

Summieren/Zählen direkt auf der Datenbank

Normalerweise wird jede Operation eines SE16XXL-Scripts **separat** ausgeführt. Eine Ausnahme bilden eine Reihe von Inner-Joins am Anfang eines Scripts, die optional in einem Zug auf der Datenbank durchgeführt werden können (Datenbank-Join).

Ab **Version 3.5B** von SE16XXL ist es auch möglich, Summier- und/oder Zähl-Operationen **direkt auf der Datenbank** durchzuführen. Dies kann zu erheblichen Performance-Verbesserungen führen, insbesondere wenn eine sehr große Anzahl von Datensätzen verarbeitet werden soll. Dieser Modus zum Ausführen der Operationen ist **optional**, genau wie bei den Inner-Joins am Anfang eines Scripts. Dies bedeutet, dass die beteiligten Summier-/Zähl-Operationen entweder wie gewohnt auf sequentielle Weise oder direkt auf der Datenbank ausgeführt werden können. Das involvierte Script wird somit wie gewohnt Schritt für Schritt erstellt und kann auf die gleiche Weise getestet werden.

Es muss jedoch erwähnt werden, dass die Summier-/Zähl-Operationen und die zugehörigen Joins bestimmte Anforderungen erfüllen müssen, die ziemlich streng sein können, um auf der Datenbank durchgeführt zu werden.

Auf den folgenden Seiten werden wir die Funktionalität zunächst anhand von Beispielen veranschaulichen. Diese Vorgehensweise scheint vernünftig, da die Anforderungen ohne praktische Beispiele vorsichtige Anwender davon abhalten könnten, von dieser Funktionalität überhaupt Gebrauch zu nutzen. Werden jedoch zunächst einfache Beispiele erörtert, können die zugegebenermaßen komplexen Anforderungen möglicherweise ihren einschüchternden Charakter verlieren.

Die Beispiele mögen trivial aussehen, allerdings sollte bedacht werden, dass der Fokus darauf liegt, wie das Resultat anhand der Summierungs-/Zählfähigkeit der Datenbank erzielt werden kann.

Beispiel #1 - Zählen

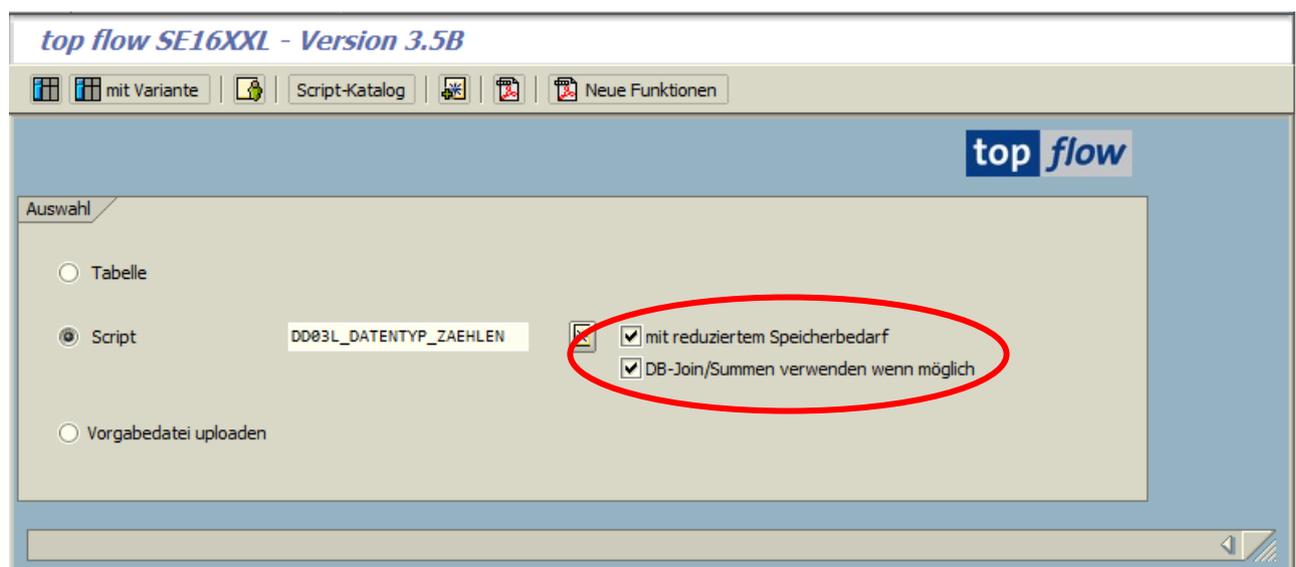
In diesem ersten Beispiel zählen wir die Anzahl der Data Dictionary-Felder, die den gleichen **Datentyp** aufweisen, d. h. CHAR, DATS usw. Unser Ausgangspunkt ist die Tabelle **DD03L**, die alle definierten Tabellenfelder enthält. Sie enthält Millionen von Datensätzen.

Zu diesem Zweck erstellen wir ein Script bestehend aus folgenden Operationen:

- 1) Selektion von Tabelle **DD03L** ohne jegliche Einschränkung.
- 2) Sortierung der Ergebnisliste nach **DATATYPE**.
- 3) *Extras* → *Eine Spalte einfügen* → *mit Anzahl sortierter Sätze*.
- 4) *Zeilen* → *Benachbarte Duplikate ausblenden*.
- 5) *Zeilen* → *Ausgeblendete Sätze verwerfen*.
- 6) Abschließend wählen wir als Felder für die Ergebnisliste nur **DATATYPE** und das **V-Feld** mit dem Zählwert.

Wir speichern dieses Script unter dem Namen “**DD03L_DATENTYP_ZAEHLEN**”.

Auf der Einstiegsmaske von SE16XXL rufen wir nun das Script mit folgenden Optionen auf:



ANMERKUNG: Die Option “*DB-Join/Summen verwenden wenn möglich*” ist nur in Kombination mit der Option “*mit reduziertem Speicherbedarf*” verfügbar.

Ein wichtiger Aspekt dieser Funktionalität wird durch den Zusatz **“wenn möglich”** angezeigt. Dies bedeutet, dass die Entscheidung, die gewünschten Operationen auf der Datenbank durchzuführen, letztendlich dem Programm überlassen wird – es analysiert das Script und prüft, ob die notwendigen Bedingungen erfüllt sind. Wenn nicht, wird das Script auf herkömmliche Weise ausgeführt, eine Operation nach der anderen.

Woher wissen wir dann, ob die Operationen tatsächlich direkt auf der Datenbank durchgeführt werden? In diesem Fall gibt das Programm am unteren Rand der Selektionsmaske eine kurze Meldung aus, die für ein paar Sekunden sichtbar bleibt:

```
**** DB-Summieren/Zählen verwendet ****
```

Zum Schluss erscheint die Ergebnisliste:

Script DD03L_DATENTYP_ZAEHLEN - 27 Einträge

Tabelle DD03L - Tabellenfelder

Datentyp	Zähler
	238.897
ACCP	656
CHAR	3.830.371
CLNT	140.993
CUKY	57.394
CURR	227.778
DATS	292.190
DEC	235.235
FLTP	16.846
INT1	9.280
INT2	29.225
INT4	78.355
LANG	38.498
LCHR	371
LRAW	1.079
NUMC	430.666
PREC	23
QUAN	95.109
RAW	43.715
REF	7.036
RSTR	996
SSTR	1.051
STRG	53.111
STRU	137.736
TIIMS	60.540
TTYP	215.212
UNIT	62.810

Wenn wir uns eine der Ergebniszeilen mittels Doppelklick genauer ansehen, werden wir feststellen, dass nur die beiden Felder vorhanden sind:

Satz von Tabelle DD03L - Tabellenfelder

Voriger Eintrag | Nächster Eintrag

Satz von Tabelle DD03L - Tabellenfelder

DATATYPE	'CHAR'	Datentyp im ABAP Dictionary
V-COUNT	'3.830.371'	SE16XXL - Standard V-Feld zum Zählen

Normalerweise sind auch im Modus “mit reduziertem Speicherbedarf” die Schlüsselfelder der Tabelle immer vorhanden. Da die Zählung jedoch direkt auf der Datenbank stattgefunden hat, wurden in diesem Fall keine solchen Felder zurückgegeben und können daher nicht angezeigt werden.

Nur um den Unterschied zu zeigen, werden wir das Script noch einmal ausführen, diesmal ohne die Datenbankoption.

Die Ergebnisliste sieht genauso aus wie oben. Wenn wir jedoch auf die gleiche Ergebniszeile doppelklicken, erhalten wir das folgende Bild:

Satz von Tabelle DD03L - Tabellenfelder

Voriger Eintrag | Nächster Eintrag

Satz von Tabelle DD03L - Tabellenfelder

TABNAME	'/1BCDWB/IQ000000000048'	'Tabellenname'
FIELDNAME	'ACPAID'	'Feldname'
AS4LOCAL	'A'	Aktivierungsstand eines Repository-Objektes
AS4VERS	'0000'	Version des Eintrags (nicht benutzt)
POSITION	'0003'	Position des Feldes in der Tabelle
DATATYPE	'CHAR'	Datentyp im ABAP Dictionary
V~COUNT	'3.830.371'	SE16XXL - Standard V-Feld zum Zählen

In diesem Fall sind die Schlüsselfelder des ersten Tabellenfeldes mit dem Datentyp CHAR sichtbar. Offenkundig sind sie in diesem Zusammenhang bedeutungslos.

WICHTIG: Die **maximale Trefferzahl** (auf der Selektionsmaske) bezieht sich auf die **Anzahl der aggregierten Sätze**, falls die Operationen auf der Datenbank ausgeführt werden. Andernfalls bezieht sie sich auf die Anzahl der selektierten Einzelsätze.

In diesem ersten Beispiel konnten wir das Script auf beide Arten ausführen, da die Tabelle DD03L auf dem verwendeten SAP-System etwa sechs Millionen Einträge enthält. Wenn wir stattdessen ein neueres System mit mehr als fünfzehn Millionen Einträgen genommen hätten, hätte das auf herkömmliche Weise ausgeführte Script wegen Speichermangels mit einem Kurzdump enden können. Es ist also nicht nur die verringerte Antwortzeit, die diese Funktionalität so interessant macht – sie ermöglicht außerdem Abfragen, die sonst nicht möglich wären.

Wir machen nun mit einem zweiten Beispiel weiter, das etwas komplizierter ist als das erste.

Beispiel #2 - Summieren

In diesem zweiten Beispiel betrachten wir **alle Fakturabelege** und summieren die Nettowerte für jede Verkaufsorganisation. Die entsprechende Tabelle ist diesmal VBRK (Faktura: Kopfdaten). Der Nettowert ist im Feld **NETWR** enthalten – das zugehörige Referenzfeld ist **WAERK**. Die Verkaufsorganisation jedes Beleges befindet sich im Feld **VKORG**.

Das Script, das wir erstellen, besteht aus folgenden Operationen:

- 1) Selektion von Tabelle **VBRK** ohne jegliche Einschränkung.
- 2) Sortierung der Ergebnisliste nach **VKORG** und **WAERK**.
- 3) *Extras → Spalten einfügen → mit (Zwischen)Summen.*
- 4) *Zeilen → Benachbarte Duplikate ausblenden* (VKORG und WAERK).
- 5) *Zeilen → Ausgeblendete Sätze verwerfen.*
- 6) Abschließend wählen wir als Felder für die Ergebnisliste nur **VKORG**, das **V-field** mit den Zwischensummen und **WAERK**.

Wir speichern das Script unter dem Namen “**VBRK_VKORG_NETWR_SUMMEN**”.

Mit einem leicht modifizierten Script würden wir das gleiche Ergebnis erzielen:

- 1) Selektion von Tabelle **VBRK** ohne jegliche Einschränkung.
- 2) Sortierung der Ergebnisliste nach **VKORG**.
- 3) *Extras → Spalten einfügen → mit (Zwischen)Summen.*
- 4) Sortierung der Ergebnisliste nach **VKORG** und **WAERK**.
- 5) *Zeilen → Benachbarte Duplikate ausblenden* (VKORG and WAERK).
- 6) *Zeilen → Ausgeblendete Sätze verwerfen.*
- 7) Abschließend wählen wir als Felder für die Ergebnisliste nur **VKORG**, das **V-field** mit den Zwischensummen und **WAERK**.

Die Operation *Extras → Spalten einfügen → mit (Zwischen)Summen* betrachtet den Währungsschlüssel (WAERK) **implizit** als Vergleichskriterium – deswegen verwendet die erste Sortierung nur VKORG als Kriterium. Die Sortier-Operation vor *benachbarte Duplikate ausblenden* ist notwendig, um die Zwischensummen für alle beteiligten Währungen im Resultat beizubehalten.

Beide Scripts liefern das gleiche Ergebnis:

Script VBRK_VKORG_NETWR_SUMMEN - 18 Einträge

Tabelle VBRK - Faktura: Kopfdaten

VerkOrg.	Netto (Summe)	Währung
<input type="checkbox"/> 0001	3.000,00	CAD
<input type="checkbox"/> 0001	10.100,00	EUR
<input type="checkbox"/> 0005	570.000,00	EUR
<input type="checkbox"/> 1000	250.774.399,57	EUR
<input type="checkbox"/> 1000	24.881.989,40	USD
<input type="checkbox"/> 2400	32.925.700,00	EUR
<input type="checkbox"/> 2700	7.500,00	CHF
<input type="checkbox"/> 3000	13.241,10	EUR
<input type="checkbox"/> 3000	20.346.489,76	USD
<input type="checkbox"/> 3020	34.272.477,24	USD
<input type="checkbox"/> 6000	102.000,00	MXN
<input type="checkbox"/> 7500	10,00	ARS
<input type="checkbox"/> CPB1	5.385,00	USD
<input type="checkbox"/> CPF1	53,00	USD
<input type="checkbox"/> R100	622.867,24	EUR
<input type="checkbox"/> R300	17.300.807,19	USD
<input type="checkbox"/> SL31	7.475,60	USD
<input type="checkbox"/> TM2	3.140,52	EUR

Wie im ersten Beispiel sind in der Detailansicht einer Ergebniszeile nur die relevanten Felder zu sehen:

Satz von Tabelle VBRK - Faktura: Kopfdaten

Voriger Eintrag Nächster Eintrag

Satz von Tabelle VBRK - Faktura: Kopfdaten

WAERK	'EUR'	Währung des Vertriebsbelegs
VKORG	'1000'	Verkaufsorganisation
V~NETWR_SUM (WAERK)'	250.774.399,57'	Nettowert in Belegwährung (Summe)

Auf der nächsten Seite werden wir ein komplexeres Beispiel mit zwei Datenbanktabellen erörtern.

Beispiel #3 – Min, Durchschnitt, Max and Anzahl

In diesem Beispiel befassen wir uns mit **Einkaufsinfosätzen**. Wir möchten für jeden Lieferanten folgende Informationen sammeln:

- 1) Niedrigsten Nettopreis
- 2) Mittleren Nettopreis
- 3) Höchsten Nettopreis
- 4) Anzahl Infosätze

Das folgende Script liefert die angeforderten Informationen:

- 1) Selektion von Tabelle **EINA** (Einkaufsinfosatz - allgemeine Daten).
- 2) Inner-Join mit Tabelle **EINE** (Einkaufsorganisationsdaten) mit **INFNR** als Join-Kriterium.
- 3) Sortierung der Ergebnisliste nach **A~LIFNR** (Kontonummer des Lieferanten) und **B~WAERS** (Währungsschlüssel von **B~NETPR**).
- 4) *Extras → Spalten einfügen → mit Minimum-Werten* für Feld **B~NETPR** (Nettopreis).
- 5) *Extras → Spalten einfügen → mit Durchschnitts-Werten* für Feld **B~NETPR**.
- 6) *Extras → Spalten einfügen → mit Maximum-Werten* für Feld **B~NETPR**.
- 7) *Extras → Eine Spalte einfügen → mit Anzahl sortierter Sätze* (**A~LIFNR** und **B~WAERS**).
- 8) *Zeilen → Benachbarte Duplikate ausblenden* (**A~LIFNR** und **B~WAERS**).
- 9) *Zeilen → Ausgeblendete Sätze verwerfen*.
- 10) Abschließend wählen wir als Felder für die Ergebnisliste **A~LIFNR**, **B~WAERS** und alle **V-Felder** mit den verschiedenen Werten.

Wir speichern das Script unter dem Namen “**INFSATZ_MIN_MVAL_MAX_CNT**”.

Dieses dritte Script zeigt, dass **mehr als eine** Summier-/Zähl-Operation in einem Zug auf der Datenbank ausgeführt werden kann. Am wichtigsten ist, dass alle beteiligten Felder **das gleiche Referenzfeld** haben müssen. Für die Zählung müssen die gleichen Vergleichskriterien wie für die Summierung verwendet werden. In unserem Beispiel muss der Währungsschlüssel **explizit** für die Zählung der Anzahl der Infosätze verwendet werden.

Das Resultat des oben beschriebenen Scripts ist wie folgt:

Script INFSATZ_MIN_MVAL_MAX_CNT - 403 Einträge

Join von EINA(A) und EINE(B)

A~Lieferant	Nettoprs (PMn)	Nettoprs (PDw)	Nettoprs (PMx)	B-Währung	Zähler
000000002	0,00	5,79	6,00	EUR	33
000000005	100,00	183,33	250,00	USD	3
000000008	0,00	0,00	0,00	MXN	3
000000015	100,00	100,00	100,00	EUR	3
000000100	0,00	495,28	10.226,35	EUR	24
000000111	0,00	26,88	51,10	EUR	4
000000111	23,00	23,00	23,00	USD	1
000000200	1,79	95,39	204,50	EUR	9
000000200	11,00	11,00	11,00	USD	1
000000222	1,00	1,00	1,00	USD	1
000000300	0,00	18,31	127,80	EUR	54
000000300	25,00	25,00	25,00	USD	1
000000954	5,00	7,75	9,00	USD	4
000001000	0,00	459,87	76.692,00	EUR	692
000001000	0,00	0,00	0,00	MXN	1
000001000	0,00	570,06	1.800,00	USD	17
000001001	0,00	2,55	44,99	EUR	48
000001001	2,00	20,08	35,00	USD	6
000001002	0,43	1.196,69	12.270,72	EUR	83
000001002	0,00	38,05	330,00	EUR	87

Es kann natürlich verbessert werden, indem Texte, aussagekräftigere Überschriften und Trennlinien hinzugefügt werden:

Script INFSATZ_MIN_MVAL_MAX_CNT - 403 Einträge

Join von EINA(A) und EINE(B)

Lieferant	Lieferant	Nettopreis (Min)	Nettopreis (Dwert)	Nettopreis (Max)	Währung	Anz. Infosätze
000000002	Electronic Components Distributor - FOSTER CITY	0,00	5,79	6,00	EUR	33
000000005	Safety Clean Inc. - New York	100,00	183,33	250,00	USD	3
000000008	Jose Fernandez - Mexico City	0,00	0,00	0,00	MXN	3
000000015	Tiedemeier Entsorgung GmbH - Berlin	100,00	100,00	100,00	EUR	3
000000100	C.E.B. BERLIN - Berlin	0,00	495,28	10.226,35	EUR	24
000000111	KBB Schwarze Pumpe - Frankenthal/Pfalz	0,00	26,88	51,10	EUR	4
000000111	KBB Schwarze Pumpe - Frankenthal/Pfalz	23,00	23,00	23,00	USD	1
000000200	SMP - ATLANTA	1,79	95,39	204,50	EUR	9
000000200	SMP - ATLANTA	11,00	11,00	11,00	USD	1
000000222	Express Vendor Inc - CHICAGO	1,00	1,00	1,00	USD	1
000000300	AluCast - HILLSBOROUGH	0,00	18,31	127,80	EUR	54
000000300	AluCast - HILLSBOROUGH	25,00	25,00	25,00	USD	1
000000954	Ander Group - ATLANTA	5,00	7,75	9,00	USD	4
000001000	C.E.B. BERLIN - Berlin	0,00	459,87	76.692,00	EUR	692
000001000	C.E.B. BERLIN - Berlin	0,00	0,00	0,00	MXN	1

Beispiel #4 – Zweimal Zählen

In diesem vierten Beispiel zeigen wir, wie die Zählung auf der Datenbank mit einer herkömmlichen Zähl-Operation kombiniert werden kann.

Wir möchten herausfinden, wie viele **transparente Tabellen** im Data Dictionary eine bestimmte Anzahl von Feldern aufweisen, d.h. **wie viele** transparente Tabellen **ein Feld, zwei Felder, drei Felder usw.** haben.

Um unser Ziel zu erreichen, benötigen wir ein Script mit folgenden Operationen:

- 1) Selektion von Tabelle **DD02L** (SAP-Tabellen) mit **AS4LOCAL = 'A'** und **TABCLASS = 'TRANSP'**.
- 2) Inner-Join mit Tabelle **DD03L** (Tabellenfelder) mit **TABNAME** als Join-Kriterium und noch **AS4LOCAL = 'A'** und **FIELDNAME** ungleich **.***.
- 3) Sortierung der resultierenden Join-Liste nach **A~TABNAME**.
- 4) *Extras → Eine Spalte einfügen → mit Anzahl sortierter Sätze* mit Zähl-V-Feld **ANZAHL_FELDER**.
- 5) *Zeilen → Benachbarte Duplikate ausblenden* (**A~TABNAME**).
- 6) *Zeilen → Benachbarte Duplikate ausblenden*.
- 7) Sortierung der Ergebnisliste nach **V~ANZAHL_FELDER**.
- 8) *Extras → Eine Spalte einfügen → mit Anzahl sortierter Sätze* mit Zähl-V-Feld **ANZAHL_TABELLEN**.
- 9) *Zeilen → Benachbarte Duplikate ausblenden* (**V~NR_OF_FIELDS**).
- 10) *Zeilen → Benachbarte Duplikate ausblenden*.
- 11) Als Listenfelder nur **V~ANZAHL_FELDER** und **V~ANZAHL_TABELLEN**.

Wir speichern das Script unter dem Namen **“TRANSP_TABELLEN_STAT”**.

Wie wir sehen werden, werden nur die **ersten sechs Operationen** auf der Datenbank ausgeführt. Die restlichen werden auf herkömmliche Weise durchgeführt, da es in der Datenbank kein **ANZAHL_FELDER** Feld gibt, das gezählt werden kann.

Das Ergebnis dieses Scripts ist wie folgt:

Script TRANSP_TABELLEN_STAT - 227 Einträge

Join von DD02L(A) und DD03L(B)

V~ANZAHL_FELDER	V~ANZAHL_TABELLEN
1	1.317
2	5.767
3	8.567
4	13.392
5	9.717
6	6.520
7	5.020
8	3.808
9	3.104
10	3.084
11	2.722

Es besagt, dass es **1317** transparente Tabellen mit nur **einem Feld**, **5767** transparente Tabellen mit **zwei Feldern** usw. gibt.

Wie bereits erwähnt, wurden nur einige der Operationen direkt auf der Datenbank durchgeführt. Dies kann über folgende Menüfunktion ermittelt werden:

Springen → Operations-Log anzeigen

Verwenden Sie dann auf der resultierenden Liste folgende Menüfunktion:

Script → DB-Summierung/Zählung erklären

Das Operations-Log erklärt nun im Detail, was getan wurde:

SE16XXL - Operations-Log

Op. Op.

Liste des Operations-Log:

SE16XXL mit Script TRANSP_TABELLEN_STAT gestartet

Die farblich hervorgehobenen Operationen sind sinngemäß auf der Datenbank durchgeführt worden

Nr. 1	SELSCREEN	Erste Selektionsmaske
	SEL_MODE=N	TABNAME=DD02L - SEL_WITH_OR= - ALIAS=A
	Selektions-Felder:	TABNAME AS4LOCAL AS4VERS TABCLASS EXCLASS
	Selektions-Kriterien	AS4LOCAL I EQ A TABCLASS I EQ TRANSP
Nr. 2	SELECT	Erste Selektion
	SEL_MODE=N	TABNAME=DD02L - SEL WITH OR= - ALIAS=A

Wenn wir die Liste nach unten scrollen, finden wir die **letzte Operation**, die direkt auf der Datenbank ausgeführt wurde:

SE16XXL - Operations-Log

Op. Op.

Liste des Operations-Log:

Nr. 9	HIDE_ADJ	Benachbarte Duplikate ausblenden	
	SEL_MODE=	TABNAME=	SEL_WITH_OR= ALIAS=B
	Comparing-Kriterien: A-TABNAME		
	Betroffene Tabellen: (A)DD02L (B)DD03L		
Nr. 10	DISC_HIDE	Ausgeblendete Sätze verwerfen	
	SEL_MODE=	TABNAME=	SEL_WITH_OR= ALIAS=B
	Betroffene Tabellen: (A)DD02L (B)DD03L		
Nr. 11	CHOOSE_LF	Felder für Ausgabeliste auswählen	
	SEL_MODE=	TABNAME=	SEL_WITH_OR= ALIAS=B
	List-Felder:	A-TABNAME V-ANZAHL_FELDER	
	Betroffene Tabellen: (A)DD02L (B)DD03L		
Nr. 12	SORT	Liste sortieren	
	SEL_MODE=	TABNAME=	SEL_WITH_OR= ALIAS=B
	Sort-Kriterien:	V-ANZAHL_FELDER A	
	Betroffene Tabellen: (A)DD02L (B)DD03L		
Nr. 13	COUNT_ROWS	Spalte m. Anzahl sortierter Sätze einf.	
	SEL_MODE=	TABNAME=	SEL_WITH_OR= ALIAS=B
	Comparing-Kriterien: V-ANZAHL_FELDER		
	V-Felder:	ANZAHL_TABELLEN TYPE /TFTO/TX_ST_STDV-VCOUNT	
	Betroffene Tabellen: (A)DD02L (B)DD03L		
Nr. 14	CHOOSE_LF	Felder für Ausgabeliste auswählen	

ANMERKUNG: Die Zahlen, die jede Operation identifizieren, entsprechen nicht denen, die wir zu Beginn dieses Beispiels verwendet haben. Der Grund dafür ist, dass SE16XXL zusätzliche Operationen wie SELSCREEN oder CHOOSE_LF in das Log einfügt.

WICHTIG: Die Menüfunktion *Script* → *DB-Summierung/Zählung erklären* im Operations-Log kann auch dazu verwendet werden, **um herauszufinden**, warum ein bestimmtes Script **nicht direkt auf der Datenbank durchgeführt wurde**.

Unser nächstes Beispiel wird eine solche Situation veranschaulichen.

Beispiel #5 – Etwas funktioniert nicht

In diesem Beispiel möchten wir einige Durchschnittswerte bezüglich der Materialien im Materialstamm (MARA) erhalten. Um jedoch zu zeigen, wie man mögliche Probleme erkennt, funktioniert das Script zunächst nicht wie gewünscht und muss ein paar Mal korrigiert werden.

Wir beginnen unser Beispiel mit folgendem Script:

- 1) Selektion von Tabelle MARA (Materialstamm) ohne Einschränkungen.
- 2) Sortierung der resultierenden Liste nach MTART (Materialart).
- 3) *Extras → Spalten einfügen → mit Durchschnitts-Werten* für Feld VOLUM (Volumen).
- 4) *Extras → Spalten einfügen → mit Durchschnitts-Werten* für Feld BRGEW (Bruttogewicht).
- 5) *Zeilen → Benachbarte Duplikate ausblenden* (MTART).
- 6) *Zeilen → Ausgeblendete Sätze verwerfen*.
- 7) Für die Ergebnisliste wählen wir nur die relevanten Felder aus.

Wir speichern das Script als “MARA_D_WERTE”.

Wenn das Script mit der Option “*DB-Join/Summen verwenden wenn möglich*” ausgeführt wird, wird die erwartete Meldung

```
**** DB-Summieren/Zählen verwendet ****
```

NICHT ausgegeben.

Was ist falsch an unserem Script? Um dies herauszufinden, konsultieren wir das Operations-Log wie bereits in **Beispiel #4** beschrieben:

Springen → Operations-Log anzeigen

An dieser Stelle wählen wir dann die Menüfunktion:

Script → DB-Summierung/Zählung erklären

In der Operations-Log-Liste finden wir die Ursache:

SE16XXL - Operations-Log

Op. Op.

Liste des Operations-Log:

SE16XXL mit script MARA_D_WERTE gestartet

Die Summierungs-/Zählungs-Operationen konnten aus folgendem Grund nicht auf der Datenbank durchgeführt werden:
Die Summierungs-Operationen verwenden unterschiedliche Referenzfelder !
Die betroffene Operation ist farblich hervorgehoben worden.

Nr. 1 SELSCREEN Erste Selektionsmaske
SEL_MODE=N - TABNAME=MARA - SEL_WITH_OR= - ALIAS=A

Durch Scrollen nach unten lokalisieren wir schnell den Übeltäter:

Betroffene Tabellen: MARA

Nr. 6 ADD_MVALS Spalten mit Durchschnittswerten einfügen
SEL_MODE= - TABNAME=MARA - SEL_WITH_OR= - ALIAS=A
Comparing-Kriterien: MTART
V-Felder: BRGEW=>V-BRGEW_D_WERT
Betroffene Tabellen: MARA

Nr. 7 CHOOSE_LF Felder für Ausgabeliste auswählen
SEL MODE= - TABNAME=MARA - SEL WITH OR= - ALIAS=A

Das Problem ist, dass VOLUM und BRGEW unterschiedliche Referenzfelder aufweisen, d.h. VOLEH und GEWEI.

Wenn wir die Operationen wirklich auf der Datenbank durchführen wollen, müssen wir auf einen der beiden Durchschnitts-Werte verzichten. Wir entscheiden uns, den Durchschnitts-Wert für **VOLUM** beizubehalten.

Mit Hilfe des Script-Editors entfernen wir die umstrittene Operation und führen anschließend das Script erneut aus.

Zu unserer Enttäuschung **taucht** die ersehnte Meldung jedoch **immer noch nicht auf**. Offenbar haben wir nicht alle Probleme aus unserem Script eliminiert.

Wir machen erneut von der Erklär-Funktion Gebrauch.

Diesmal lautet die Erklärung wie folgt:

SE16XXL - Operations-Log

Liste des Operations-Log:

SE16XXL mit Script MARA_D_WERTE_2 gestartet

Die Summierungs-/Zählungs-Operationen konnten aus folgendem Grund nicht auf der Datenbank durchgeführt werden:
Das Referenzfeld der Summen ist kein Comparing-Kriterium !
Die betroffene Operation ist farblich hervorgehoben worden.

Nr.	1	SELSCREEN	Erste Selektionsmaske
		SEL_MODE=N	- TABNAME=MARA - SEL_WITH_OR= - ALIAS=A

Die fragliche Operation ist schnell gefunden:

Liste der Operationen:

Betroffene Tabellen: MARA

Nr.	6	HIDE_ADJ	Benachbarte Duplikate ausblenden
		SEL_MODE=N	- TABNAME=MARA - SEL_WITH_OR= - ALIAS=A
		Comparing-Kriterien:	MTART
		Betroffene Tabellen:	MARA

Nr.	7	DISC_HIDE	Ausgeblendete Sätze verwerfen
		SEL_MODE=N	- TABNAME=MARA - SEL_WITH_OR= - ALIAS=A

Um unser Script zu korrigieren, müssen wir das Referenzfeld **VOLEH** zu den Sortierkriterien hinzufügen, entweder durch Ändern der anfänglichen Sortierung oder durch Einfügen einer **SORT**-Operation unmittelbar vor der Operation **Benachbarte Duplikate ausblenden**. Deren Comparing-Kriterien müssen ebenfalls angepasst werden.

Wir versuchen unser Glück noch einmal, hoffentlich zum letzten Mal.
Und tatsächlich, diesmal die wunderbare Meldung

**** DB-Summieren/Zählen verwendet ****

kommt tatsächlich zum Vorschein. Unsere Bemühungen waren erfolgreich.

Das obige Beispiel wirft die folgende Frage in Bezug auf Referenzfelder auf:
Warum sollten alle beteiligten Summen usw. das gleiche Referenzfeld haben?

Die Antwort ist nicht so offensichtlich. Es kann sich daher lohnen, dieses Thema etwas näher zu erläutern. Für unsere Diskussion betrachten wir den Materialstamm (MARA) und die Durchschnitts-Werte von VOLUM (Volumen) und BRGEW (Bruttogewicht), so wie wir es zu Beginn von Beispiel #5 tun wollten. Aus Gründen der Übersichtlichkeit beschränken wir uns auf die Materialart FERT.

Wir beginnen damit, die beiden Durchschnitts-Werte getrennt zu berechnen. Die beiden Ergebnisse lauten wie folgt:

Volumen D-Werte

MTART	V~VOLUM_D_WERT	VOLEH
FERT	5	CCM
FERT	1	CDM
FERT	70,589	FT3
FERT	0,264	GLL
FERT	41,800	IN3
FERT	6,948	L
FERT	914,971	M3
FERT	1.550,000	ML
FERT	54.000	MMQ

Bruttogewicht D-Werte

MTART	V~BRGEW_D_WERT	GEWEI
FERT	189,743	G
FERT	1.449,753	KG
FERT	289,833	LB
FERT	120	MG
FERT	3,188	OZ
FERT	1,439	TO

Angenommen, man würde beide Operationen direkt auf der Datenbank ausführen, nachdem nur **eine GROUP BY-Klausel verwendet werden kann**, müssten **MTART**, **VOLEH** und **GEWEI** zusammen erwähnt werden. Das Ergebnis wäre wie folgt:

MTART	V~VOLUM_D_WERT	VOLEH	V~BRGEW_D_WERT	GEWEI
FERT			225,041	G
FERT			384,834	KG
FERT			122,322	LB
FERT			3,188	OZ
FERT			2	TO
FERT	5	CCM	50	G
FERT	1	CDM	1	KG
FERT	239,071	FT3	24,786	KG
FERT	15,093	FT3	423,574	LB
FERT	1	FT3	0,598	TO
FERT	0,264	GLL	1	KG
FERT	44,141	IN3	5,294	KG
FERT	2	IN3	100	LB
FERT	0,957	L	111,176	G
FERT	8,352	L	3,519	KG
FERT	10	L	10,600	LB
FERT	914,971	M3	2.872,488	KG
FERT	100,000	ML	0,450	KG
FERT	3.000,000	ML	120	MG
FERT	54.000	MMQ	23	G

Wenn wir versuchen, den Durchschnitts-Wert für VOLEH = 'FT3' (70,589) zu finden, werden wir ihn nicht finden. Stattdessen gibt es **drei** verschiedene Werte für **drei** verschiedene GEWEI-Einheiten. Dieser Effekt ist auf die gleichzeitige Sortierung durch VOLEH und GEWEI zurückzuführen. Die Durchschnitts-Werte werden somit **für jede Gewichtseinheit separat** berechnet.

Die gleichen Überlegungen gelten für die Durchschnitts-Werte des Bruttogewichts.

Um keine irreführenden Ergebnisse zu produzieren, ist die Summierung auf der Datenbank auf ein Referenzfeld beschränkt.

Wir können nun die **Bedingungen** für das Summieren/Zählen in Detail auflisten.

Bedingungen für das Summieren/Zählen auf der Datenbank

Um die Summierungs-/Zähl-Funktionalität der Datenbank nutzen zu können, muss ein Script folgende Bedingungen erfüllen:

- 1) Die erste Tabelle des Scripts muss transparent sein.
- 2) Nach der ersten Selektion kann eine Reihe von Inner-Joins folgen. Die beteiligten Tabellen müssen ebenfalls transparent sein. Die Inner-Joins sollten nicht virtuell durchgeführt werden.
- 3) Die Join-Kriterien sollten weder partielle oder verkettete Felder enthalten. Und auch keine speziellen Operatoren oder Gültigkeitsprüfungen. In anderen Worten sollten die Join-Kriterien “normal” sein.
- 4) Eine SORT-Operation muss der ersten Selektion oder Reihe von Inner-Joins folgen.
- 5) Nach der SORT-Operation müssen eine oder mehrere der folgenden Operationen vorhanden sein:
 - *Extras → Eine Spalte einfügen → mit Anzahl sortierter Sätze*
 - *Extras → Spalten einfügen → mit (Zwischen)Summen*
 - *Extras → Spalten einfügen → mit Durchschnitts-Werten*
 - *Extras → Spalten einfügen → mit Minimum-Werten*
 - *Extras → Spalten einfügen → mit Maximum-Werten.*
- 6) Zwischen den obengenannten Operationen sind nur “*Sort*” oder “*Felder für Ausgabeliste auswählen*” Operationen zulässig.
- 7) Die Operationen mit Währungsbeträgen oder Mengen müssen alle **dasselbe Referenzfeld** verwenden (d. h. Währungs- oder Einheitenschlüssel).
- 8) Die Comparing-Kriterien der obengenannten Operationen sollten keine partiellen Felder enthalten. Sie müssen dieselben Felder adressieren (mit Ausnahme des Referenzfeldes), ggf. in einer anderen Reihenfolge. Die Comparing-Kriterien einer Summierungs-Operation können wahlweise das Referenzfeld enthalten. Falls eine Zähl-Operation vorhanden ist und es gibt Summierungs-Operationen mit Referenzfeld, dann müssen die Comparing-Kriterien der Zähl-Operation das Referenzfeld enthalten. Dasselbe gilt für eine Summierungs-Operation ohne Referenzfeld.

- 9) Auf den Zähl-/Summierungs-Operationen muss eine “**Benachbarte Duplikate ausblenden**” Operation mit den gleichen Comparing-Kriterien folgen.
- 10) Wenn die Summierungs-Operationen ein Referenzfeld (Währung/Einheit) beinhalten, muss dieses Referenzfeld ein Comparing-Kriterium der “**Benachbarte Duplikate ausblenden**” Operation sein. Ist dies nicht der Fall, muss zu diesem Zweck eine SORT-Operation eingefügt werden.
- 11) Auf die Operation “**Benachbarte Duplikate ausblenden**” muss eine Operation “**Ausgeblendete Sätze verwerfen**” folgen.
- 12) Die Operationen, die nach “**Ausgeblendete Sätze verwerfen**” kommen, sollten nur verfügbare Felder adressieren.
Diese Bedingung bezieht sich auf die Felder der transparenten Tabellen, die selektiert werden, bevor die Summierung/Zählung durchgeführt wird.
Es stehen nur die Felder zur Verfügung, die als Comparing-Kriterien verwendet werden.
Die Operation “**Felder für Ausgabeliste auswählen**” ist ausgenommen - nicht verfügbare Felder werden in diesem Fall ignoriert.
- 13) Die erste Selektion des Scripts sollte keine große Anzahl einzelner Selektionswerte (große Selektion) enthalten.

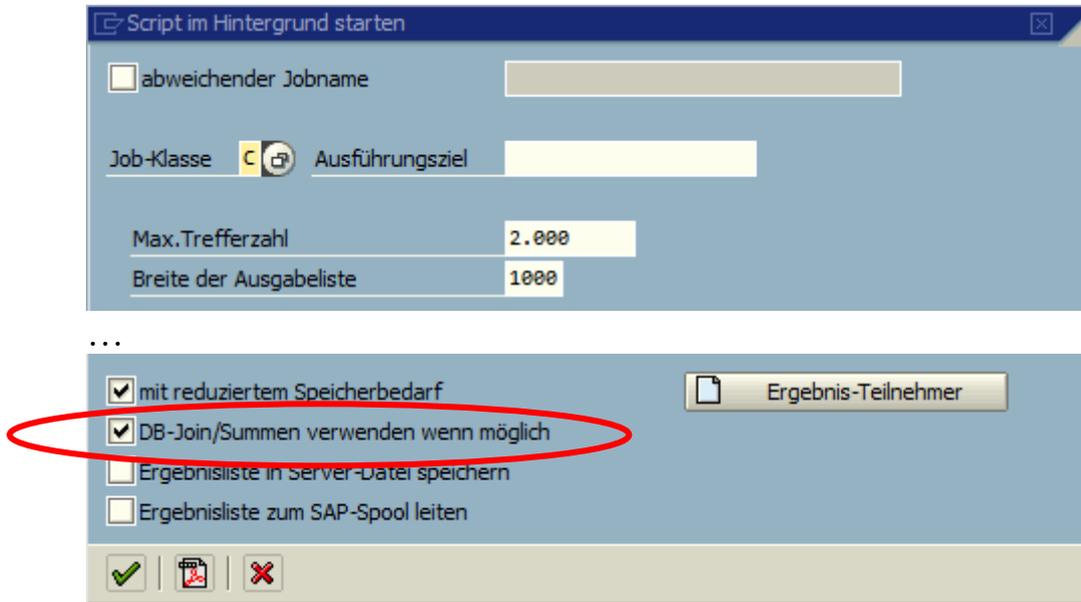
Zusätzliche Anmerkungen

Nachdem die Datenbank nur Sätze mit aggregierten Werten zurückliefert, hat SE16XXL keine Möglichkeit auf die einzelnen Tabellen-Einträge zuzugreifen. Aus diesem Grund gelten folgende Anmerkungen:

- Falls **Berechtigungsprüfungen** für die betroffenen Tabellen aktiv sind und Sie sind nicht mit einer generellen Berechtigung (alle Werte = ‘*’) für die relevanten Prüfungen ausgestattet, dann benötigen Sie eine spezielle Rolle (/TFTO/XXL_SCR_DB_SUMS_AUTH).
- Falls **spezielle Berechtigungsprüfungen** für die betroffenen Tabellen aktiv sind, benötigen Sie die Rolle /TFTO/XXL_SCR_DB_SUMS_SAC.
- Berechtigungsprüfungen werden nur für die vorhandenen Felder durchgeführt.
- Dasselbe gilt für spezielle Berechtigungsprüfungen.
- Die maximale Trefferzahl bezieht sich auf die aggregierten Sätze.

Scripts im Hintergrund

Die Option “**DB-Join/Summen verwenden wenn möglich**” steht auch für im Hintergrund ausgeführte Scripts zur Verfügung, wie im entsprechenden Dialogfenster zu sehen ist:



Wenn der Job beendet ist, zeigt das Job-Protokoll an, ob die Operationen tatsächlich auf der Datenbank ausgeführt wurden:

Datum	Uhrzeit	Nachrichtentext
23.10.2021	10:48:50	Job wurde gestartet
23.10.2021	10:48:50	Step 001 gestartet (Programm /TFTO/TX_BATCH_SCRIPT_X, Variante &0000000004869,
23.10.2021	10:48:51	Globales Script \$MARA_VOLUM_D_WERT wird ausgeführt
23.10.2021	10:48:51	Option "mit reduziertem Speicherbedarf" ist aktiv
23.10.2021	10:48:51	Die Datenbank-Zählungs-/Summierungs-Funktionalität wird verwendet
23.10.2021	10:48:51	Job wurde beendet

ANMERKUNG: Die zuvor erwähnte **Erklär**-Funktion ist auch für die Ergebnisliste von Hintergrund-Scripts verfügbar. Es gibt jedoch nur die **positive** Erklärung (d.h. ob die Operationen tatsächlich auf der Datenbank durchgeführt wurden). Eine negative Erklärung (d.h. warum es nicht funktioniert hat) wird im Gegenteil nicht gegeben. Der Grund dafür ist, dass sich in der Zwischenzeit (Hintergrundergebnisse können sehr alt sein) die gesamte Logik geändert haben könnte und ein bestimmter Grund zum aktuellen Zeitpunkt obsolet sein könnte.