Credit Card Transactions

- Project Prepared By :- Gaurav kumar

Overview:

Given dataset contains 4 files which provides information related to the type of card customers have, details about the credit card, transactions which are marked as fraudulent and the transactions done by each customer. Use this information to better understand the credit card transactions and solve given problems.

Functions and concepts are used while solving a problem in PostgreSQL :-

- FUNCTION :- count() , distinct(), min(), max(), round(), avg(), sum()
- Union all, Subqueries, JOIN, Group by, to_char
- Window function :- first_value(), last_value(), row_number()
- Used “CASE STATEMENT” to converting **row level data to column level data**.

Problem Statements:

1 :- How many customers have done transactions over 49000?

2:- What kind of customers can get a Premium credit card?

3:- Identify the range of credit limit of customer who have done fraudulent transactions.

4:- What is the average age of customers who are involved in fraud transactions based on different card type?

5:- Identify the month when highest no of fraudulent transactions occurred

6:- Identify the customer who has done the most transaction value without involving in any fraudulent transactions.

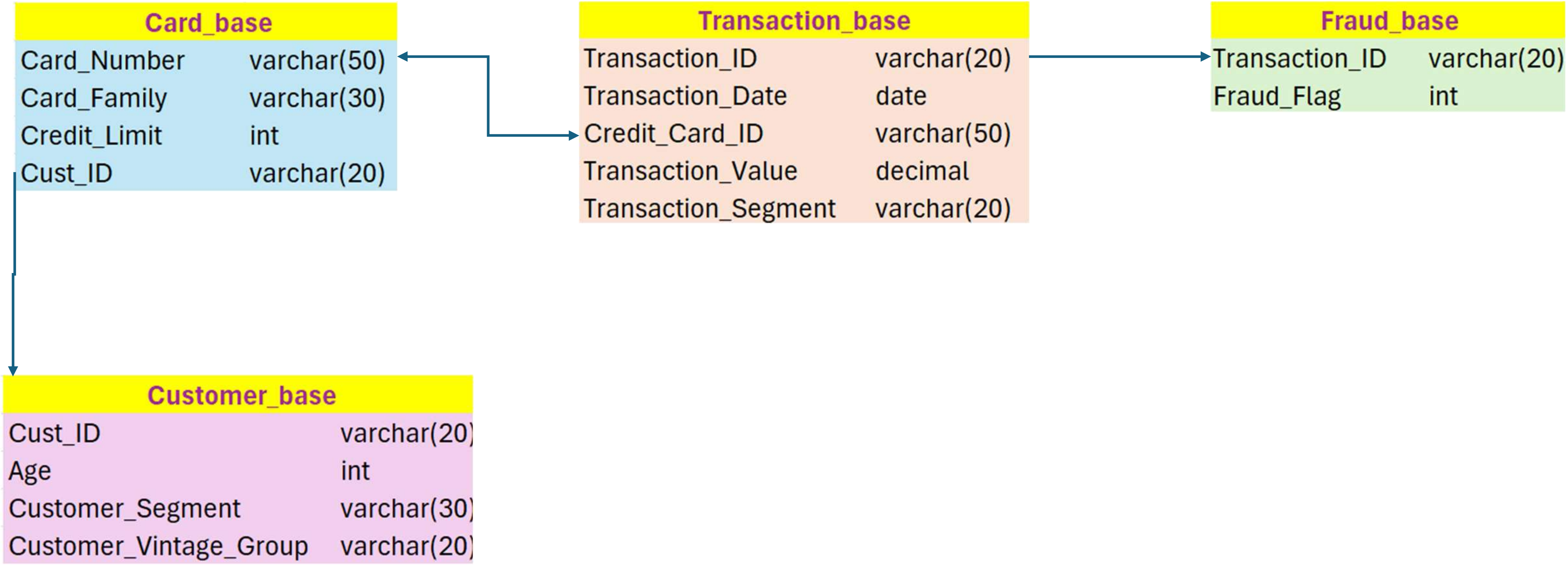
7:-Check and return any customers who have not done a single transaction.

8:-What is the highest and lowest credit limit given to each card type?

9:- What is the total value of transactions done by customers who come under the age bracket of 20-30 yrs, 30-40yrs, 40-50 yrs, 50+ yrs and 0-20 yrs.

10:- Which card type has done the most no of transactions and the total highest value of transactions without having any fraudulent transactions.

Lists of Table



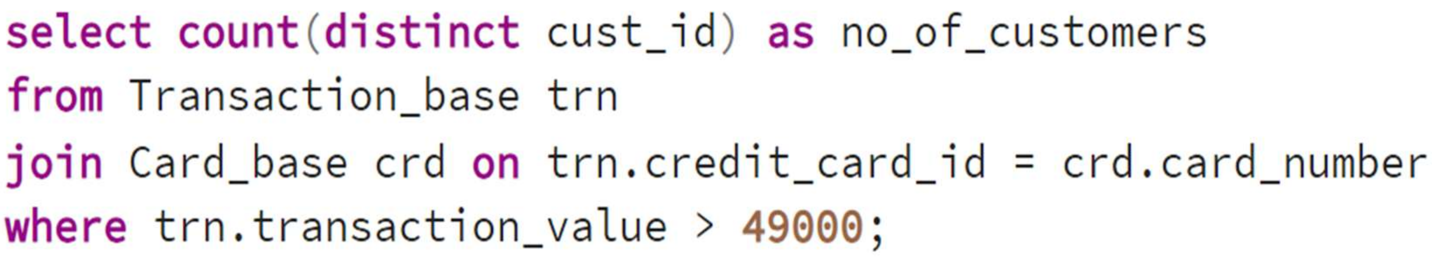
Card_base	
Card_Number	varchar(50)
Card_Family	varchar(30)
Credit_Limit	int
Cust_ID	varchar(20)

Transaction_base	
Transaction_ID	varchar(20)
Transaction_Date	date
Credit_Card_ID	varchar(50)
Transaction_Value	decimal
Transaction_Segment	varchar(20)

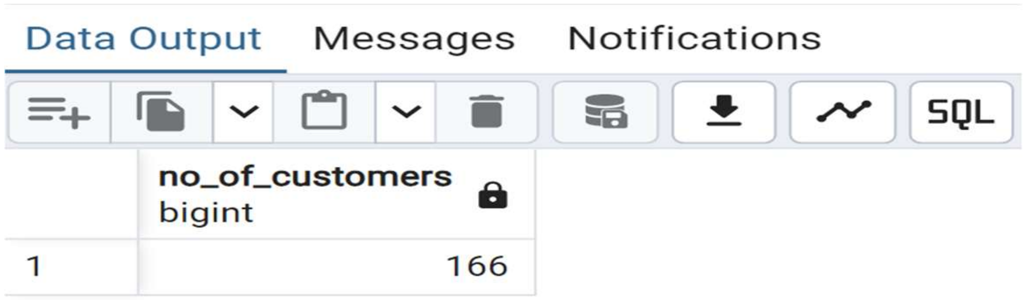
Fraud_base	
Transaction_ID	varchar(20)
Fraud_Flag	int

Customer_base	
Cust_ID	varchar(20)
Age	int
Customer_Segment	varchar(30)
Customer_Vintage_Group	varchar(20)

1 :- How many customers have done transactions over 49000?

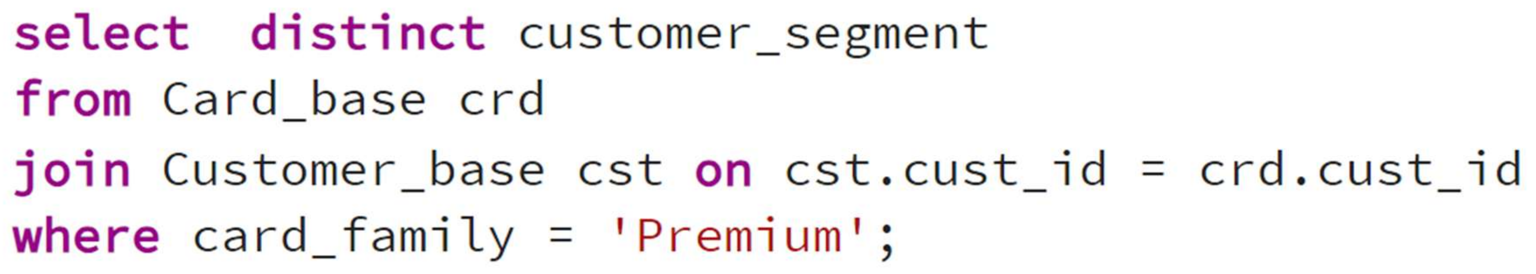


```
select count(distinct cust_id) as no_of_customers
from Transaction_base trn
join Card_base crd on trn.credit_card_id = crd.card_number
where trn.transaction_value > 49000;
```

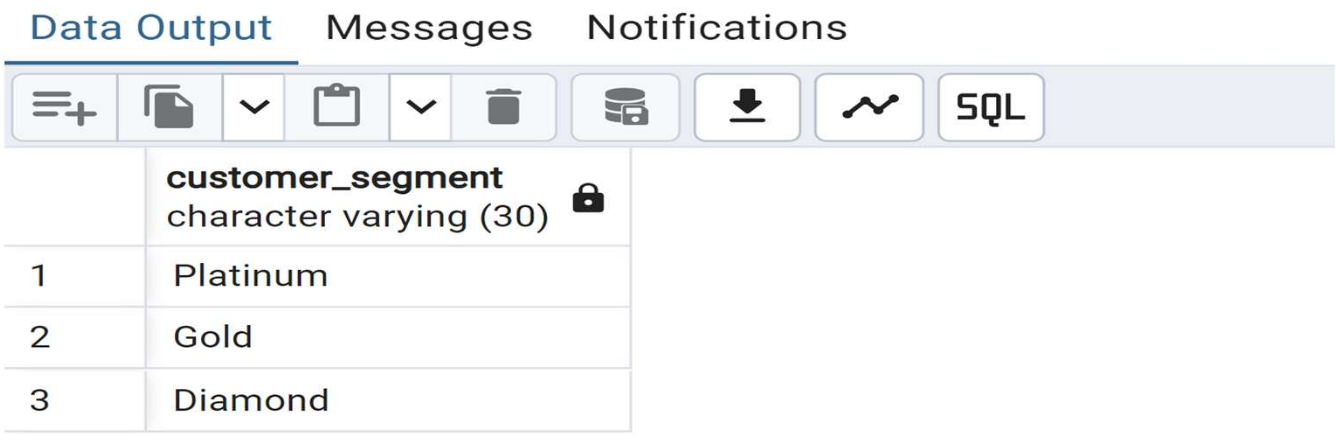


	no_of_customers bigint
1	166

2:- What kind of customers can get a Premium credit card?

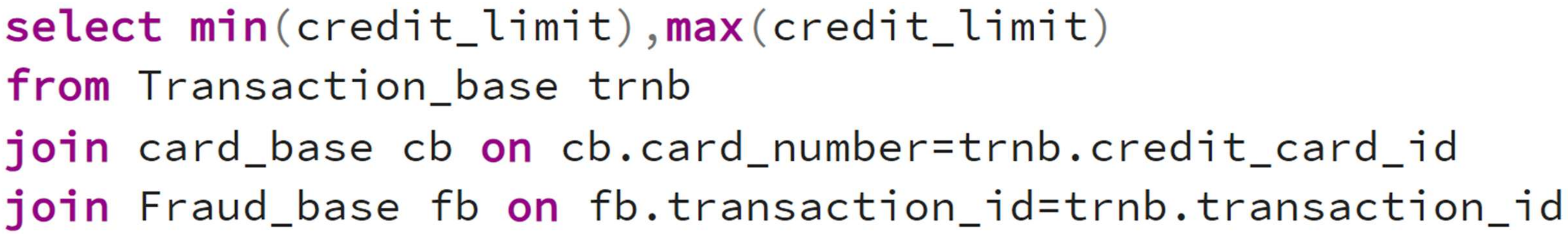


```sql
select  distinct customer_segment
from Card_base crd
join Customer_base cst on cst.cust_id = crd.cust_id
where card_family = 'Premium';
```

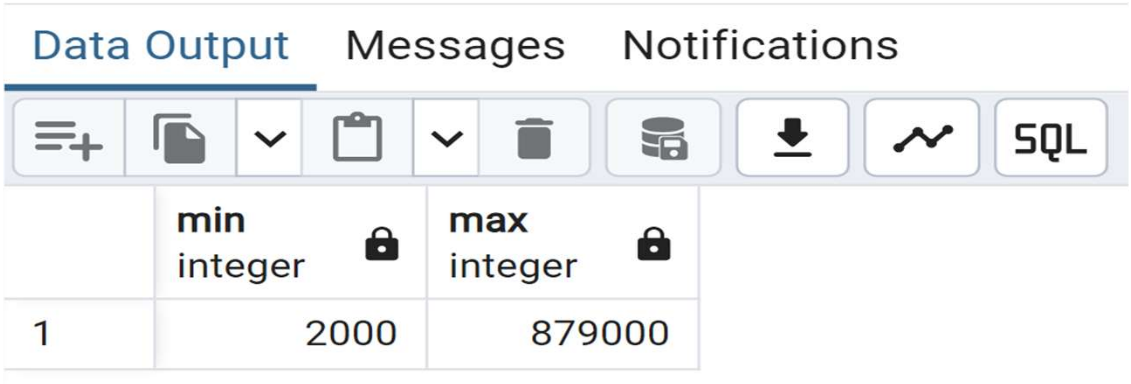


	customer_segment character varying (30)
1	Platinum
2	Gold
3	Diamond

3:- Identify the range of credit limit of customer who have done fraudulent transactions.

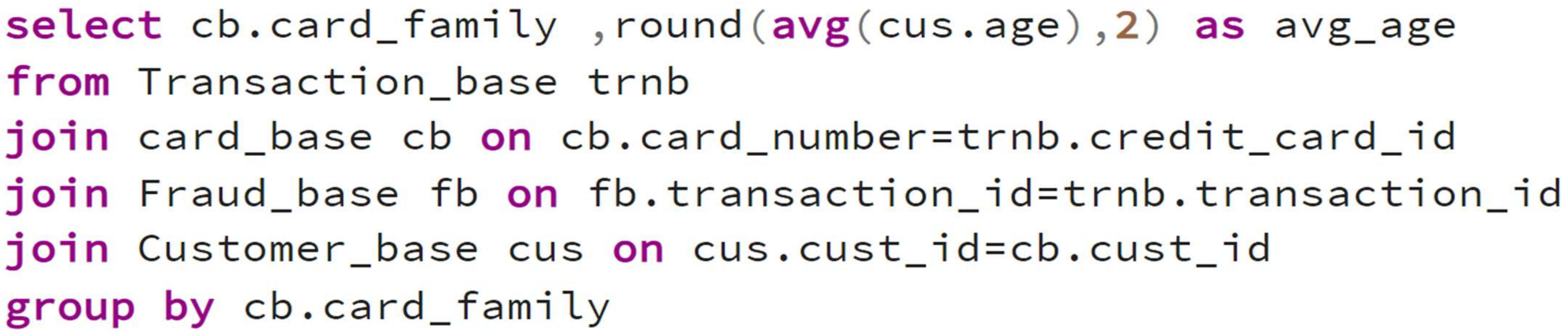


```
select min(credit_limit),max(credit_limit)
from Transaction_base trnb
join card_base cb on cb.card_number=trnb.credit_card_id
join Fraud_base fb on fb.transaction_id=trnb.transaction_id
```

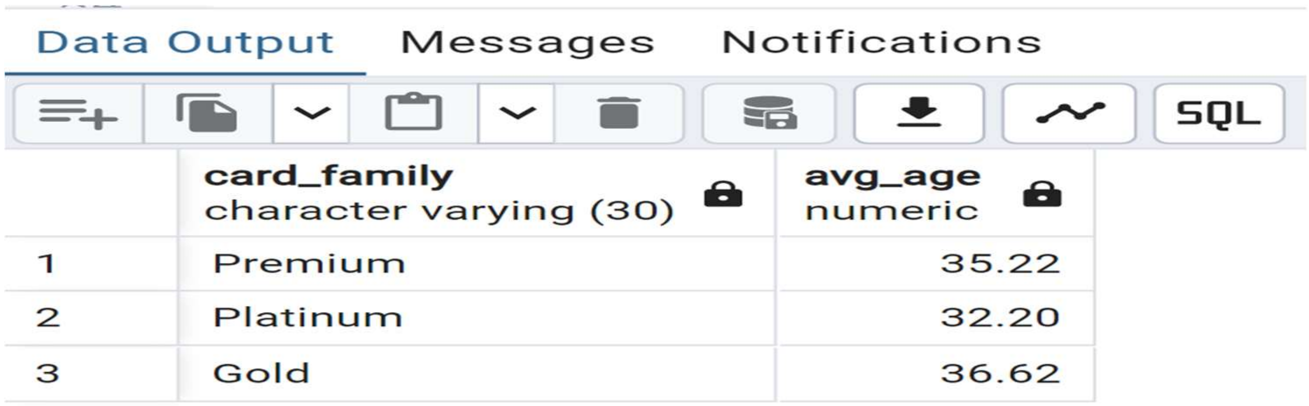


	min integer	max integer
1	2000	879000

4:- What is the average age of customers who are involved in fraud transactions based on different card type?

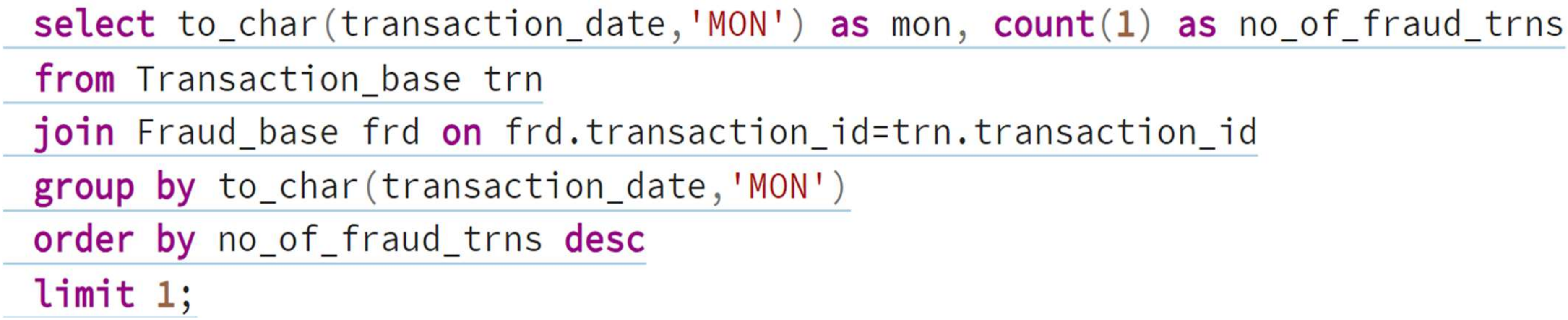


```
select cb.card_family ,round(avg(cus.age),2) as avg_age
from Transaction_base trnb
join card_base cb on cb.card_number=trnb.credit_card_id
join Fraud_base fb on fb.transaction_id=trnb.transaction_id
join Customer_base cus on cus.cust_id=cb.cust_id
group by cb.card_family
```

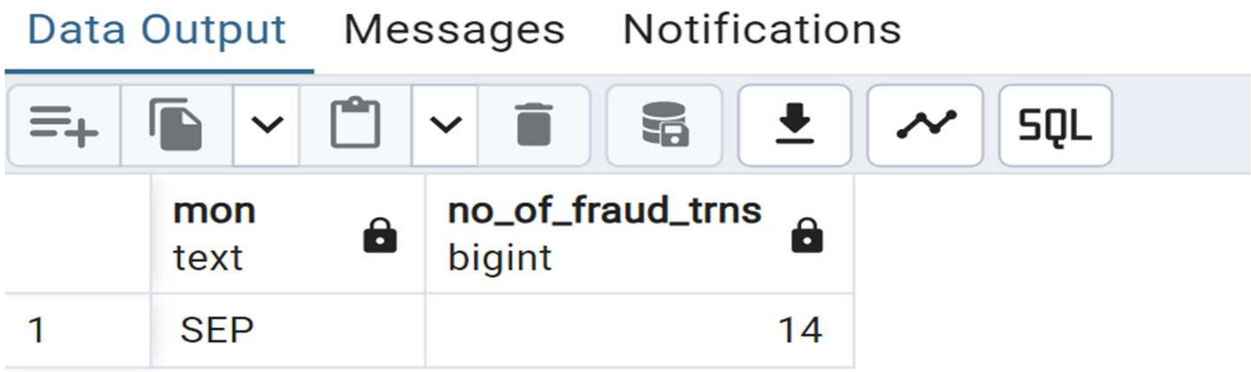


	card_family character varying (30)	avg_age numeric
1	Premium	35.22
2	Platinum	32.20
3	Gold	36.62

5:- Identify the month when highest no of fraudulent transactions occurred

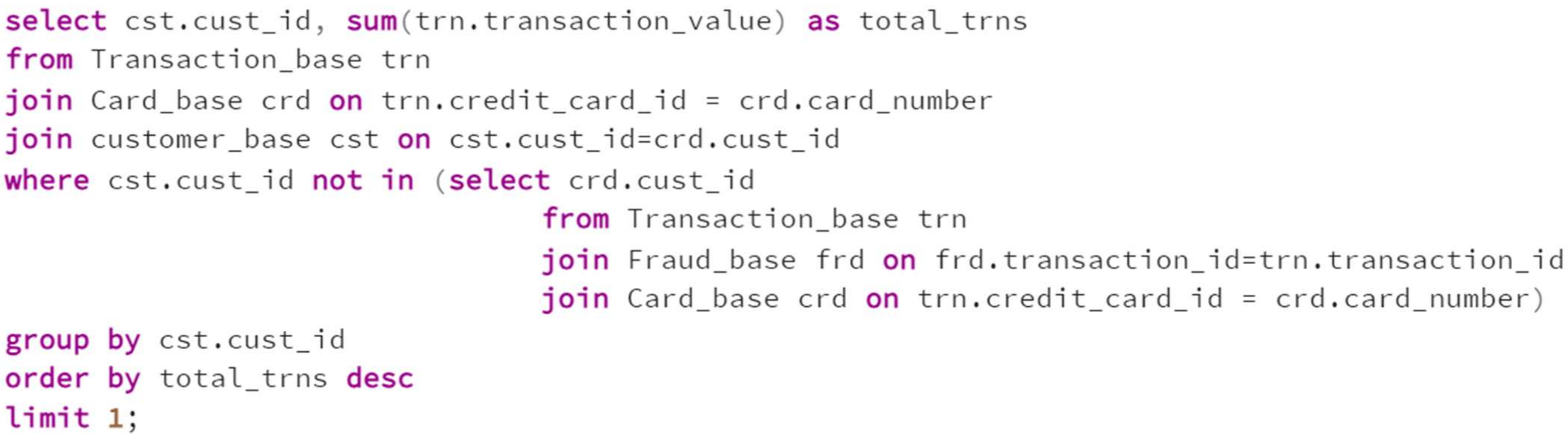


```sql
select to_char(transaction_date,'MON') as mon, count(1) as no_of_fraud_trns
from Transaction_base trn
join Fraud_base frd on frd.transaction_id=trn.transaction_id
group by to_char(transaction_date,'MON')
order by no_of_fraud_trns desc
limit 1;
```

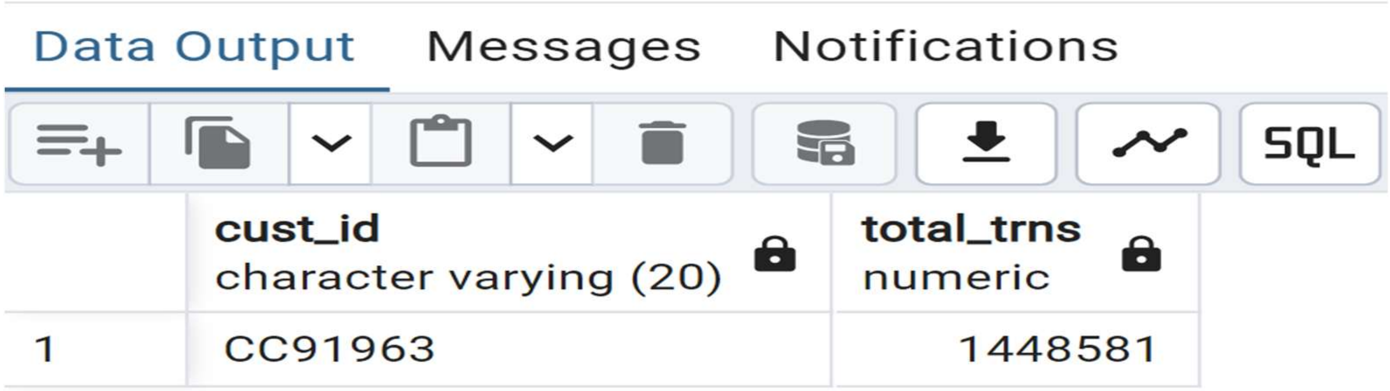


	mon text	no_of_fraud_trns bigint
1	SEP	14

6:- Identify the customer who has done the most transaction value without involving in any fraudulent transactions.

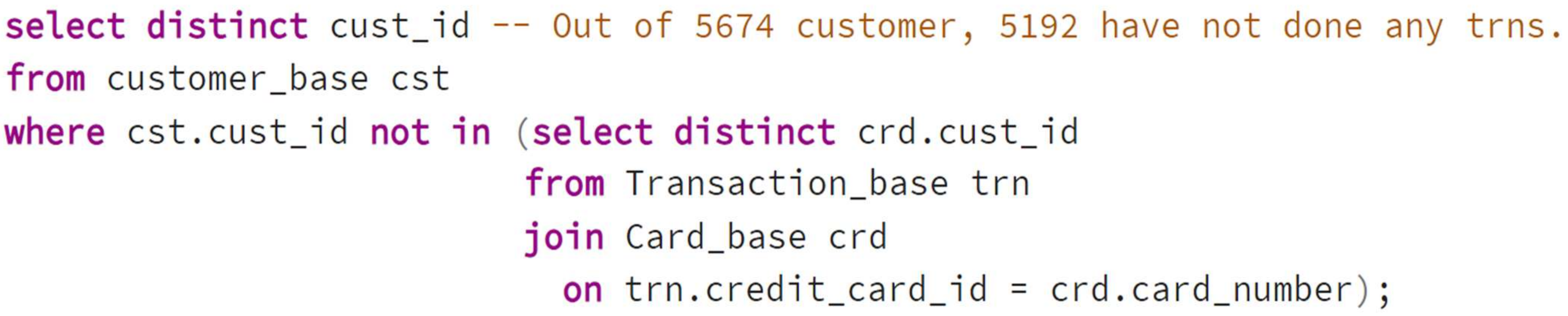


```
select cst.cust_id, sum(trn.transaction_value) as total_trns
from Transaction_base trn
join Card_base crd on trn.credit_card_id = crd.card_number
join customer_base cst on cst.cust_id=crd.cust_id
where cst.cust_id not in (select crd.cust_id
                               from Transaction_base trn
                               join Fraud_base frd on frd.transaction_id=trn.transaction_id
                               join Card_base crd on trn.credit_card_id = crd.card_number)
group by cst.cust_id
order by total_trns desc
limit 1;
```

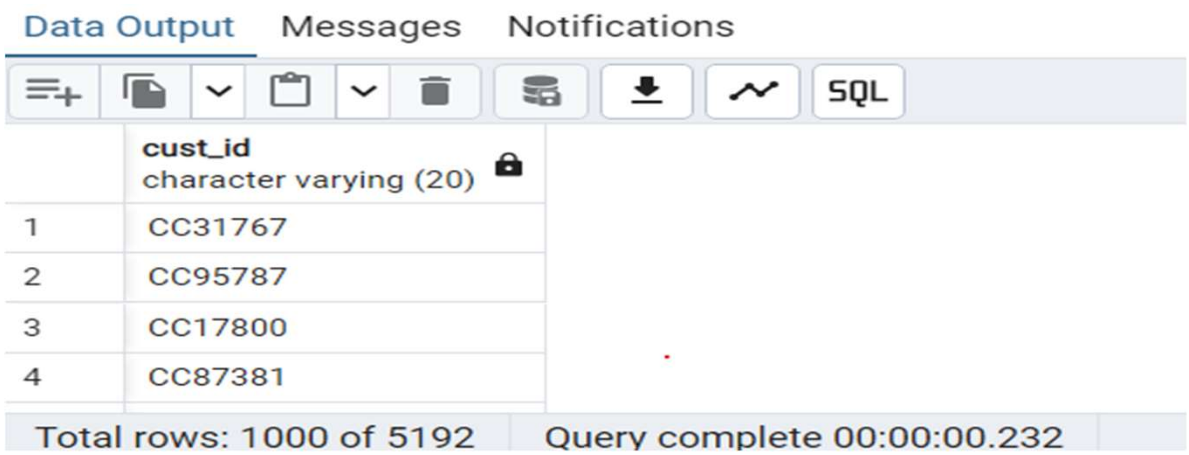


	cust_id character varying (20)	total_trns numeric
1	CC91963	1448581

7:-Check and return any customers who have not done a single transaction.



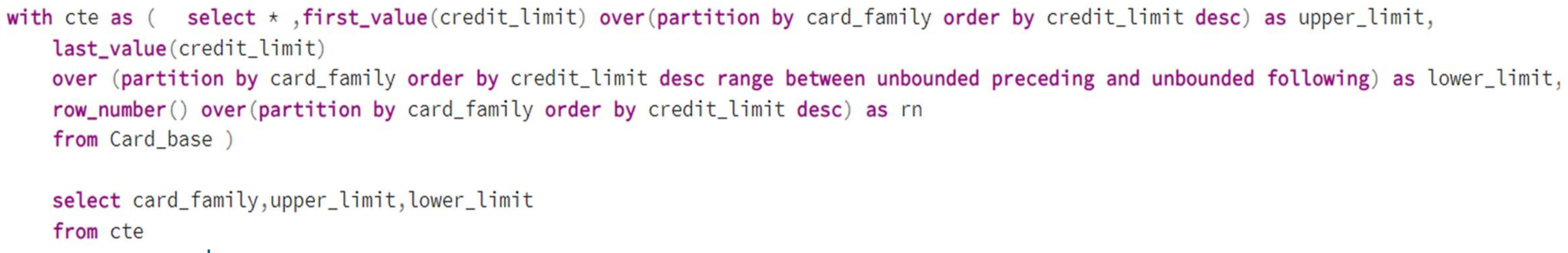
```
select distinct cust_id -- Out of 5674 customer, 5192 have not done any trns.
from customer_base cst
where cst.cust_id not in (select distinct crd.cust_id
                          from Transaction_base trn
                          join Card_base crd
                            on trn.credit_card_id = crd.card_number);
```



	cust_id character varying (20)
1	CC31767
2	CC95787
3	CC17800
4	CC87381

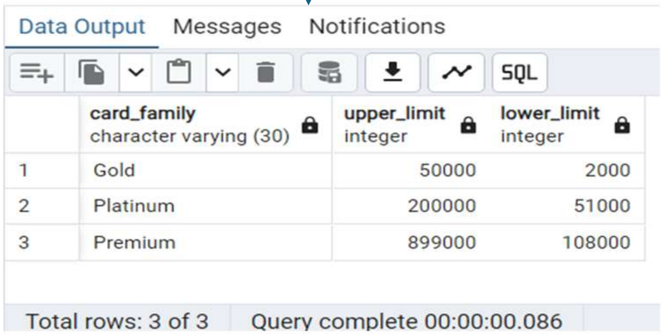
Total rows: 1000 of 5192 Query complete 00:00:00.232

8:-What is the highest and lowest credit limit given to each card type?



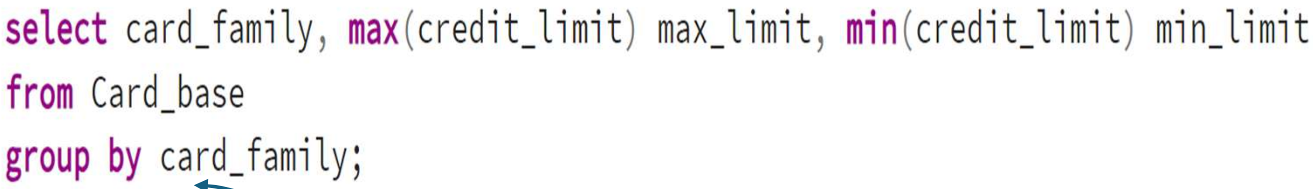
```
with cte as (   select * ,first_value(credit_limit) over(partition by card_family order by credit_limit desc) as upper_limit,
    last_value(credit_limit)
    over (partition by card_family order by credit_limit desc range between unbounded preceding and unbounded following) as lower_limit,
    row_number() over(partition by card_family order by credit_limit desc) as rn
    from Card_base )

    select card_family,upper_limit,lower_limit
    from cte
```

Solution 1



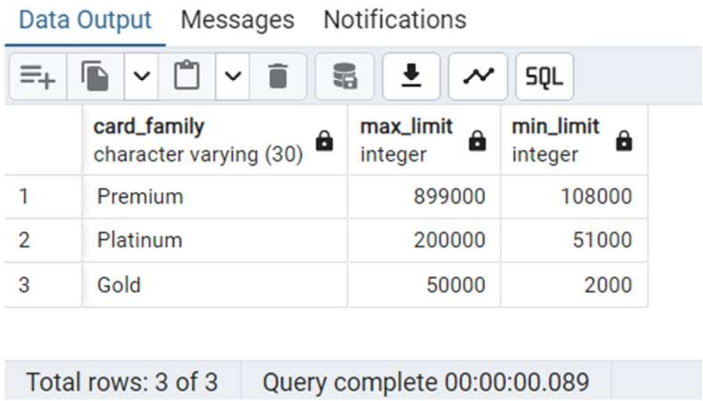
	card_family character varying (30)	upper_limit integer	lower_limit integer
1	Gold	50000	2000
2	Platinum	200000	51000
3	Premium	899000	108000

Total rows: 3 of 3   Query complete 00:00:00.086



```
select card_family, max(credit_limit) max_limit, min(credit_limit) min_limit
from Card_base
group by card_family;
```

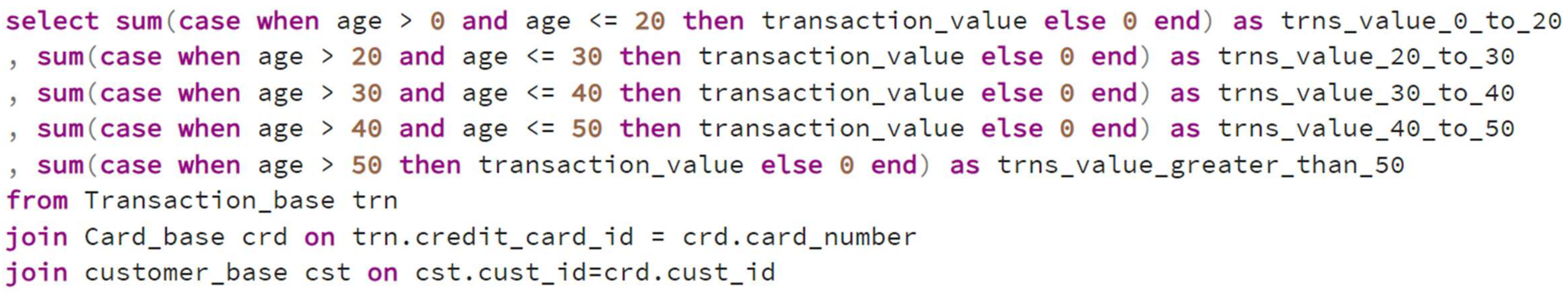
Solution 2



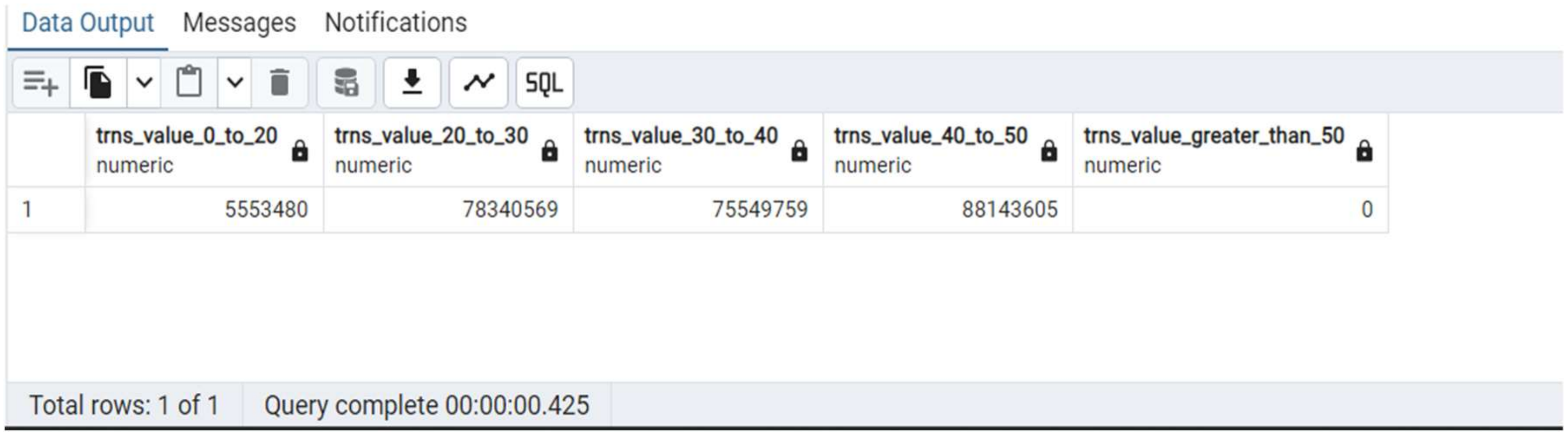
	card_family character varying (30)	max_limit integer	min_limit integer
1	Premium	899000	108000
2	Platinum	200000	51000
3	Gold	50000	2000

Total rows: 3 of 3   Query complete 00:00:00.089

9:- What is the total value of transactions done by customers who come under the age bracket of 20-30 yrs, 30-40yrs, 40-50 yrs, 50+ yrs and 0-20 yrs.



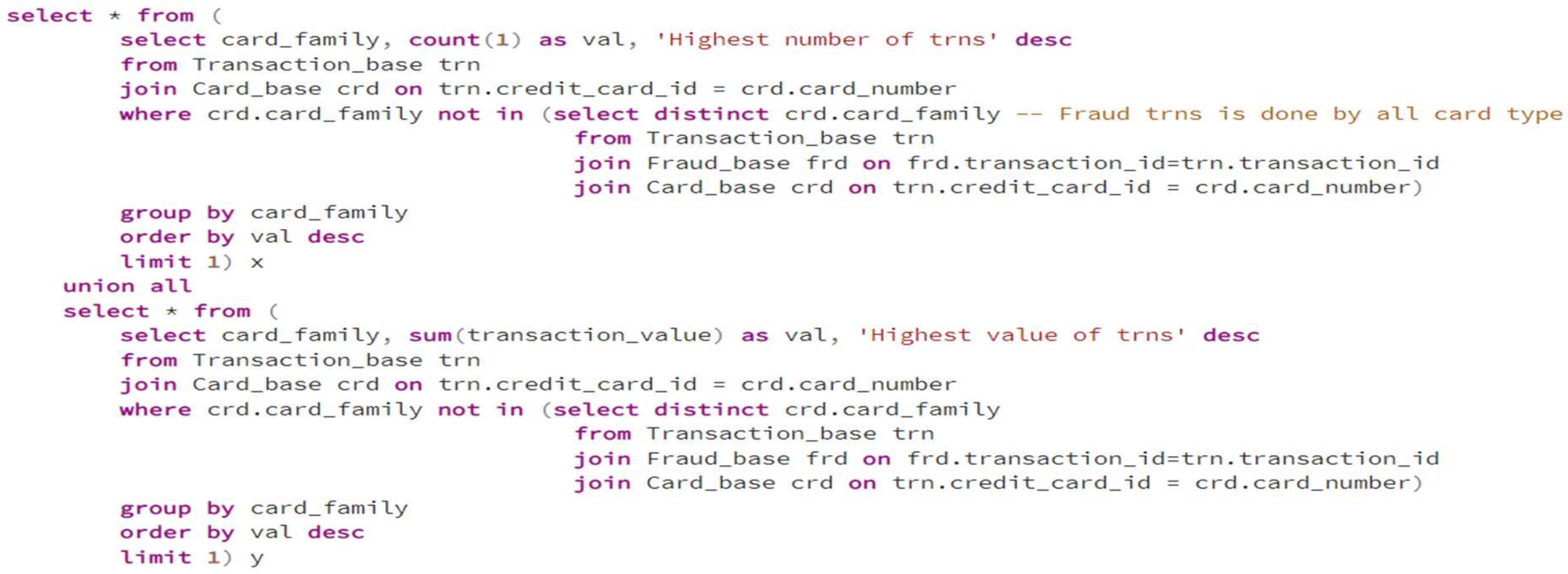
```sql
select sum(case when age > 0 and age <= 20 then transaction_value else 0 end) as trns_value_0_to_20
, sum(case when age > 20 and age <= 30 then transaction_value else 0 end) as trns_value_20_to_30
, sum(case when age > 30 and age <= 40 then transaction_value else 0 end) as trns_value_30_to_40
, sum(case when age > 40 and age <= 50 then transaction_value else 0 end) as trns_value_40_to_50
, sum(case when age > 50 then transaction_value else 0 end) as trns_value_greater_than_50
from Transaction_base trn
join Card_base crd on trn.credit_card_id = crd.card_number
join customer_base cst on cst.cust_id=crd.cust_id
```



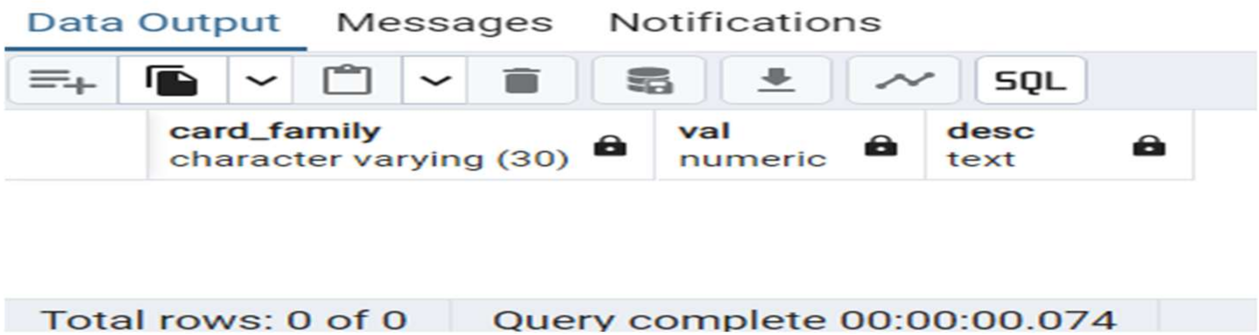
	trns_value_0_to_20 numeric	trns_value_20_to_30 numeric	trns_value_30_to_40 numeric	trns_value_40_to_50 numeric	trns_value_greater_than_50 numeric
1	5553480	78340569	75549759	88143605	0

Total rows: 1 of 1 Query complete 00:00:00.425

10:- Which card type has done the most no of transactions and the total highest value of transactions without



```
select * from (
        select card_family, count(1) as val, 'Highest number of trns' desc
        from Transaction_base trn
        join Card_base crd on trn.credit_card_id = crd.card_number
        where crd.card_family not in (select distinct crd.card_family -- Fraud trns is done by all card type
                                        from Transaction_base trn
                                        join Fraud_base frd on frd.transaction_id=trn.transaction_id
                                        join Card_base crd on trn.credit_card_id = crd.card_number)
        group by card_family
        order by val desc
        limit 1) x
    union all
    select * from (
        select card_family, sum(transaction_value) as val, 'Highest value of trns' desc
        from Transaction_base trn
        join Card_base crd on trn.credit_card_id = crd.card_number
        where crd.card_family not in (select distinct crd.card_family
                                        from Transaction_base trn
                                        join Fraud_base frd on frd.transaction_id=trn.transaction_id
                                        join Card_base crd on trn.credit_card_id = crd.card_number)
        group by card_family
        order by val desc
        limit 1) y
```



card_family character varying (30)	val numeric	desc text

Total rows: 0 of 0   Query complete 00:00:00.074

THANK YOU