**Batch: B-1 Roll No.: 16010122104**

**Experiment / assignment / tutorial No. 5**

**Grade: AA / AB / BB / BC / CC / CD /DD**

**Signature of the Staff In-charge with date**

|  |
| --- |
| **Title: Queries based Views and Triggers** |



**Objective:** To be able to use SQL view and triggers.



**Expected Outcome of Experiment:**

|  |  |
| --- | --- |
| CO 2 | Develop relational database design using the designed Entity-Relationship model. |
| CO 3 | Use SQL for Relational database creation, maintenance and query processing |

**Books/ Journals/ Websites referred:**

1. Dr. P.S. Deshpande, SQL and PL/SQL for Oracle 10g.Black book, Dreamtech Press

2. www.db-book.com

3. Korth, Slberchatz, Sudarshan : “Database Systems Concept”, 5th Edition , McGraw

Hill

4. Elmasri and Navathe,”Fundamentals of database Systems”, 4th Edition,PEARSON

Education.

**Resources used:** Postgresql

**Theory**

**Views** are pseudo-tables. That is, they are not real tables; nevertheless appear as ordinary tables to SELECT. A view can represent a subset of a real table, selecting certain columns or certain rows from an ordinary table. A view can even represent joined tables. Because views are assigned separate permissions, you can use them to restrict table access so that the users see only specific rows or columns of a table.

A view can contain all rows of a table or selected rows from one or more tables. A view can be created from one or many tables, which depends on the written PostgreSQL query to create a view.

Views, which are kind of virtual tables, allow users to do the following −

* Structure data in a way that users or classes of users find natural or intuitive.
* Restrict access to the data such that a user can only see limited data instead of complete table.
* Summarize data from various tables, which can be used to generate reports.

Since views are not ordinary tables, you may not be able to execute a DELETE, INSERT, or UPDATE statement on a view. However, you can create a RULE to correct this problem of using DELETE, INSERT or UPDATE on a view.

Syntax

CREATE [TEMP | TEMPORARY] VIEW view\_name AS

SELECT column1, column2.....

FROM table\_name

WHERE [condition];

Ex:

CREATE VIEW COMPANY-VIEW AS

SELECT ID, NAME, AGE

FROM COMPANY;

select \* from Company-View

Insert into Company-View values (123,’alpha’, 10)

select \* from Company

Dropping Views

Syntax: DROP VIEW view\_name;

Triggers

The basic syntax of creating a trigger is as follows −

CREATE TRIGGER trigger\_name [BEFORE|AFTER|INSTEAD OF] event\_name

ON table\_name

[

-- Trigger logic goes here....

];

event\_name could be INSERT, DELETE, UPDATE, and TRUNCATE database operation on the mentioned table table\_name. You can optionally specify FOR EACH ROW after table name.

The following is the syntax of creating a trigger on an UPDATE operation on one or more specified columns of a table as follows −

CREATE TRIGGER trigger\_name [BEFORE|AFTER] UPDATE OF column\_name

ON table\_name

[

-- Trigger logic goes here....

];

Example :

creates a log table and a trigger that inserts a row in the log table after any **UPDATE** statement affects the **SALARY** column of the **EMPLOYEES** table, and then pdates **EMPLOYEES**.**SALARY** and shows the log table.

CREATE TABLE Emp\_log ( Emp\_id NUMBER, Log\_date DATE, New\_salary NUMBER, Action VARCHAR2(20));

- Create trigger that inserts row in log table after EMPLOYEES.SALARY is update

CREATE OR REPLACE TRIGGER log\_salary\_increase

AFTER UPDATE OF salary ON employees

FOR EACH ROW BEGIN INSERT INTO Emp\_log (Emp\_id, Log\_date, New\_salary, Action) VALUES (:NEW.employee\_id, SYSDATE, :NEW.salary, 'New Salary');

END;

Update EMPLOYEES.SALARY:

UPDATE employees SET salary = salary + 1000.0 WHERE Department\_id = 20;

Result:

2 rows updated. Show log table:

SELECT \* FROM Emp\_log;

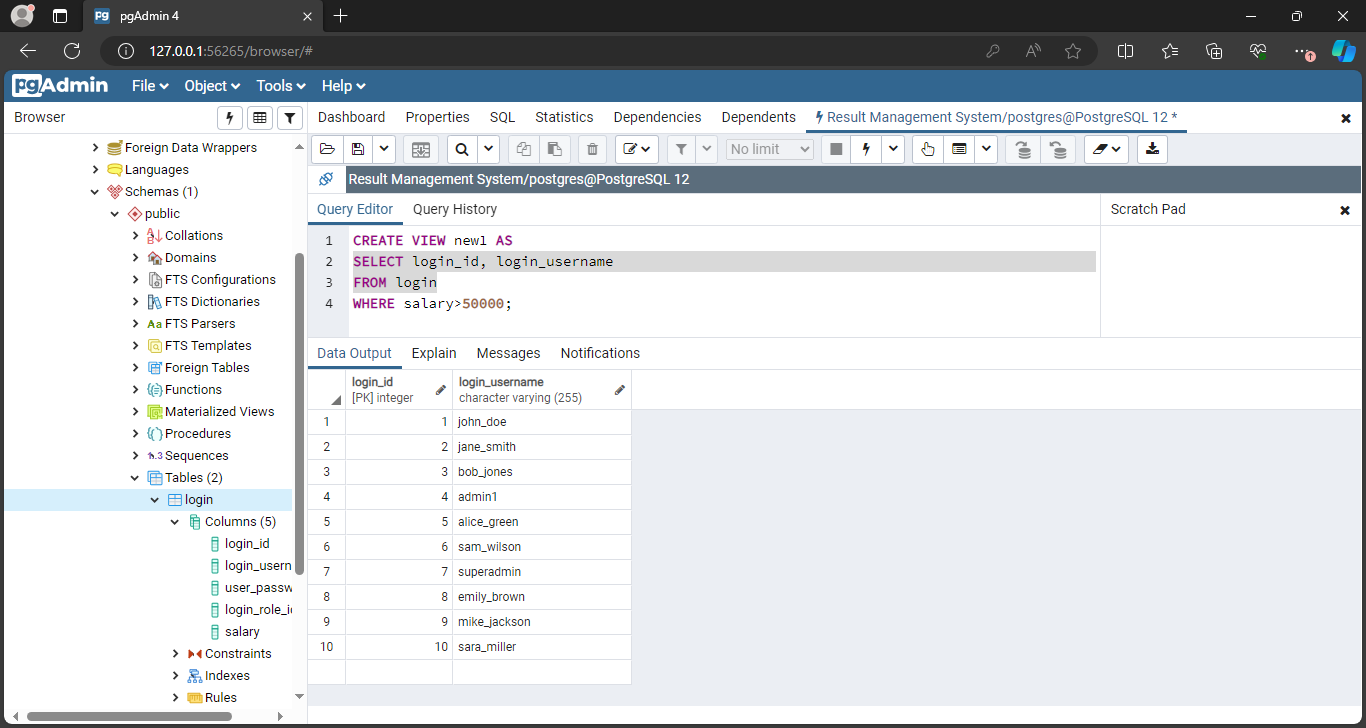
Result:

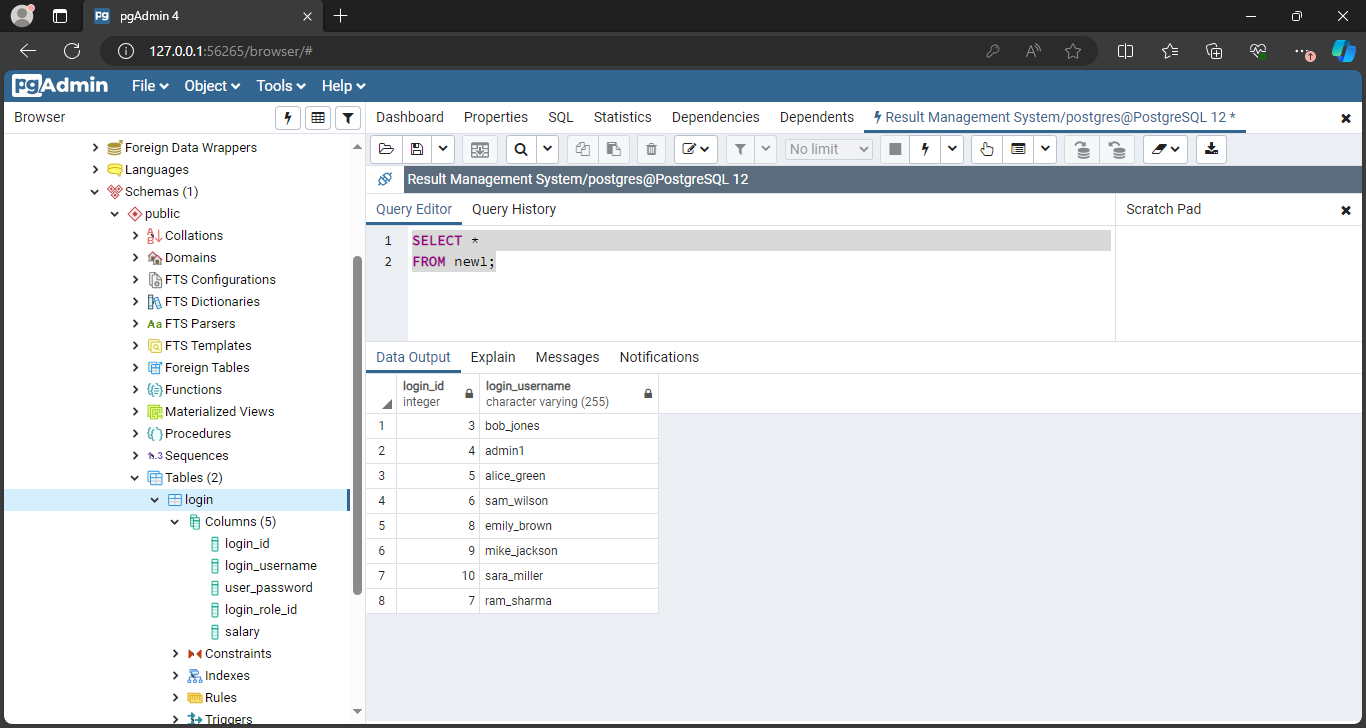
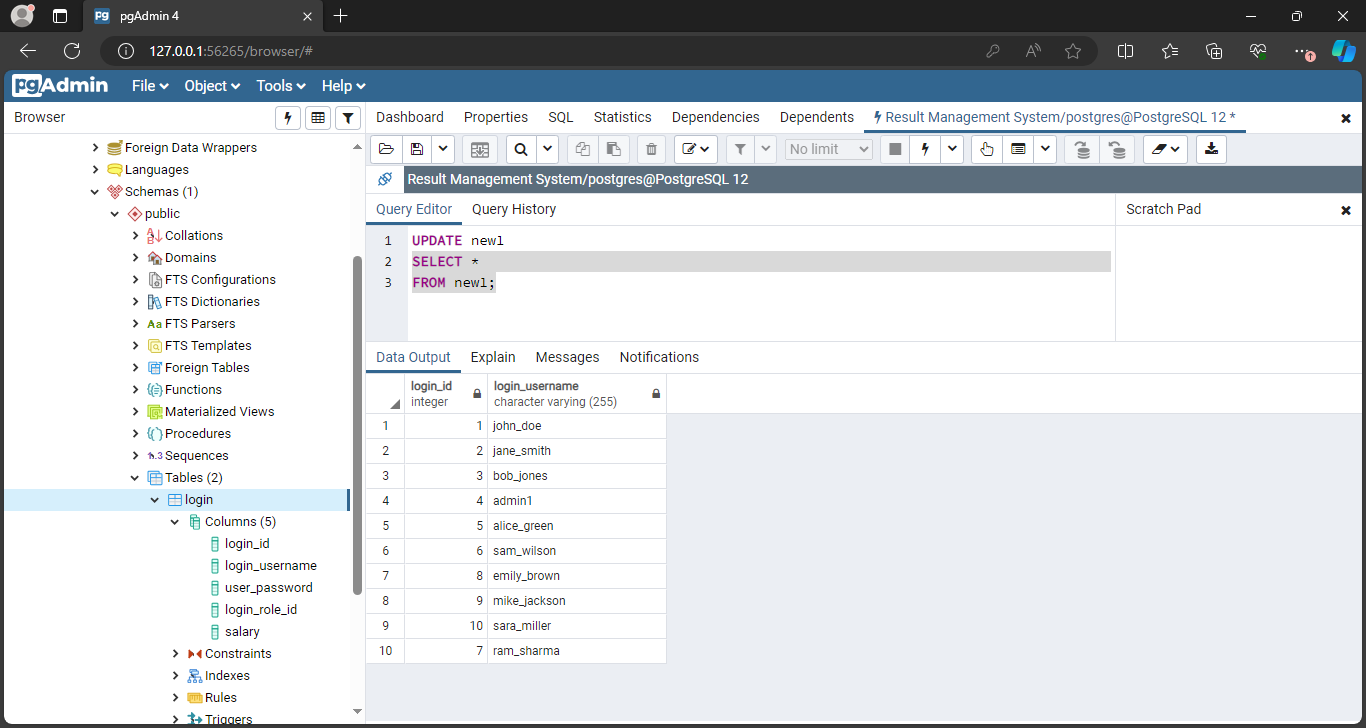
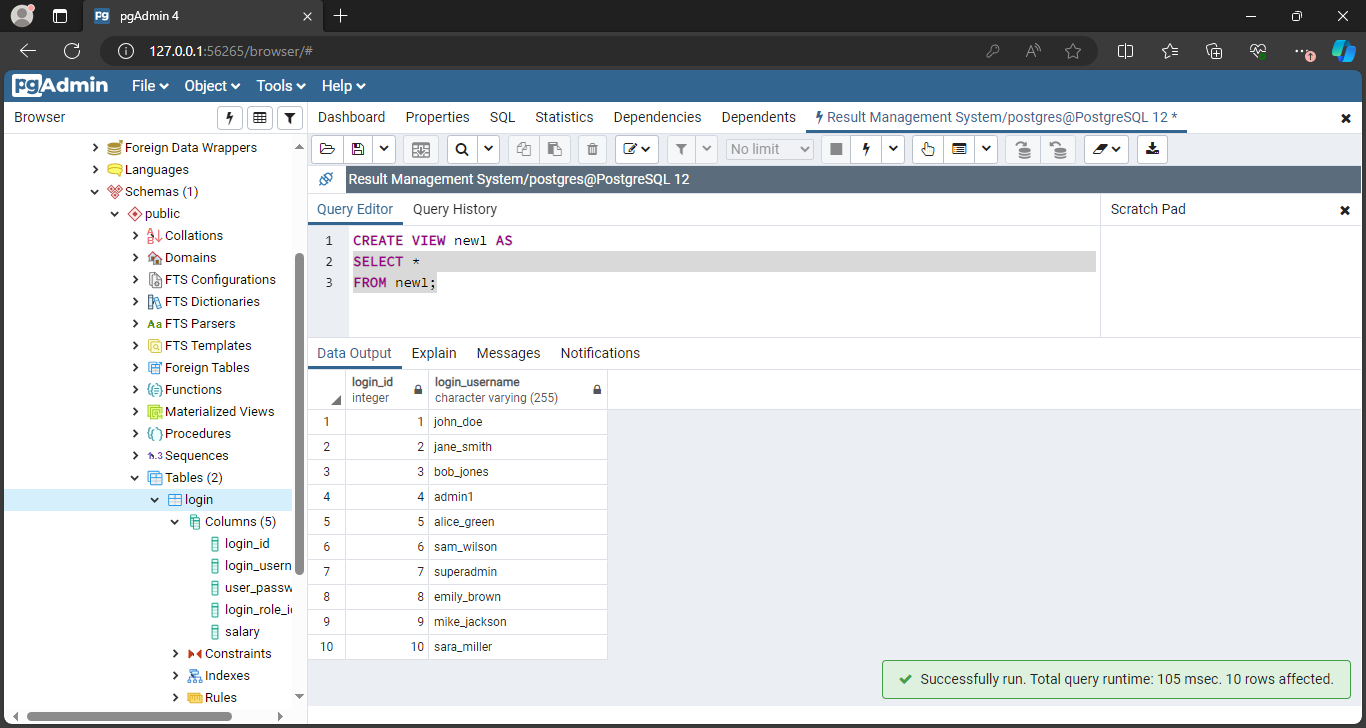
EMP\_ID LOG\_DATE NEW\_SALARY ACTION

201 28-APR-10 15049.13 New Salary

202 28-APR-10 6945.75 New Salary

2 rows selected.

**Implementation Screenshots (Problem Statement, Query and Screenshots of Results):**



# Conclusion:

# Post Lab Questions:

# What is a view?

# a) A view is a special stored procedure executed when certain event occurs

# b) A view is a virtual table which results of executing a pre-compiled query

# c) A view is a database diagram

# d) None of the Mentioned

# Ans:

**b) A view is a virtual table which results of executing a pre-compiled query.**

* Explanation: A view in a database is a virtual table that is based on the result set of a pre-compiled SQL query. It acts like a table, allowing users to query and manipulate the data stored in it, but the underlying data may come from one or more tables in the database.

# List Advantages and disadvantages of triggers

# Ans:

**Advantages:**

* **Data Integrity:** Triggers can enforce data integrity rules by automatically performing actions such as validating or modifying data before it is inserted, updated, or deleted.
* **Automated Tasks:** Triggers can automate repetitive tasks, such as logging changes, updating related tables, or enforcing business rules.
* **Complex Business Logic:** Triggers allow complex business logic to be encapsulated within the database, ensuring consistent application across all database operations.
* **Performance:** Triggers can improve performance by reducing the need for complex application logic and by executing tasks directly within the database engine.

**Disadvantages:**

* **Complexity:** Triggers can make the database schema more complex and harder to understand, especially if there are many triggers defined.
* **Hidden Logic:** Triggers may contain hidden logic that is not immediately visible in the application code, making it harder to debug and maintain.
* **Performance Overhead:** Triggers can introduce performance overhead, especially if they involve complex operations or are triggered frequently.
* **Dependency:** Triggers create dependencies between tables and objects in the database, which can make it harder to refactor the database schema or migrate to a different platform.