Lab: Table Relations

This document defines the lab assignments for [“Databases Basics with MySQL Course” at Software University](#).

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

1. 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.

2. 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
1	van	Simo Sheytanov
2	van	Roli Dimitrova
…	…	…

3.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.

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

4.Delete Mountains

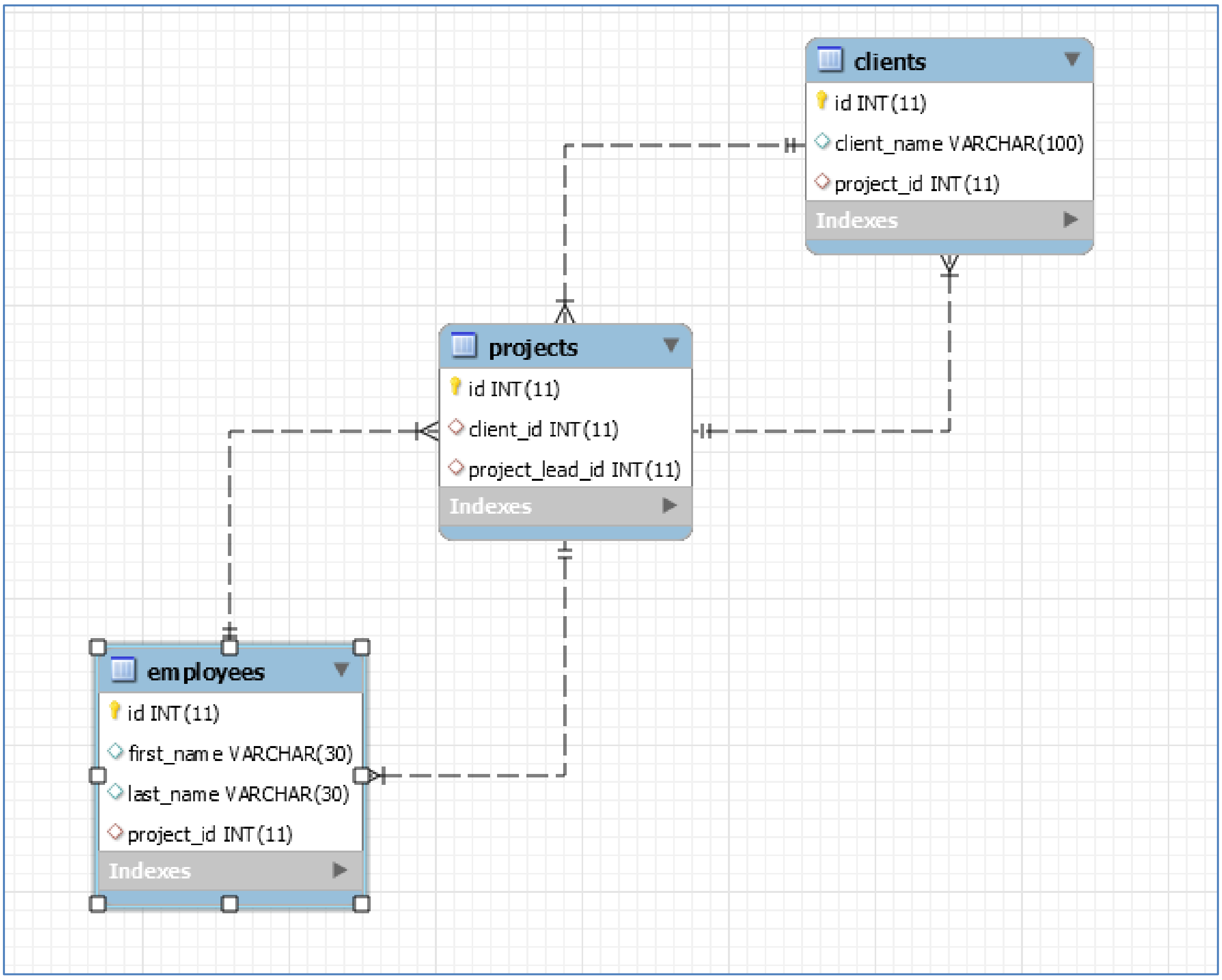
Drop tables from the task 1.

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

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

5. Project Management DB*

Write a query to create a project management db according to the following E/R Diagram:



Submit your queries using the "**MySQL Run Queries and Check DB**" strategy.

Exercises: Table Relations

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

1. One-To-One Relationship

Create two tables as follows. Use appropriate data types.

persons			
person_id	**first_name**	**salary**	**passport_id**
1	Roberto	43300.00	102
2	Tom	56100.00	103
3	Yana	60200.00	101

passports	
passport_id	**passport_number**
101	N34FG21B
102	K65LO4R7
103	ZE657QP2

Insert the data from the example above.

- Alter table **persons** and make **`person_id`** a **`primary key`**.
- Create a **`foreign key`** between **persons** and **passports** by using the **`passport_id`** column.
- Think about which passport field should be **`UNIQUE`**.

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

2. One-To-Many Relationship

Create two tables as follows. Use appropriate data types.

manufacturers		
manufacturer_id	**name**	**established_on**
1	BMW	01/03/1916
2	Tesla	01/01/2003
3	Lada	01/05/1966

models		
model_id	**name**	**manufacturer_id**
101	X1	1
102	i6	1
103	Model S	2
104	Model X	2
105	Model 3	2
106	Nova	3

Insert the data from the example above.

- Add primary and foreign keys.

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

3. Many-To-Many Relationship

Create three tables as follows. Use appropriate data types.

students	
student_id	**name**
1	Mila
2	Toni
3	Ron

exams	
exam_id	**name**
101	Spring MVC
102	Neo4j
103	Oracle 11g

students_exams	
student_id	**exam_id**
1	101
1	102
2	101
3	103
2	102
2	103

Insert the data from the example above.

- Add primary and foreign keys.
- Have in mind that the table `student_exams` should have a **composite** primary key.

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

4. Self-Referencing

Create a single table as follows. Use appropriate data types.

teachers		
teacher_id	**name**	**manager_id**
101	John	
102	Maya	106
103	Silvia	106
104	Ted	105
105	Mark	101
106	Greta	101

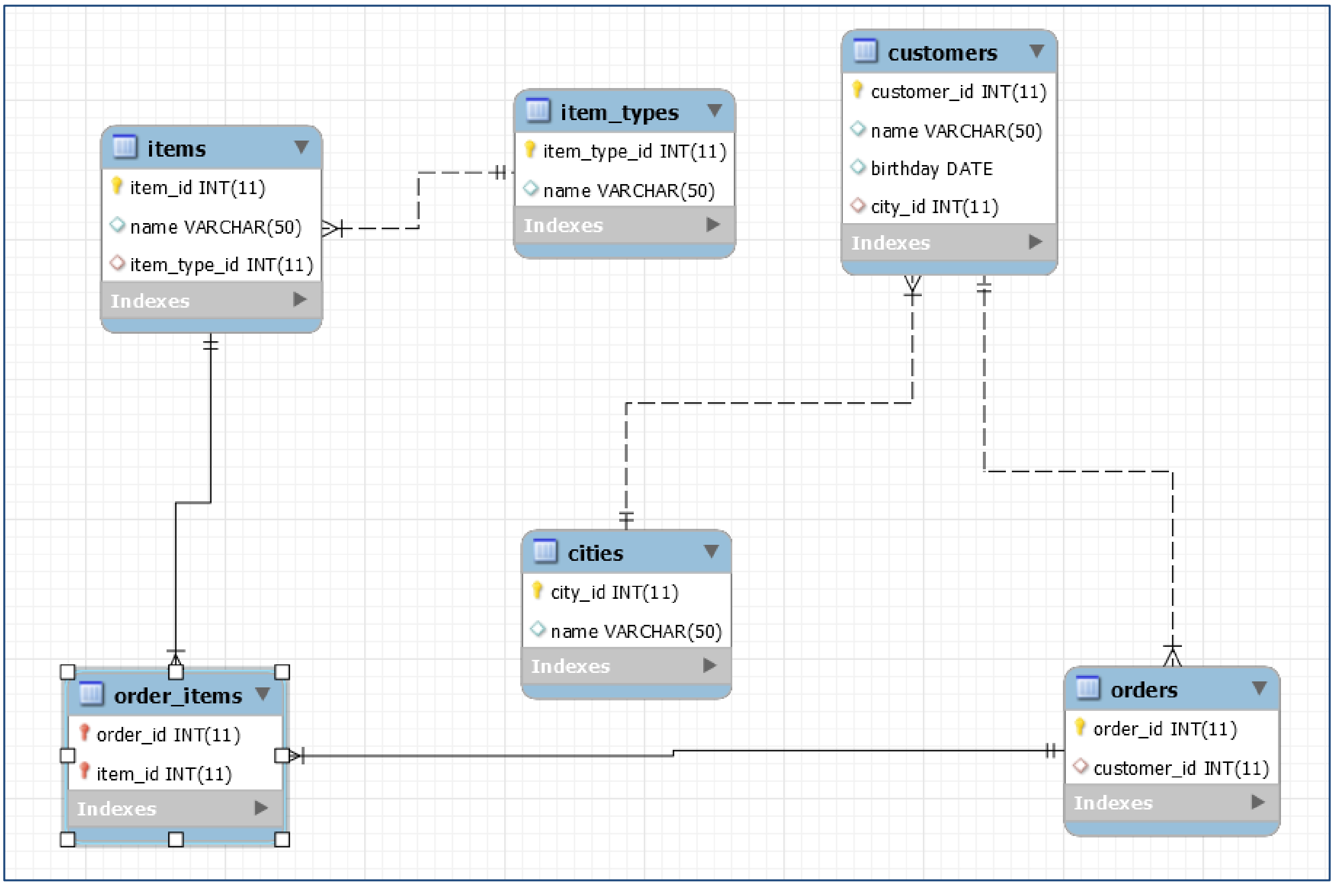
Insert the data from the example above.

- Add primary and foreign keys.
- The foreign key should be between `manager_id` and `teacher_id`.

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

5. Online Store Database

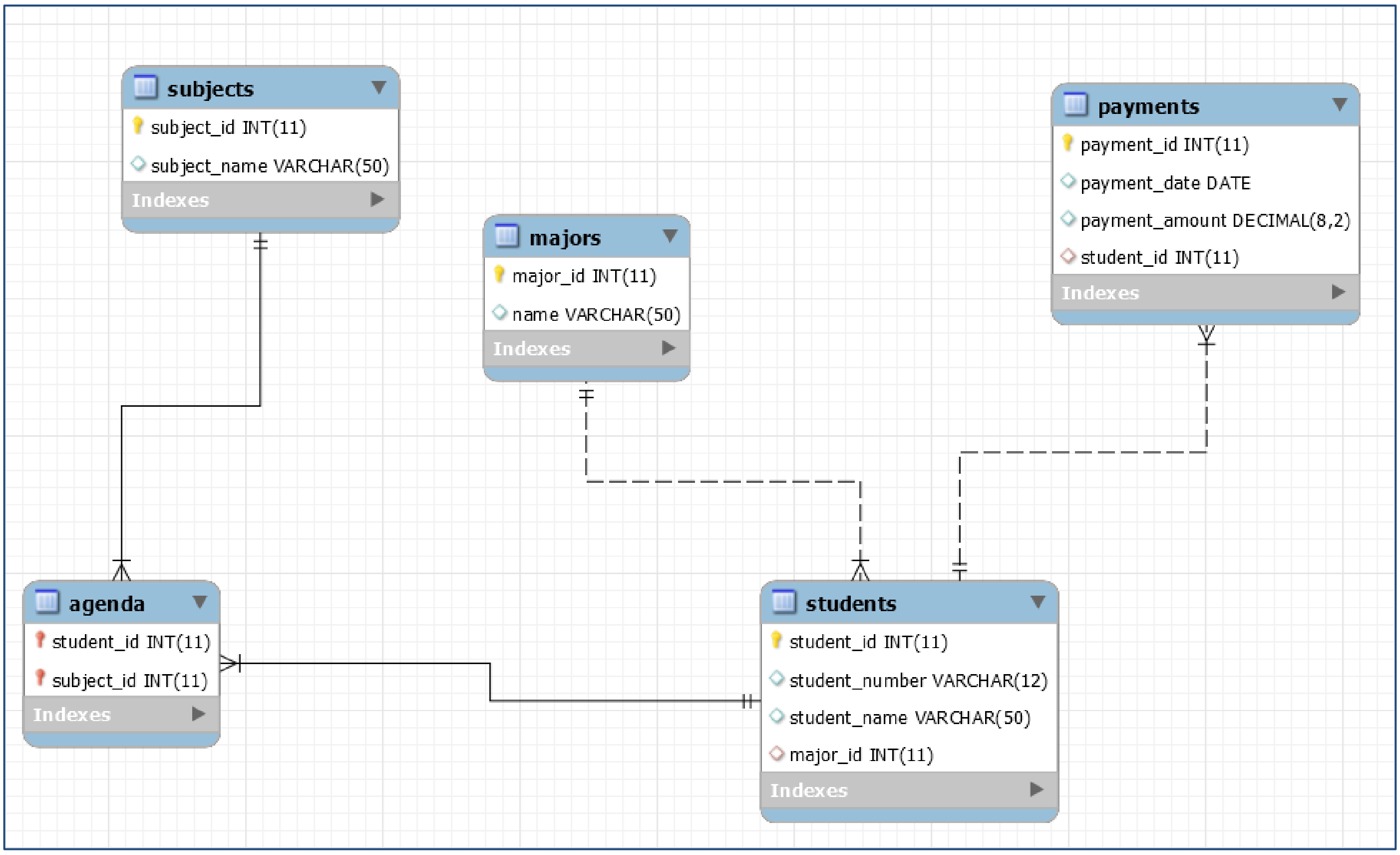
Create a new database and design the following structure:



Submit your queries by using “**MySQL run queries & check DB**” strategy.

6. University Database

Create a new database and design the following structure:



Submit your queries by using “**MySQL run queries & check DB**” strategy.

7. SoftUni Design

Create an E/R Diagram of the SoftUni Database. There are some special relations you should check out: **employees** are **self-referenced (manager_id)** and **departments** have **One-to-One** with the **employees (manager_id)** while the **employees** have **One-to-Many** (**department_id**). You might find it interesting how it looks on a diagram. ☺

8. Geography Design

Create an E/R Diagram of the Geography Database.

9. Peaks in Rila

Display all peaks for "**Rila**" **mountain_range**. Include:

- **mountain_range**
- **peak_name**
- **peak_elevation**

Peaks should be sorted by **peak_elevation** descending.

Example

mountain_range	peak_name	peak_elevation
Rila	Musala	2925
…	…	…

Submit your queries by using "**MySQL prepare DB & run queries**" strategy.