Database Management Systems Laboratory Experiments Phase 2

Subject : Stored Procedures, Views, Triggers, and SQL queries
Due Date : 10.04.2022
Advisor : Res. Assist.Ahmet ALKILINÇ, Ali Baran TAŞDEMİR

Introduction:

In this assignment your task is to define queries and objects such as procedures, views, triggers and queries which will be necessary for your project. In general, SQL (***S**tructured **Q**uery **L**anguage*) statements are divided into two major categories:

Data Definition Language (DDL): It is used for modification of the structure tables and objects in your database. [CREATE, ALTER, DROP] are few examples of DDL.

Data Manipulation Language (DML): Rather than structure, it only considers data in your database [INSERT, UPDATE, SELECT, DELETE] are a member of DML. In spite of lots of DDL phrases that exist, there are few DML phrases which are given above.

Using SQL queries summarized above you can create various objects stored in your database as well. Well-known and frequently-used objects are described below:

1. **Stored Procedure:** Stored procedures are programs that run on the database server and can be called with a single SQL statement. They are useful in situations where the processing should be done on the server side rather than the client side. Also, since the procedures are centralized to the server, code writing and maintenance is simplified, because the client programs do not have to duplicate the application logic. Stored procedures can also be used to reduce network communication; the results of a stored procedure can be analyzed and kept on the database server.[1]

   An SQL query that creates stored procedure comprising of main block, variables and conditional statements as well is given below:

   ```
   CREATE PROCEDURE [Procedure Name]
   (
       Variables with their types are defined here (if any)
   )
   WITH{RECOMPILE | ENCRYPTION}
   AS
   BEGIN
       Using BEGIN - END is optional.
   END
   ```

2. **View:** A view is a black-box database object that contains user defined SQL statements inside of it. Views are not stored in the database, but can be queried as if they existed.

[1] Ramakrishnan, Raghu, and Johannes Gehrke. Database management systems. Osborne/McGraw-Hill, 2000.

The query processor will replace the view by its definition in order to execute the query. Any relation that is not part of the logical model, but is made visible to a user as a virtual relation, is called a *view*. It is possible to support a large number of views on top of any given set of actual relations.[2]

SQL statement that creates a view is given below:

CREATE VIEW $< view\_name >$ **AS** $< view\_definition >$;

3. **Trigger or *Event-Condition-Action Rules*:** Triggers are awakened when certain events, specified by the database programmer, occur. The sorts of events allowed are usually insert, delete, or update to a particular relation. After awakened a trigger, an array of events is put in process, or not, depending on the conditions defined in trigger[3]. A trigger gives user three different options: *event*, *condition*, and *action*.

   Below, an SQL statement that constitute a trigger structure is given:

```
CREATE [OR REPLACE] TRIGGER < trigger_name >
{BEFORE | AFTER | INSTEAD OF }
    –DML operations are stated here–
WHEN (condition)
DECLARE
     Declaration-statements
BEGIN
     Executable-statements
EXCEPTION
     Exception-handling-statements
END;
```

Requirements of the assignment:

The aim of this phase is to implement following actions:

- Learn how to write SQL code using an interface
- Creating objects
- Inserting data
- Understand the logic of using views and procedures

Considering definitions above, you should perform the tasks detailed below:

1. Obtain **SQL statements of your database** you created for drawing ER model in previous assignment.

[2] Silberschatz, Abraham, Henry F. Korth, and Shashank Sudarshan. Database system concepts. Vol. 4. New York: McGraw-Hill, 1997.

[3] Garcia-Molina, Hector. Database systems: the complete book. Pearson Education India, 2008.

2. Create **sequences and triggers** for automatically incrementing primary key fields in your tables.
3. Generate insert, update and delete **stored procedures** for all tables in your database.
4. **Insert** minimum 10 records to all tables via the **procedures**.
5. Determine probably frequently-used SQL statements in your project, and place them in **views** (at least 5 views) some of which should contain Aggregate Functions.

Note: The first three bullets should be saved as *create.sql*, 4th and 5th should be *insert.sql* and *views.sql*, respectively.

Reporting:

Your reports should cover these subjects:

- Functionality of each procedure and view.
- Explain revision of your tables (if exists)

Submit Format:

→ <student id.zip>
→ sql/create.sql, insert.sql, views.sql
→ report/*.pdf

Notes

- Do not miss the deadline. Late submissions will be accepted until 3 days after the deadline. And it will be graded with a penalty.
- Save all your work until the assignment is graded.
- The assignment must be original, **individual** work. **Duplicate or very similar assignments are both going to be considered as cheating.**
- You can ask your questions via e-mail (*ahmetalkilinc, abtasdemir*[at]*cs.hacettepe.edu.tr*) or Piazza.
- Report language must be in **English**.
- You will submit your work from `https://evdekal.hacettepe.edu.tr` with the file hierarchy as above.