Experiment No 5:

Student Name: Bhavya Gupta
Branch: BE CSE (IS)
Semester: 5th
Subject:ADBMS

UID: 23BIS70147
Section/Group: 23AIT-KRG(2A)
Date of Performance:19-Sep-2025
Subject Code: 23-CSP-333

[Medium]

1. Create a Large Dataset
   - Create a table named transaction_data with the following columns:
     - id
     - value
   - Insert 1 million records into the table.
   - For id values 1 and 2, generate 1 million records each in the value column.
   - Use generate_series() and random() functions to populate the dataset.
2. Create Views for Sales Summary
   - Create a normal view named sales_summary that provides aggregated sales data.
   - Create a materialized view with the same purpose.
   - Both views should include the following aggregated information:
     - Total quantity sold
     - Total sales
     - Total orders
3. Performance and Execution Time Comparison

[Hard]

1. Create Restricted Views
   - Design views that display only summarized and non-sensitive data.
   - Ensure that confidential or sensitive information from base tables is not exposed.
2. Assign Access Using DCL Commands
   - Use GRANT to provide selected users (reporting team) with access to the restricted views.
   - Use REVOKE when access needs to be removed or modified.

SQL Quries

```
---------Medium----------

-- For id = 1
INSERT INTO transaction_data (id, value)
SELECT 1, random() * 1000  -- simulate transaction amounts 0-1000
FROM generate_series(1, 1000000);

-- For id = 2
INSERT INTO transaction_data (id, value)
SELECT 2, random() * 1000
FROM generate_series(1, 1000000);

SELECT *FROM transaction_data

--WITH NORMAL VIEW
CREATE OR REPLACE VIEW sales_summary_view AS
SELECT
    id,
    COUNT(*) AS total_orders,
    SUM(value) AS total_sales,
    AVG(value) AS avg_transaction
FROM transaction_data
GROUP BY id;


EXPLAIN ANALYZE
SELECT * FROM sales_summary_view;

--WITH MATERIALIZED VIEW
CREATE MATERIALIZED VIEW sales_summary_mv AS
SELECT
    id,
    COUNT(*) AS total_orders,
    SUM(value) AS total_sales,
    AVG(value) AS avg_transaction
FROM transaction_data
GROUP BY id;



EXPLAIN ANALYZE
SELECT * FROM sales_summary_mv;
```

```
---------hard----------
CREATE VIEW vW_ORDER_SUMMARY
AS
SELECT
    O.order_id,
    O.order_date,
    P.product_name,
    C.full_name,
    (P.unit_price * O.quantity) - ((P.unit_price * O.quantity) * O.discount_percent / 100) AS final_cost
FROM customer_master AS C
JOIN sales_orders AS O
    ON O.customer_id = C.customer_id
JOIN product_catalog AS P
    ON P.product_id = O.product_id;


SELECT * FROM vW_ORDER_SUMMARY;

CREATE ROLE Bkg
LOGIN
PASSWORD 'pass';

GRANT SELECT ON vW_ORDER_SUMMARY TO Bkg;
REVOKE SELECT ON vW_ORDER_SUMMARY FROM Bkg;
```

Output:

Output:

```
CREATE TABLE
INSERT 0 1000000
INSERT 0 1000000
CREATE VIEW
 id | total_orders | total_sales |   avg_transaction
----+--------------+-------------+----------------------
  1 |      1000000 |   500363852 | 500.3638520000000000
  2 |      1000000 |   500232543 | 500.2325430000000000
(2 rows)

SELECT 2
 id | total_orders | total_sales |   avg_transaction
----+--------------+-------------+----------------------
  1 |      1000000 |   500363852 | 500.3638520000000000
  2 |      1000000 |   500232543 | 500.2325430000000000
(2 rows)
```

Output:

```
CREATE TABLE
CREATE TABLE
CREATE TABLE
INSERT 0 10
INSERT 0 10
INSERT 0 20
CREATE VIEW
 order_id | order_date |    product_name     |  full_name  |         final_cost
----------+------------+---------------------+-------------+----------------------------
        1 | 2025-09-01 | Smartphone X100     | Amit Sharma |     47500.0000000000000000
        2 | 2025-09-02 | Laptop Pro 15       | Priya Verma |     58500.0000000000000000
        3 | 2025-09-03 | Wireless Earbuds    | Ravi Kumar  | 15000.000000000000000000000
        4 | 2025-09-04 | Smartwatch Fit      | Neha Singh  |     27600.0000000000000000
        5 | 2025-09-05 | Tablet 10.5         | Arjun Mehta |     41800.0000000000000000
        6 | 2025-09-06 | Gaming Console      | Amit Sharma |     39600.0000000000000000
        7 | 2025-09-07 | Bluetooth Speaker   | Priya Verma | 14000.000000000000000000000
        8 | 2025-09-08 | Digital Camera      | Ravi Kumar  |     49500.0000000000000000
        9 | 2025-09-09 | LED TV 55 inch      | Sneha Reddy |     51000.0000000000000000
       10 | 2025-09-10 | Power Bank 20000mAh | Vikram Das  | 10000.000000000000000000000
       11 | 2025-09-11 | Smartphone X100     | Rohit Gupta |     23750.0000000000000000
       12 | 2025-09-12 | Laptop Pro 15       | Pooja Nair  |        117000.000000000000
       13 | 2025-09-13 | Wireless Earbuds    | Ankit Yadav | 10000.000000000000000000000
       14 | 2025-09-14 | Smartwatch Fit      | Arjun Mehta |     27600.0000000000000000
       15 | 2025-09-15 | Tablet 10.5         | Sneha Reddy |     62700.0000000000000000
       16 | 2025-09-16 | Gaming Console      | Vikram Das  |     39600.0000000000000000
       17 | 2025-09-17 | Bluetooth Speaker   | Rohit Gupta | 14000.000000000000000000000
       18 | 2025-09-18 | Digital Camera      | Pooja Nair  |     49500.0000000000000000
       19 | 2025-09-19 | LED TV 55 inch      | Ankit Yadav |     51000.0000000000000000
       20 | 2025-09-20 | Power Bank 20000mAh | Neha Singh  | 12500.000000000000000000000
(20 rows)
```

Learning Outcome:

- Learned to create database views to present combined data from multiple tables.
- Practiced embedding calculations (like discounted totals) directly within a view.
- Understood how views simplify complex queries with JOINs across tables.
- Gained knowledge of using views as read-only abstractions for end-users.
- Learned to control access to sensitive data by granting/revoking view permissions.