# 💳 Credit Card Transactions Analysis - SQL Case Study

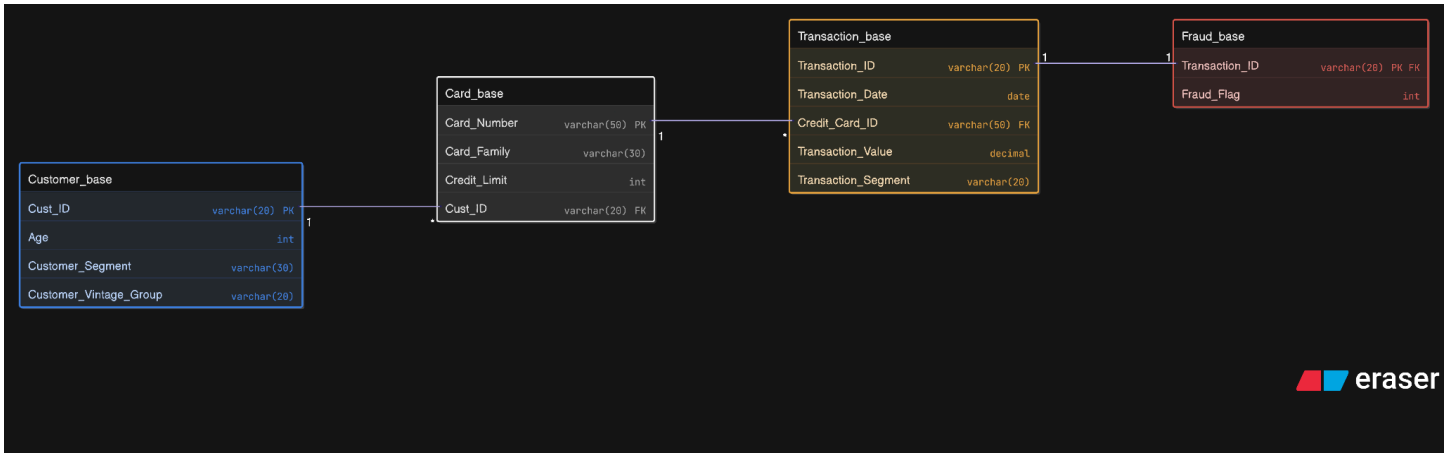
## Project Overview

A comprehensive SQL analysis of credit card transaction data, examining customer behaviour, fraud patterns, and transaction trends across different customer segments and card types.

**Dataset:** 4 tables with 16,283 total records

- `transactionbase` - 10,000 transactions
- `customerbase` - 5,674 customers
- `cardbase` - 500 credit cards
- `fraudbase` - 109 fraudulent transactions

**ERD Diagram**



# Business Questions & Solutions

## 1. High-Value Transaction Analysis

**Question:** How many customers have made transactions over $49,000?

```
SELECT
    COUNT(DISTINCT c.Cust_ID) AS Count_of_cx
FROM TransactionBase t
JOIN CardBase c ON t.Credit_Card_ID = c.card_number
WHERE t.transaction_value > 49000;
```

**Result:** 166 customers

**Key Insight:** Only 2.9% of customers make ultra-high-value transactions, indicating a concentrated premium segment.

---

## 2. Premium Card Eligibility

**Question:** Which customer segments qualify for Premium credit cards?

```
SELECT
   DISTINCT cb.Customer_Segment
FROM CustomerBase cb
JOIN CardBase crd ON crd.Cust_ID = cb.Cust_ID
WHERE crd.Card_Family = 'Premium';
```

**Result:** Gold, Diamond, Platinum segments

**Key Insight:** Premium cards are reserved for top-tier customer segments.

---

## 3. Fraud Risk by Credit Limit

**Question:** What is the credit limit range for customers involved in fraud?

```
SELECT
   MAX(Credit_Limit) AS max_limit,
   MIN(Credit_Limit) AS min_limit
FROM TransactionBase tb
JOIN FraudBase fb ON tb.Transaction_ID = fb.Transaction_ID
JOIN CardBase cb ON cb.Card_Number = tb.Credit_Card_ID;
```

**Result:** $2,000 - $879,000

**Key Insight:** Fraud occurs across all credit limit tiers, not just high-limit accounts.

---

## 4. Fraud Demographics by Card Type

**Question:** What's the average age of fraud victims by card type?

```
SELECT
   cb.Card_Family,
   AVG(cxb.Age) AS avg_age
FROM TransactionBase tb
JOIN FraudBase fb ON tb.Transaction_ID = fb.Transaction_ID
JOIN CardBase cb ON cb.Card_Number = tb.Credit_Card_ID
JOIN CustomerBase cxb ON cxb.Cust_ID = cb.Cust_ID
GROUP BY cb.Card_Family;
```

| Card Family | Avg Age |
|---|---|
| Premium | 35 |
| Gold | 36 |
| Platinum | 32 |

**Key Insight:** Younger Platinum cardholders (32) are more vulnerable to fraud.

---

## 5. Fraud Seasonality Analysis

**Question:** Which month had the highest fraud activity?

**Approach 1 - Simple:**

```
SELECT TOP 1
    DATENAME(MONTH, tb.transaction_date) AS mon,
    COUNT(1) AS no_of_fraud_trns
FROM Transactionbase tb
JOIN Fraudbase fb ON fb.transaction_id = tb.transaction_id
GROUP BY DATENAME(MONTH, tb.transaction_date)
ORDER BY no_of_fraud_trns DESC;
```

**Approach 2 - Tie-Safe (Recommended):**

```
WITH fraud_counts AS (
    SELECT
        DATENAME(MONTH, tb.transaction_date) AS month_name,
        COUNT(fb.transaction_id) AS fraud_transactions
    FROM TransactionBase tb
    JOIN FraudBase fb ON tb.transaction_id = fb.transaction_id
    GROUP BY DATENAME(MONTH, tb.transaction_date)
)
SELECT month_name
FROM fraud_counts
WHERE fraud_transactions = (SELECT MAX(fraud_transactions) FROM fraud_counts);
```

**Result:** September (14 fraudulent transactions)

**Technical Note:** Approach 2 handles ties and is more performant than window functions for this use case.

---

## 6. Top Legitimate Spender

**Question:** Who has the highest transaction value with zero fraud?

```
WITH cte AS(
    SELECT
        cb.Cust_ID,
        SUM(tb.Transaction_Value) AS total_transaction_value
    FROM Transactionbase tb
    LEFT JOIN Fraudbase fb ON fb.transaction_id = tb.transaction_id
    JOIN CardBase cb ON cb.Card_Number = tb.Credit_Card_ID
    WHERE fb.Fraud_Flag IS NULL
    GROUP BY cb.Cust_ID
)
SELECT
    cust_id,
    total_transaction_value
FROM cte
WHERE total_transaction_value = (SELECT MAX(total_transaction_value) FROM cte);
```

**Result:** Customer CC91963 - $1,448,581

**Key Insight:** Identifies high-value, low-risk customers for VIP programs.

---

## 7. Inactive Customer Identification

**Question:** How many customers have never made a transaction?

```sql
SELECT
    DISTINCT cxb.Cust_ID
FROM CustomerBase cxb
LEFT JOIN CardBase cb ON cxb.Cust_ID = cb.Cust_ID
LEFT JOIN Transactionbase tb ON cb.Card_Number = tb.Credit_Card_ID
WHERE tb.Transaction_ID IS NULL;
```

**Result:** 5,192 customers (91.5% inactive rate)

**Key Insight:** Massive opportunity for activation campaigns.

---

## 8. Credit Limit Strategy by Card Type

**Question:** What are the credit limit ranges for each card family?

```sql
SELECT
    card_family,
    MIN(Credit_Limit) AS min_limit,
    MAX(Credit_Limit) AS max_limit
FROM CardBase
GROUP BY card_family;
```

| Card Family | Min Limit | Max Limit |
|---|---|---|
| Gold | $2,000 | $50,000 |
| Platinum | $51,000 | $200,000 |
| Premium | $108,000 | $899,000 |

**Key Insight:** Clear tiered structure with no overlap between card types.

---

## 9. Transaction Value by Age Demographics

**Question:** How do transaction values distribute across age groups?

**Vertical Display:**

```
SELECT
    CASE
        WHEN cb.Age BETWEEN 0 AND 20 THEN '0-20 yrs'
        WHEN cb.Age BETWEEN 20 AND 30 THEN '20-30 yrs'
        WHEN cb.Age BETWEEN 30 AND 40 THEN '30-40 yrs'
        WHEN cb.Age BETWEEN 40 AND 50 THEN '40-50 yrs'
        ELSE '50+ yrs'
    END AS age_group,
    SUM(tb.Transaction_Value) AS total_transaction_value
FROM CustomerBase cb
JOIN CardBase crd ON crd.Cust_ID = cb.Cust_ID
JOIN Transactionbase tb ON crd.Card_Number = tb.Credit_Card_ID
GROUP BY
    CASE
        WHEN cb.Age BETWEEN 0 AND 20 THEN '0-20 yrs'
        WHEN cb.Age BETWEEN 20 AND 30 THEN '20-30 yrs'
        WHEN cb.Age BETWEEN 30 AND 40 THEN '30-40 yrs'
        WHEN cb.Age BETWEEN 40 AND 50 THEN '40-50 yrs'
        ELSE '50+ yrs'
    END
ORDER BY age_group;
```

**Horizontal Display (Alternative):**

```
SELECT
    SUM(CASE WHEN cxb.age BETWEEN 0 AND 20 THEN tb.transaction_value ELSE 0 END) AS
trns_value_0_to_20,
    SUM(CASE WHEN cxb.age BETWEEN 21 AND 30 THEN tb.transaction_value ELSE 0 END) AS
trns_value_20_to_30,
    SUM(CASE WHEN cxb.age BETWEEN 31 AND 40 THEN tb.transaction_value ELSE 0 END) AS
trns_value_30_to_40,
    SUM(CASE WHEN cxb.age BETWEEN 41 AND 50 THEN tb.transaction_value ELSE 0 END) AS
trns_value_40_to_50,
    SUM(CASE WHEN cxb.age > 50 THEN tb.transaction_value ELSE 0 END) AS
trns_value_greater_than_50
FROM TransactionBase tb
JOIN CardBase cb ON tb.credit_card_id = cb.card_number
JOIN CustomerBase cxb ON cb.cust_id = cxb.cust_id;
```

| Age Group | Total Value |
|---|---|
| 0-20 yrs | $5,553,480 |
| 20-30 yrs | $78,340,569 |
| 30-40 yrs | $75,549,759 |
| 40-50 yrs | $88,143,605 |

**Key Insight:** 40-50 age group generates highest transaction value; 0-20 contributes minimal volume.

---

## 10. Best Performing Card Type

**Question:** Which card type leads in transaction count AND value (excluding fraud)?

```sql
WITH cte AS (
    SELECT
        cb.card_family,
        COUNT(tb.transaction_id) AS transaction_count,
        SUM(tb.transaction_value) AS total_transaction_value,
        RANK() OVER (ORDER BY COUNT(tb.transaction_id) DESC) AS rnk_count,
        RANK() OVER (ORDER BY SUM(tb.transaction_value) DESC) AS rnk_value
    FROM CardBase cb
    JOIN TransactionBase tb ON tb.credit_card_id = cb.card_number
    LEFT JOIN FraudBase fb ON fb.transaction_id = tb.transaction_id
    WHERE fb.transaction_id IS NULL
    GROUP BY cb.card_family
)
SELECT
    card_family,
    transaction_count,
    total_transaction_value,
    'Highest number of transactions' AS metric
FROM cte
WHERE rnk_count = 1

UNION ALL

SELECT
    card_family,
    transaction_count,
    total_transaction_value,
    'Highest total transaction value' AS metric
FROM cte
WHERE rnk_value = 1;
```

**Result:** Premium cards dominate both metrics

- 4,054 transactions
- $100,002,750 total value

**Key Insight:** Premium cardholders drive disproportionate business value.

---

# Key Takeaways

1. **Concentrated Value:** 166 customers (2.9%) drive ultra-high-value transactions
2. **Fraud Exposure:** All credit limit tiers vulnerable; younger Platinum holders at higher risk
3. **Seasonal Pattern:** September shows peak fraud activity
4. **Inactive Base:** 91.5% customer inactivity represents a major growth opportunity
5. **Premium Dominance:** Premium cards generate the highest volume and value
6. **Age Demographics:** The 40-50 age group is the most valuable segment

---

# Skills Demonstrated

- Complex JOINs (INNER, LEFT, multiple table joins)
- CTEs & Subqueries
- Window Functions (RANK, aggregates)
- Date Functions (DATENAME)
- Conditional Logic (CASE WHEN)
- Aggregations (COUNT, SUM, AVG, MIN, MAX)
- Set Operations (UNION ALL)
- Performance Optimization
- NULL Handling

---

*Analysis performed using SQL Server / T-SQL*