# Lab: Table Relations

This document defines the lab assignments for [MySQL Course at Software University](https://softuni.bg/opencourses/databases-basics-mysql).

Get familiar with the **camp** **database**. You will use it in the following exercises.

## Mountains and Peaks

Write a query to create two tables – **mountains** and **peaks** and **link their fields** properly. Tables should have:

* Mountains:
* id
* name
* Peaks:
* id
* name
* mountain\_id

Check your solutions using the "**Run Queries and Check DB**" strategy.

**CREATE TABLE `mountains` (**

**id INT PRIMARY KEY,**

**`name` VARCHAR(20)**

**);**

**CREATE TABLE `peaks` (**

**id INT PRIMARY KEY,**

**`name` VARCHAR(20),**

**mountain\_id INT,**

**CONSTRAINT fk\_peaks\_mountains FOREIGN KEY (mountain\_id)**

**REFERENCES mountains (id)**

**);**

## Trip Organization

Write a query to retrieve information about SoftUni camp's transportation organization. Get information about the drivers (name and id) and their vehicle type. Submit your queries using the "**MySQL prepare DB and Run Queries**" strategy.

### Example

|  |  |  |
| --- | --- | --- |
| **driver\_id** | **vehicle\_type** | **driver\_name** |
| 1 | bus | Simo Sheytanov |
| 2 | van | Roli Dimitrova |
| 1 | van | Simo Sheytanov |
| … | … | … |

## SoftUni Hiking

Get information about the hiking **routes** – starting point and ending point, and their **leaders** – name and id. Submit your queries using the "**MySQL prepare DB and Run Queries**" strategy.

**SELECT**

**starting\_point,**

**end\_point,**

**leader\_id,**

**CONCAT(first\_name, ' ', last\_name) AS leader\_name**

**FROM**

**routes AS r**

**JOIN**

**campers AS c ON r.leader\_id = c.id;**

### Example

|  |  |  |  |
| --- | --- | --- | --- |
| **route\_starting\_point** | **route\_ending\_point** | **leader\_id** | **leader\_name** |
| Hotel Malyovitsa | Malyovitsa Peak | 3 | RoYaL Yonkov |
| Hotel Malyovitsa | Malyovitsa Hut | 3 | RoYaL Yonkov |
| Ribni Ezera Hut | Rila Monastery | 3 | RoYaL Yonkov |
| Borovets | Musala Peak | 4 | Ivan Ivanov |

## Delete Mountains

Drop tables from the task 1.

Write a query to create a one-to-many relationship between a table, holding information about   
mountains (id, name) and other - about peaks (id, name, mountain\_id), so that when a mountain   
gets removed from the database, all his peaks are deleted too.

Submit your queries using the "**MySQL run queries & check DB**" strategy.

**CREATE TABLE `mountains` (**

**`id` INT PRIMARY KEY AUTO\_INCREMENT,**

**`name` VARCHAR(20) NOT NULL**

**);**

**CREATE TABLE `peaks` (**

**`id` INT PRIMARY KEY AUTO\_INCREMENT,**

**`name` VARCHAR(20) NOT NULL,**

**`mountain\_id` INT,**

**CONSTRAINT `fk\_mountain\_id` FOREIGN KEY (`mountain\_id`)**

**REFERENCES `mountains` (`id`)**

**ON DELETE CASCADE**

**);**