

Data Query Language

Eine Datenbank enthält eine Vielzahl verschiedener Daten. Abfragen dienen dazu, bestimmte Daten aus der Datenbank auszugeben. Dabei kann die Ergebnismenge gemäß den Anforderungen eingegrenzt und genauer gesteuert werden.

Die SQL-Abfrage kann grundlegend in die folgenden sechs Komponenten unterteilt werden:

SELECT [* DISTINCT] <i>Spaltenbezeichner</i> [, ...]	Zeige die Werte aus folgenden Spalten ...
FROM <i>Tabellenbezeichner</i> [, ...]	... aus folgenden Tabellen
[WHERE <i>Bedingung</i>]	... schränke die Auswahl wie folgt ein ...
[GROUP BY <i>Spalte</i> [, ...]]	Gruppenbildung: Verdichte die Ausgabe von allen Zeilen mit gleichem Attributwert
[HAVING <i>Bedingung</i>]	Zusätzliche Auswahlbedingung auf Gruppenebene
[ORDER BY <i>Spalte</i> [, ...] [ASC DESC] ...]	sortiere Ausgabe anhand folgender Spalten
[LIMIT [start,] anzahl]	Gibt nur die ersten DS oder von x bis y aus

Bei der Selektion mit WHERE wird als Bedingung ein logischer Vergleich erwartet, der mit „true“ oder „false“ beantwortet werden kann:

Operator	Beispiel	Bedeutung, Wirkung
=	knr = 24	gleich
<	preisgruppe < 5	kleiner als
>		größer als
<>		ungleich
<=		kleiner gleich
>=		größer gleich
LIKE	ort LIKE 'B%'	Vergleicht das Feld Ort mit dem Textmuster
AND	plz = 96047 AND nr < 100	Datensatz wird nur berücksichtigt, wenn Bedingung 1 UND Bedingung 2 erfüllt sind
OR	nr > 200 OR nr < 100	Datensatz wird berücksichtigt, wenn Bedingung 1 ODER Bedingung 2 erfüllt ist.
NOT	NOT (plz = 96047)	Datensatz wird berücksichtigt wenn die Bedingung NICHT erfüllt ist.
BETWEEN	gebdat BETWEEN '1980/12/31' AND '1979/01/01'	Bereichsabfragen bei numerischen -, Datums- und Textfeldern
IN	wohntort IN ('ULM', MUC', 'BER')	Datensatz wird berücksichtigt wenn Wohnort entweder „ULM“ oder „MUC“ oder „BER“

	LF 8: Daten systemübergreifend bereitstellen – DQL Info	
	Klasse: ITT 11	Datum:

Pattern Matching (Jokerzeichen) – Bedeutung siehe Internet:

%, _, [B-F], [^B-F]

DISTINCT:

Das Schlüsselwort DISTINCT untersdrückt in der Abfrage doppelte Datensätze (Duplikate).

NULL-Werte:

Spalten in denen keine Werte eingegeben werden, haben den Wert NULL. Rechenoperationen mit einem Feld, das den Wert NULL enthält, erhalten als Ergebnis den WERT NULL. Mit der Eigenschaft NOT NULL können alle Felder gesucht werden die einen Eintrag enthalten.

Daten sortieren:

Mit der Anweisung **ORDER BY** können Ausgaben nach Feldinhalten sortiert werden. Dabei wird die Sortierreihenfolge mit dem Zusatz **ASC** (ascending = aufsteigend) oder **DESC** (descending = absteigend) festgelegt.

Werden zwei Feldinhalte als Sortierkriterien angegeben, dann wird nach dem ersten sortiert und bei gleichen Feldinhalten zusätzlich nach dem zweiten.

Alias in Abfrage:

Mit dem Schlüsselwort **AS** können Spaltennamen in der Abfrage ersetzt werden.

Beispiele:

```
SELECT *
FROM Mitarbeiter;
```

```
SELECT name AS Nachname, gebdat
FROM Mitarbeiter
WHERE gebdat BETWEEN '1978/01/01' AND '1979/12/31'
ORDER BY gebdat
LIMIT 1;
```

```
SELECT DISTINCT name
FROM Mitarbeiter;
```

```
SELECT name, vorname, telefon, email
FROM Mitarbeiter
WHERE email IS NOT NULL;
```