



Abfragen

Zugehörige Informationen			
Standorte, Nutzer..., Prüfregeln, Abfragen... , Masken, Prüfpläne..., Kommunikation, Nachrichten..., Vorgangssteuerung, Allgemeine Konfiguration	Allgemeine Bedienungshinweise	Abfragecode	—
Übersicht, Prüfregeln, Abfragen , Fehlertexte, Abhilfetexte			

Fachlich/inhaltliche Beschreibung

Die Perspektive **Prüfregeln, Abfragen...** umfasst die ASYS-Repositoryobjekte der Prüfregeln und Abfragen sowie der Fehlertexte und Abhilfetexte. Die beiden erstgenannten Objektklassen werden in dieser Perspektive definiert, aber größtenteils in den Konfigurationsoptionen der anderen Perspektiven zur Verwendung referenziert. Die beiden Textbausteinobjektklassen sind Hilfsobjekte für die Prüfregeln. Die genannten Objektklassen sind global für alle Repository-Standorte gültig und nutzbar.

Die in dieser Perspektive konfigurierbaren Repository-Objekte sind:

- **Prüfregeln:** Prüfregeln sind durch den Admin definierbare logische Ausdrücke oder 'Miniprogramme' in **JavaScript**. Sie können auf Inhalte von **Maskenattributen**, **Abfragen** und auf Datenfelder von **Nachrichten** zugreifen. Sie werden gebraucht, um **Regelmengen** für Masken und Nachrichten zusammenzustellen und **Bearbeitungsfilter** zu definieren. Jede Prüfregel liefert als Ergebnis einen logischen Wert, also entweder **true** (Wahr) oder **false** (Falsch). Das Ergebnis **true** bedeutet dabei immer, dass die in der Prüfregel formulierte Prüfung erfolgreich absolviert wurde. Das Ergebnis **false** stellt den Fehlerfall dar. Dieser ist mit einer Fehlerstufe verbunden, welche die Schwere des Fehlerfalls darstellt (von 0 = Steuerregel für den Ablauf bis 4 = Datensatz kann/darf nicht gespeichert werden). Für den Fehlerfall können der Prüfregel ein **Fehlertextbaustein** und ein **Abhilfetextbaustein** zugeordnet werden, die im Fehlerprotokoll des geprüften Datensatzes abgelegt werden und ggf. einem Nachrichtenabsender übermittelt werden. Prüfregeln können als 'Sofortprüfregel' einem Maskenattribut zugeordnet werden. Sie werden dann sofort nach Eingabe eines Wertes in das Maskenfeld ausgeführt und nicht erst im Rahmen eines Prüfplan beim Speichern des Datensatzes. Zur Auslieferungsausstattung von ASYS gehören umfangreiche ASYS-Standardprüfpläne, die aus **Standardprüfregeln** bestehen. Für diese Standardprüfregeln gilt ein weitgehender Schutz vor Veränderungen durch den Admin.
- **Abfragen:** Abfragen sind durch den Admin definierbare Abfrage-Anweisungen an die ASYS-Datenbank. In Abfragen lassen sich alle Datenverknüpfungen formulieren, die das **ASYS-Datenmodell (FOM)** hergibt. Abfragen gibt es in zwei Hauptvarianten, den **freien Abfragen** und den **internen Abfragen**. Freie Abfragen können den ASYS-Nutzern über die **Nutzerprofile** zur Ausführung freigegeben

werden. Sie lassen sich Masken zuordnen, damit sie im Kontext dieser Masken für die Nutzer bereitgestellt werden und aus den Masken mit Parametern versorgt werden können. Freie Abfragen gibt es darüber hinaus in den speziellen Ausprägungen der [QS-Abfragen^{1\)}](#) und der [Auswertungsabfragen](#).

Interne Abfragen sind, wie ihre Bezeichnung bereits andeutet, nicht für die Verwendung durch die ASYS-Nutzer gedacht. Interne Abfragen dienen vielmehr als Mechanismus zur Ermittlung von Informationen aus dem Datenbestand für andere Repository-Objekte. Interne Abfragen sind bedeutsam für *Prüfregeln*, die [Empfängerermittlung von Nachrichten](#) bei der ASYS-internen Kommunikation, für [Textformulardefinitionen](#), für die [Vorgangssteuerung](#) und den [Webservice](#).

- **Fehlertexte:** Fehlertexte sind Textbaustein-Objekte, die den *Prüfregeln* zugeordnet werden. Fehlertextbausteine lassen sich den Prüfregeln auf zwei Arten zuordnen: Als Standardfehlertextbaustein in der Definition der *Prüfregel* und als individueller Fehlertext in der Zuordnung einer Prüfregel zu einer [Regelmenge](#).
- **Abhilfetexte:** Abhilfetexte sind Textbaustein-Objekte, die den *Prüfregeln* zugeordnet werden. Abhilfetextbausteine lassen sich den Prüfregeln auf zwei Arten zuordnen: Als Standardabhilfetextbaustein in der Definition der *Prüfregel* und als individueller Abhilfetext in der Zuordnung einer Prüfregel zu einer [Regelmenge](#).

2013/12/16 12:40 · eflor

Bedienung

IKA STD Doppelte Betriebe andere FKB

Schriftgröße 9

Abfrage

Name: IKA STD Doppelte Betriebe andere FKB

Sprechender Name: Doppelte Betriebe an unterschiedlichen FKB

Autor: Consist ITU Letzte Änderung:

Code 1 Code 2 (2) Eigenschaften SQD-Code Masken (0) Prüfregeln (0) VG-Skripte (0) Vorgänge (0) »6

CLASSES (Tabellen und Sichten)

Fkb=Fkb
Fkb.Betrieb=Btr

RESULTS (Ausgabefelder)

Btr.Behoerdliche Nummer=BEH_NUMMER
Btr.Rolle=BETRIEBSTYP
ANZAHL=count()
\$FSCHLUESSEL_MAX=max({%Fkb.Firmenschluessel%})
\$FSCHLUESSEL_MIN=min({%Fkb.Firmenschluessel%})

CONDITIONS (Kriterien)



{%Fkb.Gueltig Bis%} is null
{%Btr.FL gestrichen%}=#false#
{%Fkb.FL gestrichen%}=#false#

GROUP (Gruppierungen - nicht gruppierte Ausgabefelder müssen mit einer Aggregatfunktion (max, min...) versehen werden)

{%Btr.Behoerdliche Nummer%}, {%Btr.Rolle%}

Zuoberst befinden sich auf dem Tab-Reiter eine Zeile mit Buttons:

- Der Button **Tochteranwendung** startet die [Administrator-Tochteranwendung](#) und übergibt ihr die Abfrage für einen Test mit Daten aus der ASYS-Datenbank.
- Der Button **Syntaxcheck** startet eine formale Prüfung des Abfragecodes. Dabei wird nur geprüft, ob der Abfragecode den formalen Regeln von SQDs²⁾ gehorcht. Geprüft wird, ob der ASYS-SQD-Interpreter Fehler zurückmeldet. Diese werden dem Admin angezeigt. Ob die Abfrage im Kontext von Masken, Datensätzen oder Nachrichten technisch und fachlich sinnvoll arbeitet, kann dieser Test nicht ermitteln.
- Der Button **SQD-Import** ermöglicht den Import von Abfragecode. Es öffnet sich ein Dialog mit einem Eingabefeld, in welches SQD-Code eingegeben (z.B. via Zwischenablage) und angepasst werden kann. Über den **Ok**-Button wird der Code in die aktuelle Abfrage übernommen. **Achtung:** Die Übernahme überschreibt vorhandenen SQD-Code der aktuellen Abfragen in allen Bestandteilen! In den Dialog eingegebene Bestandteile, die sich nicht interpretieren lassen, gehen verloren. Diese Option ist primär dazu gedacht, vorhandene und funktionsfähige Abfragen, die vor ASYS 6.x als SQD-Dateien abgespeichert wurden, weiterhin nach ASYS importieren zu können.
- Der Button **Aktualisieren** startet die Prüfung, in welchen Konfigurationsobjekten die Abfrage genutzt wird. Fundstellen werden in die Tabellen der Listenbereichs (s.u.) eingetragen bzw. die Listen werden aktualisiert.

-  Der Button **Großbuchstaben** wandelt die Aliasnamen³⁾ im Abschnitt *Results* in Großbuchstaben um. Insbesondere in Abfragen für [Textformulare](#) lassen sich damit Probleme bei der Erstellung der Word-Dokumente vermeiden.
-  Der Button **Import** öffnet einen Dateiauswahldialog, mit dem sich aus der *ASYS7-Oberfläche* exportierte Suchen und Datenbereichssuchen für einen Import auswählen lassen (der Export erfolgt als *.zip-Datei). Bei einem Import werden die vorhandenen Eintragungen in den Tab-Reitern Code 1 und Code 2 vollständig und ohne Rückfrage ersetzt. Die importierte Suche kann als Grundlage für eine Abfragendefinition genutzt werden.
- Das Auswahlfeld Schriftgröße ermöglicht die Anpassung der Schriftgröße in den SQD-Code-Eingabefeldern. Eine Änderung an dieser Auswahl dient ausschließlich der Darstellung und hat keinerlei Auswirkung auf die Funktionsweise der Abfrage.

Der darunter befindliche Bereich des Tab-Reiters einer Abfrage im Bearbeitungsbereich des Administrators gliedert sich in zwei Abschnitte:

Abfrage

- Name: Der interne und eindeutige Name der Abfrage. Diese wurde bei der Erstellung durch den Admin vergeben und ist nachträglich nicht mehr änderbar.
- Sprechender Name: Der Name der Abfrage, wenn diese in der ASYS7-Oberfläche angeboten wird. Kann nachträglich angepasst werden.
- Autor: Der Name des Nutzers, der die Abfrage definiert hat. Hier wird per Default der Login-Name des Administrators vorgeschlagen, es kann aber auch der tatsächliche Name derjenigen Person eingetragen werden, der oder die die Abfrage erstellt hat.
- Letzte Änderung: Das Datum der letzten Änderung der Abfrage. Wird automatisch verwaltet.

Listenbereich

Den unteren Teil des Bearbeitungsbereichs nimmt ein Abschnitt mit vierzehn Tab-Reitern ein. Aufgrund der Anzahl konfigurierbarer Eigenschaften einer Abfrage und der Menge anderer Objektklassen, in denen Abfragen verwendet werden können, lassen sich in der Standardbildschirmauflösung leider nicht alle Tab-Reiter nebeneinander anzeigen.

Die Tab-Reiter können in zwei Gruppen eingeteilt werden:

- Die Gruppe der Tab-Reiter, auf denen die Abfrage konfiguriert wird: **Code 1**, **Code 2**, **Eigenschaften** und **Masken**.
- Die Gruppe der Tab-Reiter, auf denen nur Informationen angezeigt werden: alle anderen Tab-Reiter.



Auf die Bestandteile des Abfragecodes wird an dieser Stelle nicht eingegangen, hierzu gibt es eine gesonderte Seite über den [Abfragecode](#).

Die erläuternden Abbildungen zu den via **Strg+Leertaste** oder automatisch öffnenden kontextsensitiven Eingabehilfen stellen hinsichtlich der in den Abbildungen enthaltenen Funktions- oder Codefragmentlisten bzw. der Auswahlmöglichkeit von Klassen oder Attributen nur das jeweilige Prinzip als



Symbolbild dar. Diese Abbildungen werden nicht laufend an den tatsächlichen Stand der Eingabehilfe angepasst!



Code 1

Der Tab-Reiter Code 1 nimmt die vier wichtigeren Codegruppen der Abfragen auf:

- CLASSES (Tabellen und Sichten/Views)
- RESULTS (Ausgabefelder)
- CONDITIONS (Kriterien)
- GROUP (Gruppierungen)

Die restlichen Codegruppen folgen auf dem Tab-Reiter Code 2.

Die beiden ersten Codegruppen **dürfen nicht leer** bleiben, damit eine ausführbare Abfrage erstellt werden kann. Zumindest muss angegeben werden, aus welcher/welchen Tabelle(n) der ASYS-Datenbank Datensätze abgefragt und welche Tabellenspalte(n) ausgegeben werden sollen. Alle weiteren Gruppen können bedarfsweise leer bleiben.

Leere Codegruppen werden automatisch minimiert. Ihr Platz im Bearbeitungsbereich der Abfrage wird gleichmäßig auf die gefüllten Codegruppen aufgeteilt. Alle Codegruppen können manuell geöffnet und geschlossen werden. Hierzu befindet sich rechts oben in der Überschriftenzeile der Gruppe eine Schaltfläche, die zwischen schließen  und öffnen  in Abhängigkeit vom Zustand der Gruppe umschaltet.

Alle Codegruppen werden zeilenweise ausgewertet, d.h. ein Ausdruck in der jeweiligen Gruppe **muss in einer einzigen Zeile** beginnen und abgeschlossen werden. Komplexe Ausdrücke über mehrere Zeilen sind nicht möglich, da mehrere Zeilen nicht als zusammengehörig markiert werden können. Dies gilt auch für Klammerausdrücke.

Verwenden Sie für die in den folgenden Abschnitten einzutragenden Codefragmente ausschließlich Standardzeichen des Alphabets!



Deutsche Sonderzeichen (äÄ, öÖ, üÜ, ß) sind unbedingt zu vermeiden, da die nicht als Platzhalter für Datenbankattribute dienenden Bestandteile der Abfragendefinition unverändert in die generierten SQL-Kommandos an die Datenbank durchgereicht werden. Die Datenbanken können aber die vom lateinischen Standardalphabet abweichenden Zeichen meist nicht korrekt verarbeiten.



Die nachfolgenden Abbildungen zu den Eingabehilfen stellen nur das Prinzip dar. Welche Eingabehilfen angeboten werden, ist abhängig von der jeweils eingesetzten Programmversion des Repository-Administrators. Der Umfang der auswählbaren Funktionen kann sich daher mit der Zeit ändern, weshalb die



Abbildungen einen älteren Versionsstand wiedergeben können.

CLASSES

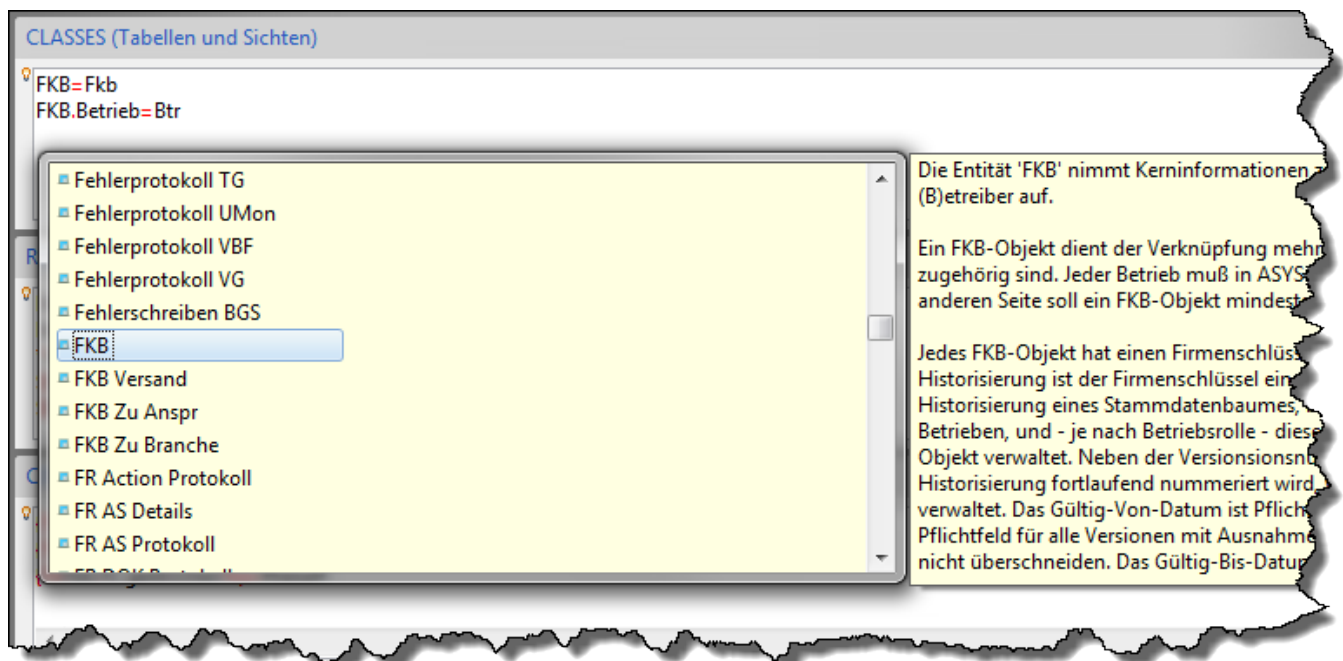
Hinweise zum Abfragecode und den Ausdrücken finden Sie im [Abschnitt CLASSES](#).

In diesem Abschnitt werden die Tabellen und Sichten (= Views) der ASYS-Datenbank eingetragen, die in der Abfrage verwendet werden sollen. Alle Tabellen und Views, aus denen Datenfelder in den weiteren Abschnitten (→ Codegruppe RESULTS bis HAVING) verwendet werden sollen, müssen hier eingetragen werden. Für jede genutzte Tabelle und jeden View muss eine Zeile eingetragen werden. Damit eine Abfrage ausgeführt werden kann, muss mindestens eine Tabelle oder ein View eingetragen werden.

Jede Tabelle wird durch einen Alias-Namen vertreten, um bei langen Tabellen- oder Viewnamen mit kurzen Alias-Namen arbeiten zu können.

Im Datenmodell bereits verbundene Tabellen können automatisch in der Abfrage verknüpft werden. Hierzu ist die Verkettung in Form von 'Tabellenname1.Tabellenname2' anzugeben. Bei der Interpretation der Abfragendefinition wird daraus automatisch das passende Verknüpfungsstatement zwischen den beiden Tabellen generiert.

Wird in einer leeren Zeile **Strg+Leertaste** gedrückt, öffnet sich ein Auswahlfenster der Datenmodellobjekte (= Tabellen & Views) der ASYS-Datenbank:



Die alphabetisch sortierte Liste der Datenmodellobjekte wird ergänzt durch eine Anzeige rechts der Liste mit einem Erläuterungstext zur Objekt aus dem [Fachobjektemodell \(FOM\)](#) von ASYS (liegt kein Erläuterungstext vor, bleibt das Anzeigefenster geschlossen). Ein Eintrag aus der Auswahlliste wird per **Doppelklick** ausgewählt.

Wird hinter ein Datenmodellobjekt ein Punkt geschrieben, so öffnet sich das Auswahlfenster mit

denjenigen Datenmodellobjekten, die von dem Objekt vor dem Punkt im Datenmodell erreichbar sind:



Die **Verknüpfung der Tabellen** erfolgt seit ASYS V6.02 auf neue Weise und automatisch mittels JOIN-Ausdrücken⁴⁾. Diese Verknüpfungsausdrücke werden weiterhin automatisch aus den hier im CLASSES-Abschnitt eingetragenen Tabellennamenverkettungen generiert.

Alle bisherigen Tabellenverknüpfungen werden hierbei in **INNER-JOIN**-Statements übersetzt - dies entspricht der bisherigen Abfragenlogik. Das bedeutet, dass als Treffer nur diejenigen Datensätze in Frage kommen, die für jede der angegebenen Tabellenverknüpfungen auch mindestens eine Datensatzverknüpfung besitzen (**Beispiel**: Werden die Entsorgerbetriebsstätte und die Teilanlagen in einer Anlage verknüpft, so kommen nur noch Datensätze als Treffer in Frage, bei denen für den Entsorgerbetrieb auch mindestens eine Teilanlage in der Datenbank eingetragen ist. Entsorgerbetriebe ohne Teilanlage werden kein Bestandteil der Trefferliste, auch wenn sie den übrigen Filterbedingungen des CONDITIONS-Abschnittes genügen.).

Mit dem Wechsel zu JOIN-Ausdrücken für Abfragen ist nun auch die Formulierung von **OUTER-JOIN**-Statements möglich. Das bedeutet, dass als Treffer auch diejenigen Datensätze in Frage kommen, die keinen verknüpften Datensatz besitzen (**Beispiel**: Ein Outer-Join von Entsorgerbetrieb und Teilanlagen, liefert auch die Entsorgerbetriebe ohne Teilanlagen, die den übrigen Bedingungen im CONDITIONS-Abschnitt genügen.).

Die üblichen CLASSES-Ausdrücke werden immer als **INNER-JOIN**-Statement ausgewertet:

```
FKB=Fkb
FKB.Betrieb=Btr
```

Diese Tabellenverknüpfung liefert nur FKB-Datensätze mit mindestens einem zugehörigen Betriebsstättendatensatz.

Durch einen * am Ende des Klassen-Aliasnamens wird aus dem CLASSES-Ausdruck ein **OUTER-JOIN**-Statement:


```
FKB=Fkb  
FKB.Betrieb=Btr*
```

Diese Tabellenverknüpfung liefert zusätzlich auch FKB-Datensätze, denen keine Betriebsstättendatensatz zugeordnet ist.

In beiden Fällen müssen aber zusätzlich auch noch die Bedingungen aus dem CONDITIONS- und dem HAVING-Abschnitt für die Trefferdatensätze erfüllt sein!

Damit die Verknüpfung der Tabellen mittels JOIN-Statements erwartungsgemäß funktioniert, müssen **ALLE** Tabellen im CLASSES-Abschnitt mittels der Punktnotation 'Tabellenname1.Tabellenname2' untereinander verbunden sein!

Ob eine OUTER-JOIN-Bedingung von einer INNER-JOIN-Bedingung gefolgt werden darf, ist möglicherweise von der verwendeten Datenbank abhängig und muss ggf. ausprobiert werden. Beispiel:



```
FKB=Fkb  
FKB.Betrieb=Btr*  
Betrieb.Adresse=Adr
```

Eine Mischform aus JOIN-Statements und zusätzlichen Statements im CONDITIONS-Abschnitt ist im Prinzip möglich (s. [Abfragecode und Ausdrücke](#)). Dies kann aber auch von der verwendeten Datenbank und der konkreten Formulierung der Tabellenverknüpfung abhängen.

RESULTS

In diesem Abschnitt werden die Spalten definiert, die im Ergebnis angezeigt werden. Für jede Ergebnisspalte muss genau eine Zeile in diesem Abschnitt eingetragen werden. Damit eine Abfrage ausführbar ist, muss zumindest eine Ergebnisspalte definiert sein. Alle Spalten, die hier definiert werden, müssen sich aus Attributen der Tabellen oder Views (→ Codegruppe CLASSES) ermitteln lassen⁵⁾.

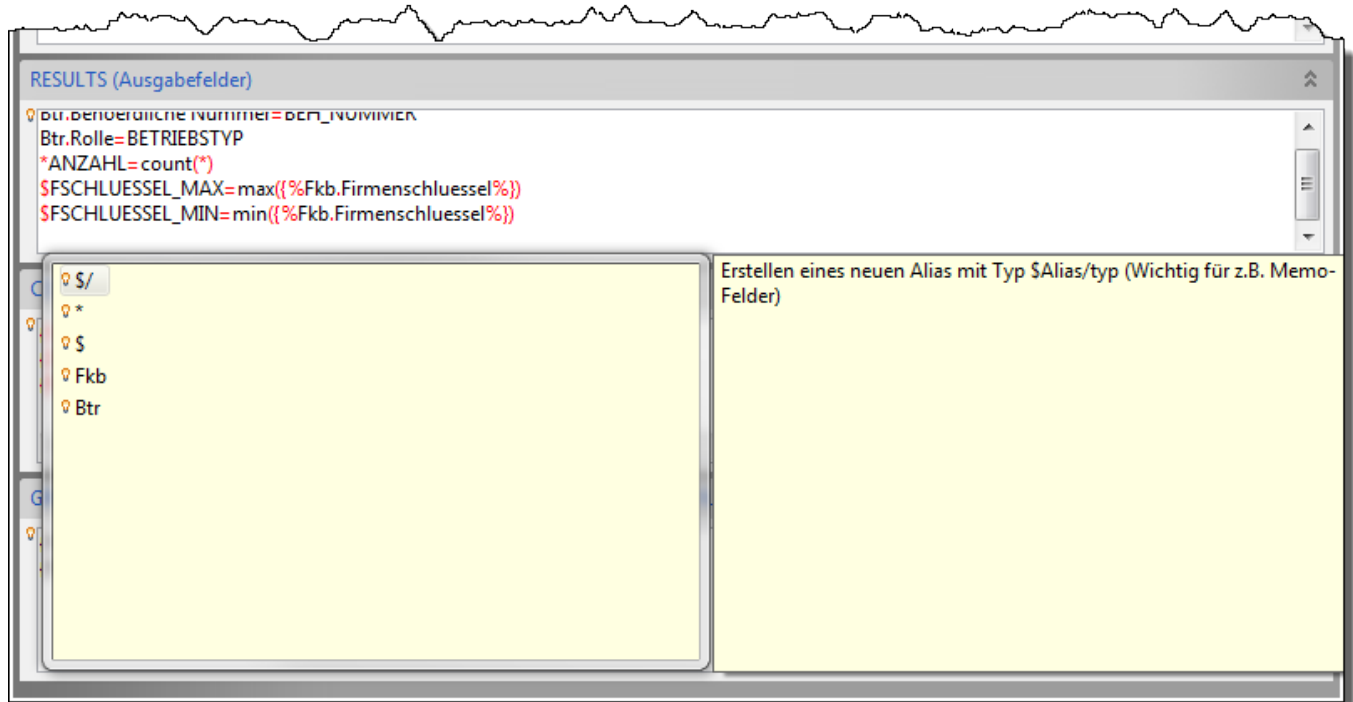
Jede Spalte wird durch einen Alias-Namen vertreten, da für die Referenzierung der Attributnamen nicht die physischen Attributnamen der Datenbank, sondern die Modellnamen des [ASYS-Fachobjektemodells](#) verwendet werden.



Bitte verwenden Sie für die Alias-Namen bei Abfragen für Textformulare **IMMER GROSSBUCHSTABEN**. Dies verhindert Probleme, die gelegentlich auftreten, wenn das gleiche Attribut mehrfach mit verschiedenen Alias-Namen ausgegeben wird!

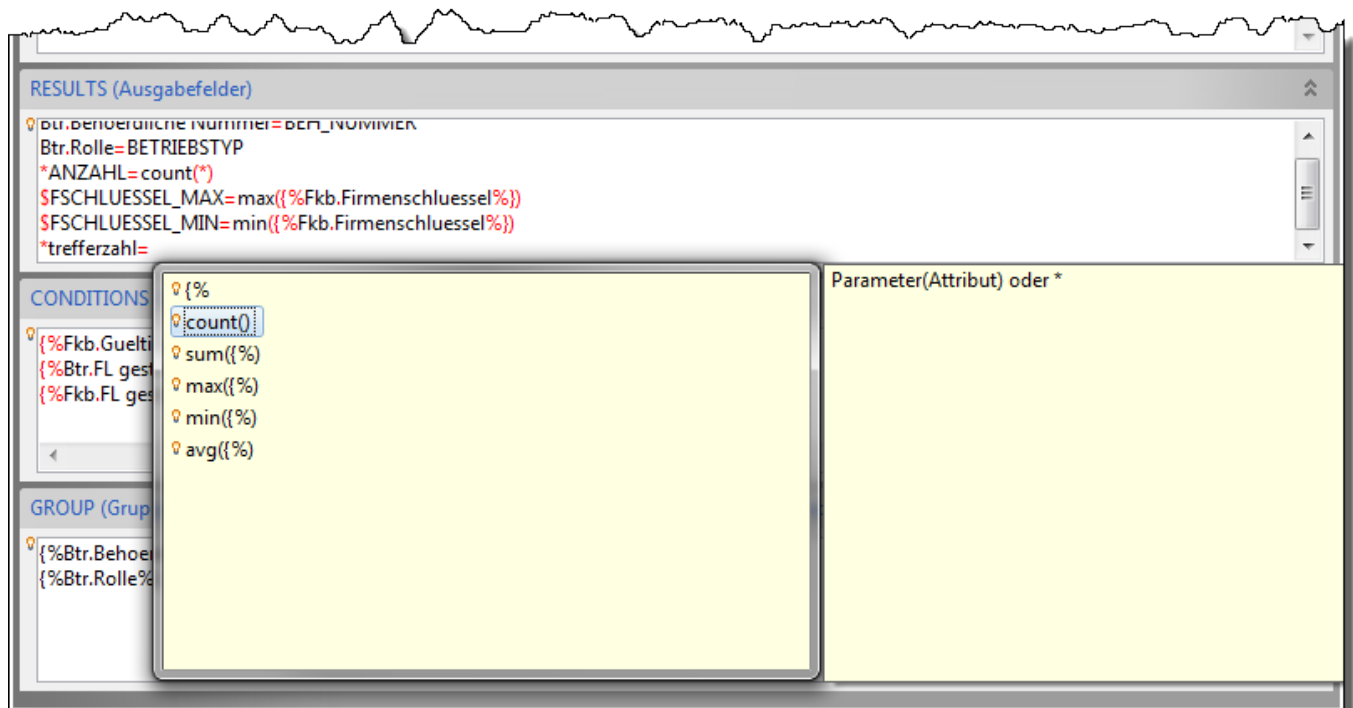
Wird in einer leeren Zeile **Strg+Leertaste** gedrückt, öffnet sich ein Auswahlfenster mit Vorschlägen zur Definition einer Ausgabespalte. Da ein Spaltenalias entweder vor dem Platzhalter mit dem

Modellnamen des Attributs, oder dahinter definiert werden kann, enthält die Auswahlliste Einträge für beide Alternativen. Aus diesem Grunde werden auch die Tabellenalias aus dem CLASSES-Abschnitt hier wieder zur Auswahl angeboten.



Die Liste wird ergänzt durch eine Anzeige rechts der Liste mit einem Erläuterungstext. Bei den Tabellenaliasen kommen die Informationen aus dem [Fachobjektemodell \(FOM\)](#) von ASYS (liegt kein Erläuterungstext vor, bleibt das Anzeigefenster geschlossen). Ein Eintrag aus dieser und den nachfolgend beschriebenen Auswahllisten wird per **Doppelklick** ausgewählt.

Wird hinter einem Gleichheitszeichen (=) die Tastenkombination **Strg+Leertaste** gedrückt, öffnet sich ein Auswahlfenster mit Aggregatfunktionen - die erste Zeile dieser Liste ermöglicht aber auch, einen einfachen nicht aggregierten Attributausdruck anzulegen.



Die Liste wird ergänzt durch eine Anzeige rechts der Liste mit einem Erläuterungstext.



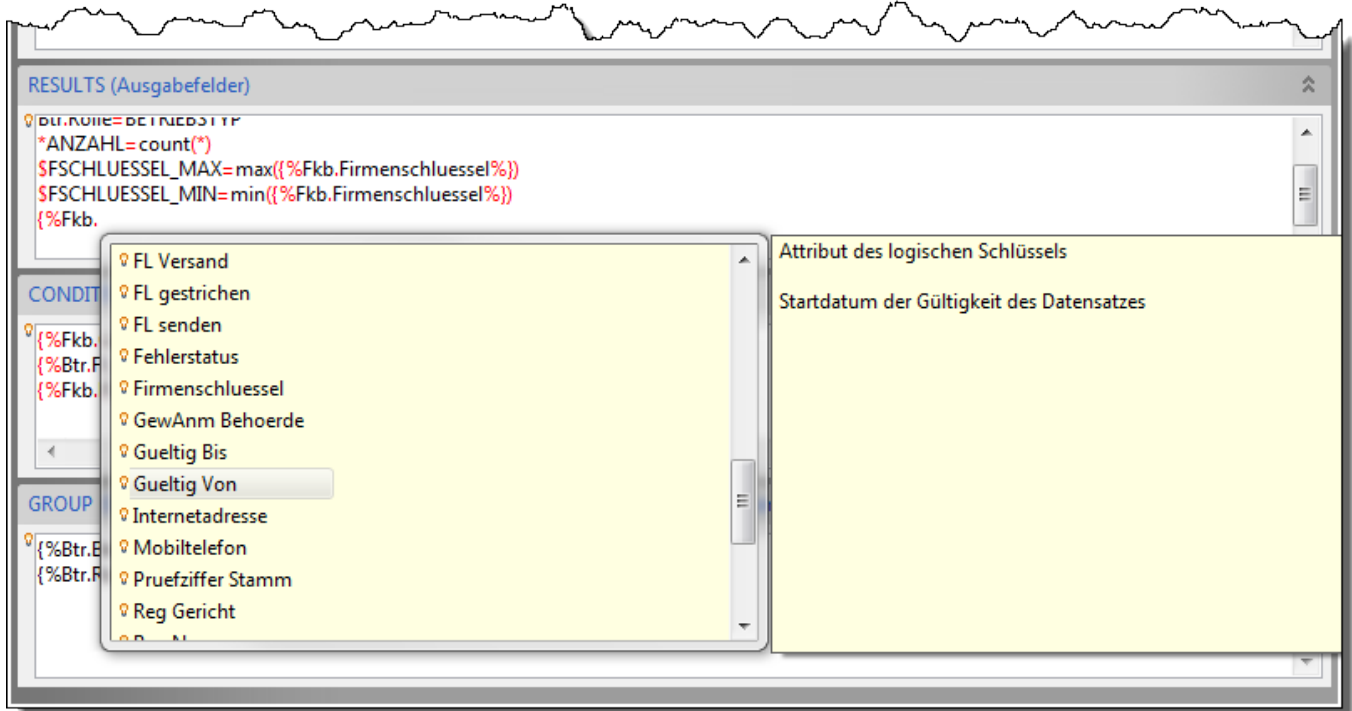
Achten Sie bitte auf diese Erläuterungstexte!

Für einige Einträge dieser Auswahl kann die Einschränkung gelten, dass sie nur mit einer bestimmten Datenbank - Oracle, MS SQL-Server oder Access - genutzt werden können.

Wird in eine Zeile die Zeichenkombination `{%` geschrieben, öffnet sich ein Auswahlfenster mit den Class-Aliasen aus dem CLASSES-Abschnitt:

The screenshot shows the ASYS software interface. At the top, a window titled "RESULTS (Ausgabefelder)" displays a query: `DEL.KUHE=DEINBESTITP`, `*ANZAHL=count(*)`, `$FSCHLUESSEL_MAX=max({%Fkb.Firmenschlüssel%})`, `$FSCHLUESSEL_MIN=min({%Fkb.Firmenschlüssel%})`, and `{%`. Below the query, a small dialog box is open, showing a list of class aliases: `Fkb` and `Btr`. To the right of the dialog, a text box provides detailed information about the `Fkb` class. The text explains that the `Fkb` entity contains core information for a company (Firma), a company (Körperschaft), or a company operator (Betreiber). It states that an `Fkb` object is used to link multiple companies that belong to a legal person. Every company must be assigned to an `Fkb` object in ASYS, and each `Fkb` object must have at least one associated company. It also mentions that each `Fkb` object has a company key (Firmenschlüssel), which, along with the version number of the history, forms a unique key for the `Fkb` object. The history of a company data tree, consisting of `Fkb` objects, one or more companies, and - depending on the company role - the associated information, is managed in the `Fkb` object. Besides the version number, the company key is continuously numbered in the history, and for each version, a validity period is managed. The validity period is managed for each version.

Wird hinter einen Class-Alias ein Punkt geschrieben, öffnet sich ein Auswahlfenster mit den Attributen der zugehörigen Tabelle. Die Auswahlliste ist alphabetisch sortiert:



Wird ein Attribut ausgewählt, so wird der Platzhalteraussdruck automatisch mit der Zeichenkombination **%}** abgeschlossen.

CONDITIONS

In diesem Abschnitt werden die Bedingungen definiert, denen die Datensätze der Ergebnismenge gehorchen müssen. Der Bedingungsabschnitt kann leer bleiben, die Ergebnismenge ist dann der komplette Inhalt einer Tabelle oder Tabellenverknüpfung (je nach Inhalt des CLASSES-Abschnittes).

Jede Bedingung muss in genau einer Zeile formuliert werden (komplexere Ausdrücke mit AND- oder OR-Verknüpfungen können daher zu entsprechend langen Zeilen führen!). Alle Bedingungen (= Zeilen) werden grundsätzlich mittels AND verknüpft, d.h. ein Datensatz der Ergebnismenge muss ALLE Bedingungen erfüllen.

Werden Klammern benötigt, um Bedingungen zusammenzufassen bzw. zu hierarchisieren, so muss der Ausdruck in der obersten Klammerenebene in eine Zeile geschrieben werden.

Beispiele:

Die Buchstaben A bis F sind nachfolgend in diesem Beispiel als Platzhalter für beliebige Bedingungsausdrücke zu verstehen.

- A und B und C und D und E und F
wird geschrieben als

A
B
C
D
E

F

- A und (B oder C) und (D oder E) und F
wird geschrieben als

A

(B or C)

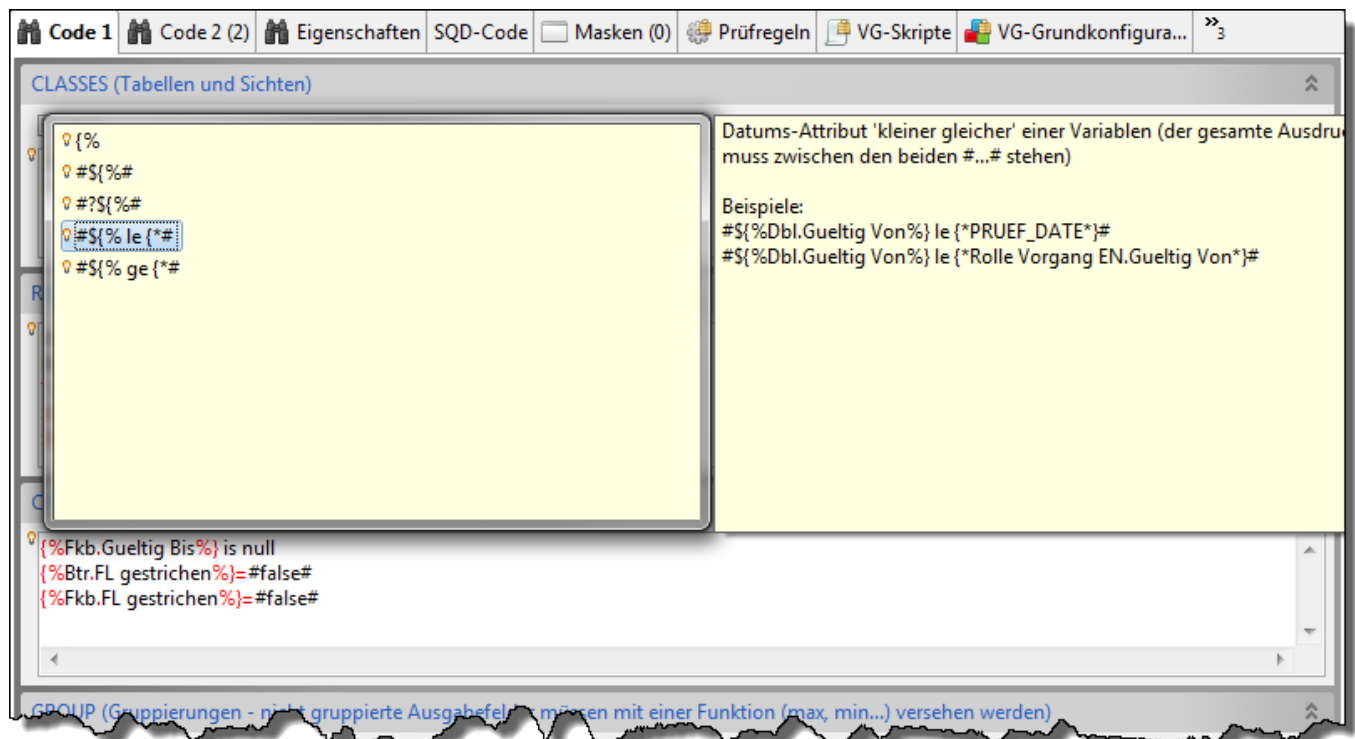
(D or E)

F

- A oder ((B oder C) und (D oder E)) oder F
wird geschrieben als

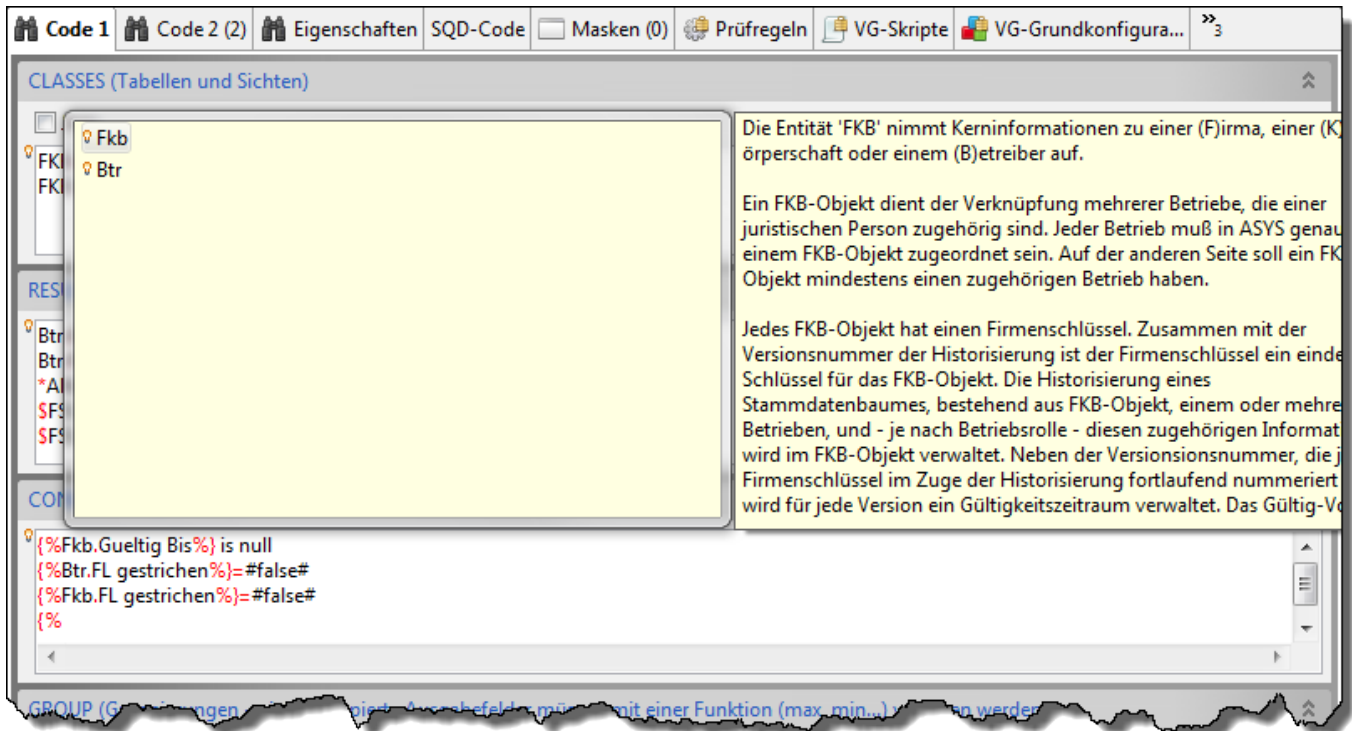
(A or ((B or C) and (D or E)) or F)

Wird in einer leeren Zeile **Strg+Leerzeichen** gedrückt, öffnet sich ein Auswahlfenster mit typischen Fragmenten für Bedingungsausdrücke:

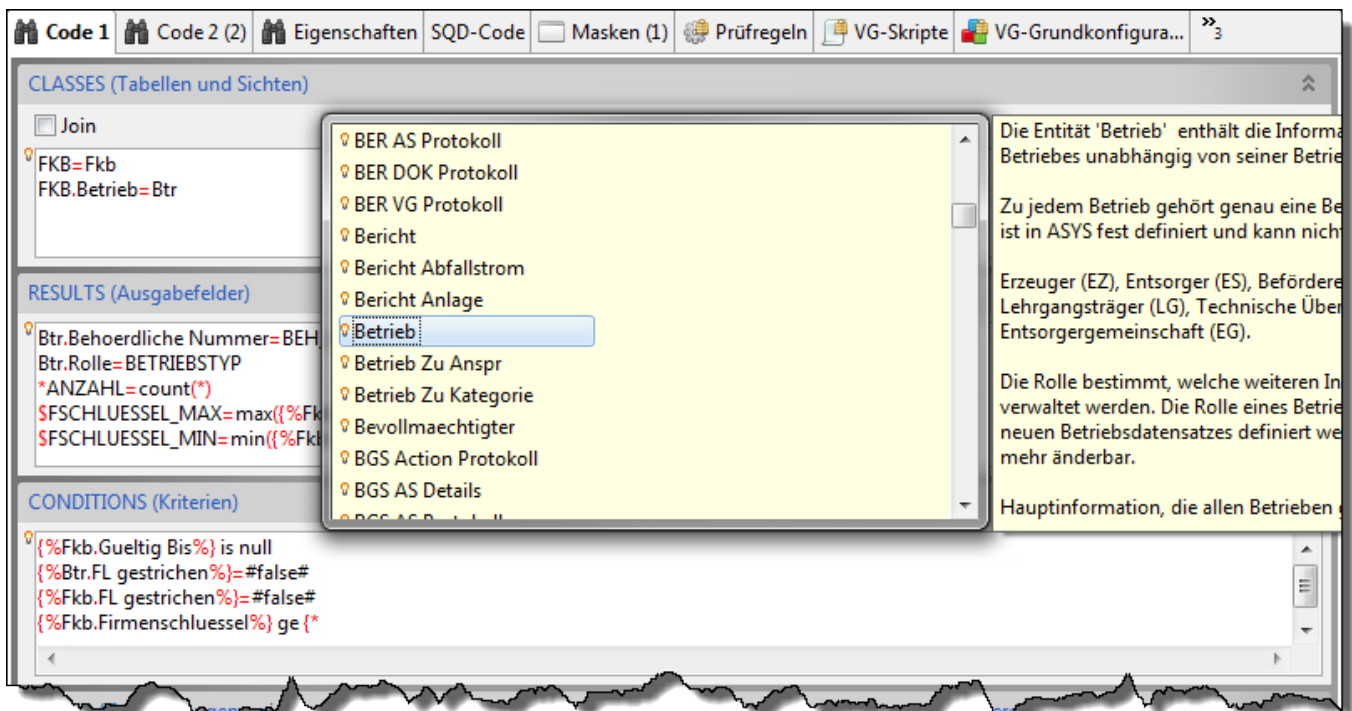


Die Liste wird ergänzt durch eine Anzeige rechts der Liste mit einem Erläuterungstext zum jeweiligen Fragment sowie ggf. typischen Anwendungsbeispielen. Ein Eintrag aus dieser und den nachfolgend beschriebenen Auswahllisten wird per **Doppelklick** ausgewählt.

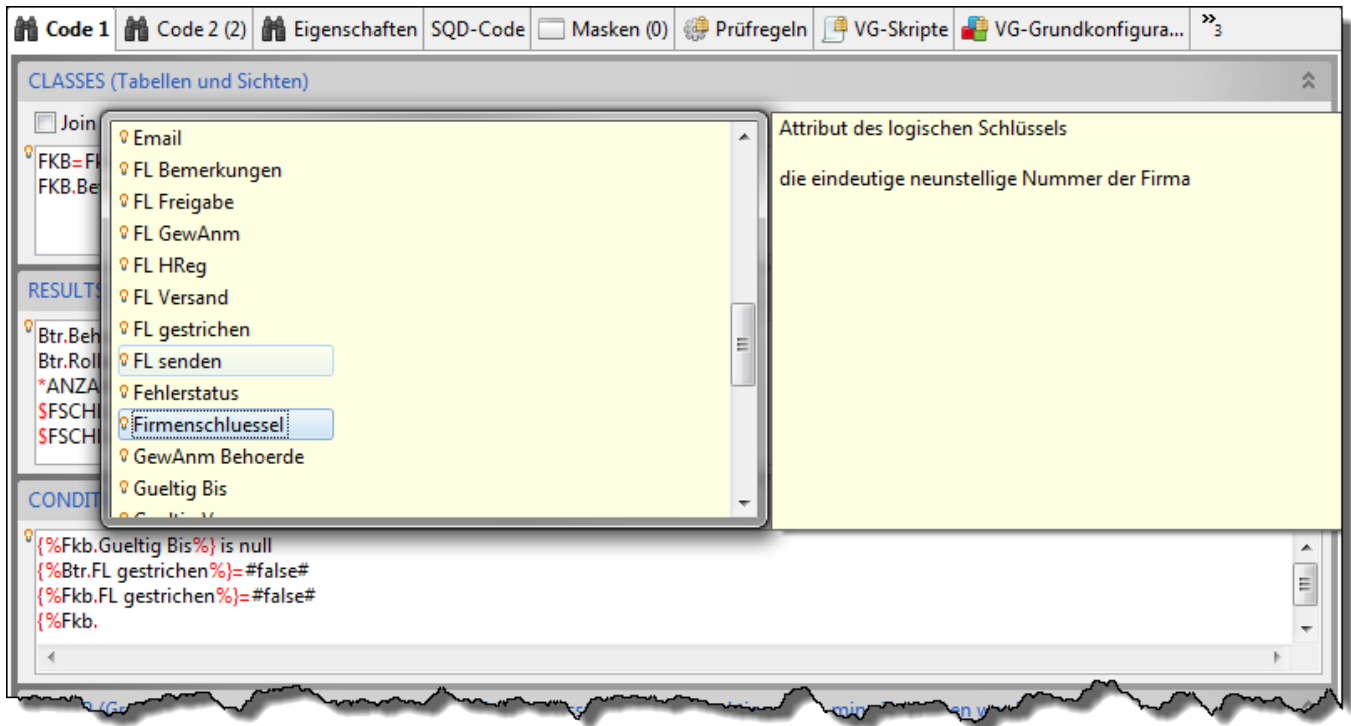
Wird die Zeichenkombination **{%** eingegeben, öffnet sich ein Auswahlfenster mit den Class-Aliassen aus dem CLASSES-Abschnitt:



Wird die Zeichenkombination `{*` eingegeben, öffnet sich ein Auswahlfenster mit den Modellnamen der Tabellen aus dem ASYS-Fachobjektemodell:



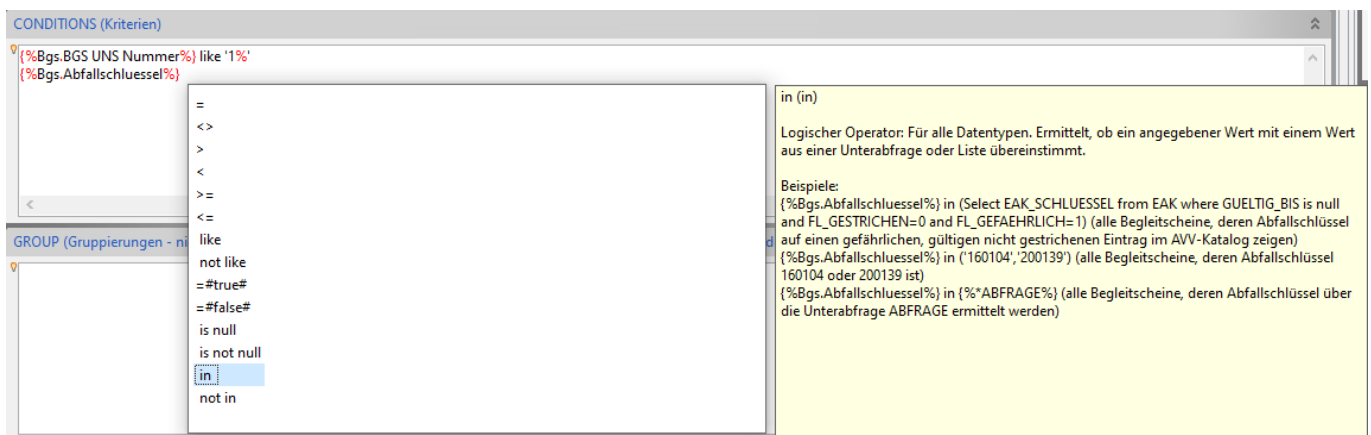
Wird hinter einen Class-Alias oder einen Fachobjektemodell-Tabellennamen ein Punkt geschrieben, öffnet sich ein Auswahlfenster mit den Attributen der zugehörigen Tabelle. Die Auswahlliste ist alphabetisch sortiert:



Wird ein Attribut ausgewählt, so wird der Platzhalterausdruck automatisch mit der Zeichenkombination **%}** bzw. ***}** abgeschlossen.

Vergleichsoperatoren

Wird hinter einer schließenden geschweiften Klammer (**}**) die Tastenkombination **Strg+Leerzeichen** gedrückt, erscheint ein Auswahlfenster der Vergleichsoperatoren.



- Wenn Sie einen der Vergleichsoperatoren **'='** bis **'=#false#'** auswählen und erneut **Strg+Leerzeichen** drücken, erscheint eine Auswahlliste mit typischen Ausdrucksfragmenten und Funktionen, die nach dem Operator genutzt werden können.
- Wenn Sie einen der Vergleichsoperatoren **'is null'** oder **'is not null'** wählen, erscheint bei erneutem drücken von **Strg+Leerzeichen** die Auswahlliste mit dem bereits gewählten Operator.
- Wenn Sie einen der Vergleichsoperatoren **'in'** oder **'not in'** wählen und danach **Strg+Leerzeichen** drücken, erscheint eine Auswahlliste mit drei Alternativen:
 - **(%*)**: Die Menge, in welcher der Wert des Ausdrucks vor dem 'in'-Operator enthalten sein muss, wird von einer anderen Abfrage geliefert.

- Nachdem Sie diese Alternative gewählt haben, können Sie erneut **Strg+Leerzeichen** drücken, um die Abfrage auszuwählen, welche die Menge für den Vergleich liefert. Aus dieser Abfrage wird nur die erste Ergebnisspalte aus dem RESULTS-Abschnitt für den Vergleich verwendet.
- **(Select)**: Die Menge, in welcher der Wert des Ausdrucks vor dem 'in'-Operator enthalten sein muss, wird von einer Abfrage geliefert, die Sie an dieser Stelle fest definieren (muss in einer Zeile stehen!).
- **()**: Die Menge, in welcher der Wert des Ausdrucks vor dem 'in'-Operator enthalten sein muss, wird an dieser Stelle als feste Aufzählung angegeben (z.B. ('A', 'E', 'I') für drei Bundeslandskennbuchstaben).

GROUP

In diesem Abschnitt werden die Datenbankattribute eingetragen, nach denen die Datensätze des Ergebnisses gruppiert werden soll. Alle Datensätze, die in einem Gruppierungsattribut den gleichen Wert enthalten, bilden zusammen eine Gruppe. Jeder Datensatz im Abfrageergebnis ergibt sich daher aus einem oder mehreren Datensätzen der Datenbank.

Alle Attribute, nach denen gruppiert werden soll, müssen in eine Zeile dieses Abschnitts eingetragen werden und sind durch Kommata (,) zu trennen.

Für alle Spalten im RESULTS-Abschnitt, die nicht gleichzeitig ein Gruppierungsattribut sind, muss grundsätzlich festgelegt werden, wie der Wert der Spalte im Ergebnis für die Gruppe ermittelt werden soll, da sich die Datensätze einer Gruppe in diesen Spalten ja unterscheiden können. Hierfür sind alle diese Spalten mit einer Aggregatfunktion zu definieren. Dies erfolgt im Abschnitt RESULTS (s.o.).

Eine Gruppierung der Datensätze kann über eine oder mehrere Spalten erfolgen. Wird über mehrere Spalten gruppiert, ist die Reihenfolge der Spalten für die Gruppierung zu beachten. Alle Datensätze werden nach der ersten Gruppierungsspalte gruppiert. Die Datensätze dieser Gruppen werden anschließend gruppenweise nach der folgenden Gruppierungsspalte gruppiert. Entsprechend wird vorgegangen, wenn weitere Gruppierungsattribute definiert werden.

Eine Gruppierung muss nicht notwendigerweise über Spalten erfolgen, die auch im Ergebnis (→ Abschnitt RESULTS) erscheinen. Sie muss aber über Spalten erfolgen, die Bestandteil der Tabellen oder Views sind (→ Abschnitt CLASSES) oder sich aus diesen ermitteln lassen. Eine Gruppierung kann über eine berechnete Spalte erfolgen, wenn die Datenbank dies unterstützt⁶⁾.

Wird in einer leeren Zeile **Strg+Leerzeichen** gedrückt, öffnet sich ein Auswahlfenster mit nur einem auswählbaren Eintrag:

```
{%
```

Wird die Zeichenkombination **{%** eingegeben, öffnet sich ein Auswahlfenster mit den Class-Aliasen aus dem CLASSES-Abschnitt (vergl. Abbildungen weiter oben). Ein Eintrag aus dieser und der nachfolgend beschriebenen Auswahlliste wird per **Doppelklick** ausgewählt.

Wird hinter einen Class-Alias ein Punkt geschrieben, öffnet sich ein Auswahlfenster mit den Attributen der zugehörigen Tabelle. Die Auswahlliste ist alphabetisch sortiert (vergl. Abbildungen weiter oben). Wird ein Attribut ausgewählt, so wird der Platzhalterausdruck automatisch mit der Zeichenkombination **%}** abgeschlossen.

Code 2



Der Tab-Reiter Code 2 nimmt die vier seltener benötigten Codegruppen der Abfragen auf:

- ORDER (Sortierung)
- HAVING (Filterung auf berechnete Ergebnisspalten)
- VARIABLES (Variablendefinition für die Bedingungen → CONDITIONS)
- UNION (Abfragenverknüpfung)
- SKRIPTS (Skriptdefinitionen)

Die wichtigeren Codegruppen befinden sich auf dem Tab-Reiter Code 1.

Keine dieser Codegruppen muss mit Inhalt gefüllt werden, um eine ausführbare Abfrage erstellen zu können.

Leere Codegruppen werden automatisch minimiert. Ihr Platz im Bearbeitungsbereich der Abfrage wird gleichmäßig auf die gefüllten Codegruppen aufgeteilt. Alle Codegruppen können manuell geöffnet und geschlossen werden. Hierzu befindet sich rechts oben in der Überschriftenzeile der Gruppe eine

Schaltfläche, die zwischen schließen  und öffnen  in Abhängigkeit vom Zustand der Gruppe umschaltet.

Alle Codegruppen werden zeilenweise ausgewertet, d.h. ein Ausdruck in der jeweiligen Gruppe **muss in einer einzigen Zeile** beginnen und abgeschlossen werden. Komplexe Ausdrücke über mehrere Zeilen sind nicht möglich, da mehrere Zeilen nicht als zusammengehörig markiert werden können. Dies gilt auch für Klammerausdrücke.

Verwenden Sie für die in den folgenden Abschnitten einzutragenden Codefragmente ausschließlich Standardzeichen des Alphabets!



Deutsche Sonderzeichen (äÄ, öÖ, üÜ, ß) sind unbedingt zu vermeiden, da die nicht als Platzhalter für Datenbankattribute dienenden Bestandteile der Abfragendefinition unverändert in die generierten SQL-Kommandos an die Datenbank durchgereicht werden. Die Datenbanken können aber die vom lateinischen Standardalphabet abweichenden Zeichen meist nicht korrekt verarbeiten.

ORDER

In diesem Abschnitt werden die Datenbankattribute eingetragen, nach denen die Datensätze des Ergebnisses sortiert werden sollen.

Alle Attribute, nach denen sortiert werden soll, müssen in einer Zeile dieses Abschnitts eingetragen werden. Eine Sortierung kann aufsteigend oder absteigend erfolgen. Für eine aufsteigende Sortierung genügt die Benennung des Sortierattributs, für eine absteigende Sortierung ist hinter das Sortierattribut das Kürzel DESC zu schreiben. Wird nach mehreren Attributen sortiert, müssen sie in der Zeile durch Kommata (,) getrennt werden.

Eine Sortierung der Datensätze kann über eine oder mehrere Spalten erfolgen. Wird über mehrere Spalten sortiert, ist die Reihenfolge der Spaltenangaben in diesem Abschnitt für die Sortierung zu beachten. Alle Datensätze werden nach der ersten Sortierungsspalte sortiert. Die Datensätze, die in dieser aktuellen Sortierungsspalte gleiche Werte enthalten, werden nach der folgenden Sortierungsspalte sortiert. Entsprechend wird vorgegangen, wenn weitere Sortierungsattribute definiert werden. Die Sortierreihenfolge kann je Sortierungsspalte individuell festgelegt werden (z.B. nach der ersten Spalte absteigend, nach der zweiten aufsteigend).

Eine Sortierung muss nicht notwendigerweise über Spalten erfolgen, die auch im Ergebnis (→ Abschnitt RESULTS) erscheinen. Sie muss aber über Spalten erfolgen, die Bestandteil der Tabellen oder Views sind (→ Abschnitt CLASSES) oder sich aus diesen ermitteln lassen. Eine Sortierung kann über eine berechnete Spalte erfolgen, wenn die Datenbank dies unterstützt⁷⁾.

Wird in einer leeren Zeile **Strg+Leerzeichen** gedrückt, öffnet sich ein Auswahlfenster mit zwei auswählbaren Einträgen:

```
{%  
{% DESC
```

Der erste Eintrag ist für die Definition einer aufsteigenden, der zweite für eine absteigende Sortierung vorgesehen.

Wird die Zeichenkombination **{%** eingegeben, öffnet sich ein Auswahlfenster mit den Class-Aliassen aus dem CLASSES-Abschnitt (vergl. Abbildungen weiter oben). Ein Eintrag aus dieser und der nachfolgend beschriebenen Auswahlliste wird per **Doppelklick** ausgewählt.

Wird hinter einen Class-Alias ein Punkt geschrieben, öffnet sich ein Auswahlfenster mit den Attributen der zugehörigen Tabelle. Die Auswahlliste ist alphabetisch sortiert (vergl. Abbildungen weiter oben). Wird ein Attribut ausgewählt, so wird der Platzhalterausdruck automatisch mit der Zeichenkombination **%}** abgeschlossen.

HAVING

Die Bedingungen im CONDITIONS-Abschnitt filtern Datensätze in der Datenbank. Jeder einzelne Datensatz muss diesen Bedingungen gehorchen.

Ein Having-Ausdruck ist eine zusätzliche Bedingung an die Ergebnisdatensätze, die sich nicht aus den zugrundeliegenden Datensätzen der Datenbank direkt ermitteln lässt, sondern nur aus dem Ergebnis selbst. Typisches Beispiel hierfür ist eine Abfrage mit Gruppierung, bei der die Gruppen im Ergebnis eine bestimmte Mindeststärke an Mitgliedern haben sollen; diese Information lässt sich nicht aus den Einzeldatensätzen der Datenbank ermitteln.

Für diesen Abschnitt steht nur eine eingeschränkte Kontextunterstützung mit Auswahllisten zur Verfügung. Jede Having-Bedingung muss in eine Zeile des Abschnitts geschrieben werden. Weiter Informationen zu Having-Ausdrücken finden sich im Abschnitt zu den [Ausdrücken in Abfragen](#).

VARIABLES

In diesem Abschnitt lassen sich Variablen für die Abfrage definieren, die vor allem im CONDITIONS-

Abschnitt genutzt werden können, um die Definition von Bedingungen zu vereinfachen und leichter lesbar zu machen. Insbesondere ist es möglich, mit Datumswerten zu rechnen und die Ergebnisse einer Variablen zuzuweisen, die für Bedingungen genutzt wird.

Für diesen Abschnitt steht keine Kontextunterstützung mit Auswahllisten zur Verfügung. Jede Variablen-Definition muss in eine Zeile des Abschnitts geschrieben werden. Weiter Informationen zu Variablen in Abfragen finden sich im Abschnitt zu den [Ausdrücken in Abfragen](#).

UNION

Die Ergebnisse mehrerer Abfragen können zusammengefasst werden, wenn folgende Bedingungen erfüllt sind:

- Die Abfragen müssen die **gleiche Anzahl an Ergebnisspalten** im RESULTS-Abschnitt enthalten.
- Die **Reihenfolge der Ergebnisspalten** im RESULTS-Abschnitt **ist maßgeblich für die Zuordnung** der Ergebnisspalten aus den zusammengefassten Abfragen.
- Die einander zugeordneten Ergebnisspalten müssen den **gleichen Datentyp** besitzen⁸⁾.
- Die UNION-Unterabfragen dürfen **keine Einträge im ORDER-Abschnitt** enthalten.

Eine Abfrage bildet die Hauptabfrage. Per UNION zuzuordnende weitere Abfragen, werden bei ihr in den UNION-Abschnitt eingetragen. Alle auf diese Weise zusammengefassten Abfragen liefern eine Ergebnisliste. **Hinweis:** Doppelte Einträge - also Treffer aus den verschiedenen per UNION gemeinsam ausgeführten Abfragen mit identischen Inhalten in allen Ergebnisspalten - werden zu einer Ergebniszeile zusammengefasst. Die Anzahl der Treffer einer UNION-Abfrage kann daher geringer als die Summe der Treffer der Einzelabfragen sein!

Um eine Abfrage als UNION-Abfrage mit einer Hauptabfrage zu verbinden, gehen Sie wie folgt vor:

1. Öffnen Sie den UNION-Abschnitt der Hauptabfrage.
2. Öffnen Sie per **Strg+Leerzeichen** die Auswahlliste der Abfragen im Repository und wählen Sie die mit der Hauptabfrage zu verbindende UNION-Abfrage aus.
3. Mehrere UNION-Abfragen zu einer Hauptabfrage werden per 'Komma' getrennt hintereinander geschrieben (vgl. ORDER-Abschnitt)

Die UNION-Abfragen werden als Link präsentiert. **Strg+Linksklick** öffnet die jeweilige UNION-Abfrage.



1. Die Auswahlliste der Abfragen im UNION-Abschnitt aktualisiert sich nicht automatisch, wenn neue Abfragen hinzukommen. Definieren Sie daher die UNION-Abfragen bevor Sie die Hauptabfrage erstellen, damit Sie die UNION-Abfragen zur Auswahl erhalten. Alternativ schließen Sie die Hauptabfrage und öffnen Sie sie erneut, wenn Sie Ihre UNION-Abfragen definiert haben, damit die Auswahlliste neu eingelesen wird. Der Name der UNION-Abfrage kann aber auch per Zwischenablage in den Abschnitt UNION der Hauptabfrage kopiert werden.
2. Die ersten beiden Bedingungen oben lassen sich leichter einhalten, wenn Sie die Möglichkeit nutzen, eine Abfrage (Haupt- oder UNION-Abfrage) zu kopieren.

SKRIPTS

In diesem Abschnitt lassen sich Ausdrücke definieren, die auf den in den RESULTS ermittelten Spalten und Werten (Rechen-)Operationen auszuführen, die weiteren Ergebnisspalten liefern. Ursprünglich wurde dieser Abschnitt benötigt, um in Abfragen an die Inhalte von Memofeldern zu gelangen - und nicht nur an die Nummer ihres Datensatzes. Hierfür ist dieser Abschnitt nicht mehr erforderlich.

Für diesen Abschnitt steht keine Kontextunterstützung mit Auswahllisten zur Verfügung. Jede Skript-Definition muss in eine Zeile des Abschnitts geschrieben werden. Weiter Informationen zu Skripten in Abfragen finden sich im Abschnitt zu den [Ausdrücken in Abfragen](#).

Eigenschaften

The screenshot shows the 'Eigenschaften' tab in a software interface. The top bar contains several icons and labels: 'Code 1', 'Code 2', 'Eigenschaften' (selected), 'SQD-Code', 'Masken (0)', 'Prüfregeln (2)', 'VG-Skripte (0)', 'Vorgänge (0)', and a right arrow icon. The main area is divided into several sections:

- Standardabfrage**: A checkbox labeled 'Standardabfrage' is checked. Below it is a text field labeled 'Info (für Anwender)'.
- deaktiviert**: A checkbox labeled 'deaktiviert' is unchecked.
- Spezielle freie Abfrage**: A section with several checkboxes: 'QS-Abfrage' (unchecked), 'Auswertungen' (unchecked), 'Reihenfolge' (checked, with a value of '0' in a text field), and 'Abfrage für Archivierung' (unchecked). A button 'Reihenfolge prüfen' is next to the 'Reihenfolge' field.
- Interne Abfrage für**: A section with checkboxes: 'Empfängerermittlung' (unchecked), 'Textformular' (unchecked), 'Prüfregel' (checked), 'Webservice' (unchecked), and 'Vorgangssteuerung' (unchecked).
- Zu öffnende Maske**: A text field containing 'Begleitschein'.
- Hauptklasse**: A text field containing 'BGS UNS'.
- Notwendige Zusatzbedingung**: A text field containing 'BGS.BGS_UNNUMMER like '1%''.
- Notizen**: A large text area at the bottom for notes.

Der Tab-Reiter Eigenschaften enthält zusätzliche Informationen zu einer Abfrage:

- **Standardabfrage**: Abfragen, die zum Standardumfang von ASYS gehören - insbesondere Abfragen, die in [Standardprüfregeln](#) genutzt werden - sind durch Governikus ITU und die IKA als Standardabfragen gekennzeichnet. An diesen Abfragen lassen sich nur in eingeschränktem Maße Änderungen vornehmen. Die Einstufung als Standardabfrage kann nicht verändert werden.
- **deaktiviert**: Eine Abfrage die als deaktiviert gekennzeichnet ist, kann nicht ausgeführt werden. Bei einer freien Abfrage wird sie den Anwendern nicht zur Auswahl angeboten.
- **Anwenderinfo**: Für eine freie Abfrage kann hier ein erläuternder Text eingetragen werden, der in der ASYS-Oberfläche bei der Auswahl der Abfrage angezeigt wird.
- **QS-Abfrage**: Die Abfrage wird als Qualitätssicherungsabfrage genutzt⁹⁾.
- **Auswertung**: Die Abfrage wird als Auswertungsabfrage genutzt¹⁰⁾.
- **Reihenfolge**: Die QS-Abfragen und Auswertungsabfragen sollen in einer vorgebbaren Reihenfolge ausgeführt werden. Der Zahlenwert gibt die laufende Nummer der Abfrage in einer

Folge von entsprechenden Abfragen für eine Maske an.

- **Reihenfolge prüfen:** Mit dem Button kann überprüft werden, ob die Nummern der QS-Abfragen und Auswertungsabfragen einer Maske sich nicht überschneiden. Geprüft werden nur die Folgen der Masken, denen auch die aktuell geprüfte Abfrage zugeordnet ist.
 - **Abfrage für Archivierung:** **Ab V7.12** Die Abfrage kann für die Ermittlung von zu archivierenden Daten mittels der [Tochteranwendung](#) genutzt werden. Hierzu sind zwei wichtige Bedingungen einzuhalten:
 - a) als 'zu öffnende Maske' ist die Maske einer Hauptdatenkategorie einzutragen.
 - b) Von dieser Hauptdatenkategorie muss das ID-Feld im RESULTS-Abschnitt dem Alias THE_ID zugewiesen sein (z.B. beim BGS: *THE_ID={%Bgs.%}).
 - **Interne Abfrage:** Die Abfrage ist eine interne Abfrage (Ankreuzfeld gesetzt) und keine freie Abfrage (Ankreuzfeld nicht gesetzt). Interne Abfrage können nicht durch die Nutzer in der ASYS-Oberfläche direkt aufgerufen werden.
 - **Empfängerermittlung:** Die Abfrage wird im Rahmen der [Empfängerermittlung](#) in der Kommunikation genutzt¹¹⁾.
 - **Textformular:** Die Abfrage wird im Rahmen einer [Textformulardefinition](#) genutzt¹²⁾.
 - **Prüfregel:** Die Abfrage wird im Rahmen einer [Prüfregeldefinition](#) genutzt¹³⁾.
 - **Webservice:** Die Abfrage wird im Rahmen des [Webservice-Zugangs](#) genutzt¹⁴⁾.
 - **Vorgangssteuerung:** Die Abfrage wird im Rahmen der [Vorgangssteuerung](#) genutzt¹⁵⁾.
 - **Zu öffnende Maske:** Handelt es sich um eine Abfrage, aus deren Ergebnisliste heraus eine Maske mit dem jeweiligen Datensatz geöffnet werden können soll, so ist diese Maske hier anzugeben. Mit dem -Button wird ein Auswahldialog der Maskendefinitionen aufgerufen, aus denen genau eine Maske als Ziel ausgewählt werden kann. Der -Button löscht eine ausgewählte Maske.
 - Im Feld **Hauptklasse** wird der Name der Hauptklasse der ausgewählten Maske angezeigt. Im Abschnitt RESULTS des Tab-Reiters 'Code 1' **muss** die erste Zeile mit dem Ergebnislistenspaltennamen **THE_ID** enthalten sein, welche die Datensatz-IDs aus dieser Hauptklasse erhält. **Ab V7.12** Dies gilt auch, wenn die Abfrage für die Ermittlung von zu archivierenden Daten in der Tochteranwendung genutzt werden soll (vgl. oben [Abfrage für Archivierung](#)).
 - Das Feld **Notwendige Zusatzbedingung** kann einen Bedingungsausdruck für den CONDITIONS-Bereich enthalten, der sicherstellt, dass nur Daten passend zu der ausgewählten Maske ermittelt werden. Ist das Feld leer, ist keine Zusatzbedingung erforderlich - dies wird in den meisten Fällen zutreffen. Die notwendige Zusatzbedingung kann als übliche Bedingung in {%...%}-Notation im CONDITIONS-Bereich definiert werden. Alternativ kann die Inhalt diese Feldes auch markiert und via Zwischenablage kopiert werden.
- Hinweis: Eine Bedingung der Art '...DEL_STS=0' kann ignoriert werden!**
- **Notizen:** Freitextfeld für eine interne Dokumentation der Abfrage.

Die Einstufung einer Abfrage muss durch den Admin erfolgen. Die Nutzung einer Abfrage in einem bestimmten Kontext (z.B. Prüfregel) führt nicht dazu, dass für die Abfrage das entsprechende Ankreuzfeld gesetzt wird. Andererseits werden aber in den Auswahllisten des Administrators regelmäßig nur Abfragen angeboten, die eine passende Einstufung besitzen.

Summierungs- , Auswertungsabfragen

Was sind Auswertungsabfragen? Auswertungsabfragen sind normale Abfragen, die bestimmten

zusätzlichen Regeln gehorchen (s.w.u.).

Wozu dienen die? Mit Auswertungsabfragen können dem Nutzer zusätzliche Informationen zum aktuellen Datensatz in der Maske angezeigt werden. Diese Informationen können aus beliebigen anderen Bereichen des ASYS-Datenbestandes kommen (schließlich handelt es sich um Abfragen).

Die **Auswertungsabfragen** stellen eine Möglichkeit dar, den Anwendern Auswertungen, Summierungen oder sonstige Informationen für den Datensatz, der gerade bearbeitet wird, zur Anzeige zu bringen. Bei einem Begleitschein z.B. für die Fragestellung, wie viele Begleitscheine es bisher für den Nachweis gibt, wie hoch die gesamte entsorgte Menge bisher ist, welche Angaben in der VE stehen etc. Für jede Einzelfragestellung ist hierbei eine eigene Abfrage zu erstellen. Nach Betätigung der Schaltfläche aus der Anwendungsoberfläche werden die Abfragen nacheinander ausgeführt und jedes Ergebnis in eine eigene Zeile in eine Tabelle geschrieben. Es können beliebig viele Abfragen definiert werden. Die Abfragen werden nach ihrer laufenden Reihenfolgennummer sortiert ausgeführt (s. [Eigenschaften der Abfrage](#)). Je Maske kann es genau ein Set an Auswertungsabfragen geben.

Bei der Definition der (Auswertungs-)Abfragen müssen einige Rahmenbedingungen eingehalten werden:

- Die Abfrage muss der gewünschten Maske (Aufgabenbereich) zugeordnet sein.
- Die Abfrage muss als Auswertungsabfrage in den Eigenschaften gekennzeichnet sein und eine Reihenfolgennummer erhalten.
- Es darf nur ein Ausgabeparameter definiert sein (bzw. nur der erste Ausgabeparameter wird angezeigt, alle anderen werden ignoriert). D.h. im RESULTS-Abschnitt steht nur eine Zeile (bzw. alle nachfolgenden Zeilen sind ohne Belang)!
- Die Abfrage darf nur einen (oder keinen) Datensatz zurückliefern (bzw. nur der erste Datensatz wird angezeigt, alle anderen werden ignoriert).
- Für die Beschreibung, um was für einen Wert es sich handelt, wird das Anwenderinfo-Feld der Abfrage verwendet.
- Es muss auf der Maske ein Datensatz selektiert worden sein.
- Die Parameter im CONDITIONS-Bereich der Abfrage müssen automatisch durch Werte von der Maske ersetzt werden können. Können bei einer Abfrage die Parameter während der Ausführung nicht automatisch durch Werte von der Maske ersetzt werden, kommt es zu einer Fehlermeldung und die Ausführung wird abgebrochen.



Auswertungsabfragen müssen wie die freien Abfragen im **Nutzerprofil** des ausführenden Nutzers enthalten sein. Sie können also über die Zuordnung von Auswertungsabfragen zu verschiedenen Nutzerprofilen steuern, welche Nutzer bestimmte zusätzliche Informationen auf diese Weise angezeigt bekommen.

QS-Abfragen


Die **QS-Abfragen** dienen der Auswertung des ASYS-Datenbestandes unter dem Aspekt der Qualitätssicherung. Je Maske kann es genau ein Subset an QS-Abfragen geben. Alle Abfragen müssen hierbei eine Trefferzahl als Ergebniss liefern. Die erste Abfrage (Reihenfolge: 1) gibt die Grundgesamtheit der Treffer an (100%). Alle folgenden Abfragen enthalten die gleichen Parameter

wie die erste Abfrage sowie weitere 'feste' Einschränkungen, die festzulegende Qualitätsmerkmale überprüfen. An der Oberfläche ausgegeben werden die Anzahl der Treffer sowie ein prozentualer Wert im Verhältnis zur Grundgesamtheit (dies geschieht durch die Oberfläche, muss also in der Abfrage nicht definiert werden). Es können beliebig viele Abfragen definiert werden. Die Abfragen werden nach ihrer laufenden Reihenfolgennummer sortiert ausgeführt (s. Eigenschaften der Abfrage). An der Oberfläche werden die Parameter einmal abgefragt und dann für alle Abfragen ausgeführt. Damit die Abfragen ausgeführt werden können, müssen sie einem Nutzerprofil des ausführenden Nutzers zugeordnet sein.

Bei der Definition der QS-Abfragen müssen einige Rahmenbedingungen eingehalten werden:


- Die Abfrage muss der gewünschten Maske (Aufgabenbereich) zugeordnet sein.
- Die Abfrage muss als QS-Abfrage in den Eigenschaften gekennzeichnet sein und eine Reihenfolgennummer erhalten.
- Es darf nur ein Ausgabeparameter definiert sein und bei diesem muss es sich um eine Trefferzahl handeln.
- Die Abfrage darf nur einen (oder keinen) Datensatz zurückliefern (bzw. nur der erste Datensatz wird angezeigt, alle anderen werden ignoriert).
- Für die Beschreibung, um was für einen Wert es sich handelt, wird das Anwenderinfo-Feld der Abfrage verwendet.
- Es muss auf der Maske ein Datensatz selektiert worden sein.
- Alle variablen Parameter im CONDITIONS-Bereich der Abfragen müssen für alle Abfragen identisch sein.

SQD-Code


Auf dem Tab-Reiter SQD-Code wird der Abfragecode angezeigt, der sich aus den Eintragungen in den Abschnitten der Tab-Reiter Code 1 und Code 2 ergibt. Die Anzeige dient der Überprüfung im Fehlerfall. Der Inhalt kann markiert und kopiert (Strg+C) werden. Über den Button des SQD-Imports () kann derart kopierter Abfragecode in eine neue Abfrage importiert werden.

Auf diesem Tab-Reiter können keine Änderungen an den angezeigten SQD-Codes vorgenommen werden.

Masken

Auf dem Tab-Reiter Masken kann die aktuelle Abfrage einer oder mehreren Masken zugewiesen werden. Die Zuweisung einer Maske erfolgt aus der Auswahlliste der Masken rechts via **Drag&Drop**. Alternativ kann über den Button **Maske hinzufügen** () ein Auswahldialog für die Masken geöffnet werden.

In beiden Fällen ist eine Mehrfachauswahl möglich. Bei der Übernahme der Masken in die Zuordnungsliste werden Mehrfachzuweisungen einer Maske ignoriert.

Um die Zuordnung einer Maske zu löschen, wird sie in der Liste markiert und anschließend der Button **Maske entfernen** () betätigt. Nach einer Sicherheitsabfrage wird die Zuordnung gelöscht.

Prüfregeln, VG-Skripte, Vorgänge, Vorgangsabläufe, Textformulare, Empfängerermittlung, Maskenattribute, Nutzerprofile, Attributsteuerung

Die weiteren Tab-Reiter zum Abfrageobjekt dienen nur der Anzeige, in welchem Kontext eine Abfrage genutzt wird. Damit kann ein Überblick erreicht werden, welche Konfigurationsbestandteile eine Abfrage nutzen und durch eine Änderung an der Abfragendefinition betroffen sein können.

Ein **Doppelklick** auf einen Eintrag in den Listen öffnet das entsprechende Objekt der Konfiguration, welches die Abfrage nutzt, im Bearbeitungsbereich. Meist wird dabei zuvor die Perspektive entsprechend gewechselt.

Weitere Informationen zu dieser Maske															
Ausdrücke in Abfragen															
landesspezifische Zusatzinformationen: SH HH NI HB NW HE RP BW BY SL BE MV ST BB TH SN															

1)

QS = Qualitätssicherung; Abfragen, die Informationen zusammentragen, welche Auskunft über die Qualität des ASYS-Datenbestandes in einem bestimmten Bereich des Datenmodells geben

2)

SQD = Simplified Query Definition; ASYS-spezifische Abfragesprache, ermöglicht die Verwendung von Platzhaltern zur Einbindung von ASYS-Datenfeldern in Datenbankabfragen

3)

Beispiel: Aus der Zeile *Btr.Behoerdliche Nummer=Beh_Nummer* wird *Btr.Behoerdliche Nummer=BEH_NUMMER*

4)

Die bis dahin praktizierte klassische Methode, die Tabellen einer Datenbankabfrage mittels Bedingungen - also automatisch im Hintergrund zu den CONDITIONS-Ausdrücken ergänzten Verknüpfungsbedingungen - zu verbinden, wurde durch diese Methode ersetzt.

5)

Die Möglichkeiten und Beschränkungen ergeben sich aus der Datenbankabfragesprache SQL bzw. dem Angebot an Funktionen der genutzten Datenbank

6)

Beispiel: Eine Tabelle von Rechtecken mit Spalten für die Längen der beiden Seiten kann nach der Fläche der Rechtecke gruppiert werden mittels 'GROUP BY (Rechteck.LaengeX * Rechteck.LaengeY)'

7)

Beispiel: Eine Tabelle von Rechtecken mit Spalten für die Längen der beiden Seiten kann nach der Fläche der Rechtecke sortiert werden mittels 'ORDER BY (Rechteck.LaengeX * Rechteck.LaengeY)'

8)

Bei Zeichenketten - Typ String - spielt die Länge aber keine Rolle.

9) 10)

S.U.

11) 12) 13) 14) 15)

Die Angabe dient nur der Dokumentation und/oder der Filterung der Abfragen in den Auswahllisten und -masken des Administrators. Sie hat keine Auswirkung auf die Ausführung der Abfrage

From:

<https://hilfe.gadsys.de/asyshilfe/> - **ASYS-Onlinehilfe**

Permanent link:

<https://hilfe.gadsys.de/asyshilfe/doku.php?id=adm6:pat:abfragen>

Last update: **2024/05/02 13:41**

