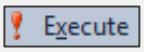
SQL DDL / Intro Assignment

PLEASE SUBMIT ALL YOUR **.sql** FILES TO MOODLE

In this assignment, always execute the given SQL statements ***one by one*** as follows:

First, highlight the statement and then click on 

Download this PDF file and ***copy/paste*** the SQL code from this PDF file to the Query Editor window.

TASK 1. Introduction

=> Save your work on Task 1 to a file named **DDL_Intro_1_***YourSurname***.sql**

Create tables **Department** and **Employee** by executing the following two **CREATE TABLE** statements:

```
-- Task 1
CREATE TABLE Department (
    deptno INTEGER NOT NULL,
    deptname VARCHAR(50) NOT NULL,
    CONSTRAINT PK_Department PRIMARY KEY(deptno)
)

CREATE TABLE Employee (
    empno INTEGER NOT NULL,
    empname VARCHAR(50) NOT NULL,
    deptno INTEGER NOT NULL,
    CONSTRAINT PK_Employee PRIMARY KEY(empno),
    CONSTRAINT FK_Employee_Department FOREIGN KEY(deptno) REFERENCES Department(deptno)
)
```

Insert some data into these tables by executing the following **INSERT** statements:

```
INSERT INTO Department(deptno, deptname) VALUES (10, 'Sales')

INSERT INTO Department(deptno, deptname) VALUES (20, 'Marketing')

INSERT INTO Employee(empno, empname, deptno) VALUES
(1, 'John',  10),
(2, 'Susan', 20),
(3, 'Mary',  10)
```

Execute the following **SELECT** statements ***one by one*** to see some of the ***data*** that is saved in the database:

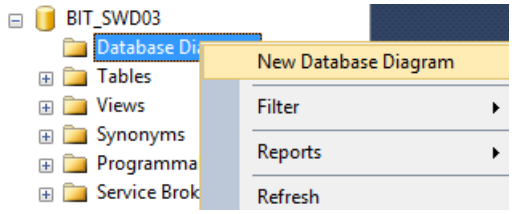
```
SELECT deptno, deptname
  FROM Department

SELECT deptname
  FROM Department
 WHERE deptno = 10

SELECT deptname
  FROM Department
  ORDER BY deptname

SELECT empname, deptname
  FROM Employee
  JOIN Department ON (Department.deptno = Employee.deptno)
  ORDER BY empname
```

NB! If you want to ***recreate an existing table***, then you have to remove the table first by executing a DROP TABLE statement. For example, as follows: `DROP TABLE Employee`

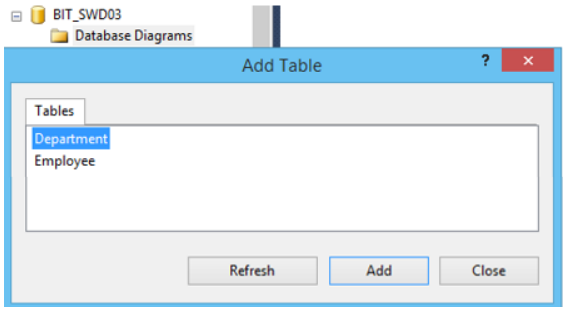
Execute the following **SELECT** statements ***one by one*** to see some of the ***metadata*** that the DBMS has saved in the database:

```
SELECT *
  FROM INFORMATION_SCHEMA.TABLES

SELECT TABLE_NAME, COLUMN_NAME, DATA_TYPE, CHARACTER_MAXIMUM_LENGTH
  FROM INFORMATION_SCHEMA.COLUMNS

SELECT TABLE_NAME, CONSTRAINT_NAME, COLUMN_NAME
  FROM INFORMATION_SCHEMA.CONSTRAINT_COLUMN_USAGE
  ORDER BY TABLE_NAME
```

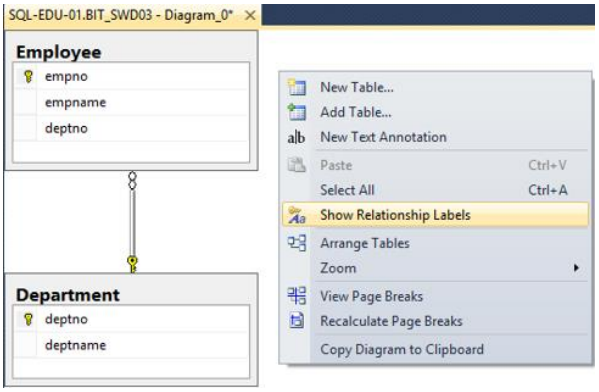
TASK 2. Creating a database diagram (Nothing is required to be submitted on TASK 2)



Step 1: Create a new empty Database Diagram.



Step 2: Select the tables that you want to add to the diagram and click Add.

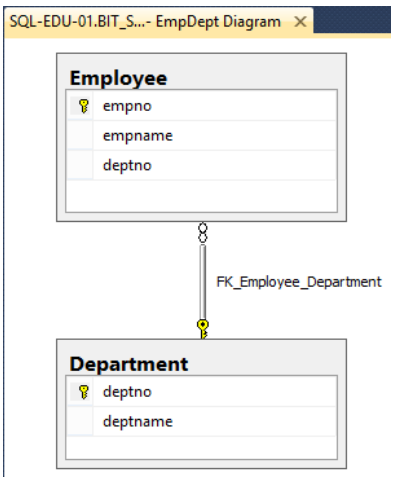


Step 3: Finalise the diagram

Right-click on the canvas and select **Show Relationship Labels**

Right-click on the canvas and select **Arrange Tables**

When you save the diagram SSMS asks you to enter a name for the diagram. Save the diagram as "EmpDept Diagram"



Finally, the database diagram should look like the one on the left.

Congratulations! You have created your first database tables and a database diagram that documents the structure.

TASK 3. Database Queries => Save your work on Task 3 to a file named **DDL_Intro_3_***YourSurname***.sql**

SQL Queries (please *see Task 1 for examples*)

3.1 Write and execute an INSERT statement that ***inserts a new department*** into the database. The department number is 30 and department name is 'Management'.

3.2 Write and execute an INSERT statement that ***inserts a new employee*** into the database. The employee number is 4 and employee name is 'Athena'. She works for the Management department.

3.3 Write and execute a SELECT statement that ***lists names of all employees***. Do not display anything else but employee name.

3.4 Write and execute a SELECT statement that ***displays the name of the employee whose employee number is 2***.

3.5 Write and execute a SELECT statement that ***lists names of those employees who work for department 10***. Display employee name and department number.

Task 4. => Save your work on Task 4 to a file named **DDL_Intro_4_***YourSurname***.sql**

Create Tables (Please *see Task 1 for code examples*)

Suppose the following relation schemas:

Cyclist (cyclistNumber , familyName, givenName, teamNumber)
Team (teamNumber , teamName)

4.1a Create **Cyclist** and **Team** tables. Determine column data types yourself. Create the required ***primary key*** and ***foreign key*** constraints. NB! There will be exactly one foreign key constraint.

4.1b Create a new database diagram ***Team_Cyclist_diagram*** that shows the Cyclist table, Team table and the relationship between these two tables.

4.2 Write and execute INSERT statements that populate the Cyclist table with at least 3 rows and Team table at least with 2 rows. You can create the data (cyclist names etc.) yourself.

4.3 Test that the primary key and foreign key constraints work as required. That is, *try to insert some rows that the DBMS should not accept*.

SQL Queries (Please *see Task 1 for code examples*)

4.4 Write and execute a SELECT statement that ***lists all teams*** in alphabetical order. Display team name only.

4.5 Write and execute a SELECT statement that ***lists all cyclists.*** Display family name and given name only.

4.6 Write and execute a SELECT statement that ***lists all cyclists with team names.*** Display family name, given name, and team name for each cyclist.