Grundlagen Datenbanken

Grundlegende Konzepte und Befehle

Datenbank Definition

Eine Datenbank ist eine systematische Sammlung von Daten, die in einer strukturierten Form gespeichert, organisiert und verwaltet werden. Sie dient dazu, Informationen effizient abzurufen, zu aktualisieren, zu speichern und zu verarbeiten.

Datenbanken spielen eine zentrale Rolle in der Informationsverarbeitung und sind in verschiedenen Anwendungen und Bereichen weit verbreitet, darunter Unternehmensanwendungen, Webanwendungen, wissenschaftliche Forschung, Bibliotheken und mehr.

Strukturierte Daten

In einer relationalen Datenbank werden Daten in strukturierter Form gespeichert, wobei Informationen in Tabellen, Zeilen und Spalten organisiert sind. Diese Struktur ermöglicht es, Daten logisch zu verknüpfen und abzufragen.

id	Produkt	Kategorie
1	Snickers XXL	Food
2	TV	Electronics

Datenintegrität

Datenbanken bieten Mechanismen zur Sicherstellung der Datenintegrität. Dies bedeutet, dass die Daten konsistent und korrekt sind und die Integritätsregeln eingehalten werden.

Integritätsregel könnte zum Beispiel sein, dass einer Bestellung immer ein Kunde zugeordnet ist.

Abfragen

Durch die Verwendung von Abfragesprachen wie SQL (Structured Query Language) können Benutzer Daten aus der Datenbank abrufen und komplexe Abfragen durchführen.

Jedes Datenbanksystem besitzt in der Regel SQL-Konstrukte, die von anderen Systemen **nicht** verstanden werden. SQL ist somit zwischen den Systemen häufig nicht direkt austauschbar.

In diesem Tutorial beziehen wir uns auf das SQL im MySQL-Flavour.

Datenbank erstellen

Um eine neue Datenbank zu erstellen, muss mit dem folgendem Befehl eine erstellt werden. In einem Datenbank-Management-System kann man mehrere Datenbanken parallel nutzen.

Die folgenden Beispiele sind für das DBMS MySQL und können mit dem MySQL-Commandline-Client ausgeführt werden.

```
CREATE DATABASE game;
```

Schema

Ein Schema ist ein logischer Container für Datenbankobjekte wie Tabellen, Sichten und Prozeduren. In einigen Datenbanksystemen ist ein Schema ein Teil einer Datenbank und ermöglicht die Organisation von Objekten innerhalb einer Datenbank.

In MySQL ist ein Schema gleichbedeutend mit einer Datenbank. Wenn Sie in MySQL eine neue Datenbank erstellen, erstellen Sie auch ein neues Schema. Man kann also eine Datenbank in MySQL auch so erstellen:

```
CREATE SCHEMA game;
```

Datenbank auswählen

Um mit der Datenbank zu arbeiten, muss man erstmal eine auswählen. Dazu gibt es den Befehl use

USE game;

Das Datenbanksystem ist jetzt bereit, mit der Datenbank zu arbeiten. Die Befehle müssen nicht groß geschrieben werden, es ist aber von Vorteil, um sie von anderen Bezeichnern abzugrenzen.

Datentypen

In MySQL (und anderen RDBMS) gibt es eine Vielzahl von Datentypen, die verwendet werden können, um verschiedene Arten von Daten in einer Datenbank zu speichern. Hier sind einige der gängigsten Datentypen und ihre Verwendung:

Datentyp	Beschreibung	Beispiel
INT	ganzzahlig, 32 bit, mit Vorzeichen oder ohne	INT
FLOAT / DOUBLE	Fließkommazahlen, 32 / 64 bit	FLOAT
DECIMAL	Festkommazahlen mit Vor- und Nachkommastellen	DECIMAL(10, 2)
VARCHAR	Zeichen, bis 65.535 Zeichen	VARCHAR(255)
TEXT	für lange Texte	TEXT
DATE	Datumswerte	DATE
BOOLEAN	Wahrheitswert	BOOL

Anmerkung zu int

INT-Datentypen werden immer mit einer zusätzlichen Zahl in Klammer angegeben, zb. INT(4).

Bei dieser Zahl handelt es sich nur um die sogenannte Display-Width, die von manchen Programmen im Zusammenhang mit ZEROFILL genutzt wird, um die Zahl entsprechend zu mit einem Padding zu formatieren. Sie gibt nicht an, wie groß die Zahl im Feld sein darf!

id INT(4) ZEROFILL

Zum Beispiel könnte die Zahl 32 dann so dargestellt weden:

0032

Tabelle erstellen

Wir erstellen eine erste Tabelle.

```
CREATE TABLE player (
    id INT AUTO_INCREMENT PRIMARY KEY,
    vorname VARCHAR(50),
    nachname VARCHAR(50),
    is_active BOOL,
    points INT
);
```

Die Tabelle besteht aus 4 Spalten id, vorname, nachname und geburtsdatum. Die Spalte id ist der Primärschlüssel.

AUTO_INCREMENT erzeugt eine eindeutige, fortlaufende Nummer für eine Spalte.

Best Practices bei der Namensgebung von Tabellen

Folgende Schreibweisen sollten vermieden werden:

camelCase, es ist schwer schnell zu erfassen.

keine Umlaute, idealerweise englisch

Beschreibende Präfixe oder ungarische Notation wie tbl_

Namen müssen mit einem Buchstaben beginnen.

Best Practices bei Namensgebung von Spalten

- Immer den Singularnamen nutzen, zb. city und nicht cities
- nicht den gleichen Namen wie die Tabelle nutzen
- lowercase
- Beschreibende Präfixe oder ungarische Notation wie column_sport.
- Namen müssen mit einem Buchstaben beginnen.

Show Tables

Um sich alle Tabellen innerhalb einer MySQL-Datenbank anzeigen lassen, wählen wir zuerst die Datenbank aus.

```
use game;
```

Dann lassen wir uns alle Tabellen anzeigen:

```
show tables;
```

Datensatz einfügen

Um Daten in die Tabelle einzufügen gibt es den Befehlt INSERT INTO.

```
INSERT INTO player (vorname, nachname, is_active, points)
VALUES
('Max', 'Mustermann', True, 200);
```

Da die Spalte id der Primary-Key ist und die Eigenschaft AUTO_INCREMENT hat, wird dieser Wert automatisch hochgezählt. Man braucht ihn nicht extra einzufügen.

Datensätze abfragen

Um Datensätze abzufragen, gibt es den Befehl SELECT

```
SELECT
        *
FROM
        player;
```

Wir selektieren alle Spalten aus der Tabelle benutzer. Falls uns nur die Spalten vorname und nachname interessieren, würde man folgendes schreiben:

SELECT vorname, nachname FROM player;

Datensätze abfragen

Ein Select gibt uns alle Datensätzen einer Tabelle zurück. Wenn wir nur eine Sub-Menge erhalten wollen, müssen wir die Menge mit einer WHERE-Bedingung einschränken. Das Sternchen steht für alle Spalten.

```
SELECT
        *
FROM
        player
WHERE
        points < 300;
```

Select Auswahl an Spalten

Ein Select gibt uns alle Datensätzen einer Tabelle zurück. Wenn wir nicht alle Spalten in der Ausgabe haben wollen, sondern nur eine Auswahl, kann man das beim Selektieren angeben.

```
SELECT
        id, firstname
FROM
        player
```

Datensätze aktualisieren

Um Datensätze zu aktualisieren, gibt es den Befehl UPDATE. Auch hier arbeitet man idealerweise mit Einschränkungen, ansonsten werden alle Datensätze upgedated. Der SET-Befehl gibt an, welche Spalten ersetzt werden sollen.

```
UPDATE
        player
SET
        firstname = 'John',
        points=34
WHERE
        id=3;
```