1. Create Database

You now know how to create database using the GUI of the Workbench. Now it’s time to create it using SQL queries.

In that task (and the several following it) you will be required to create the database from the previous exercise

using only SQL queries. Firstly, just create new database named minions.

2. Create Tables

In the newly created database Minions add table minions (id, name, age). Then add new table towns (id, name). Set

id columns of both tables to be primary key as constraint. Submit your create table queries in Judge together for

both tables (one after another separated by “;”) as Run queries and check DB.

3. Alter Minions Table

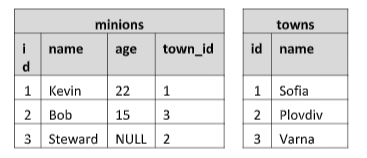
Change the structure of the Minions table to have new column town\_id that would be of the same type as the id

column of towns table. Add new constraint that makes town\_id foreign key and references to id column of towns

table. Submit your create table query in Judge as MySQL run skeleton, run queries and check DB

4. Insert Records in Both Tables

Populate both tables with sample records given in the table below.



Use only insert SQL queries. Submit your INSERT statements in Judge as Run skeleton, run queries and check DB.

5. Truncate Table Minions

Delete all the data from the minions table using SQL query. Submit your query in Judge as Run skeleton, run

queries and check DB.

6. Drop All Tables

Delete all tables from the minions database using SQL query. Submit your query in Judge as Run skeleton, run

queries and check DB.

7. Create Table People

Using SQL query create table “people” with columns:

id – unique number for every person there will be no more than 2 31 -1people. (Auto incremented)

name – full name of the person will be no more than 200 Unicode characters. (Not null)

picture – image with size up to 2 MB. (Allow nulls)

height – In meters. Real number precise up to 2 digits after floating point. (Allow nulls)

weight – In kilograms. Real number precise up to 2 digits after floating point. (Allow nulls)

gender – Possible states are m or f. (Not null)

birthdate – (Not null)

biography – detailed biography of the person it can contain max allowed Unicode characters. (Allow nulls)

Make id primary key. Populate the table with 5 records. Submit your CREATE and INSERT statements in Judge as

Run queries and check DB.

8. Create Table Users

Using SQL query create table users with columns:

id – unique number for every user. There will be no more than 2 63-1 users. (Auto incremented)

username – unique identifier of the user will be no more than 30 characters (non Unicode). (Required)

password – password will be no longer than 26 characters (non Unicode). (Required)

profile\_picture – image with size up to 900 KB.

last\_login\_time

is\_deleted – shows if the user deleted his/her profile. Possible states are true or false.

Make id primary key. Populate the table with 5 records. Submit your CREATE and INSERT statements. Submit your

CREATE and INSERT statements as Run queries and check DB.

9. Change Primary Key

Using SQL queries modify table users from the previous task. First remove current primary key then create new

primary key that would be combination of fields id and username. The initial primary key name on id is pk\_users.

Submit your query in Judge as Run skeleton, run queries and check DB.

10. Set Default Value of a Field

Using SQL queries modify table users. Make the default value of last\_login\_time field to be the current time.

Submit your query in Judge as Run skeleton, run queries and check DB.

11. Set Unique Field

Using SQL queries modify table users. Remove username field from the primary key so only the field id would be

primary key. Now add unique constraint to the username field. The initial primary key name on (id, username) is

pk\_users. Submit your query in Judge as Run skeleton, run queries and check DB.

12. Create SoftUni Database

Now create bigger database called soft\_uni. You will use database in the future tasks. It should hold information

about

towns (id, name)

addresses (id, address\_text, town\_id)

departments (id, name)

employees (id, first\_name, middle\_name, last\_name, job\_title, department\_id, hire\_date,

salary, address\_id)

Id columns are auto incremented starting from 1 and increased by 1 (1, 2, 3, 4…). Make sure you use appropriate

data types for each column. Add primary and foreign keys as constraints for each table. Use only SQL queries.

Consider which fields are always required and which are optional. Submit your CREATE TABLE statements as Run

queries and check DB.

13. Basic Insert

Use the SoftUni database and insert some data using SQL queries.

towns: Sofia, Plovdiv, Varna, Burgas

departments: Engineering, Sales, Marketing, Software Development, Quality Assurance

employees:



Submit your INSERT queries in Judge as Run skeleton, run queries and check DB.

14. Basic Select All Fields

Use the soft\_uni database and first select all records from the towns, then from departments and finally from

employees table. Use SQL queries and submit them to Judge at once. Submit your query statements as Prepare DB

and Run queries.

15. Basic Select All Fields and Order Them

Modify queries from previous problem by sorting:

towns - alphabetically by name

departments - alphabetically by name

employees - descending by salary

Submit your query statements as Prepare DB and Run queries.

16. Basic Select Some Fields

Modify queries from previous problem to show only some of the columns. For table:

towns – name

departments – name

employees – first\_name, last\_name, job\_title, salary

Keep the ordering from the previous problem. Submit your query statements as Prepare DB and Run queries.

17. Increase Employees Salary

Use softuni database and increase the salary of all employees by 10%. Select only salary column from the

employees table. Submit your query statements as Prepare DB and Run queries.