# Exercises: Table Relations

This document defines the **exercise assignments** for the ["Databases Basics - MSSQL" course @ Software University.](https://softuni.bg/trainings/1436/databases-basics-mssql-september-2016)

## One-To-One Relationship

Create two tables as follows. Use appropriate data types.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Persons** | | | |  | **Passports** | |
| **PersonID** | **FirstName** | **Salary** | **PassportID** |  | **PassportID** | **PassportNumber** |
| 1 | Roberto | 43300.00 | 102 |  | 101 | N34FG21B |
| 2 | Tom | 56100.00 | 103 |  | 102 | K65LO4R7 |
| 3 | Yana | 60200.00 | 101 |  | 103 | ZE657QP2 |

**Insert** the data from the example above.

Alter the **customers** table and make **PersonID** a **primary key**. Create a **foreign key** between **Persons** and **Passports** by using **PassportID** column.

CREATE TABLE Passports(

PassportID INT NOT NULL,

PassportNumber NVARCHAR(30) NOT NULL

)

INSERT INTO Passports

VALUES

(101, 'N34FG21B'),

(102, 'K65LO4R7'),

(103, 'ZE657QP2')

CREATE TABLE Persons(

PersonID INT NOT NULL,

FirstName NVARCHAR(30) NOT NULL,

Salary DECIMAL(20, 2),

PassportID INT NOT NULL

)

INSERT INTO Persons

VALUES

(1, 'Roberto', 43300.00, 102),

(2, 'Tom', 56100.00, 103),

(3, 'Yana', 60200.00, 101)

ALTER TABLE Persons

ADD CONSTRAINT PK\_PersonIDAsPK

PRIMARY KEY(PersonID)

ALTER TABLE Passports

ADD CONSTRAINT PK\_PassportIDAsPK

PRIMARY KEY(PassportID)

ALTER TABLE Persons

ADD FOREIGN KEY(PassportID)

REFERENCES Passports(PassportID)

## One-To-Many Relationship

Create two tables as follows. Use appropriate data types.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Models** | | |  | **Manufacturers** | | |
| **ModelID** | **Name** | **ManufacturerID** |  | **ManufacturerID** | **Name** | **EstablishedOn** |
| 101 | X1 | 1 |  | 1 | BMW | 07/03/1916 |
| 102 | i6 | 1 |  | 2 | Tesla | 01/01/2003 |
| 103 | Model S | 2 |  | 3 | Lada | 01/05/1966 |
| 104 | Model X | 2 |  |  | | |
| 105 | Model 3 | 2 |  |  | | |
| 106 | Nova | 3 |  |  | | |

**Insert** the data from the example above. Add **primary keys** and **foreign keys**.

CREATE TABLE Manufacturers(

ManufacturerID INT PRIMARY KEY IDENTITY,

[Name] NVARCHAR(30) NOT NULL,

EstablishedOn DATE NOT NULL

)

INSERT INTO Manufacturers

VALUES

('BMW', '07/03/1916'),

('Tesla', '01/01/2003'),

('Lada', '01/05/1966')

CREATE TABLE Models(

ModelID INT PRIMARY KEY,

[Name] NVARCHAR(30) NOT NULL,

ManufacturerID INT FOREIGN KEY REFERENCES Manufacturers(ManufacturerID)

)

INSERT INTO Models

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.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Students** | |  | **Exams** | |  | **StudentsExams** | |
| **StudentID** | **Name** | **ExamID** | **Name** | **StudentID** | **ExamID** |
| 1 | Mila | 101 | SpringMVC | 1 | 101 |
| 2 | Toni | 102 | Neo4j | 1 | 102 |
| 3 | Ron | 103 | Oracle 11g | 2 | 101 |
|  | |  | | 3 | 103 |
| 2 | 102 |
| 2 | 103 |

Insert the data from the example above.  
Add **primary keys** and **foreign keys**. Have in mind that table **StudentsExams** should have a **composite primary key**.

CREATE TABLE Students(

StudentID INT PRIMARY KEY NOT NULL,

[Name] NVARCHAR(30) NOT NULL

)

CREATE TABLE Exams(

ExamID INT PRIMARY KEY NOT NULL,

[Name] NVARCHAR(30) NOT NULL

)

INSERT INTO Students

VALUES

(1, 'Mila'),

(2, 'Toni'),

(3, 'Ron')

INSERT INTO Exams

VALUES

(101, 'SpringMVC'),

(102, 'Neo4j'),

(103, 'Oracle 11g')

CREATE TABLE StudentsExams(

StudentID INT FOREIGN KEY REFERENCES Students(StudentID),

ExamID INT FOREIGN KEY REFERENCES Exams(ExamID),

CONSTRAINT PK\_Students\_and\_Exams

PRIMARY KEY(StudentID, ExamID),

)

## Self-Referencing

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

|  |  |  |
| --- | --- | --- |
| **Teachers** | | |
| **TeacherID** | **Name** | **ManagerID** |
| 101 | John | NULL |
| 102 | Maya | 106 |
| 103 | Silvia | 106 |
| 104 | Ted | 105 |
| 105 | Mark | 101 |
| 106 | Greta | 101 |

Insert the data from the example above. Add **primary keys** and **foreign keys**. The **foreign key** should be between **ManagerId** and **TeacherId**.

CREATE TABLE Teachers(

TeacherID INT PRIMARY KEY NOT NULL,

[Name] NVARCHAR(30) NOT NULL,

ManagerID INT FOREIGN KEY REFERENCES Teachers(TeacherID)

)

INSERT INTO Teachers

VALUES

(101, 'John', NULL),

(102, 'Maya', 106),

(103, 'Silvia', 106),

(104, 'Ted', 105),

(105, 'Mark', 101),

(106, 'Greta', 101)

## Online Store Database

Create a new database and design the following structure:



CREATE TABLE ItemTypes(

ItemTypeID INT PRIMARY KEY NOT NULL,

[Name] VARCHAR(50)

)

CREATE TABLE Items(

ItemID INT PRIMARY KEY NOT NULL,

[Name] VARCHAR(50),

ItemTypeID INT FOREIGN KEY REFERENCES ItemTypes(ItemTypeID)

)

CREATE TABLE Cities(

CityID INT PRIMARY KEY NOT NULL,

[Name] VARCHAR(50)

)

CREATE TABLE Customers(

CustomerID INT PRIMARY KEY NOT NULL,

[Name] VARCHAR(50),

Birthday DATE,

CityID INT FOREIGN KEY REFERENCES Cities(CityID)

)

CREATE TABLE Orders(

OrderID INT PRIMARY KEY NOT NULL,

CustomerID INT FOREIGN KEY REFERENCES Customers(CustomerID)

)

CREATE TABLE OrderItems(

OrderID INT FOREIGN KEY REFERENCES Orders(OrderID),

ItemID INT FOREIGN KEY REFERENCES Items(ItemID),

CONSTRAINT PR\_OrderID\_ItemID

PRIMARY KEY(OrderID, ItemID)

)

## University Database

Create a new database and design the following structure:



CREATE TABLE Majors(

MajorID INT PRIMARY KEY NOT NULL,

[Name] NVARCHAR(50)

)

CREATE TABLE Subjects(

SubjectID INT PRIMARY KEY NOT NULL,

SubjectName NVARCHAR(50)

)

CREATE TABLE Students(

StudentID INT PRIMARY KEY NOT NULL,

StudentNumber INT NOT NULL,

StudentName NVARCHAR(50),

MajorID INT FOREIGN KEY REFERENCES Majors(MajorID)

)

CREATE TABLE Payments(

PaymentID INT PRIMARY KEY NOT NULL,

PaymentDate DATE,

PaymentAccount NVARCHAR(50),

StudentID INT FOREIGN KEY REFERENCES Students(StudentID)

)

CREATE TABLE Agenda(

StudentID INT FOREIGN KEY REFERENCES Students(StudentID),

SubjectID INT FOREIGN KEY REFERENCES Subjects(SubjectID),

CONSTRAINT PK\_StudentID\_SubjectID

PRIMARY KEY(StudentId, SubjectID)

)

## SoftUni Design

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



## Geography Design

Create an **E/R Diagram** of the Geography Database.



## \*Peaks in Rila

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

* **MountainRange**
* **PeakName**
* **Elevation**

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

SELECT m.MountainRange, p.PeakName, p.Elevation

FROM Mountains AS m

JOIN Peaks AS p

ON p.MountainId = m.Id

WHERE m.MountainRange = 'Rila'

ORDER BY p.Elevation DESC

### Example

|  |  |  |
| --- | --- | --- |
| **MountainRange** | **PeakName** | **Elevation** |
| Rila | Musala | 2925 |
| … | … | … |