**AY: 2024-25**

| **Class:** | **SE** | **Semester:** | **IV** |
| --- | --- | --- | --- |
| **Course Code:** | **CSL402** | **Course Name:** | **Database Management System Lab** |

| **Name of Student:** | **Shruti Gauchandra** |
| --- | --- |
| **Roll No. :** | **16** |
| **Experiment No.:** | **8** |
| **Title of the Experiment:** | **Perform DCL and TCL Commands.** |
| **Date of Performance:** | **20/02/25** |
| **Date of Submission:** | **06/03/25** |

**Evaluation**

| **Performance Indicator** | **Max. Marks** | **Marks Obtained** |
| --- | --- | --- |
| Performance | 5 |  |
| Understanding | 5 |  |
| Journal work and timely submission | 10 |  |
| Total | 20 |  |

| **Performance Indicator** | **Exceed Expectations (EE)** | **Meet Expectations (ME)** | **Meet Expect Below Expectations (BE)** |
| --- | --- | --- | --- |
| Performance | 4-5 | 2-3 | 1 |
| Understanding | 4-5 | 2-3 | 1 |
| Journal work and timely submission | 8-10 | 5-8 | 1-4 |

**Checked by**

**Name of Faculty : Ms. Neha Raut**

**Signature :**

**Date:**

**Experiment No 8**

**Aim :-** **Write a SQL query to implement views and triggers**

**Objective :-** To learn about virtual tables in the database and also PLSQL constructs

**Theory:**

**SQL Views:**

In SQL, a view is a virtual table based on the result-set of an SQL statement.

A view contains rows and columns, just like a real table. The fields in a view are fields from one or more real tables in the database.

You can add SQL statements and functions to a view and present the data as if the data were coming from one single table.

A view is created with the CREATE VIEW statement.

CREATE VIEW Syntax

CREATE VIEW view\_name AS

SELECT column1, column2, ...

FROM table\_name

WHERE condition;

SQL Updating a View

A view can be updated with the CREATE OR REPLACE VIEW statement.

SQL CREATE OR REPLACE VIEW Syntax

CREATE OR REPLACE VIEW view\_name AS

SELECT column1, column2, ...

FROM table\_name

WHERE condition;

SQL Dropping a View

A view is deleted with the DROP VIEW statement.

SQL DROP VIEW Syntax

DROP VIEW view\_name;

Trigger: A trigger is a stored procedure in the 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 columns are being updated.

Syntax:

create trigger [trigger\_name]

[before | after]

{insert | update | delete}

on [table\_name]

[for each row]

[trigger\_body]

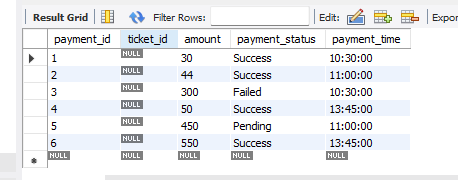
Explanation of syntax:

1. create trigger [trigger\_name]: Creates or replaces an existing trigger with the trigger\_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 the table associated with the 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

Implementation:

For VIEWS:

Code:



**CREATE VIEW View\_SuccessPayments AS**

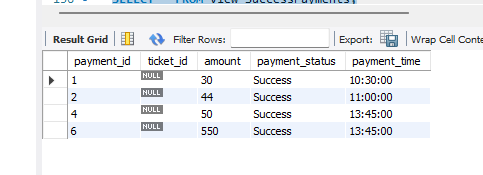
**SELECT \***

**FROM Payment**

**WHERE payment\_status = 'Success';**

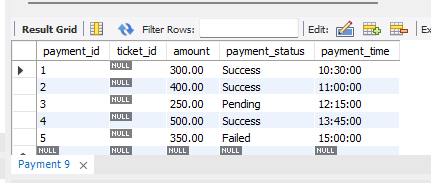
**SELECT \* FROM View\_SuccessPayments;**

Output:



**For TRIGGER:**

**Code:**

****

**CREATE TRIGGER apply\_discount**

**BEFORE UPDATE ON Payment**

**FOR EACH ROW**

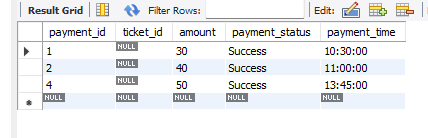
**BEGIN**

**SET NEW.amount = NEW.amount \* 0.10;**

**END apply\_discount;**

**SELECT \* FROM Payment;**

**Output:**

****

Conclusion:

A) Brief about the benefits for using views and triggers.

B) Explain different strategies to update views