Databases with *MYSQL*

5. *SQL*

Dante – Digital Area for Networking Teachers and Educators

Funded by the Erasmus+ Programme of the European Union

Learning Outcomes

After this lesson, the learner will be able to use all the following SQL commands:

- CREATE DATABASE
- CREATE TABLE
- ALTER TABLE
- DROP TABLE
- INSERT
- UPDATE
- DELETE

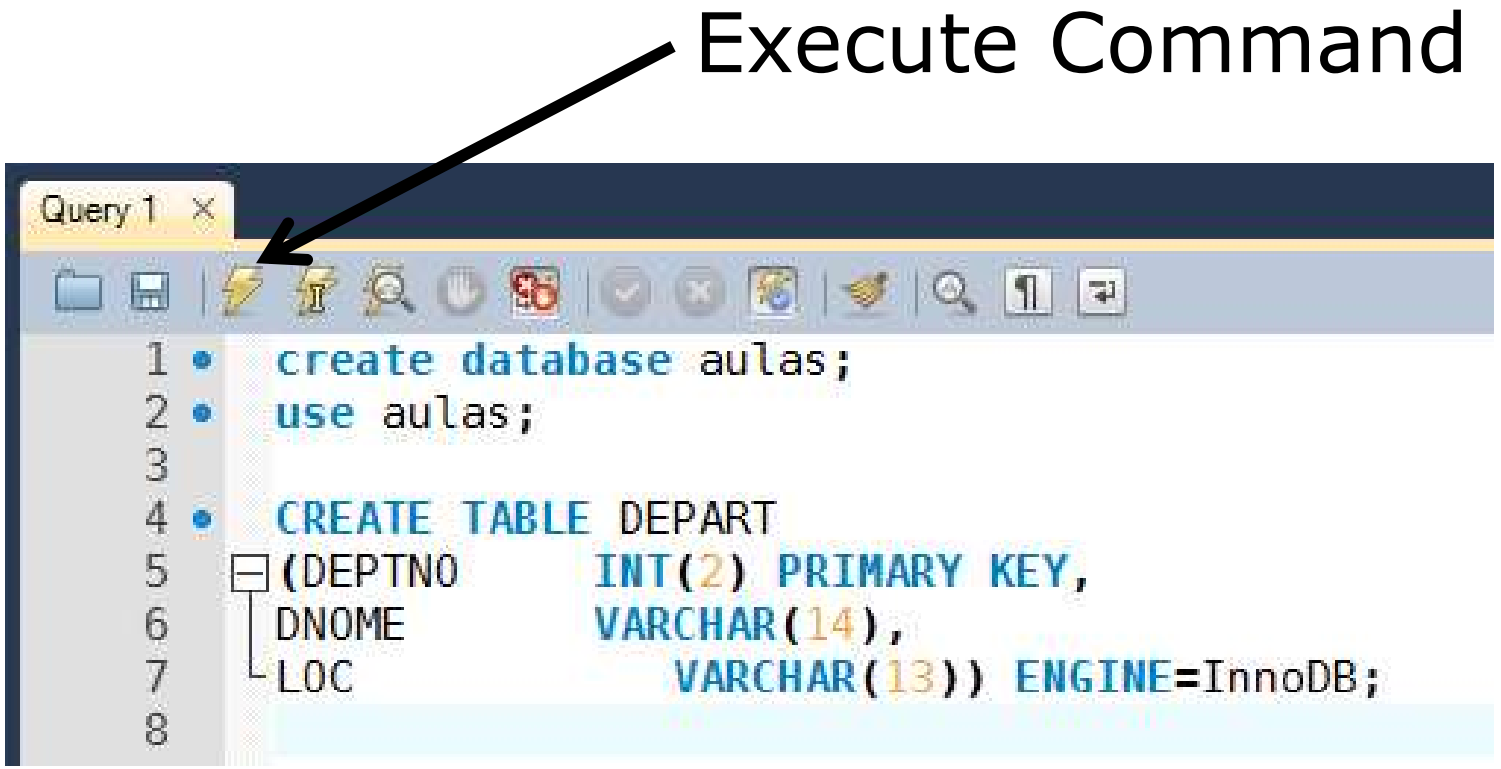
Funded by the Erasmus+ Programme of the European Union

Structured Query Language (SQL)

- Standard database language.
- For all accesses to the database.
- Types of commands:
  - Data Definition Language (DDL) : Create, Alter, Drop
  - Data Manipulation Language (DML): Insert, Delete, Update
  - Data Retrieval: Select
  - Transaction Control: Commit, Rollback, Savepoint
  - Data Control Language (DCL): Grant, Revoke

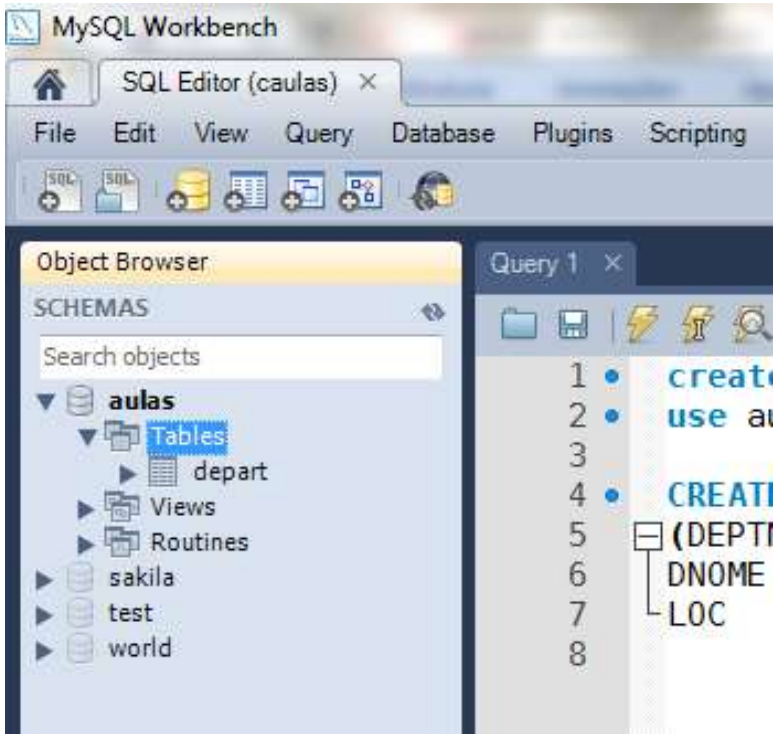
Funded by the Erasmus+ Programme of the European Union

Execute Script in *MYSQL Workbench*



Funded by the Erasmus+ Programme of the European Union

Execute *script*



Funded by the Erasmus+ Programme of the European Union

Example Database

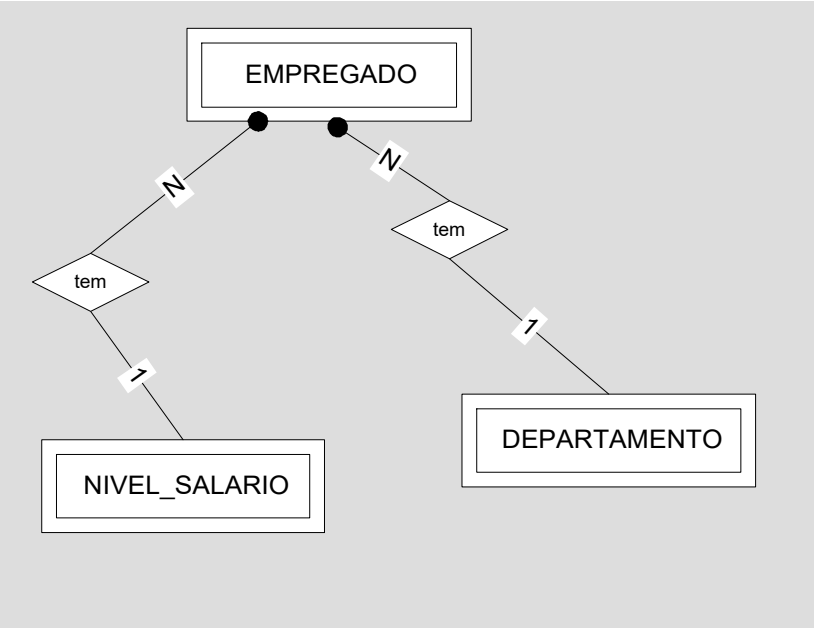


Table names:

EMP – employees;
DEPT – departments;
SALGRADE – salary grades.

EMP (Empno, Ename, Job, Mgr, Hiredate, Sal, Comm, Deptno)
DEPT (Deptno, Dname, Loc)
SALGRADE (Grade, Losal, Hisal)

Funded by the Erasmus+ Programme of the European Union

Create and Select a Database

Command to create the classes database:

```
CREATE DATABASE CLASSES;
```

Command to Select the classes database:

```
USE CLASSES;
```

The database only needs to be created once, however it is always necessary to select the database you are going to use when entering *MySql*.

Funded by the Erasmus+ Programme of the European Union

Data Types

The types of data that a field can have can be grouped into three large groups:

- Numeric types
- Date Types
- Chain Types

Funded by the Erasmus+ Programme of the European Union

Data Types

`CHAR(w)`
Fixed length alphanumeric string; maximum *w*=256 caracteres

`VARCHAR(w)`
Variable length alphanumeric string ; maximum *w*=256 caracteres

`NUMERIC(w)`
numeric values; maximum w=38 digits

`INT(w)`
integer values; maximum w=38 digits

`NUMERIC(w,d)`
numeric values; *w* digits and *d* decimal digits

`DATE`
dates (includes hour)

`DATETIME`
dates (includes hour)

Funded by the Erasmus+ Programme of the European Union

Creating a table

To create a table, CREATE TABLE command is used to specify

- The name of the table
- The name of each column
- The type of data to be stored in each column
- The width of each column

Command to create DEPT table:

```
CREATE TABLE DEPT
(DEPTNO      INT(2) PRIMARY KEY,
DNAME        VARCHAR(14),
LOC          VARCHAR(13));
```

Funded by the Erasmus+ Programme of the European Union

Creating a table

To store the simple information of departments in a table.

We will identify columns

```
DEPTNO      INT(2) primary key,
DNAME       VARCHAR(14),
LOC         VARCHAR(13)
```

Funded by the Erasmus+ Programme of the European Union

Creating tables

```
CREATE TABLE DEPT
(DEPTNO      INT(2),
DNAME        VARCHAR(14),
LOC          VARCHAR(13));
```

Funded by the Erasmus+ Programme of the European Union

Creating tables / Primary key

```
CREATE TABLE DEPT
(DEPTNO      INT(2) PRIMARY KEY,
DNAME        VARCHAR(14),
LOC          VARCHAR(13));
```

Funded by the Erasmus+ Programme of the European Union

Creating tables / foreign key

```
CREATE TABLE EMP
(EMPNO    INT(4)   PRIMARY KEY,
ENAME     VARCHAR(10),
JOB       VARCHAR(9),
MGR       NUMERIC(4),
HIREDATE  DATE,
SAL       NUMERIC(7,2),
COMM      NUMERIC(7,2),
DEPTNO    NUMERIC(2),
FOREIGN KEY (DEPTNO) REFERENCES DEPT(DEPTNO));
```

Funded by the Erasmus+ Programme of the European Union

Foreign key

FK

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	80.12.17	800	(null)	20
7499	ALLEN	SALESMAN	7698	81.02.20	1600	300	30
7521	WARD	SALESMAN	7698	81.02.22	1250	500	30
7566	JONES	MANAGER	7839	81.04.02	2975	(null)	20
7654	MARTIN	SALESMAN	7698	81.09.28	1250	1400	30
7698	BLAKE	MANAGER	7839	81.05.01	2850	(null)	30
7782	CLARK	MANAGER	7839	81.06.09	2450	(null)	10
7788	SCOTT	ANALYST	7566	87.04.19	3000	(null)	20
7839	KING	PRESIDENT	(null)	81.11.17	5000	(null)	10
7844	TURNER	SALESMAN	7698	81.09.08	1500	0	30
7876	ADAMS	CLERK	7788	87.05.23	1100	(null)	20
7900	JAMES	CLERK	7698	81.12.03	950	(null)	30
7902	FORD	ANALYST	7566	81.12.03	3000	(null)	20
7934	MILLER	CLERK	7782	82.01.23	1300	(null)	10

PK

DEPTNO	DNAME	LOC
10	ACCOUNTING	LISBON
20	RESEARCH	PORTO
30	SALES	COIMBRA
40	OPERATIONS	FARO

Funded by the Erasmus+ Programme of the European Union

Creating tables / Restrictions NULL/NOT NULL

NULL/NOT NULL

Columns that can't have null values.

```
CREATE TABLE DEPT
(DEPTNO      INT(2) PRIMARY KEY,
DNAME        VARCHAR(14) NOT NULL,
LOC          VARCHAR(13));
```

DNAME – can´t be null.

Funded by the Erasmus+ Programme of the European Union

Creating tables / Restrictions CHECK

CHECK Specifies a condition in a table.

```
CREATE TABLE DEPT
(DEPTNO      INT(2) PRIMARY KEY,
DNAME        VARCHAR(14) NOT NULL,
LOC          VARCHAR(13) CHECK (LOC='LISBON' OR
                        LOC='PARIS' OR LOC='NEW YORK');
```

The localities introduced can only be LISBON, PARIS or NEW YORK

Funded by the Erasmus+ Programme of the European Union

Creating tables / Restrictions
AUTO INCREMENT

AUTO_INCREMENT - automatically increments the column value. It should only be used with INT datatype.

```
CREATE TABLE EMP
(EMPNO     INT(4)   AUTO_INCREMENT,
ENAME     VARCHAR(10),
JOB       VARCHAR(14),
MGR      NUMERIC(4),
HIREDATE DATE,
SAL      NUMERIC(7,2),
COMM     NUMERIC(7,2),
DEPTNO NUMERIC(2),
PRIMARY KEY (EMPNO),
CONSTRAINT FK FOREIGN KEY (DEPTNO) REFERENCES DEPT(DEPTNO));
```

Funded by the Erasmus+ Programme of the European Union

Altering table structure / Adding columns

```
ALTER TABLE <table name>
ADD (column name   datatype (size)        [<restrictions>],
    -----);
```

Adding the new column "NOTE":

```
ALTER TABLE DEPT
ADD ( NOTE CHAR(1) CHECK(TYPE='S' OR TYPE='L'));
```

Funded by the Erasmus+ Programme of the European Union

Droping a table

```
DROP TABLE <Table Name>;
```

```
DROP TABLE DEPT;
```

All data int the DEPT table will be lost.

Funded by the Erasmus+ Programme of the European Union

Inserting data into tables

```
INSERT INTO <Table Name> [(column1,column2,....)]
VALUES (value1, value2,.....);
```

```
INSERT INTO DEPT (DNAME, DEPTNO)
VALUES ('PRODUCTION',20);
```

Department 20 will be inserted.

```
INSERT INTO DEPT VALUES (10,"SALES", NULL);
```

Department 10 will be inserted.

```
INSERT INTO DEPT VALUES (NULL,"OPERATIONS", NULL);
```

Since the first column is DEPTNO, which is auto incremente, The next department in the auto increment order will be inserted because a NULL value is being used.

Funded by the Erasmus+ Programme of the European Union

View data in Tables

```
SELECT * FROM <table name>;
```

```
SELECT * FROM DEPT;
```

Show all data in the DEPT table.

Funded by the Erasmus+ Programme of the European Union

Inserting data into tables

- Only one line per command;
- May omit column names if the data is inserted in the order that the columns were create.
- Use word NULL for null values.

Funded by the Erasmus+ Programme of the European Union

Deleting data from tables

```
DELETE FROM <Table Name> WHERE
<condition>;
```

```
DELETE FROM DEPT;
```

Deletes all lines from the department table.

Funded by the Erasmus+ Programme of the European Union

Deleting data from tables

```
DELETE FROM DEPT WHERE DEPTNO=10;
```

Only the lines that respect the WHERE condition will be deleted.

Department 10 will be deleted.

Updating data

```
UPDATE <table name>
       SET    column = (value, expression,...),
              column = (value, expression,...),....
       WHERE <condition>;
```

```
UPDATE DEPT
       SET LOC = 'LISBON',
              DNAME='OPERATIONS'
WHERE DEPTNO=10;
```

Only updates the lines that respect the conditions LOC = 'LISBON', DNAME='FABRICO'

Deleting a Database

Command to delete the database CLASSES and all the tables included.

```
DROP DATABASE CLASSES;
```

Funded by the Erasmus+ Programme of the European Union

Database Integrity

MySQL validates the intregrity of the database in the following commands:

- INSERT;
- UPDATE;
- DELETE.