# Exercises: MySQL and PHP Basics

This document defines the **exercise assignments** for the ["PHP Fundamentals @ Software University.](https://softuni.bg/courses/php-basics/)

### WARNING!

For all tasks export the SQL structure and submit for review alongside with the PHP code. You can combine the exported SQL structure for some tasks, where makes sense to avoid duplications. Just put in comments to clarify. Do not submit any exported data, it is not necessary. You will be using both HeidiSQL and PHP. Every task specifies what to use, but feel free to use PHP even if not explicitly specified.

## Create Database

You now know how to create database using the GUI of the HeidiSQL. Now it’s time to create it using SQL queries and PHP. In that task (and the several following it) you will be required to create the database from the previous exercise **using mainly SQL queries**. Firstly, just **create new database** named **minions.**

## 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**. Test queries with **HeidiSQL** and implement them with **PHP**.

## 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. Use only **HeidiSQL**.

## Insert Records in Both Tables

**Populate both tables** with sample records given in the table below. Test queries with **HeidiSQL** and implement them with **PHP**.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **minions** | | | |  | **towns** | |
| **id** | **name** | **age** | **town\_id** |  | **id** | **name** |
| 1 | Kevin | 22 | 1 |  | 1 | Sofia |
| 2 | Bob | 15 | 3 |  | 2 | Plovdiv |
| 3 | Steward | NULL | 2 |  | 3 | Varna |

## Truncate Table Minions

**Delete all the data** from the **minions** table**.** Test queries with **HeidiSQL** and implement them with **PHP**.

## Drop All Tables

**Delete all tables** from the **minions** database. Test queries with **HeidiSQL** and implement them with **PHP**.

## Create Table People

Test queries with **HeidiSQL** and implement them with **PHP**:

* **id** – unique number for every person there will be **no more than 231-1** **people.** (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**.

## Create Table Users

Test queries with **HeidiSQL** and implement them with **PHP**:

* **id** – unique number for every user. There will be **no more than 263-1 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** – use you skill to decide on the right type
* **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**.

## Change Primary Key

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**. Use **HeidiSQL.**

## Set Default Value of a Field

Make the **default value** of **last\_login\_time** field to be the **current time.** Use **HeidiSQL**.

## 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. Use **HeidiSQL**.

## Movies Database

Create **Movies** database with the following entities. Test queries with **HeidiSQL** and implement them with **PHP**.

* **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 are always required and which are optional.

## Car Rental Database

Create **car\_rental** database with the following entities. Test queries with **HeidiSQL** and implement them with **PHP**.

* **categories** (id, category, daily\_rate, weekly\_rate, monthly\_rate, weekend\_rate)
* **cars** (id, plate\_number, make, model, car\_year, category\_id, doors, picture, 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.

## Hotel Database

Create **Hotel** database with the following entities. Test queries with **HeidiSQL** and implement them with **PHP**.

* **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.

## Create SoftUni Database

Now create bigger database called **SoftUni.** You will use database in the future tasks. It should hold information about. Test queries with **HeidiSQL** and implement them with **PHP**.

* **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 **HeidiSQL**. Consider which fields are always required and which are optional.

## Backup Database

Backup the database **SoftUni** from the previous tasks into a file named “**softuni-backup.bak**”. Delete your database from HeidiSQL. Then restore the database from the created backup.

## Basic Insert

Use the **SoftUni** database and insert some data. Test queries with **HeidiSQL** and implement them with **PHP**.

* **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 Teziev Ivanov | CEO | Sales | 09/12/2007 | 3000.00 |
| Peter Pan Pan | Intern | Marketing | 28/08/2016 | 599.88 |

## Basic Select All Fields

Use the **softuni** database and first select all records from the **towns**, then from **departments** and finally from **employees** table. Test queries with **HeidiSQL** and implement them with **PHP**.

## Basic Select All Fields and Order Them

Modify queries from previous problem by sorting. Test queries with **HeidiSQL** and implement them with **PHP**.

* **towns** - alphabetically by name
* **departments** - alphabetically by name
* **employees** - descending by salary

## Basic Select Some Fields

Modify queries from previous problem to show only **some of the columns**. Test queries with **HeidiSQL** and implement them with **PHP**. For table:

* **towns** – name
* **departments** – name
* **employees** – first\_name, last\_name, job\_title, salary

**Keep the ordering** from the previous problem.

## Increase Employees Salary

Use **softuni** database and **increase the salary** of all employees by **10%.** Select **only salary** column from the **employees** table. Test queries with **HeidiSQL** and implement them with **PHP**.

## Decrease Tax Rate

Use **hotel** database and **decrease tax rate by** **3%** to all payments. Select **only** **tax\_rate** column from the **payments** table. Test queries with **HeidiSQL** and implement them with **PHP**.

## Delete All Records

Use **Hotel** database and **delete all records** from the **occupancies** table. Test queries with **HeidiSQL** and implement them with **PHP**.