NAME: CHETAN PRAKASH

UID: 23BCS13776

SECTION: 23_KRG_1(B)

DATE : 22-SEP-2025

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

ANSWER:

CREATE TABLE transaction_info (

user_id INT,

purchase_value DECIMAL

);

INSERT INTO transaction_info (user_id, purchase_value)

SELECT 1, random() * 1000

FROM generate_series(1, 1000000);

INSERT INTO transaction_info (user_id, purchase_value)

SELECT 2, random() * 1000

FROM generate_series(1, 1000000);

CREATE OR REPLACE VIEW purchase_summary_view AS

SELECT

user_id,

COUNT(*) AS order_count,

SUM(purchase_value) AS total_amount,

AVG(purchase_value) AS average_amount

FROM transaction_info

GROUP BY user_id;

CREATE MATERIALIZED VIEW purchase_summary_mv AS

SELECT

user_id,

COUNT(*) AS order_count,

```sql
  SUM(purchase_value) AS total_amount,
  AVG(purchase_value) AS average_amount
FROM transaction_info
GROUP BY user_id;


CREATE TABLE random_data (
group_id INT,
metric_value DECIMAL
);


INSERT INTO random_data
SELECT 1, random() FROM generate_series(1, 1000000);


INSERT INTO random_data
SELECT 2, random() FROM generate_series(1, 1000000);


SELECT
group_id,
AVG(metric_value),
COUNT(*)
FROM random_data
GROUP BY group_id;


CREATE MATERIALIZED VIEW mv_random_data AS
SELECT
group_id,
AVG(metric_value),
COUNT(*)
FROM random_data
GROUP BY group_id;


SELECT * FROM mv_random_data;
```

```
REFRESH MATERIALIZED VIEW mv_random_data;
```

QUESTION 2: Views: Securing Data Access with Views and Role-Based Permissions (hard)

ANSWER:

```
CREATE TABLE customer_data (
    customer_id SERIAL PRIMARY KEY,
    customer_name TEXT NOT NULL
);


CREATE TABLE product_data (
    item_id SERIAL PRIMARY KEY,
    item_name TEXT NOT NULL,
    price DECIMAL(10,2) NOT NULL
);


CREATE TABLE sales_data (
    order_id SERIAL PRIMARY KEY,
    customer_id INT REFERENCES customer_data(customer_id),
    item_id INT REFERENCES product_data(item_id),
    qty INT NOT NULL,
    discount DECIMAL(5,2),
    order_date DATE NOT NULL
);


INSERT INTO customer_data (customer_name) VALUES ('Rohit'), ('Meera'), ('Karan');

INSERT INTO product_data (item_name, price) VALUES ('Laptop', 55000), ('Phone', 25000), ('Tablet', 18000);


INSERT INTO sales_data (customer_id, item_id, qty, discount, order_date) VALUES
(1, 1, 2, 10, '2025-09-01'),
(2, 2, 1, 5, '2025-09-05'),
(3, 3, 3, 0, '2025-09-10');
```

```sql
CREATE OR REPLACE VIEW vw_order_summary AS
SELECT
    s.order_id,
    s.order_date,
    p.item_name,
    (p.price * s.qty) - ((p.price * s.qty) * s.discount / 100) AS net_amount
FROM sales_data AS s
JOIN product_data AS p ON p.item_id = s.item_id;


SELECT * FROM vw_order_summary;


CREATE ROLE chetan LOGIN PASSWORD '1234';


GRANT SELECT ON vw_order_summary TO chetan;


CREATE TABLE staff_records (
    staff_id SERIAL PRIMARY KEY,
    staff_name TEXT NOT NULL,
    division TEXT NOT NULL
);


INSERT INTO staff_records (staff_name, division) VALUES
('Aman', 'Sales'),
('Deepak', 'Finance'),
('Anita', 'Sales');


CREATE OR REPLACE VIEW vw_sales_team AS
SELECT staff_id, staff_name, division
FROM staff_records
WHERE division = 'Sales'
WITH CHECK OPTION;
```

```
SELECT * FROM vw_sales_team;


INSERT INTO vw_sales_team (staff_id, staff_name, division) VALUES (10, 'Pooja',
'Admin');
```

OUTPUT:

```
Output:

CREATE TABLE
CREATE TABLE
CREATE TABLE
INSERT 0 3
INSERT 0 3
INSERT 0 3
CREATE VIEW
 order_id | order_date | item_name |          net_amount
----------+------------+-----------+---------------------------
        1 | 2025-09-01 | Laptop    |          99000.000000000000
        2 | 2025-09-05 | Phone     |       23750.0000000000000000
        3 | 2025-09-10 | Tablet    | 54000.00000000000000000000
(3 rows)


psql:commands.sql:40: ERROR:  permission denied to create role
```