SCS 2209 - DATABASE II

Instructions

- Download the ***.sql*** file in the UGVLE and import the file to ***phpmyadmin***.
- Use the ***salescompany*** database to write SQL queries for each question.
- Save your sql queries and result sets to a text document with the relevant question number. Make sure you have saved the text file using your index number.
- Then upload it to the UGVLE.

What is a Database Trigger?

A database trigger is special stored procedure that is run when specific actions occur within a database. Most triggers are defined to run when changes are made to a table's data. For example, a trigger can be invoked when a row is inserted into a specified table or when certain table columns are being updated.

Why triggers?

- Global enforcement of business rules. Define a trigger once and then reuse it for any application that uses the database.
- Validate input data.
- Generate a unique value for a newly-inserted row in a different file.
- Write to other files for audit trail purposes.
- Replicate data to different files to achieve data consistency.
- Faster application development. Because the database stores triggers, you do not have to code the trigger actions into each database application.

How to create MySQL triggers?

Syntax:

```
DELIMITER //

DROP TRIGGER IF EXISTS trigger_name //

CREATE TRIGGER trigger_name

trigger_time { BEFORE | AFTER }

trigger_event { INSERT | UPDATE | DELETE }

ON tbl_name

FOR EACH ROW

BEGIN

        trigger_body

END //

DELIMITER ;
```

Each trigger requires:

- CREATE TRIGGER [trigger_name]: Creates or replaces an existing trigger with the trigger_name. All triggers must have a unique name.
- [before | after]: This specifies when the trigger will be executed.
- {insert | update | delete}: This specifies the DML operation which can be an insert, update or delete.
    - The trigger activates whenever a new row is inserted into the table. (Ex : INSERT statement)
    - The trigger activates whenever a row is modified. (Ex : UPDATE statement)

    - The trigger activates whenever a row is deleted from the table. (Ex: DELETE and REPLACE statements.) DROP TABLE and TRUNCATE TABLE statements on the table do not activate this trigger, because they do not use DELETE.
- ON [table_name]: This specifies the name of the table associated with the trigger. You can only associate a trigger with a permanent table, not with a TEMPORARY table or a view.
- [for each row]: This specifies a row-level trigger, i.e., the trigger will be executed for each row being affected.
- [trigger_body]: This provides the operation to be performed as trigger is fired. To execute multiple statements, use the BEGIN ... END compound statement construct.

There are two MySQL extensions for triggers as '**OLD**' and '**NEW**' (not case sensitive) which allows you to access columns in the rows affected by the DML statement in the trigger.

```
CREATE TRIGGER ins_sum

BEFORE INSERT

ON account

FOR EACH ROW

DECLARE sum INT DETAULT 100;

SET sum = sum + NEW.amount;
```

- In an INSERT trigger, only NEW.col_name can be used.
- In an UPDATE trigger, you can use OLD.col_name to refer to the columns of a row before it is updated and NEW.col_name to refer to the columns of the row after it is updated.
- In a DELETE trigger, only OLD.col_name can be used.

How to create Triggers in PHPMyAdmin?

1. In phpMyAdmin, select the database that you want to work with.
2. Go to the SQL tab at the top of the page.

3. Enter your SQL trigger into the main dialog box on the form. In entering the syntax for the trigger, make sure you include the Delimiter and finally at the end change the delimiter to normal semicolons.
4. Finally hit "GO".

Example 01: MySQL Trigger AFTER INSERT

Question: Once a customer makes a new purchase order, basic details regarding the purchase order should be inserted in the *purchase_order* table and the items in that purchase order should be inserted to the *order_item* table. Consider the below invoice.

Purchase Order Number : 542420			Customer ID : 88	
Order Date : 13 January 2020			Customer Name: Ms. Paula Parente	
Product ID	Product Name	Unit Price	Quantity	Total
14	Tofu	23.25	6 Packs	139.50
3	Aniseed Syrup	10.00	12 Bottles	120.00
13	Konbu	6.00	6 Boxes	36.00
			Total	**295.50**

Create a trigger called *'orderItem_details'* to insert the products in above invoice to *order_item* table when the purchase order details are inserted to the *purchase_order* table.

Note: Go to the 'Structure' tab in the phpMyAdmin and change the structure of Primary Key (Id) as AUTO_INCREMENT in both *purchase_order* and *order_item* table and save the changes.

Further change the structure of *OrderDate* column in *purchase_order* table to save current system date & time. (Change '**Type'** as TIMESTAMP and '**Default'** as CURRENT_ TIMESTAMP. Then Save the changes)

Answer

```
//First create the trigger 'orderItem_details'

DELIMITER //

DROP TRIGGER IF EXISTS orderItem_details //

CREATE TRIGGER orderItem_details

AFTER INSERT ON purchase_order

FOR EACH ROW

BEGIN

    INSERT INTO orderitem (`OrderId`, `ProductId`, `UnitPrice`, `Quantity`)

    VALUES (NEW.id, 14, 23.25, 6);

    INSERT INTO orderitem (`OrderId`, `ProductId`, `UnitPrice`, `Quantity`)

    VALUES (NEW.id, 3, 10, 12);

    INSERT INTO orderitem (`OrderId`, `ProductId`, `UnitPrice`, `Quantity`)

   VALUES (NEW.id, 13, 6, 6);

END //

DELIMITER ;
```

// Then insert purchase order details to *purchase_order* table

```
INSERT INTO purchase_order (`OrderNumber`, `CustomerId`, `TotalAmount`) VALUES (542420, 88, 295.50),
```

// View the data in 'orderitem' table and check whether 'orderItem_details ' trigger has been executed and inserted product details for the purchase order with the OrderNumber 542420.

Example 02: MySQL Trigger BEFORE INSERT

Before inserting a new record to the *customer* table, a trigger called '*validate_customer'* check the column value of FirstName, LastName, and Country. If there are any space(s) before or after the FirstName , LastName and Country, trigger should use the TRIM() function to remove those space(s). The value of the Country should be converted to upper cases by UPPER() function.

First create the *validate_customer* trigger and then Insert the below row into *customer* table. Then check the FirstName, LastName and Country column in the *customer* table.

```
INSERT INTO customer VALUES (92,' Ana  ', '  King', 'Oulu' ,' Finland ','981-447755');
```

Answer

```
//Create the trigger 'validate_customer'

DELIMITER //

DROP TRIGGER IF EXISTS validate_customer //

CREATE TRIGGER validate_customer

BEFORE INSERT

ON customer

FOR EACH ROW

        BEGIN

                SET NEW. FirstName = TRIM (NEW. FirstName);

                SET NEW. LastName = TRIM (NEW. LastName);

                SET NEW. Country = TRIM (UPPER (NEW. Country));

          END //

DELIMITER ;
```

Example 03: MySQL Trigger AFTER UPDATE

Create a new table called '*AuditLog'* to record Audit Activities in '*salescompany'* database. Whenever a DML operation happens in the database, relevant audit record should be inserted to the *AuditLog table.* Consider only the *update* operation in *Customer* table and create the trigger '*AuditLog_customer'* that shows you the updated records in *Customer* table.

Answer

```
//First Create the table 'AuditLog'

CREATE TABLE IF NOT EXISTS AuditLog (

    Id INT AUTO_INCREMENT PRIMARY KEY,

    User_id VARCHAR (100) NOT NULL,

    Description VARCHAR (255) NOT NULL);

//Then Create the trigger 'AuditLog_customer'

DELIMITER //

DROP TRIGGER IF EXISTS AuditLog_customer //

CREATE TRIGGER AuditLog_customer

AFTER UPDATE

ON customer

FOR EACH ROW

    BEGIN

     DECLARE UserKey VARCHAR (100);

    SET UserKey= (SELECT USER());

    INSERT INTO AuditLog (User_id,Description)
```

```
VALUES (UserKey, CONCAT('Updated First name ',OLD.FirstName,' and Last name ', OLD.LastName,' as First Name ',NEW.FirstName,' and Last name ',NEW.LastName,'. On ',Now()));

END //

DELIMITER ;

// Then run below record and check 'AuditLog' table for new audit logs.

UPDATE customer SET FirstName='Reeta',LastName='Miller' WHERE id=86;
```

How to delete a MySQL trigger?

To delete or destroy a trigger, use a DROP TRIGGER statement. You must specify the schema name if the trigger is not in the default (current) schema.

```
DROP TRIGGER [IF EXISTS] [schema_name.]trigger_name;
```

If you drop a table, any triggers for the table are also dropped.

Questions

01. Company has decided to give a 10% discount for Total Payable amount in every customer purchase order. Create the table 'Discount' in the *salescompany* database. *Discount* table is recording Purchase Order Number, Total payable amount without discount and total payable amount with 10% discount. Create the trigger *'DiscountAmount'* on *purchase_order* table.Then insert a new record to the *purchase_order* table and view the data in *Discount* table.
02. Assume suddenly the table which displays the total amount of the orders was deleted from the database and you are assigned with the task of recreating the table for the calculation of the total order amount. You are not required to process anything regarding the previous orders. The only requirement is that you create a new table for displaying

total order amount for the order id, product id, unit price, quantity and total amount upon an insert of a new record.

03. Company has decided to keep track of customers who have not provided their phone numbers. Create a table named *Reminders* to store the reminders to insert the missing phone numbers with id INT AUTO_INCREMENT, memberId INT, message VARCHAR(255) NOT NULL, PRIMARY KEY (id , memberId). Then create an AFTER INSERT trigger as '***after_customer_insert'*** which inserts a reminder into the reminders table if the customer Phone number is NULL. Insert a new record without a phone number to the customer table and view the data in reminders table.