# Exercises: Table Relations

This document defines the **exercise assignments** for the [MySQL course @ Software University.](https://softuni.bg/opencourses/databases-basics-mysql)

## One-To-One Relationship

Create two tables as follows. Use appropriate data types.

|  |  |  |  |
| --- | --- | --- | --- |
| **people** | | | |
| **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 **people** and make person\_ida primary key.
* Create a foreign key between **people** and **passports** by using the passport\_id column.
* Think about which passport field should be UNIQUE.
* Format salary to **second** digit after decimal point.

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

**CREATE TABLE `people` (**

**`person\_id` INT NOT NULL,**

**`first\_name` VARCHAR(20),**

**`salary` DECIMAL(7 , 2 ),**

**`passport\_id` INT UNIQUE**

**);**

**CREATE TABLE `passports` (**

**`passport\_id` INT NOT NULL UNIQUE,**

**`passport\_number` VARCHAR(20)**

**);**

**ALTER TABLE `people`**

**ADD PRIMARY KEY (`person\_id`);**

**ALTER TABLE `people`**

**ADD CONSTRAINT `fk\_people\_passports`**

**FOREIGN KEY (`passport\_id`) REFERENCES passports(passport\_id);**

**INSERT INTO `passports`**

**(`passport\_id`, `passport\_number`)**

**VALUES**

**('101', 'N34FG21B'),**

**('102', 'K65LO4R7'),**

**('103', 'ZE657QP2');**

**INSERT INTO `people`**

**(`person\_id`, `first\_name`, `salary`, `passport\_id`)**

**VALUES**

**('1', 'Roberto', ROUND('43300', 2), '102'),**

**('2', 'Tom', ROUND('56100', 2), '103'),**

**('3', 'Yana', ROUND('60200', 2), '101');**

## 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.

**CREATE TABLE `manufacturers` (**

**`manufacturer\_id` INT PRIMARY KEY AUTO\_INCREMENT,**

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

**`established\_on` date**

**);**

**CREATE TABLE `models` (**

**`model\_id` INT PRIMARY KEY AUTO\_INCREMENT,**

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

**`manufacturer\_id` INT NOT NULL,**

**CONSTRAINT `fk\_models\_manufacturers`**

**FOREIGN KEY (`manufacturer\_id`) REFERENCES `manufacturers`(`manufacturer\_id`)**

**);**

**INSERT INTO `manufacturers`**

**(`manufacturer\_id`, `name`, `established\_on`)**

**VALUES**

**(1, 'BMW', '1916-03-01'),**

**(2, 'Tesla', '2003-01-01'),**

**(3, 'Lada', '1966-05-01');**

**INSERT INTO `models`**

**(`model\_id`, `name`, `manufacturer\_id`)**

**VALUES**

**('101', 'X1', 1),**

**('102', 'i6', 1),**

**('103', 'Model S', 2),**

**('104', 'Model X', 2),**

**('105', 'Model 3', 2),**

**('106', 'Nova', 3);**

## Many-To-Many Relationship

Create three tables as follows. Use appropriate data types.

|  |  |
| --- | --- |
| **exams** | |
| **exam\_id** | **name** |
| 101 | Spring MVC |
| 102 | Neo4j |
| 103 | Oracle 11g |

|  |  |
| --- | --- |
| **students** | |
| **student\_id** | **name** |
| 1 | Mila |
| 2 | Toni |
| 3 | Ron |

|  |  |
| --- | --- |
| **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**.**

**CREATE TABLE `students` (**

**`student\_id` INT PRIMARY KEY,**

**`name` VARCHAR(50)**

**);**

**CREATE TABLE `exams` (**

**`exam\_id` INT PRIMARY KEY,**

**`name` VARCHAR(225)**

**);**

**CREATE TABLE `students\_exams` (**

**`student\_id` INT,**

**`exam\_id` INT,**

**CONSTRAINT PK\_StudentID\_ExamID PRIMARY KEY (student\_id , exam\_id),**

**CONSTRAINT FK\_StudentsExams\_Students FOREIGN KEY (student\_id)**

**REFERENCES `students` (student\_id),**

**CONSTRAINT FK\_StudentsExams\_ExamID FOREIGN KEY (exam\_id)**

**REFERENCES `exams` (exam\_id)**

**);**

**INSERT INTO `students`**

**(`student\_id`, `name`)**

**VALUES**

**(1, 'Mila'),**

**(2, 'Toni'),**

**(3, 'Ron');**

**INSERT INTO `exams`**

**(`exam\_id`, `name`)**

**VALUES**

**('101', 'Spring MVC'),**

**('102', 'Neo4j'),**

**('103', 'Oracle 11g');**

**INSERT INTO `students\_exams`(`student\_id`,`exam\_id`) VALUES**

**(1, 101),**

**(1, 102),**

**(2, 101),**

**(3, 103),**

**(2, 102),**

**(2, 103);**

## 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**.**

**CREATE TABLE `teachers` (**

**`teacher\_id` INT PRIMARY KEY,**

**`name` VARCHAR(50),**

**`manager\_id` INT);**

**INSERT INTO `teachers`**

**VALUES**

**(101,'John', NULL),**

**(102,'Maya', 106),**

**(103,'Silvia', 106),**

**(104,'Ted', 105),**

**(105,'Mark', 101),**

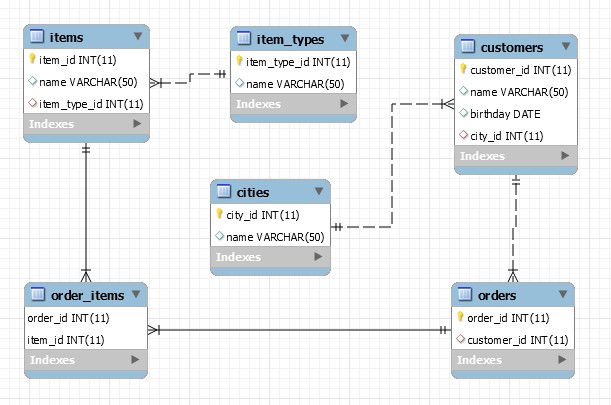
**(106,'Greta', 101);**

**ALTER TABLE `teachers`**

**add constraint fk\_teacher foreign key (manager\_id) references `teachers`(`teacher\_id`);**

## Online Store Database

Create a new database and design the following structure:



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

**CREATE TABLE `items` (**

**`item\_id` INT(11) PRIMARY KEY,**

**`name` VARCHAR(50),**

**`item\_type\_id` INT(11)**

**);**

**CREATE TABLE `item\_types` (**

**`item\_type\_id` INT(11) PRIMARY KEY,**

**`name` VARCHAR(50)**

**);**

**CREATE TABLE `customers` (**

**`customer\_id` INT(11) PRIMARY KEY,**

**`name` VARCHAR(50),**

**`birthday` DATE,**

**`city\_id` INT(11)**

**);**

**CREATE TABLE `order\_items` (**

**`order\_id` INT(11),**

**`item\_id` INT(11),**

**CONSTRAINT PRIMARY KEY (`order\_id` , `item\_id`)**

**);**

**CREATE TABLE `orders` (**

**`order\_id` INT(11) PRIMARY KEY,**

**`customer\_id` INT(11)**

**);**

**CREATE TABLE `cities` (**

**`city\_id` INT(11) PRIMARY KEY,**

**`name` VARCHAR(50)**

**);**

**ALTER TABLE `items`**

**add constraint fk\_item\_type foreign key (item\_type\_id) references `item\_types`(`item\_type\_id`);**

**ALTER TABLE `order\_items`**

**add constraint fk\_item\_id foreign key (item\_id) references `items`(`item\_id`);**

**ALTER TABLE `order\_items`**

**add constraint fk\_order\_id foreign key (order\_id) references `orders`(`order\_id`);**

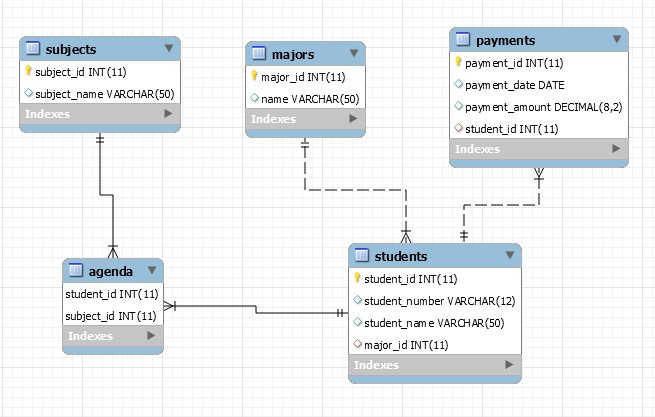
**ALTER TABLE `customers`**

**add constraint fk\_city\_id foreign key (city\_id) references `cities`(`city\_id`);**

**ALTER TABLE `orders`**

**add constraint fk\_customer\_id foreign key (customer\_id) references `customers`(`customer\_id`);**

## University Database

Create a new database and design the following structure:

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

**CREATE TABLE `subjects` (**

**`subject\_id` INT(11) PRIMARY KEY,**

**`subject\_name` VARCHAR(50));**

**CREATE TABLE `majors` (**

**`major\_id` INT(11) PRIMARY KEY,**

**`name` VARCHAR(50));**

**CREATE TABLE `payments` (**

**`payment\_id` INT(11) PRIMARY KEY,**

**`payment\_date` date,**

**`payment\_amount` decimal(8,2),**

**`student\_id` int(11));**

**CREATE TABLE `students` (**

**`student\_id` INT(11) PRIMARY KEY,**

**`student\_number` VARCHAR(12),**

**`student\_name` VARCHAR(50),**

**`major\_id` int(11));**

**CREATE TABLE `agenda` (**

**`student\_id` INT(11),**

**`subject\_id` int(11),**

**constraint primary key(student\_id,subject\_id)) ;**

**ALTER TABLE `students`**

**add constraint fk\_major\_id foreign key (major\_id) references `majors`(`major\_id`);**

**ALTER TABLE `payments`**

**add constraint fk\_student\_id foreign key (student\_id) references `students`(`student\_id`);**

**ALTER TABLE `agenda`**

**add constraint fk\_student\_subject\_id foreign key (student\_id) references `students`(`student\_id`),**

**add constraint fk\_subject\_student\_id foreign key (subject\_id) references `subjects`(`subject\_id`);**

## 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.

**SELECT**

**m.mountain\_range, p.peak\_name, p.elevation AS peak\_elevation**

**FROM**

**mountains AS m**

**JOIN**

**peaks AS p ON p.mountain\_id = m.id**

**WHERE**

**mountain\_range = 'Rila'**

**ORDER BY peak\_elevation DESC;**

### Example

|  |  |  |
| --- | --- | --- |
| **mountain\_range** | **peak\_name** | **peak\_elevation** |
| Rila | Musala | 2925 |
| … | … | … |

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