

# Wiederholung SQL-Grundlagen

## 1. Grundlagen & Anwendung

**Frage:** Was ist die Hauptaufgabe von SQL und in welchem Verhältnis steht es zu einem Datenbankmanagementsystem (DBMS)?

**Lösung:** SQL ist eine Spezialsprache zur Kommunikation mit relationalen Datenbanken. Während das DBMS (z. B. MySQL oder SQLite) die Software ist, welche die Daten verwaltet, ist SQL die „Schnittstelle“, mit der wir Befehle an das System senden (Daten abfragen, einfügen, ändern oder löschen).

## 2. Projektion

**Frage:** In einer Tabelle „Schüler“ gibt es die Spalten Vorname, Nachname, Klasse und Geburtsdatum. Wie sieht ein SQL-Befehl aus, der nur die Namen aller Schüler anzeigt? Wie nennt man diesen Vorgang?

**Lösung:** `SELECT Vorname, Nachname FROM Schüler;` Diesen Vorgang nennt man **Projektion**, da man eine Auswahl aus den vorhandenen Spalten (Attributen) trifft.

## 3. Selektion

**Frage:** Du möchtest aus der Tabelle „Schüler“ nur diejenigen finden, die in der Klasse „9b“ sind. Welches Schlüsselwort benötigst du und wie nennt man diesen Filtervorgang?

**Lösung:** Man nutzt das Schlüsselwort `WHERE`, also: `SELECT * FROM Schüler WHERE Klasse = '9b';`. Man nennt dies **Selektion**, da hierbei bestimmte Zeilen (Datensätze) nach einer Bedingung ausgewählt werden.

## 4. Projektion vs. Selektion (Kombination)

**Frage:** Erkläre den Unterschied: Was passiert bei einer Projektion und was bei einer Selektion?

**Lösung:**

**Projektion:** Wählt bestimmte **Spalten** aus (vertikaler Ausschnitt).

**Selektion:** Wählt bestimmte **Zeilen** aus (horizontaler Ausschnitt).

Meistens werden beide kombiniert, um genau die Infos zu bekommen, die man braucht.

## 5. Zeichensetzung bei Zeichenketten

**Frage:** Warum funktioniert der Befehl `SELECT * FROM Produkte WHERE Name = Apfel;` wahrscheinlich nicht? Was muss korrigiert werden?

**Lösung:** Texte (Zeichenketten/Strings) müssen in SQL immer in **einfache**

**Anführungszeichen** gesetzt werden. Richtig wäre: `... WHERE Name = 'Apfel';`. Ohne die Anführungszeichen denkt SQL, dass Apfel der Name einer Spalte ist.

## 6. Platzhalter (Wildcards)

**Frage:** Du suchst alle Kunden, deren Nachname mit „Schm...“ beginnt (z. B. Schmidt, Schmitt, Schmiede). Welches Zeichen nutzt du als Platzhalter und wie lautet die Bedingung?

**Lösung:** Man nutzt das Prozentzeichen % als Wildcard. Die Bedingung lautet: WHERE Nachname LIKE 'Schm%';. Das Schlüsselwort LIKE ist hierbei zwingend erforderlich.

## 7. Platzhalter für Einzelzeichen

**Frage:** Was ist der Unterschied zwischen den Wildcards % und \_?

**Lösung:**

% steht für eine **beliebige Anzahl** von Zeichen (auch null Zeichen).

\_ steht für exakt **ein einzelnes** Zeichen (z. B. findet L\_sa sowohl 'Lisa' als auch 'Lesa').

## 8. Aggregation

**Frage:** Was versteht man unter Aggregatfunktionen? Nenne drei Beispiele.

**Lösung:** Aggregationen fassen mehrere Datensätze zu einem einzigen Ergebniswert zusammen. Beispiele:

COUNT(): Zählt die Anzahl der Datensätze.

SUM(): Berechnet die Summe von Werten.

AVG(): Berechnet den Durchschnitt.

## 9. Join (Tabellenverknüpfung)

**Frage:** Warum benötigt man in einer relationalen Datenbank den JOIN-Befehl?

**Lösung:** Um Redundanzen (doppelte Daten) zu vermeiden, werden Informationen oft auf mehrere Tabellen verteilt (z. B. „Schüler“ und „Klasse“). Mit einem JOIN kann man diese Tabellen über gemeinsame Spalten (Primär- und Fremdschlüssel) wieder logisch zusammenführen, um eine Gesamtaufstellung zu erhalten.

## 10. Praxis-Check: Die Reihenfolge

**Frage:** Bring die folgenden SQL-Bausteine in die richtige logische Reihenfolge: FROM, SELECT, WHERE, ORDER BY.

**Lösung:**

1. SELECT (Was will ich sehen? → Datenfelder)
2. FROM (Woher kommen die Daten? → Datenbank /-basis)
3. WHERE (Welche Bedingungen gelten? → Auswahl der Datensätze)
4. ORDER BY (Wie soll sortiert werden? → ASC/DESC)
5. LIMIT (Anzahl der Datensätze)

---

Oft hilft es, sich SQL als einen „Satzbaukasten“ vorzustellen.

„Wähle [Spalten] aus [Tabelle], wo [Bedingung] erfüllt ist.“