Lab 2

Uniwersytet Warszawski
Wydział Nauk Ekonomicznych

SQL Basic Syntax - CREATE/DROP database

- **Creating an empty database dbTest:**

```
create database dbTest;
```

- **Drop(remove) a database dbTest:**

```
drop database dbTest;
```

- **Show all tables:**

```
show tables; /*in sqlite use .tables*/
```

- **Show all columns in database dbTest:**

```
show columns from dbTest;  /* .schema dbTest*/
```

SQL Basic Syntax – CREATE TABLE

Create table table_name:

```
CREATE TABLE table_name (
    column1 datatype,
    column2 datatype,
    column3 datatype,
   ....
);
```

Create table customers with single Primary Key:

```
CREATE TABLE customers (
    idCustomer int NOT NULL AUTO_INCREMENT,
    lastName varchar(255) NOT NULL ,
    firstName varchar(255) NOT NULL ,
    address varchar(255) DEFAULT NULL,
    timeStamp time NULL,
    PRIMARY KEY (idCustomer)
);
```

AUTO_INCREMENT :
- column it is used on must be indexed
-column that is it used on cannot have a DEFAULT value
-can be only one AUTO_INCREMENT column per table

SQL Basic Syntax – CREATE TABLE (examples)

Create table, which contains one enumarated filed:

```
CREATE TABLE new_fruits_enum (
      fruit_name ENUM('Apple', 'Orange', 'Pear') DEFAULT 'Pear,
      );
```

Create table, which contains compound Primary Key {artist_id, album_id}:

```
CREATE TABLE album (artist_id INT(5) NOT NULL,
            album_id INT(4) NOT NULL AUTO_INCREMENT,
            album_name CHAR(128) DEFAULT NULL,
            PRIMARY KEY (artist_id, album_id)
      );
```

SQL Basic Syntax – ALTER TABLE

- **Change the name of column {played} to {last_played}**

```
ALTER TABLE Songs CHANGE played last_played TIMESTAMP;
```

- **Modify the data type and clauses of a column {name} in table artist :**

```
ALTER TABLE artist MODIFY name CHAR(64) DEFAULT "Unknown";
```

- **Add an extra column {bandSize} to an existing table artist:**

```
ALTER TABLE artist ADD bandSize int;
```

- **Add column {bandSize} to table artist in a specific position:**

```
ALTER TABLE artist ADD bandSize int AFTER artist_id;
```

- **Remove column {bandSize}**

```
ALTER TABLE artist DROP bandSize
```

- **Rename table artist to playlist**

```
ALTER TABLE artist RENAME TO playlist;
```

ALTER TABLE
- add new columns to a table
- remove existing columns
- change columns names, types, lengths

SQL Basic Syntax – INSERT data

We insert into table artist values of artist_id and surname with below query:

```
INSERT INTO artist VALUES (7, "Adamson");
```

Analogically, we can insert many records in single SQL query:

```
INSERT INTO artist VALUES (7, "Adamson"),
                          (8, ”Brick"),
                          (9,”Samuelson”);
```

Alternatively,we can use syntex as below:

```
INSERT INTO played (artist_id, album_id, track_id)
      VALUES (7,1,2),(7,1,3),(7,1,4);
```

Or

```
INSERT INTO played SET artist_id = 7, album_id = 1, track_id= 1;
```

SQL Basic Syntax – DELETE data

We delete all rows in table songs:

```
DELETE FROM songs;
```

We delete row with satisfies given condition artist_id=3, using following query :

```
DELETE FROM artist WHERE artist_id = 3;
```

SQL Basic Syntax – SELECT data

- **We can select all rows and all columns from a table:**

```
SELECT * FROM table_name;
```

- **Or restrict the list of columns that we are interested in:**

```
SELECT column1, column2, ...
FROM table_name;
```

- **Sometimes we are not interested in all the rows but only in unique values in column1 but also containing information from column2.**

```
SELECT DISTINCT column1, column2, ...
FROM table_name;
```

Distinct :
- allows to selet unique values from given table

SQL Basic Syntax – SELECT data

- **Usually we want to retrieve just some rows (for example most recent or fulfilling certain criteria)**

```
SELECT * FROM customers WHERE lastName='Smith';
SELECT * FROM orders WHERE orderValue >=10000;
```

- **We may not be sure what is the exact phrase we are looking for:**

```
SELECT * FROM customers WHERE lastName LIKE 'Sm%';
SELECT * FROM customers WHERE lastName LIKE '_i%';
```

- **Or we know that there are many rows but we want only 5 most recent ones**

```
SELECT * FROM Orders ORDER BY orderDate DESC LIMIT 5;
```

- **Of course we can order records basing on many columns:**

```
SELECT time, track_name FROM track ORDER BY time, track_name;
```

- **We can also select the maximum/minimum value from column artist_id**

```
SELECT MAX(artist_id) FROM artist;
SELECT MIN(artist_id) FROM artist;
```

SQL Basic Syntax – UPDATE data

Change the values in {artist_name} column to uppercase:

```
UPDATE artist SET artist_name = UPPER(artist_name);
```

Change the values in {artist_name} column to NULL in table songs:

```
UPDATE songs SET artist_name = NULL;
```

We update values in rows,which satisfy given condition using WHERE clause:

```
UPDATE album SET album_name =”Dragon” WHERE artist_id = 1 AND
album_id = 2;
```

We can also update 10 lowest scores to 0 in table songs:

```
UPDATE songs SET score = 0 ORDER BY score ASC LIMIT 10;
```

OREDR BY :

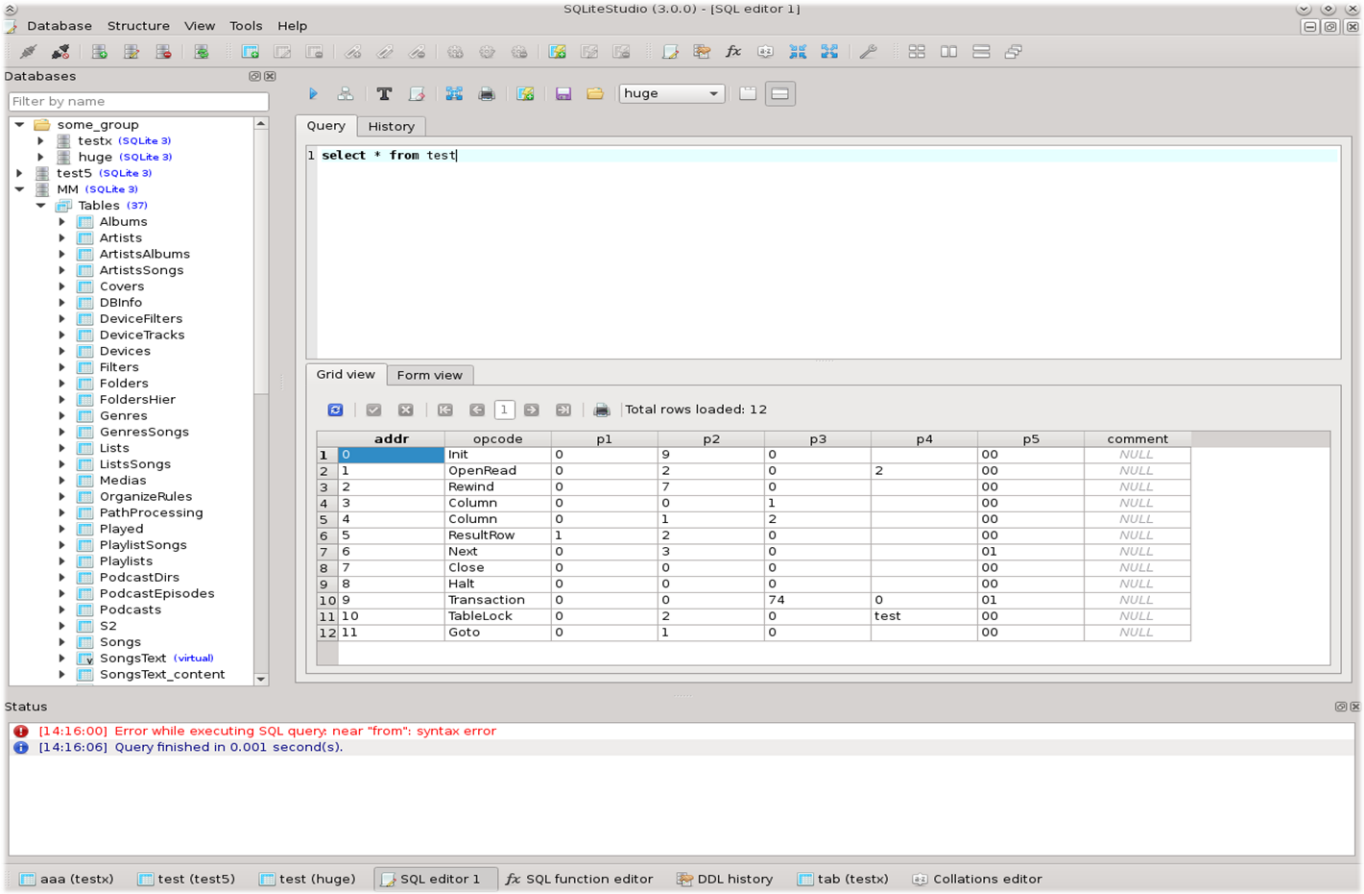
- ORDER BY *key_part1* DESC, *key_part2* DESC;
- -ORDER BY *key_part1* ASC, *key_part2* ASC;

SQLITE

- relational database management system contained in a C programming library
- designed in the spring of 2000 by Richard Hipp (part of the software for guided missile destroyers)
- open source
- light (takes only 0,3MB)
- one file (.sqlite or .db)
- implements most of the SQL-92 standard for SQL
- suitable for embedded devices and small-medium websites

SQLiteStudio

- freeware, portable (no installation needed)
- https://sqlitestudio.pl



Student's Task

1. Download SQLiteStudio from https://sqlitestudio.pl
2. Create database uni_{your_surname}
3. Create tables and choose suitable data types for their columns:

   student(student_id,name,surname,dateOfBirth,yearEnrolled),

   course(course_id,name,creditPoints,yearCommenced)

   staff(employee_id,name,surname,jobTitle)

   program(program_id,name,creditPoints, yearCommenced)

3. Write instert queries and fill tables with random data (min. 10 rows per table)
4. Present all students whose name starts with „W"
5. Present all students who are currenty at the 4th year

Student's Task

7. Present all courses from table course starting with these which have the highest numer of credit points
8. Change the name of the student with lowest student_id into Adam
9. Change all values from column {name} in course table into uppercase
10. Delete the oldest student in table student
11. Remove column yearCommenced from course table
12. Rename table staff into employee
13. Send data base file via email to instructor

References

1. **Williams H., S. Tahaghoghi (2009). Learning MySQL., O'Reilly Media**
2. **Churcher C. (2007), Beginning Database Design.,Apress**

Links

- **https://www.python.org/**
- **https://sqlitestudio.pl**
- **https://www.sqlite.org/index.html**