EXPERIMENT - 5

Student Name: Tanay Manish Nesari	**UID: 23BCS13761**
Branch: BE-CSE	**Section/Group: KRG 1-B**
Semester: 5th	**Date of Performance: 16/09/2025**
Subject Name: ADBMS	**Subject Code: 23CSP-333**

Aim: To understand and implement views.

Objective:

1. Views: Performance Benchmarking : Normal View vs. Materialized View (Medium)

i. Create a large dataset:

Create a table names transaction_data (id , value) with 1 million records.
take id 1 and 2, and for each id, generate 1 million records in value column
Use Generate_series () and random() to populate the data._

ii. Create a normal view and materialized view to for sales_summary, which includes total_quantity_sold, total_sales, and total_orders with aggregation.

iii. Compare the performance and execution time of both.

2. Views: Securing Data Access with Views and Role-Based Permissions(Hard)
The company TechMart Solutions stores all sales transactions in a central database.
A new reporting team has been formed to analyze sales but they should not have direct access to the base tables for security reasons._The database administrator has decided to:

i. Create restricted views to display only summarized, non-sensitive data.

ii. Assign access to these views to specific users using DCL commands (GRANT, REVOKE).

Code:

1)

```
drop table if exists transaction_data;

create table transaction_data
(id int, val int);

insert into transaction_data
select 1, random()* 500000 + 100 from generate_series(1,1500000);

insert into transaction_data
select 2, random()* 500000 + 100 from generate_series(1,1500000);

select count(*)
from transaction_data
where id = 2 and val > 400000;

create or replace view v_sales_summary
as
select count(*) as total_orders,
sum(val) as total_sales,
round(avg(val),2) as average_price
from
transaction_data;

explain analyze
select * from v_sales_summary;

drop materialized view if exists mv_sales_summary;

create materialized view mv_sales_summary
as
select count(*) as total_orders,
sum(val) as total_sales,
round(avg(val),2) as average_price
from
transaction_data;

explain analyze
select * from mv_sales_summary;
```

2)

```
drop table if exists sales_orders;
drop table if exists product_catalog;
drop table if exists customer_master;
drop role if exists "AuditTeam";

CREATE TABLE customer_master (
    customer_id VARCHAR(5) PRIMARY KEY,
    full_name VARCHAR(50) NOT NULL,
    phone VARCHAR(15),
    email VARCHAR(50),
    city VARCHAR(30)
);

CREATE TABLE product_catalog (
    product_id VARCHAR(5) PRIMARY KEY,
    product_name VARCHAR(50) NOT NULL,
    brand VARCHAR(30),
    unit_price NUMERIC(10,2) NOT NULL
);

CREATE TABLE sales_orders (
    order_id SERIAL PRIMARY KEY,
    product_id VARCHAR(5) REFERENCES product_catalog(product_id),
    quantity INT NOT NULL,
    customer_id VARCHAR(5) REFERENCES customer_master(customer_id),
    discount_percent NUMERIC(5,2),
    order_date DATE NOT NULL
);

INSERT INTO customer_master (customer_id, full_name, phone, email, city) VALUES
('C11', 'Zoya Khan', '8765432109', 'zoya.k@newcorp.com', 'Pune'),
('C12', 'Arun Das', '8888888888', 'arun.d@newcorp.com', 'Chennai');
```

```sql
INSERT INTO product_catalog (product_id, product_name, brand, unit_price) VALUES
('P11', 'VR Headset', 'Oculus', 45000.00),
('P12', 'Monitor 4K', 'BenQ', 35000.00);

INSERT INTO sales_orders (product_id, quantity, customer_id, discount_percent, order_date) VALUES
('P11', 1, 'C11', 15.00, '2025-10-01'),
('P12', 2, 'C12', 5.00, '2025-10-02'),
('P11', 1, 'C12', 0.00, '2025-10-03');

create view v_sales_audit
as
        select
        o.order_id,
        o.order_date,
        p.product_name,
        p.brand,
        c.city,
        p.unit_price * o.quantity * (1 - o.discount_percent/100) as final_sale_value
        from
        customer_master as c
        join
        sales_orders as o
        on c.customer_id = o.customer_id
        join
        product_catalog as p
        on o.product_id = p.product_id;

create role "AuditTeam";

grant select on v_sales_audit to "AuditTeam";

revoke select on customer_master from "AuditTeam";
revoke select on sales_orders from "AuditTeam";
```