MySQL – Relational Databases: Workshop 1

Workshop contents:

- Writing a SQL script that
  - o Creates tables in a database.
  - o Inserts test data in a database.

Create the tables for the database described here (see the schema below).

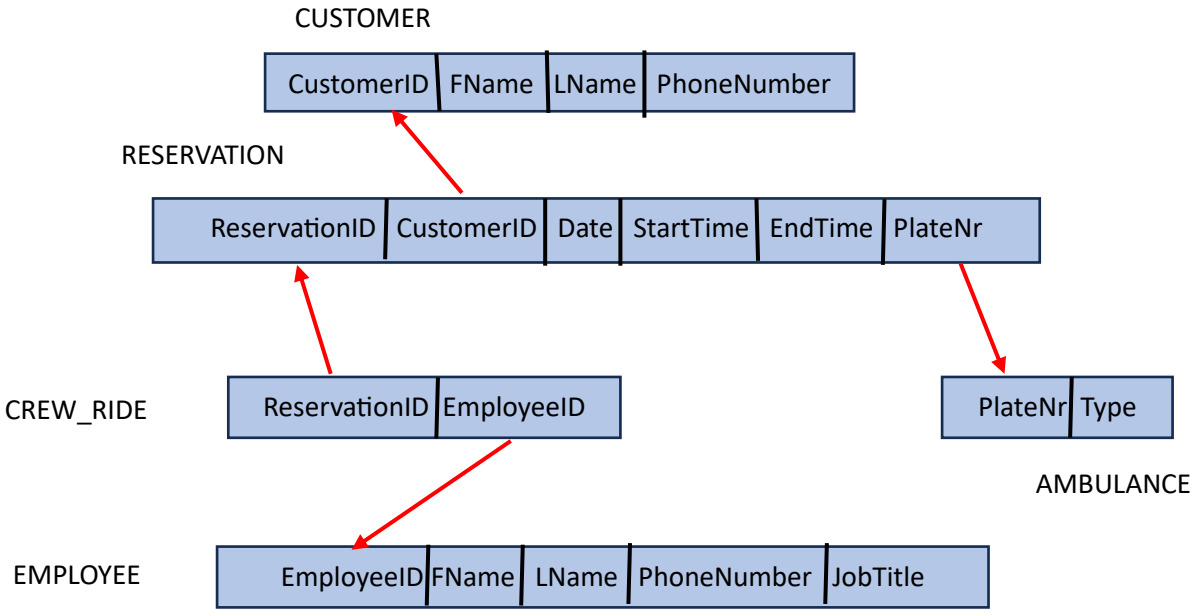
We will use the "Ambulance Transportation Service" case in this workshop. The scenario description of this case is:

A small transportation company provides ambulance services in the Helsinki area. The company has few ambulances equipped for different types of medical transportation situations and employs free-lance professionals (drivers, nurses, doctors, first-aid specialists) to attend a transportation call.
To better manage reservations and allocation of crews and vehicles, the company is planning to implement a simple online reservation system.
The following conditions are given by the company:

- One reservation (call) is associated to one single customer, and the same customer can make many reservations for different situations
- One reservation is always associated to one and only one ambulance, and the same ambulance can be reserved for different situations in different date&times
- One employee can serve in several ambulances, and the crew of an ambulance can be composed by one or more employees

The picture below shows a Relational Schema of the database you should implement during this workshop.

RELATIONAL SCHEMA – AMBULANCE



To create the database tables use the following guidelines:

- Create a separate table for each of the entities: Customer, Reservation, Crew_Ride,Ambulance, Employee

- When adding the columns in your tables, you can decide what's the most appropriate data type for each of the fields. For example: CustomerID can be of type Integer, CUSTOMER.FName can be of type Varchar(16)
- Remember to configure the primary keys in each table

Executing SQL queries on your database

The database you just created only contains tables and fields, but it does not contain any data yet.

Let's now execute one INSERT sql query to add one row into the "customer" table. The query is:

```
INSERT INTO customer values
(1, 'Robert', 'Smith', '035-444673');
```

To add more rows on your tables you can continue with a similar process, writing one INSERT query for each new row you want to add on the table. You will soon however notice that this is a time consuming process if you are planning to add a large amount of rows as it will require many mouse clicks.

A better approach to add a large amount of data (rows) in a database is to first write a SQL script containing all the INSERT queries, and then you run this script on the database.

For example, to add many rows on a table you can write the following query on a text (.sql) file:

```
INSERT INTO CUSTOMER VALUES
(1, 'Robert', 'Smith', '035-444673'),
(2, 'Maria', 'Gimenez', '044778901'),
(3, 'Markku', 'Nieminen', '+3582298976'),
(4, 'Patrick', 'Johansson', '09120987');
```

Creating a SQL script to insert test data into tables

Write a SQL script that contains INSERT SQL sentences to insert the data shown in the tables below. Write your script with for example Notepad++ and save it with an extension ".sql" (for example "ambulance_data.sql"). Once the SQL scrip is ready, import it the same way as you did to import the script that creates the database tables. After importing the script, check that all tables contain the expected rows.

CUSTOMER

CustomerID	FName	LName	PhoneNr
1	Robert	Smith	035-444673
2	Maria	Gimenez	044778901
3	Markku	Nieminen	+3582298976
4	Patrick	Johansson	09120987

EMPLOYEE

EmployeeID	FName	LName	PhoneNr	JobTitle
101	Julia	Fernandez	0987465534	Nurse
102	Tommi	Mäkinen	0357787653	Driver
103	Ella	Jokinen	+3589876223	Driver
104	Hoang	Nguyen	0976353423	Doctor
105	Julius	Nyqvist	045-98764524	Nurse
106	Eveliina	Järvinen	035-998567	Doctor

AMBULANCE

PlateNr	Type
IXC-233	Basic
KJM-001	Emergency
FCE-873	Emergency

RESERVATION

ReservationID	CustomerID	Reservation Date	StartTime	EndTime	PlateNr
2334	4	2024-Jan-20	08:00:00	09:00:00	IXC-233
3442	4	2024-Feb-03	14:00:00	15:30:00	IXC-233
1101	1	2024-Mar-05	13:15:00	14:00:00	KJM-001
4565	3	2024-Apr-30	21:40:00	22:30:00	FCE-873

CREW_RIDE

ReservationID	EmployeeID
2334	102
3442	102
1101	103
1101	104
1101	105
4565	102
4565	105