Lab3 Create Populate Delete

Lab3 Agenda

- Create tables
- Insert / Modify data
- Εργαστηριακές Ασκήσεις
- Εξαμηνιαία Εργασία

CREATE DATABASE statement

- *CREATE DATABASE* statement is used to create a new SQL database

```
mysql> create database test1;
Query OK, 1 row affected (0.00 sec)
mysql> use test1;
Database changed
mysql> SHOW CREATE DATABASE test1;
+----------+-----------------------------------------------------------------+
| Database | Create Database                                                 |
+----------+-----------------------------------------------------------------+
| test1    | CREATE DATABASE `test1` /*!40100 DEFAULT CHARACTER SET utf8mb4 */ |
+----------+-----------------------------------------------------------------+
1 row in set (0.01 sec)
```

Create Table Statement

- *CREATE TABLE* statement allows you to create a new table in a database
- basic syntax

```
CREATE TABLE [IF NOT EXISTS] table_name(
   column_1_definition,
   column_2_definition,
   ...,
   table_constraints
) [ENGINE=storage_engine];
```

- storage engine: InnoDB(default) and MyISAM.

Create Table Statement

- syntax for a column's definition:

```
column_name data_type(length) [NOT NULL] [DEFAULT value] [AUTO_INCREMENT] column_constraint;
```

- *column_name*: name of the column. Each column has a specific *data type* and optional *size*
  - e.g. VARCHAR(255)
- *NOT NULL* constraint ensures that the column will not contain NULL.
  - a column may have additional constraint such as CHECK, and UNIQUE.
- *DEFAULT* specifies a default value for the column.
- *AUTO_INCREMENT*: the value of the column is incremented by one automatically whenever a new row is inserted into the table. (Max one AUTO_INCREMENT column per table)

Create Table

- table constraints
  - UNIQUE
    - values in a column or group of columns to be unique
  - CHECK
    - a Boolean expression which must evaluate to TRUE or UNKNOWN for each row of the table
  - PRIMARY KEY
    - unique values
    - a primary key column cannot have NULL values
    - a table can have one an only one primary key.
  - FOREIGN KEY
    - a column(s) in a table that links to a column(s) in another table
    - maintain referential integrity

Create Table Example

```
CREATE TABLE movies(
    title VARCHAR(50) NOT NULL,
    genre VARCHAR(30) NOT NULL,
    director VARCHAR(60) NOT NULL,
    release_year INT NOT NULL,
    PRIMARY KEY(title)
);
```

Create Table Example (from a file)

```
CREATE DATABASE movies1;
USE movies1;
CREATE TABLE movies(
    title VARCHAR(50) NOT NULL,
    genre VARCHAR(30) NOT NULL,
    director VARCHAR(60) NOT NULL,
    release_year INT NOT NULL,
    PRIMARY KEY(title)
);
INSERT INTO movies VALUE ("Joker", "psychological thriller", "Todd Phillips", 2019);
```

- save as file: myddl.sql
- connect with

```
mysql -u root -p <myddl.sql
```

Create Table Example (from a file)

```
MariaDB [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| movies1            |
+--------------------+
MariaDB [(none)]> use movies;
Database changed
MariaDB [(movies1)]>show tables;
+-------------------+
| Tables_in_movies1 |
+-------------------+
| movies            |
+-------------------+
1 row in set (0.00 sec)
```

Create Table Example (from a file)

```
MariaDB [(movies1)]>desc movies;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| title        | varchar(50) | NO   | PRI | NULL    |       |
| genre        | varchar(30) | NO   |     | NULL    |       |
| director     | varchar(60) | NO   |     | NULL    |       |
| release_year | int(11)     | NO   |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
4 rows in set (0.01 sec)
MariaDB [(movies1)]>select * from movies;
+-------+-----------------------+---------------+--------------+
| title | genre                 | director      | release_year |
+-------+-----------------------+---------------+--------------+
| Joker | psychological thriller | Todd Phillips |         2019 |
+-------+-----------------------+---------------+--------------+
1 row in set (0.00 sec)
```

Create Table Example

```
CREATE TABLE parts (
    part_no VARCHAR(18),
    description VARCHAR(40),
    cost DECIMAL(10,2) NOT NULL CHECK (cost >= 0),
    price DECIMAL(10,2) NOT NULL CHECK (price >= 0),
    PRIMARY KEY (part_no)
);
```

CREATE VIEW

- View: a query that is stored in the data dictionary

```
MariaDB [(movies1)]>CREATE VIEW minimum_release_year AS SELECT title FROM movies
    -> WHERE release_year > 1990;
Query OK, 0 rows affected (0.01 sec)
```

```
MariaDB [(movies1)]>select * from minimum_release_year;
+-------+
| title |
+-------+
| Joker |
+-------+
1 row in set (0.04 sec)
```

DROP Table

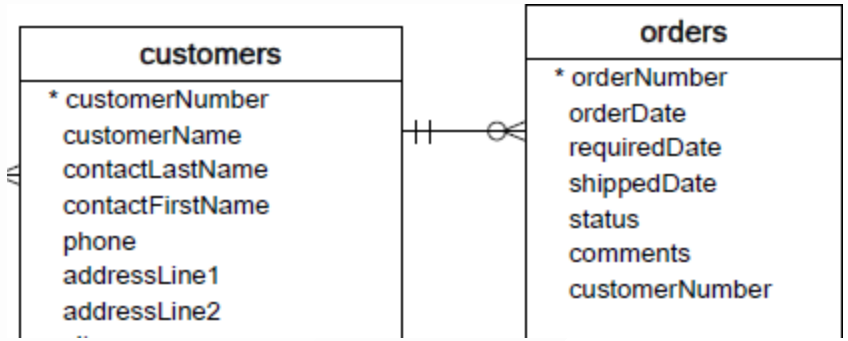
- **TRUNCATE TABLE**
  - used to delete the data inside a table, but not the table itself
    - *truncate table movies1;*
- **DROP TABLE**
  - used to drop an existing table in a database
    - *drop table movies1;*

DROP Table 😟

```
MariaDB [(movies1)]>drop table movies1;
Query OK, 0 rows affected (0.02 sec)
>show tables;
+----------------------+
| Tables_in_movies1    |
+----------------------+
| minimum_release_year |
+----------------------+
1 row in set (0.00 sec)

MariaDB [(movies1)]>select * from minimum_release_year;
ERROR 1356 (HY000): View 'movies1.minimum_release_year' references
invalid table(s) or column(s) or function(s) or
definer/invoker of view lack rights to use them
```

Create Tables Example



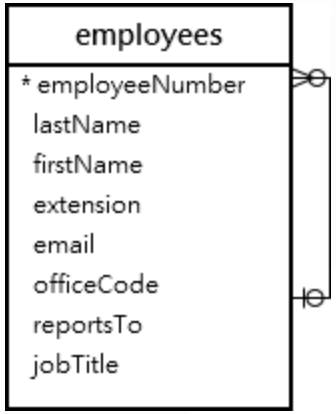
- each *customer* can have zero or many *orders* and each *order* belongs to one *customer*
- the relationship between *customers* and *orders* is *one-to-many*
  - established by the foreign key in the orders table specified by the customerNumber column
- The customers table is called the **parent** table or referenced table, and the orders table is known as the **child** table or referencing table.

Create Tables Example

```
CREATE TABLE customers (
  customerNumber int(11) NOT NULL,
  customerName varchar(50) NOT NULL,
  PRIMARY KEY (customerNumber)
) ENGINE=InnoDB;

CREATE TABLE orders (
  orderNumber int(11) NOT NULL,
  orderDate date NOT NULL,
  customerNumber int(11) NOT NULL,
  PRIMARY KEY (orderNumber),
  CONSTRAINT orders_ibfk_1 FOREIGN KEY (customerNumber) REFERENCES customers (customerNumber)
) ENGINE=InnoDB;
```

Self-referencing foreign key



- the **child** and **parent** tables may refer to the **same table**
- **reportTo** column is a *foreign key* that refers to the **employeeNumber** column which is the primary key of the employees table
- *Each employee reports to zero or one employee and an employee can have zero or many subordinates*

Populating Data

- **insert** statement: INSERT INTO .... VALUES ...
  - The name of the table into which to add the data
  - The names of the columns in the table to be populated
  - The values with which to populate the columns

🔔 provide data for every column in the table defined as **not null**

Populating Data

```
INSERT INTO actor (first_name, last_name) VALUES ('Zach',  'Galifianakis');
```

```
MariaDB [sakila]> select * from actor;
+----------+-------------+--------------+---------------------+
| actor_id | first_name  | last_name    | last_update         |
+----------+-------------+--------------+---------------------+
|        1 | PENELOPE    | GUINESS      | 2006-02-15 04:34:33 |
                        ...
|      200 | THORA       | TEMPLE       | 2006-02-15 04:34:33 |
|      201 | Zach        | Galifianakis | 2022-03-17 16:08:57 |
+----------+-------------+--------------+---------------------+
201 rows in set (0.001 sec)
```

? actor_id, last_update

```
MariaDB [sakila]> desc actor;
+-------------+------------------+------+-----+---------------------+-------------------------------+
| Field       | Type             | Null | Key | Default             | Extra                         |
+-------------+------------------+------+-----+---------------------+-------------------------------+
| actor_id    | int(10) unsigned | NO   | PRI | NULL                | auto_increment                |
| first_name  | varchar(45)      | NO   |     | NULL                |                               |
| last_name   | varchar(45)      | NO   | MUL | NULL                |                               |
| last_update | timestamp        | NO   |     | current_timestamp() | on update current_timestamp() |
+-------------+------------------+------+-----+---------------------+-------------------------------+
4 rows in set (0.018 sec)
```

how values are generated for numeric primary keys

1. pick a number out of thin air ❗
2. look at the largest value currently in the table and add one
3. let the database server provide the value for you
   - Oracle: a separate schema object is used (called a sequence)
   - MySQL: auto-increment
   - PostgreSQL: smallserial, serial and bigserial

insert again

```
INSERT INTO actor (actor_id, first_name, last_name, last_update)
VALUES (null, 'Maria',  'Menounos', null);
```

```
MariaDB [sakila]> select * from actor;
+----------+-------------+--------------+---------------------+
| actor_id | first_name  | last_name    | last_update         |
+----------+-------------+--------------+---------------------+
                         ...
|      201 | Zach        | Galifianakis | 2022-03-17 16:08:57 |
|      202 | Maria       | Menounos     | 2022-03-17 16:19:50 |
+----------+-------------+--------------+---------------------+
```

and again

```
INSERT INTO actor (actor_id, first_name, last_name, last_update) VALUES (300, 'Maria',  'Menounos', null);
```

```
MariaDB [sakila]> select * from actor;
+----------+-------------+-------------+---------------------+
| actor_id | first_name  | last_name   | last_update         |
+----------+-------------+-------------+---------------------+
|      201 | Zach        | Galifianakis | 2022-03-17 16:08:57 |
|      202 | Maria       | Menounos    | 2022-03-17 16:19:50 |
|      300 | Maria       | Menounos    | 2022-03-17 16:22:14 |
+----------+-------------+-------------+---------------------+
```

Updating Data

- **update** statement: UPDATE ... SET .... WHERE ...
  - The name of the table into which to update the data
  - The names of the columns in the table to be updated & new values for the columns
  - where: Filter data

🔔 **double check the where clause**

Updating Data

```
UPDATE actor
SET first_name = 'ZACH', last_name = upper('Galifianakis')
WHERE actor_id = 201;
```

```
Query OK, 1 row affected (0.016 sec)
Rows matched: 1  Changed: 1  Warnings: 0
MariaDB [sakila]> select * from actor where actor_id=201;
+----------+------------+--------------+---------------------+
| actor_id | first_name | last_name    | last_update         |
+----------+------------+--------------+---------------------+
|      201 | ZACH       | GALIFIANAKIS | 2022-03-17 16:31:24 |
+----------+------------+--------------+---------------------+
1 row in set (0.000 sec)
```

Updating Data 😢

- what would happen if my where clause looked as follows

```
UPDATE actor
SET first_name = 'ZACH', last_name = upper('Galifianakis')
WHERE actor_id < 201;
```

- what would happen if i ommited the where

```
UPDATE actor
SET first_name = 'ZACH', last_name = upper('Galifianakis') ;
```

Deleting Data 😥

- **delete** statement: DELETE FROM .... WHERE ...
  - The name of the table from which to DELETE the data
  - where: Filter data

```
DELETE FROM actor WHERE actor_id = 300;
```

```
Query OK, 1 row affected (0.017 sec)
```

⛔ check, recheck and check again

When Good Statements Go Bad

1. Nonunique Primary Key

```
INSERT INTO actor (actor_id, first_name, last_name)  VALUES (1, 'Irene',  'Papas');
```

```
MariaDB [sakila]> INSERT INTO actor (actor_id, first_name, last_name)  VALUES (1, 'Irene',  'Papas');
ERROR 1062 (23000): Duplicate entry '1' for key 'PRIMARY'
```

2. Nonexistent Foreign Key

```
desc film_actor;
INSERT INTO film_actor (actor_id, film_id)  VALUES (201, 1050);
```

```
MariaDB [sakila]> desc film_actor;
+-------------+------------------+------+-----+---------------------+-------------------------------+
| Field       | Type             | Null | Key | Default             | Extra                         |
+-------------+------------------+------+-----+---------------------+-------------------------------+
| actor_id    | int(10) unsigned | NO   | PRI | NULL                |                               |
| film_id     | int(10) unsigned | NO   | PRI | NULL                |                               |
| last_update | timestamp        | NO   |     | current_timestamp() | on update current_timestamp() |
+-------------+------------------+------+-----+---------------------+-------------------------------+
MariaDB [sakila]> INSERT INTO film_actor (actor_id, film_id)  VALUES (201, 1050);
ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint fails (`sakila`.`film_actor`, CONSTRAINT
`fk_film_actor_film` FOREIGN KEY (`film_id`)
REFERENCES `film` (`film_id`) ON UPDATE CASCADE)
```

- add film before film_actor

3. Column Value Violations

```
UPDATE film SET length = -1 WHERE film_id = 1;
```

```
MariaDB [sakila]> UPDATE film SET length = -1 WHERE film_id = 1;
Query OK, 1 row affected, 1 warning (0.015 sec)
Rows matched: 1  Changed: 1  Warnings: 1

MariaDB [sakila]> show warnings;
+---------+------+-----------------------------------------------+
| Level   | Code | Message                                       |
+---------+------+-----------------------------------------------+
| Warning | 1264 | Out of range value for column 'length' at row 1 |
+---------+------+-----------------------------------------------+
1 row in set (0.000 sec)
```

```
MariaDB [sakila]> UPDATE film SET rating = 'all' WHERE film_id = 1;
Query OK, 1 row affected, 1 warning (0.029 sec)
Rows matched: 1  Changed: 1  Warnings: 1

MariaDB [sakila]> show warnings;
+---------+------+-----------------------------------------+
| Level   | Code | Message                                 |
+---------+------+-----------------------------------------+
| Warning | 1265 | Data truncated for column 'rating' at row 1 |
+---------+------+-----------------------------------------+
```

- rating : enum('G','PG','PG-13','R','NC-17')

4. Invalid Date Conversions

```
MariaDB [sakila]> update customer set create_date =  'Aug 10 2017' WHERE customer.customer_id = 1;
Query OK, 1 row affected, 1 warning (0.016 sec)
Rows matched: 1  Changed: 1  Warnings: 1

MariaDB [sakila]> show warnings;
+---------+------+-----------------------------------------------------+
| Level   | Code | Message                                             |
+---------+------+-----------------------------------------------------+
| Warning | 1265 | Data truncated for column 'create_date' at row 1 |
+---------+------+-----------------------------------------------------+
```

```
update customer set create_date =  str_to_date("Aug 10 2017", "%b %d %Y")
WHERE customer.customer_id = 1;
```

- check your DB date formats
- check your DB date-formatting strings
  - Mysql %b The short month name, such as Jan, Feb, ...
  - Mysql %d The numeric day of the month (00..31)
  - Mysql %Y The four-digit year

Εργαστηριακές Ασκήσεις 💻

1. Actor 'EMILY TEMPLE' was accidentally entered in the actor table as 'EMILY DEE' Write a query to fix the problem.
2. You cannot locate the schema of the actor table. Which query would you use to re-create it?
3. Your manager thinks it is a good idea to store actors middle name. Write a query to achieve that.
4. Populate the newly created column 'middle name' with data for each actor.
5. Your manager is having second thoughts about the actors 'middle name'. Write a query to fix the problem.
6. Your manager thinks a table holding counts for actors, customers and rentals is needed. Write a query to achieve that.

Εργαστηριακές Ασκήσεις 💻

7. Your manager is having second thougths on your sql skills. He asked you to write a query that will create a table 'users', insert data and select data as following sample.

ΑΡ_ΤΑΥΤ	ΕΠΩΝΥΜΟ	ΟΝΟΜΑ	ΗΜ_ΓΕΝΝΗΣΗΣ
Π702538	ΚΑΛΛΙΓΕΡΟΣ	ΔΗΜΗΤΡΗΣ	10/11/1985
Χ234678	ΣΕΜΠΟΣ	ΒΑΣΙΛΗΣ	1/1/1976
Χ297200	ΜΙΧΑΣ	ΑΓΓΕΛΟΣ	25/3/1981

“ hint: use column aliases ”

Εξαμηνιαία Εργασία 💻

- ~~Database Schema Design~~
  1. Start thinking about the entities you need
     - Identify entities, attributes and relationships from the problem description
     - identify cardinality ratios of the relationships found
  2. Design an E/R diagram for your database
     - Look for any issues that are apparent in the E/R diagram

Εξαμηνιαία Εργασία 💻

- Materialize Schema: DDL statements
  1. Create your tables
     - create a table for each entity
     - a table (representing an entity) should have:
       - a column for each attribute, with appropriate data type

Wrap Up

1. [x] `Create tables`
2. [x] `Insert / Modify data`
3. [x] `Εργαστηριακές Ασκήσεις`
4. [x] `Εξαμηνιαία Εργασία`

Wrap Up

😱 **Απορίες** https://discord.gg/g3fFxWVPfD

Εργαστηριακές Ασκήσεις / Απαντήσεις

1. Actor 'EMILY TEMPLE' was accidentally entered in the actor table as 'EMILY DEE'
   Write a query to fix the problem

```
UPDATE actor SET last_name = 'TEMPLE'
WHERE first_name = 'EMILY' AND last_name = 'DEE';
```

Εργαστηριακές Ασκήσεις / Απαντήσεις

2. You cannot locate the schema of the actor table. Which query would you use to re-create it?

```
SHOW CREATE TABLE sakila.actor;
```

Εργαστηριακές Ασκήσεις / Απαντήσεις

3. Your manager thinks it is a good idea to store actors middle name. Write a query to achieve that.

```
ALTER TABLE actor
ADD COLUMN middle_name VARCHAR(45);
```

Εργαστηριακές Ασκήσεις / Απαντήσεις

4. Populate the newly created column 'middle name' with data for each actor.

```
update actor set  middle_name = CONCAT(first_name, ", ",  last_name);
```

“ hint: use first and last name ”

Εργαστηριακές Ασκήσεις / Απαντήσεις

5. Your manager is having second thoughts about the actors 'middle name'. Write a query to fix the problem

```
ALTER TABLE actor
DROP COLUMN middle_name;
```

Εργαστηριακές Ασκήσεις / Απαντήσεις

6. Your manager thinks a table holding counts for actors, customers and rentals is needed. Write a query to achieve that.

```
create table stats as
SELECT
  (SELECT count(*) FROM actor) as actors,
  (SELECT count(*) FROM customer) as cust,
  (SELECT count(*) FROM rental) as rental ;
```

Εργαστηριακές Ασκήσεις / Απαντήσεις

7. Your manager is having second thougths on your sql skills. He asked you to write a query that will create a table 'users', insert data and select data as following sample.

ΑΡ_ΤΑΥΤ	ΕΠΩΝΥΜΟ	ΟΝΟΜΑ	ΗΜ_ΓΕΝΝΗΣΗΣ
Π702538	ΚΑΛΛΙΓΕΡΟΣ	ΔΗΜΗΤΡΗΣ	10/11/1985
Χ234678	ΣΕΜΠΟΣ	ΒΑΣΙΛΗΣ	1/1/1976
Χ297200	ΜΙΧΑΣ	ΑΓΓΕΛΟΣ	25/3/1981

“ hint: use column aliases ”

Εργαστηριακές Ασκήσεις / Απαντήσεις

```
CREATE TABLE  users (
  id INT AUTO_INCREMENT PRIMARY KEY,
  adt varchar(45) NOT NULL,
  first_name varchar(45) NOT NULL,
  last_name varchar(45) NOT NULL,
  birth_date date not null
) ;

INSERT INTO users (adt, first_name, last_name, birth_date)  values
('Π702538', 'ΚΑΛΛΙΓΕΡΟΣ', 'ΔΗΜΗΤΡΗΣ', str_to_date("10/11/1985", "%d/%m/%Y") ),
('Χ234678', 'ΣΕΜΠΟΣ', 'ΒΑΣΙΛΗΣ', str_to_date("1/1/1976", "%d/%m/%Y") ),
('Χ297200', 'ΜΙΧΑΣ', 'ΑΓΓΕΛΟΣ', str_to_date("25/3/1981", "%d/%m/%Y") );

select
  adt as  ΑΡ_ΤΑΥΤ,
  first_name as ΕΠΩΝΥΜΟ,
  last_name as ΟΝΟΜΑ,
  birth_date as ΗΜ_ΓΕΝΝΗΣΗΣ
from users;
```