//31139

प्र.क्र.
Q. No.

Assignment No. 8

Title : Database Trigger

PROBLEM STATEMENT :
Write database triggers to keep track records on library table. Include 6 triggers, namely; before insert, after insert, before update, after update, before delete, after delete

OBJECTIVE :
1. Understand the concept of database triggers.
2. Understand mySQL commands.

OUTCOMES : I will be able to
1. Create database triggers to enhance functionality
2. Understand and implement required MySQL commands.

S/W & H/W Requirements :
MySQL, PC with configuration as latest version of 64 bit OS, opensource Fedora, 8 GB RAM, 500 GB HDD.

THEORY :

→ Trigger :
A trigger is a stored procedure in database which automatically invokes whenever a special event in the database occurs. For example, a trigger can be invoked when a row is inserted into a specified table or when certain table colums are being updated.

F. No.

→ Differences between procedures and Triggers:
A procedure is executed explicitly from another block via a procedure call with passing arguments, while a trigger is executed (or fired) implicitly, whenever the triggering event (DML: INSERT, UPDATE OR DELETE) happens, and a trigger doesn't accept arguments.

→ When are triggers used?
- Maintaining complex integrity constraints (referential integrity) or business rules.
- Auditing information in a table by recording the changes.
- Automatically signalling ~~of~~ other programs that action needs to take place when changes are made to a table.
- Collecting/maintaining statistical data

→ Types of Triggers:
SQL Server provides 3 types of triggers

1. Data manipulation language (DML) triggers:
They are invoked automatically in response to INSERT, UPDATE, DELETE events against tables.

2. Data Definition language (DDL) triggers:
They fire in response to CREATE, ALTER, DROP statements. DDL triggers also fire in response to some system stored procedure that perform DDL-like operations.

Q. No. | Q. No.

3. Logon triggers: They fire in response to LOGON events.

→ BEFORE & AFTER Triggers:

Since triggers occur because of events, they may be set to occur immediately before or after these events. The events that execute triggers are database transactions, triggers can be executed immediately BEFORE or AFTER the DML statements.

Syntax:

```
Create trigger [trigger_name]
[before|after]
{insert | update | delete}
on [table_name]
[for each row]
[trigger_body]
```

Explanation of syntax:

1. Create trigger [trigger_name]:
→ Creates or replaces an existing trigger with the name.

2. [before | after]
→ This specifies when the trigger will be executed

3. {insert | update | delete}
→ This specifies the DML operation.

4. on [table_name]:
This specifies the name of table associated with trigger.

5. [For each row]:
This specifies a row-level trigger, i.e. the trigger will be executed for each row being affected.

6. [trigger_body]
This provides the operation to be performed as trigger is fired.

CONCLUSION: Successfully understood & implemented database triggers.