



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 sections such as 'highcharts-series-group', 'highcharts-exporting-group', and 'highcharts-axis-labels'. 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;'. There is also a preview of the chart area.

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.