Group No :- 18

Vaghasiya Nikhil (i.d. :- 202003042)

Hardikkumar jani (i.d. :- 202003041)

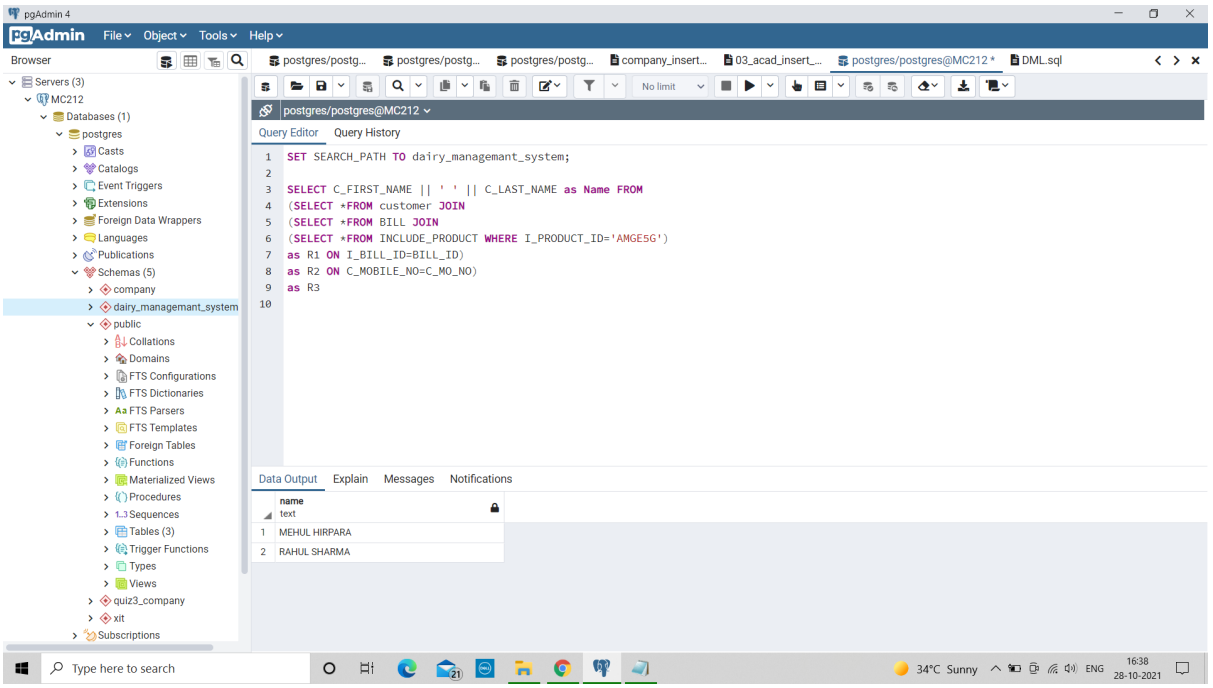
Lab No.:- 10(25-11-2021)

Dairy product management system

Queries:-

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

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



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

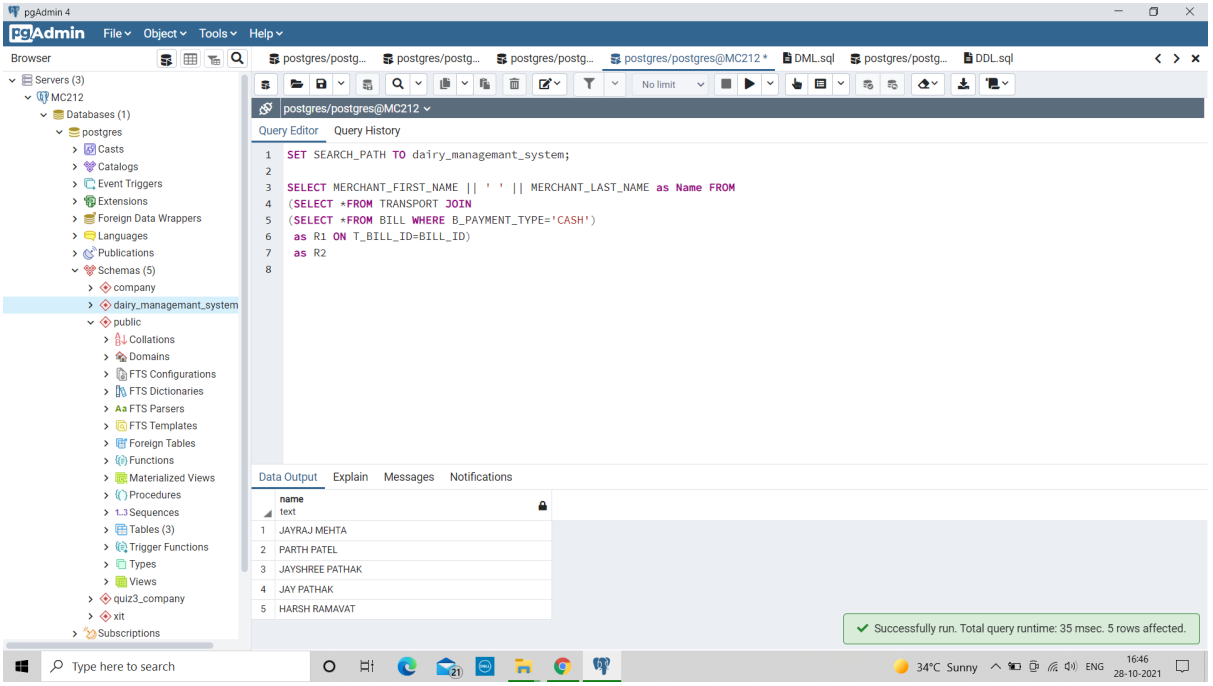
- Relational algebra:-

- $\Pi_{(MERCHANT\_FIRST\_NAME,MERCHANT\_LAST\_NAME)}\rho(R2,TRANSPORT \bowtie_{(T\_BILL\_ID=BILL\_ID)} \rho(R1,\sigma_{(B\_PAYMENT\_TYPE=CASH)} BILL))$

❖ 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')
 as R1 ON T_BILL_ID=BILL_ID)
 as R2
```

❖ Output:-



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

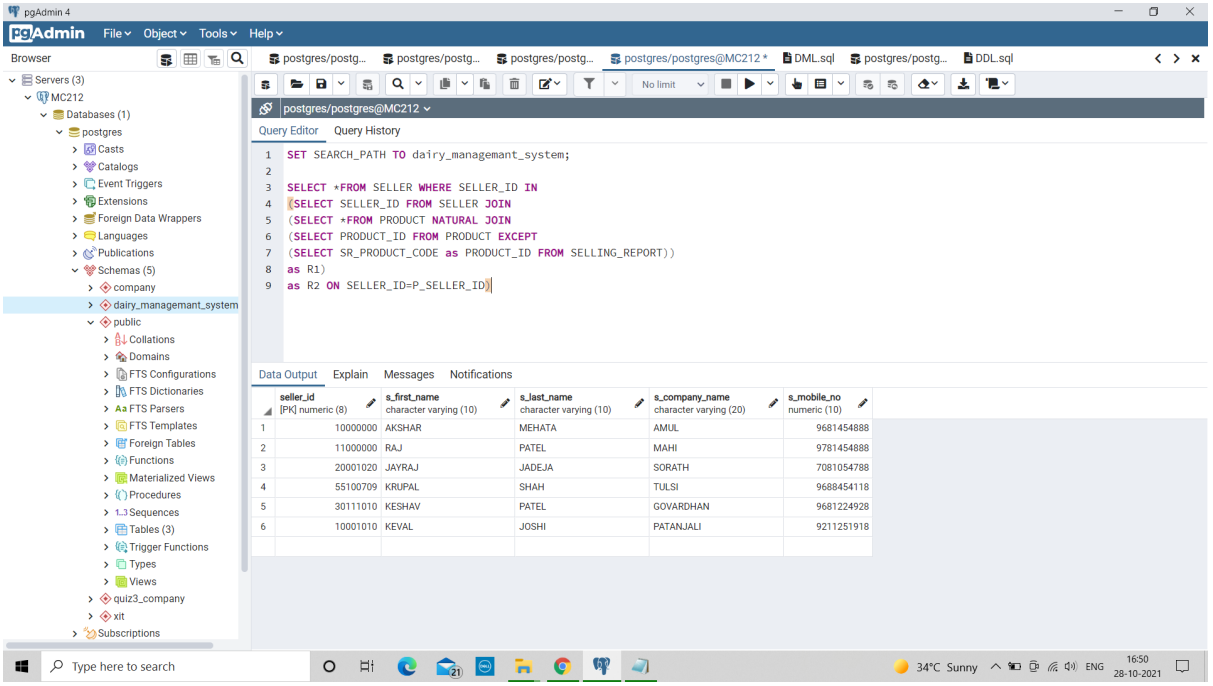
❖ Relational algebra:-

- $\Pi_{SELLER*}(SELLER \bowtie_{(SELLER\_ID=P\_SELLER\_ID)} \rho(R2,PRODUCT\ NATURAL\ JOIN\ \rho(R1,\Pi_{(PRODUCT\_ID)}PRODUCT - \Pi_{(SR\_PRODUCT\_CODE \rightarrow PRODUCT\_ID)}SELLING\_REPORT)))$

❖ SQL code:-

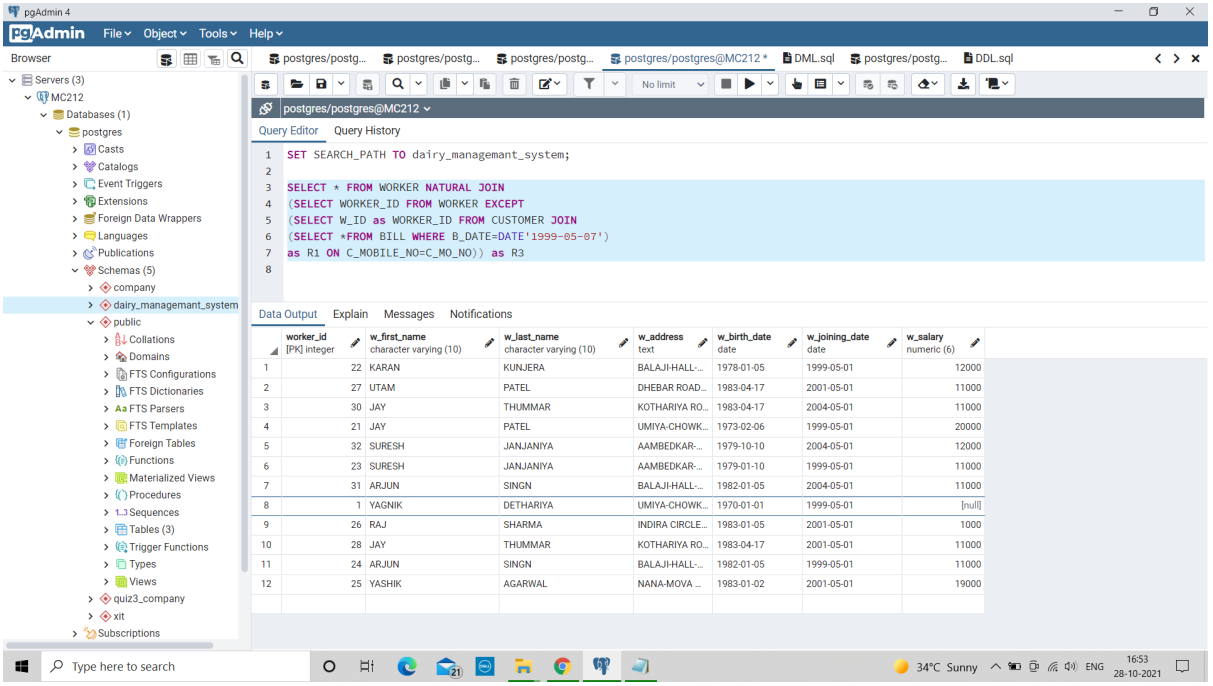
```
SELECT *FROM SELLER WHERE SELLER_ID IN
(SELECT SELLER_ID FROM SELLER JOIN
(SELECT *FROM PRODUCT NATURAL JOIN
(SELECT PRODUCT_ID FROM PRODUCT EXCEPT
(SELECT SR_PRODUCT_CODE as PRODUCT_ID FROM
SELLING_REPORT))
as R1)
as R2 ON SELLER_ID=P_SELLER_ID)
```

❖ Output:-



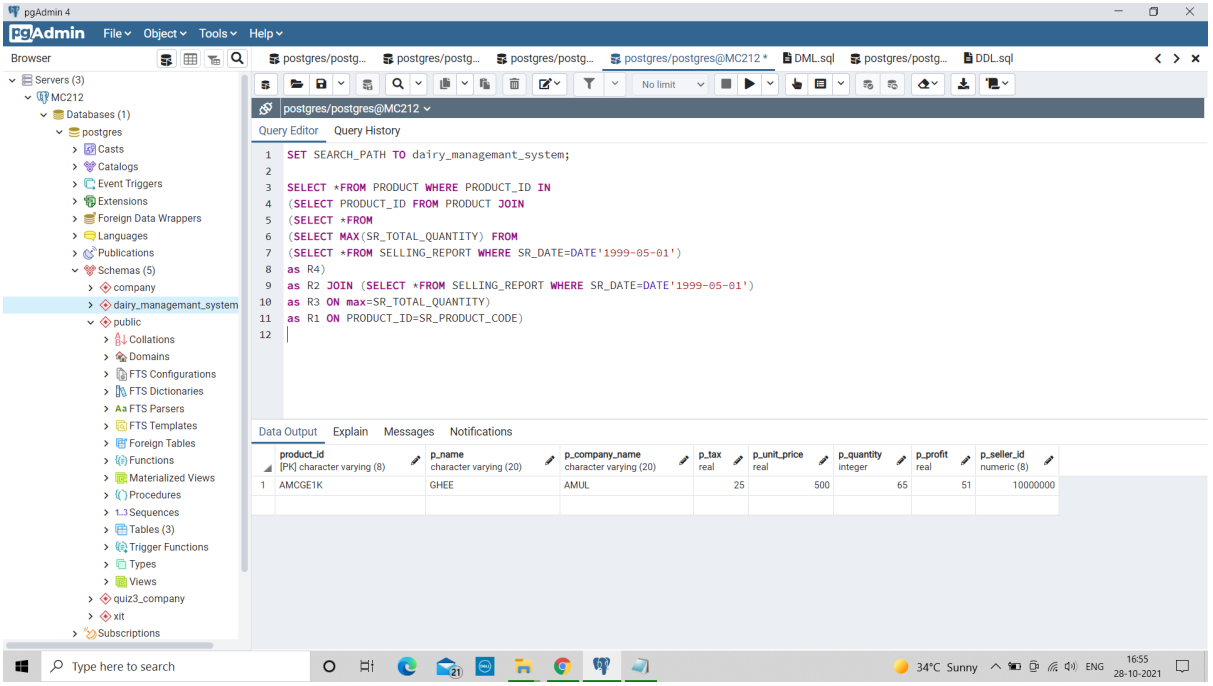
4. Find a worker who did not interact with a customer who bought a product today(DATE'1999-05-07') at the given outlet.

- Relational algebra:-
  - WORKER NATURAL JOIN $\rho(R3,\Pi_{(WORKER\_ID)}WORKER - \Pi_{(W\_ID \rightarrow WORKER\_ID)}\rho(R2,CUSTOMER \bowtie_{(C\_MOBILE\_NO=C\_MO\_NO)} \rho(R1,\sigma_{(B\_DATE=DATE'1999-05-07')}BILL)))$
- 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:-



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

- Relational algebra:-
  - $\Pi_{PRODUCT*}$(PRODUCT $\bowtie_{(PRODUCT\_ID=SR\_PRODUCT\_CODE)}$ ρ(R1,($\mathcal{F}_{MAX(SR\_TOTAL\_QUANTITY)}$ ($\sigma_{(SR_DATE=DATE'1999-05-01')}$SELLING_REPORT)) $\bowtie_{(max=SR\_TOTAL\_QUANTITY)}$ ($\sigma_{(SR_DATE=DATE'1999-05-01')}$SELLING_REPORT))
- 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:-



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

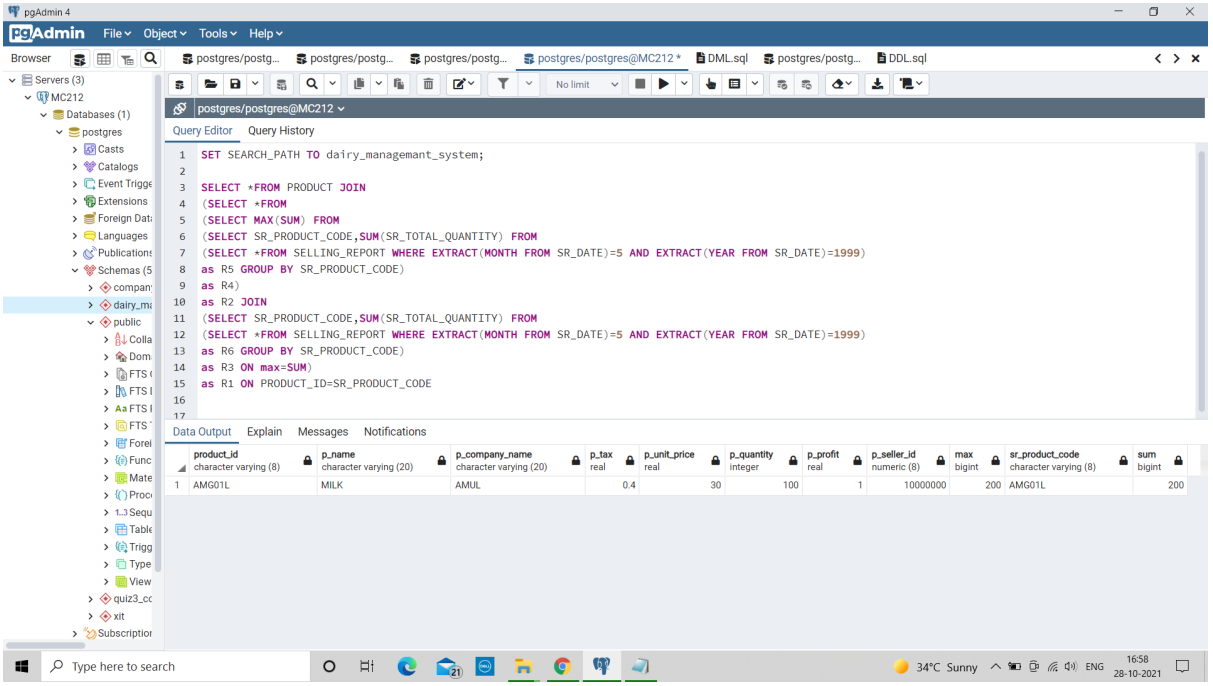
❖ Relational algebra:-

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

❖ SQL code:-

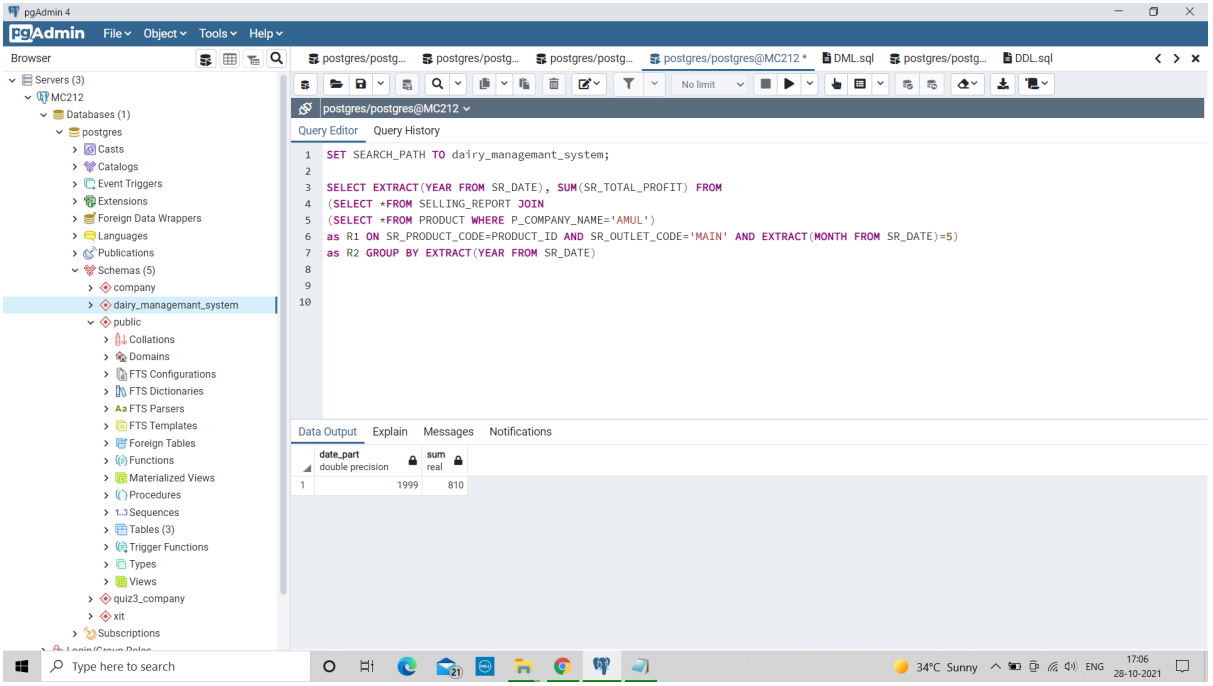
➢ SELECT *FROM PRODUCT JOIN
➢ (SELECT *FROM
➢ (SELECT MAX(SUM) FROM
➢ (SELECT SR_PRODUCT_CODE,SUM(SR_TOTAL_QUANTITY) FROM (SELECT *FROM SELLING_REPORT WHERE EXTRACT(MONTH FROM SR_DATE)=5 AND EXTRACT(YEAR FROM SR_DATE)=1999)
➢ as R5 GROUP BY SR_PRODUCT_CODE)
➢ as R4)
➢ as R2 JOIN (SELECT SR_PRODUCT_CODE,SUM(SR_TOTAL_QUANTITY) 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:-



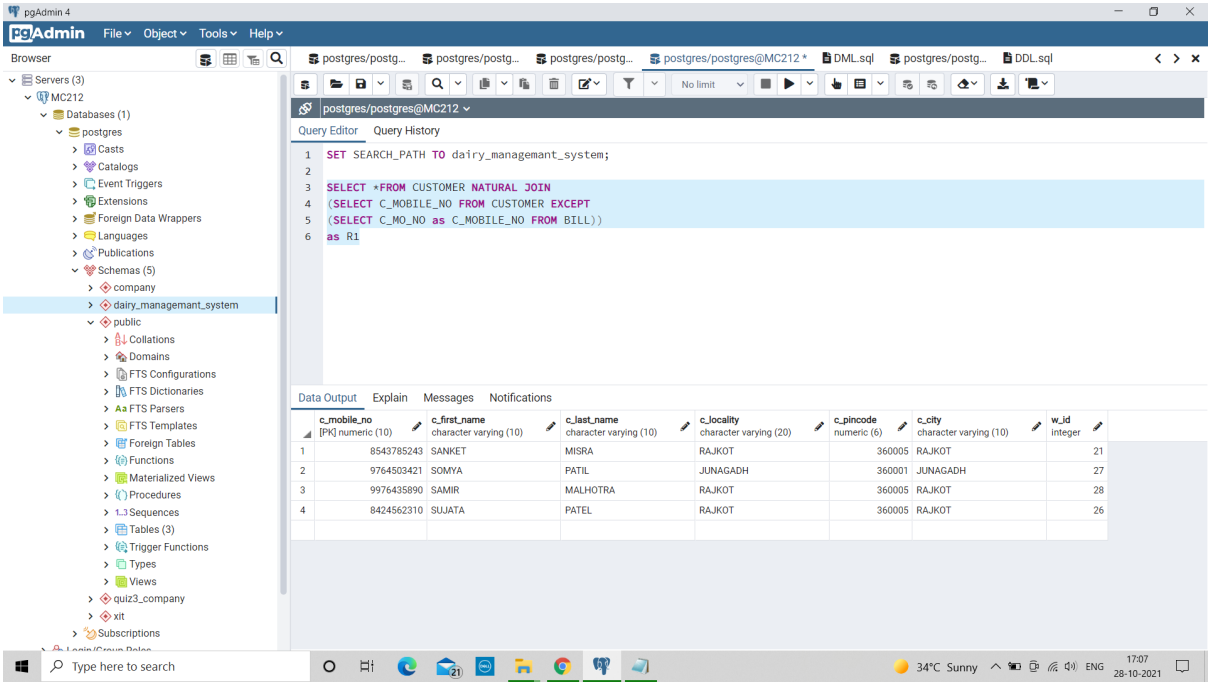
7. Give total profit for a given month(month=5) on a given outlet(OUTLET_CODE= ‘Main’) for a given company(name=’AMUL’).

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



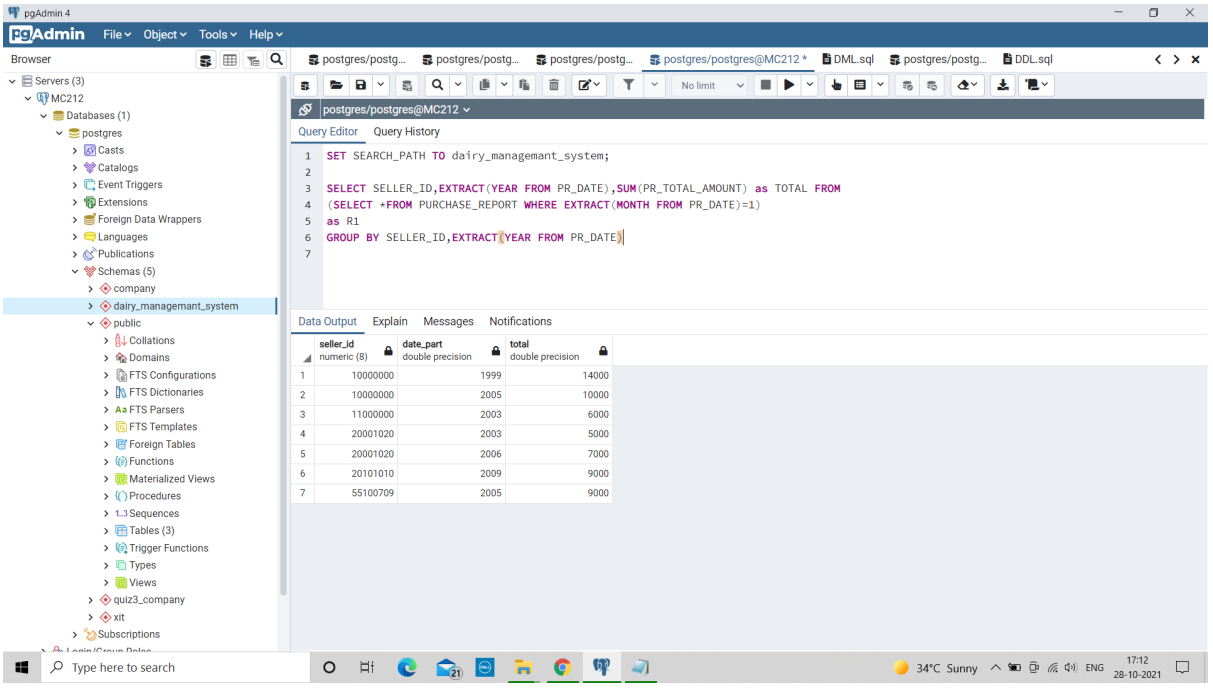
8. Give a customer list who has not to buy the product.

- 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:-



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

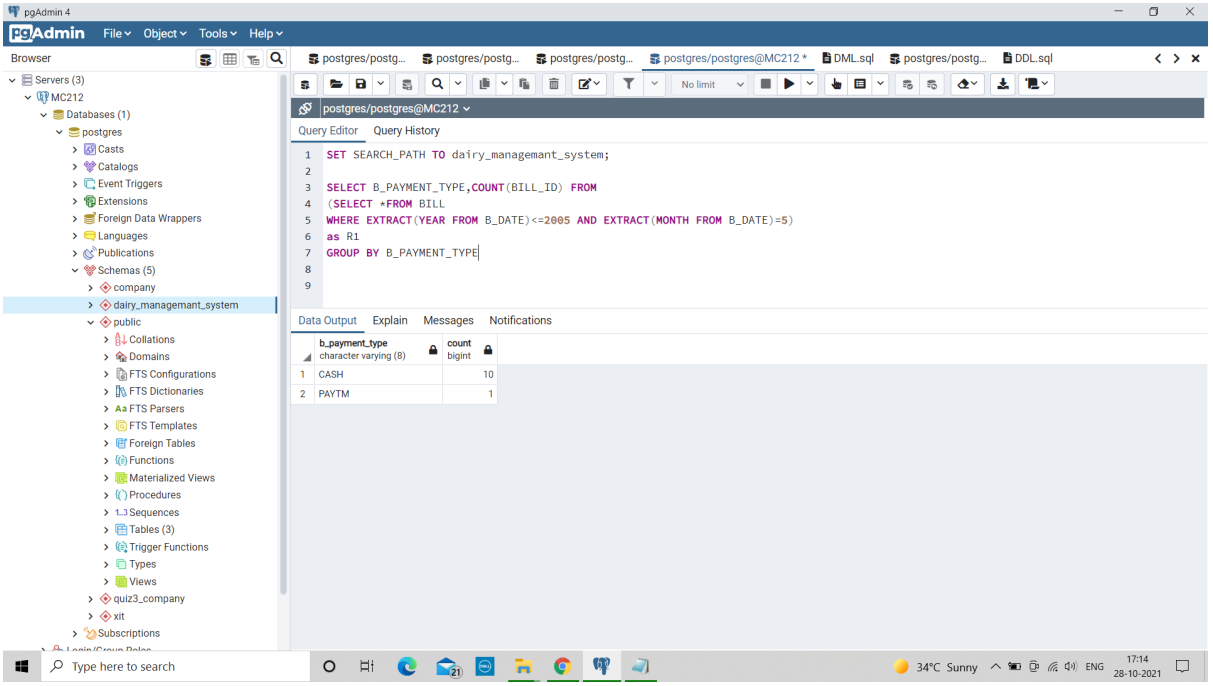
- Relational algebra:-
  - $_{\text{year(PR\_DATE),SELLER\_ID}}\mathcal{F}_{\text{(SUM(PR\_TOTAL\_AMOUNT)}\rightarrow\text{TOTAL)}}\rho(R1,\sigma_{\text{(month(PR\_DATE)=1)}}\text{PURCHASE\_REPORT})$
- SQL code:-
  - SELECT SELLER_ID,EXTRACT(YEAR FROM PR_DATE),SUM(PR_TOTAL_AMOUNT) as TOTAL FROM
  - (SELECT *FROM PURCHASE_REPORT WHERE EXTRACT(MONTH FROM PR_DATE)=1)
  - as R1
  - GROUP BY SELLER_ID,EXTRACT(YEAR FROM PR_DATE)
- Output:-



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

- Relational algebra:-
  - $_{\text{B\_PAYMENT\_TYPE}}\mathcal{F}_{\text{COUNT(BILL\_ID)}}\ \rho(R1,\sigma_{\text{(month(B\_DATE)=5 and (year(B\_DATE)<=2005)}}\text{BILL})$
- SQL code:-
  - SELECT B_PAYMENT_TYPE,COUNT(BILL_ID) FROM
  - (SELECT *FROM BILL
  - WHERE EXTRACT(YEAR FROM B_DATE)<=2005 AND EXTRACT(MONTH FROM B_DATE)=5)
  - as R1
  - GROUP BY B_PAYMENT_TYPE

❖ Output:-



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

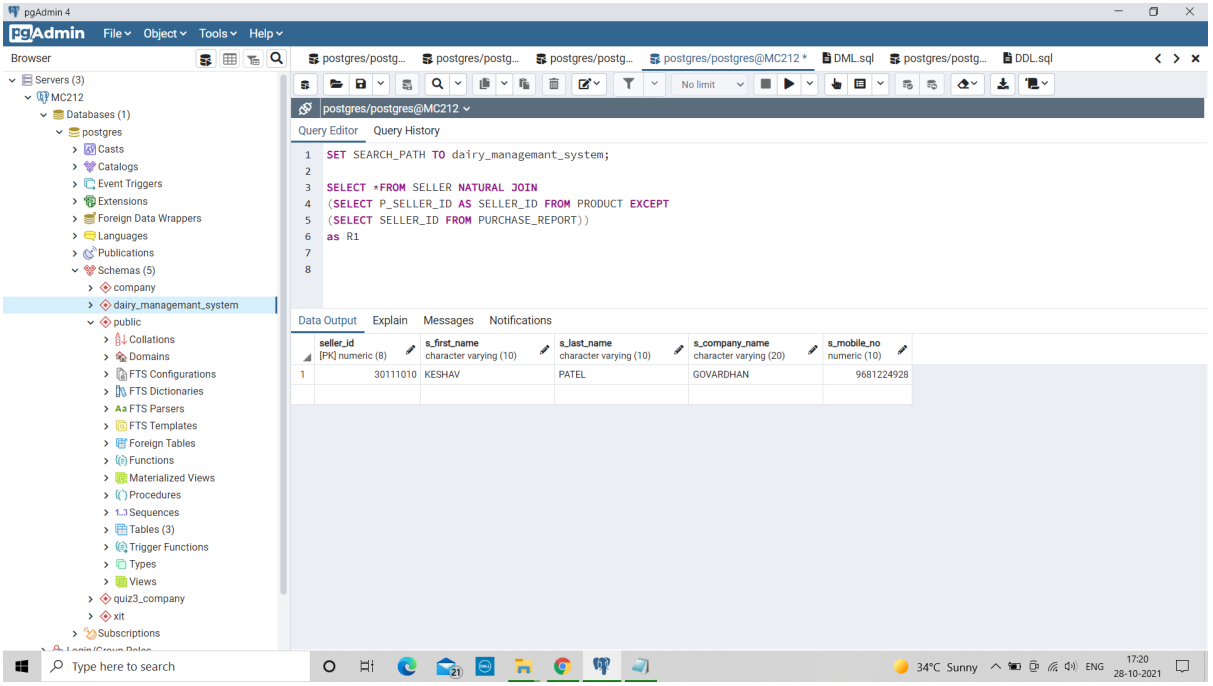
❖ Relational algebra:-

➢ SELLER NATURAL JOIN $\rho(R1,\prod_{(P\_SELLER\_ID \rightarrow SELLER\_ID)}PRODUCT - \prod_{(SELLER\_ID)}PURCHASE\_REPORT)$

❖ SQL code:-

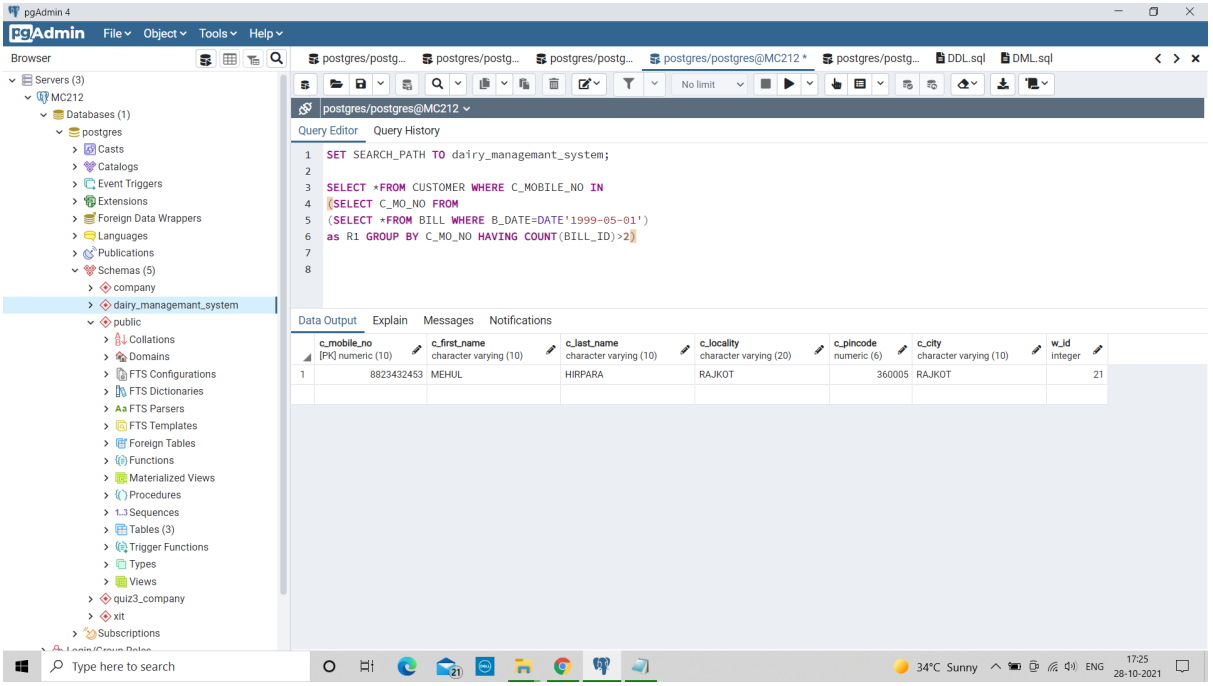
- SELECT *FROM SELLER NATURAL JOIN
- (SELECT P_SELLER_ID AS SELLER_ID FROM PRODUCT EXCEPT
- (SELECT SELLER_ID FROM PURCHASE_REPORT))
- as R1

❖ Output:-



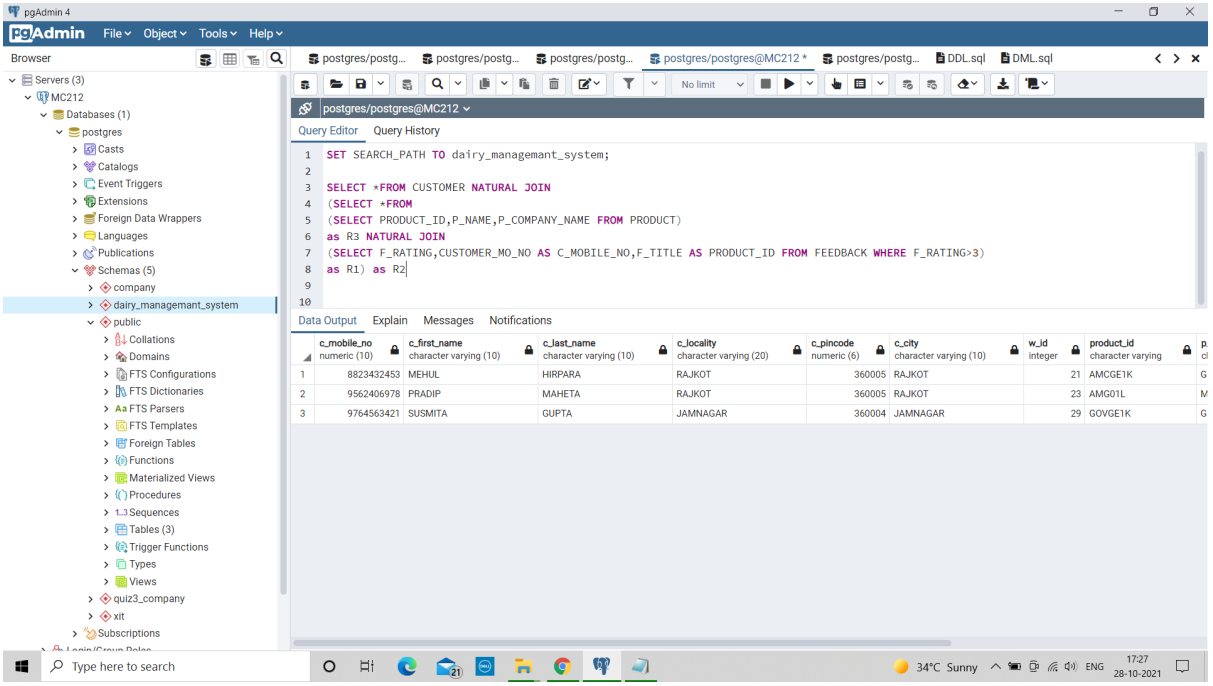
12. Find a customer who visits a dairy more than 2 times on a given day(DATE'1999-05-01').

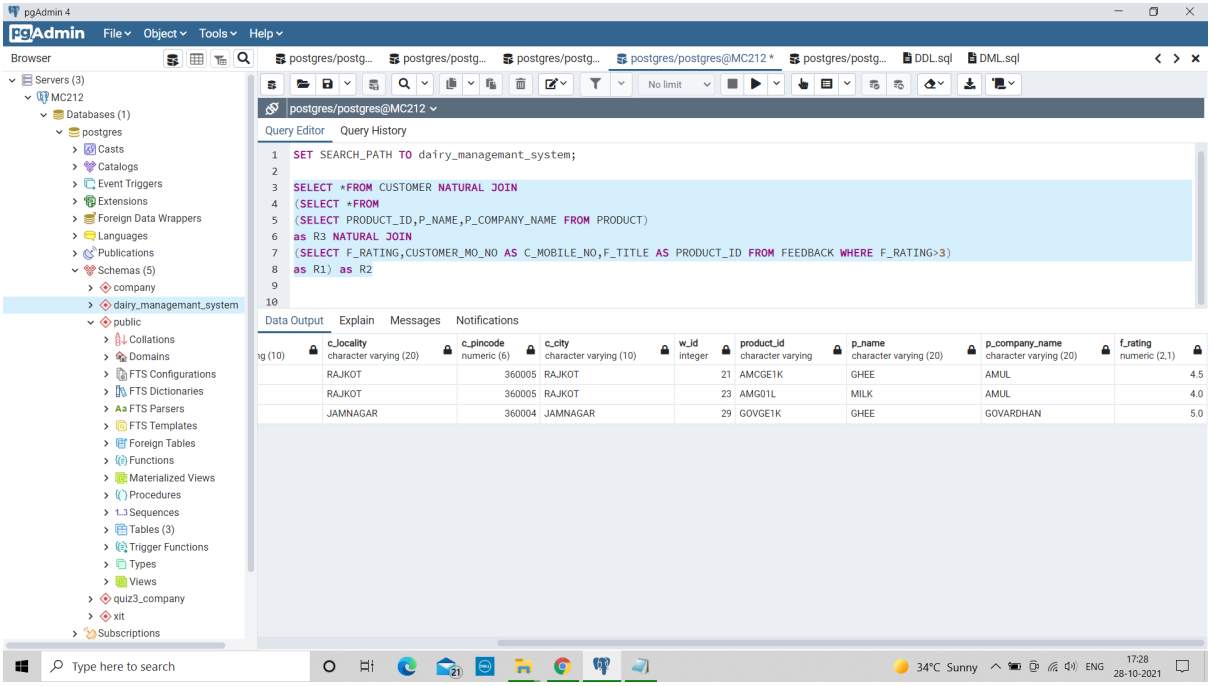
- Relational algebra:-
  - $\Pi_{(CUSTOMER*)}$(CUSTOMER SEMI-JOIN$_{(C_MOBILE_NO=C_MO_NO)}$ ($\sigma_{(COUNT(BILL)>2)}$($_{(C\_MO\_NO)}\mathcal{F}_{(COUNT(BILL\_ID))}$ ρ(R1,$\sigma_{(B_DATE=DATE'1999-05-01')}$BILL))))
- SQL code:-
  - SELECT *FROM CUSTOMER WHERE C_MOBILE_NO IN
  - (SELECT C_MO_NO FROM
  - (SELECT *FROM BILL WHERE B_DATE=DATE'1999-05-01')
  - as R1 GROUP BY C_MO_NO HAVING COUNT(BILL_ID)>2)
- Output:-



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:-



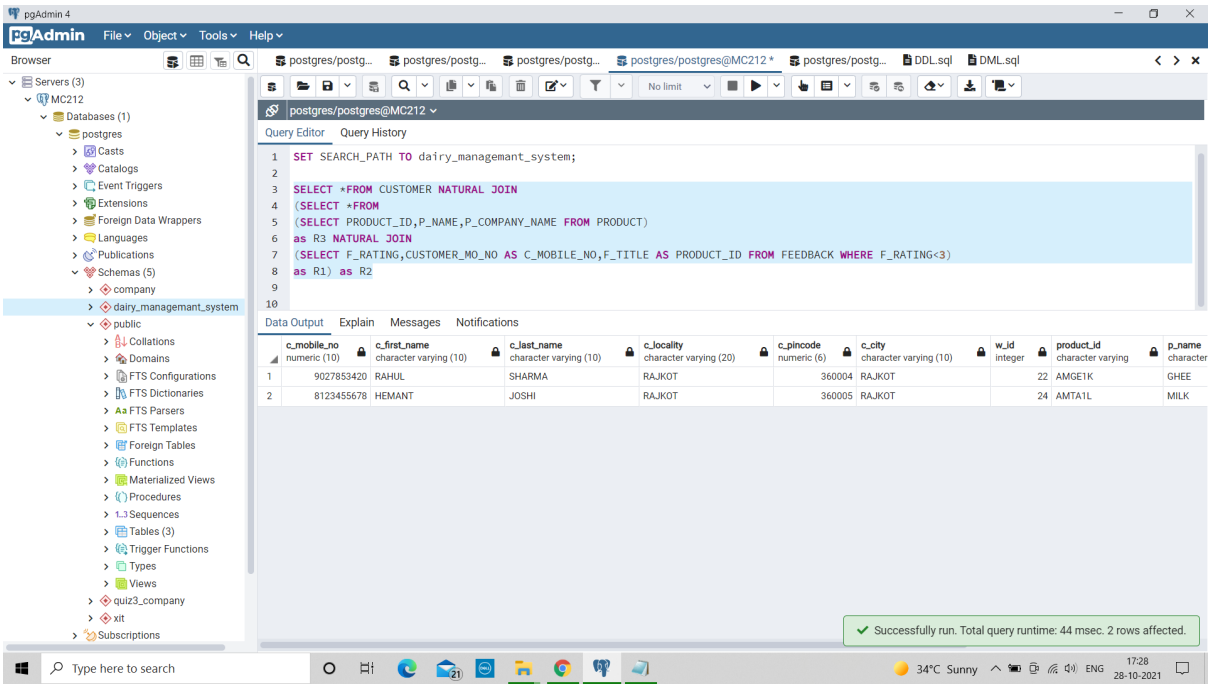


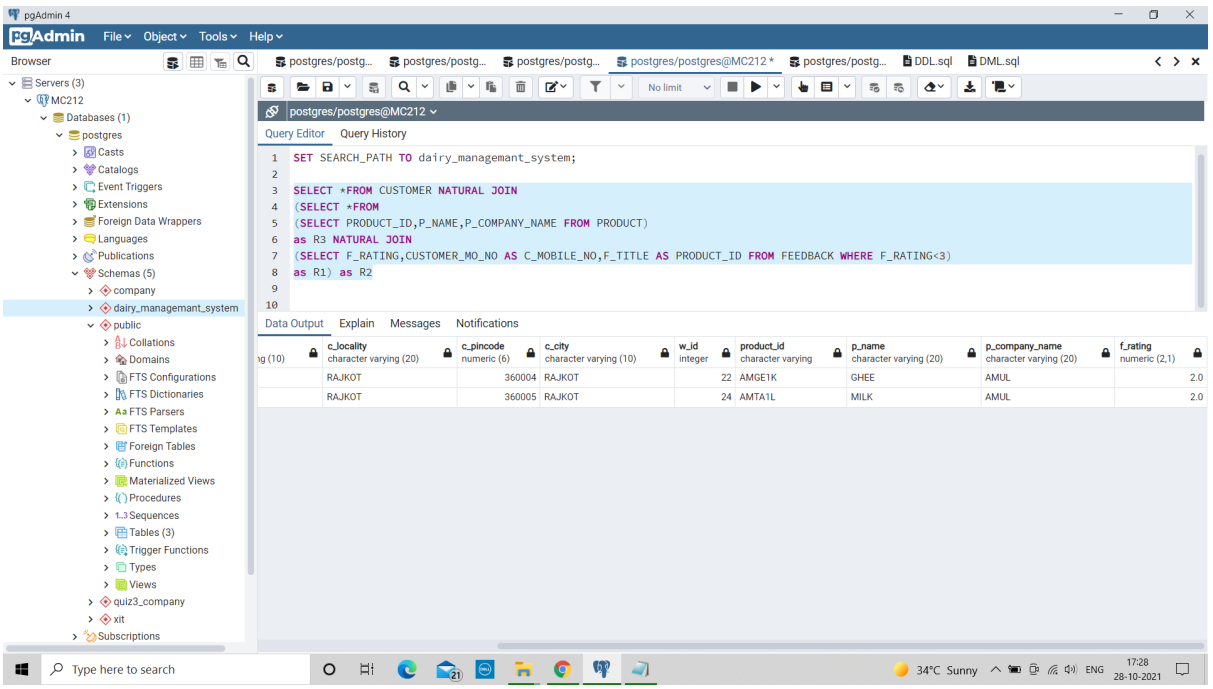
14. Find details of customers who give us bad(less than 3) ratings and list that products.

- 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

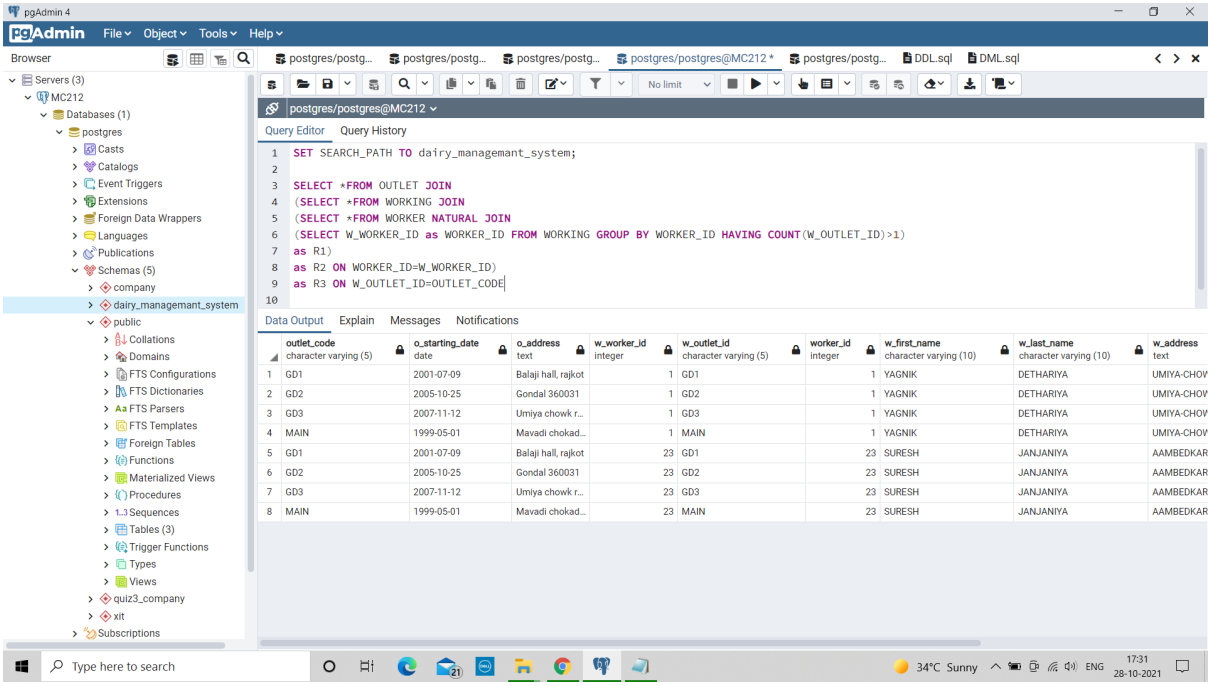
❖ Output:-

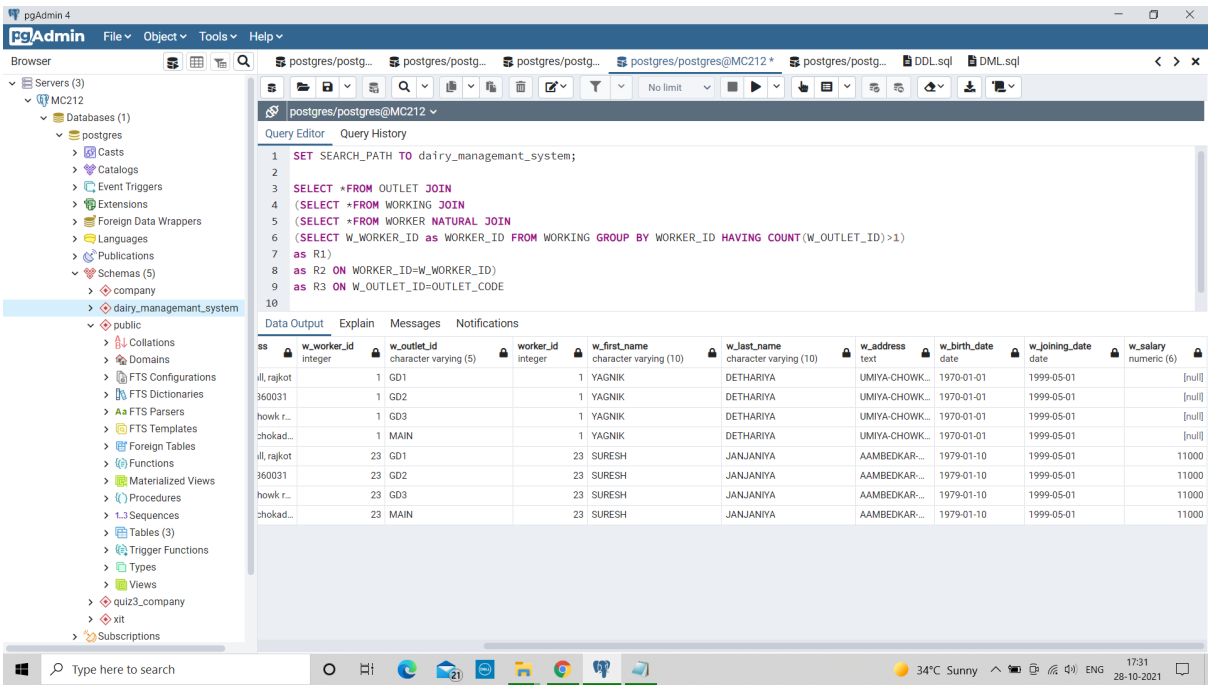




15. Find workers who work on more than 1 outlet and print their outlets.

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



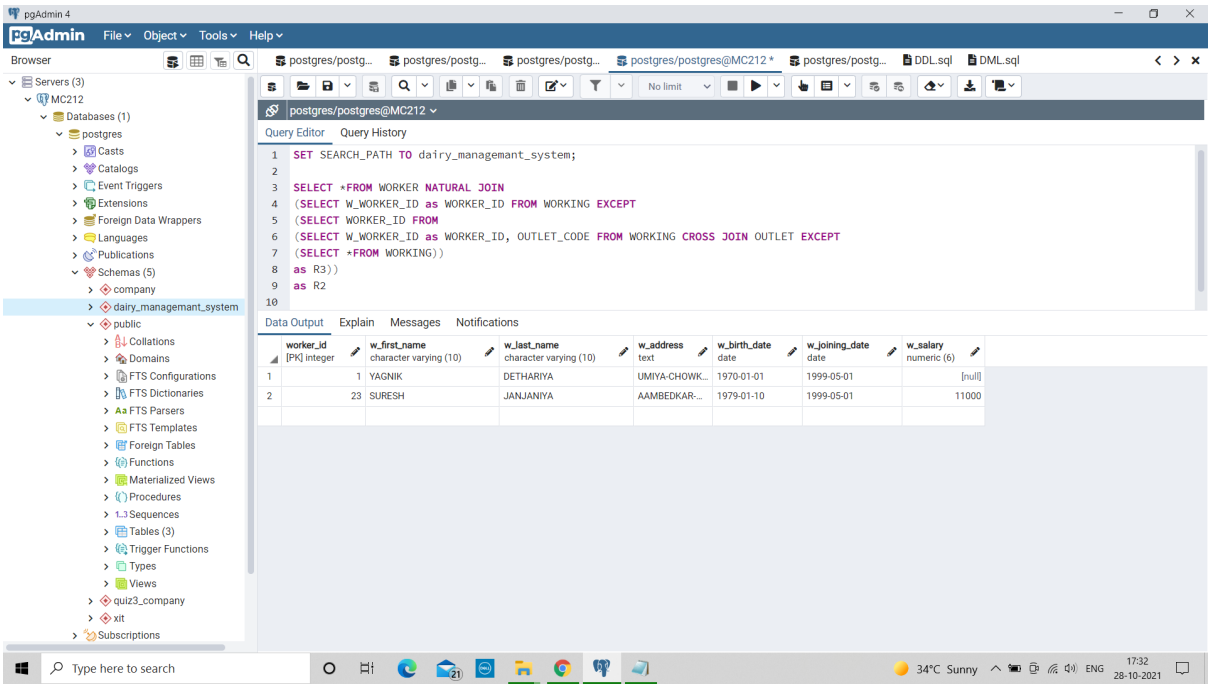


16. Find a worker who works on all outlets.(DIVISION QUERY)

- Relational algebra:-
  - WORKER NATURAL JOIN $\rho(R2,\prod_{(W\_WORKER\_ID \rightarrow WORKER\_ID)}WORKING - \prod_{(WORKER\_ID)}(R3,\prod_{(W\_WORKER\_ID \rightarrow WORKER\_ID,OUTLET\_CODE)}(WORKING \times OUTLET) - WORKING))$
- 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:-



17.Find transports which do not include any products.

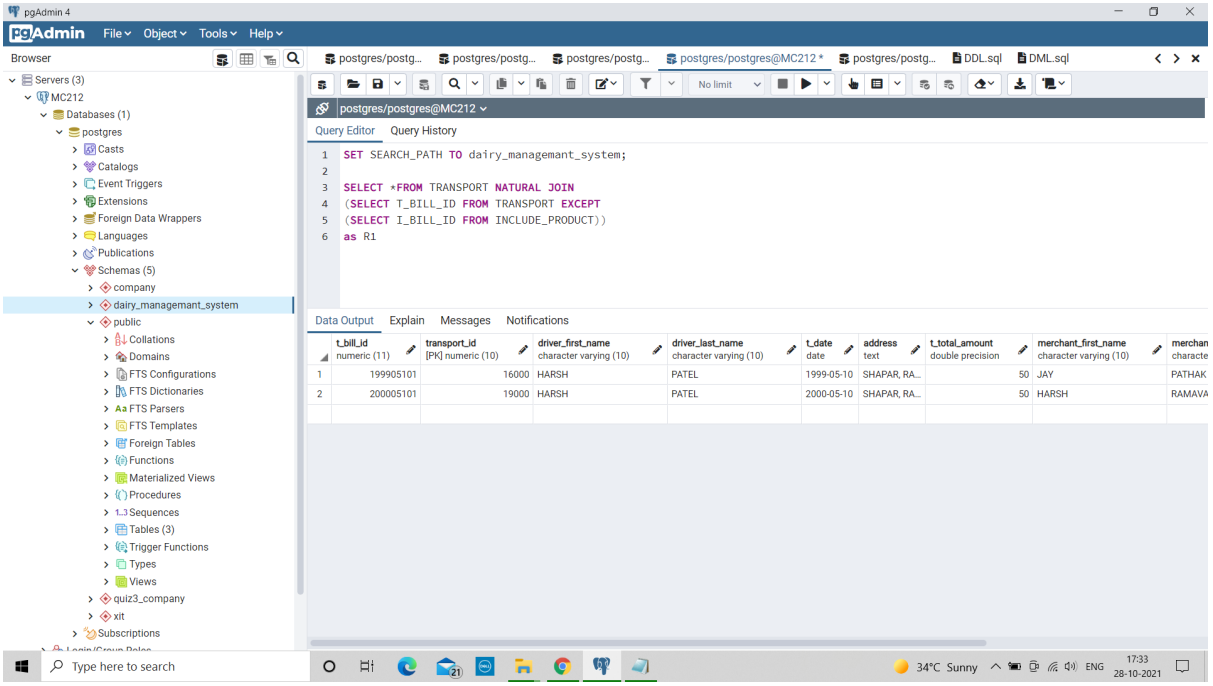
❖ Relational algebra:-

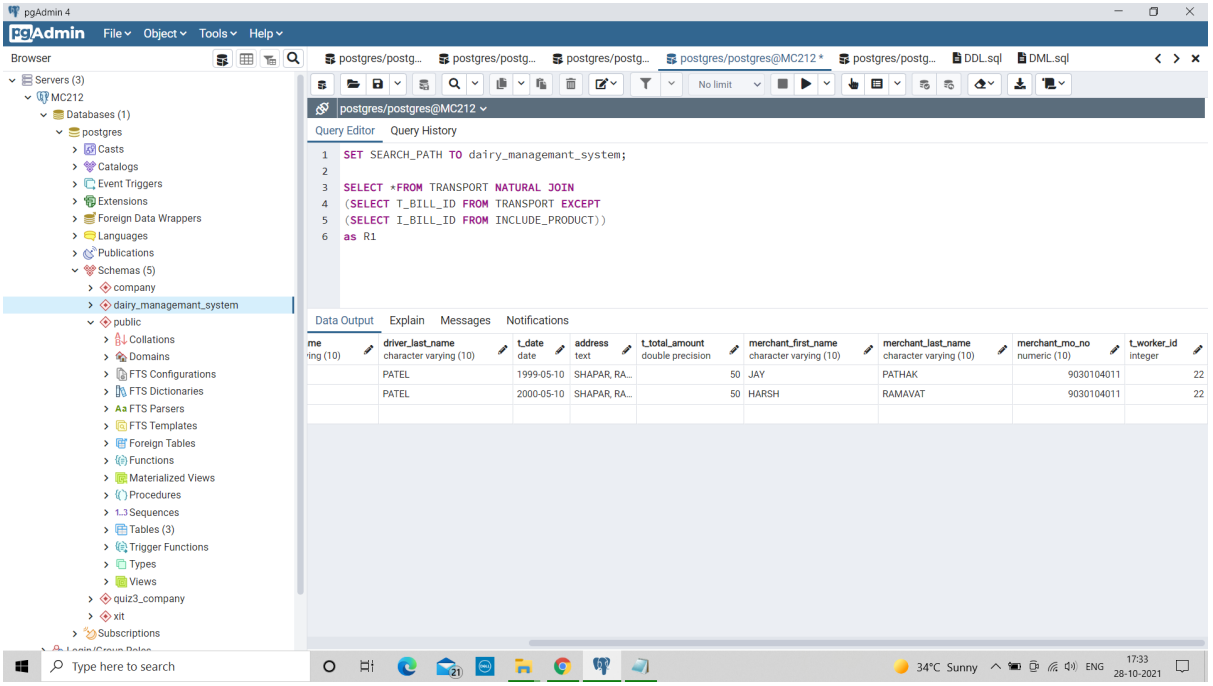
- ➢ TRANSPORT NATURAL JOIN ρ(R1,$\prod_{(T_BILL_ID)}$TRANSPORT - $\prod_{(I_BILL_ID)}$INCLUDE_PRODUCT)

❖ SQL code:-

- ➢ SELECT *FROM TRANSPORT NATURAL JOIN
- ➢ (SELECT T_BILL_ID FROM TRANSPORT EXCEPT
- ➢ (SELECT I_BILL_ID FROM INCLUDE_PRODUCT))
- ➢ as R1

❖ Output:-





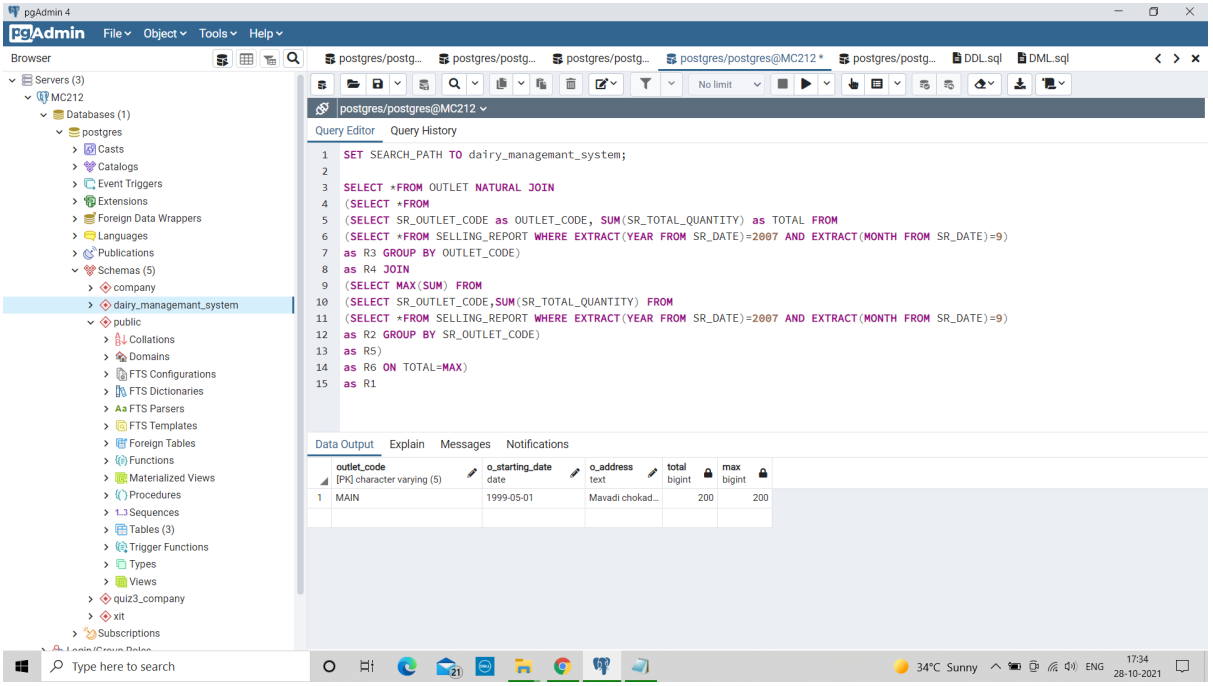
18. Which outlet sold the maximum product on a given month(9) and year(2007).

- Relational algebra:-
  - OUTLET NATURAL JOIN

    $\rho(R1,\rho(R4,_{(SR\_OUTLET\_CODE \rightarrow OUTLET\_CODE)}\mathcal{F}_{(SUM(SR\_TOTAL\_QUANTITY) \rightarrow total)}$
    $\rho(R3,\sigma_{(YEAR(SR\_DATE)=2007\ AND\ MONTH(SR\_DATE)=9}\ SELLING\_REPORT)) \bowtie_{(total=MAX)}$
    $\rho(R6,\mathcal{F}_{(MAX(SUM))}\ \rho(R5,_{(SR\_OUTLET\_CODE)}\mathcal{F}_{(SUM(SR\_TOTAL\_QUANTITY))}$
    $\rho(R2,\sigma_{(YEAR(SR\_DATE)=2007\ AND\ MONTH(SR\_DATE)=9}\ SELLING\_REPORT))))$
- SQL code:-

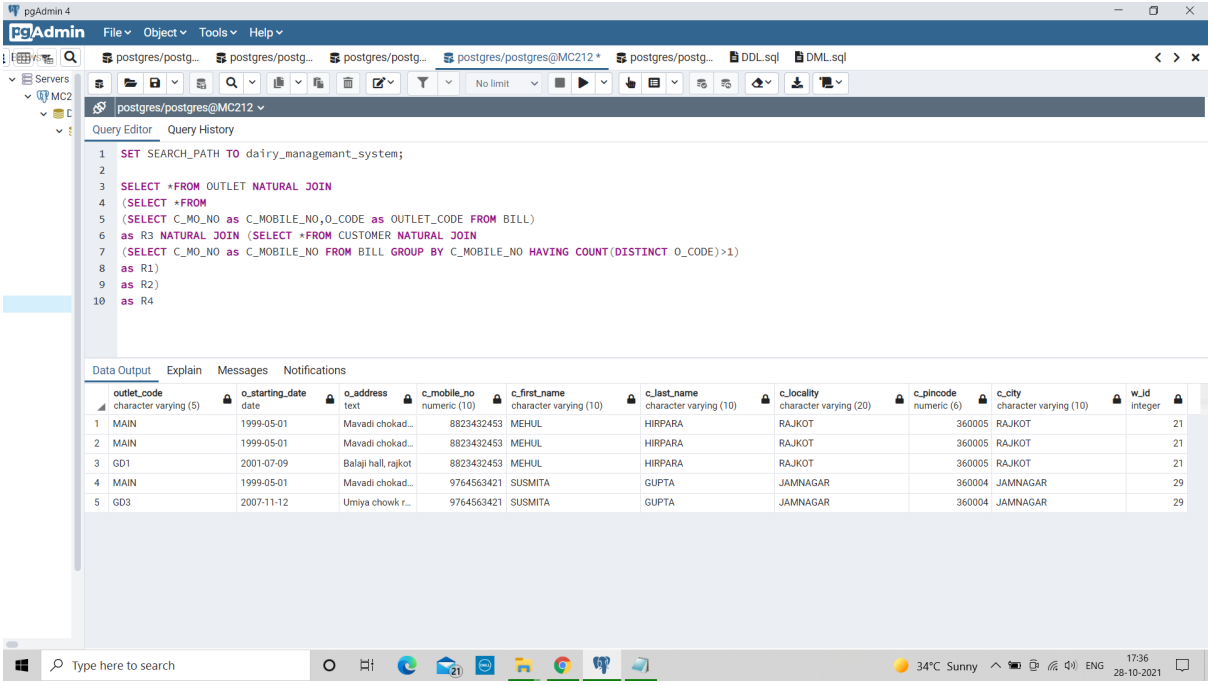
- SELECT *FROM OUTLET NATURAL JOIN
- (SELECT *FROM
- (SELECT SR_OUTLET_CODE as OUTLET_CODE, SUM(SR_TOTAL_QUANTITY) as TOTAL FROM
- (SELECT *FROM SELLING_REPORT WHERE EXTRACT(YEAR FROM SR_DATE)=2007 AND EXTRACT(MONTH FROM SR_DATE)=9)
- as R3 GROUP BY OUTLET_CODE)
- as R4 JOIN
- (SELECT MAX(SUM) FROM
- (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:-



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

- Relational algebra:-
  - OUTLET NATURAL JOIN
    $(R4,(R3,\Pi_{(C\_MO\_NO \rightarrow C\_MOBILE\_NO, O\_CODE \rightarrow OUTLET\_CODE)}BILL)$ NATURAL JOIN
    (R2,CUSTOMER NATURAL JOIN
    $(R1,\sigma_{(COUNT>1)}(_{(C\_MO\_NO \rightarrow C\_MOBILE\_NO)}\mathcal{F}_{COUNT(DISTINCT\ O\_CODE)}\ BILL))))$
- 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:-



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

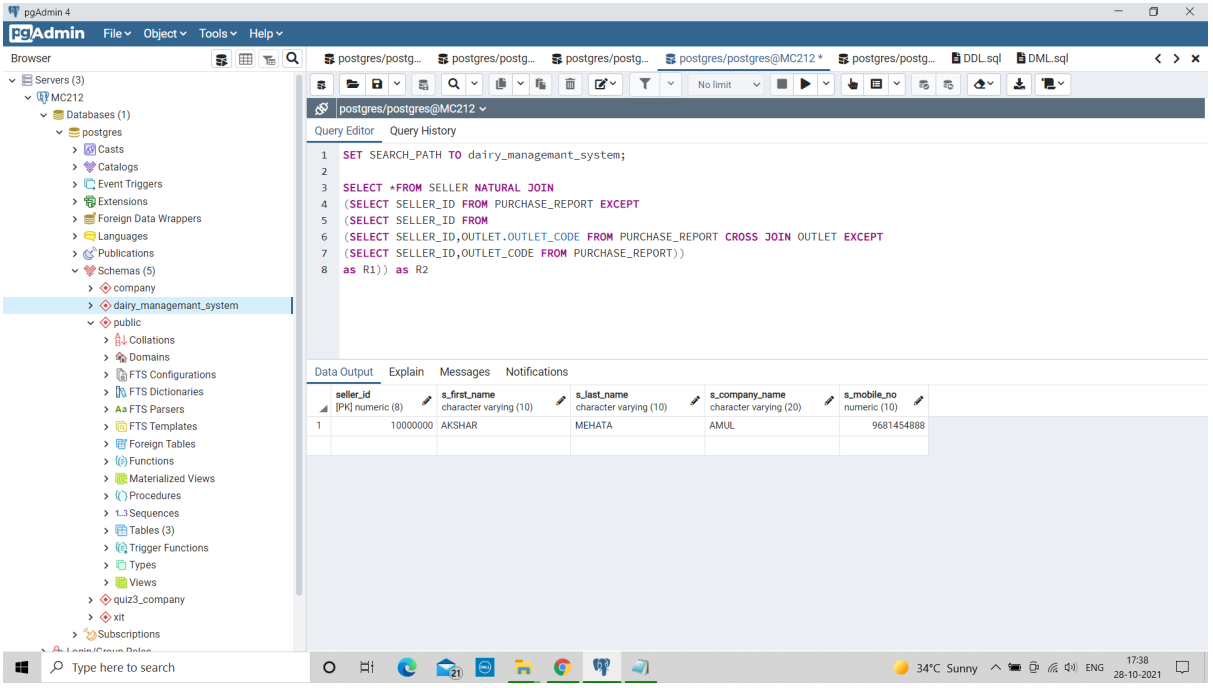
- Relational algebra:-

- ➢ SELLER NATURAL JOIN $\rho(R2,\Pi_{(SELLER\_ID)}PURCHASE\_REPORT - \Pi_{(SELLER\_ID)}\rho(R1,\Pi_{(SELLER\_ID,OUTLET\_CODE)}(PURCHASE\_REPORT\ X\ OUTLET) - \Pi_{(SELLER\_ID,OUTLET\_CODE)}PURCHASE\_REPORT))$

❖ 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:-



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

❖ Relational algebra:-

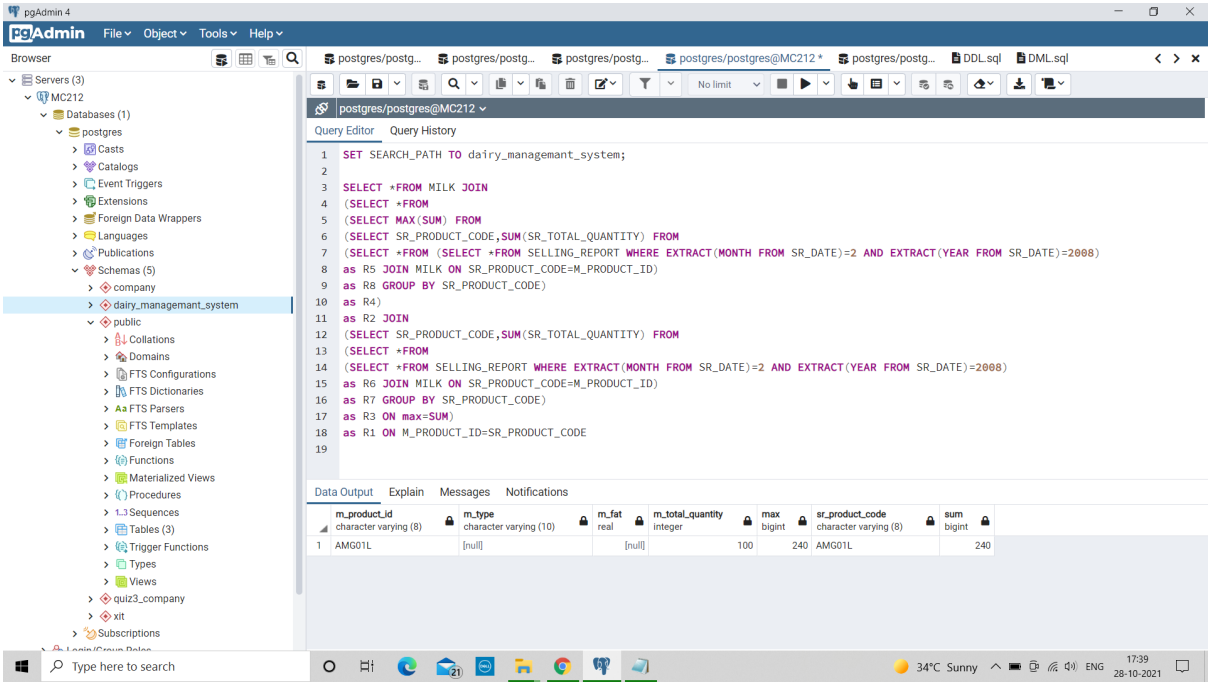
- ➢ MILK $\bowtie_{(M\_PRODUCT\_ID=SR\_PRODUCT\_CODE)} \rho(R1,\rho(R2,\mathcal{F}_{MAX(sum)}\rho(R4,_{SR\_PRODUCT\_CODE}\mathcal{F}_{SUM(SR\_TOTAL\_QUANTITY)} \rho(R8,\rho(R5,\sigma_{(month(SR\_DATE)=5\ and\ year(SR\_DATE)=1999)}SELLING\_REPORT) \bowtie_{(SR\_PRODUCT\_CODE=M\_PRODUCT\_ID)} MILK))) \bowtie_{(max=sum)}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

❖ 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:-



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

❖ Relational algebra:-

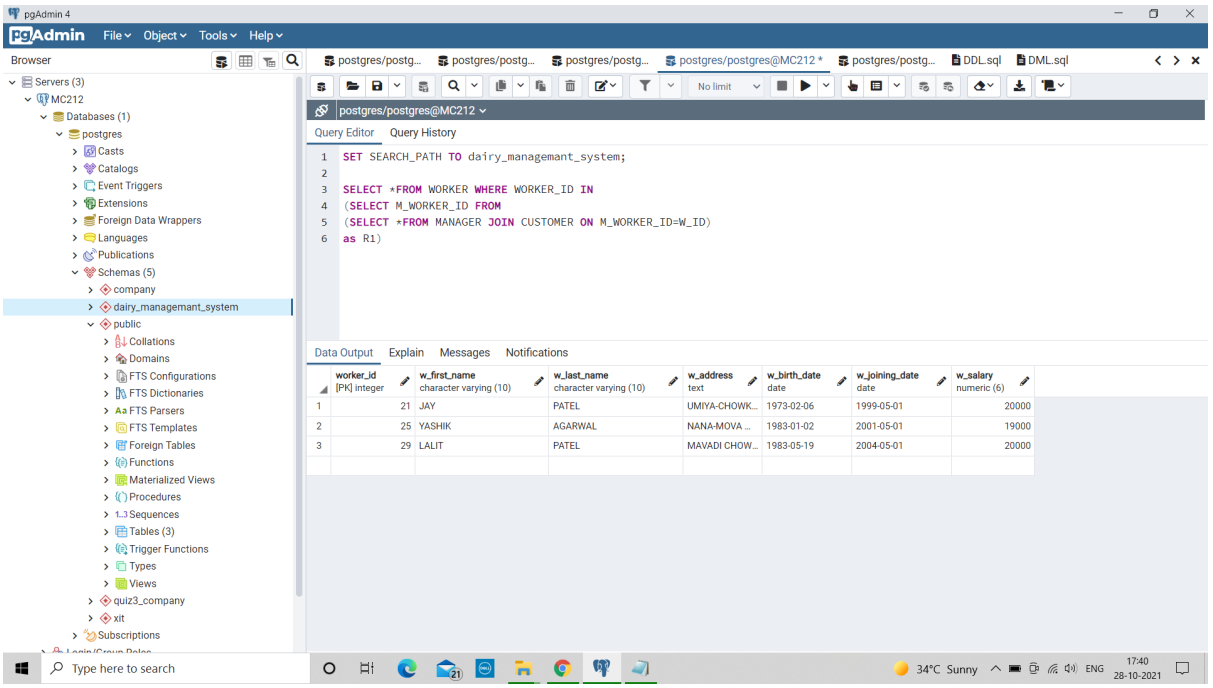
- $\Pi_{(WORKER*)}$ WORKER $\bowtie_{(WORKER\_ID=M\_WORKER\_ID)}$ ρ(R1,MANAGER $\bowtie_{(M\_WORKER\_ID=W\_ID)}$ CUSTOMER)

❖ 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:-



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

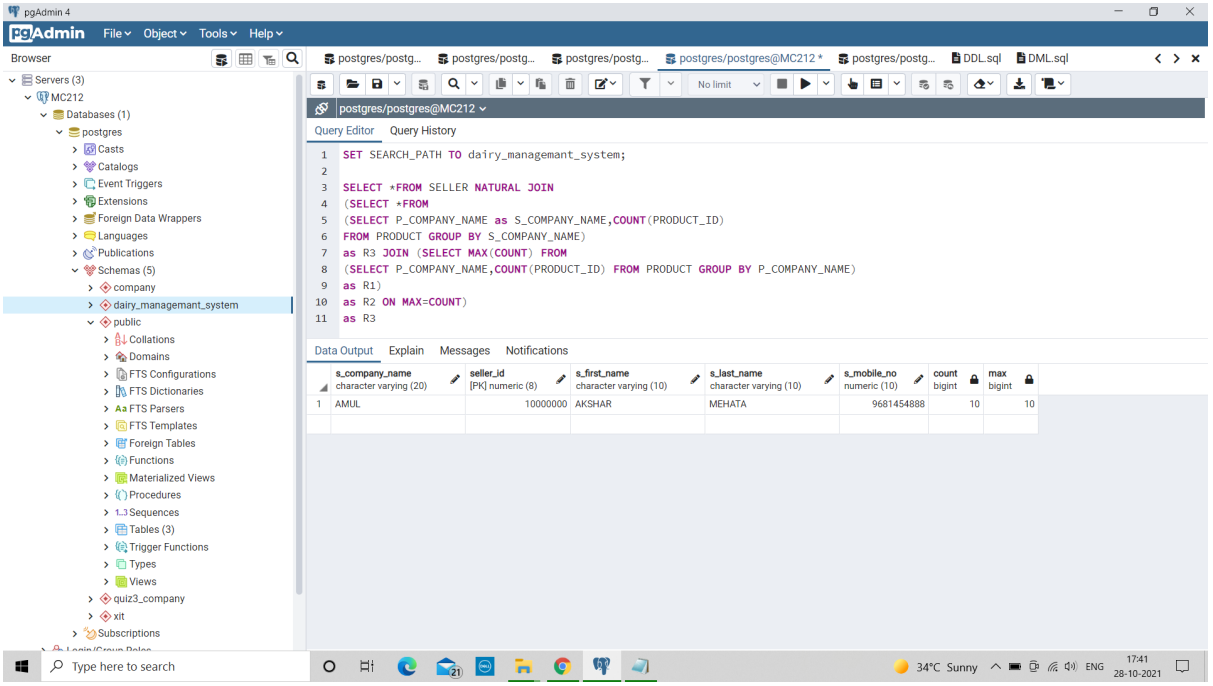
❖ Relational algebra:-

- ➢ SELLER NATURAL JOIN $\rho(R3,_{P\_COMPANY\_NAME \rightarrow S\_COMPANY\_NAME}\mathcal{F}_{COUNT(PRODUCT\_ID)}\ PRODUCT) \bowtie_{(MAX=COUNT)} \rho(R2,\mathcal{F}_{MAX(COUNT)}\ \rho(R1,_{P\_COMPANY\_NAME}\mathcal{F}_{COUNT(PRODUCT\_ID)}\ PRODUCT))$

❖ SQL code:-

- ➢ SELECT *FROM SELLER NATURAL JOIN
- ➢ (SELECT *FROM
- ➢ (SELECT P_COMPANY_NAME as S_COMPANY_NAME,COUNT(PRODUCT_ID)
- ➢ FROM PRODUCT GROUP BY S_COMPANY_NAME)
- ➢ as R3 JOIN (SELECT MAX(COUNT) FROM
- ➢ (SELECT P_COMPANY_NAME,COUNT(PRODUCT_ID) FROM PRODUCT GROUP BY P_COMPANY_NAME)
- ➢ as R1)
- ➢ as R2 ON MAX=COUNT)
- ➢ as R3

❖ Output:-


pgAdmin 4
File Object Tools Help
Browser
Servers (3)
MC212
Databases (1)
postgres
Casts
Catalogs
Event Triggers
Extensions
Foreign Data Wrappers
Languages
Publications
Schemas (5)
company
dairy_managemant_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
postgres/postgres@MC212 *
DDL.sql
DML.sql
No limit
Query Editor
Query History
SET SEARCH_PATH TO dairy_managemant_system;
SELECT *FROM SELLER NATURAL JOIN
(SELECT *FROM
(SELECT P_COMPANY_NAME as S_COMPANY_NAME,COUNT(PRODUCT_ID)
FROM PRODUCT GROUP BY S_COMPANY_NAME)
as R3 JOIN (SELECT MAX(COUNT) FROM
(SELECT P_COMPANY_NAME,COUNT(PRODUCT_ID) FROM PRODUCT GROUP BY P_COMPANY_NAME)
as R1)
as R2 ON MAX=COUNT)
as R3
Data Output
Explain
Messages
Notifications
s_company_name character varying (20)
seller_id [PK] numeric (8)
s_first_name character varying (10)
s_last_name character varying (10)
s_mobile_no numeric (10)
count bigint
max bigint
1 AMUL 10000000 AKSHAR MEHATA 9681454888 10 10
Type here to search
34°C Sunny
ENG
17:41
28-10-2021