# **Exercises: Data Definition and Data Types**

This document defines the exercise assignments for the "Databases Basics - MySQL" course @ Software University.

#### O. 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.

#### 1. 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 & check DB.

#### 2. 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 & check DB

### 3. Insert Records in Both Tables

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

minions						
id	name	age	town_id			
1	Kevin	22	1			
2	Bob	15	3			
3	Steward	NULL	2			

towns					
id name					
1	Sofia				
2	Plovdiv				
3	Varna				

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

#### 4. Truncate Table Minions

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

## 5. Drop All Tables

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

## 6. Create Table People

Using **SQL query** create table "people" with columns:

id – unique number for every person there will be no more than 2<sup>31</sup>-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 & check DB.

#### 7. Create Table Users

Using **SQL query** create table **users** with columns:

- id unique number for every user. There will be no more than 2<sup>63-1</sup> 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 & check DB.

## 8. 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 & check DB.

#### 9. 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 & check DB.

# 10. 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 & check DB.

#### 11. Movies Database

Using **SQL queries** create **Movies** database with the following entities:

- directors (id, director name, notes)
- genres (id, genre\_name, notes)
- categories (id, category name, notes)
- movies (id, title, director\_id, copyright\_year, length, genre\_id, category\_id, rating, notes)

Set most appropriate data types for each column. Set primary key to each table. Populate each table with 5 records. Make sure the columns that are present in 2 tables would be of the same data type. Consider which fields



© Software University Foundation. This work is licensed under the CC-BY-NC-SA license.















are always required and which are optional. Submit your CREATE TABLE and INSERT statements as Run gueries & check DB.

#### 12. Car Rental Database

Using **SQL queries** create **car\_rental** database with the following entities:

- categories (id, category, daily\_rate, weekly\_rate, monthly\_rate, weekend\_rate)
- cars (id, plate\_number, make, model, car\_year, category\_id, doors, picture, car\_condition, available)
- employees (id, first\_name, last\_name, title, notes)
- customers (id, driver licence number, full name, address, city, zip code, notes)
- rental\_orders (id, employee\_id, customer\_id, car\_id, car\_condition, tank\_level, kilometrage start, kilometrage end, total kilometrage, start date, end date, total days, rate applied, tax rate, order status, notes)

Set most appropriate data types for each column. Set primary key to each table. Populate each table with 3 records. Make sure the columns that are present in 2 tables would be of the same data type. Consider which fields are always required and which are optional. Submit your CREATE TABLE and INSERT statements as Run queries & check DB.

#### 13. Hotel Database

Using **SQL queries** create **Hotel** database with the following entities:

- employees (id, first\_name, last\_name, title, notes)
- customers (account number, first name, last name, phone number, emergency name, emergency\_number, notes)
- room\_status (room\_status, notes)
- room\_types (room\_type, notes)
- bed\_types (bed\_type, notes)
- rooms (room\_number, room\_type, bed\_type, rate, room\_status, notes)
- payments (id, employee\_id, payment\_date, account\_number, first\_date\_occupied, last date occupied, total days, amount charged, tax rate, tax amount, payment total, notes)
- occupancies (id, employee id, date occupied, account number, room number, rate applied, phone\_charge, notes)

Set most appropriate data types for each column. Set primary key to each table. Populate each table with 3 records. Make sure the columns that are present in 2 tables would be of the same data type. Consider which fields are always required and which are optional. Submit your CREATE TABLE and INSERT statements as Run queries & check DB.

## 14. 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 & check DB.

# 15. Backup Database

By using mysqldump command from MySql command line make a backup of the database soft uni, from the previous tasks, into a file named "softuni-backup.sql". Drop your database from Heidi or MySQL Workbench. Then restore the database from the created backup file by using mysql command line.

#### 16. 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:

name	job_title	department	hire_date	salary
Ivan Ivanov Ivanov	.NET Developer	Software Development	01/02/2013	3500.00
Petar Petrov Petrov	Senior Engineer	Engineering	02/03/2004	4000.00
Maria Petrova Ivanova	Intern	Quality Assurance	28/08/2016	525.25
Georgi Terziev Ivanov	CEO	Sales	09/12/2007	3000.00
Peter Pan Pan	Intern	Marketing	28/08/2016	599.88

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

#### 17. 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 & Run queries.

#### 18. 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 & Run queries.

#### 19. 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 & Run queries.

## 20. 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 & Run queries.

#### 21. Decrease Tax Rate

Use hotel database and decrease tax rate by 3% to all payments. Select only tax\_rate column from the payments table. Submit your query statements as Prepare DB & Run queries.

#### 22. Delete All Records

Use Hotel database and delete all records from the occupancies table. Use SQL query. Submit your query statements as Run skeleton, run queries & check DB.















