# Database Basics MS SQL Exam – 10.08.2022

Exam problems for the ["Database Basics" course @ SoftUni](https://softuni.bg/trainings/3714/ms-sql-may-2022)

Submit your solutions in the SoftUni Judge system at <https://judge.softuni.bg/>

# 100 National Tourist Sites of Bulgaria

*100 Tourist Sites of Bulgaria is a Bulgarian national movement established in 1966 to promote tourism among Bulgaria's most significant cultural, historic, and natural landmarks. You are given the task to create a database in which you will store information about the sites and the tourists who are visiting them.*

# Section 1. DDL (30 pts)

You have been given the following E/R Diagram.

Graphical user interface

Description automatically generated

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

* **Categories** – contains information about the different **categories of the tourist sites**
* **Locations** – contains information about the **locations of the tourist sites**
* **Sites** – contains information about the **tourist** **sites**
* **Tourists** – contains information about the **tourists**, who are visiting the **tourist sites**
* **SitesTourists** – a many to many mapping table between the sites and the tourists
* **BonusPrizes** – contains information about the **bonus prizes**, which are given to an annual raffle
* **TouristsBonusPrizes** – a many to many mapping table between the tourists and the bonus prizes.

**N.B.**

**Use VARCHAR for strings, not NVARCHAR. Keep in mind that Judge doesn’t accept the “ALTER” statement and square brackets naming when the names are not keywords.**

**Categories**

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

**Locations**

|  |  |  |
| --- | --- | --- |
| **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 |
| **Municipality** | **String** up to **50** symbols | **Null** isallowed |
| **Province** | **String** up to **50** symbols | **Null** isallowed |

**Sites**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| **Id** | **Integer** from **0** to **2,147,483,647** | **PK**, **Unique** table identification, **Identity** |
| **Name** | **String** up to **100** symbols | **Null** is **not** allowed |
| **LocationId** | **Integer** from **0** to **2,147,483,647** | **Relationship** with table **Locations**, **Null** is **not** allowed |
| **CategoryId** | **Integer** from **0** to **2,147,483,647** | **Relationship** with table **Categories**, **Null** is **not** allowed |
| **Establishment** | **String** up to **15** symbols | **Null** is 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 **50** symbols | **Null** is **not** allowed |
| **Age** | **Integer** from **0 to 2,147,483,647** | In range between **0 (inclusive)** and **120 (inclusive).**  **Null** is **not** allowed |
| **PhoneNumber** | **String** up to **20** symbols | **Null** is **not** allowed |
| **Nationality** | **String** up to **30** symbols | **Null** is **not** allowed |
| **Reward** | **String** up to **20** symbols | **Null** is **allowed** |

**SitesTourists**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| **TouristId** | **Integer** from **0** to **2,147,483,647** | **PK**, **Unique** table identification, **Relationship** with table **Tourists.**  **Null** is **not** allowed |
| **SiteId** | **Integer** from **0** to **2,147,483,647** | **PK**, **Unique** table identification, **Relationship** with table **Sites.**  **Null** is **not** allowed |

**BonusPrizes**

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

**TouristsBonusPrizes**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| **TouristId** | **Integer** from **0** to **2,147,483,647** | **PK**, **Unique** table identification, **Relationship** with table **Tourists.**  **Null** is **not** allowed |
| **BonusPrizeId** | **Integer** from **0** to **2,147,483,647** | **PK**, **Unique** table identification, **Relationship** with table **BonusPrizes.**  **Null** is **not** allowed |

## Database design

Submit all of your **created** **statements** to Judge (only creation of tables).

# Section 2. DML (10 pts)

**Before you start you have to import "*01.* *DDL\_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 should be **auto-generated**.

****Tourists****

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Age** | **PhoneNumber** | **Nationality** | **Reward** |
| Borislava Kazakova | 52 | +359896354244 | Bulgaria | NULL |
| Peter Bosh | 48 | +447911844141 | UK | NULL |
| Martin Smith | 29 | +353863818592 | Ireland | Bronze badge |
| Svilen Dobrev | 49 | +359986584786 | Bulgaria | Silver badge |
| Kremena Popova | 38 | +359893298604 | Bulgaria | NULL |

**Sites**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **LocationId** | **CategoryId** | **Establishment** |
| Ustra fortress | 90 | 7 | X |
| Karlanovo Pyramids | 65 | 7 | NULL |
| The Tomb of Tsar Sevt | 63 | 8 | V BC |
| Sinite Kamani Natural Park | 17 | 1 | NULL |
| St. Petka of Bulgaria – Rupite | 92 | 6 | 1994 |

## Update

For some of the **tourist** **sites** there are no clear records when they were established, so you need to **update** the column **'Establishment'** for those records by putting the text **'(not defined)'**.

## Delete

For this year´s raffle it was decided to **remove** the **Sleeping bag** fromthe **bonus prizes.**

**N.B.** 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 (01. DDL\_Dataset.sql). DO NOT CHANGE OR INCLUDE DATA FROM DELETE, INSERT AND UPDATE TASKS!!!**

## Tourists

Extract information about all the **Tourists** – **name, age, phone number and nationality. Order** the result by **nationality (ascending)**, then by **age (descending),** and then by **tourist name (ascending).**

### Example:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Age** | **PhoneNumber** | **Nationality** |
| Danny Kane | 39 | +32487454880 | Belgium |
| Krasen Krasenov | 62 | +359897653265 | Bulgaria |
| Pavel Mateev | 51 | +359879632123 | Bulgaria |
| Kameliya Dimitrova | 42 | +359898645326 | Bulgaria |
| Dobroslav Mihalev | 39 | +359889632200 | Bulgaria |
| Mariya Petrova | 37 | +359887564235 | Bulgaria |
| … | … | … | … |

## Sites with Their Location and Category

Select all **sites** and find their **location** and **category.** Select the name of the **tourist site**, name of the **location**, **establishment** **year/ century** and name of the **category**. **Order** the result by name of the **category** (**descending**), then by nameof the **location (ascending)** and then by name of the **site** itself **(ascending).**

### Example:

|  |  |  |  |
| --- | --- | --- | --- |
| **Site** | **Location** | **Establishment** | **Category** |
| Clock Tower – Botevgrad | Botevgrad | 1866 | Spare time in the city |
| Clock Tower of Etropole | Etropole | 1710 | Spare time in the city |
| House of Humour and Satire Museum – Gabrovo | Gabrovo | 1972 | Spare time in the city |
| Museum of Education – Gabrovo | Gabrovo | 1974 | Spare time in the city |
| Antique Theater – Plovdiv | Plovdiv | II | Spare time in the city |
| Salt Museum – Pomorie | Pomorie | 2002 | Spare time in the city |
| … | … | … |  |

## Count of Sites in Sofia Province

Extract **all** **locations** which are in **Sofia province**. Find the **count** **of sites** in every **location**. Select the name of the **province**, name of the **municipality**, name of the **location** and **count** of the **tourist sites** in it. **Order** the result by **count of tourist sites** (**descending**) and then by **name of the** **location (ascending)**.

### Example:

|  |  |  |  |
| --- | --- | --- | --- |
| **Province** | **Municipality** | **Location** | **CountOfSites** |
| Sofia | Sofia | Sofia | 11 |
| Sofia | Etropole | Etropole | 3 |
| Sofia | Botevgrad | Botevgrad | 1 |
| Sofia | Koprivshtitsa | Koprivshtitsa | 1 |
| Sofia | Svoge | Osenovlag village | 1 |
| Sofia | Samokov | Samokov | 1 |

## Tourist Sites established BC

Extract information about the **tourist sites,** which have a **location** **name** that **does** **NOT** **start** with the letters **'B', 'M'** or **'D'** and which are **established Before Christ (BC)**. Select the **site** **name**, **location** **name**, **municipality**, **province** and **establishment**. **Order** the result by name of the **site (ascending).**

**NB. If the establishment century is Before Christ (BC), it will always be in the following format: 'RomanNumeral BC'.**

### Example:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Site** | **Location** | **Municipality** | **Province** | **Establishment** |
| Asen’s Fortress | Asenovgrad | Asenovgrad | Plovdiv | V BC |
| National archaeological reserve Kabile | Yambol | Yambol | Yambol | II BC |
| Perperikon – Medieval Archaeological Complex | Rhodope Mountain | NULL | NULL | V BC |
| Shumen Fortress Historical-Archaeological Preserve | Shumen | Shumen | Shumen | I BC |
| Starosel – Thracian Temple Complex | Starosel village | Hisarya | Plovdiv | V BC |
| Thracian Tomb of Kazanlak | Kazanlak | Karlovo | Plovdiv | IV BC |

## Tourists with their Bonus Prizes

Extract information about the **tourists**, along with their **bonus** **prizes**. If there is **no data** for the **bonus** **prize** put **'(no bonus prize)'**. Select tourist’s **name**, **age**, **phone** **number**, **nationality** and **bonus** **prize**. **Order** the result by the **name** of the **tourist (ascending)**.

**NB. There will never be a tourist with more than one prize.**

### Example:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Age** | **PhoneNumber** | **Nationality** | **Reward** |
| Alonzo Conti | 36 | +393336258996 | Italy | (no bonus prize) |
| Brus Brown | 42 | +447459881347 | UK | (no bonus prize) |
| Claudia Reuss | 54 | +4930774615846 | Germany | Sleeping bag |
| Cosimo Ajello | 51 | +393521112654 | Italy | (no bonus prize) |
| Cyrek Gryzbowski | 64 | +48503435735 | Poland | (no bonus prize) |
| Danny Kane | 39 | +32487454880 | Belgium | Water filter jug |
| … | … | … | … | … |

## Tourists visiting History and Archaeology sites

Extract all **tourists**, who have visited sites from **category 'History and archaeology'**.Selecttheir **last name, nationality, age** and **phone number. Order** the result by their **last name (ascending).**

**NB. The name of the tourists will always be in the following format: 'FirstName LastName'.**

### Example:

|  |  |  |  |
| --- | --- | --- | --- |
| **LastName** | **Nationality** | **Age** | **PhoneNumber** |
| Bauer | Germany | 49 | +496913265224 |
| Becker | Germany | 36 | +491711234567 |
| Bianchi | Italy | 51 | +393125965845 |
| Brown | UK | 42 | +447459881347 |
| Conti | Italy | 36 | +393336258996 |
| Dimitrova | Bulgaria | 42 | +359898645326 |
| … | … | … | … |

# Section 4. Programmability (20 pts)

## Tourists Count on a Tourist Site

Create a **user-defined function** named **udf\_GetTouristsCountOnATouristSite** (**@Site)** which receives a **tourist site** and returns the count of **tourists,** who have visited it.

### Examples:

|  |
| --- |
| **Query** |
| **SELECT dbo.udf\_GetTouristsCountOnATouristSite ('Regional History Museum – Vratsa')** |
| **Output** |
| **6** |
| **Query** |
| **SELECT dbo.udf\_GetTouristsCountOnATouristSite ('Samuil’s Fortress')** |
| **Output** |
| **8** |

|  |
| --- |
| **Query** |
| **SELECT dbo.udf\_GetTouristsCountOnATouristSite ('Gorge of Erma River')** |
| **Output** |
| **7** |

## Annual Reward Lottery

*A reward scheme has been developed to encourage collection of as many stamps as possible. Depending on the number of stamps collected, participants may receive bronze, silver or gold badges.*

Create a **stored procedure**, named **usp\_AnnualRewardLottery(@TouristName). Update** the **reward** of the given **tourist** according to the **count of the sites** he have **visited**:

* **>= 100** receives **'Gold badge'**
* **>= 50** receives **'Silver badge'**
* **>= 25** receives **'Bronze badge'**

Extract the name of the **tourist** and the **reward** he has.

### Example:

|  |
| --- |
| **Query** |
| **EXEC usp\_AnnualRewardLottery 'Gerhild Lutgard'** |

#### Result

|  |  |
| --- | --- |
| **Name** | **Reward** |
| Gerhild Lutgard | Gold badge |

### Example:

|  |
| --- |
| **Query** |
| **EXEC usp\_AnnualRewardLottery 'Teodor Petrov'** |

#### Result

|  |  |
| --- | --- |
| **Name** | **Reward** |
| Teodor Petrov | Silver badge |

### Example:

|  |
| --- |
| **Query** |
| **EXEC usp\_AnnualRewardLottery 'Zac Walsh'** |

#### Result

|  |  |
| --- | --- |
| **Name** | **Reward** |
| Zac Walsh | Bronze badge |

### Example:

|  |
| --- |
| **Query** |
| **EXEC usp\_AnnualRewardLottery 'Brus Brown'** |

#### Result

|  |  |
| --- | --- |
| **Name** | **Reward** |
| Brus Brown | NULL |