

Database Basics MS SQL Exam – 22 Oct 2017

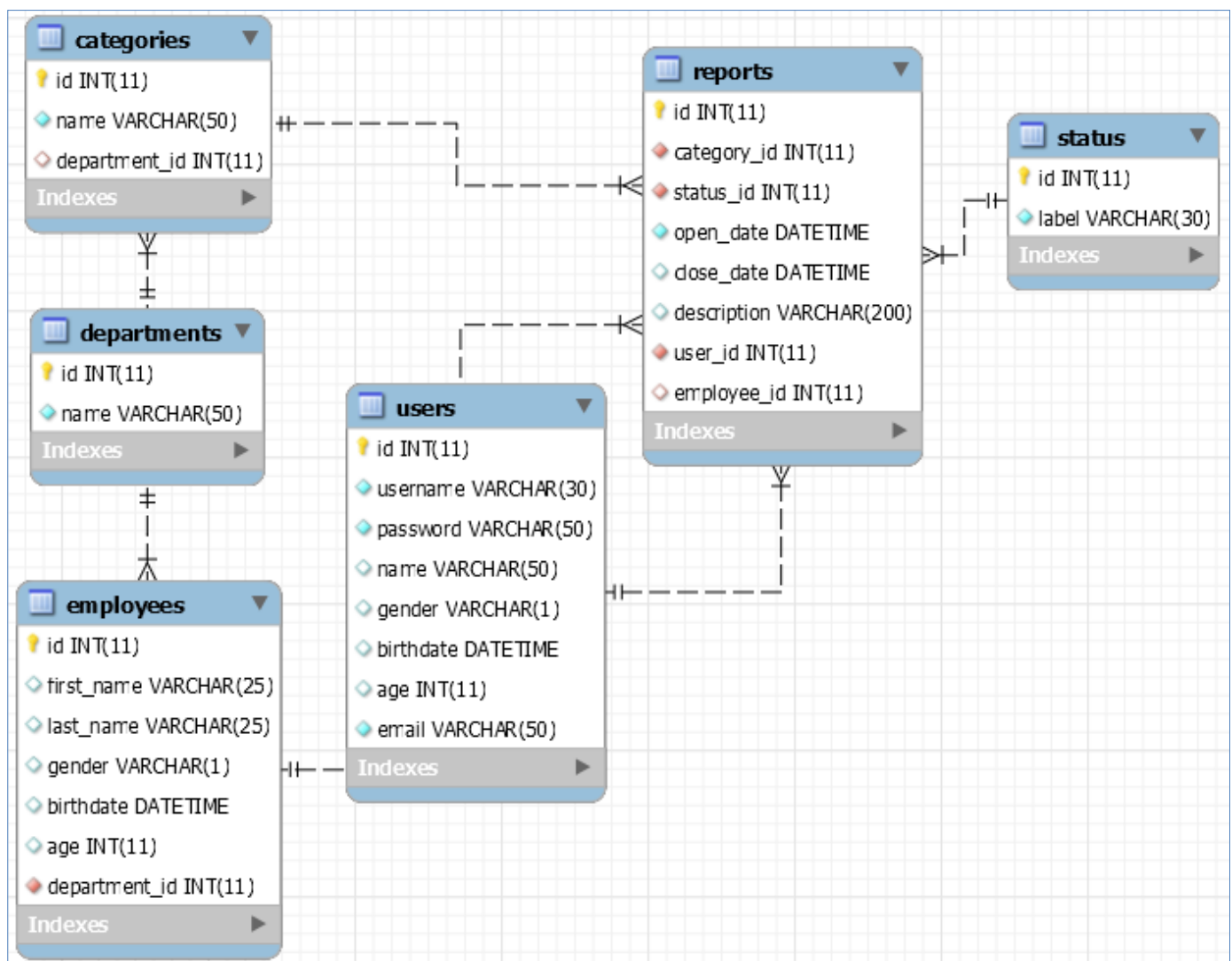
Exam problems for the [“Database Basics MySQL” course @ SoftUni](#). Submit your solutions in the SoftUni judge system at Software University.

Report Service

Mrs. Y. Fandukova, 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 Mrs. Fandukova 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 **report_service**. You need to create **6 tables**:

- **users** – contains information about the people who submit reports
- **reports** - contains information about the problems
- **employees** – contains information about the employees
- **departments** – mapping table between products and ingredients.
- **categories** – contains information about categories in reports.
- **status** – contains information about the possible

users

Column Name	Data Type	Constraints
id	Integer from 0 to 4,294,967,295	Unique table identifier
username	String up to 30 symbols	Unique for each user
password	String up to 50 symbols	NULL is NOT permitted
name	String up to 50 symbols	NULL is permitted
gender	String with exactly 1 symbol	
birthdate	Date with time	NULL is permitted
age	Integer from 0 to 4,294,967,295	NULL is permitted
email	String up to 50 symbols	NULL is NOT permitted

departments

Column Name	Data Type	Constraints
id	Integer from 0 to 4,294,967,295	Unique table identifier
name	String up to 50 symbols	NULL is NOT permitted

employees

Column Name	Data Type	Constraints
id	Integer from 0 to 4,294,967,295	Unique table identifier
first_name	String up to 25 symbols	NULL is permitted
last_name	String up to 25 symbols	NULL is permitted
gender	String with exactly 1 symbol	
birthdate	Date with time	NULL is permitted
age	Integer from 0 to 4,294,967,295	NULL is permitted
department_id	Integer from 0 to 4,294,967,295	Relationship with table departments .

categories

Column Name	Data Type	Constraints
id	Integer from 0 to 4,294,967,295	Unique table identifier
name	String up to 50 symbols	NULL is NOT permitted
department_id	Integer from 0 to 4,294,967,295	Relationship with table departments .

status

Column Name	Data Type	Constraints
id	Integer from 0 to 4,294,967,295	Unique table identificator
label	String up to 30 symbols	NULL is NOT permitted

reports

Column Name	Data Type	Constraints
id	Integer from 0 to 4,294,967,295	Unique table identificator
category_id	Integer from 0 to 4,294,967,295	Relationship with table categories .
status_id	Integer from 0 to 4,294,967,295	Relationship with table status .
open_date	Date with time	NULL is permitted
close_date	Date with time	NULL is permitted
description	String up to 200 symbols	NULL is permitted
user_id	Integer from 0 to 4,294,967,295	Relationship with table users .
employee_id	Integer from 0 to 4,294,967,295	Relationship with table employees .

1. Table design

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

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).

employees

first_name	last_name	gender	birthdate	department_id
Marlo	O'Malley	M	9/21/1958	1
Niki	Stanaghan	F	11/26/1969	4
Ayrton	Senna	M	03/21/1960	9
Ronnie	Peterson	M	02/14/1944	9
Giovanna	Amati	F	07/20/1959	5

reports

category_id	status_id	open_date	close_date	description	user_id	employee_id
1	1	04/13/2017		Stuck Road on Str.133	6	2
6	3	09/05/2015	12/06/2015	Charity trail running	3	5
14	2	09/07/2015		Falling bricks on Str.58	5	2
4	3	07/03/2017	07/06/2017	Cut off streetlight on Str.11	1	1

3. Update

Switch all report's status to **2** where it is currently **1** for the **4** category.

4. 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 (Data_ReportService.sql).

5. Users by Age

Select all **usernames** and **age** ordered by **age (ascending)** then by **username (descending)**.

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**.

Example:

description	open_date
Falling bricks on Str.13	2014-11-13 00:00:00:
Sky Run competition on September 21	2014-11-25 00:00:00
Art exhibition on July 24	2014-12-17 00:00:00

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 (again ascending)** and by **report_id ascending**.

Example:

first_name	last_name	description	open_date
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 **ascending** order and then **alphabetically** by name.

Example:

category_name	reports_number
Green Areas	1
Illegal Construction	1
Street animal	1

9. Employees in Category

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

Example:

category_name	employees_number
Animal in Danger	3
Art Events	5
Dangerous Building	1
...	...

10. Birthday Report

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

Duplicates are not needed.

Example:

category_name
Dangerous Trees
Homeless Elders
Snow Removal

11. 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 **first_name** and **last_name** and a space between them
- User's count

Order by **users_number descending** and then by **name ascending**.

Example:

name	users_count
Bron Ledur	3
Adelind Benns	2
Dick Wentworth	2
...	...

12. Emergency Patrol

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

- are **not closed** yet (they don't have a **close_date**)
- 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 **open_date** and then by **Reporter's Email and report_id ascending**.

Example:

open_date	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
...

13. Numbers Coincidence

Select all 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**

Order them **alphabetically**.

Example:

username
1qiskowf
5omarkwelleyc
fdenrico3
...

14. Open/Closed Statistics

Select **all employees** who have **at least one** assigned **closed** (have a **closed_date** value) / **open** report through year **2016** and **their number**. Reports that have been **opened before** 2016 but were **closed in** 2016 are counted as **closed only**! Order the results by **name** alphabetically.

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(in days)** for **closing a report** for each department. If there is **no information** (e.g. none closed reports) about any **department** fill in the Average Duration column **"no info"**. Round the average duration to the nearest smaller integer value.

Order them by department name.

Example:

department_name	average_duration
Aged Care	no info
Animals Care	17
Emergency	no info
...

16. Most Reported Category

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.

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 (20 pts)

For this section put your queries in judge and use: "SQL Server run skeleton, run queries and check DB".

17. Get Reports

Create a **user defined function** with the name `udf_get_reports_count(employee_id INT, status_id INT)` that receives an **employee's Id** and a **status Id** returns the sum of the reports he is assigned to with the given status.

Example usage:

Query			
<pre>SELECT id, first_name, last_name, udf_get_reports_count(id, 2) AS reports_count FROM employees AS e ORDER BY e.id;</pre>			
id	first_name	last_name	reports_count
1	Marlo	O'Malley	0
2	Nolan	Meneyer	0
3	Tarah	McWaters	0
4	Bernetta	Bigley	0
5	Gregory	Stithe	0
6	Bord	Hambleton	0
7	Humphrey	Tamblyn	0
8	Dinah	Zini	1

18. Assign Employee

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

Example usage:

Query
<pre>CALL usp_assign_employee_to_repor(30, 1); SELECT employee_id FROM reports WHERE id = 2</pre>
Response
Employee doesn't belong to the appropriate department!
Query
<pre>CALL usp_assign_employee_to_report(17, 2) SELECT employee_id FROM reports WHERE id = 2</pre>
Response
17

19. Close Reports

Create a **trigger** which changes the **status_id** to “**completed**” of each report after a **close_date** is **entered** for the report.

Example usage:

Query
<pre>UPDATE reports SET close_date = now() WHERE employee_id = 5;</pre>
Response
(1 row affected)
(1 row affected)

Section 5. Bonus (10 pts)

For this section put your queries in judge and use: “SQL Server prepare DB and run queries”.

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**”.

Example:

category_name	reports_number	main_status
Animal in Danger	1	in progress
Art Events	2	equal
Dangerous Building	1	waiting
...

