

**Group No :- 18**  
**Vaghasiya Nikhil (i.d. :- 202003042)**  
**Hardikkumar jani (i.d. :- 202003041)**  
**Lab No.: - 10(25-11-2021)**  
**Dairy product management system**

### Queries:-

- Find the name of customers who buy Amul ghee(id=AMGE5G).

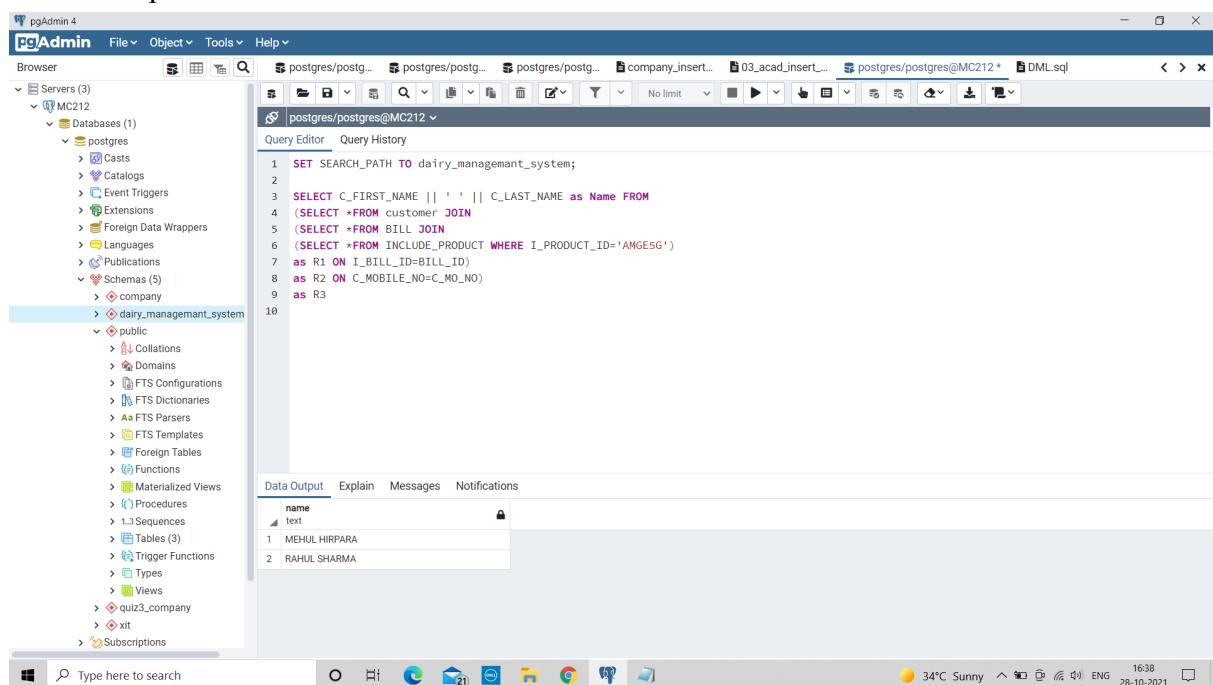
❖ Relational algebra:-

➤  $\Pi_{(C\_FIRST\_NAME, C\_LAST\_NAME)} \rho(R3, CUSTOMER \bowtie_{(C\_MOBILE\_NO=C\_MO\_NO)} \rho(R2, BILL \bowtie_{(I\_BILL\_ID=BILL\_ID)} \rho(R1, \sigma_{(I\_PRODUCT\_ID=AMGE5G)}(\text{INCLUDE\_PRODUCT})))$

❖ SQL code:-

```
> SELECT DISTINCT C_FIRST_NAME || ' ' || C_LAST_NAME as Name
  FROM
    (SELECT *FROM customer JOIN
     (SELECT *FROM BILL JOIN
      (SELECT *FROM INCLUDE_PRODUCT WHERE
       I_PRODUCT_ID='AMGE5G')
     as R1 ON I_BILL_ID=BILL_ID)
    as R2 ON C_MOBILE_NO=C_MO_NO)
   as R3
```

❖ Output:-



The screenshot shows the pgAdmin 4 interface with the following details:

- Browser:** Shows the database structure under "Servers (3) / MC212 / databases (1) / postgres".
- Query Editor:** Contains the following SQL code:

```
1 SET SEARCH_PATH TO dairy_management_system;
2
3 SELECT C_FIRST_NAME || ' ' || C_LAST_NAME as Name
4   (SELECT *FROM customer JOIN
5    (SELECT *FROM BILL JOIN
6     (SELECT *FROM INCLUDE_PRODUCT WHERE
7      I_PRODUCT_ID='AMGE5G')
8    as R1 ON I_BILL_ID=BILL_ID)
9    as R2 ON C_MOBILE_NO=C_MO_NO)
10   as R3
```
- Data Output:** Displays the results of the query:

name
MEHUL HIRPARA
RAHUL SHARMA
- System Bar:** Shows the date (28-10-2021), time (16:38), and system status (34°C Sunny, ENG).

- Find a merchant name whose transport bill payment type is cash.

❖ Relational algebra:-

- $\Pi_{(MERCHANT\_FIRST\_NAME, MERCHANT\_LAST\_NAME)} \rho(R2, TRANSPORT \bowtie_{(T\_BILL\_ID=BILL\_ID)} \rho(R1, \sigma_{(B\_PAYMENT\_TYPE=CASH)} BILL))$
- ❖ SQL code:-
- SELECT DISTINCT MERCHANT\_FIRST\_NAME || ' ' || MERCHANT\_LAST\_NAME as Name FROM (SELECT \*FROM TRANSPORT JOIN (SELECT \*FROM BILL WHERE B\_PAYMENT\_TYPE='CASH') R1 ON T\_BILL\_ID=BILL\_ID) R2
- ❖ Output:-

```

1 SET SEARCH_PATH TO dairy_management_system;
2
3 SELECT MERCHANT_FIRST_NAME || ' ' || MERCHANT_LAST_NAME as Name FROM
4 (SELECT *FROM TRANSPORT JOIN
5 (SELECT *FROM BILL WHERE B_PAYMENT_TYPE='CASH')
6 as R1 ON T_BILL_ID=BILL_ID)
7 as R2
8

```

name
JAYRAJ MEHTA
PARTH PATEL
JAYSRIE PATHAK
JAY PATHAK
HARSH RAMAVAT

Successfully run. Total query runtime: 35 msec. 5 rows affected.

### 3. Find the seller whose product we purchased but can't sell.

- ❖ Relational algebra:-
- $\Pi_{SELLER^*} (SELLER \bowtie_{(SELLER\_ID=P\_SELLER\_ID)} \rho(R2, PRODUCT \text{ NATURAL JOIN } \rho(R1, \Pi_{(PRODUCT\_ID)} PRODUCT - \Pi_{(SR\_PRODUCT\_CODE \rightarrow PRODUCT\_ID)} SELLING\_REPORT)))$
- ❖ SQL code:-
- SELECT \*FROM SELLER WHERE SELLER\_ID IN (SELECT SELLER\_ID FROM SELLER JOIN (SELECT \*FROM PRODUCT NATURAL JOIN (SELECT PRODUCT\_ID FROM PRODUCT EXCEPT (SELECT SR\_PRODUCT\_CODE as PRODUCT\_ID FROM SELLING\_REPORT))) R1 ON SELLER\_ID=P\_SELLER\_ID) R2 ON SELLER\_ID=P\_SELLER\_ID)
- ❖ Output:-

The screenshot shows the pgAdmin 4 interface. In the left sidebar, under 'Servers', there is one entry: 'MC212' which contains a single database named 'postgres'. Under 'postgres', there are various objects like Casts, Event Triggers, Extensions, Foreign Data Wrappers, Languages, Publications, Schemas, and Tables. The 'Tables' section is expanded, showing 'dairy\_management\_system' which contains six rows of data.

In the main area, a query editor window is open with the following SQL code:

```

1 SET SEARCH_PATH TO dairy_management_system;
2
3 SELECT *FROM SELLER WHERE SELLER_ID IN
4 (SELECT SELLER_ID FROM SELLER JOIN
5 (SELECT *FROM PRODUCT NATURAL JOIN
6 (SELECT PRODUCT_ID FROM PRODUCT EXCEPT
7 (SELECT SR_PRODUCT_CODE AS PRODUCT_ID FROM SELLING_REPORT)))
8 AS R1)
9 AS R2 ON SELLER_ID=P_SELLER_ID;

```

Below the query editor, a data output table is displayed with the following data:

seller_id	s_first_name	s_last_name	s_company_name	s_mobile_no
10000000	AKSHAR	MEHATA	AMUL	9681454888
11000000	RAJ	PATEL	MAHI	9781454888
20001020	JAYRAJ	JADEJA	SORATH	7081054788
55100709	KRUPAL	SHAH	TULSI	9688454118
30111010	KESHAV	PATEL	GOVARDHAN	9681224928
10001010	KEVAL	JOSHI	PATANJALI	9211251918

4. Find a worker who did not interact with a customer who bought a product today(DATE'1999-05-07') at the given outlet.

❖ Relational algebra:-

➤ WORKER NATURAL JOIN  $\rho_{(W\_ID \rightarrow WORKER\_ID)} \Pi_{(WORKER\_ID)} WORKER - \Pi_{(W\_ID \rightarrow WORKER\_ID)} \rho_{(R2, CUSTOMER \bowtie_{(C\_MOBILE\_NO=C\_MO\_NO)}} \rho_{(R1, \sigma_{(B\_DATE=DATE'1999-05-07')} BILL))}$

❖ SQL code:-

➤ SELECT \* FROM WORKER NATURAL JOIN  
 ➤ (SELECT WORKER\_ID FROM WORKER EXCEPT  
 ➤ (SELECT W\_ID as WORKER\_ID FROM CUSTOMER JOIN  
 ➤ (SELECT \* FROM BILL WHERE B\_DATE=DATE'1999-05-07')  
 ➤ as R1 ON C\_MOBILE\_NO=C\_MO\_NO)) as R3

❖ Output:-

The screenshot shows the pgAdmin 4 interface. On the left, the 'Browser' pane displays a tree view of the database structure under 'Servers (3)'. One node is expanded to show 'dairy\_management\_system' with various objects like 'Tables (3)', 'Views', and 'Triggers'. The main area is a 'Query Editor' window titled 'postgres@MC212'. It contains the following SQL code:

```

1 SET SEARCH_PATH TO dairy_management_system;
2
3 SELECT * FROM WORKER NATURAL JOIN
4 (SELECT WORKER_ID FROM WORKER EXCEPT
5 (SELECT W_ID AS WORKER_ID FROM CUSTOMER JOIN
6 (SELECT * FROM BILL WHERE B_DATE=DATE '1999-05-01')
7 AS R1 ON C_MOBILE_NO=C_MO_NO)) AS R3
8

```

Below the code, the 'Data Output' tab is selected, showing a table with 12 rows of data. The columns are:

worker_id	w_first_name	w_last_name	w_address	w_birth_date	w_joining_date	w_salary	
1	22	KARAN	KUNJERA	BALAJI-HALL...	1978-01-05	1999-05-01	12000
2	27	UTAM	PATEL	DHEBAR ROAD...	1983-04-17	2001-05-01	11000
3	30	JAY	THUMMAR	KOTHARIYA RO...	1983-04-17	2004-05-01	11000
4	21	JAY	PATEL	UMIYA-CHOWK...	1973-02-06	1999-05-01	20000
5	32	SURESH	JANJANIYA	AAMBEDKAR...	1979-10-10	2004-05-01	12000
6	23	SURESH	JANJANIYA	AAMBEDKAR...	1979-01-10	1999-05-01	11000
7	31	ARJUN	SINGH	BALAJI-HALL...	1982-01-05	2004-05-01	11000
8	1	YAGNIK	DETHARIYA	UMIYA-CHOWK...	1970-01-01	1999-05-01	[null]
9	26	RAJ	SHARMA	INDIRA CIRCLE...	1983-01-05	2001-05-01	1000
10	28	JAY	THUMMAR	KOTHARIYA RO...	1983-04-17	2001-05-01	11000
11	24	ARJUN	SINGH	BALAJI-HALL...	1982-01-05	1999-05-01	11000
12	25	YASHIK	AGARWAL	NANA-MOVA ...	1983-01-02	2001-05-01	19000

The system tray at the bottom right shows the date as 28-10-2021, time as 16:53, and weather as 34°C Sunny.

5. Which product sold much on a given day(DATE'1999-05-01').

❖ Relational algebra:-

$$\begin{aligned}
 & \Rightarrow \Pi_{\text{PRODUCT}^*}(\text{PRODUCT} \Join_{(\text{PRODUCT\_ID} = \text{SR\_PRODUCT\_CODE})} \\
 & \quad \rho(R1, (\mathcal{F}_{\text{MAX}(\text{SR\_TOTAL\_QUANTITY})}(\sigma_{(\text{SR\_DATE} = \text{DATE}'1999-05-01')} \text{SELLING\_REPORT})) \\
 & \quad \Join_{(\max = \text{SR\_TOTAL\_QUANTITY})}(\sigma_{(\text{SR\_DATE} = \text{DATE}'1999-05-01')} \text{SELLING\_REPORT}))
 \end{aligned}$$

❖ SQL code:-

- SELECT \*FROM PRODUCT WHERE PRODUCT\_ID IN
- (SELECT PRODUCT\_ID FROM PRODUCT JOIN
- (SELECT \*FROM
- (SELECT MAX(SR\_TOTAL\_QUANTITY) FROM
- (SELECT \*FROM SELLING\_REPORT WHERE
- SR\_DATE=DATE'1999-05-01')
- as R4)
- as R2 JOIN (SELECT \*FROM SELLING\_REPORT WHERE
- SR\_DATE=DATE'1999-05-01')
- as R3 ON max=SR\_TOTAL\_QUANTITY)
- as R1 ON PRODUCT\_ID=SR\_PRODUCT\_CODE)

❖ Output:-

```

1 SET SEARCH_PATH TO dairy_management_system;
2
3 SELECT *FROM PRODUCT WHERE PRODUCT_ID IN
4 (SELECT PRODUCT_ID FROM PRODUCT JOIN
5 (SELECT *FROM
6 (SELECT MAX(SR_TOTAL_QUANTITY) FROM
7 (SELECT *FROM SELLING_REPORT WHERE SR_DATE=DATE'1999-05-01')
8 AS R4)
9 AS R2 JOIN (SELECT *FROM SELLING_REPORT WHERE SR_DATE=DATE'1999-05-01')
10 AS R3 ON max=SR_TOTAL_QUANTITY)
11 AS R1 ON PRODUCT_ID=SR_PRODUCT_CODE)
12

```

product_id	p_name	p_company_name	p_tax	p_unit_price	p_quantity	p_profit	p_seller_id
AMCGE1K	GHEE	AMUL	25	500	65	51	1000000

6. Which product sold much in a given month(month=5) of a given year(year=1999).

❖ Relational algebra:-

➤ PRODUCT  $\bowtie_{(PRODUCT\_ID=SR\_PRODUCT\_CODE)}$   
 $\rho(R1, \rho(R2, \mathcal{F}_{\text{MAX}}(\text{SUM}) \rho(R4, SR\_PRODUCT\_CODE \mathcal{F}_{\text{SUM}}(SR\_TOTAL\_QUANTITY)})$   
 $\rho(R5, \sigma_{(\text{month}(SR\_DATE)=5 \text{ and } \text{year}(SR\_DATE)=1999)} \text{SELLING\_REPORT})) \bowtie_{(\text{max}=\text{SUM})}$   
 $\rho(R3, SR\_PRODUCT\_CODE \mathcal{F}_{\text{SUM}}(SR\_TOTAL\_QUANTITY)) \rho(R6, \sigma_{(\text{month}(SR\_DATE)=5 \text{ and } \text{year}(SR\_DATE)=1999)} \text{SELLING\_REPORT}))$

❖ SQL code:-

- SELECT \*FROM PRODUCT JOIN
- (SELECT \*FROM
- (SELECT MAX(SUM) FROM
- (SELECT SR\_PRODUCT\_CODE,SUM(SR\_TOTAL\_QUANTITY) FROM (SELECT \*FROM SELLING\_REPORT WHERE EXTRACT(MONTH FROM SR\_DATE)=5 AND EXTRACT(YEAR FROM SR\_DATE)=1999)
- as R5 GROUP BY SR\_PRODUCT\_CODE)
- as R4)
- as R2 JOIN (SELECT SR\_PRODUCT\_CODE,SUM(SR\_TOTAL\_QUANTITY) FROM (SELECT \*FROM SELLING\_REPORT WHERE EXTRACT(MONTH FROM SR\_DATE)=5 AND EXTRACT(YEAR FROM SR\_DATE)=1999))
- as R6 GROUP BY SR\_PRODUCT\_CODE)
- as R3 ON max=SUM)
- as R1 ON PRODUCT\_ID=SR\_PRODUCT\_CODE)

❖ Output:-

The screenshot shows the pgAdmin 4 interface. The left sidebar displays a tree view of database objects under 'Servers' (MC212) and 'dairy\_management\_system'. The main area contains a 'Query Editor' tab with the following SQL code:

```

1 SET SEARCH_PATH TO dairy_management_system;
2
3 SELECT *FROM PRODUCT JOIN
4 (SELECT *FROM
5 (SELECT MAX(SUM) FROM
6 (SELECT SR_PRODUCT_CODE,SUM(SR_TOTAL_QUANTITY) FROM
7 (SELECT *FROM SELLING_REPORT WHERE EXTRACT(MONTH FROM SR_DATE)=5 AND EXTRACT(YEAR FROM SR_DATE)=1999)
8 AS R5 GROUP BY SR_PRODUCT_CODE)
9 AS R4)
10 AS R2 JOIN
11 (SELECT SR_PRODUCT_CODE,SUM(SR_TOTAL_QUANTITY) FROM
12 (SELECT *FROM SELLING_REPORT WHERE EXTRACT(MONTH FROM SR_DATE)=5 AND EXTRACT(YEAR FROM SR_DATE)=1999)
13 AS R3 GROUP BY SR_PRODUCT_CODE)
14 AS R2 ON max=SUM)
15 AS R1 ON PRODUCT_ID=SR_PRODUCT_CODE
16
17

```

Below the code, the 'Data Output' tab is selected, showing a table with one row of data:

product_id	p_name	p_company_name	p_tax	p_unit_price	p_quantity	p_profit	p_seller_id	max	sr_product_code	sum
AMG01L	MILK	AMUL	0.4	30	100	1	10000000	200	AMG01L	200

7. Give total profit for a given month(month=8) on a given outlet(OUTLET\_CODE= ‘Main’) for a given company(name=’AMUL’).

❖ Relational algebra:-

- $\rho(\sigma_{\text{year}(\text{SR\_DATE}) = 8} \mathcal{F}_{\text{COUNT}(\text{SR\_TOTAL\_PROFIT})} \rho(\text{R2}, \text{SELLING\_REPORT})$
- $\bowtie_{(\text{SR\_PRODUCT\_CODE} = \text{PRODUCT\_ID} \text{ AND } \text{OUTLET\_CODE} = \text{‘Main’} \text{ AND } \text{month}(\text{SR\_DATE}) = 8)}$
- $\rho(\text{R1}, \sigma_{\text{P\_COMPANY\_NAME} = \text{‘AMUL’}} \text{PRODUCT})$

❖ SQL code:-

- SELECT EXTRACT(YEAR FROM SR\_DATE),  
SUM(SR\_TOTAL\_PROFIT) FROM
- (SELECT \*FROM SELLING\_REPORT JOIN
- (SELECT \*FROM PRODUCT WHERE P\_COMPANY\_NAME='AMUL')
- as R1 ON SR\_PRODUCT\_CODE=PRODUCT\_ID AND  
SR\_OUTLET\_CODE='MAIN' AND EXTRACT(MONTH FROM  
SR\_DATE)=5)
- as R2 GROUP BY EXTRACT(YEAR FROM SR\_DATE)

❖ Output:-

```

SET SEARCH_PATH TO dairy_management_system;
SELECT EXTRACT(YEAR FROM SR_DATE), SUM(SR_TOTAL_PROFIT) FROM
(SELECT *FROM SELLING_REPORT JOIN
(SELECT *FROM PRODUCT WHERE P_COMPANY_NAME='AMUL')
AS R1 ON SR_PRODUCT_CODE=PRODUCT_ID AND SR_OUTLET_CODE='MAIN' AND EXTRACT(MONTH FROM SR_DATE)=5)
AS R2 GROUP BY EXTRACT(YEAR FROM SR_DATE);

```

date_part	sum
1	1999
	810

## 8. Give a customer list who has not to buy the product.

❖ Relational algebra:-

➢ CUSTOMER NATURAL JOIN  $\rho(R1, \Pi_{(C\_MOBILE\_NO)} CUSTOMER - \Pi_{(C\_MO\_NO \rightarrow C\_MOBILE\_NO)} BILL)$

❖ SQL code:-

- SELECT \*FROM CUSTOMER NATURAL JOIN
- (SELECT C\_MOBILE\_NO FROM CUSTOMER EXCEPT
- (SELECT C\_MO\_NO as C\_MOBILE\_NO FROM BILL))
- as R1

❖ Output:-

```

SET SEARCH_PATH TO dairy_management_system;
SELECT *FROM CUSTOMER NATURAL JOIN
(SELECT C_MOBILE_NO FROM CUSTOMER EXCEPT
(SELECT C_MO_NO as C_MOBILE_NO FROM BILL))
AS R1

```

c_mobile_no	c_first_name	c_last_name	c_locality	c_pincode	c_city	w_id
8543785243	SANKET	MISRA	RAJKOT	360005	RAJKOT	21
9764503421	SOMYA	PATIL	JUNAGADH	360001	JUNAGADH	27
9976435890	SAMIR	MALHOTRA	RAJKOT	360005	RAJKOT	28
8424562310	SUJATA	PATEL	RAJKOT	360005	RAJKOT	26

9. Give total payment of purchase report for a given MONTH(month=1) for each seller and year.

❖ Relational algebra:-

$\Rightarrow \text{year(PR_DATE),SELLER_ID} \mathcal{F}_{(\text{SUM(PR_TOTAL_AMOUNT}) \rightarrow \text{TOTAL})} \rho(R1, \sigma_{(\text{month(PR_DATE})=1)} \text{PURCHASE\_REPORT})$

❖ SQL code:-

- `SELECT SELLER_ID,EXTRACT(YEAR FROM PR_DATE),SUM(PR_TOTAL_AMOUNT) as TOTAL FROM PURCHASE_REPORT WHERE EXTRACT(MONTH FROM PR_DATE)=1`
- `as R1`
- `GROUP BY SELLER_ID,EXTRACT(YEAR FROM PR_DATE)`

❖ Output:-

```

SET SEARCH_PATH TO dairy_management_system;
SELECT SELLER_ID,EXTRACT(YEAR FROM PR_DATE),SUM(PR_TOTAL_AMOUNT) as TOTAL FROM
(SELECT *FROM PURCHASE_REPORT WHERE EXTRACT(MONTH FROM PR_DATE)=1)
as R1
GROUP BY SELLER_ID,EXTRACT(YEAR FROM PR_DATE);
    
```

seller_id	date_part	total	
1	10000000	1999	14000
2	10000000	2005	10000
3	11000000	2003	6000
4	20001020	2003	5000
5	20001020	2006	7000
6	20101010	2009	9000
7	55100709	2005	9000

10. Count total payment by payment type of customers for a given month(month=5) and year(year<=2005).

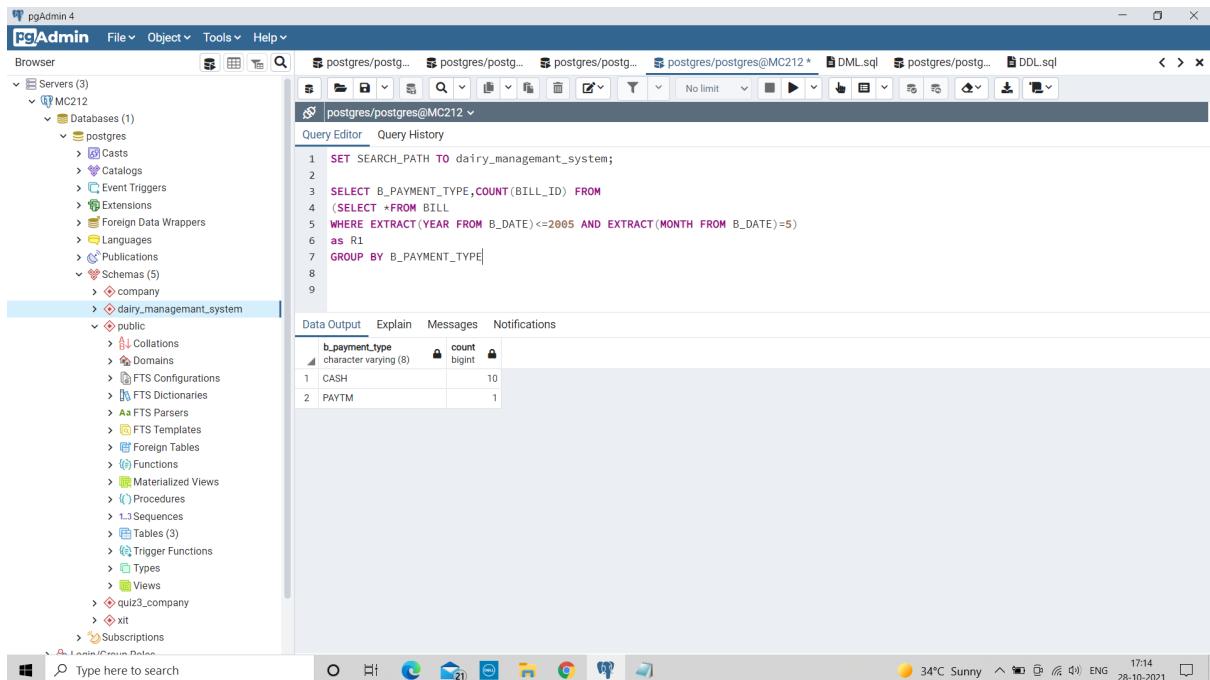
❖ Relational algebra:-

$\Rightarrow \text{B_PAYMENT_TYPE} \mathcal{F}_{\text{COUNT(BILL_ID)}} \rho(R1, \sigma_{(\text{month(B_DATE})=5 \text{ and } (\text{year(B_DATE}) \leq 2005)}) \text{BILL})$

❖ SQL code:-

- `SELECT B_PAYMENT_TYPE,COUNT(BILL_ID) FROM BILL`
- `WHERE EXTRACT(YEAR FROM B_DATE)<=2005 AND EXTRACT(MONTH FROM B_DATE)=5`
- `as R1`
- `GROUP BY B_PAYMENT_TYPE`

❖ Output:-



The screenshot shows the pgAdmin 4 interface. In the left sidebar, under the 'dairy\_management\_system' database, the 'Tables' section is expanded, showing a table named 'b\_payment\_type'. The main area displays a SQL query and its results.

```

1 SET SEARCH_PATH TO dairy_management_system;
2
3 SELECT B_PAYMENT_TYPE,COUNT(BILL_ID) FROM
4 (SELECT *FROM BILL
5 WHERE EXTRACT(YEAR FROM B_DATE)<=2005 AND EXTRACT(MONTH FROM B_DATE)=5)
6 as R1
7 GROUP BY B_PAYMENT_TYPE
8
9

```

The results table shows the count of bills for each payment type:

b_payment_type	count
CASH	10
PAYTM	1

11. Find sellers whose product we purchase but our payments not done yet means they have no purchase report reports.

❖ Relational algebra:-

➤ SELLER NATURAL JOIN  $\rho_{(P\_SELLER\_ID \rightarrow SELLER\_ID)} \Pi_{(SELLER\_ID)} PRODUCT - \Pi_{(SELLER\_ID)} PURCHASE\_REPORT$

❖ SQL code:-

- SELECT \* FROM SELLER NATURAL JOIN
- (SELECT P\_SELLER\_ID AS SELLER\_ID FROM PRODUCT EXCEPT
- (SELECT SELLER\_ID FROM PURCHASE\_REPORT))
- as R1

❖ Output:-

The screenshot shows the pgAdmin 4 interface. On the left, the 'Servers' tree view is expanded to show 'MC212' and its 'dairy\_management\_system' database. The 'Tables' node under 'dairy\_management\_system' is selected. In the center, the 'Query Editor' tab is active, displaying the following SQL code:

```

1 SET SEARCH_PATH TO dairy_management_system;
2
3 SELECT *FROM SELLER NATURAL JOIN
4 (SELECT P_SELLER_ID AS SELLER_ID FROM PRODUCT EXCEPT
5 (SELECT SELLER_ID FROM PURCHASE_REPORT))
6 as R1
7
8

```

Below the query editor, the 'Data Output' tab is selected, showing a table with one row of data:

seller_id	s_first_name	s_last_name	s_company_name	s_mobile_no
30111010	KESHAV	PATEL	GOVARDHAN	9681224928

12. Find a customer who visits a dairy more than 2 times on a given day(DATE'1999-05-01').

❖ Relational algebra:-

➢  $\Pi_{(\text{CUSTOMER}^*)}(\text{CUSTOMER SEMI-JOIN}_{(C\_MOBILE\_NO=C\_MO\_NO)} (\sigma_{(\text{COUNT}(\text{BILL})>2)}((C\_MO\_NO) \mathcal{F}_{(\text{COUNT}(\text{BILL\_ID}))} \rho(R1, \sigma_{(B\_DATE=DATE'1999-05-01')}(\text{BILL}))))$

❖ SQL code:-

➢ SELECT \*FROM CUSTOMER WHERE C\_MOBILE\_NO IN  
 ➢ (SELECT C\_MO\_NO FROM  
 ➢ (SELECT \*FROM BILL WHERE B\_DATE=DATE'1999-05-01')  
 ➢ as R1 GROUP BY C\_MO\_NO HAVING COUNT(BILL\_ID)>2)

❖ Output:-

The screenshot shows the pgAdmin 4 interface. On the left, the 'Servers' tree view is expanded to show 'MC212' and its 'dairy\_management\_system' database. The 'Tables' node under 'dairy\_management\_system' is selected. In the center, the 'Query Editor' tab contains the following SQL code:

```

1 SET SEARCH_PATH TO dairy_management_system;
2
3 SELECT *FROM CUSTOMER WHERE C_MOBILE_NO IN
4 (SELECT C_MO_NO FROM
5 (SELECT *FROM BILL WHERE B_DATE='1999-05-01')
6 as R1 GROUP BY C_MO_NO HAVING COUNT(BILL_ID)>2)
7
8

```

The 'Data Output' tab shows the results of the query:

c_mobile_no	c_first_name	c_last_name	c_locality	c_pincode	c_city	w_id
8823432453	MEHUL	HIRPARA	RAJKOT	360005	RAJKOT	21

13. Find feedback that have ratings greater than 3 and list customer and product for that feedback.

❖ Relational algebra:-

➢ CUSTOMER NATURAL JOIN

$$\rho(R2, \rho(R3, \Pi_{(PRODUCT\_ID, P\_NAME, P\_COMPANY\_NAME)} PRODUCT) NATURAL JOIN \\ \Pi_{(F\_RATING, CUSTOMER\_MO\_NO \rightarrow C\_MO\_NO, F\_TITLE \rightarrow PRODUCT\_ID)} \\ \rho(R1, \sigma_{(F\_RATE > 3)} FEEDBACK))$$

❖ SQL code:-

➢ SELECT \*FROM CUSTOMER NATURAL JOIN

➢ (SELECT \*FROM

➢ (SELECT PRODUCT\_ID, P\_NAME, P\_COMPANY\_NAME  
FROM PRODUCT)

➢ as R3 NATURAL JOIN

➢ (SELECT F\_RATING, CUSTOMER\_MO\_NO AS  
C\_MOBILE\_NO, F\_TITLE AS PRODUCT\_ID FROM  
FEEDBACK WHERE F\_RATING > 3)

➢ as R1) as R2

❖ Output:-

```

1 SET SEARCH_PATH TO dairy_management_system;
2
3 SELECT *FROM CUSTOMER NATURAL JOIN
4 (SELECT *FROM
5 (SELECT PRODUCT_ID,P_NAME,P_COMPANY_NAME FROM PRODUCT)
6 as R3 NATURAL JOIN
7 (SELECT F_RATING,CUSTOMER_MO_NO AS C_MOBILE_NO,F_TITLE AS PRODUCT_ID FROM FEEDBACK WHERE F_RATING<3)
8 as R1) as R2
9
10

```

c_mobile_no	c_first_name	c_last_name	c_locality	c_pincode	c_city	w_id	product_id	p_name	p_company_name	f_rating
8823432453	MEHUL	HIRPARA	RAJKOT	360005	RAJKOT	21	AMCGE1K	GHEE	AMUL	4.5
9562406978	PRADIP	MAHETA	RAJKOT	360005	RAJKOT	23	AMG01L	MILK	AMUL	4.0
9764563421	SUSMITA	GUPTA	JAMNAGAR	360004	JAMNAGAR	29	GOVGE1K	GHEE	GOVARDHAN	5.0

```

1 SET SEARCH_PATH TO dairy_management_system;
2
3 SELECT *FROM CUSTOMER NATURAL JOIN
4 (SELECT *FROM
5 (SELECT PRODUCT_ID,P_NAME,P_COMPANY_NAME FROM PRODUCT)
6 as R3 NATURAL JOIN
7 (SELECT F_RATING,CUSTOMER_MO_NO AS C_MOBILE_NO,F_TITLE AS PRODUCT_ID FROM FEEDBACK WHERE F_RATING<3)
8 as R1) as R2
9
10

```

c_locality	c_pincode	c_city	w_id	product_id	p_name	p_company_name	f_rating
RAJKOT	360005	RAJKOT	21	AMCGE1K	GHEE	AMUL	4.5
RAJKOT	360005	RAJKOT	23	AMG01L	MILK	AMUL	4.0
JAMNAGAR	360004	JAMNAGAR	29	GOVGE1K	GHEE	GOVARDHAN	5.0

14. Find details of customers who give us bad(less than 3) ratings and list that products.

❖ Relational algebra:-

➢ CUSTOMER NATURAL JOIN

$$\rho(R2, \rho(R3, \Pi_{(PRODUCT\_ID, P\_NAME, P\_COMPANY\_NAME)} PRODUCT) NATURAL JOIN \\ \Pi_{(F\_RATING, CUSTOMER\_MO\_NO \rightarrow C\_MO\_NO, F\_TITLE \rightarrow PRODUCT\_ID)} \\ \rho(R1, \sigma_{(F\_RATE < 3)} FEEDBACK))$$

❖ SQL code:-

➢ SELECT \*FROM CUSTOMER NATURAL JOIN

- (SELECT \*FROM
- (SELECT PRODUCT\_ID,P\_NAME,P\_COMPANY\_NAME  
FROM PRODUCT)
- as R3 NATURAL JOIN
- (SELECT F\_RATING,CUSTOMER\_MO\_NO AS  
C\_MOBILE\_NO,F\_TITLE AS PRODUCT\_ID FROM  
FEEDBACK WHERE F\_RATING<3)
- as R1) as R2

❖ Output:-

```

1 SET SEARCH_PATH TO dairy_management_system;
2
3 SELECT *FROM CUSTOMER NATURAL JOIN
4 (SELECT *FROM
5 (SELECT PRODUCT_ID,P_NAME,P_COMPANY_NAME FROM PRODUCT)
6 as R3 NATURAL JOIN
7 (SELECT F_RATING,CUSTOMER_MO_NO AS C_MOBILE_NO,F_TITLE AS PRODUCT_ID FROM FEEDBACK WHERE F_RATING<3)
8 as R1) as R2
9
10

```

c_mobile_no	c_first_name	c_last_name	c_locality	c_pincode	c_city	w_id	product_id	p_name
9027853420	RAHUL	SHARMA	RAJKOT	360004	RAJKOT	22	AMGETK	GHEE
8123455678	HEMANT	JOSHI	RAJKOT	360005	RAJKOT	24	AMTA1L	MILK

Successfully run. Total query runtime: 44 msec. 2 rows affected.

```

1 SET SEARCH_PATH TO dairy_management_system;
2
3 SELECT *FROM CUSTOMER NATURAL JOIN
4 (SELECT *FROM
5 (SELECT PRODUCT_ID,P_NAME,P_COMPANY_NAME FROM PRODUCT)
6 as R3 NATURAL JOIN
7 (SELECT F_RATING,CUSTOMER_MO_NO AS C_MOBILE_NO,F_TITLE AS PRODUCT_ID FROM FEEDBACK WHERE F_RATING<3)
8 as R1) as R2
9
10

```

c_locality	c_pincode	c_city	w_id	product_id	p_name	p_company_name	f_rating
RAJKOT	360004	RAJKOT	22	AMGETK	GHEE	AMUL	2.0
RAJKOT	360005	RAJKOT	24	AMTA1L	MILK	AMUL	2.0

15. Find workers who work on more than 1 outlet and print their outlets.

❖ Relational algebra:-

➤ OUTLET  $\bowtie_{(W\_OUTLET\_ID=OUTLET\_CODE)}$   $\rho(R3, WORKING)$   
 $\bowtie_{(WORKER\_ID=W\_WORKER\_ID)}$   $\rho(R2, WORKER)$  NATURAL JOIN  
 $\rho(R1, \sigma_{COUNT>1}(W\_WORKER\_ID \rightarrow WORKER\_ID, \mathcal{F}_{COUNT(W\_OUTLET\_ID)} WORKING)))$

❖ SQL code:-

➤ SELECT \* FROM OUTLET JOIN  
➤ (SELECT \* FROM WORKING JOIN  
➤ (SELECT \* FROM WORKER NATURAL JOIN  
➤ (SELECT W\_WORKER\_ID as WORKER\_ID FROM WORKING  
GROUP BY WORKER\_ID HAVING  
COUNT(W\_OUTLET\_ID)>1)  
➤ as R1)  
➤ as R2 ON WORKER\_ID=W\_WORKER\_ID)  
➤ as R3 ON W\_OUTLET\_ID=OUTLET\_CODE

❖ Output:-

```

1 SET SEARCH_PATH TO dairy_management_system;
2
3 SELECT *FROM OUTLET JOIN
4 (SELECT *FROM WORKING JOIN
5 (SELECT *FROM WORKER NATURAL JOIN
6 (SELECT W_WORKER_ID as WORKER_ID FROM WORKING GROUP BY WORKER_ID HAVING COUNT(W_OUTLET_ID)>1)
7 as R1)
8 as R2 ON WORKER_ID=W_WORKER_ID)
9 as R3 ON W_OUTLET_ID=OUTLET_CODE
10

```

outlet_code	o_starting_date	o_address	w_worker_id	w_outlet_id	worker_id	w_first_name	w_last_name	w_address
GD1	2001-07-09	Balaji hall, rajkot	1	GD1	1	YAGNIK	DETHARIYA	UMIYA-CHOW
GD2	2005-10-25	Gondal 360031	1	GD2	1	YAGNIK	DETHARIYA	UMIYA-CHOW
GD3	2007-11-12	Umiya chowk r...	1	GD3	1	YAGNIK	DETHARIYA	UMIYA-CHOW
MAIN	1999-05-01	Mavadi chokad...	1	MAIN	1	YAGNIK	DETHARIYA	UMIYA-CHOW
GD1	2001-07-09	Balaji hall, rajkot	23	GD1	23	SURESH	JANJANIYA	AAMBEDKAR
GD2	2005-10-25	Gondal 360031	23	GD2	23	SURESH	JANJANIYA	AAMBEDKAR
GD3	2007-11-12	Umiya chowk r...	23	GD3	23	SURESH	JANJANIYA	AAMBEDKAR
MAIN	1999-05-01	Mavadi chokad...	23	MAIN	23	SURESH	JANJANIYA	AAMBEDKAR

ss	w_worker_id	w_outlet_id	worker_id	w_first_name	w_last_name	w_address	w_birth_date	w_joining_date	w_salary
rajkot	1	GD1	1	YAGNIK	DETHARIYA	UMIYA-CHOW...	1970-01-01	1999-05-01	[null]
360031	1	GD2	1	YAGNIK	DETHARIYA	UMIYA-CHOW...	1970-01-01	1999-05-01	[null]
hawk r...	1	GD3	1	YAGNIK	DETHARIYA	UMIYA-CHOW...	1970-01-01	1999-05-01	[null]
chokad...	1	MAIN	1	YAGNIK	DETHARIYA	UMIYA-CHOW...	1970-01-01	1999-05-01	[null]
rajkot	23	GD1	23	SURESH	JANJANIYA	AAMBEDKAR...	1979-01-10	1999-05-01	11000
360031	23	GD2	23	SURESH	JANJANIYA	AAMBEDKAR...	1979-01-10	1999-05-01	11000
hawk r...	23	GD3	23	SURESH	JANJANIYA	AAMBEDKAR...	1979-01-10	1999-05-01	11000
chokad...	23	MAIN	23	SURESH	JANJANIYA	AAMBEDKAR...	1979-01-10	1999-05-01	11000

## 16. Find a worker who works on all outlets.(DIVISION QUERY)

❖ Relational algebra:-

➢ WORKER NATURAL JOIN  $\rho_{(R2, \Pi_{(W\_WORKER\_ID \rightarrow WORKER\_ID)} WORKING - \Pi_{(WORKER\_ID)}(R3, \Pi_{(W\_WORKER\_ID \rightarrow WORKER\_ID, OUTLET\_CODE)} (WORKING \times OUTLET - WORKING))}$

❖ SQL code:-

➢ SELECT \*FROM WORKER NATURAL JOIN  
 ➢ (SELECT W\_WORKER\_ID as WORKER\_ID FROM WORKING EXCEPT

- (SELECT WORKER\_ID FROM
- (SELECT W\_WORKER\_ID as WORKER\_ID, OUTLET\_CODE  
FROM WORKING CROSS JOIN OUTLET EXCEPT
- (SELECT \*FROM WORKING))
- as R3))
- as R2

❖ Output:-

```

SET SEARCH_PATH TO dairy_management_system;
SELECT *FROM WORKER NATURAL JOIN
(SELECT W_WORKER_ID as WORKER_ID FROM WORKING EXCEPT
(SELECT WORKER_ID FROM
(SELECT W_WORKER_ID as WORKER_ID, OUTLET_CODE FROM WORKING CROSS JOIN OUTLET EXCEPT
(SELECT *FROM WORKING))
as R3)
as R2

```

worker_id	w_first_name	w_last_name	w_address	w_birth_date	w_joining_date	w_salary
1	YAGNIK	DETHARIYA	UMIYA-CHOWK...	1970-01-01	1999-05-01	[null]
2	SURESH	JANJANIYA	AAMBEDKAR...	1979-01-10	1999-05-01	11000

17. Find transports which do not include any products.

❖ Relational algebra:-

- TRANSPORT NATURAL JOIN  $\rho_{(R1, \Pi_{(T\_BILL\_ID)} TRANSPORT - \Pi_{(I\_BILL\_ID)} INCLUDE\_PRODUCT)}$

❖ SQL code:-

- SELECT \*FROM TRANSPORT NATURAL JOIN
- (SELECT T\_BILL\_ID FROM TRANSPORT EXCEPT
- (SELECT I\_BILL\_ID FROM INCLUDE\_PRODUCT))
- as R1

❖ Output:-

```

1 SET SEARCH_PATH TO dairy_management_system;
2
3 SELECT * FROM TRANSPORT NATURAL JOIN
4 (SELECT T_BILL_ID FROM TRANSPORT EXCEPT
5 (SELECT I_BILL_ID FROM INCLUDE_PRODUCT))
6 as R1

```

t_bill_id	transport_id	driver_first_name	driver_last_name	t_date	address	t_total_amount	merchant_first_name	merchant_last_name
199905101	16000	HARSH	PATEL	1999-05-10	SHAPAR, RA...	50	JAY	PATHAK
200005101	19000	HARSH	PATEL	2000-05-10	SHAPAR, RA...	50	HARSH	RAMAVAT

```

1 SET SEARCH_PATH TO dairy_management_system;
2
3 SELECT * FROM TRANSPORT NATURAL JOIN
4 (SELECT T_BILL_ID FROM TRANSPORT EXCEPT
5 (SELECT I_BILL_ID FROM INCLUDE_PRODUCT))
6 as R1

```

merch_id	driver_last_name	t_date	address	t_total_amount	merchant_first_name	merchant_last_name	merchant_mo_no	t_worker_id
PATEL	1999-05-10	SHAPAR, RA...	50	JAY	PATHAK		9030104011	22
PATEL	2000-05-10	SHAPAR, RA...	50	HARSH	RAMAVAT		9030104011	22

18. Which outlet sold the maximum product on a given month(9) and year(2007).

❖ Relational algebra:-

➤ OUTLET NATURAL JOIN

$\rho(R1, \rho(R4, (SR\_OUTLET\_CODE \rightarrow OUTLET\_CODE) \mathcal{F}_{(SUM(SR\_TOTAL\_QUANTITY) \rightarrow total)})$

$\rho(R3, \sigma_{(YEAR(SR\_DATE)=2007 \text{ AND } MONTH(SR\_DATE)=9)} SELLING\_REPORT)) \bowtie_{(total=MAX)}$

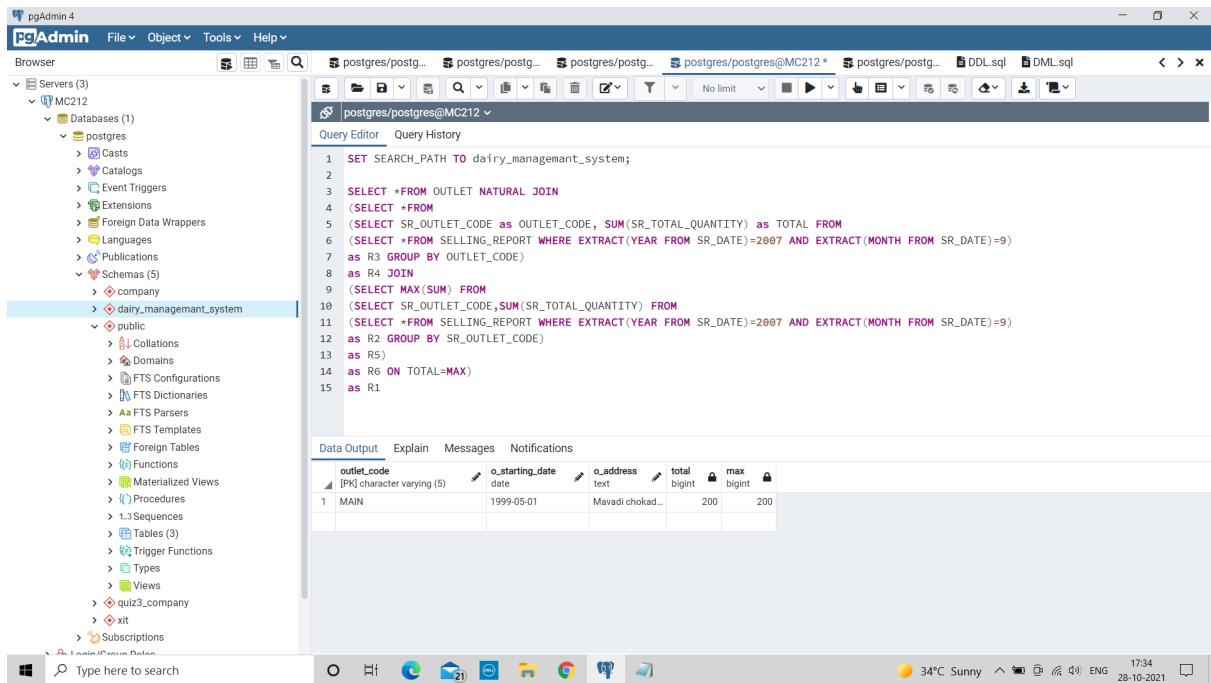
$\rho(R6, \mathcal{F}_{(MAX(SUM))}) \rho(R5, (SR\_OUTLET\_CODE) \mathcal{F}_{(SUM(SR\_TOTAL\_QUANTITY))})$

$\rho(R2, \sigma_{(YEAR(SR\_DATE)=2007 \text{ AND } MONTH(SR\_DATE)=9)} SELLING\_REPORT)))$

❖ SQL code:-

- SELECT \*FROM OUTLET NATURAL JOIN
- (SELECT \*FROM
- (SELECT SR\_OUTLET\_CODE as OUTLET\_CODE,  
SUM(SR\_TOTAL\_QUANTITY) as TOTAL FROM
- (SELECT \*FROM SELLING\_REPORT WHERE  
EXTRACT(YEAR FROM SR\_DATE)=2007 AND  
EXTRACT(MONTH FROM SR\_DATE)=9)
- as R3 GROUP BY OUTLET\_CODE)
- as R4 JOIN
- (SELECT MAX(SUM) FROM
- (SELECT SR\_OUTLET\_CODE,SUM(SR\_TOTAL\_QUANTITY)  
FROM
- (SELECT \*FROM SELLING\_REPORT WHERE  
EXTRACT(YEAR FROM SR\_DATE)=2007 AND  
EXTRACT(MONTH FROM SR\_DATE)=9)
- as R2 GROUP BY SR\_OUTLET\_CODE)
- as R5)
- as R6 ON TOTAL=MAX)
- as R1

#### ❖ Output:-



The screenshot shows the pgAdmin 4 interface. On the left is a tree view of database objects under 'Servers' (MC212). The 'dairy\_management\_system' database is selected. The main area contains a 'Query Editor' tab with the following SQL code:

```

1 SET SEARCH_PATH TO dairy_management_system;
2
3 SELECT *FROM OUTLET NATURAL JOIN
4 (SELECT *FROM
5 (SELECT SR_OUTLET_CODE as OUTLET_CODE, SUM(SR_TOTAL_QUANTITY) as TOTAL FROM
6 (SELECT *FROM SELLING_REPORT WHERE EXTRACT(YEAR FROM SR_DATE)=2007 AND EXTRACT(MONTH FROM SR_DATE)=9)
7 as R3 GROUP BY OUTLET_CODE)
8 as R4 JOIN
9 (SELECT MAX(SUM) FROM
10 (SELECT SR_OUTLET_CODE,SUM(SR_TOTAL_QUANTITY) FROM
11 (SELECT *FROM SELLING_REPORT WHERE EXTRACT(YEAR FROM SR_DATE)=2007 AND EXTRACT(MONTH FROM SR_DATE)=9)
12 as R2 GROUP BY SR_OUTLET_CODE)
13 as R5)
14 as R6 ON TOTAL=MAX)
15 as R1

```

Below the code, the 'Data Output' tab is active, showing the results of the query:

outlet_code	o_starting_date	o_address	total	max
MAIN	1999-05-01	Mavadi chokad...	200	200

19. Find customers who purchased products from more than one outlet and print outlet.

❖ Relational algebra:-

➤ OUTLET NATURAL JOIN

$(R4, (R3, \Pi_{(C\_MO\_NO \rightarrow C\_MOBILE\_NO, O\_CODE \rightarrow OUTLET\_CODE)} BILL) \text{ NATURAL JOIN}$

$(R2, CUSTOMER \text{ NATURAL JOIN}$

$(R1, \sigma_{(\text{COUNT} > 1)}((C\_MO\_NO \rightarrow C\_MOBILE\_NO) \mathcal{F}_{\text{COUNT}(\text{DISTINCT } O\_CODE)} BILL)))$

❖ SQL code:-

➤ SELECT \* FROM OUTLET NATURAL JOIN

➤ (SELECT \* FROM

➤ (SELECT C\_MO\_NO AS C\_MOBILE\_NO, O\_CODE AS  
OUTLET\_CODE FROM BILL)

➤ as R3 NATURAL JOIN (SELECT \* FROM CUSTOMER  
NATURAL JOIN

➤ (SELECT C\_MO\_NO AS C\_MOBILE\_NO FROM BILL GROUP  
BY C\_MOBILE\_NO HAVING COUNT(DISTINCT  
O\_CODE)>1)

➤ as R1)

➤ as R2)

➤ as R4

❖ Output:-

The screenshot shows the pgAdmin 4 interface. In the top bar, it says 'pgAdmin 4' and 'File Object Tools Help'. Below the bar, there's a toolbar with various icons. The main area has a title bar 'postgres/postgres@MC212' and tabs for 'Query Editor' and 'Query History'. The 'Query Editor' tab is active, containing the following SQL code:

```

1 SET SEARCH_PATH TO dairy_management_system;
2
3 SELECT * FROM OUTLET NATURAL JOIN
4 (SELECT * FROM
5 (SELECT C_MO_NO AS C_MOBILE_NO, O_CODE AS OUTLET_CODE FROM BILL)
6 AS R3 NATURAL JOIN (SELECT * FROM CUSTOMER NATURAL JOIN
7 (SELECT C_MO_NO AS C_MOBILE_NO FROM BILL GROUP BY C_MOBILE_NO HAVING COUNT(DISTINCT O_CODE)>1)
8 AS R1)
9 AS R2)
10 AS R4

```

Below the code, the 'Data Output' tab is selected, showing a table with the following data:

outlet_code	o_starting_date	c_address	c_mobile_no	c_first_name	c_last_name	c_locality	c_pincode	c_city	w_id
1 MAIN	1999-05-01	Mavadi chokad...	8823432453	MEHUL	HIRPARA	RAJKOT	360005	RAJKOT	21
2 MAIN	1999-05-01	Mavadi chokad...	8823432453	MEHUL	HIRPARA	RAJKOT	360005	RAJKOT	21
3 GD1	2001-07-09	Balaji hall, rajkot	8823432453	MEHUL	HIRPARA	RAJKOT	360005	RAJKOT	21
4 MAIN	1999-05-01	Mavadi chokad...	9764563421	SUSMITA	GUPTA	JAMNAGAR	360004	JAMNAGAR	29
5 GD3	2007-11-12	Umiya chowk r...	9764563421	SUSMITA	GUPTA	JAMNAGAR	360004	JAMNAGAR	29

20. Find a seller who sold products in all outlets.(DIVISION QUERY)

❖ Relational algebra:-

- SELLER NATURAL JOIN  $\rho(R2, \Pi_{(SELLER\_ID)} PURCHASE\_REPORT - \Pi_{(SELLER\_ID)} \rho(R1, \Pi_{(SELLER\_ID, OUTLET\_CODE)} (PURCHASE\_REPORT \times OUTLET) - \Pi_{(SELLER\_ID, OUTLET\_CODE)} PURCHASE\_REPORT))$
- ❖ SQL code:-

  - SELECT \* FROM SELLER NATURAL JOIN
  - (SELECT SELLER\_ID FROM PURCHASE\_REPORT EXCEPT
  - (SELECT SELLER\_ID FROM
  - (SELECT SELLER\_ID, OUTLET.OUTLET\_CODE FROM PURCHASE\_REPORT CROSS JOIN OUTLET EXCEPT
  - (SELECT SELLER\_ID, OUTLET\_CODE FROM PURCHASE\_REPORT))
  - as R1)) as R2

❖ Output:-

```

pgAdmin 4
PgAdmin File Object Tools Help
Servers (3)
  MC212
    Databases (1)
      postgres
        Casts
        Catalogs
        Event Triggers
        Extensions
        Foreign Data Wrappers
        Languages
        Publications
      Schemas (5)
        company
        dairy_management_system
      public
        Collations
        Domains
        FTS Configurations
        FTS Dictionaries
        FTS Parsers
        FTS Templates
        Foreign Tables
        Functions
        Materialized Views
        Procedures
        Sequences
        Tables (3)
        Trigger Functions
        Types
        Views
      quiz3_company
      xit
      Subscriptions
  Login (Create Data)
  Type here to search
  34°C Sunny 17:38 28-10-2021
  
```

```

1 SET SEARCH_PATH TO dairy_management_system;
2
3 SELECT *FROM SELLER NATURAL JOIN
4 (SELECT SELLER_ID FROM PURCHASE_REPORT EXCEPT
5 (SELECT SELLER_ID FROM
6 (SELECT SELLER_ID,OUTLET.OUTLET_CODE FROM PURCHASE_REPORT CROSS JOIN OUTLET EXCEPT
7 (SELECT SELLER_ID,OUTLET_CODE FROM PURCHASE_REPORT))
8 as R1)) as R2
  
```

	seller_id	s_first_name	s_last_name	s_company_name	s_mobile_no
1	1000000	AKSHAR	MEHATA	AMUL	9681454888

21. Find milk which is purchased most by customers on month=2 and year=2008.

❖ Relational algebra:-

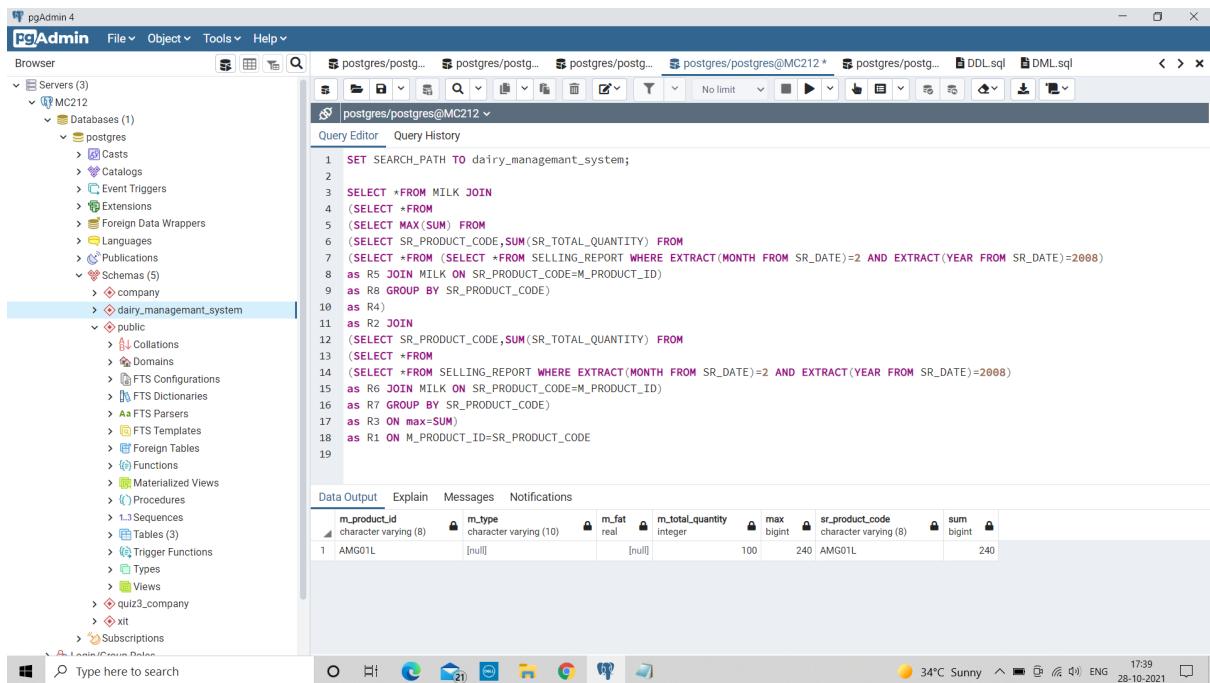
➤ MILK  $\bowtie_{(M\_PRODUCT\_ID=SR\_PRODUCT\_CODE)}$   
 $\rho(R1, \rho(R2, \mathcal{F}_{\text{MAX}(\text{sum})} \rho(R4, SR\_PRODUCT\_CODE \mathcal{F}_{\text{SUM}(SR\_TOTAL\_QUANTITY)})$   
 $\rho(R8, \rho(R5, \sigma_{(\text{month}(SR\_DATE)=5 \text{ and } \text{year}(SR\_DATE)=1999)} SELLING\_REPORT)$   
 $\bowtie_{(SR\_PRODUCT\_CODE=M\_PRODUCT\_ID)} MILK))) \bowtie_{(\text{max}=\text{sum})}$   
 $\rho(R3, SR\_PRODUCT\_CODE \mathcal{F}_{\text{SUM}(SR\_TOTAL\_QUANTITY)}) \rho(R7, \rho(R6, \sigma_{(\text{month}(SR\_DATE)=5 \text{ and } \text{year}(SR\_DATE)=1999)} SELLING\_REPORT) \bowtie_{(SR\_PRODUCT\_CODE=M\_PRODUCT\_ID)} MILK))$

❖ SQL code:-

➤ SELECT \* FROM MILK JOIN

- (SELECT \*FROM
- (SELECT MAX(SUM) FROM
- (SELECT SR\_PRODUCT\_CODE,SUM(SR\_TOTAL\_QUANTITY) FROM
- (SELECT \*FROM (SELECT \*FROM SELLING\_REPORT WHERE EXTRACT(MONTH FROM SR\_DATE)=2 AND EXTRACT(YEAR FROM SR\_DATE)=2008)
- as R5 JOIN MILK ON SR\_PRODUCT\_CODE=M\_PRODUCT\_ID)
- as R8 GROUP BY SR\_PRODUCT\_CODE)
- as R4)
- as R2 JOIN
- (SELECT SR\_PRODUCT\_CODE,SUM(SR\_TOTAL\_QUANTITY) FROM
- (SELECT \*FROM
- (SELECT \*FROM SELLING\_REPORT WHERE EXTRACT(MONTH FROM SR\_DATE)=2 AND EXTRACT(YEAR FROM SR\_DATE)=2008)
- as R6 JOIN MILK ON SR\_PRODUCT\_CODE=M\_PRODUCT\_ID)
- as R7 GROUP BY SR\_PRODUCT\_CODE)
- as R3 ON max=SUM)
- as R1 ON M\_PRODUCT\_ID=SR\_PRODUCT\_CODE

❖ Output:-



The screenshot shows the pgAdmin 4 interface. On the left, the 'Servers' tree view shows a connection to 'MC212' with a single database 'postgres'. The 'Tables' node under 'postgres' is expanded, showing tables like 'Casts', 'Catalogs', 'Event Triggers', 'Extensions', 'Foreign Data Wrappers', 'Languages', 'Publications', 'Schemas', 'company', 'dairy\_management\_system', 'public', 'Collations', 'Domains', 'FTS Configurations', 'FTS Dictionaries', 'FTS Parsers', 'FTS Templates', 'Foreign Tables', 'Functions', 'Materialized Views', 'Procedures', 'Sequences', 'Tables', 'Trigger Functions', 'Types', 'Views', 'quiz3.company', 'xit', and 'Subscriptions'. A search bar at the bottom left says 'Type here to search'. The main window contains a 'Query Editor' tab with the following SQL code:

```

1 SET SEARCH_PATH TO dairy_management_system;
2
3 SELECT *FROM MILK JOIN
4 (SELECT *FROM
5 (SELECT MAX(SUM) FROM
6 (SELECT SR_PRODUCT_CODE,SUM(SR_TOTAL_QUANTITY) FROM
7 (SELECT *FROM (SELECT *FROM SELLING_REPORT WHERE EXTRACT(MONTH FROM SR_DATE)=2 AND EXTRACT(YEAR FROM SR_DATE)=2008)
8 as R5 JOIN MILK ON SR_PRODUCT_CODE=M_PRODUCT_ID)
9 as R8 GROUP BY SR_PRODUCT_CODE)
10 as R4)
11 as R2 JOIN
12 (SELECT SR_PRODUCT_CODE,SUM(SR_TOTAL_QUANTITY) FROM
13 (SELECT *FROM
14 (SELECT *FROM SELLING_REPORT WHERE EXTRACT(MONTH FROM SR_DATE)=2 AND EXTRACT(YEAR FROM SR_DATE)=2008)
15 as R6 JOIN MILK ON SR_PRODUCT_CODE=M_PRODUCT_ID)
16 as R7 GROUP BY SR_PRODUCT_CODE)
17 as R3 ON max=SUM)
18 as R1 ON M_PRODUCT_ID=SR_PRODUCT_CODE
19

```

Below the code, there are tabs for 'Data Output', 'Explain', 'Messages', and 'Notifications'. The 'Data Output' tab displays the results of the query in a table:

m_product_id	m_type	m_fat	m_total_quantity	max	sr_product_code	sum
AMG01L	[null]	[null]	100	240	AMG01L	240

22. Find a manager who manages outlets and also interacts with customers.

❖ Relational algebra:-

➤  $\Pi_{(\text{WORKER}^*)} \text{WORKER} \bowtie_{(\text{WORKER\_ID}=\text{M\_WORKER\_ID})} \rho(\text{R1}, \text{MANAGER} \bowtie_{(\text{M\_WORKER\_ID}=\text{W\_ID})} \text{CUSTOMER})$

❖ SQL code:-

- SELECT \*FROM WORKER WHERE WORKER\_ID IN
- (SELECT M\_WORKER\_ID FROM

- (SELECT \*FROM MANAGER JOIN CUSTOMER ON M\_WORKER\_ID=W\_ID)
- as R1)

❖ Output:-

```

1 SET SEARCH_PATH TO dairy_management_system;
2
3 SELECT *FROM WORKER WHERE WORKER_ID IN
4 (SELECT M_WORKER_ID FROM
5 (SELECT *FROM MANAGER JOIN CUSTOMER ON M_WORKER_ID=W_ID)
6 as R1)

```

worker_id	w_first_name	w_last_name	w_address	w_birth_date	w_joining_date	w_salary
1	JAY	PATEL	UMIYA-CHOWK...	1973-02-06	1999-05-01	20000
2	YASHIK	AGARWAL	NANA-MOVA ...	1983-01-02	2001-05-01	19000
3	LALIT	PATEL	MAVADI CHOW...	1983-05-19	2004-05-01	20000

23. From which company(seller) we have the maximum product.

❖ Relational algebra:-

- SELLER NATURAL JOIN

$$\begin{aligned} & \rho(R3, P\_COMPANY\_NAME \rightarrow S\_COMPANY\_NAME \mathcal{F}_{COUNT(P\_PRODUCT\_ID)} PRODUCT) \bowtie_{(MAX=COUNT)} \\ & \rho(R2, \mathcal{F}_{MAX(COUNT)} \rho(R1, P\_COMPANY\_NAME \mathcal{F}_{COUNT(P\_PRODUCT\_ID)} PRODUCT)) \end{aligned}$$

❖ SQL code:-

- SELECT \*FROM SELLER NATURAL JOIN
- (SELECT \*FROM
- (SELECT P\_COMPANY\_NAME as
- S\_COMPANY\_NAME,COUNT(P\_PRODUCT\_ID)
- FROM PRODUCT GROUP BY S\_COMPANY\_NAME)
- as R3 JOIN (SELECT MAX(COUNT) FROM
- (SELECT P\_COMPANY\_NAME,COUNT(P\_PRODUCT\_ID) FROM
- PRODUCT GROUP BY P\_COMPANY\_NAME)
- as R1)
- as R2 ON MAX=COUNT)
- as R3

❖ Output:-

pgAdmin 4

Browser    File ▾ Object ▾ Tools ▾ Help ▾

Servers (3)  
MC212  
Databases (1)  
postgres  
Casts  
Catalogs  
Event Triggers  
Extensions  
Foreign Data Wrappers  
Languages  
Publications  
Schemas (5)  
company  
dairy\_management\_system  
public  
Collations  
Domains  
FTS Configurations  
FTS Dictionaries  
FTS Parsers  
FTS Templates  
Foreign Tables  
Materialized Views  
Procedures  
Sequences  
Tables (3)  
Trigger Functions  
Types  
Views  
quiz3.company  
xit  
Subscriptions

postgres/postgres@MC212 ~

Query Editor    Query History

```
1 SET SEARCH_PATH TO dairy_management_system;
2
3 SELECT * FROM SELLER NATURAL JOIN
4 (SELECT * FROM
5 (SELECT P_COMPANY_NAME AS S_COMPANY_NAME,COUNT(PRODUCT_ID)
6 FROM PRODUCT GROUP BY S_COMPANY_NAME)
7 AS R3 JOIN (SELECT MAX(COUNT) FROM
8 (SELECT P_COMPANY_NAME,COUNT(PRODUCT_ID) FROM PRODUCT GROUP BY P_COMPANY_NAME)
9 AS R1)
10 AS R2 ON MAX=COUNT)
11 AS R3
```

Data Output    Explain    Messages    Notifications

s_company_name	seller_id	s.first_name	s.last_name	s_mobile_no	count	max
AMUL	1000000	AKSHAR	MEHATA	9681454888	10	10

34°C Sunny 17:41 28-10-2021

Type here to search

