# Database Basics MS SQL Exam – 8 April 2021

Exam problems for the [“Database Basics MSSQL Server” course @ SoftUni](https://softuni.bg/opencourses/ms-sql)" Submit your solutions in the SoftUni judge system at Software University.

# Service

The city mayor, came up with the idea to create an online platform where all the citizens can **report about different problems** and a special organization will work to resolve all the incoming reports. This organization has a few **departments each of which is responsible for a set of problem's categories** in which **users can submit a report**. In each department there are employees who get assigned to a report. Of course, this huge platform needs a reliable database to store and process the information and the mayor has asked for the best specialist in this area. That’s why you got chosen! Congratulations and good luck!

# Section 1. DDL (30 pts)

You have been given the E/R Diagram of the Report Service:



Create a database called Service. You need to create **6 tables**:

* Users - contains information about the people who submist reports
* Reports **- contains information about the problems**
* Employees - contains information about the employees
* Departments - contains information about the departments
* Categories - contains information about categories of the reports
* Status- contains information about the possible status

Users

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| Id | Integer from **0** **to 2,147,483,647** | **Unique** table identificator, **Identity.** |
| Username | String up to **30 symbols** | **Unique** for each user, NULLis **NOT** permitted. |
| Password | String up to **50 symbols** | NULLis **NOT** permitted. |
| Name | String up to **50 symbols** | NULLispermitted. |
| Birthdate | Date **with time** | NULLispermitted. |
| Age | Integer from **0 to 2,147,483,647** | In range between **14** and **110 (inclusive).** |
| Email | String up to **50 symbols** | NULLis **NOT** permitted. |

Departments

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| Id | Integer from **0 to 2,147,483,647** | **Unique** table identificator, **Identity.** |
| Name | String up to 50 symbols | NULLis **NOT** permitted. |

Employees

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| Id | Integer from **0 to 2,147,483,647** | **Unique** table identificator, **Identity.** |
| FirstName | String up to **25 symbols** | NULLispermitted. |
| LastName | String up to **25 symbols** | NULLispermitted. |
| Birthdate | Date **with time** | NULLispermitted. |
| Age | Integer from **0 to 2,147,483,647** | In range between **18** and **110 (inclusive).** |
| DepartmentId | Integer from **0 to 2,147,483,647** | Relationship with table departments. |

Categories

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| Id | Integer from **0 to 2,147,483,647** | **Unique** table identificator, **Identity.** |
| Name | String up to **50 symbols** | NULLis **NOT** permitted. |
| DepartmentId | Integer from **0 to 2,147,483,647** | Relationship with table departments. NULLis **NOT** permitted . |

Status

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| Id | Integer from **0 to 2,147,483,647** | **Unique** table identificator, **Identity.** |
| Label | String up to **30 symbols** | NULLis **NOT** permitted. |

Reports

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| Id | Integer from **0 to 2,147,483,647** | **Unique** table identificator, **Identity.** |
| CategoryId | Integer from **0 to 2,147,483,647** | Relationship with table categories. NULLis **NOT** permitted. |
| StatusId | Integer from **0 to 2,147,483,647** | Relationship with table status. NULLis **NOT** permitted. |
| OpenDate | Date **with time** | NULLis **NOT** permitted. |
| CloseDate | Date **with time** | NULLis permitted. |
| Description | String up to **200 symbols** | NULLis **NOT** permitted. |
| UserId | Integer from **0 to 2,147,483,647** | Relationship with table users. NULLis **NOT** permitted. |
| EmployeeId | Integer from **0 to 2,147,483,647** | Relationship with table employees. |

## Table design

Submit all of your **create statements** to Judge.

# Section 2. DML (10 pts)

**Before you start you have to import "DataSet-Service.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 Id's should be auto-generated.

****Employees****

|  |  |  |  |
| --- | --- | --- | --- |
| **FirstName** | **LastName** | **Birthdate** | **DepartmentId** |
| Marlo | O'Malley | 1958-9-21 | 1 |
| Niki | Stanaghan | 1969-11-26 | 4 |
| Ayrton | Senna | 1960-03-21 | 9 |
| Ronnie | Peterson | 1944-02-14 | 9 |
| Giovanna | Amati | 1959-07-20 | 5 |

****Reports****

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **CategoryId** | **StatusId** | **OpenDate** | **CloseDate** | **Description** | **UserId** | **EmployeeId** |
| 1 | 1 | 2017-04-13 |  | Stuck Road on Str.133 | 6 | 2 |
| 6 | 3 | 2015-09-05 | 2015-12-06 | Charity trail running | 3 | 5 |
| 14 | 2 | 2015-09-07 |  | Falling bricks on Str.58 | 5 | 2 |
| 4 | 3 | 2017-07-03 | 2017-07-06 | Cut off streetlight on Str.11 | 1 | 1 |

## Update

Update the CloseDate with the current date of all reports, which don't have CloseDate.

## Delete

Delete **all reports** who have a **Status** **4**.

# Section 3. Querying (40 pts)

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

## Unassigned Reports

Find all **reports** that **don't** have an **assigned employee**. **Order** the results by OpenDate in **ascending** order, then by description **ascending**. **OpenDate** must be in format - **'dd-MM-yyyy'**

### Example:

|  |  |
| --- | --- |
| **Description** | **OpenDate** |
| Art exhibition on July 24 | 17-12-2014 |
| Stuck Road on Str.133 | 20-06-2015 |
| Burned facade on Str.560 | 26-08-2015 |

## Reports & Categories

Select all **descriptions** from reports, which have **category**. Order them by description (**ascending**) then by category name (**ascending**).

### Example:

|  |  |
| --- | --- |
| **Description** | **CategoryName** |
| 162 kg plastic for recycling. | Green Areas |
| 246 kg plastic for recycling. | Snow Removal |
| 366 kg plastic for recycling. | Recycling |

## Most Reported Category

Select the **top 5 most reported categories** and **order** them **by** the number of **reports** **per category** in **descending** order and then **alphabetically** by name.

### Example:

|  |  |
| --- | --- |
| **CategoryName** | **ReportsNumber** |
| Recycling | 8 |
| Snow Removal | 5 |
| Streetlight | 4 |

## Birthday Report

Select the user's **username** and **category name** in all **reports** in which **users** have submitted a report **on their birthday**. **Order** them by **username** (**ascending**)and then by **category name** (**ascending**).

### Example:

|  |  |
| --- | --- |
| **Username** | **CategoryName** |
| 5omarkwelleyc | Snow Removal |
| dpennid | Dangerous Trees |
| llankham6 | Homeless Elders |

## Users per Employee

Select **all** **employees** and show how many **unique** users each of them has served to.

Order by **users count** (**descending**) and then by **full** name (**ascending**).

### Example:

|  |  |
| --- | --- |
| **FullName** | **UsersCount** |
| Bron Ledur | 3 |
| Adelind Benns | 2 |
| Dick Wentworth | 2 |
| … | … |

## Full Info

Select **all info** for reports along with employee **first name** and **last name** (**concataned with space**), **their** **department name**, **category name**, **report description**, **open date**, **status label** and **name of the user**. Order them by first name (**descending**), last name (**descending**), department (**ascending**), category (**ascending**), description (**ascending**), open date (**ascending**), status (**ascending**) and user (**ascending**).

Date should be in format - **dd.MM.yyyy**

If there are empty records, replace them with '**None**'.

### Example:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Employee** | **Department** | **Category** | **Description** | **OpenDate** | **Status** | **User** |
| Niki Stranaghan | Event Management | Sports Events | Sky Run competition on September 8 | 08.06.2015 | Completed | Emlynn Alliberton |
| Marlo O'Malley | Infrastructure | Streetlight | Fallen streetlight columns on Str.14 | 12.09.2017 | Blocked | Erhart Alpine |
| Leonardo Shopcott | Animals Care | Animal in Danger | Parked car on green area on the sidewalk of Str.74 | 10.11.2016 | In Progress | Jocko Greggor |
| … | …. | … | … | … | … | … |

# Section 4. Programmability (20 pts)

## Hours to Complete

Create a **user defined function** with the name **udf\_HoursToComplete**(**@StartDate** **DATETIME, @EndDate** **DATETIME**) that receives a start date and end date and mustreturns the total hours which has been taken for this task. If start date is null or end is null return 0.

### Example usage:

|  |
| --- |
| **Query** |
| SELECT dbo.udf\_HoursToComplete(OpenDate, CloseDate) AS TotalHours  FROM Reports |
| TotalHours |
| 0 |
| 288 |
| 0 |

## Assign Employee

Create a **stored procedure** with the name **usp\_AssignEmployeeToReport**(@E**mployeeId INT**, @**ReportI**d INT) that receives an **employee's Id** and a **report's Id** and assigns the employee to the report **only if** the **department of the employee** and the **department of the report's category** are the same. Otherwise throw an **exception** with message: "Employee doesn't belong to the appropriate department!".

### Example usage:

|  |
| --- |
| **Query** |
| EXEC usp\_AssignEmployeeToReport 30, 1 |
| **Response** |
| Employee doesn't belong to the appropriate department! |
| **Query** |
| EXEC usp\_AssignEmployeeToReport 17, 2 |
| **Response** |
| (1 row affected) |