

Kapitel 4 - SQL-Grundlagen

- 4.1 SELECT, FROM und WHERE
- 4.2 ORDER BY und DISTINCT
- 4.3 Vergleichsoperatoren
- 4.4 LIKE, IN, BETWEEN und IS NULL

4.1 SELECT, FROM und WHERE

Definition

SELECT, **FROM** und **WHERE** gehören zu den wichtigsten SQL-Befehlen für Datenbankabfragen.

Mit ihnen werden Daten aus Tabellen ausgewählt und gefiltert.

Grundaufbau

```
SELECT spalte  
FROM tabelle  
WHERE bedingung;
```

SELECT

SELECT legt fest, welche Spalten ausgegeben werden sollen.

Beispiel:

```
SELECT name  
FROM kunde;
```

FROM

FROM legt fest, aus welcher Tabelle die Daten gelesen werden.

Beispiel:

```
SELECT name  
FROM kunde;
```

WHERE

WHERE filtert Datensätze nach einer Bedingung.

Beispiel:

```
SELECT name
FROM kunde
WHERE ort = 'Berlin';
```

Alle Spalten ausgeben

```
SELECT *
FROM kunde;
```

Das Sternchen * steht für alle Spalten.

Merksatz

SELECT = welche Spalten

FROM = aus welcher Tabelle

WHERE = welche Bedingung

Prüfungshinweis

In IHK-Aufgaben wird häufig gefragt, welche Datensätze mit einer SQL-Abfrage ausgegeben werden.

Typische Fehler:

- SELECT und FROM vertauschen.
- WHERE vergessen.
- Textwerte nicht in Anführungszeichen setzen.
- = mit LIKE verwechseln.
- SELECT * verwenden, obwohl nur bestimmte Spalten gefragt sind.

Quellen

- IT-Handbuch für Fachinformatiker

4.2 ORDER BY und DISTINCT

Definition

ORDER BY sortiert das Ergebnis einer SQL-Abfrage.

DISTINCT entfernt doppelte Werte aus dem Ergebnis.

ORDER BY

Mit **ORDER BY** werden Datensätze sortiert.

```
SELECT name, ort
FROM kunde
ORDER BY name;
```

ASC und DESC

Schlüsselwort	Bedeutung
ASC	aufsteigend sortieren
DESC	absteigend sortieren

Beispiel aufsteigend

```
SELECT name, ort
FROM kunde
ORDER BY name ASC;
```

Beispiel absteigend

```
SELECT name, ort
FROM kunde
ORDER BY name DESC;
```

DISTINCT

Mit **DISTINCT** werden doppelte Werte entfernt.

```
SELECT DISTINCT ort  
FROM kunde;
```

Merksatz

ORDER BY sortiert.

DISTINCT entfernt doppelte Werte.

Prüfungshinweis

In IHK-Aufgaben wird häufig gefragt, in welcher Reihenfolge Datensätze ausgegeben werden oder wie doppelte Werte vermieden werden.

Typische Fehler:

- ORDER BY vor WHERE schreiben.
- ASC und DESC verwechseln.
- DISTINCT mit GROUP BY verwechseln.
- DISTINCT auf einzelne Werte falsch verstehen.

Quellen

- IT-Handbuch für Fachinformatiker

4.3 Vergleichsoperatoren

Definition

Vergleichsoperatoren werden in SQL verwendet, um Werte miteinander zu vergleichen.

Sie werden häufig in der **WHERE-Klausel** eingesetzt.

Wichtige Vergleichsoperatoren

Operator	Bedeutung
=	gleich
<>	ungleich
<	kleiner als
>	größer als
<=	kleiner oder gleich
>=	größer oder gleich

Beispiele

Gleich

```
SELECT *  
FROM kunde  
WHERE ort = 'Berlin';
```

Ungleich

```
SELECT *  
FROM kunde  
WHERE ort <> 'Berlin';
```

Größer als

```
SELECT *  
FROM artikel  
WHERE preis > 100;
```

Kleiner oder gleich

```
SELECT *  
FROM artikel  
WHERE bestand <= 10;
```

Textwerte

Textwerte werden in SQL normalerweise in einfache Anführungszeichen gesetzt.

```
WHERE name = 'Müller'
```

Zahlenwerte

Zahlenwerte werden ohne Anführungszeichen geschrieben.

```
WHERE preis >= 50
```

Merksatz

Vergleichsoperatoren prüfen Bedingungen in der WHERE-Klausel.

Prüfungshinweis

In IHK-Aufgaben wird häufig geprüft, ob die richtige Bedingung für eine Abfrage gewählt wurde.

Typische Fehler:

- `=` und `<>` verwechseln.
- Textwerte ohne Anführungszeichen schreiben.
- Zahlenwerte unnötig in Anführungszeichen setzen.
- `<` und `>` vertauschen.
- WHERE-Klausel vergessen.

Quellen

- IT-Handbuch für Fachinformatiker

4.4 LIKE, IN, BETWEEN und IS NULL

Definition

LIKE, IN, BETWEEN und **IS NULL** werden in SQL verwendet, um Daten gezielt zu filtern.

Sie werden meistens in der **WHERE-Klausel** eingesetzt.

LIKE

LIKE sucht nach Mustern in Textwerten.

```
SELECT *  
FROM kunde  
WHERE name LIKE 'M%';
```

Bedeutung:

Alle Kunden, deren Name mit **M** beginnt.

Platzhalter bei LIKE

Platzhalter	Bedeutung
%	beliebig viele Zeichen
_	genau ein Zeichen

IN

IN prüft, ob ein Wert in einer Liste enthalten ist.

```
SELECT *  
FROM kunde  
WHERE ort IN ('Berlin', 'Potsdam');
```

BETWEEN

BETWEEN prüft, ob ein Wert in einem Bereich liegt.


```
SELECT *  
FROM artikel  
WHERE preis BETWEEN 50 AND 100;
```

IS NULL

IS NULL prüft, ob ein Feld keinen Wert enthält.

```
SELECT *  
FROM kunde  
WHERE telefon IS NULL;
```

Übersicht

Ausdruck	Bedeutung
LIKE	Mustervergleich
IN	Wert ist in einer Liste enthalten
BETWEEN	Wert liegt zwischen zwei Grenzen
IS NULL	Feld ist leer / ohne Wert

Merksatz

LIKE sucht Muster.

IN sucht in Listen.

BETWEEN sucht Bereiche.

IS NULL sucht fehlende Werte.

Prüfungshinweis

In IHK-Aufgaben wird häufig gefragt, welcher Operator für eine bestimmte Filterung geeignet ist.

Typische Fehler:

- `=` statt `LIKE` für Muster verwenden.
- `%` und `_` verwechseln.
- `NULL` mit `= NULL` prüfen.
- `BETWEEN` mit zwei Einzelbedingungen verwechseln.
- Textwerte ohne Anführungszeichen schreiben.

Quellen

- IT-Handbuch für Fachinformatiker