



The screenshot shows the browser's developer tools with the DOM tree and style inspector open. The DOM tree on the left lists various HTML elements like `<g>`, `<text>`, and `<g>` under a root node, many of which have the class prefix "highcharts-". The style inspector on the right shows the CSS properties for a selected element, including `position: relative`, `width: 300px`, `height: 300px`, and `text-align: left`. The color palette at the bottom indicates a color of #3399FF.

# Modul 106

## Day 1

### Auftrag 2

db\_verändern.sql

schule\_[nur\_üK].sql

Grundlagen der DML-Befehle(Data Manipulation Language) und DDL-Befehle(Data Definition Language).

### DDL(Data Definition Language)

#### 1. ALTER

```
ALTER TABLE person  
ADD COLUMN firma VARCHAR(300);
```

Dieser Befehl fügt der Tabelle `person` eine weitere Spalte mit dem Namen `firma` und dem Datentyp `VARCHAR(300)` zu.

```
ALTER TABLE person  
CHANGE COLUMN firma firma VARCHAR(60);
```

Dieser Befehl bearbeitet die Spalte `firma` in der Tabelle `person`. Die abgeänderte Spalte heisst nun `firma` und der Datentyp ist neu `VARCHAR(60)`.

## 2. DROP

```
DROP TABLE person;  
DROP DATABASE schule;
```

`DROP TABLE` löscht eine Tabelle samt deren Inhalt.

`DROP DATABASE` löscht eine Tabelle samt deren Tabellen.

# DML(Data Manipulation Language)

## 1. INSERT INTO

```
INSERT INTO person  
(klasse, name, vorname, firma)  
VALUES  
( 'INA1C', 'Müller', 'Yannick', 'Swisscom Schweiz AG' )  
/*, ('INA1C/A', 'Burgstaller', 'Robin', 'Merkle')*/;
```

In der Klammer nach dem Tabellennamen gibt man an welche Werte man setzen möchte und in der Klammer nach `VALUES` gibt man die Werte in der oben festgelegten Reihenfolge ein.

## 2. UPDATE

```
UPDATE person  
SET name = 'Simpson',  
    vorname = 'Bart'  
WHERE vorname = 'Yannick';
```

Mit `UPDATE` können vorhandene Datensätze bearbeitet werden, im oben stehenden Beispiel wird bei jedem Datensatz bei welchem der aktuelle `vorname`

Yannick ist der `name` mit Simpson und der `vorname` mit Bart ersetzt.

### 3. DELETE

```
DELETE FROM person  
WHERE vorname = 'Bart';  
  
DELETE FROM person;
```

Im obenstehenden Beispiel werden zuerst alle Datensätze bei welchen der `vorname` Bart ist aus der Tabelle gelöscht bevor dann mit `DELETE FROM person;` alle Datensätze gelöscht werden.

## DQL(Data Query Language)

### 1. SELECT

Der `SELECT` Befehl ist der einzige Befehl der DQL und mit ihm kann man sich die Daten einer Tabelle anzeigen lassen.

```
SELECT * FROM person  
WHERE name = 'Yannick';
```

## Auftrag 3

db\_spital.sql

Ein ERD erstellen und danach mit Database à Forward Engineering SQL-Script zur Erstellung der Datenbank und Tabellen generieren.

## Auftrag 4

spiele.sql

Daten aus einer .csv Datei importieren.

## Befehle (neu)

```
LOAD DATA INFILE  
'C:/Tabelle-hersteller.csv'  
INTO TABLE hersteller  
CHARACTER SET utf8mb4  
FIELDS TERMINATED BY ';'   
LINES TERMINATED BY '\r\n'  
IGNORE 1 ROWS  
(id, standort, name);
```

Mit diesem Befehl importiert man die Daten aus der CSV-Datei.

`CHARACTER SET` Setzt die Codierung in der die CSV-Datei gelesen werden muss.

`FIELDS TERMINATED BY ';'`  setzt das Trennungszeichen der Attribute auf ein Semikolon.

`LINES TERMINATED BY '\r\n'` setzt das Datensatztrennzeichen auf einen Carriage Return Line Feed.

`IGNORE 1 ROWS` sorgt dafür, dass die Headerzeile ignoriert wird.

In den Klammern wird angegeben in welcher Reihenfolge die Attribute in der CSV-Datei stehen.

## Day 2

### Aufgabe 5

[benutzer\\_rechte\\_verwalten.sql](#)

[schule.sql](#)

### Benutzer erstellen

```
CREATE USER 'jan'@'localhost' IDENTIFIED BY 'V3ry-53cur3-Pa55';
CREATE USER 'Tim'@'%' IDENTIFIED BY 'V3ry-53cur3-Pa55w0rd';
```

Dieser Befehl erstellt einen Benutzer mit dem Benutzernamen `jan`, welcher nur vom Server selber (`localhost`) auf das DBMS zugreifen kann. Das Passwort wird in diesem Fall auf `V3ry-53cur3-Pa55w0rd` gesetzt.

Er erstellt noch einen zweiten Benutzer mit dem Benutzernamen `Tim`, dieser kann nun aber von überall auf das DBMS zugreifen. Auch `Tim` bekommt das Passwort `V3ry-53cur3-Pa55w0rd` zugewiesen.

## Benutzerrechte erteilen

```
GRANT SELECT, UPDATE ON schule.* TO 'jan'@'localhost';
```

Dieser Befehl fügt dem Benutzer `jan` die Rechte `SELECT` und `UPDATE` auf allen Tabellen der Datenbank `schule` hinzu.

## Benutzerrechte entziehen

```
REVOKE UPDATE ON schule.* FROM 'jan'@'localhost';
```

Dieser Befehl entzieht dem Benutzer `jan` das `UPDATE` Recht auf allen Tabellen der Datenbank `schule`.

## Benutzer löschen

```
USE mysql;
DELETE FROM user
WHERE User = 'Tim';

DROP USER 'jan'@'localhost';
```

Es gibt wie oben dargestellt zwei Möglichkeiten einen Benutzer zu löschen.

## Aufgabe 6

tierheim.sql

tierheim\_abfragen.sql

## JOINS

```
SELECT t.name AS TierName, h.name AS HalterName  
FROM tier AS t  
RIGHT JOIN halter AS h ON t.halter_id = h.id;
```

Dieser Befehl gibt den Tiernamen und den Namen des Halters aus zu welchem das Tier gehört ebenfalls wird mit `AS` gearbeitet um eine schöne Darstellung der Tabelle und einen leserlichen Code zu gewährleisten.

## Aggregatsfunktionen

```
SELECT name AS TierName  
FROM tier  
WHERE name LIKE 'B%';
```

Dieser Befehl selektiert alle Tiere mit einem Namen der mit `B` beginnt.

```
SELECT name AS TierName  
FROM tier  
WHERE name IN ('Benno', 'Bilde', 'Tilda', 'Mia');
```

Dieser Befehl selektiert nur die Tiere mit den Namen zwischen den runden Klammern.

```
SELECT COUNT(*) AS AnzahlTiere FROM tier;  
SELECT MIN(jahrgang) AS ÄltestesTier FROM tier;
```

Der erste Befehl gibt **die Anzahl der Datensätze** in der Tabelle `tier` zurück.

Der zweite Befehl gibt **den kleinsten Wert** des Attributs `jahrgang` zurück.

