SQL

Zusammenfassung für den 26.09.19

Begriff

SQL ist eine Programmiersprache zum Abfragen und manipulieren von Daten. Es ist die Abkürzung für "**s**tructured **q**uery **l**anguage". Die Befehle werden in folgende 4 Bereiche eingeteilt:

- (DQL) zur Abfrage und Aufbereitung der gesuchten Informationen
- (DML) zur Datenmanipulation (Ändern, Einfügen, Löschen) und lesendem Zugriff
- (DDL) zur Definition des Datenbankschemas
- (DCL) für die Rechteverwaltung und Transaktionskontrolle

Ein wichtiger Bestandteil sind die **Schlüssel** (Primary Key, Foreign Key), welche eine eindeutige Zuordnung möglich machen, um Daten miteinander zu verbinden und gezielt anzusteuern (z.B. für UPDATE).

Redundanz beschreibt die Existenz von überflüssigen / doppelten Informationen und ist nicht erwünscht.

Die **referentielle Integrität** (Beziehungsintegrität) beschreibt die Bedingung, dass Informationen miteinander verbunden sind. So muss es für einen Fremdschlüssel den jeweiligen Primärschlüssel geben.

Tabellenstruktur

```
CREATE TABLE table_name (
 column_name1 data_type(size) optional_property,
 column_name2 data_type(size) optional_property,
 ...
);
```

Erstellt eine Tabellenstruktur.

Datentypen

- Int kurz für Intiger. Das sind ganzzahlige Werte
- Decimal(x,n) speichert Dezimalzahlen; x: alle Stellen, n: alle nach dem Komma
- Varchar(x) für X Zeichen

Eigenschaften

- NOT NULL das Feld darf nicht leer bleiben
- PRIMARY KEY kombiniert UNIQUE und NOT NULL und darf nur einmal in einer Tabelle vorkommen
- DEFAULT wenn kein Wert festgelegt wird, wird dieser Wert eingefügt anstatt NULL
- AUTO_INCREMENT erzeugt automatisch eine einzigartige Zahl

```
ALTER TABLE table_name
 ADD column_nameX data_type(size) optional_property,
 MODIFY column_name data_type(size) optional_property,
 DROP column_name,
 ADD property;
```

Ist zum Verändern einer bereits existierenden Tabellenstruktur. Mit „DROP TABLE table_name;“ kann eine Tabelle auch samt Inhalt gelöscht werden.

Daten

Daten warden hiermit eingefügt:

```
INSERT INTO name (cloumn1, column2,...) VALUES
 (value1_1, value2_1, ...),
 (value1_2, value2_2, ...);
```

Und bereits bestehende Zeilen können hiermit verändert werden:

```
UPDATE name
 SET column1 = new_value1, column2 = new_value2, ...
 WHERE condition;
```

Zum Schluss, mit „DELETE FROM name WHERE condition;" gelöscht.

Abfragen

```
SELECT [DISTINCT] column1, column2, ...
 FROM tablename
 [WHERE condition]
 [ORDER BY column ASC | DESC]
```

- DISTINCT lässt nur unterschiedliche Werte zu. So wird "DE, EN, US" angezeigt anstatt "DE, DE, EN, DE, US, EN, US, DE".
- Mit ORDER BY werden die Ergebnisse nach der Spalte mit entweder aufsteigend (ASC) oder absteigend (DESC)
- Mehrere Bedinungen müssen mit AND und OR verknüpft werden.
- Nutzt man BETWEEN value AND value, werden die alle Werte dazwischen ausgegeben, falls vorhanden. Mit NOT kann dies umgekehrt werden.
- IN (value, value, value) gibt Zeilen mit diesen Werten, wieder. Es kann mit NOT umgekehrt werden oder auch als Langform mit OR geschrieben werden.

Funktionen

Sie können nicht in dem "WHERE"-Statement genutzt werden. Dazu wird eine extra Abfrage geschrieben. Z.B.: ... WHERE bezeichnung='Obama-Tower' AND Kosten=(SELECT max(kosten) FROM ...)...

- MIN() gibt den kleinsten Wert an
- AVG() zeigt den Mittelwert aus allen Daten
- MAX() bringt das Maximum zum Vorschein
- SUM() addiert alle Werte
- COUNT() zählt die Werte