

SQL DDL on MySQL – tasks for students:

The database UNIVERSITY consists of the following tables:
TEAMS, EMPLOYEES, PROJECTS, SALARIES

1. Connect to database UNIVERSITY as the **root** user.
2. Create new tables: **teams1** and **emp1** that satisfy the following conditions:
 - a) The table **teams1** should consist of the following columns:
name char(15)
teamid smallint
manid smallint,
The teamid is the primary key of the table **team1**. The table should be created in the **InnoDB** database engine!
 - b) The table **emp1** should consist of the following columns:
empid smallint
gender char(1)
birthdate datetime
name char(15)
teamid smallint

The table should be created in the **InnoDB** database engine!
The **empid** is the primary key of the table **emp1**. Additionally, the columns: **name** and **birthdate** should not have null values.
An index should be created on the column **teamid**.
3. Inserting data to the tables:
 - a) Insert all rows from the table **TEAMS** into the table **teams1**, using the *INSERT* command.
 - b) Insert all rows from the table **EMPLOYEES** to the table **emp1** using the *INSERT* command.
4. Modify the tables as follows (keep the proper order):
 - a) In the table **team1** should exist a foreign key on **manid**, which references to the primary key in the **emp1** table. Define the necessary constraints that deny deletion of a row in the **emp1** table, if there are related rows in the **team1** table.
 - b) In the table **emp1** should exist a foreign key on **teamid**, which references to primary key in the **team1** table. Define the necessary constraints that for any deletion of a row from the **team1** table, the related rows of the **emp1** table are set to null (in the foreign key).
 - c) Draw a diagram of the tables, make picture and add to the report.
5. Check the created integrity constraints in the database. (insert Null values, delete dependent/related rows, change values of a primary key or a foreign key in a way that violates the integrity constraints).
6. Creating users and granting them privileges:
 - a) *Connect to database as the root user:* Create user *user1* with password *user1* and grant him the SELECT privilege on the EMPLOYEES table.
 - b) *Connect as user1:* Display the EMPLOYEES table

- c) Try to drop the EMPLOYEES table
- d) Display the SUBJECTS table
- e) *Connect as root*: Grant the *user1* user the delete privilege on the TEAMS table
- f) *Connect as user1*: Display the TEAMS table
- g) Drop the team with *teamid =1*
what has happened? Why?
- h) *Connect as root*: Grant the user *user1* the SELECT privilege on the TEAMS table.
- i) *Connect as user1*: Drop the team with *teamid =1*
what has happened? Why?
- j) *Connect as root*: Create the view called DATABASES, which includes the employee ids and names of these employees that belong to the team DATABASES.
- k) Grant the *user1* user the SELECT privilege on the view DATABASES
- l) *Connect as user1*: Display the DATABASES view
- m) Display the EMPLOYEES table
- n) Update data using the DATABASES view (try change name of a chosen employee)
what has happened? Why?
- o) *Connect as root*: Grant the *user1* user the UPDATE privilege on the DATABASES view
- p) *Connect as user1*: Update data using the DATABASES view (try change name of chosen employee)
what has happened? Why?
- q) *Connect as root*: Update data using the DATABASES view (try change name of chosen employee)
what has happened? Why?
- r) Create the view DATABASES1 including the names of employees.
- s) Update data using the DATABASES1 view (try change name of chosen employee)
what has happened? Why?
- t) Grant *user1* user the SELECT and UPDATE privileges on the DATABASES1 view.
- u) *Connect as user1*: Update data using the view DATABASES1 (try change name of chosen employee)
what has happened? Why?
- v) *Connect as root*: Create user *sec1*
- w) Grant the *sec1* user the SELECT privilege on the SUBJECTS table.
- x) *Connect as sec1*: Display the SUBJECTS table
- y) Grant the *user1* the SELECT privilege on the SUBJECTS table
what has happened? Why?
- z) *Connect as root*: Grant the *sec1* user the SELECT privilege on the SUBJECTS table with grant option.
- aa) *Connect as sec1*: Grant the *user1* user the SELECT privilege on the SUBJECTS table
what has happened? Why?
- bb) *Connect as user1*: Display the SUBJECTS table