



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Experiment No: 5

Student Name: Sohit

Branch: B.E./C.S.E.

Semester: 5th

Subject Name: ADBMS

Subject Code: 23CSP-333

UID: 23BCS12492

Section/Group: KRG_3A

Question 1 :Medium Level Problem

Problem Title: Normal View vs. Materialized View

1. 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.

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

3. Compare the performance and execution time of both.

Solution:

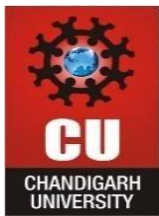
```
CREATE TABLE transaction_data (  
    id INT,  
    value INT  
);
```

```
-- 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)
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
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;
```



Question 2 :Hard Level Problem

Problem Title : Securing Data Access with Views and Role-Based Permissions

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:

1. Create restricted views to display only summarized, non-sensitive data.
2. Assign access to these views to specific users using DCL commands (GRANT, REVOKE).

Solution:

```
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;
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
CREATE ROLE CLIENT_USER
```

```
LOGIN
```

```
PASSWORD 'client_password';
```

```
GRANT SELECT ON vW_ORDER_SUMMARY TO CLIENT_USER;
```

```
REVOKE SELECT ON vW_ORDER_SUMMARY FROM CLIENT_USER;
```