Mandatory Assignment A

You have to hand-in this assignment on Fronter before October 9th at 23:59.

There should be *one* PDF file containing your name, link to the video, and answers to all the questions. You have to explain what you have done, show what you have done (screenshot of SQL statements) and the outcome (screenshot of the server's response – for INSERT assignments you need to show the table both before and after the INSERT). This is an individual report, it is allowed to work in groups on the SQL assignments and data model, but the explanations and test data should be your own.

The two assignments marked with "extra" are not mandatory. However, a good solution on one of these might make up for a mistake you made in one of the previous assignments. Not doing these will not count as negative though.

1. Start of the blog

1.1 Data model

You have to build the data model for a blog system that has the following characteristic:

- Users can register
- Users can have different roles, eg. author, administrator, commentator
- Blog posts can be established with eg. headline, content, picture
- Blog posts can belong to a number of categories
- Blog posts can be marked with a number of tags
- Blog posts can have different states, eg. drafts, published, archived, hidden

The data model (the design) have to include:

- Entities with specification of attributes and keys
- Relations with specification of cardinality and participation

You should describe the thoughts behind your design choices, especially where there are many-to-many relations. Chen's notation and Crow's foot notation are accepted. SQL diagrams are not accepted for this assignment.

1.2 SQL

For the solution of the following assignments, it is mandatory to establish the tables in SQL Server Management Studio and put up some realistic data, so that it is possible to test whether the SQL expressions work as planned or not – and show the results in the report.

1. What SQL expression(s) can be used to establish a blog post, when you consider that there has to be a user for the post, it has to belong to one category and two tags, and it is supposed to be a draft?

   For this one, you have to include screenshots of the query and screenshots where you show how the blog post table was before and after the establishment.

2. What SQL expression(s) can be used to collect all the tags that a given blog post has attached to it?

3. What SQL expression(s) can be used to show the 10 latest published blog posts? *For this one, it is important to remember that the date columns are not established yet. So you have to figure out a way to find the 10 newest posts without using dates.*

4. What SQL expression(s) can be used to show the number of blog posts for the categories that have at least one blog post attached?

5. What SQL expression(s) can be used to do a text search in both a headline and the content of a blog post?

Screenshots of the queries and results must be included.

2. Expansion of blog

2.1 Expansion of the data model

By now, it should be possible to handle the following data in your blog system:

- Users can register with relevant metadata
- Users can have different roles, eg. author, admin, commentator
- Blog posts can be established with a headline, content and a picture
- A blog post can be categorized in multiple categories
- A blog post can be tagged with multiple tags
- A blog post can have a status, eg pending, published, archived, hidden

It should also be possible to define what a user has rights to do. We need to distinguish between “reading”, “establish”, “edit” and “delete”. These rights should be tied up to a role, so that it is possible to specify what that role has rights to do. Eg. an author should be able to read, establish and edit.

You should describe the thoughts behind the expansion of the data model and explain how a user, a right and a role are connected with an example.

2.2 Date

It should be possible to find out, at what time a blog post has been established and when it has been edited. If that is not possible in your database design, you should now edit it, so that blog posts can be stamped with a date.

Then you need to work out the SQL expression(s) that makes it possible to get blog posts from the latest 10 days that are allowed to be read by all.

The following data needs to be included in your dataset:

- Title
- Publishing date
- Date for the latest editing
- Author
- Tags
- Categories

For this SQL expression, it is useful to insert at least one blog post that has a date from more than 10 days ago. Show how this blog post is there if you include all dates, but not there if you only include the latest 10 days.

2.3 Recursive categories

Now, a blog post can belong to multiple categories. We now wish that categories could belong to each other.

1. What changes do this take to the ER diagram?
   Tip: *Self Join* could be worth looking into for this one
2. What SQL expression can be used to collect all categories on the first level?
3. What SQL expression can be used to collect all categories on first and second level?

Hint: the first level categories are the ones that does not have any parent-category. Remember that there might exist categories on third level as well, that should not be shown in this exercise.

3. Automation of SQL in blog

The purpose of the assignments is to train VIEW and Stored Procedures. Remember to execute the stored procedure/views before you move on to the next assignment.

3.1 What goes where?

You shall decide what type of statements so far in the previous assignments you will store as VIEW and as PROCEDURE. All your statements shall be either VIEW or PROCEDURE.

3.2 Create Table script

You shall now create a new database for the whole assignment. Create a stored procedure that makes it possible to create all tables in the database for the blogging system.
Hint: Use the CREATE TABLE command.

3.3 Create Foreign Keys

You shall now create a stored procedure that makes it possible to create all the foreign keys between all the tables created in assignment 3.2.

3.4 Create test data

You shall now create a stored procedure that populates the tables with relevant test data.

3.5 Create relevant views

You shall now create the relevant views you have specified in the first assignment on this page.

3.6 Create relevant stored procedures

You shall now create the relevant stored procedures you have specified in the first assignment on this page.

3.7 Create new views and stored procedures

You shall now create two views and two stored procedures that you may think will make the life easier for the developer to communicate with the database on a daily basis. As documentation, you shall write your assumptions and arguments for why these views and stored procedures are relevant.

Hint: a good documentation for this third assignment is to include the new database diagram as documentation for your tables and foreign keys.

4. Transactions and triggers

4.1 Create transactions

You shall now find one example from assignment 3, where it would make sense to use a transaction. Argument for your choice, and show how the

new stored procedure looks like after the change.

4.2 Create new transactions (extra)

You shall now create two new stored procedures that include a transaction. Think about what kinds of stored procedures would benefit most from the ACID principles.

4.3 Create new triggers (extra)

Create a new table called BlogLog. Now you need to make sure, that every time a user either inserts, deletes or edits a blogpost, some data should be inserted into the BlogLog table.
- The action that caused the log (either INSERT, DELETE or EDIT)
- The username that caused the log
- The time of the change
- The id of the blogpost
- The headline of the blogpost

Consider in what other cases a trigger might be used in this blog system.