

Schulungsunterlagen Makroassistent

© Copyright 2023 by SelectLine Software AG, CH-9016 St. Gallen

Kein Teil dieses Dokumentes darf ohne ausdrückliche Genehmigung in irgendeiner Form ganz oder in Auszügen reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden. Wir behalten uns vor, ohne besondere Ankündigung, Änderungen am Dokument und am Programm vorzunehmen. Die im Dokument verwendeten Soft- und Hardware-Bezeichnungen sind überwiegend eingetragene Warenbezeichnungen und unterliegen als solche den gesetzlichen Bestimmungen des Urheberrechtsschutzes.

03.11.2023/pe/V4.1

Inhalt

1	Vorwort	3
2	Struktur und Verwaltung des Makroassistenten	4
2.1	Überblick	4
2.2	Bedienung	5
2.2.1	Benutzereinstellungen.....	6
2.1.1	Die Funktionsleiste, Bedienelemente und Symbole des Makro-Assistenten.....	7
2.1.2	Das Kontextmenü (rechte Maustaste) der Baumstruktur.....	8
2.3	Der Makro Editierbereich	9
2.3.1	Einstellungen.....	9
2.3.2	Untermaske Bearbeiten	10
2.3.3	Parameter	10
2.3.4	Bestätigungsdialoge für „ausgeführt“, „gedruckt“, „gespeichert“	12
2.3.5	Anlegen einer Folge von Makros	12
3	Datenbankabfragesprache „Transact SQL“	14
3.1	Anweisungen.....	14
3.2	Klauseln	15
3.3	Funktionen	16
3.3.1	Stringfunktionen	16
3.3.2	Aggregatfunktion	17
3.3.3	Datenfunktion	17
3.4	Operatoren	18
3.5	Prädikate	18
3.6	Datendefinition	18
3.7	Besonderheiten der Datenbankabfragesprache T-SQL	19
3.8	Formularentwicklung für gedruckte Ergebnismengen	20
3.9	Datenbankabhängige Unterschiede in der SQL-Syntax	22
4	Anhang	24
4.1	Glossar	24
4.2	Dank	25
4.3	Ihre Notizen und Erkenntnisse.....	26

1 Vorwort

Vielen Dank für Ihr Interesse an SelectLine und dem Besuch dieses Kurses „Makroassistent“. Wir freuen uns sehr über Ihre Teilnahme und sind überzeugt, dass Ihnen diese Software eine grosse Unterstützung in Ihrer täglichen Arbeit sein wird. Die bedienerfreundliche Benutzeroberfläche wird es Ihnen ermöglichen, dass Sie schnell erste Erfolge erzielen können und Ihnen die Arbeit leicht von der Hand gehen wird. Aber lassen Sie sich nicht täuschen, auch Sie werden stets wieder neue Funktionalitäten und Möglichkeiten entdecken, welche dieses Programm bietet.



Ziel dieses Kurses ist es zu verstehen, wie der Makroassistent bedient werden kann und welche Besonderheiten im Vergleich zum regulären Transact-SQL verwendet werden können.

Um Ihnen das Arbeiten mit diesem Lehrmittel so einfach wie möglich zu machen, verwenden wir in diesem Kurs – und später auch in den weiteren Kursen – Symbole, die Ihnen einen raschen Überblick der wichtigsten Punkte geben sollen. Dies, da auch das Programm über Symbole oder sogenannte „Icons“ gesteuert wird. Hier deshalb kurz ein Überblick über die Symbole, die Ihnen in diesen Unterlagen begegnen werden. Das Erste haben Sie bereits kennengelernt.



Lernziele

Neben diesem Symbol sehen Sie, was das Ziel dieser Einheit ist oder welches Wissen Sie neu erwerben.



Hinweise

Hier erfahren Sie wichtige Tipps, Hinweise und Funktionen des Programms oder Einstellungen, welche Sie vornehmen können.



Übungen

Wenn Sie dieses Icon sehen, sind Sie an der Reihe. Hier geht es darum, das erworbene, theoretisches Wissen in die Praxis umzusetzen anhand von Fallbeispielen.



Infos

Diese Möglichkeit steht Ihnen nur in den Versionen Gold oder Platin zur Verfügung. Da der Makroassistent generell erst ab der Skalierung Gold zur Verfügung steht, wird in diesen Unterlagen weitestgehend auf dieses Icon verzichtet.

Nun wünschen wir Ihnen viel Spass und Erfolg in dieser Schulung und anschliessend beim Erkunden der Software und natürlich auch im täglichen Praxiseinsatz.

Wir weisen darauf hin, dass Sie nur dann optimal vom Kurs profitieren können, wenn Sie sich bereits mit der Datenbankstruktur von SelectLine auseinandergesetzt haben und grundlegende Kenntnisse in Transact-SQL haben.

Beachten Sie auch, dass alle Funktionen des Makroassistenten im „SelectLine System Handbuch“ entsprechend ausführlich detailliert geschildert werden. Die Kursunterlagen dienen lediglich als Ergänzung dazu. Für weitere Informationen empfehlen wir Ihnen daher, ebenfalls das Handbuch zu konsultieren. Zudem können Sie an nahezu jeder Stelle des Programms mit der Taste [F1] die Hilfe aufrufen. So werden Ihnen direkt zum aktuellen Programmpunkt weitere Informationen angezeigt. Eine Übersicht des Funktionsumfangs und der Abgrenzung zwischen den Skalierungen Standard, Gold und Platin entnehmen Sie der Leistungsübersicht, welche auf dem USB-Stick oder auf der Homepage zu finden ist. Weiter empfehlen wir Ihnen auch das **Neuerungsdocument** auf dem USB-Stick oder Homepage zu beachten.

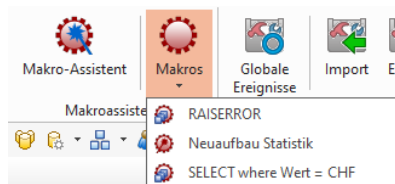
2 Struktur und Verwaltung des Makroassistenten

2.1 Überblick

Mit Hilfe des Makroassistenten können Abfragen aus den Datenbeständen des Programms erstellt werden. Dazu wird die genormte Datenbankabfragesprache Structured Query Language (SQL) verwendet. Mit Hilfe dieser Sprache ist es auch möglich, Daten direkt in der Datenbank zu ändern und zu löschen.

In älteren Programmversionen auf Basis der BDE (Borland Database Engine) steht nur ein eng begrenzter Funktionsumfang (Stichwort: Local SQL) zur Verfügung. Auch gibt es Unterschiede hinsichtlich der „Schreibweise“ von SQL-Anweisungen zur neueren **T-SQL**.

In den neueren Programmversionen auf Basis des MS-SQL-Servers kann der umfangreichere Funktionsbestand des Transact-SQL (**Kurzform: T-SQL**) verwendet werden. Eine wertvolle Informationsquelle bietet hierzu auch die Online-Hilfe des MS-SQL-Servers. Diese Schulungsunterlage wird nur die häufigsten Funktionalitäten beispielhaft aufgreifen.



Der Makro-Assistent wird über das Haupt-Menü „Eigene Daten“ gestartet. Zum Ausführen bereits erstellter Makros gibt es einen Menü-Punkt „Makros“.

Über die Passwortverwaltung kann der Makro-Assistent gesperrt werden. Bestimmte Makros können im Makro-Assistenten für den Menüpunkt „Makros“ freigegeben werden (gilt für alle Benutzer). In der Passwortverwaltung können wiederum „freigegebene“ Makros einzelnen Benutzer gesperrt werden, so dass jedem Benutzer sein individuelles „Makros“-Menü zugewiesen werden kann.

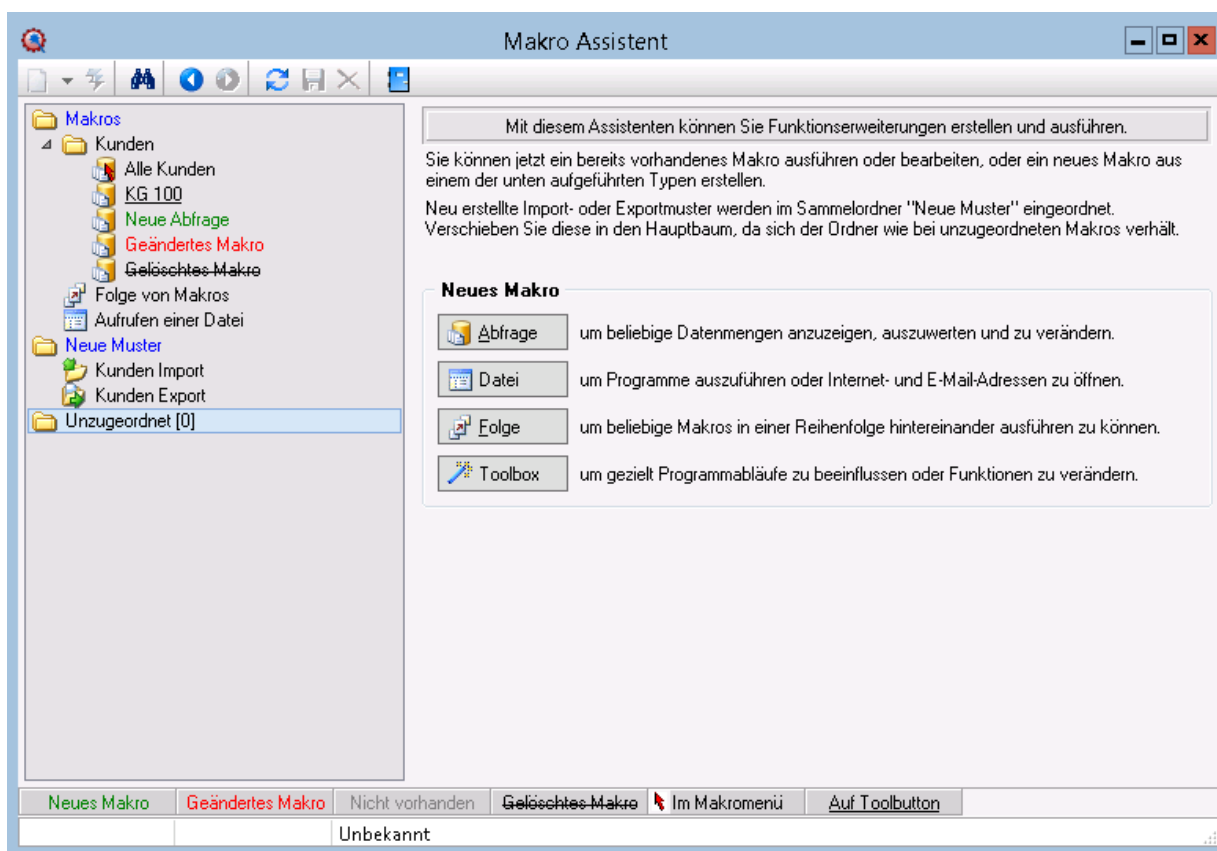


Der Makroassistent sollte für die Anwender nur stark eingeschränkt zur Verfügung gestellt werden! Über Makros ist es möglich die eingestellten Benutzerrechte aus der Passwortverwaltung von SelectLine zu umgehen, da direkt auf die Datenbank zugegriffen wird. Aus diesem Grund sollten den Anwendern nur ausgewählte Makros im Menü „Makros“ zur Verfügung gestellt werden.



Makros werden als .QDF-Dateien im Installationsverzeichnis von SelectLine im Ordner „Makro“ abgespeichert. Um Makros von einer anderen Installation zu übertragen empfehlen wir Ihnen allerdings die Funktion „Einschleusen“ / „Ausschleusen“ zu verwenden.

2.2 Bedienung



Die Eingabemaske des Makroassistenten ist in zwei Bereiche aufgeteilt. Im linken Maskenbereich werden in einer Baumstruktur alle vorhandenen Elemente aufgelistet, deren Typ und Status durch Symbol, Farbe bzw. Schrifttyp unterschiedlich dargestellt wird. Die Gestaltung der Maskenansicht auf der rechten Seite variiert entsprechend dem Typ des gewählten Elements. Die Breite des linken bzw. rechten Bereiches kann durch verschieben den „Mittelsteges“ angepasst werden.

Der Status eines Makros wird wie folgt dargestellt:

- Statusfarbe **grün** - aktiv neu erstelltes, noch nicht gespeichertes Element.
- Statusfarbe **rot** - aktiv geändertes, noch nicht gespeichertes Element.
- Statusfarbe **grau** - Makrodatei ist physisch im Makroverzeichnis nicht vorhanden.
- **durchgestrichen** - als gelöscht markiertes Element.
- **unterstrichen** - einem Toolbutton zugeordnetes Element.
- mit rotem Pfeil am Symbol - dem Makromenü zugeordnetes Element.



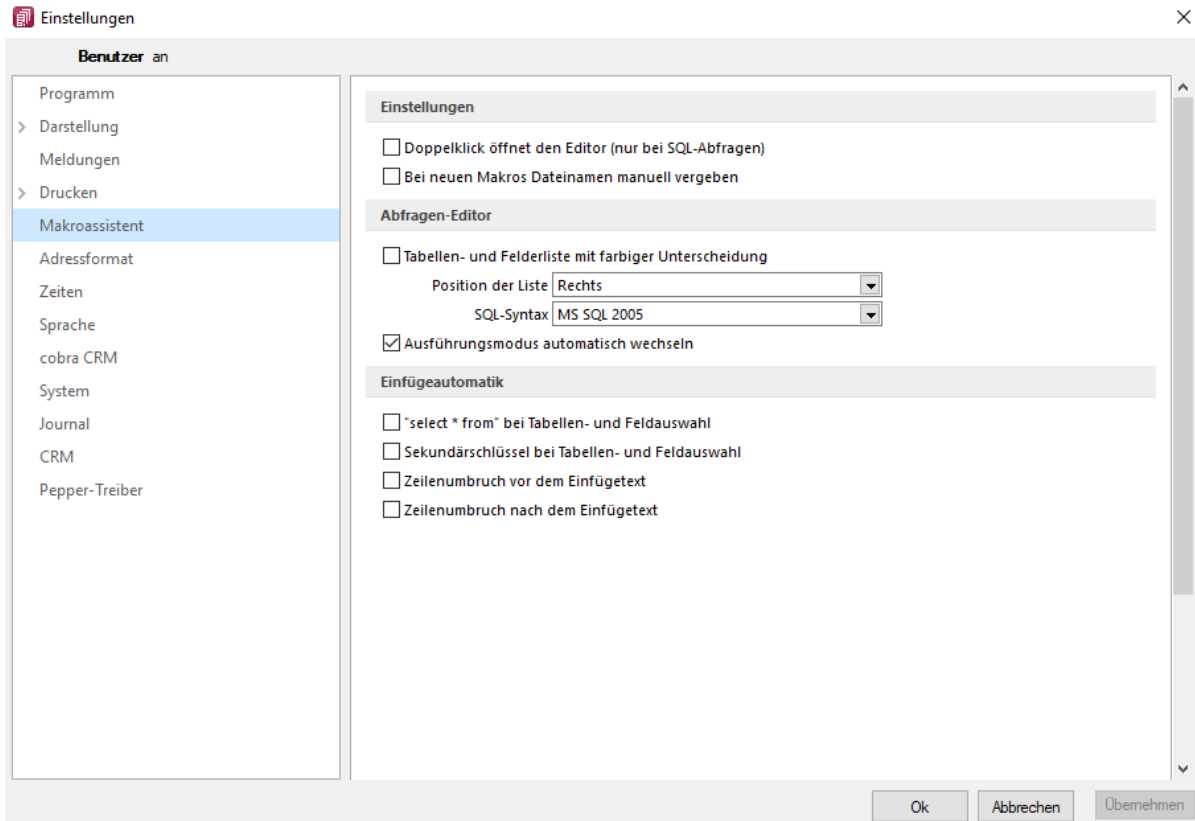
Am unteren Rand der Maske finden Sie eine Legende zur unterschiedlichen Darstellung der Einträge im linken Maskenbereich.



Der Ordner „Unzugeordnet“ enthält Makros, welche von Hand in den Makro-Ordner in das Installationsverzeichnis von SelectLine kopiert wurden, oder Elemente, welche nicht korrekt abgespeichert werden konnten. **Falls Sie befürchten ein Makro verloren zu haben, lohnt sich ein Blick in diesen Ordner.**

2.2.1 Benutzereinstellungen

Unter „Applikationsmenü/Einstellungen“ finden Sie einige Einstellungen, welche das Verhalten des Makroassistenten optimieren. Diese Einstellungen werden pro Benutzer gespeichert!



Makro-Assistent

- Ein Doppelklick auf einen Makro-Eintrag (nur bei SQL-Abfragen) öffnet den Makro-Editor, sonst Ausführen.
- Bei Neuanlage eines Makros wird ein Eingabedialog zum Dateinamen des Makros angezeigt.

Abfragen-Editor

Die Leiste mit der Anzeige der verfügbaren Tabellen und Tabellenfelder im Abfrage-Editor kann für die linke bzw. rechte Maskenseite eingestellt oder ganz ausgeblendet werden. Weiter kann für diese Leiste eine farbliche Markierung eingestellt werden, die kenntlich macht, in welchem Verzeichnis (Daten- oder Mandant) sich die Tabelle befindet bzw. um welchen Feldtyp (String, Float, ...) es sich handelt. Darüber hinaus wählen Sie die zu verwendende SQL-Syntax, d.h. nach welchem Dialekt die Farben in bestimmten Anweisungen dargestellt werden.

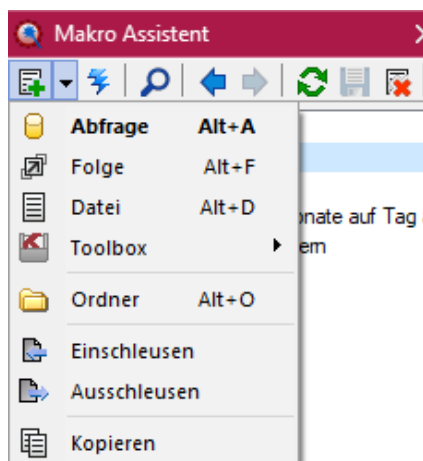
Mit gesetzter Option "Ausführungsmodus automatisch wechseln" wird die Makro-Aktion bei Eingabe der Befehlsanweisung automatisch angepasst. So wird z. B. automatisch auf die Aktion "Ausführen" gewechselt, wenn Befehlseingaben mit update, create, insert erfolgen.

Einfügeautomatik

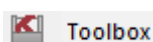
- Die Auswahl einer Tabelle bzw. eines Tabellenfeldes wird automatisch um die entsprechende "select"-Syntax ergänzt.
- Für die Auswahl wird zusätzlich der Sekundärschlüssel der Tabelle als "where"-Klausel eingefügt.
- Vor dem Einfügetext wird zusätzlich ein Zeilenumbruch eingefügt.
- Nach dem Einfügetext wird zusätzlich ein Zeilenumbruch eingefügt.

2.1.1 Die Funktionsleiste, Bedienelemente und Symbole des Makro-Assistenten

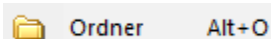
Über den Button  und das entsprechende Untermenü lassen sich nachfolgende Elemente anlegen.



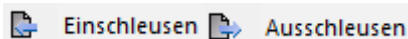
- **Abfrage Alt+A**
SQL-Anweisungen für Tabellenansichten, editierbare Datenmengen, Import- und Exportmuster, Druckauswertungen, Exporte.
- **Folge Alt+F**
Sie können sich beliebige Abfolgen, die aus Abfragen oder Dateikommandos bestehen, zusammenstellen, die dann vom Makro-Assistenten nacheinander ausgeführt werden.
- **Datei Alt+D**
Es werden Ausführungskommandos für Dateien oder Programme abgelegt. Sie können aus dem Makro-Assistenten heraus z.B. Excel- oder Worddokumente bearbeiten, sich Ordner auf Ihrer Festplatte anzeigen lassen oder beliebige Programme starten.



- Diese Funktion steht nur zur Verfügung, wenn die Toolbox lizenziert und für den entsprechenden Benutzer aktiviert ist. Hier werden Einstellungen zu Schnittstellen-Makros vorgenommen.



- Mit Hilfe von Ordnern können Sie sich eine Baumstruktur aufbauen, in der Sie Ihre Abfragen oder Folgen ablegen. Diese Struktur wird nicht physikalisch auf der Betriebssystem-Ebene erzeugt. Sie dient ausschliesslich der internen, übersichtlichen Verwaltung der abgelegten Elemente im Makroassistenten.



- Über diese Funktion ist es möglich Makros von einem in ein anderes System zu übertragen. Wir empfehlen für die Übertragung von Makros immer diese Funktion zu nutzen. Nur so ist gewährleistet, dass gedruckte Auswertungen auch komplett mit den Druckvorlagen ins Zielsystem übernommen werden. Ab der Version 18.3.x wird beim Einschleusen von Makros der aktuell ausgewählte Ordner zugeordnet.



- Alle Import- / Exportmuster, welche über die entsprechenden Assistenten unter „Schnittstellen/Assistenten“ angelegt werden, sind hier aufgelistet. Diese stehen auch für die Verwendung in Makrofolgen zur Verfügung.



- **Ausführen**
Das aktuelle ausgewählte Element wird über diesen Schalter ausgeführt. Alternativ kann dies auch mit einem Doppelklick oder [CTRL]+[Enter] bewirkt werden.



- **Makros Suchen**
Startet die Suche nach speziellen Elementen. Alternativ über [F3].



- **Makroliste neu einlesen**
Achtung nicht gespeicherte Änderungen werden allenfalls verworfen. Alternativ über [F5].



- **Löschen**
Ein gelöscht Element wird durchgestrichen dargestellt. Ordner müssen leer sein bevor sie gelöscht werden können. Als gelöscht markierte Elemente können wieder zurückgeholt werden, solange die Maske nicht geschlossen wurde. Alternativ [CTRL]+[DEL].



- **Beenden**
Beendet den Makro-Assistenten. Gibt es dabei noch nicht gespeicherte oder als gelöscht markierte Elemente, wird in der folgenden Maske zum Speichern/Verwerfen aufgefordert. Alternativ über [Esc].

2.1.2 Das Kontextmenü (rechte Maustaste) der Baumstruktur

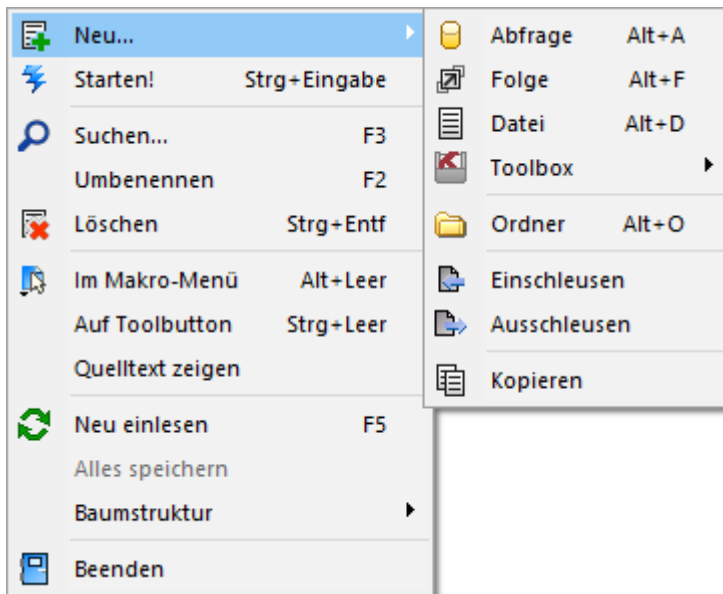
Im Kontextmenü der Baumstruktur finden Sie fast alle Funktionen, welche im vorhergehenden Kapitel beschrieben wurden. Zusätzlich sind jedoch noch folgende Funktionen verfügbar:

Umbenennen [F2]

Hier kann die Bezeichnung eines selektierten Elements geändert werden, nicht der Dateiname.

Quelltext anzeigen

Öffnet das selektierte Element mit dem Texteditor.



Alles speichern

Speichert über Vorauswahl in der folgenden Maske alle aktiv angelegten, geänderten bzw. gelöschten Elemente.

Im Makro-Menü [Alt] + [Leertaste]

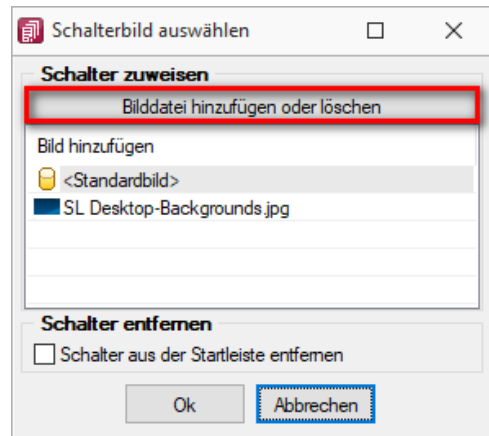
Hiermit haben Sie die Möglichkeit, einen Eintrag im Makromenü (Menüpunkt Makros) zu erzeugen. Somit sind oft benötigte Funktionen bequemer und schneller erreichbar. Wenn sich ein Eintrag im Menü befindet, wird sein Icon im Baum mit einem kleinen roten Mauspfel versehen. Die Struktur des Makromenüs und die im Baum des Makroassistenten sind identisch. Einem Ordner untergeordnete Einträge, werden im Makromenü auch als Untermenüpunkte angezeigt.

Auf Toolbutton [CTRL] + [Leertaste]

Eine weitere komfortable Möglichkeit, oft benötigte Abfragen oder Auswertungen schnell auszuführen, besteht darin, einen Schalter in der Schnellzugriffsleiste des Hauptfensters für das Makro zu erzeugen.

Es folgt die Maske zur Auswahl des Schalterbildes.

Mit entsprechenden Schalterbildern, können Sie die unterschiedlichen Makros besser unterscheiden. Die Reihenfolge der Schalter in der Funktionsleiste entspricht der Reihenfolge der Makros in der Baumstruktur. Die Schnellzugriffsleiste wird erst aktualisiert, wenn Sie den Makroassistent komplett schliessen.



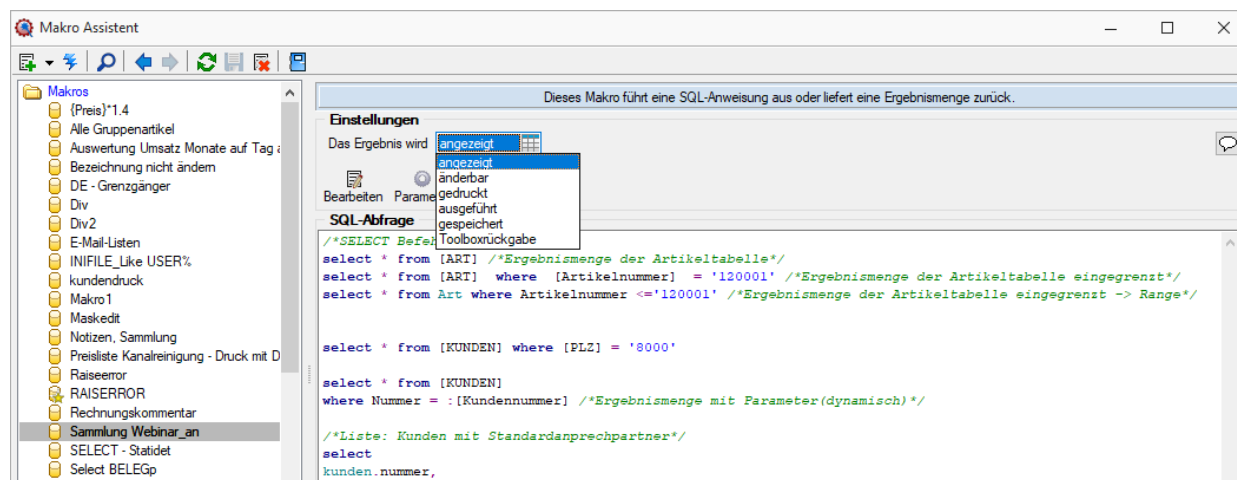
Baumstruktur

Die gesamte Strukturansicht der vorhandenen Elemente in der Liste kann nach Rückfrage verändert bzw. umsortiert werden:

- Alles aufklappen [CTRL] + [A] - zeigt zu allen Elemente die untergeordneten Einträge.
- Alles zuklappen [CTRL] + [Z] - zeigt nur die übergeordneten Elemente ohne Untereinträge.
- Gruppieren [CTRL] + [S] - sortiert alle Elemente alphabetisch nach Ordnern und Typen
- Alphabetisch - sortiert alle Elemente in alphabetischer Reihenfolge
- nach Datum - sortiert alle Elemente nach Datum

2.3 Der Makro Editierbereich

Im rechten Bereich des Makro-Assistenten wird das in der Baumstruktur markierte Makro bearbeitet.



2.3.1 Einstellungen

Unabhängig vom Inhalt und Umfang eines Makros kann gesteuert werden, was mit der „Ergebnismenge“ passiert.

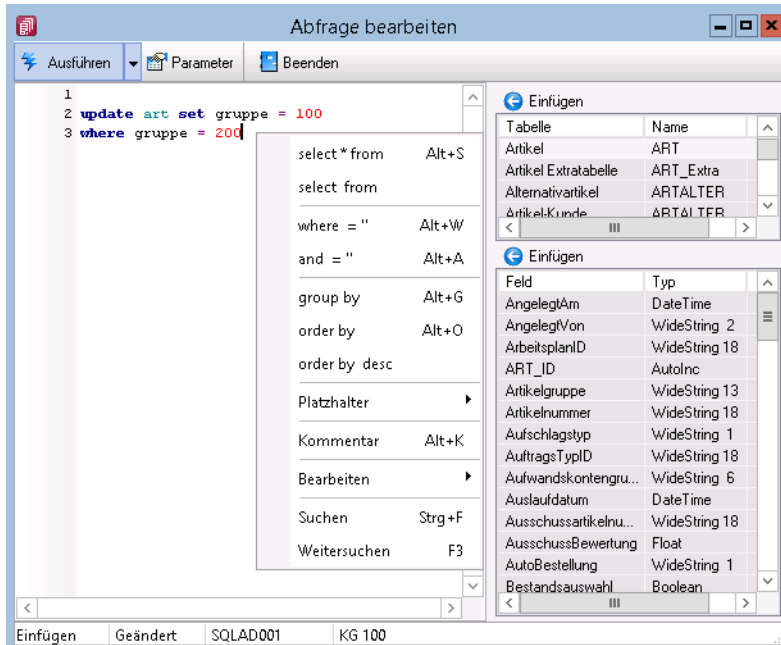
Das Ergebnis wird:

- **angezeigt**
Nach dem Betätigen des Schalters wird die Abfrage ausgeführt und die Ergebnismenge in einer Tabelle angezeigt. Diese Darstellung besitzt rein informativen Charakter (Lese-Modus).
- **änderbar**
Nach dem Betätigen des Schalters wird die Abfrage ausgeführt und die Ergebnismenge in einer Tabelle angezeigt. Die generierte Datenmenge wird im Edit-Modus zurückgeliefert (nicht bei komplexen (z. B. JOINS) Anweisungen möglich).
- **gedruckt**
Nach dem Ausführen der Abfrage erfolgt die Übergabe der Ergebnismenge an die Standarddruckausgabe. In der zuvor angezeigten Auswahl des Druckzieles sollten Sie beim ersten Mal auf jeden Fall den Bildschirm als Ziel auswählen, da die Druckvorlage zu Beginn leer ist und erst noch bearbeitet werden muss.
- **ausgeführt**
Wenn die Abfrage zur Modifikation von Daten dient (INSERT oder UPDATE statt SELECT), wird diese nach einer Bestätigung, deren Text man in der allgemeinen Bildschirmmaske angeben kann, ausgeführt (Standardtext ist: „SQL-Aktion wirklich ausführen?“). Hat diese Textzeile keinen Inhalt, wird der Bildschirmdialog unterdrückt. Anschliessend erfolgt die Meldung „SQL - Aktion ausgeführt“, wenn die Option „Meldung nach beendeter Ausführung zeigen“ in der Maske eingeschaltet ist (Standard ist ein).
- **gespeichert**
Hiermit haben Sie die Möglichkeit, die Ergebnismenge der Abfrage in externe Dateien zu exportieren. Um den Export zu konfigurieren wird der im Programm integrierte Exportassistent genutzt. Anschliessend kann der Export, ohne den Assistenten, aus dem SQL-Assistenten oder aus dem Makromenü heraus aufgerufen werden. Wenn Sie die Ergebnismenge einer neuen Abfrage erstmalig speichern wollen, wird der Exportassistent gestartet, da Angaben wie Dateiname- und Format noch nicht bekannt sind. Falls Sie zu einem späteren Zeitpunkt diese Angaben noch einmal ändern wollen, wird mit dem Schalter Exportmuster bearbeiten der Exportassistent mit dem Exportmuster zur Bearbeitung angezeigt.
- **Toolbox-Rückgabe**
Nur dann möglich, wenn die SelectLine-Toolbox lizenziert ist und mind. Ein Benutzer aktiviert ist.

2.3.2 Untermaske Bearbeiten



Die Syntax für ein Makro kann direkt im Langtext-Bereich unter „SQL-Abfrage“ eingegeben werden. Um weitere Hilfsmittel wie Tabellen- und Feldnamen für das Erstellen eines Makros zu erhalten, kann mit dem „Bearbeiten“-Schalter ein Unterdialog geöffnet werden.

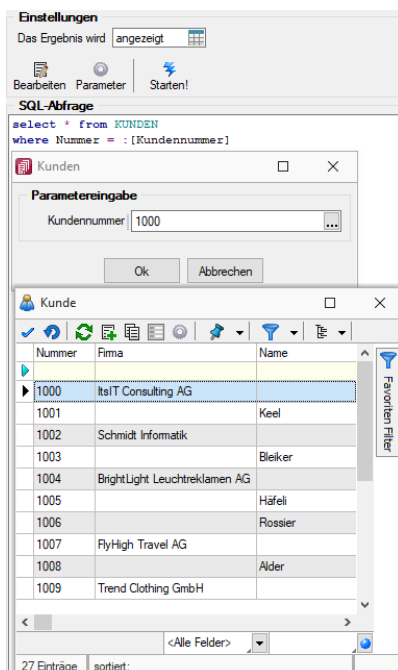


Im rechten Bereich werden zweigeteilt, oben alle Tabellen und unten die dazugehörigen Felder eingeblendet.

Über die Schalter **Einfügen** kann entweder der Tabellennamen oder der Feldname in die Makrosyntax im linken Bereich (rechts vom Cursor) übernommen werden. So wird die richtige Schreibweise von Tabelle und Feldname erleichtert.

Zusätzlich steht auch ein Kontextmenü (rechte Maustaste) im linken Bereich zur Auswahl, um häufig genutzte Syntax hinein zu kopieren.

2.3.3 Parameter



Mit Hilfe von sogenannten Parametern können Makros variabel gestaltet und Benutzereingaben beim Starten des Makros abgefordert werden.

Abfrageparameter für dynamische SQL-Anweisungen werden mit einem Doppelpunkt gekennzeichnet. Wird die Abfrage gestartet, erscheint ein Dialog in welchem der variable Parameter eingegeben werden kann.

In diesem Beispiel ist der variable Parameter die Kundennummer.

Ab der Version 16.3 kann es notwendig sein, den Parameter in eckige Klammern zu setzen, beispielsweise:

bis 16.2.x

: 'Datum zum Beginn'

bis 16.3.0

: [Datum zum Beginn]



Mit diesem Schalter können Sie alle vorhandenen Parameter in einem Makro bearbeiten. Folgende Einstellungen sind dabei möglich:

Parameter	Datentyp	Auswahlfeld	Auswahl aktiv	Letzter Wert	Vorgabewert
Kundennummer	WideString	Kunde	<input checked="" type="checkbox"/>	1000	
Kundengruppe	WideString	Kundengruppe	<input checked="" type="checkbox"/>		
Datum	Date		<input type="checkbox"/>		

Datentyp

Wählen Sie den passenden Datentyp für den definierten Parameter.

Auswahlfeld

Hier können Sie eine Tabelle als Auswahlfeld definieren. So können Sie den Hauptschlüssel der ausgewählten Tabelle als Parameter übernehmen.

Vorgabewert

Der Parameter wird immer mit dem hier erfassten Wert voreingetragen

Letzter Wert

Zeigt den zuletzt verwendeten Wert an.

Hinweistext

Hier können Sie einen Hinweistext für den Anwender erfassen. Dieser Text wird dem Anwender als Tooltipp angezeigt (mit der Maus auf das Eingabefeld des Parameters zeigen).

Parameterautomatik

Für die Parameterübernahme aus dem aktiven Dialog muss die Parameterautomatik definiert werden. Bei den Parametern einer SQL-Anweisung kann man einstellen, ob:

- die Parameter vor dem Starten immer abgefragt werden.
- im aktiven Stammdialog betroffene Tabellenfelder gesucht und als Vorschlag für die Parameter angeboten werden.
- im aktiven Stammdialog betroffene Tabellenfelder gesucht werden, falls alle Parameter gefunden wurden, die Anweisung ohne Parameterkontrolle gestartet wird.
- in der aktiven Datenmenge betroffene Tabellenfelder gesucht und als Vorschlag für die Parameter angeboten werden.
- in der aktiven Datenmenge betroffene Tabellenfelder gesucht werden, falls alle Parameter gefunden wurden, die Anweisung ohne Parameterkontrolle gestartet wird.

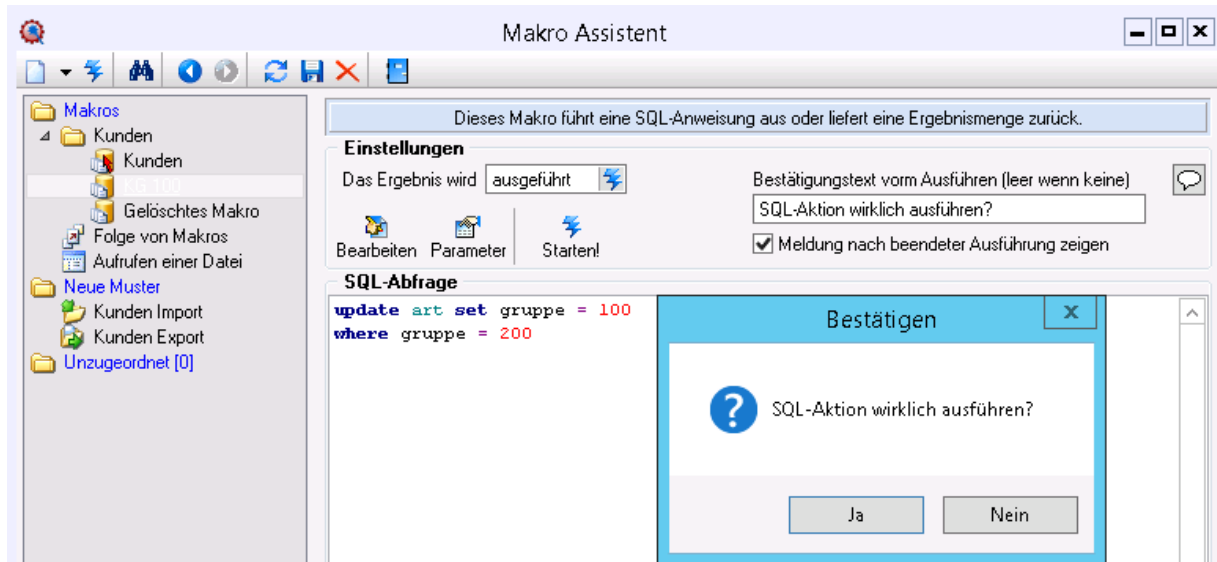


Erfassen Sie das unten aufgeschriebene Makro und legen Sie es auf den Toolbutton. Anschliessend öffnen Sie die Kundenmaske und führen Sie das Makro aus der Schnellzugriffsliste aus. Testen Sie die verschiedenen Einstellungen für die Parameterautomatik.

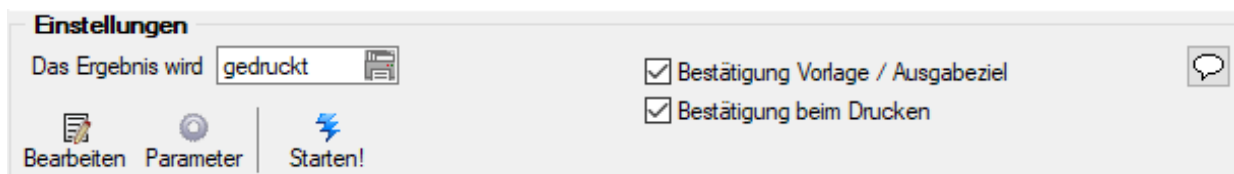
`select nummer, anzeigename, plz, ort, gruppe from Kunden where gruppe = :gruppe`

2.3.4 Bestätigungsdialoge für „ausgeführt“, „gedruckt“, „gespeichert“

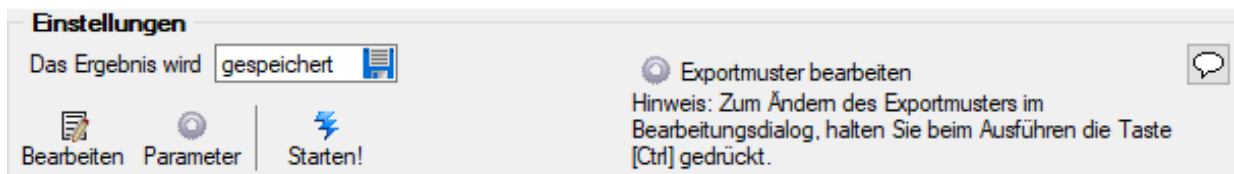
Bei einer Abfrage mit dem Typ „ausgeführt“ können Sie vor der Ausführung einen Bestätigungstext anzeigen lassen. Die Aktion wird erst nach der Bestätigung ausgeführt. (Ist der Bestätigungstext leer und die Checkbox für die Ausführungsmeldung inaktiv, wird das SQL-Statement dialogfrei abgearbeitet).



Bei einer Abfrage mit dem Typ „gedruckt“ können Sie ebenfalls wählen ob zuerst der Bestätigungsdialog angezeigt werden soll oder der Druck mit Ausführung der Abfrage gleich ohne Dialog erfolgen soll.



Bei der ersten Ausführung einer Abfrage vom Typ „gespeichert“ erhalten Sie den gleichen Dialog wie wenn der Exportassistent ausgeführt wird und die Ergebnismenge kann in eines der vordefinierten Formate gespeichert werden.



2.3.5 Anlegen einer Folge von Makros

Das Anlegen einer Makrofolge ermöglicht Ihnen mehrere Makros nacheinander ausführen zu lassen. Dies ermöglicht Ihnen beispielsweise in einem ersten Makro eine neue Tabelle zu erstellen und diese mit

Daten zu füllen. Im zweiten Makro werden die abgefüllten Daten ausgedruckt. Dazu verwenden wir folgendes Beispiel:

Makro 1 – Aktualisieren der Bankverbindungen

UPDATE [BANKVERB]

SET kasebank = : [Banbezug neu]

WHERE blobkey = 'KU' -- Nur Kunden

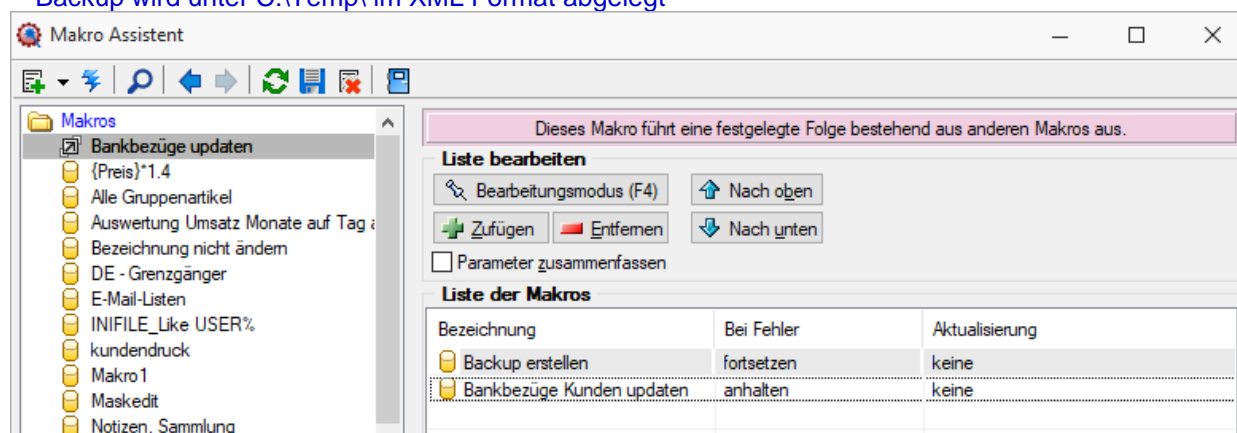
AND kasebank = : [Banbezug Alt] – falls mehrere Bankbezüge (je Währung) vorhanden sind

Makro 2 – Backup

SELECT * FROM bankverb

Where blobkey = 'KU'

-- Backup wird unter C:\Temp\ im XML Format abgelegt



Bearbeitungsmodus (F4)

Wenn Sie den Bearbeitungsmodus einer Makrofolge aktivieren, können Sie Makros aus dem linken Maskenbereich per Drag&Drop in die Folge ziehen. Durch erneuten Klick auf den Schalter wird der Bearbeitungsmodus deaktiviert

Nach oben Nach unten

Verschieben Sie Makros in der Reihenfolge nach oben oder unten.

Zufügen Entfernen

Verwenden Sie diese Schalter um ein Makro hinzuzufügen oder zu entfernen.

Parameter zusammenfassen

Sofern in den einzelnen Makros der Folge Parameter verwendet wurden, so werden diese zusammengefasst, wenn sie identisch sind. Ansonsten werden die Parameter jedes Makros einzeln abgefragt.

Makro Details

- Für einzelne Makros der Folge können Sie einstellen, ob die Folge im Fehlerfall abgebrochen oder fortgesetzt werden soll.
- Für die Abfragen können Sie die Aktualisierung von Formular oder Datenquelle einstellen.
- Wir empfehlen unter "Programmeinstellungen/Makroassistent" die Einstellung "Ausführungsmodus automatisch wechseln" zu deaktivieren, mit gesetzter Option wird die Makro-Aktion bei Eingabe der Befehlsanweisung automatisch angepasst. So wird z.B. automatisch auf die Aktion Ausführen gewechselt, wenn Befehlseingaben mit update, create, insert erfolgen (Testingmodus).

3 Datenbankabfragesprache „Transact SQL“

In SelectLine kommt die international gebräuchliche „Structured Query Language“ (SQL) zum Einsatz. Je nach Datenbank gibt es entsprechende „Feinheiten“, doch sind die Haupt-Befehle identisch. Die von Microsoft im SQL-Server verwendete Abfragesprache wird als T-SQL (Transact-SQL) bezeichnet.

3.1 Anweisungen

Data Manipulation Language- (DML-) Anweisungen:

Anweisung	Beschreibung / Syntax / Beispiel
SELECT Felder aus einer Tabelle: Felder aus zwei Tabellen über qualifizierte Bezeichner:	Listet bestehende Daten aus einer Tabelle auf. select [Feld1], [Feld2] from [Tabelle] select nummer, Anzeigename from kunden select Kunden.Nummer, Kunden.Name, Liefer.Nummer, Liefer.Name from kunden, liefer oder mit Alias select k.Nummer, k.Name, l.Nummer, l.Name from kunden k, liefer l
DELETE Beispiel Löscht Kunden, welche nicht in einem Beleg verwendet wurden.	Löscht bestehende Datensätze aus einer Tabelle. delete from [Tabelle] where [Feld] = [Wert] delete from kunden where kunden.nummer not in (select adressnummer from beleg)
INSERT Beispiel Legt einen neuen Kunden mit der Nummer 9999 und Name „Test“ an.	Fügt einer Tabelle neue Daten hinzu. insert into [Tabelle] values ([Wert1], [Wert2]) oder mit spezifischen Feldern, welche zu füllen sind insert into [Tabelle] ([Feld1], [Feld2]) values ([Wert1], [Wert2]) insert into kunden (nummer, Name) values (9999, 'Test')
UPDATE Beispiel Setz bei allen Kunden der Gruppe „100“ die Gruppe „200“.	Verändert bestehende Daten in einer Tabelle update [Tabelle] set [Feld1] = [Wert1], [Feld2] = [Wert2] update kunden set gruppe = 200 where gruppe = 100

3.2 Klauseln

Klauseln sind Bedingungen und Einschränkungen für SQL-Anweisungen.

Hier die am häufigsten verwendeten:

Klausel	Beschreibung / Syntax / Beispiel
FROM	Gibt die für die Anweisung verwendeten Tabellen an.
WHERE	Gibt Filterkriterien zum Begrenzen der abgerufenen Zeilen an.
ORDER BY Beispiel Sortiert die Kundentabelle nach dem Feld ‚Gruppe‘ absteigend. Für eine aufsteigende Sortierung ist desc wegzulassen.	Gibt die Spalte an, anhand deren die Ergebnismengen zu sortieren ist. <code>select * from kunden order by gruppe desc</code>
GROUP BY Beispiel Zeigt alle Kunden, welche in den Belegen verwendet wurden. Gleiche nummern werden zusammengefasst.	Gibt die Spalten an, die zum Gruppieren von Zeilen verwendet werden. <code>select adressnummer from beleg group by adressnummer</code>
HAVING Beispiel Zeigt alle Artikel, welche in mehr als zwei Stücklisten verwendet werden.	Gibt Filterkriterien an, die zum Aggregieren (z. B. Gruppieren) von Daten verwendet werden. <code>select setartikelnummer from artset group by setartikelnummer having count (setartikelnummer) > 2</code> <code>select * from ARTLIEF where bestellnummer in (select bestellnummer from ARTLIEF group by Bestellnummer having count (Bestellnummer)>1)</code>

3.3 Funktionen

3.3.1 Stringfunktionen

Eine Auswahl der am häufigsten verwendeten Stringfunktionen:

String-Funktionen	Beschreibung / Syntax / Beispiel
CONCAT Beispiel Verbindet die Felder Name und Vorname jeweils mit einem Leerzeichen.	Verbindet mehrere Strings zu einem <code>select concat ([String1], [String2]) as [Spaltenname]</code> <code>select concat(name, ' ', vorname) as NameVorname from kunden</code>
UPPER Beispiel Beispiel von oben in Grossbuchstaben.	Wandelt einen String in Grossbuchstaben um. <code>select upper ([String])</code> <code>select upper(concat(name, ' ', vorname)) as NameVorname from kunden</code>
LOWER Beispiel Beispiel von oben in Kleinbuchstaben	Wandelt einen String in Kleinbuchstaben um <code>select lower ([String])</code> <code>select lower(concat(name, ' ', vorname)) as NameVorname from kunden</code>
SUBSTRING Beispiel Extrahiert nur die Belegnummer aus dem Feld „Adresstyp“.	Extrahiert einen Teil eines Stringwertes <code>select substring ([String], [Startzeichen], [Länge])</code> <code>select substring(adresstyp, 3, 18) from adress as Belegnummer</code>
TRIM, LTRIM, RTRIM Beispiel Entfernt die Leerschläge links.	Entfernt Leerschläge links oder rechts eines Strings <code>select rtrim([String])</code> <code>select ltrim(' Dies ist ein Test')</code>

3.3.2 Aggregatfunktion

Eine Auswahl der am häufigsten verwendeten Aggregatsfunktionen.

Aggregatfunktion	Beschreibung / Syntax / Beispiel
AVG Beispiel Durchschnittlicher Umsatz des Kunden 1000.	Erzeugt den Mittelwert aller numerischen Spaltenwerte, die ungleich Null sind. <code>select avg([Zahl]) from [Tabelle]</code> <code>select avg(umsatz) from statidet where adressnummer = '1000' and EA = 'A'</code>
COUNT Beispiel Anzahl der Kunden.	Zählt die Anzahl der Zeilen in einer Ergebnismenge. <code>select count([Feld]) from [Tabelle]</code> <code>select count(nummer) from kunden</code>
MAX Beispiel Alphabetisch der letzte Kunde.	Ermittelt den maximalen Wert in einer Spalte (auch für String). <code>select max([Feld]) from [Tabelle]</code> <code>select max(Anzeigenname) from kunden</code>
MIN Beispiel Kleinsten Umsatz.	Ermittelt den minimalen Wert in einer Spalte (auch für String). <code>select min([Feld]) from [Tabelle]</code> <code>select min(umsatz) from statidet</code>
SUM Beispiel Summe aller Umsätze der Kunden im Jahr 2022	Summiert alle numerischen Werte in einer Spalte. <code>select sum([Zahl]) from [Tabelle]</code> <code>select nummer, sum(AUmsatz) as '2022' from statisum where bk = 'KU' and Jahr = '2022' group by nummer</code>

3.3.3 Datenfunktion

Eine Auswahl von Datenfunktionen:

Datenfunktion	Beschreibung / Syntax / Beispiel
CAST / CONVERT Beispiel Umwandlung einer Zahl in einen String.	Wandelt Werte von einem Datentyp in einen anderen um. <code>select cast([Wert] as [Datentyp] ([Länge]))</code> <code>select cast(1000 as char(4))</code>
DAY / MONTH / YEAR Beispiel Monat aus Datum.	Extrahiert Jahres-, Monats- oder Tagesfeld eines Datums. <code>select year([Datum])</code> <code>select month('21.03.2022')</code> <code>select Nummer, Ort, year(AngelegtAm) as Jahr from Kunden</code>

3.4 Operatoren

Typ	Operator
Arithmetisch	+, -, *, /
Logisch	AND, OR, NOT

3.5 Prädikate

Prädikat	Beschreibung
BETWEEN	Vergleicht einen Wert mit einem durch zwei Werte gebildeten Bereich.
EXISTS	Bestimmt, ob ein Wert in einer Nachschlagetabelle vorhanden ist und gibt einen Wert vom Typ boolean zurück.
IN	Bestimmt, ob ein Wert in einer Werteliste oder in einer Tabelle vorhanden ist.
LIKE	Vergleicht einen Wert mit einem anderen (ganz oder teilweise).
IS NULL	Vergleicht einen Wert mit einem leeren oder NULL-Wert.
SOME / ANY / ALL	Führt quantifizierte Vergleiche durch.

3.6 Datendefinition

T- SQL unterstützt Data Definition Language (DDL) zum Erstellen, Verändern und Löschen von Tabellen und Indizes.

DDL-Anweisung	Beschreibung / Syntax / Beispiel
CREATE TABLE Beispiel Erstellt eine neue Tabelle mit Feldern von folgendem Typ: String, boolean, Datum	Legt eine neue Tabelle an <code>create table [Tabellenname] ([Feld1] [Datentyp] ([Grösse]), [Feld2] [Datentyp] ([Grösse]))</code> <code>create table TestTabelle (feld1 char(20), feld2 bit, feld3 datetime,)</code>
ALTER TABLE ADD / ALTER COLUMN Beispiel Feld hinzufügen Feld verändern Feld löschen	Fügt einer bestehenden Tabelle Spalten hinzu, löscht Spalten raus oder modifiziert diese. <code>alter table TestTabelle add neuesfeld nvarchar (10)</code> <code>alter table TestTabelle alter column neuesfeld bit</code> <code>alter table TestTabelle drop column neuesfeld</code>
DROP TABLE Beispiel Löscht die Tabelle TestTabelle.	Löscht eine bestehende Tabelle. <code>drop table TestTabelle</code>
CREATE INDEX Beispiel Neuer Index auf das Feld1	Erstellt einen neuen Sekundärindex für eine bestehende Tabelle. Indizes beschleunigen das Suchen nach Daten in einer Tabelle. Allerdings dauert es länger Daten anzupassen in einer Tabelle mit vielen Indizes! <code>create index idx1 on TestTabelle (feld1)</code>
DROP INDEX Beispiel Löscht den Index idx1.	Löscht einen bestehenden Primär- oder Sekundärindex. <code>drop index idx1 on TestTabelle</code>

3.7 Besonderheiten der Datenbankabfragesprache T-SQL

- **Kommentare**

Für das Auskommentieren innerhalb von SQL-Statements ist die Syntax `/*Kommentar*/` zu verwenden. Zudem steht für das auskommentieren einer einzelnen Zeile die Syntax `--Kommentar` zur Verfügung.

- **Datentyp String**

Für das Kennzeichnen von Datenfeldern vom Typ „String“ sind keine doppelten Hochkommas (Gänsefüßchen) zulässig, es müssen einfache Hochkommas verwendet werden, z. B.:

```
select * from art where artikelnummer = '100001'
```

- **Datentyp Boolean**

Dieser Feldtyp kann auch im SQL-Server nur zwei Werte besitzen. Aus dem 'False' wird '0' und aus 'True' wird '1'. Als einzige Abweichung dazu kann ein solches Feld auch keinen Wert (null) besitzen.

- **Anlegen von Feldern vom Typ „char“ (widestring)**

Der SQL-Server erwartet für neu anzulegende Datenfelder vom Typ "char" zwingend die entsprechenden Zeichensatzinformationen zur korrekten Anlage dieser Felder. Diese müssen natürlich dieselben sein, wie sie das Quellfeld mitbringt.

```
create table mengetmp (menge float, mengeneinheit char(10) COLLATE Latin1_General_CS_AS)
```

- **UPDATE-Anweisung**

Die Qualifizierung der Alias-Namen erfolgt im Transact-SQL in abgewandelter Form, z.B.:

```
update arpreis set preis = k.FreieZahl1 from arpreis a, kunden k
where a.artikelnummer = '100001' and a.preistyp = 'K'
and a.Typnummer = k.nummer
```

- **Verwendung von dateibasierenden Datenbanktabellen**

Für die Verwendung von Informationen aus dBase- oder Paradox-Dateien muss nachfolgende Syntax benutzt werden:

Paradox

```
select * from opendatasource
```

```
('Microsoft.Jet.OLEDB.4.0', 'Data Source=C:\Temp\Extended properties=PARADOX 4.X') ...Extra
```

Dieses Makro öffnet die Tabelle Extra.db im Pfad c:\temp.

dBase

```
select * from opendatasource
```

```
('Microsoft.Jet.OLEDB.4.0', 'Data Source=c:\temp\Extended properties=dBASE IV') ...Extra
```

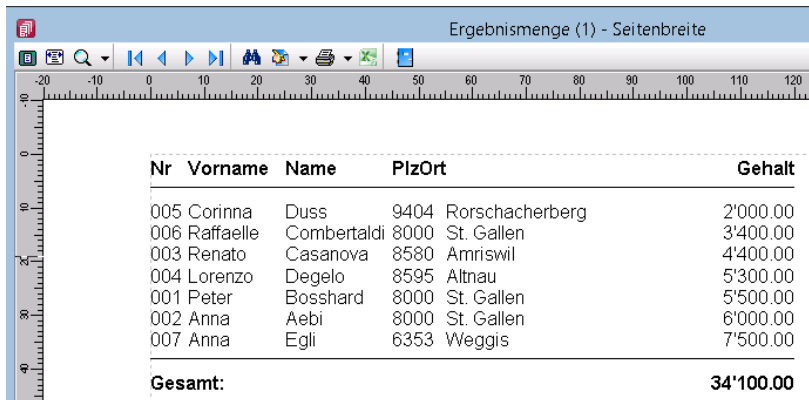
Dieses Makro öffnet die Tabelle Extra.dbf im Pfad c:\temp.

Dabei ist zu beachten, dass das "c:\temp\" das lokale Temp-Verzeichnis des Rechners spezifiziert, auf dem der SQL-Serverdienst läuft.

3.8 Formularentwicklung für gedruckte Ergebnismengen

Nach dem Ausführen der Aktion „Drucken“ einer Abfrage erfolgt die Ausgabe auf Standarddruckausgabe (siehe Einstellungen). In der zuvor angezeigten Auswahl des Druckzieles sollten Sie beim ersten Mal auf jeden Fall den Bildschirm als Ziel auswählen, da die Druckvorlage zum Ausdruck der Ergebnismenge erst noch bearbeitet werden muss.

Beispiel:



Nr	Vorname	Name	PlzOrt	Gehalt
005	Corinna	Duss	9404 Rorschacherberg	2'000.00
006	Raffaele	Combataldi	8000 St. Gallen	3'400.00
003	Renato	Casanova	8580 Amriswil	4'400.00
004	Lorenzo	Degelo	8595 Altnau	5'300.00
001	Peter	Bosshard	8000 St. Gallen	5'500.00
002	Anna	Aebi	8000 St. Gallen	6'000.00
007	Anna	Egli	6353 Weggis	7'500.00
Gesamt:				34'100.00


Im Feld FreieZahl1 jedes Mitarbeiters kann bspw. das Gehalt hinterlegt werden. Eine Gehaltsliste mit allen Mitarbeitern soll, nach Gehalt sortiert, per SQL erstellt und gedruckt werden.

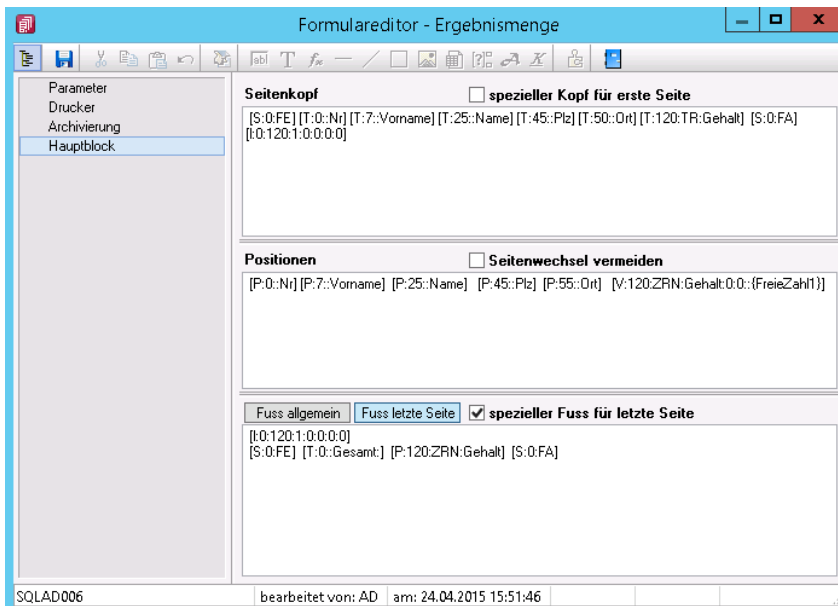
```
select * from mitarbw
order by freiezahl1
```

```
select * from mitarbw
order by freiezahl1 desc
```

(oder asc)

Aktion „Drucken“ druckt als Bildschirmvorschau eine leere Seite.

 Druckvorlage bearbeiten – der Formulareditor wird geöffnet.



Die Listenüberschrift wird über Text-platzhalter, Schriftstil und Linie im Seitenkopf definiert.

Alle Felder der Ergebnismenge stehen im Positionsbereich als Platzhalter Datenfeld unter Ergebnismenge zur Verfügung.

Die Linie im Seitenfuß schliesst den Positionsbereich ab.

Alle numerischen Datenfelder weisen im Fußbereich die Summe aus.

Damit die Summenzeile nicht am unteren Seitenrand sondern unter der letzten Position erscheint, ist auf der Seite „Parameter“ die Option „Seitenfuss heranziehen“ zu aktivieren.

The screenshot shows the 'Formulareditor - Ergebnismenge' window. The 'Parameter' tab is active. The 'Druckziel' section has 'Standardausgabe' selected. The 'Ränder' section shows 'Links' at 20 mm, 'Oben' at 10 mm, 'Rechts' at 10 mm, and 'Unten' at 10 mm. The 'Zeilenabstand' section has 'Fix' selected with a value of 40 and a unit of 1/10 mm. The 'Exemplare' section has 'Anzahl' set to 1. The 'Spaltendruck' section has 'Spaltenanzahl' set to 1. The 'Seitenfuss' section has the 'heranziehen' checkbox checked. The 'Etikett' section has 'Nummer' set to '...' and 'Block' set to 'Hauptblock'. The 'Separatoren' section has 'Dezimaltrennzeichen' checked. The status bar at the bottom shows 'SQLAD006' and 'bearbeitet von: AD am: 24.04.2015 15:51:46'.

Seit der Version 13 sind alle Druckvorlagen nun in der SQL-Datenbank hinterlegt. Auch der „Druckanteil“ der QDF-Dateien wird nun in der SQL-Datenbank gespeichert.



Es ist nun wichtig, ein Makro, welches in eine andere Installation übertragen werden soll, über die Funktion Ausschleusen/Einschleusen zu transferieren. Andernfalls fehlt der „Druckanteil“ komplett.

3.9 Datenbankabhängige Unterschiede in der SQL-Syntax

Ausgewählte String-Funktionen

Local-SQL	Advantage Streamline SQL	Transact SQL
	ASCII(str)	ASCII(str)
	CHAR(int)	CHAR(int)
	LEFT(str, anzahl)	LEFT(str, anzahl)
	RIGHT(str, anzahl)	RIGHT(str, anzahl)
	REPLACE(str1, str2, str3)	REPLACE(str1, str2, str3)
	SPACE(anzahl)	SPACE(anzahl)
	LENGTH(str)	LEN(str)
	LOCATE(str1, str2 [,start])	CHARINDEX(str1, str2 [,start])
TRIM(LEADING " " FROM str)	LTRIM(str)	LTRIM(str)
TRIM(TRAILING " " FROM str)	RTRIM(str)	RTRIM(str)
SUBSTRING(str FROM start FOR anzahl)	SUBSTRING(str, start, anzahl)	SUBSTRING(str, start, anzahl)

Überall funktioniert:

+

LOWER(str)

UPPER(str)

Ausgewählte mathematische Funktionen

Local-SQL	Advantage Streamline SQL	Transact SQL
siehe *1)	ABS(str)	ABS(str)
siehe *2)	CEILING(float)	CEILING(float)
siehe *2)	FLOOR(float)	FLOOR(float)
	ROUND(float, anzahl)	ROUND(float, anzahl)
siehe *3)	MOD(int1, int2)	int1 % int2

Überall funktioniert:

+, -, *, /

*1) Nachbildung von ABS unter Local-SQL

cast(trim(leading '-' from cast(freiezahl2 as char(10))) as float)

*2) Nachbildung von FLOOR bzw. CEILING unter Local-SQL

cast(cast(freiezahl2 as int) as float)

wirkt bei positiven Zahlen wie FLOOR, bei negativen wie CEILING

*3) Nachbildung von MOD unter Local-SQL

int1 – (int1 / int2) * int2

Ausgewählte Datumsfunktionen

Local-SQL	Advantage Streamline SQL	Transact SQL
EXTRACT(YEAR FROM datum)	YEAR(datum)	YEAR(datum)
EXTRACT(MONTH FROM datum)	MONTH(datum)	MONTH(datum)
EXTRACT(DAY FROM datum)	DAYOFMONTH(datum)	DAY(datum)
	CURDATE()	
	NOW()	GETDATE()
	DAYOFYEAR(datum)	DATEPART(DY, datum)
	DAYOFWEEK(datum) (1=Sonntag ... 7 = Sonnabend)	DATEPART(DW, datum) (1 = Montag ... 7 = Sonntag)

Sonstige Funktionen:

Local-SQL	Advantage Streamline SQL	Transact SQL
Kommentare mit /* und */	Kommentare mit /* und */ oder //	Kommentare mit /* und */ oder --
CAST(ausdruck AS FLOAT)	CONVERT(ausdruck, SQL_DOUBLE)	CAST(ausdruck AS FLOAT) oder auch CONVERT(FLOAT, 100)
	IFNULL(ausdruck, wert)	ISNULL(ausdruck, wert)
		ISNUMERIC(ausdruck)
siehe *4)	IIF(log.ausdruck, truewert, falsewert)	
	Einfache Case-Anweisung CASE ausdruck WHEN fall1 THEN wert1 ... ELSE wertn END Komplexe Case-Anweisung CASE WHEN log.ausdruck1 THEN wert1 ... ELSE wertn END	Einfache Case-Anweisung CASE ausdruck WHEN fall1 THEN wert1 ... ELSE wertn END Komplexe Case-Anweisung CASE WHEN log.ausdruck1 THEN wert1 ... ELSE wertn END

*4) Umwandlung in einen Faktor 0 oder 1 zur Nachbildung von IIF unter Local-SQL

IIF(monat = 8, 1, 0) kann man ersetzen durch 1 - cast(cast(monat-8 as boolean) as integer)

4 Anhang

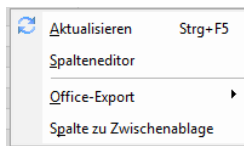
4.1 Glossar

Alphanumerik:

Es ist wichtig, dass Sie sich bei der Erfassung von Stammdaten oder Belegen mit der Alphanumerik auseinandersetzen. Dies kann sich auf die Sortierung, Darstellung und Auswertung der Daten weiterführend auswirken.

Machen Sie sich Gedanken über die ungefähre Anzahl an Stammdaten und Belegen, die als Anzahl Stellen (inkl. führenden Nullen) definiert werden. Dies bedeutet konkret, wenn Sie etwa 1'000 Kunden haben, beginnen Sie mit der Kundennummer 1001 oder 0001. Die Daten werden ansonsten immer nach der vordersten Zahl gegliedert, wie z. B. folgendermassen:
1,10,11,...,100,101,...,2,20,21,...,etc.

Spalteneditor:



In allen Tabellenansichten haben Sie die Möglichkeit, diese auf Ihre individuellen Bedürfnisse anzupassen. Dies ist auf verschiedene Arten möglich: einerseits kommen Sie in den „Spalteneditor“ indem Sie in der Tabellenansicht in der Tabelle über das Kontextmenü der rechten Maustaste klicken und anschliessend die Spaltenüberschriften mit der linken Maustaste verschieben.

Andererseits können Sie auch in der Tabelle selber die Spalten mit der rechten Maustaste, in der Kopfzeile, an die gewünschte Position verschieben.

Quickfilter:



Den Quickfilter finden Sie in den meisten Fenstern des Programmes. Durch diesen ist es möglich, im geöffneten Fenster nach einem gewünschten Datensatz zu suchen. Es kann in allen Feldern oder nur in einer gewünschten Spalte gesucht werden. Der Kreis ganz rechts ändert die Farbe von **blau** zu **rot** wenn er aktiviert ist. Sie sehen dann nur die Auswahl gemäss Ihren Suchkriterien.

Icons:

	Wechseln zum ersten Eintrag der Datenbank		Löschen eines Datensatzes
	Einen Datensatz Rückwärts blättern		Wechseln in die Tabellenansicht
	Wechseln zum letzten Eintrag der Datenbank		Wechseln in den Bearbeitungsmodus
	Einen Datensatz Vorwärts blättern		Marke setzen, ▽ = zum markierten Artikel
	Historie-Button (die letzten 10 Datensätze)		Filter setzen, ▽ = neuen Filter anlegen
	Neuen Datensatz erstellen		Einstellungen und Zusatzfunktionen
	Kopie eines bestehenden Datensatzes anlegen		Einzeldruck, Listendruck
	Speichern		Eingabemaske schliessen und speichern
	Verwerfen einer Eingabe		

4.2 Dank

Vielen Dank für Ihre Aufmerksamkeit und herzliche Gratulation zur erfolgreichen Teilnahme an diesem Kurs. Wir wünschen Ihnen viel Spass und Erfolg beim Umsetzen in Ihrem Geschäftsalltag. Wenn nur einige Punkte dabei waren, die Sie für sich mitnehmen und anwenden können und sich damit Ihr Alltag etwas vereinfacht, ist dies schon einiges an Profit, den Sie gewonnen haben. Denn Zeit ist und bleibt eine der knappsten Ressourcen, die wir haben und diese gilt es, möglichst effizient einzusetzen.

Um diese erworbenen Kompetenzen erweitern und ausbauen zu können empfehlen wir Ihnen, die Erkenntnisse in Ihrem täglichen Arbeiten mit SelectLine Produkten einzusetzen und Ihre Fähigkeiten zu erweitern und aufzufrischen. Deshalb freuen wir uns schon jetzt, Sie bei einem weiteren Kurs wieder bei uns zu begrüßen. Als Beilage finden Sie ein Formular mit den nächsten Kursen und Terminen, womit Sie sich anmelden können. Die Anmeldung ist aber auch weiterhin über unsere Website www.selectline.ch unter „Unterstützung/Schulungen“ möglich.

Freundliche Grüsse

SelectLine Software AG

4.3 Ihre Notizen und Erkenntnisse