TECH STARTER

SQL Grundlagen

Structured Query Language (SQL)

SQL

- Eine Datenbanksprache, mit der wir Abfragen für die Datenbank schreiben können
- Existiert in vielen “Dialekten”, z.B.:
  - Postgres
  - MariaDB
  - MySQL
  - SQLite
- Teilweise als synonym für “relational”
  - Daher auch “NoSQL-Datenbanken”, als nicht relationale Datenbanken

TECH STARTER

SQL

- Wir können mit SQL beispielsweise:
  - Tabellen erstellen
  - Tabellen löschen
  - Werte hinzufügen / ändern / löschen
  - Tabellen ausgeben lassen (komplett oder nur bestimmte Spalten)
  - Ausgabe Filtern
  - Tabellen kombiniert ausgeben lassen
  - …

Tabellen (Zusammenfassung)

- Haben klar definierte Spalten
- Jede Spalte hat einen festen Datentypen
- (mindestens) eine Spalte bildet den Primärschlüssel
- Jede Zeile ist ein Eintrag der Tabelle und muss einen eindeutigen Wert für den Primärschlüssel haben

TECH STARTER

Die Hausaufgabe anhand eines Beispiels

TECH STARTER

SQL anhand eines Beispiels

- Idee: Videospielsammlung

- Tabellen: Spiele, Publisher, Spielstände

Videospielsammlung:
Die Tabellen und ihre Spalten

- Spiele:
  - Spielid (int) (PK)
  - Titel (varchar)
  - Publisherid (int) (FK)
  - Genre (varchar)
- Publisher:
  - Publisherid (int) (PK)
  - Name (varchar)
- Spielstände:
  - Spielstandid (int) (PK)
  - Spielid (int) (FK)
  - Spielzeit [in Minuten] (int)
  - Vollständigkeit [in %] (decimal 3,2)

TECH STARTER

CREATE TABLE

```
1 CREATE TABLE publisher (
2   publisher_id int,
3   name varchar(50),
4   PRIMARY KEY (publisher_id)
5 );
```

Habe hier ein fehlendes Komma ergänzt, nicht wundern! Mein fehler :)

TECH STARTER

CONSTRAINTS

- Zu dt.: “Beschränkungen”
- Kann “Regeln” für eine Spalte festlegen
- Darunter unter Anderem:
  - PRIMARY KEY (ist Primärschlüssel)
  - NOT NULL (darf nicht leer sein)
  - UNIQUE (Kein Wert darf doppelt vorkommen)
  - FOREIGN KEY (referenziert einen fremden Primärschlüssel)

TECH STARTER

CREATE TABLE

```
1 CREATE TABLE spielstaende(
2     spielstand_id int PRIMARY KEY,
3     spiel_id int FOREIGN KEY REFERENCES spiele(spiel_id),
4     spielzeit time,
5     vollstaendigkeit_in_prozent decimal(5,2),
6   );
```

- Beachte Syntax

  “spalte int FOREIGN KEY REFERENCES tabelle(spalte)”

CREATE TABLE

```sql
CREATE TABLE spiele (
    spiel_id int,
    titel varchar(255) NOT NULL,
    publisher_id int,
    genre varchar(255),
    PRIMARY KEY (spiel_id),
    FOREIGN KEY (publisher_id) REFERENCES publisher(publisher_id)
);
```

- Hier zu sehen: Constraints können auch unten gesetzt werden, statt direkt hinter die Spaltendefinitionen

INSERT INTO

```
1 INSERT INTO tabellen_name (spalteA, spalteB, spalteC)
2 VALUES (WertA, "WertB", WertC)
```

- Entweder wir geben die Spalten, die wir befüllen wollen, an und Schreiben die Werte in gleicher Reihenfolge

```
1 INSERT INTO tabellen_name
2 VALUES (WertA, "WertB", WertC)
```

- Oder wenn wir alle Spalten befüllen, geben wir die Werte direkt in der Reihenfolge der Spalten der Tabelle an

INSERT INTO

```
1 INSERT INTO tabellen_name
2 VALUES
3     (WertA, "WertB", WertC),
4     (WertD, "WertE", WertF),
5     (WertG, "WertH", WertI9);
```

Habe hier zwei fehlende Kommata ergänzt, nicht wundern! Mein fehler :)

Mehrere Zeilen auf einen Schlag sind auch erlaubt!

TECH STARTER

SELECT

- SELECT können wir nutzen, um uns Daten aus der Datenbank geben zu lassen
- Hinter SELECT schreiben wir die Spalten die wir haben wollen
- Dazu gehört stets ein FROM hinter das wir die Tabelle schreiben, die wir ansprechen wollen

TECH STARTER

SELECT

```sql
SELECT spiel_id, titel, genre
FROM spiele;
```

```sql
SELECT spiel_id, titel, genre FROM spiele;
```

Ob mit Zeilenumbruch oder ohne, ist hier egal

SELECT

```
1 SELECT * FROM spiele;
```

Wollen wir die gesamte Tabelle, schreiben wir “SELECT * “

WHERE

- WHERE können wir nutzen, um einen SQL-Befehl nur für einen Teil der Einträge in einer Tabelle auszuführen
- Unser Befehl wird nur auf die Zeilen angewandt, die die Bedingung hinterm WHERE erfüllen

```
1 SELECT *
2 FROM spielstaende
3 WHERE spiel_id = 1
```

TECH STARTER

Bedingungen Verknüpfen

- Wir können Bedingungen bei WHERE verknüpfen, indem wir sie mit AND oder OR verbinden

```
1 SELECT * FROM spiele
2 WHERE spiel_id > 1 AND spiel_id < 4
```

- Bedingungen können außerdem mit NOT zur gegensätzlichen Bedingung gedreht werden

```
1 SELECT * FROM spiele
2 WHERE spiel_id > 1 AND NOT titel = "Elden Ring"
```

Aufgabe!

Finde den passenden SQL-Befehl!

1. Wie erhalte ich ausschließlich alle Spieletitel?
2. Wie erhalte ich alle Spielstände mit allen Spalten?
3. Wie erhalte ich nur die Spielstände für Super Mario 64 (spiele_id ist 1), mit mehr als 60 Minuten Spielzeit?

TECH STARTER

UPDATE

```
1 UPDATE spielstaende
2 SET spielzeit = 60
3 WHERE spielstand_id = 1
```

- Mithilfe von UPDATE können wir Zeilen in einer Tabelle bearbeiten
- Über die ID einzelne Werte zu ändern geht gut
- Mit Vorsicht zu genießen, gerade bei mehreren Einträgen

TECH STARTER

DELETE

```
1 DELETE FROM spielstaende
2 WHERE spielstand_id = 3
```

- DELETE löscht alle Zeilen, die unsere Bedingung unter WHERE erfüllen
- Ohne Bedingung, werden alle Zeilen gelöscht

TECH STARTER

Wie mache ich mir die Beziehungen der Tabelle zu Nutze?

TECH STARTER

Problembeispiel

- Wie gebe ich mir alle Spielstände inklusive Spieletitel aus? Der Spieletitel ist nicht Teil der Spielstaende Tabelle!

JOIN

- JOIN lässt uns zwei Tabellen aneinanderhängen, indem wir eine Spalte angeben, die beide Tabellen enthalten (Primär-/Fremdschlüssel)

(INNER) JOIN

```
1 SELECT spiele.spiel_id, spiele.titel, spielstaende.spielstand_id
2 FROM spiele
3 INNER JOIN spielstaende ON spiele.spiel_id=spielstaende.spiel_id
```

- Wenn wir mehrere Tabellen in der gleichen Query haben, sollten wir “tabelle.spalte” schreiben, für alle Spalten
- Für die Ausgabe wählen wir SELECT
- Für den JOIN wählen wir zunächt eine Tabelle mit FROM und verbinden sie danach mit INNER JOIN mit einer weiteren Tabelle

Aufgabe!

Finde den SQL Befehl!

1. Gib alle Spieletitel mit jeweiligem Publisher aus
2. Gib alle Spielstaende aus mit spielstand_id, Spielzeit und Spieltitel
3. (Bonus:) So wie “2.”, nur dass der Publishername der Spiele dabei stehen soll!