

SQL & Relational Databases – Wiederholung

In den letzten Wochen hast du dir viel Wissen über Relationale Datenbanken angeeignet und zum ersten Mal die Abfragesprache SQL genutzt. Erinnere dich daran, dass du in mehreren Kapiteln grundlegende Begriffe, neue Konzepte und ihre praktische Umsetzung kennengelernt hast.

Gibt es Inhalte, die du noch nicht ganz verstanden hast? Oder Übungen, mit denen du Schwierigkeiten hattest? Um sicherzugehen, dass du das Gelernte wirklich verinnerlicht hast, kann es sinnvoll sein, einige Inhalte zu wiederholen. Die folgenden Fragen können dir dabei helfen, deine Kenntnisse zu überprüfen und sicherzustellen, dass du die Kernkonzepte des Kurses gut verstanden hast.

Grundlagen von Datenbanken und relationalen Modellen

1. Was sind die Hauptaufgaben von Datenbanken und wie können sie im Alltag angewendet werden?
2. Erkläre den Unterschied zwischen Primärschlüsseln und Fremdschlüsseln in einem relationalen Datenmodell.
3. Warum sind Integritätsbedingungen wichtig und wie tragen sie zur Datenkonsistenz bei?

SQL-Grundlagen und Datenabfragen

1. Welche SQL-Anweisungen verwendest du, um eine Liste aller Tabellen in einer Datenbank anzuzeigen?
2. Wie kannst du mit SQL Daten aus einer Tabelle auswählen und bestimmte Spalten anzeigen lassen?
3. Welche Aggregationsfunktionen in SQL verwendest du, um den Durchschnitt oder die Summe von Werten zu berechnen?

Abfragen optimieren und Daten filtern

1. Wie filterst du Daten in einer SQL-Abfrage mithilfe der WHERE-Klausel?
2. Erkläre den Unterschied zwischen **GROUP BY** und **HAVING** in Bezug auf das Aggregieren von Daten.
3. Wie führst du eine **JOIN**-Operation zwischen zwei oder mehr Tabellen durch und welche **JOIN**-Arten kennst du?

Fortgeschrittene Abfragen und Subqueries

1. Welche booleschen Operatoren werden in SQL verwendet und wie kombinierst du sie, um komplexe Bedingungen zu erstellen?
2. Was ist eine Subquery und wie kannst du sie verwenden, um Daten in einer Hauptabfrage zu filtern?
3. Wann würdest du einen `LEFT JOIN` im Vergleich zu einem `INNER JOIN` verwenden?

Datenbankdesign und Normalisierung

1. Warum sind Normalisierungsprinzipien wichtig für ein effizientes Datenbankdesign?
2. Erkläre den Unterschied zwischen einem Star Schema und einem Snowflake Schema.
3. Was sind die Vor- und Nachteile von relationalen Datenbanken im Vergleich zu nicht-relationalen Ansätzen?

Tabellenverwaltung und Views

1. Welche SQL-Anweisungen verwendest du, um eine neue Tabelle in einer Datenbank zu erstellen?
2. Wie änderst du die Daten einer vorhandenen Tabelle mit SQL?
3. Was sind Views in SQL und wie können sie zur Vereinfachung von Datenabfragen genutzt werden?

Wenn du Inhalte oder Übungen wiederholen möchtest, nutze gerne auch die folgende Übersicht, um die Lektionen zu finden, zu denen du zurückkehren möchtest.

Kapitel 1: Einführung in relationale Datenbanken und SQL

Relationale Datenbanken:

- Aufgaben und Anwendung von Datenbanken (Video 2, Video 3)
- Relationales Datenmodell und Relationenschema (Text 2)
- Primär- und Fremdschlüssel (Text 2)
- Integritätsbedingungen (Text 2)

SQL:

- Übersichten mit `SHOW tables;` und `DESC table;` (Notebook 1)
- Tabellen erkunden mit `SELECT` und `FROM` (Notebook 2)
- Ergebnis kürzen mit `LIMIT` (Notebook 2)
- Eindeutige Werte mit `DISTINCT` (Notebook 2)
- Rechenoperationen durch Expressions (Notebook 3)
- Aliasse (Notebook 3)
- Aggregationsfunktionen wie `MIN()`, `MAX()`, `AVG()`, `COUNT()` (Notebook 3)
- Filtern mit `WHERE` (Notebook 4)
- Boolesche Operatoren wie `AND`, `OR` (Notebook 5)

Kapitel 2: Vertiefung SQL

SQL:

- SQL-Dialekte (Handout 2)
- Allgemeiner SQL-Workflow (Handout 3)
- Gruppieren mit **GROUP BY** (Notebook 2)
- Gruppierte Daten filtern mit **HAVING** (Notebook 2)
- Sortieren mit **ORDER BY** (Notebook 2)
- Tabellen verbinden mit **JOIN** (Notebook 3)
- Fortgeschrittene Joins wie **LEFT JOIN**, **RIGHT JOIN** (Notebook 5)
- Subqueries (Notebook 4)

Kapitel 3: Vertiefung Datenbankdesign

Relationale Datenbanken:

- Nicht-relationale Datenbanken (Video 2)
- Vor- und Nachteile relationaler Datenbanken (Text 1)
- Entscheidungsorientierte und Operative Datenbanken (Text 2)
- Datenbankdesign (Text 2)
- Normalisierung (Text 3)
- Star und Snowflake Schemata (Text 4)

SQL:

- Tabellen erstellen mit **CREATE TABLE** (Notebook 2)
- Tabellen verändern mit **ALTER TABLE** (Notebook 2)
- Tabellen löschen mit **DROP TABLE** (Notebook 2)
- Tabellen mit Werten füllen mit **INSERT INTO** (Notebook 3)
- Tabellenwerte ändern mit **UPDATE** (Notebook 3)
- Zeilen löschen mit **DELETE FROM** (Notebook 3)
- Views erstellen mit **CREATE OR REPLACE VIEW** (Notebook 4)
- Views löschen mit **DROP VIEW** (Notebook 4)