

Die Aufgaben erläutern in aufsteigendem Niveau die Anwendung des SQL-Befehls SELECT.

Sie beziehen sich auf die Übungsdatenbank ,05_2SQL_Erlaeuterungen_und_Uebungen', die Sie im Klassenordner finden.

Mit SELECT – FROM werden Attribute (Spalten) aus einer Tabelle aus der Datenbank ausgegeben.

Die SELECT- FROM Anweisung liefert eine virtuelle Tabelle, bestehend aus Zeilen und Spalten zurück.

SELECT * gibt alle Spalten der Tabelle aus. Der ORDER BY -Abschnitt kann in Abhängigkeit der Werte einer Spalte die Datensätze aufsteigend (**ASC**) oder absteigend (**DESC**) geordnet ausgegeben.

SELECT	Spalte_1, Spalte_2, ...	* bedeutet alle Spalten
FROM	Tabellen	
ORDER BY	Spalte_n (Optional)	

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

1	SELECT A_NR, A_NAME, A_PREIS FROM ARTIKEL;	Kommagetrennte Auflistung der gewünschten Spalten, die in der Tabellen-Ausgabe 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;	Dieselbe Abfrage wie im ersten Beispiel, aber mit Aliasname A (Abkürzung) für die Tabelle ARTIKEL und der 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. Die Abfrage gibt drei Spalten aus: Netto, MwSt und Brutto. MwSt enthält einen konstanten Wert, in Brutto wird der Inhalt von A_PREIS multipliziert mit 0.19. 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;	Für die Tabelle wird ein Alias A verwendet und mithilfe von * alle Spalten ausgewählt . Das Ergebnis wird aufsteigend nach Artikel-Name und bei gleichen Artikel-Namen absteigend nach dem 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 Zeile mit dem Wert 'Oberhemd' wurde entfernt.

Mit WHERE einzelne Datensätze nach festgelegten Bedingungen auswählen

Der WHERE-Abschnitt kann Bedingungen festzulegen für die Anzeige von Datensätzen festlegen. Für jede Zeile wird geprüft, ob die durch den Spaltennamen gegebene Zelle die Bedingungen erfüllt.

```
SELECT   Spalte_1, Spalte_2, . . .
FROM     Tabellen
WHERE    Bedingung 1 Vergleichsoperator Bedingung 2 ...
```

Es können mehrere Bedingungen mit logischen Operatoren (**NOT**, **AND**, **OR**) 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
```

Zulässige logische Operatoren:

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

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 <i>gleichwertig zu WHERE A.A_NR <> 11</i>
3	Select A.* FROM ARTIKEL AS A WHERE A.A_PREIS IS NULL;	Wählt die Zeilen ohne Preis aus. <i>Zelle mit dem Zahlenwert 0 ist nicht leer Zelle mit Zeichenkette "" ist nicht leer</i>
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' und gleichzeitig 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 oder beides.

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. <i>Das AND gehört zum BETWEEN und hat hier keine logische Bedeutung.</i>
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 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. <i>Der Ausdruck ist gleichwertig zu WHERE A.A_NAME = 'Hose' OR A.A_NAME = 'Mantel' OR A.A_NAME = 'Strümpfe'</i>
9	SELECT V.* FROM VERTRETER AS V - WHERE V.V_NAME LIKE 'Me_er, Franz' oder - WHERE V.V_NAME LIKE 'Me%' oder - WHERE V.V_NAME CONTAINS 'Me';	Mit LIKE werden Textmuster geprüft. - Unterstrich _ ist Platzhalter für ein einzelnes Zeichen. - % ist Platzhalter für kein oder mehrere Zeichen. - CONTAINS prüft, ob eine Zeichenkette im Feldinhalt enthalten ist.

JOIN - Normalisierte Tabellen für eine Abfrage wieder zusammenführen

Sollen Daten aus normalisierten Tabellen einander zugeordnet werden, macht das ein JOIN.

Ein JOIN fügt zwei Tabellen zu einer neuen, virtuellen Tabelle zusammen.

FROM Tabellen_1 INNER JOIN Tabelle_2

ON Spalte_1 VERGLEICHS-OPERATOR Spalte_2

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 an, was die Abfrage tut!

SELECT A.*, U.* FROM ARTIKEL AS A INNER JOIN UMSATZ AS U ON A.A_NR = U.A_NR WHERE A.A_NR < 13;	
SELECT A.A_NR, A.A_NAME, 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 WHERE A.A_NAME LIKE '%o%';	

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