Project / Case Study in SQL

Credit Card Transactions

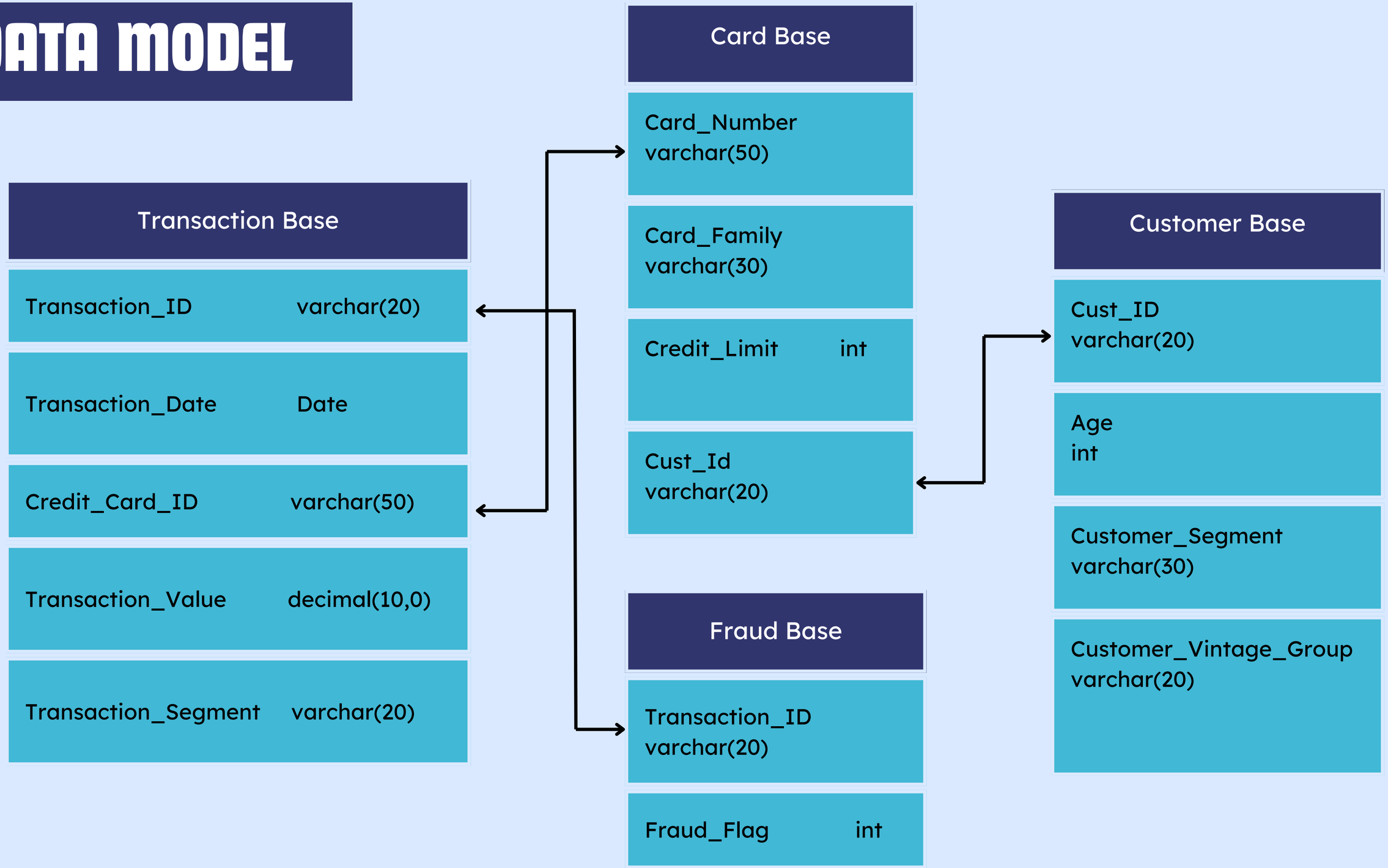
OVERVIEW

We have a dataset that provides comprehensive details on credit card ownership, transaction history, and fraudulent activities, organized into four key tables:

1. Card_base: Contains details about the credit cards held by customers, including card type and credit limit.
2. Customer_base: Provides basic information about the customers who own these credit cards.
3. Transactions_base: Records details of transactions made by customers, including transaction amounts and dates.
4. Fraud_base: Identifies transactions that have been flagged as fraudulent.

Task: Use the information from these tables to gain insights into credit card transactions. Link the tables to perform analyses and solve the given problem statements.

DATA MODEL


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(10,0)
Transaction_Segment varchar(20)
Customer Base
Cust_ID varchar(20)
Age int
Customer_Segment varchar(30)
Customer_Vintage_Group varchar(20)
Fraud Base
Transaction_ID varchar(20)
Fraud_Flag int

HOW MANY CUSTOMERS HAVE DONE TRANSACTIONS OVER 49000.

```
select count(distinct cust_id) as Customers
from transaction_base tb
join card_base cb on tb.credit_card_id = cb.card_number
where transaction_value > 49000;
```

Result Grid

	Customers
▶	166

IDENTIFY THE RANGE OF CREDIT LIMIT OF CUSTOMERS WHO HAVE DONE FRAUDULENT TRANSACTIONS.

```
select concat(min(cb.credit_limit), ' - ', max(cb.credit_limit)) as Credit_Range
from fraud_base fb
join transaction_base tb on fb.transaction_id = tb.transaction_id
join card_base cb on tb.credit_card_id = cb.card_number;
```

Result Grid Filte

	Credit_Range
▶	2000 - 879000

WHAT IS THE AVERAGE AGE OF CUSTOMERS WHO ARE INVOLVED IN FRAUD TRANSACTIONS BASED ON DIFFERENT CARD TYPE.

```
select cb.card_family, round(avg(csb.age),2) as Avg_age_of_customers
from fraud_base fb
join transaction_base tb on fb.transaction_id = tb.transaction_id
join card_base cb on tb.credit_card_id = cb.card_number
join customer_base csb on cb.cust_id = csb.cust_id
group by cb.card_family;
```

Result Grid | Filter Rows:

card_family	Avg_age_of_customers
Gold	36.62
Platinum	32.20
Premium	35.22

IDENTIFY THE MONTH WHEN HIGHEST NO OF FRAUDULENT TRANSACTIONS OCCURED.

```
select month(transaction_date) as month ,count(*) as max_fraud_trans
from fraud_base fb
join transaction_base tb on tb.transaction_id = fb.transaction_id
group by month(transaction_date)
order by max_fraud_trans desc
limit 1;
```

Result Grid | Filter Rows:

	month	max_fraud_trans
▶	9	14

IDENTIFY THE CUSTOMER WHO HAS DONE THE MOST TRANSACTION VALUE WITHOUT INVOLVING IN ANY FRAUDULENT TRANSACTIONS.

```
select cb.cust_id as Customer, sum(tb.Transaction_Value) as Total_sum
from transaction_base tb
left join fraud_base fb on tb.transaction_id = fb.transaction_id
join card_base cb on tb.credit_card_id = cb.card_number
where cb.cust_id not in ( select cust_id
                          from card_base cb
                          join transaction_base tb on cb.Card_Number = tb.Credit_Card_ID
                          join fraud_base fb on tb.Transaction_ID = fb.Transaction_ID )
 group by cb.Cust_ID
order by total_sum desc
limit 1;
```

Result Grid Filter Rov

	Customer	Total_sum
▶	CC91963	1448581

CHECK AND RETURN ANY CUSTOMERS WHO HAVE NOT DONE A SINGLE TRANSACTION.

```
select csb.cust_id as Customers
from customer_base csb
left join card_base cb on csb.cust_id = cb.cust_id
where cb.card_number is null;
```

Result Grid

Customers
CC25034
CC59625
CC69314
CC67036
CC25597

5192 row(s) returned

OR

```
select distinct cust_id as Customers
from customer_base csb
where csb.cust_id not in (select distinct cb.cust_id
                          from Transaction_base tb
                          join Card_base cb on tb.credit_card_id = cb.card_number);
```

Result Grid

Customers
CC25034
CC59625
CC69314
CC67036

5192 row(s) returned

WHAT IS THE HIGHEST AND LOWEST CREDIT LIMIT GIVEN TO EACH CARD TYPE.

```
select card_family, min(credit_limit) as Min_limit, max(credit_limit) as Max_limit
from card_base
group by card_family ;
```

Result Grid | Filter Rows:

card_family	Min_limit	Max_limit
Premium	108000	899000
Gold	2000	50000
Platinum	51000	200000

WHAT IS THE TOTAL VALUE OF TRANSACTIONS DONE BY CUSTOMERS WHO COME UNDER THE AGE BRACKET OF 20-30 YRS, 30-40 YRS, 40-50 YRS, 50+ YRS AND 0-20 YRS.

```
with cte as
            ( select csb.age,tb.transaction_value
              from customer_base csb
              join card_base cb on cb.cust_id = csb.cust_id
              join transaction_base tb on cb.card_number = tb.credit_card_id),
      age_group as
            ( select case when (age<= 20) then '0-20'
                          when (age > 20 and age <= 30) then '20-30'
                          when (age > 30 and age <= 40) then '30-40'
                          when (age > 40 and age <= 50) then '40-50'
                          else '50+' end as Age_group,
                     transaction_value
              from cte )
select age_group, sum(transaction_value) as Total_Sum_Value
from age_group
group by age_group
order by age_group;
```

Result Grid | Filter Rows:

Age_group	Total_Sum_Value
0-20	5553480
20-30	78340569
30-40	75549759
40-50	88143605

PTO (OR)

SAME PROBLEM STATEMENT

```
select sum(case when (age <= 20) then transaction_value else 0 end) as 'Age Group 0-20',
        sum(case when (age>20 and age <= 30) then transaction_value else 0 end) as 'Age Group 20-30',
        sum(case when (age>30 and age <= 40) then transaction_value else 0 end) as 'Age Group 30-40',
        sum(case when (age>40 and age <= 50) then transaction_value else 0 end) as 'Age Group 40-50',
        sum(case when (age>50) then transaction_value else 0 end) as 'Age Group 50+'
from customer_base csb
join card_base cb on csb.cust_id = cb.cust_id
join transaction_base tb on cb.card_number = tb.credit_card_id ;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Age Group 0-20	Age Group 20-30	Age Group 30-40	Age Group 40-50	Age Group 50+
5553480	78340569	75549759	88143605	0

EXCLUDING FRAUDULENT TRANSACTIONS, WHICH CARD TYPE HAS DONE THE MOST NO OF TRANSACTIONS AND THE TOTAL HIGHEST VALUE OF TRANSACTIONS.

```
select *
from  ( select card_family, count(1) as Trans
          from transaction_base tb
          join card_base cb on tb.credit_card_id = cb.card_number
         where transaction_id not in (select transaction_id from fraud_base)
         group by card_family
         order by trans desc
          limit 1) x
union all
select *
from ( select card_family, sum(transaction_value) as Total
         from transaction_base tb
         join card_base cb on tb.credit_card_id = cb.card_number
         where transaction_id not in (select transaction_id from fraud_base)
         group by card_family
         order by total desc
          limit 1) y;
```

Result Grid | Filter Ro

card_family	Trans
Premium	4054
Premium	100002750