SELECT - Der Grundbefehl zur Auswahl von Daten

Die SELECT-Anweisung fragt Daten aus einer Datenbank ab und stellt diese in einer virtuellen Tabelle zur Verfügung. Diese virtuelle Tabelle, eine Menge von Datensätzen existiert zunächst nur temporär im Arbeitsspeicher und wird nach dem Ende der Befehlsausführung verworfen.

Die Ausgabe kann aber auch mit SELECT ... INTO ... FROM in eine neue Tabelle kopiert oder mit INSERT INTO ... SELECT ... zu einer bestehenden Tabelle hinzugefügt werden.

Grundlegender Syntax → [Befehle in eckigen Klammern sind optional]

```
SELECT      [DISTINCT]
            < Spaltenname>
            <Konstante>
            <Berechnung>
            <einer der obigen Ausdrücke> AS Spaltenalias

FROM        <Tabellenname> AS Tabellenalias

[WHERE ...]

[GROUP BY ...]

[HAVING ...]

[ORDER BY   <Spaltenname> [ASC | DESC] ]
```

Details zu den Ausdrücken nach FROM, WHERE, GROUP BY und HAVING finden Sie in den Abschnitten über JOIN, WHERE und GROUP BY.

Die Beispiele unten behandeln nur den Abschnitt zwischen SELECT und FROM.

```
SELECT      <Spaltenname ... - wie oben >

[INTO        <neue Tabelle>]

FROM        <Tabellenname> AS Tabellenalias
```

Geben Sie für alle Beispiele das konkrete Ergebnis der Abfrage an!

Beispiele:

1	SELECT A_NR, A_NAME, A_PREIS FROM ARTIKEL;	Einfache, kommagetrennte Auflistung der gewünschten Spalten, die im Tabellen-Ausdruck vorkommen. Ohne Alias für die Tabelle.
2	SELECT A.A_NR, A.A_NAME, A.A_PREIS FROM ARTIKEL AS A ORDER BY A.A_PREIS;	Dasselbe wie im ersten Beispiel, aber mit Aliasname A für die Tabelle und Sortierung nach der Spalte A_PREIS.
3	SELECT A_NR, A_PREIS AS Netto, 0.19 AS MwSt, A_PREIS * 1.19 AS Brutto FROM ARTIKEL ORDER BY A_PREIS DESC;	Es werden Alias-Ausdrücke für die Spaltennamen verwendet - das Ergebnis gibt drei Spalten Netto, MwSt und Brutto aus. MwSt enthält einen konstanten Wert, in Brutto wird der Inhalt von A_PREIS multipliziert mit einer Konstanten. Die Ausgabe wird nach A_PREIS absteigend sortiert.
4	SELECT A.* FROM ARTIKEL AS A ORDER BY A.A_NAME ASC, A.A_PREIS DESC;	Hier wird für die Tabelle ein Alias A verwendet und mithilfe von * alle Spalten ausgewählt. Das Ergebnis wird aufsteigend nach Artikel-Name und absteigend nach Artikel-Preis sortiert.
5	SELECT DISTINCT A.A_NAME FROM ARTIKEL AS A;	Das Schlüsselwort DISTINCT entfernt alle mehrfach vorkommenden Zeilen mit Ausnahme einer. Die Abfrage liefert deshalb nur drei Zeilen zurück, eine Zelle mit dem Wert 'Oberhemd' wurde entfernt.
6	SELECT V.* INTO KOPIEVONVERTRETER FROM VERTRETER AS V;	INTO speichert die virtuelle Tabelle in ein reales neues Tabellenobjekt. Es wird die neue Tabelle 'KOPIEVONVERTRETER' erstellt und mit den Zeilen gefüllt, die von der SELECT-Anweisung zurückgegeben wurden.

Bemerkungen

- Verwenden Sie Alias-Namen für Tabellen, damit die Spalten einfacher identifiziert werden.
- Wenn der Spalten-, Alias- oder Tabellenname Sonderzeichen, etwa ein Leerzeichen oder Minus (-) enthält, so schließen Sie den Ausdruck in eckige Klammern ein.
  Beispiel: Select A.A_PREIS * 1.19 AS [Aktueller Bruttopreis] FROM ARTIKEL
- Soll ein konstanter Text angegeben werden, so setzen Sie diesen in einfache Hochkommata. Beispiel: SELECT A.A_NAME, 'aktuell' AS Info FROM ARTIKEL AS A

Mit WHERE einzelne Datensätze nach Bedingungen aus der Datenbank auswählen

Die SELECT-Anweisung liefert eine virtuelle Tabelle, bestehend aus Zeilen und Spalten zurück. Der nach FROM folgende WHERE-Abschnitt kann Spaltennamen verwenden, um Bedingungen festzulegen. Für jede Zeile wird geprüft, ob die durch den Spaltennamen festgelegte Zelle die Bedingung erfüllt. Falls dies der Fall ist, wird die Zeile zur Ausgabe hinzugefügt.

Syntax

```
SELECT ...
FROM ...
WHERE    <Bedingung 1> [<logischer Operator> <Bedingung 2>]
```

Es können mehrere Bedingungen angegeben und mit logischen Operatoren (NOT, AND, OR) miteinander verknüpft werden.

Zulässige Vergleichsoperatoren:

=	Gleichheit
<>	verschieden
<	kleiner
>	größer
<=	kleiner oder gleich
>=	größer oder gleich
IS NULL	prüft, ob die Zelle leer ist
IS NOT NULL	prüft, ob die Zelle einen Wert enthält
BETWEEN	zwischen zwei Werten liegend
IN	prüft, ob der linke Ausdruck in einem der rechten vorkommt
LIKE	vergleicht Strings
EXISTS	prüft, ob die folgende Unterabfrage mindestens eine Zeile zurückliefert

Zulässige logische Operatoren:

NOT	der folgende Ausdruck darf nicht erfüllt sein
AND	beide Bedingungen müssen erfüllt sein
OR	mindestens eine Bedingung muss erfüllt sein

Geben Sie für alle Beispiele das konkrete Ergebnis der Abfrage an!

Beispiele:

1	Select A.* FROM ARTIKEL AS A WHERE A.A_NR = 11;	Wählt die Zeile aus, bei welcher die Zelle A_NR den Wert 11 hat.
2	Select A.* FROM ARTIKEL AS A WHERE NOT A.A_NR = 11;	Wählt alle Zeilen aus, deren A_NR verschieden von 11 aber auch nicht leer (Not Null) ist *Beachten Sie:* *gleichwertig zu WHERE A.A_NR <> 11*

3	Select A.* FROM ARTIKEL AS A WHERE A.A_PREIS IS NULL;	Wählt die Zeilen ohne Preis aus. *Beachten Sie:* *Zelle mit dem Zahlenwert 0 ist nicht leer* *Zelle mit Zeichenkette "" ist nicht leer*
4	Select A.* FROM ARTIKEL AS A WHERE A.A_NAME = 'Oberhemd' AND A.A_PREIS < 40.00;	Wählt die Zeilen aus, bei denen sowohl der Wert in A_NAME gleich 'Oberhemd' als auch der Wert in A_PREIS kleiner als 40.00 ist.
5	Select A.* FROM ARTIKEL AS A WHERE A.A_NAME = 'Oberhemd' OR A.A_PREIS > 40.00;	Wählt die Zeilen aus, bei denen der Wert in A_NAME gleich 'Oberhemd' oder der Wert in A_PREIS größer als 40.00 ist.
6	Select A.* FROM ARTIKEL AS A WHERE A.A_PREIS BETWEEN 39.8 AND 100.00;	Wählt die Zeilen aus, deren Preis zwischen 39.8 und 100.00 liegt. Die Randwerte werden mitgezählt. *Beachten Sie: Das AND gehört zum BETWEEN und hat hier keine logische Bedeutung.*
7	Select A.* FROM ARTIKEL AS A WHERE 39.8 <= A.A_PREIS AND A.A_PREIS <= 100;	Diese Version ist gleichwertig zur vorherigen Version.
8	Select A.* FROM ARTIKEL AS A WHERE A.A_NAME IN ('Hose', 'Mantel', 'Strümpfe');	Vergleicht den Zelleninhalt mit jedem der in der Klammer angegebenen Werte. Stimmt er mit einem überein, so wird die Zeile ausgegeben. Es können Konstante, berechnete Werte oder das Ergebnis einer Abfrage angegeben werden. *Beachten Sie: Der Ausdruck ist gleichwertig zu WHERE A.A_NAME = 'Hose' OR A.A_NAME = 'Mantel' OR A.A_NAME = 'Strümpfe'*
9	SELECT V.* FROM VERTRETER AS V WHERE V.V_NAME LIKE 'Me_er, Franz' WHERE V.V_NAME LIKE 'Me%' WHERE V.V_NAME CONTAINS 'Me';	Mit LIKE werden Textmuster geprüft. Unterstrich _ ist Platzhalter für ein einzelnes Zeichen. Prozentzeichen % ist Platzhalter für kein oder mehrere Zeichen. CONTAINS prüft, ob eine Zeichenkette im Feldinhalt enthalten ist.
10	SELECT A.A_NR FROM ARTIKEL AS A WHERE EXISTS (SELECT B.UMSATZ_NR FROM UMSATZ AS B WHERE B.A_NR = A.A_NR);	EXISTS prüft, ob eine Unterabfrage Werte enthält. EXISTS bricht ab, wenn eine Zeile gefunden wurde.

Bemerkungen

- Zur Suche nach ganzen Zahlen werden diese direkt notiert. Für Dezimalzahlen ist der Punkt das Trennzeichen. Texte werden in Hochkommata (') gesetzt.
- Operatorreihenfolge: Ein Vergleich mit = bindet am stärksten, so dass ein Ausdruck A_NAME = 11 OR A_PREIS > 100 so interpretiert wird: (A_NAME = 11) OR (A_PREIS > 100) Verwenden Sie verschiedene logische Operatoren in einem WHERE-Ausdruck, so nutzen Sie am besten Klammern.

JOIN - Normalisierte Tabellen für eine Abfrage wieder zusammenführen

Eine Datenbank wurde normalisiert, also der Datenbestand in verschiedene Tabellen aufgeteilt. Sollen nun Daten aus verschiedenen Tabellen einander zugeordnet werden. leistet das die JOIN-Verknüpfung. Ein JOIN fügt zwei Tabellen zu einer neuen, virtuellen Tabelle zusammen.

Syntax

Komma-Version

FROM <Tabellenname> , <Tabellenname>

Besser! → Einfachste Form des INNER JOIN

FROM <Tabellenname> INNER JOIN <Tabellenname>

ON <Spaltenname1> OPERATOR < Spaltenname2>

Eine Tabelle kann auch zweimal angegeben werden, man spricht von einer Selbstverknüpfung, da Zeilen der Tabelle zu anderen Zeilen der Tabelle in Beziehung gesetzt werden

Geben Sie für alle Beispiele das konkrete Ergebnis der Abfrage an!

Beispiele

1	SELECT A.*, U.* FROM ARTIKEL AS A, UMSATZ AS U WHERE U.A_NR = A.A_NR;	Der Join kombiniert jede Zeile der Tabelle ARTIKEL mit jeder Zeile der Tabelle UMSATZ und gibt das Ergebnis aus 4 * 9 = 36 Zeilen vollständig aus … also sinnlos. Durch WHERE werden dann nur relevante Datensätze angezeigt.
2	SELECT A.A_NR, A.A_NAME, A.A_PREIS,U.DATUM, U.A_STUECK, A.A_PREIS * U.A_STUECK AS Preis FROM UMSATZ AS U INNER JOIN ARTIKEL AS A ON U.A_NR = A.A_NR;	Dies ist die einfachste und am häufigsten genutzte Form des JOIN. Durch die in der ON-Klausel festgelegte Bedingung werden die Datensätze gefiltert.

Bemerkungen

Falls Sie Tabellen verknüpfen, so sollten Sie im Regelfall die 'alte' Technik, welche die Klammerversion nutzt und die ON-Klausel in die WHERE-Bedingung verschiebt vermeiden. Vergleichen Sie die beiden folgenden Darstellungen:

SELECT A.*, U.* FROM ARTIKEL AS A, UMSATZ AS U WHERE A.A_NR = U.A_NR AND A.A_NR < 13;	SELECT A.*, U.* FROM ARTIKEL AS A INNER JOIN UMSATZ AS U ON A.A_NR = U.A_NR WHERE A.A_NR < 13;

Mit Aggregat-Funktionen und GROUP BY Daten auswerten

Rückgabe einer einzelnen Zelle - einfache Aggregation

```
SELECT      <Aggregatfunktion(Spaltenname) As Spaltenname>
FROM ...
[WHERE ...]
 [WHERE ...]
GROUP BY    Spaltenname,
```

Geben Sie für alle Beispiele das konkrete Ergebnis der Abfrage an!

Beispiele

1	SELECT COUNT(U.DATUM) AS ZAHLDERUMSÄTZE FROM UMSATZ AS U;	Sind alle Zellen einer Tabelle belegt, so ist es unerheblich, welche Spalte Sie verwenden. Alle nichtleeren Zellen werden gezählt.
2	SELECT U.DATUM, COUNT(U.DATUM) AS ZAHLDERUMSÄTZEPROTAG FROM UMSATZ As U GROUP BY U.DATUM;	Sie erhalten als Ergebnis zwei Zeilen, für jede der unterschiedlichen Datumsangaben eine Zeile. Jede Zeile enthält die zusätzliche Zelle 'ZAHLDER-UMSÄTZEPROTAG' .
3	SELECT MAX(A.A_PREIS) AS TEUERSTERARTIKEL FROM ARTIKEL As A;	Dies liefert Ihnen den Preis des teuersten Artikels (360.00).

Tabellen im Anhang

Beispieldatenbank aus drei Tabellen bestehend

Artikel

A_NR	A_NAME	A_PREIS
12	Oberhemd	39,80
22	Mantel	360,00
11	Oberhemd	44,20
13	Hose	110,50

Vertreter

V_NR	V_NAME	V_ANSCH	V_PROV	V_KONTO
8413	Meyer, Emil	Wendeweg 10, 2800 Bremen	0,07	725,15
5016	Meier, Franz	Kohlstr. 1, 2800 Bremen	0,05	200,00
1215	Schulze, Fritz	Gemüseweg 3, 2800 Bremen	0,06	50,50

Umsatz

UMSATZ_NR	V_NR	A_NR	A_STUECK	DATUM
1	8413	12	40	24.06.99
2	5016	22	10	24.06.99
3	8413	11	70	24.06.99
4	1215	11	20	25.06.99
5	5016	22	35	25.06.99
6	8413	13	35	24.06.99
7	1215	13	5	24.06.99
8	1215	12	10	24.06.99
9	8413	11	20	25.06.99

Zusammengestellt von J. Rau 02/2015

Beispieldatenbank aus drei Tabellen bestehend

Artikel

A_NR	A_NAME	A_PREIS
12	Oberhemd	39,80
22	Mantel	360,00
11	Oberhemd	44,20
13	Hose	110,50

Vertreter

V_NR	V_NAME	V_ANSCH	V_PROV	V_KONTO
8413	Meyer, Emil	Wendeweg 10, 2800 Bremen	0,07	725,15
5016	Meier, Franz	Kohlstr. 1, 2800 Bremen	0,05	200,00
1215	Schulze, Fritz	Gemüseweg 3, 2800 Bremen	0,06	50,50

Umsatz

UMSATZ_NR	V_NR	A_NR	A_STUECK	DATUM
1	8413	12	40	24.06.99
2	5016	22	10	24.06.99
3	8413	11	70	24.06.99
4	1215	11	20	25.06.99
5	5016	22	35	25.06.99
6	8413	13	35	24.06.99
7	1215	13	5	24.06.99
8	1215	12	10	24.06.99
9	8413	11	20	25.06.99