Experiment 5

Student Name: Arpeet

UID: 23BAI70396

Branch: BE-AIT-CSE

Section/Group: 23AIT_KRG-1

Semester: 5th

Date of Performance: 25 Sept, 2025

Subject Name: ADBMS

Subject Code: 23CSP-333

MEDIUM - LEVEL

1. **Problem Title:** Performance Benchmarking: Normal view vs Materialised view
2. **Problem Tasks and Description:**
   a) Create a large dataset:
   Create a table named 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.

   b) Create a normal view and materialized view to for sales_summary, which includes total_quantity_sold, total_sales, and total_orders.

   c) Compare the performance and execution time of both.

3. **SQL Commands:**

   a. Creating the table Employee and generating 1 million records for both ids:

```sql
CREATE TABLE TBL_transaction_data(
    id INT,
    value DECIMAL
);

INSERT INTO TBL_transaction_data(id,value)
SELECT 1, RANDOM()
FROM GENERATE_SERIES(1,1000000);

INSERT INTO TBL_transaction_data(id,value)
SELECT 2, RANDOM()
FROM GENERATE_SERIES(1,1000000);
```

b. Creating both the Normal view and the Materialised view:

```
CREATE VIEW VW_NormalView_salesSummary
AS
SELECT
    id,
    COUNT(*) AS Total_Orders,
    SUM(value) AS Total_Sales,
    AVG(value) AS Avg_transaction
FROM TBL_transaction_data
GROUP BY id;
```

```
CREATE MATERIALIZED VIEW VW_MaterialisedView_salesSummary AS
SELECT
    id,
    COUNT(*) AS Total_Orders,
    SUM(value) AS Total_Sales,
    AVG(value) AS Avg_transaction
FROM TBL_transaction_data
GROUP BY id;
```

c. Use the "Explain Analyze" query to compare both their performances:

```
EXPLAIN ANALYZE
SELECT * FROM VW_MaterialisedView_salesSummary

EXPLAIN ANALYZE
SELECT * FROM VW_NormalView_salesSummary
```

4. **Output:**

	QUERY PLAN text
1	Finalize GroupAggregate (cost=26399.39..26399.93 rows=2 width=76) (actual time=901.979..926.396 rows=2 loops=1)
2	Group Key: tbl_transaction_data.id
3	-> Gather Merge (cost=26399.39..26399.86 rows=4 width=76) (actual time=900.649..924.978 rows=6 loops=1)
4	Workers Planned: 2
5	Workers Launched: 2
6	-> Sort (cost=25399.37..25399.37 rows=2 width=76) (actual time=752.027..752.029 rows=2 loops=3)
7	Sort Key: tbl_transaction_data.id
8	Sort Method: quicksort Memory: 25kB
9	Worker 0: Sort Method: quicksort Memory: 25kB
10	Worker 1: Sort Method: quicksort Memory: 25kB
11	-> Partial HashAggregate (cost=25399.33..25399.36 rows=2 width=76) (actual time=750.982..751.294 rows=2 loops=3)
12	Group Key: tbl_transaction_data.id
13	Batches: 1 Memory Usage: 24kB
14	Worker 0: Batches: 1 Memory Usage: 24kB
15	Worker 1: Batches: 1 Memory Usage: 24kB
16	-> Parallel Seq Scan on tbl_transaction_data (cost=0.00..19149.33 rows=833333 width=15) (actual time=0.032..137.191 rows=66...
17	Planning Time: 2.855 ms
18	Execution Time: 928.475 ms

Output of Explain Analyze of Normal view(Execution time is 928ms)

	QUERY PLAN text
1	Seq Scan on vw_materialisedview_salessummary (cost=0.00..17.80 rows=780 width=76) (actual time=0.026..0.028 rows=2 loops=...
2	Planning Time: 0.104 ms
3	Execution Time: 0.045 ms

Output of Explain Analyze of Materialised view(Execution time is 0.045ms)

5. **Learning Outcome:**
    a. I learnt the practical uses of views
    b. I learnt about different types of views and their applications
    c. I learnt the advantage of materialized views for large amounts of data.

HARD - LEVEL

1. **Problem Title:** Securing Data Access with views and Role Based Permissions
2. **Problem Task and Description:**
   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 basetables for security reasons.

   The database administrator has decided to:
   - Create restricted views to display only summarized, non-sensitive data.
   - Assign access to these views to specific users using DCL commands (GRANT, REVOKE).

3. **SQL Commands:**
   a. Create the user.

```sql
CREATE USER CLIENT_1
WITH PASSWORD '123';
```

   b. Grant User certain permissions as required

```sql
GRANT SELECT ON VW_NormalView_salesSummary TO CLIENT_1;
GRANT SELECT ON VW_MaterialisedView_salesSummary TO CLIENT_1;
```

   c. Revoke any permissions if required:

```sql
REVOKE SELECT ON VW_NormalView_salesSummary FROM CLIENT_1;
```

4. **Learning Outcomes:**

   a. Learned about the use of DCL commands with views for security

   b. Learnt how to implement DCL commands in hand with views to ensure no data breach.