Lernzettel

SQL-Grundlagen und fortgeschrittene Abfragen: SELECT, Joins, Aggregationen, Views, Transaktionen

Universität: Technische Universität Berlin
Kurs/Modul: Informationssysteme und Datenanalyse

Erstellungsdatum: September 19, 2025

Lernzettel: SQL-Grundlagen und fortgeschrittene Abfragen: SELECT, Joins, Aggregationen, Views, Transaktionen

(1) Grundlagen der SQL-Sprache.
Die Structured Query Language (SQL) dient der Abfrage und Modifikation relationaler Datenbanken. Wichtige Unterscheidungen:

- DDL (Data Definition Language): CREATE, ALTER, DROP
- DML (Data Manipulation Language): SELECT, INSERT, UPDATE, DELETE

(2) SELECT-Abfragen.
Mit SELECT extrahieren Sie Daten aus Tabellen.

```
SELECT vorname, nachname, geburtsjahr
FROM mitarbeiter
WHERE abteilung = 'Verkauf'
ORDER BY nachname ASC
LIMIT 100;
```

(3) Joins.
Joins verbinden Tabellen über gemeinsame Spalten.

```
-- Inner Join
SELECT a.name, b.bestellwert
FROM kunden AS a
JOIN bestellungen AS b ON a.kundennr = b.kundennr;

-- Left/Outer Join
SELECT a.name, b.bestellwert
FROM kunden AS a
LEFT JOIN bestellungen AS b ON a.kundennr = b.kundennr;
```

(4) Aggregationen.
Funktionen wie COUNT, SUM, AVG, MIN, MAX arbeiten mit Gruppen.

```
SELECT abteilung, COUNT(*) AS anzahl_mitarbeiter
FROM mitarbeiter
GROUP BY abteilung
HAVING COUNT(*) > 5
ORDER BY anzahl_mitarbeiter DESC;
```

(5) Views.
Views sind gespeicherte Abfragen, die als virtuelle Tabellen fungieren.

```
CREATE VIEW aktive_kunden AS
SELECT id, name, status
FROM kunden
WHERE status = 'aktiv';
```

(6) Transaktionen.
Transaktionen gewährleisten Atomizität, Konsistenz, Isolation, Dauerhaftigkeit (ACID).

```
BEGIN TRANSACTION;
UPDATE konten SET saldo = saldo - 100 WHERE kontonr = 1;
UPDATE konten SET saldo = saldo + 100 WHERE kontonr = 2;
COMMIT;
```

(7) Beispiel-Szenario: Abfragen mit Joins, Aggregationen und Views.
Beispiel 1: Top-Kunden nach Umsatz

```
SELECT c.name, SUM(o.total) AS gesamtumsatz
FROM kunden AS c
JOIN bestellungen AS o ON o.kunde_id = c.id
GROUP BY c.name
ORDER BY gesamtumsatz DESC
LIMIT 10;
```

Beispiel 2: View für aktive Kunden mit Umsatz-Filter

```
CREATE VIEW top_kunden AS
SELECT c.id, c.name, SUM(o.total) AS gesamtumsatz
FROM kunden AS c
JOIN bestellungen AS o ON o.kunde_id = c.id
GROUP BY c.id, c.name
HAVING SUM(o.total) > 1000
ORDER BY gesamtumsatz DESC;
```