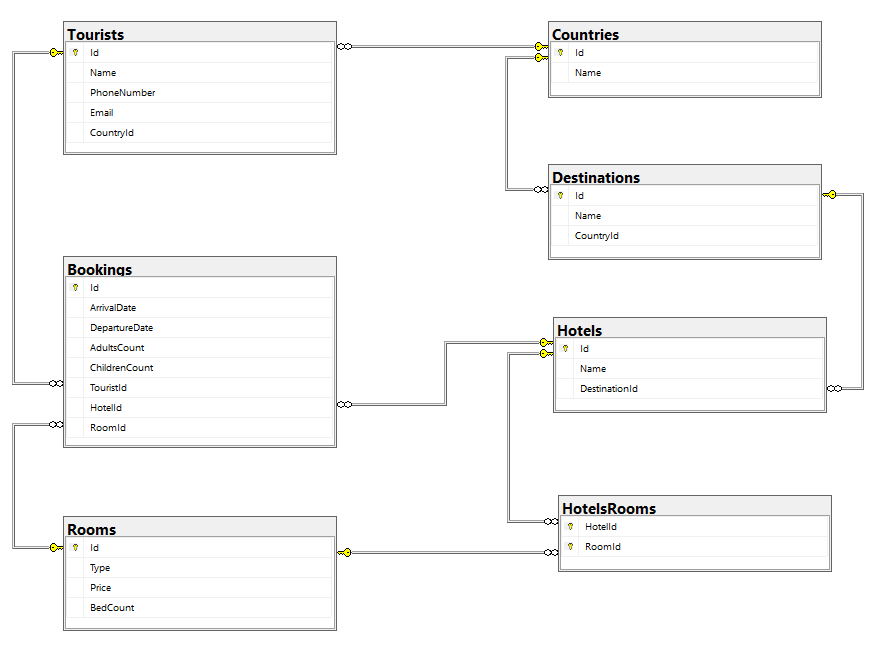
# Database Basics MS SQL Regular Exam – 15 Oct 2023

Exam problems for the ["Database Basics" course @ SoftUni](https://softuni.bg/trainings/4182/ms-sql-september-2023).  
Submit your solutions in the SoftUni Judge system at [Judge](https://judge.softuni.org/Contests/4402/Databases-MSSQL-Server-Regular-Exam-15-Oct-2023).

# Tourist Agency

# Section 1. DDL (30 pts)

You have been given the E/R Diagram of the **TouristAgency** database.



Create a database called **TouristAgency**. You need to create **7 tables**:

* **Countries** – contains information about the countries, in which the destinations and hotels are located, each tourist will also has a country;
* **Destinations** – contains information about the holiday destinations(areas, resorts, etc.);
* **Rooms** – contains information about the rooms (type of room, count of beds);
* **Hotels**– contains information about each hotel;
* **Tourists** – containts information about each tourist, that has booked a room in a hotel;
* **Bookings** – contains information about each booking;
* **HotelsRooms**  – mapping table between hotels and rooms;

**NOTE: Keep in mind that Judge doesn't accept "ALTER" statement and square brackets naming (when the names are not keywords).**

**NOTE: Please keep in mind that in case you have to work with a date, you have to use the exact same data type, described in the models tables. If you don't use the correct type, the Judge system won't accept your submission as correct.**

You have been tasked to **create the tables in the database by the following models**:

### ****Countries****

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| **Id** | **Integer** from **0** to **2,147,483,647** | PK, Unique table identification, Identity |
| **Name** | **String** up to **50** symbols, Unicode | **Null** is **not** allowed |

### Destinations

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| **Id** | **Integer** from **0** to **2,147,483,647** | PK, Unique table identification, Identity |
| **Name** | **String** up to **50** symbols | **Null** is **not** allowed |
| **CountryId** | **Integer** from **0** to **2,147,483,647** | **Relationship** with table **Countries**, **Null** is **not** allowed |

### Rooms

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| **Id** | **Integer** from **0** to **2,147,483,647** | PK, Unique table identification, Identity |
| **Type** | **String** up to **40** symbols | **Null** is **not** allowed |
| **Price** | **Decimal**, up to **18 digits**, **2** of which after the **decimal point** | **Null** is **not** allowed |
| **BedCount** | **Integer** from **0** to **2,147,483,647** | In range between **0 (exclusive)** and **10 (inclusive).**  **Null** is **not** allowed |

### ****Hotels****

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| **Id** | **Integer** from **0** to **2,147,483,647** | PK, Unique table identification, Identity |
| **Name** | **String** up to **50** symbols | **Null** is **not** allowed |
| **DestinationId** | **Integer** from **0** to **2,147,483,647** | **Relationship** with table **Destinations**, **Null** is **not** allowed |

### ****Tourists****

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| **Id** | **Integer** from **0** to **2,147,483,647** | PK, Unique table identification, Identity |
| **Name** | **String** up to **80** symbols, Unicode | **Null** is **not** allowed |
| **PhoneNumber** | **String** up to **20** symbols | **Null** is **not** allowed |
| **Email** | **String** up to **80** symbols | **Null** is allowed |
| **CountryId** | **Integer** from **0** to **2,147,483,647** | Relationship with table **Countries**, **Null** is **not** allowed |

### ****Bookings****

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| **Id** | **Integer** from **0** to **2,147,483,647** | PK, Unique table identification, Identity |
| **ArrivalDate** | **DateTime2** | **Null** is **not** allowed |
| **DepartureDate** | **DateTime2** | **Null** is **not** allowed |
| **AdultsCount** | **Integer** from **0** to **2,147,483,647** | In range between **1 (inclusive)** and **10 (inclusive).**  **Null** is **not** allowed |
| **ChildrenCount** | **Integer** from **0** to **2,147,483,647** | In range between **0 (inclusive)** and **9 (inclusive).**  **Null** is **not** allowed |
| **TouristId** | **Integer** from **0** to **2,147,483,647** | Relationship with table **Tourists**, **Null** is **not** allowed |
| **HotelId** | **Integer** from **0** to **2,147,483,647** | Relationship with table **Hotels**, **Null** is **not** allowed |
| **RoomId** | **Integer** from **0** to **2,147,483,647** | Relationship with table **Rooms**, **Null** is **not** allowed |

### ****HotelsRooms****

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| **HotelId** | **Integer** from **0** to **2,147,483,647** | PK, Unique table identification, Relationship with table **Hotels**, **Null** is **not** allowed |
| **RoomId** | **Integer** from **0** to **2,147,483,647** | PK, Unique table identification, Relationship with table **Rooms**, **Null** is **not** allowed |

## Database design

Submit all of yours **CREATE** **statements** to Judge (only the creation of tables).

# Section 2. DML (10 pts)

**Before you start, you have to import "Dataset.sql". If you have created the structure correctly, the data should be successfully inserted.**

In this section, you have to do some data manipulations:

## Insert

Let's **insert** some sample data into the database. Write a query to add the following records into the corresponding tables. All IDs (Primary Keys) should be **auto-generated**.

### ****Tourists****

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **PhoneNumber** | **Email** | **CountryId** |
| John Rivers | 653-551-1555 | john.rivers@example.com | 6 |
| Adeline Aglaé | 122-654-8726 | adeline.aglae@example.com | 2 |
| Sergio Ramirez | 233-465-2876 | s.ramirez@example.com | 3 |
| Johan Müller | 322-876-9826 | j.muller@example.com | 7 |
| Eden Smith | 551-874-2234 | eden.smith@example.com | 6 |

### ****Bookings****

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ArrivalDate** | **DepartureDate** | **AdultsCount** | **ChildrenCount** | **TouristId** | **HotelId** | **RoomId** |
| 2024-03-01 | 2024-03-11 | 1 | 0 | 21 | 3 | 5 |
| 2023-12-28 | 2024-01-06 | 2 | 1 | 22 | 13 | 3 |
| 2023-11-15 | 2023-11-20 | 1 | 2 | 23 | 19 | 7 |
| 2023-12-05 | 2023-12-09 | 4 | 0 | 24 | 6 | 4 |
| 2024-05-01 | 2024-05-07 | 6 | 0 | 25 | 14 | 6 |

## Update

We've decided to change the **departure date** of the **bookings** that are scheduled to arrive in **December 2023**. The updated departure date for these bookings should be set to **one day later than their original departure date**.

We need to update the **email addresses** of **tourists, whose names contain "MA".** The **new** value of their **email addresses** should be set to **NULL.**

## Delete

In table **Tourists**, delete every tourist, whose Name contains family name "**Smith**". Keep in mind that there could be foreign key constraint conflicts.

# Section 3. Querying (40 pts)

**You need to start with a fresh dataset, so recreate your DB and import the sample data again ("Dataset.sql**"**).**

## Bookings by Price of Room and Arrival Date

Select all **bookings,** ordered by **price** of room (descending), then by **arrival date** (ascending). The **arrival date** should be formatted in the **'yyyy-MM-dd'** format in the query results.

Required columns:

* **ArrivalDate**
* **AdultsCount**
* **ChildrenCount**

### Example

|  |  |  |
| --- | --- | --- |
| **ArrivalDate** | **AdultsCount** | **ChildrenCount** |
| 2023-10-05 | 3 | 1 |
| 2023-11-19 | 4 | 2 |
| 2023-12-10 | 5 | 1 |
| 2023-10-01 | 2 | 0 |
| … | … | … |

## Hotels by Count of Bookings

Select all **hotels** with "**VIP Apartment**" available. Order results by the **count of bookings** (count of all bookings for the specific hotel, not only for VIP apartment) made for every hotel (descending).

Required columns:

* **Id**
* **Name**

### Example

|  |  |
| --- | --- |
| **Id** | **Name** |
| 5 | Saint Ouen Marche Aux Puces |
| 11 | Silken Al-Andalaus Palace |
| 20 | Kivotos |
| 3 | Antica Panada |
| ... | ... |

## Tourists without Bookings

Select all tourists **that haven’t booked a hotel yet**. Order them by name (ascending).

Required columns:

* **Id**
* **Name**
* **PhoneNumber**

### Example

|  |  |  |
| --- | --- | --- |
| **Id** | **Name** | **PhoneNumber** |
| 19 | Ahmet Yilmaz | 777-777-7707 |
| 14 | Friedrich Weber | 434-444-4414 |
| ... | ... | ... |

## First 10 Bookings

Select the first **10** bookings that will **arrive** **before 2023-12-31.** You will need to select the bookings in **hotels with odd-numbered IDs**. Sort the results in **ascending** order, first by **CountryName**, and then by **ArrivalDate**.

Required columns:

* **HotelName**
* **DestinationName**
* **CountryName**

### Example

|  |  |  |
| --- | --- | --- |
| **HotelName** | **DestinationName** | **CountryName** |
| Royal Promenade des Anglais | Nice | France |
| Saint Ouen Marche Aux Puces | Paris | France |
| Saint Ouen Marche Aux Puces | Paris | France |
| … | … | … |

## Tourists booked in Hotels

Select all of the tourists that have a name, **not** ending in "**EZ**", and display the **names of the hotels**, that they have booked a room in. Order by the **price of the room** (descending).

Required columns:

* **HotelName**
* **RoomPrice**

### Example

|  |  |
| --- | --- |
| **HotelName** | **RoomPrice** |
| Kivotos | 600.00 |
| Silken Al-Andalaus Palace | 280.50 |
| Liebesbier Urban Art & Smart Hotel | 280.50 |
| Anklamer Hof | 250.00 |
| Silken Al-Andalaus Palace | 250.00 |
| Silken Al-Andalaus Palace | 250.00 |
| … | … |

## Hotels Revenue

### In this task, you will write an SQL query to calculate the total price of all bookings for each hotel based on the room price and the number of nights guests have stayed. The result should list the hotel names and their corresponding revenue.

* Foreach **Booking** you should **join** data for the **Hotel** and the **Room, using the Id references**;
* **NightsCount** – you will need the **ArrivalDate** and **DepartureDate** for a **DATEDIFF function;**

### Calculate the TotalRevenue by summing the price of each booking, using Price of the Room that is referenced to the specific Booking and multiply it by the NightsCount.

### Group all the bookings by HotelName using the reference to the Hotel.

### Order the results by TotalRevenue in descending order.

Required columns:

* **HotelName**
* **TotalRevenue**

### Example

|  |  |
| --- | --- |
| **HotelName** | **HotelRevenue** |
| Bonvecchiati | 10521.50 |
| Kivotos | 6530.00 |
| Cavo Tagoo | 3666.75 |
| … | … |

# Section 4. Programmability (20 pts)

## Rooms with Tourists

Create a user-defined function, named **udf\_RoomsWithTourists(@name)** that receives a room's type.

The function should return the **total number of tourists** that the **specific room type** has been booked for (**adults + children**).

Hint: A Double Room could be booked for: 2 adults + 0 children, 1 adult + 1 children, 1 adult + 0 children.

### Example

|  |
| --- |
| **Query** |
| **SELECT dbo.udf\_RoomsWithTourists('Double Room')** |
| **Output** |
| **17** |

## Search for Tourists from a Specific Country

Create a **stored procedure**, named **usp\_SearchByCountry(@country)** that receives a country name. The procedure must print full information about all tourists that have an booked a room and have origin from the given country: **Name, PhoneNumber, Email** and **CountOfBookings** (the count of all bookings made)**.** Order them by **Name** (ascending) and **CountOfBookings**(decending).

### Example

|  |  |  |  |
| --- | --- | --- | --- |
| **Query** | | | |
| **EXEC usp\_SearchByCountry 'Greece'** | | | |
| **Output** | | | |
| **Name** | **PhoneNumber** | **Email** | **CountOfBookings** |
| Maria Papadopoulou | 333-303-3333 | maria.papadopoulou@example.com | 3 |
| Nikos Papadopoulos | 303-323-3333 | nikos.papadopoulos@example.com | 3 |