Database Basics MS SQL Exam – 22 Oct 2017

Exam problems for the "Database Basics" course @ SoftUni.

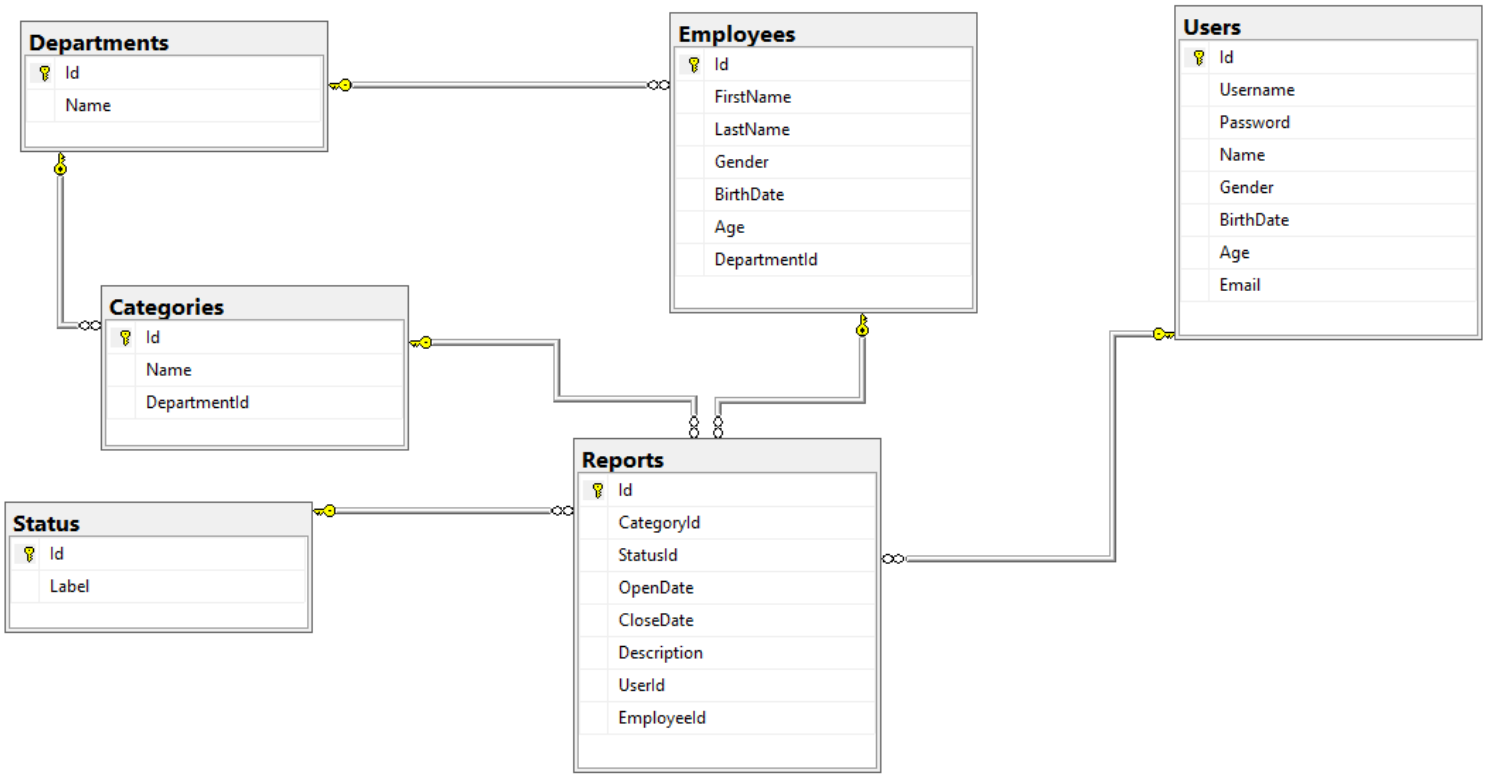
Submit your solutions in the SoftUni Judge system at https://judge.softuni.bg/Contests/819

Report Service

--Mrs. F.Y, the city's 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 Mrs. Y 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 **ReportService**. You need to create **6 tables**:

- **Users** – contains information about the people who submit reports
- **Reports** - contains information about the submitted problems
- **Employees** – contains information about the people employees who work on reports
- **Departments** – contains information about the departments
- **Categories** – contains information about categories inside the departments.
- **Status** - contains information about the possible statuses of a report

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, Unicode	**NULL** is **not** allowed, **Unique** values only
Password	**String** up to 50 symbols, Unicode	**NULL** is **not** allowed
Name	**String** up to **50** symbols, Unicode	
Gender	**Character** with **exactly 1** symbol	Could be: '***M***' or '***F***'
BirthDate	DateTime	
Age	**Integer** from **0** to **2,147,483,647**	
Email	**String** up to **50** symbols, Unicode	**NULL** is **not** allowed

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, Unicode	**NULL** is **not** allowed

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, Unicode	
LastName	**String** up to **25** symbols, Unicode	
Gender	**Character** with **exactly** 1 symbol	Could be: '***M***' or '***F***'
BirthDate	DateTime	
Age	**Integer** from **0** to **2,147,483,647**	
DepartmentId	**Integer** from **0** to **2,147,483,647**	**NULL** is **not** allowed, 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	**NULL** is **not** allowed
DepartmentId	**Integer** from **0** to **2,147,483,647**	Relationship with table Departments

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	**NULL** is **not** allowed

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**	**NULL** is **not** allowed, Relationship with table Categories
StatusId	**Integer** from **0** to **2,147,483,647**	**NULL** is **not** allowed, Relationship with table Status
OpenDate	DateTime	**NULL** is **not** allowed
CloseDate	DateTime	
Description	**String** up to **200** symbols	
UserId	**Integer** from **0** to **2,147,483,647**	**NULL** is **not** allowed, Relationship with table Users
EmployeeId	**Integer** from **0** to **2,147,483,647**	Relationship with table Employees

1. Database design

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

Section 2. DML (10 pts)

Before you start you have to import "DataSet-ReportService.sql". If you have created the structure correctly the data should be successfully inserted.

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

2. 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. Replace names that relate to other tables with the appropriate ID (look them up manually, there is no need to perform table joins).

Emlpoyees

FirstName	LastName	Gender	Birthdate	Department Id
Marlo	O'Malley	M	9/21/1958	Infrastructure
Niki	Stanaghan	F	11/26/1969	Emergency
Ayrton	Senna	M	03/21/1960	Event Management
Ronnie	Peterson	M	02/14/1944	Event Management
Giovanna	Amati	F	07/20/1959	Roads Maintenance

Reports

CategoryId	StatusId	OpenDate	CloseDate	Description	UserId	EmployeeId
Snow Removal	waiting	04/13/2017		Stuck Road on Str.133	6	2
Sports Events	completed	09/05/2015	12/06/2015	Charity trail running	3	5
Dangerous Building	in progress	09/07/2015		Falling bricks on Str.58	5	2
Streetlight	completed	07/03/2017	07/06/2017	Cut off streetlight on Str.11	1	1

3. Update

Switch **all report's status** to "**in progress**" where it is **currently** "**waiting**" for the "**Streetlight**" **category** (look up the category ID and status ID's manually, there is no need to use table joins).

4. Delete

Delete **all reports** who have a **status** "**blocked**".

Section 3. Querying (40 pts)

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

If not specified the ordering will be ascending.

5. Users by Age

Select all **Usernames** with their **age** ordered by **age** (**ascending**) then by **username** (**descending**).

Required columns:

- Username
- Age

Example:

Username	Age
5omarkwelleyc	19
bkaasg	21
dfinicj5	24
…	…

6. Unassigned Reports

Find all **reports** that **don't** have an **assigned employee**. **Order** the results by **open date** in **ascending** order, then by **description** (**ascending**).

Required columns:

- Description

- OpenDate

Example:

Description	OpenDate
Art exhibition on July 24	2014-12-17 00:00:00.000
Stuck Road on Str.133	2015-06-20 00:00:00.000
Burned facade on Str.560	2015-08-26 00:00:00.000
…	…

7. Employees & Reports

Select **only employees** who **have** an **assigned report** and show **all reports** of **each** found **employee**. Show the open date column in the format "**yyyy-MM-dd**". Order them by **employee id** (ascending) **then** by **open date** (ascending) and then by **report Id** (again ascending).

Required columns:

- FirstName
- LastName
- Description
- OpenDate

Example:

FirstName	LastName	Description	OpenDate
Marlo	O'Malley	Fallen streetlight columns on Str.14	2017-09-12
Gregory	Stithe	Stuck Road on Str.14	2017-04-13
Humphrey	Tamblyn	Burned facade on Str.793	2016-07-20
…	…	…	…

8. Most reported Category

Select **ALL categories** and **order** them **by** the number of **reports per category** in **descending** order and then **alphabetically** by name.

Required columns:

- CategoryName
- ReportsNumber

Example:

CategoryName	ReportsNumber
Recycling	8

Snow Removal	5
Streetlight	4
…	…

9. Employees in Category

Select **ALL categories** and the number of employees in each category and **order** them **alphabetically** by category name.

Required columns:

- CategoryName
- Employees Number

Example:

CategoryName	Employees Number
Animal in Danger	3
Art Events	5
Dangerous Building	1
…	…

10. Users per Employee

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

Required columns:

- Employee's name - Full name consisting of FirstName and LastName and a space between them
- User's number

Order by Users Number **descending** and then by Name **ascending**.

Example:

Name	Users Number
Bron Ledur	3
Adelind Benns	2
Dick Wentworth	2
…	…

11. Emergency Patrol

Select **all reports** which **satisfy all** the following criteria:

- are **not closed** yet (they don't have a CloseDate)
- the **description** is longer than **20 symbols** and the word "**str" is mentioned anywhere**
- are **assigned** to one of the **following departments**: "Infrastructure", "Emergency", "Roads Maintenance"

Order the results by **OpenDate** (ascending), **then** by **Reporter's Email** (ascending) and then by **Report Id** (ascending).

Required columns:

- OpenDate
- Description
- Reporter Email

Example:

OpenDate	Description	Reporter Email
2015-06-20 00:00:00.000	Stuck Road on Str.133	bkaasg@g.co
2015-08-26 00:00:00.000	Burned facade on Str.560	dpennid@arizona.edu
2015-11-17 00:00:00.000	Gigantic crater ?n Str.19	ealpine0@squarespace.com
…	…	…

12. Birthday Report

Select **all categories** in which users have **submitted** a **report** on **their birthday**. Order them by name **alphabetically.**

Required columns:

- Category Name

Example:

Category Name
Dangerous Trees
Homeless Elders
Snow Removal

13. Numbers Coincidence

Select all **unique usernames** which:

- **starts** with a **digit** and have reported in a **category** with **id equal** to the **digit**

OR

- **ends** with a **digit** and have reported in a **category** with **id equal** to the **digit**

Required columns:

- Username

Order them **alphabetically**.

Example:

Username

1qiskowf
5omarkwelleyc
fdenrico3
…

14. Open/Closed Statistics

Select **all employees** who have **at least one** assigned closed **or** open report **through** year **2016** and **their total sum**. Open reports don't have a **CloseDate**. Reports that have been **opened before** 2016 but were **closed in** 2016 are counted as **closed only**! Order by **Name** (ascending), and then by employee Id

Required columns:

- Name - name - Full name consisting of FirstName and LastName and a space between them
- Closed /Open reports number

Example:

Name	Closed Open Reports
Dick Wentworth	1/1
Eldon Gaze	0/1
Hewet Juschke	0/1
…	…

15. Average Closing Time

Select **all departments** that have been reported in and **the average time** for **closing** a **report** for each department rounded to the closest integer part**.** If there is **no information** (e.g. none closed reports) about any **department** fill in the Average Duration column "**no info**".

Required columns:

- Department Name
- Average Duration - in days

Order them by department name.

Example:

Department Name	Average Duration
Aged Care	no info
Animals Care	17
Emergency	no info
…	….

16. Favorite Categories

Select **all departments** with **their categories** where **users** have **submitted** a **report**. Show the **distribution** of reports **among** the **categories** of each department in **percentages** without decimal part.

Required columns:

- Department Name
- Category Name
- Percentage

Order them by **department** name, then by **category** name and **then** by **percentage** (all in **ascending** order).

Example:

Department Name	Category Name	Percentage
Aged Care	Homeless Elders	100
Animals Care	Animal in Danger	75
Animals Care	Street animal	25
…	…	…

Section 4. Programmability (14 pts)

17. Employee's Load

Create a **user defined function** with the name **udf_GetReportsCount(@employeeId, @statusId)** that receives an **employee's Id** and a **status Id** returns the sum of the reports he is assigned to with the given status.

Parameters:

- Employee's Id
- Status Id

Example usage:

Query

```
SELECT Id, FirstName, Lastname, dbo.udf_GetReportsCount(Id, 2) AS ReportsCount
FROM Employees
ORDER BY Id
```

Id	FirstName	LastName	ReportsCount
1	Marlo	O'Malley	0
2	Nolan	Meneyer	0
3	Tarah	McWaters	0
…	…	…	…

18. Assign Employee

Create a **user defined stored procedure** with the name **usp_AssignEmployeeToReport(@employeeId, @reportId)** 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. If the assigning is not successful **rollback** any changes and throw an **exception** with message: "`Employee doesn't belong to the appropriate department!`".

Parameters:

- Employee's Id
- Report's Id

Example usage:

Query
`EXEC usp_AssignEmployeeToReport 17, 2;` `SELECT EmployeeId FROM Reports WHERE id = 2`
Response
17

19. Close Reports

Create a **trigger** which changes the StatusId to "**completed**" of each report after a **CloseDate** is **entered** for the report.

Example usage:

Query

`UPDATE Reports`
`SET CloseDate = GETDATE()`
`WHERE EmployeeId = 5;`

Response
`(1 row affected)` `(1 row affected)`

Section 5. Bonus (10 pts)

20. Categories Revision

Select **all categories** which have **reports** with **status** "**waiting**" or "**in progress**" and show their **total number** in the column "**Reports Number**". In the **third column** fill the **main** status **type** of reports for the category (e.g. **2** reports with status "**waiting**" and **3** reports with status "**in progress**" **result in** value "**in progress**"). If they are equal just fill in "**equal**". Order by category **Name**, then by **Reports Number** and then by **Main Status.**

Required columns:

- Category Name
- Reports Number
- Main Status

Example:

Category Name	Reports Number	Main Status
Animal in Danger	1	in progress
Art Events	2	equal
Dangerous Building	1	waiting
…	…	…