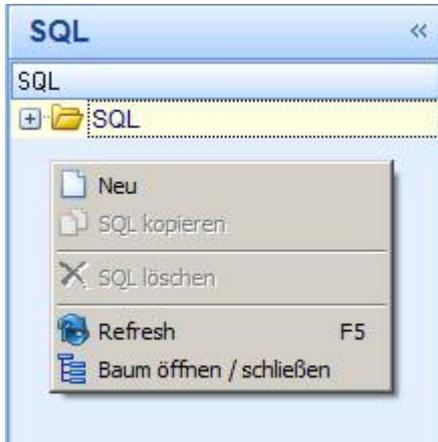
Hier gibt es zwei Modi, je nachdem ob es sich um eine gespeicherte SQL oder um eine Historien-SQLs (freie SQL) handelt:

- **Gespeicherte SQL:** Dieser Modus ist aktiv, wenn man im TreeView eine gespeicherte SQL gewählt oder über den Laden-Dialog auf eine SQL-Datei zugegriffen hat. Erkennbar ist dieser Modus auch an der Caption/Überschrift des SQL-Systems, wenn dort eine SQL-Datei angezeigt wird. Beim Speichern einer solchen SQL wird eine gleichnamige SQLSel-Datei mitgespeichert/erzeugt. Auch das Speichern einer Selektion im Parametergrid bewirkt die Anlage/Änderung der SQLSel-Datei. Neben den "Nutzdaten" (Beschreibung, Ausgabedatei und Werte von Parametern) wird auch das Layout des Parametergrids und der Aufbau des Datengrids gespeichert.
- **Historien-SQL (freie SQL):** Dieser Modus ist aktiv, wenn man (nach Klick auf den Button "Neu") SQLs verfasst und absetzt, ohne diese explizit zu speichern oder wenn man bereits auf eine Historien-SQL zugreift. Erkennbar ist dieser Modus wenn in der Caption/Überschrift des SQL-Systems "Neu" angezeigt wird. Hier gibt es keine SQLSel-Datei. Die Historie wird pro Bediener im Verzeichnis Programmpfad\SQL als [BedName].sqls gespeichert. Die Selektion einer freien SQL wird nur bei deren Ausführung in die Historienliste übernommen. Das Layout des Parametergrids sowie der Aufbau des Datengrids werden nicht gespeichert.

Kontextmenüs

An den diversen Punkten des SQL-System findet man Kontextmenüs mit folgenden Funktionen:

Kontextmenü SQL-Baumansicht (TreeView)



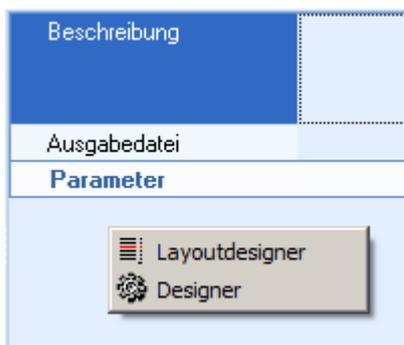
- **Neu:** Funktion wie Button "Neu" in der Symbolleiste. Aktiviert den Karteireiter "2 - SQL" mit leerem SQL-Editor
- **SQL kopieren:** Nur anwählbar, wenn im TreeView eine SQL markiert ist. Öffnet einen Windows "Speichern unter-" Dialog zum Abspeichern der SQL unter anderem Namen und/oder an anderem Ort. Eine ggfs. vorhandene SQLSel-Datei wird ebenfalls mitkopiert.
- **SQL löschen:** Löscht SQL- und SQLSel-Datei von Platte
- **Refresh (F5):** Aktualisieren des TreeViews
- **Baum öffnen / schließen:** Expandiert bzw. schließt alle Knoten des SQL-TreeView

Kontextmenü Tabellen-Übersicht



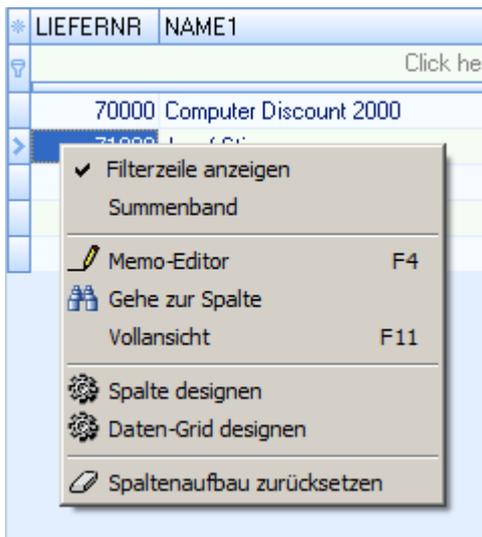
- **System Tabellen:** Blendet die normalerweise nicht sichtbaren System-Tabellen der Firebird-Datenbank ein
- **Refresh (F5):** Aktualisieren der Tabellenansicht

Kontextmenü Parameter-Grid



- **Layoutdesigner:** Wie gleichnamiger Menüpunkt im Pulldown-Menü des Designer-Rades in der Symbolleiste. Öffnet den Layoutdesigner zur Gestaltung des Layouts des Parameter-Grids
- **Designer:** Wie Menüpunkt "Parameter-Grid" im Pulldown-Menü des Designer-Rades in der Symbolleiste. Öffnet den Gridesigner zur Gestaltung des Parameter-Grids

Kontextmenü Ergebnistabelle



- **Filterzeile anzeigen:** Filterzeile oberhalb des Grids anzeigen/ausblenden (in der Hardcopy aktiviert = eingeblendet)
- **Summenband:** Anzeigen des Summenbandes unterhalb des Grids. Dort können wiederum per rechter Maustaste Summen, Durchschnittswerte ... pro Ergebnisspalte angezeigt werden. Hinweis: Kann bei großen Datenmengen einige Zeit beanspruchen, da die Daten in den Speicher geladen werden.
- **Memo-Editor (F4):** Bei Ergebnisspalten des Datentyps GDI_Memo (Blob-Felder) lässt sich hier deren Inhalt in einem separaten Editor anzeigen
- **Gehe zur Spalte:** Öffnet ein Suchfenster zur Eingabe eines Spaltennamens. Sofern in der Ergebnistabelle vorhanden kann die Spalte direkt angesprungen werden (somit entfällt das horizontale Scrollen durch den Grid). Die Suche erfolgt nach dem Prinzip "Spaltenname beginnt mit ohne Beachtung der Groß-/Kleinschreibung", die Suche nach Name würde somit auch NAME1, NAME2 und NAME3 finden
- **Vollansicht (F11):** Blendet des über der Ergebnistabelle liegenden Bereich mit Parameter-Grid, SQL-Editor und Statistik-Anzeige weg, so dass für die Ergebnistabelle die komplette Maskenhöhe zur Verfügung steht
- **Spalte designen:** Ruft den CRM-Designer für die aktive Ergebnisspalte auf
- **Daten-Grid designen:** Wie Menüpunkt "Daten-Grid" im Pulldown-Menü des Designer-Rades in der Symbolleiste. Ruft den CRM-Designer für die Ergebnistabelle auf
- **Spaltenaufbau zurücksetzen:** Bei wiederholtem Absetzen einer SQL wird der Aufbau der Ergebnistabelle nicht verändert. Somit bleibt die Anzeige z.B. bzgl. eingestellter Spaltenbreiten oder bewusst umsortierter/verschobener Spalten erhalten. Die Funktion ordnet die Ergebnisspalten wieder gemäß der (ursprünglichen) Reihenfolge im select-Abschnitt des SQL-Statements an und stellt die Standardformatierung wieder her (Löschen von geänderten Spalteneinstellungen wie Farbe, Ausrichtung, Formatierungen). Die Anwendung dieser Funktion ist z.B. auch sinnvoll bei Veränderungen im select-Abschnitt der SQL.

Mehrschrittige SQLs / SQLBatch

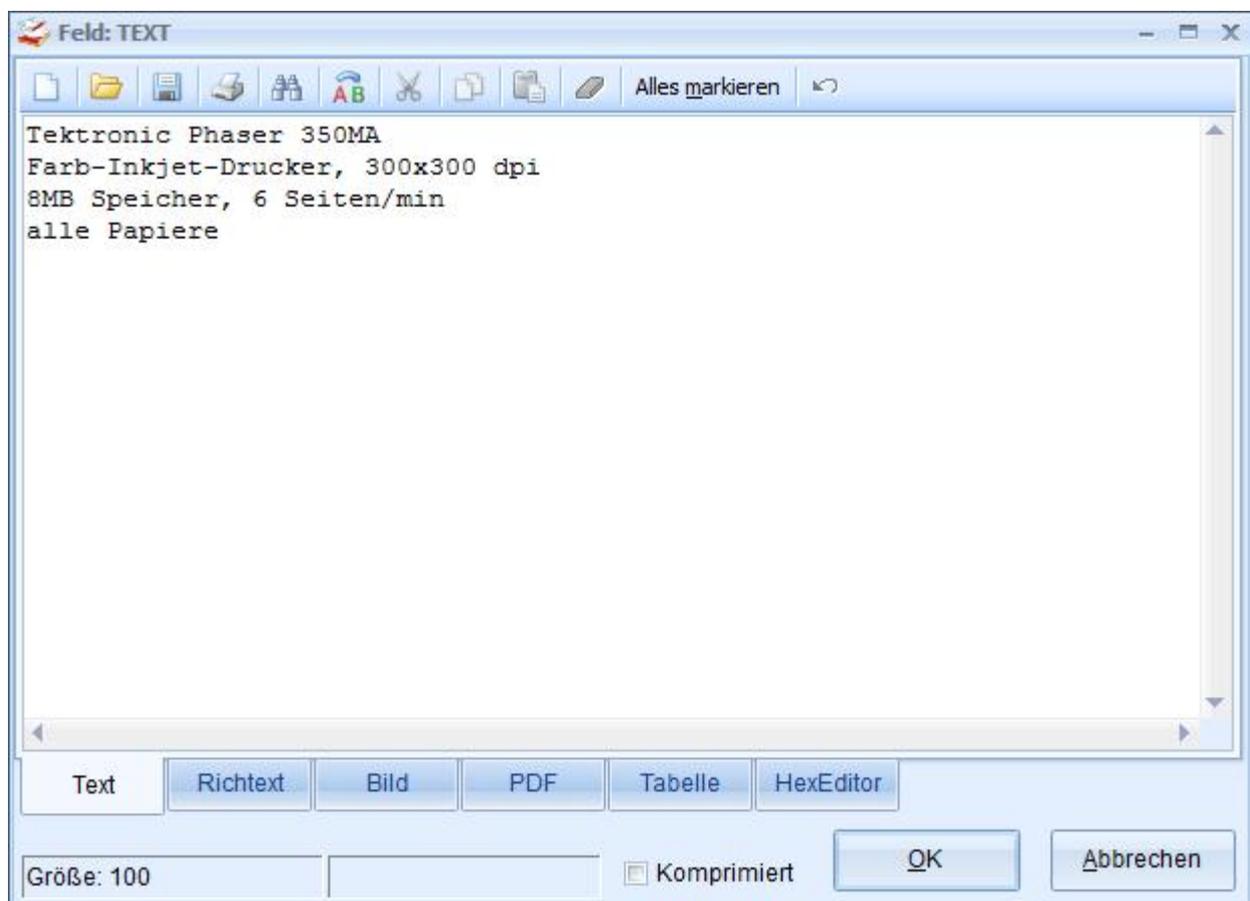
Man kann im SQL-System mehrere Abfragen hintereinander absetzen (also praktisch ein "SQL-Batch" schreiben). Hierbei ist zu beachten:

- Jede SQL-Anweisung muß mit einem Semikolon (;) als Befehlstrenner abgeschlossen werden. Das Semikolon sollte unmittelbar am Ende der jeweiligen SQL-Anweisung stehen.
- Die einzelnen Anweisungen sind durch einen Zeilenumbruch getrennt zu erfassen.
- Das Ergebnis jeder einzelnen Abfrage wird temporär im Speicher festgehalten. Dies ermöglicht innerhalb des Batches den Zugriff auf die Ergebnistabelle einer vorausgegangenen Abfrage. Eine solche Ergebnistabelle kann unter dem Namen "&&SQLn" angesprochen werden. Das n steht hierin für die Nummer der SQL-Abfrage innerhalb des Batches.

Wichtig: Dies gilt nur für das SQL-System bzw. den ExecuteSQL-Befehl innerhalb von GDI-BASIC. Mehrschrittige SQLs sind beispielsweise nicht bei den Abfrageobjekten innerhalb des FastReporters möglich.

Der Memo-Editor

Bei Ergebnisspalten des Datentyps GDI_Memo (Blob-Felder) lässt sich hier deren Inhalt in einem separaten Editor anzeigen. Er kann per rechter Maustaste oder über den Shortcut F4 geöffnet werden. Je nach Inhalt des Blob-Feldes sind verschiedene Ansichten/Bearbeitungsmodi über Karteireiter anwählbar. Die Hardcopy zeigt den Memo-Editor am Beispiel des Feldes "Text" einer Belegposition:



Berechtigungen innerhalb des SQL-Systems

Die über das Menüsystem eingestellten Rechtstufen werden im SQL-System wie folgt ausgewertet:

Rechststufe	Auswirkung
grün (alle Rechte)	<ul style="list-style-type: none"> • Volle Berechtigung • Nur in dieser Rechtstufe ist der Import-Button aktiv (Importe über Import-Button möglich)
dunkelgrün (Einfügen/Ändern)	<ul style="list-style-type: none"> • Kein Löschen einer SQL • Das Recht wirkt auf den Daten-Grid, so dass das Löschen von Datensätzen im Grid der Ergebnistabelle nicht möglich ist. Jedoch können Delete-Aktionen per SQL-Statement durchgeführt werden.
gelb (Ändern)	<ul style="list-style-type: none"> • keine Neuanlage einer SQL, kein Kopieren einer SQL, Änderung von SQL-Statements erlaubt • Änderung von Datensätzen erlaubt. Das Recht wirkt auf den Daten-Grid, so dass hier Einfügen und Löschen von Datensätzen im Grid der Ergebnistabelle nicht möglich ist. Jedoch können Insert- oder Delete-Aktionen per SQL-Statement durchgeführt werden.
türkis (Nur Lesen)	<ul style="list-style-type: none"> • keine Neuanlage einer SQL, kein Kopieren einer SQL, keine Änderung bestehender SQLs • keine Bearbeitung von Datensätzen im Datengrid <u>und auch keine Datenänderung per SQL</u> (nur lesende Transaction ist zugelassen)
rot (kein Recht)	<ul style="list-style-type: none"> • SQL-System kann nicht aufgerufen werden, Menüpunkt ist gesperrt/inaktiv

Weiterhin wird der Laden-Button zum Laden von SQLs deaktiviert, wenn der Aufruf des SQL-Systems mit Übergabe eines Verzeichnisses erfolgt, es sei denn der Bediener besitzt die volle Berechtigung (grün – alle Rechte). Siehe hierzu auch den folgenden Abschnitt.

Hinweis: Es gibt Überlegungen zwecks Erhöhung der Transparenz die Vermischung von Datenberechtigung und Funktionsberechtigung aufzuheben und dies dahingehend umzugestalten, dass

- 1) die Rechtstufen des Menüs (grün bis rot) nur für die Datenbearbeitung verwendet werden
- 2) Berechtigungen anderer Art wie z.B. darf jemand SQLs erstellen, ändern, löschen, ist der Export- Button aktiv etc. über Parameter eingestellt werden (im Menü in den Feldern Parameter 1-4, bei Startdialog durch Übergabe im Zusatzparameter, siehe auch folgenden Abschnitt)

Optionen für den Aufruf des SQL-Systems

In analoger Weise wie z.B. beim Reportmanager stehen Aufrufoptionen für das SQL-System zur Verfügung, welche entweder über Parameter im Menü oder bei Aufrufen aus GDI-Basic heraus über die Startdialog-Funktion angegeben werden. So lässt sich voreinstellen, welches Unterverzeichnis (zum Verzeichnis SQL) beim Start des SQL-Systemes im TreeView (Baum) angezeigt wird, ob nur bestimmte SQLs angezeigt werden sollen und ob eine bestimmte SQL markiert/angewählt sein soll.

- *Verzeichnis=Verzeichnisname* → Im TreeView des SQL-Systems werden nur die SQLs des übergebenen Verzeichnisses (=Unterverzeichnis zum Verzeichnis SQL) angezeigt. Der Laden-Button zum Laden von SQLs ist deaktiviert, es sei denn der Aufruf erfolgt in der vollen Berechtigung (grün – alle Rechte)
- *IncludeSQLs=Dateifilter* → Übergabe eines Dateifilters, so dass nur SQL-Auswertungen mit passenden Dateinamen angezeigt werden
- *ExcludeSQLs=Dateifilter* → Übergabe eines Dateifilters, um SQL-Auswertungen mit passenden Dateinamen von der Anzeige auszuschliessen, also "unsichtbar" zu machen
- *SQLName=Dateiname* → Markiert die angegebene SQL-Auswertungen im TreeView

Beispiel-Aufruf des SQL-Systems über das Menü

Das Beispiel zeigt den Einsatz der drei o.a. Optionen beim Aufruf des SQL-Systems über das Menü. Somit werden nur bestimmte Auswertungen angeboten, andere Auswertungen sind für den Bediener nicht sichtbar.

Programmaufruf: TSQLSystem
Parameter1: Verzeichnis=Tagesauswertungen
Parameter2: IncludeSQLs=RE_GU_Uebersicht*.sql
Parameter3: SQLname=RE_GU_Uebersicht1.sql

Beispiel-Aufruf des SQL-Systems per Startdialog

Das neue SQL-System kann auch über die GDI-Basic-Funktion Startdialog aufgerufen werden. Die Optionen werden hierbei über den Parameter "Zusatz" übergeben.

Analog sieht obiges Beispiel per Basic über die Startdialogfunktion wie folgt aus:

```
Dialog := "TSQLSystem" ;
Modal := true;
Start := "";
Default := "";
Return := "";
Filter := "";
Zusatz := "IncludeSQLs=RE_GU_Uebersicht*.sql" + CHR(13,10) +
          "SQLname=RE_GU_Uebersicht1.sql" + CHR(13,10) +
          "Verzeichnis=Tagesauswertungen";
Startdialog(Dialog,Modal,Start,Default,Return,Filter,Zusatz);
```

Export- und Importfunktionen des SQL-Systems

Export

Unter diesem Menüpunkt öffnet sich ein DropDown-Menü mit verschiedenen Exportfunktionen:



Griddaten exportieren bzw. **Exportieren und öffnen**: Die oberen beiden Einträge betreffen Exporte der Ergebnistabelle als Datei und sind aktiv, wenn zuvor eine SQL ausgeführt wurde. Über sie kann der Inhalt des Datengrids exportiert werden, entweder als alleiniger Export ("Griddaten exportieren") oder als Export mit anschließendem Öffnen der erzeugten Datei mit einer verknüpften Anwendung ("Exportieren und öffnen"). Für den Datenexport steht eine Auswahl diverser Datenformate zur Verfügung. Zu den aus dem alten SQL-System bekannten GDI-eigenen Exportroutinen wurden Exportroutinen des Komponenten-Herstellers des Datengrids aufgenommen. Letztere sind an dem Namensbestandteil "DevExpress" bei den Exporteinstellungen erkennbar. (D.h. bei gleicher Dateiendung sind inhaltlich Unterschiede gegeben).

SQL für Serienbrief: Über diesen Punkt kann eine SQL zur Erstellung von Serienbriefen angetestet werden (siehe auch Abschnitt "Serienbrief mit erweiterter Datenbasis" in der technischen Dokumentation zur Bline). Man benötigt eine SQL, welche als Rückgabewerte (zwingend) die AdressID plus ggfs. gewünschte Zusatzinformationen/Felder umfasst. Diese SQL wird an den Briefeditor übergeben, der daraus eine erweiterte Kontaktliste erstellt. Im Unterschied zur "normalen" Kontaktliste bei Erstellen eines Serienbriefes aus der Adresstabelle heraus umfasst die erweiterte Kontaktliste neben den eigentlichen Kontaktdaten auch die Zusatzdaten aus der SQL. Diese können über Variablen im Serienbrief ausgegeben werden. Die Syntax für eine Variable lautet "_" + Feldname der SQL. Beispiel:

```
select K.Adressnr as AdressID, P.ProjektNr as KFZNR, p.ausfab as HU from
Kunden k, projekt P where K.kundennr = P.Kundennr and getMonth(p.ausfab)= 6
and getYear(p.ausfab)= 2012
```

In diesem Falle kann die KFZNR über "_KFZNR" und der HU-Termin über "_HU" ausgegeben werden.

Hinweise:

- Die Ausführung der SQL im SQL-System ist nicht erforderlich, wird allerdings empfohlen um das Ergebnis vorab kontrollieren zu können.
- Die SQL darf weder mehrschrittig sein noch Parameter enthalten.

Der Button ist in erster Linie zum Testen der Funktion vorgesehen. Für die eigentliche Anwendung ist die Einbettung in GDI-Basic gedacht, dort lassen sich weitere Steuerungen vornehmen, wie z.B. die Übergabe von Betreff und Briefvorlage.

Adressen kopieren: Diese Funktion ist anwählbar, sofern im Ergebnis eine Spalte "AdressID" enthalten ist. Es werden die AdressIDs aller Datensätze in dem GDI-internen Format zum Kopieren/Einfügen von Adressen in die Zwischenablage überstellt. Somit können diese an

anderen Stellen im Programm wie z.B. in einem Adressordner der Adresstabelle oder in die Adressliste eines Serienbriefes per rechter Maustaste "Adressen einfügen" verwendet werden.

Beispiele:

```
Select AdressID from Beleg where ...
```

à Selektion von Adressen aus Belegen

```
Select Adressnr as AdressID from Kunden where Adressgrp = '200'
```

à im Kundenstamm ist die AdressID zwar enthalten, allerdings im Datenfeld "Adressnr". Über die Angabe des Aliasnamens (as AdressID) kann das gewünschte Ergebnis erreicht werden.

Import

Das DropDown-Menü des Import-Buttons umfasst zwei Einträge: "Import" und "Import in temporäre Tabelle":



Import: Beim Import können CSV-Dateien oder dBase-Dateien angesprochen werden. Aktuell findet dabei die Verarbeitung nach folgendem festen Schema statt, eine Art Assistent zur individuellen Gestaltung eines Imports ist als künftige Erweiterung vorgesehen:

- Vor dem Einlesen ist eine Tabelle auszuwählen, in welche die Daten eingelesen werden sollen (Doppelclick öffnet die Tabelle zur Bearbeitung)
- Schlüsselfelder müssen in der Quelldatei enthalten sein (z.B. Kundennr oder Artikelnr)
- Namensgleiche Felder werden eingelesen
- Nicht namensgleiche Felder werden überlesen.
- Es erfolgt kein Import von Daten in andere Tabellen (z.B. VK-Preise in Tabelle Preise beim Einlesen von Artikeldaten aus einer Artikel.csv).
- Vorhandene Datensätze werden aktualisiert, nicht vorhandene Datensätze werden angelegt

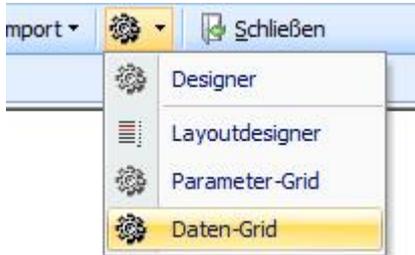
Nach dem Import wird die Quelldatei in Quelldateiname.bak umbenannt z.B. Kunden.csv à Kunden.csv.bak

Weitere grundlegende Informationen zum Thema Datenimport siehe auch in der Dokumentation zum Thema *CSV-Datenimport in die Bline* (unter Zuhilfenahme eines universellen GDI-Basic-Programms).

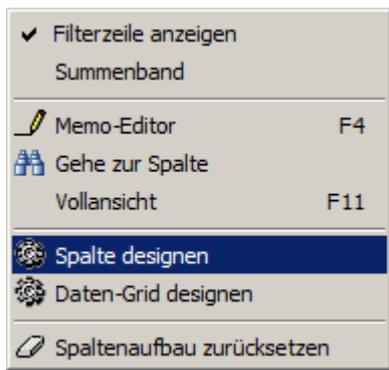
Import in temporäre Tabelle: Diese Funktion bietet die Möglichkeit dBase-Tabellen zu öffnen, einzusehen und bei Bedarf auch über den Datengrid zu bearbeiten. Zu diesem Zweck wird die dBase-Tabelle in eine temporäre FireBird-Tabelle der Datenbank überführt. Sofern man diese bearbeitet hat kann man sie über die Export-Funktion wiederum als dBase-Tabelle exportieren. Beim Verlassen des SQL-Systemes wird die temporäre FireBird-Tabelle aus der Datenbank gelöscht.

Formatierung der Ergebnisspalten

Sofern die Spalten der Ergebnistabelle speziell formatiert werden sollen (per Defaulteinstellung erfolgt die Ausgabe von alphanumerischen Feldtypen linksbündig, Zahlenwerte werden rechtsbündig dargestellt) kann über das Designer-Rad der Designer des Daten-Grids aufgerufen werden:



Alternativ kann man diesen Aufruf auch per rechter Maustaste auf der Ergebnistabelle vornehmen oder aber - was in vielen Fällen ausreichend sein dürfte – per rechter Maustaste den Designer gezielt die aktive Spalte aufrufen:



Es wird der aus dem CRM-Bereich der Business-Line bekannte Griddesigner für den vertikalen Grid aufgerufen. Um beispielsweise eine Zahlendarstellung mit Tausendertrenner zu erreichen ist folgendes einzustellen:

Properties → psEdit einstellen → es erscheint das "+" zum Einblenden weiterer Unterproperties. Dort ist die Eigenschaft GDIMask → R###,###,##0.00 einzustellen.

Desweiteren sollte man darauf achten, dass bei den Eigenschaften
EchoMode → eemNormal
MaskKind → emkStandard
CharCase → ecNormal
EditMask → "leer"
eingestellt ist (dies sind normalerweise auch die jeweiligen Defaulteinstellungen dieser Eigenschaften)

