



# Credit Card Fraud Analysis

---



# Database Schema

Customer Table

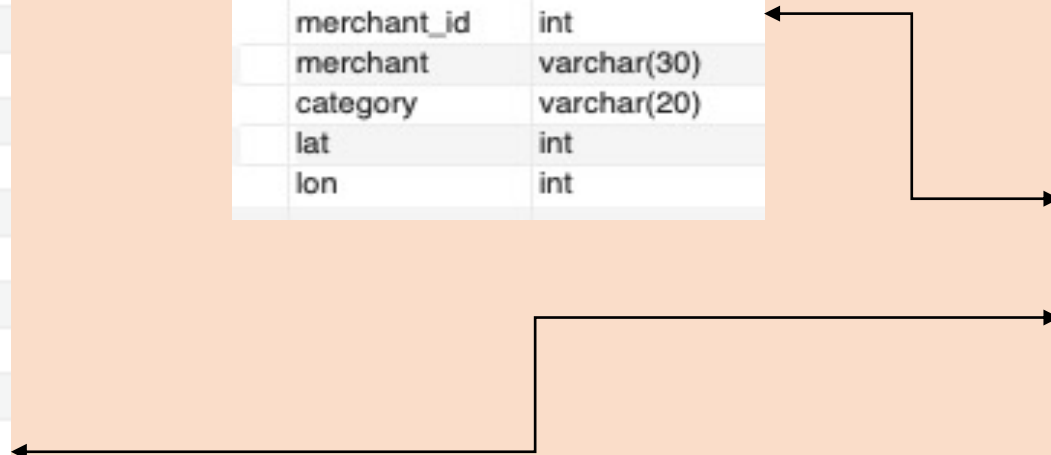
Field	Type	^
customer_id	int	
zip	int	
lat	int	
lon	int	
city_pop	int	
state	varchar(10)	
dob	varchar(15)	
city	varchar(20)	
name	varchar(255)	
street	varchar(30)	
job	varchar(30)	
gender	varchar(5)	

Merchant Table

Field	Type
merchant_id	int
merchant	varchar(30)
category	varchar(20)
lat	int
lon	int

Transactions Table

Field	Type
trans_num	varchar(50)
trans_date_tr...	varchar(70)
cc_num	varchar(50)
amt	int
unix_time	int
merchant_id	int
is_fraud	int
customer_id	int

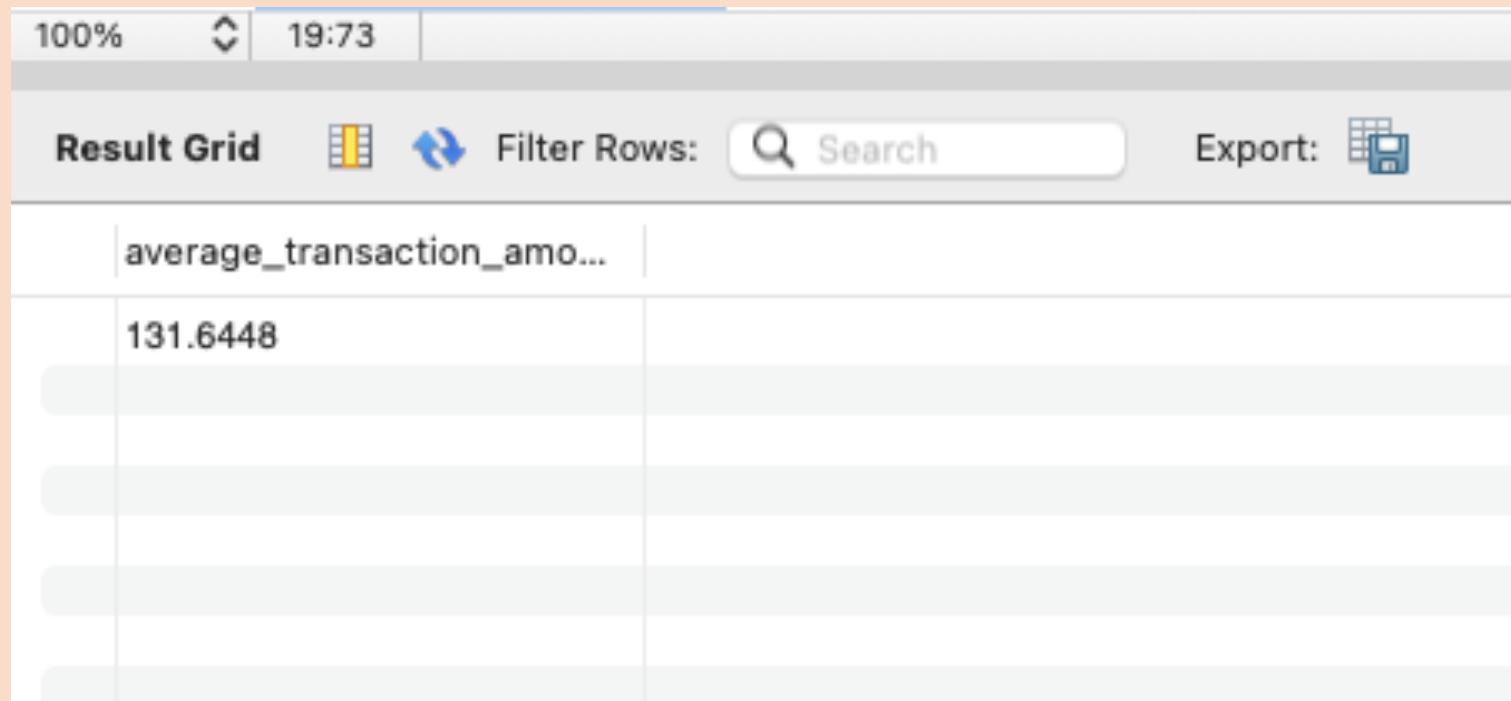




# Objective

- The objectives of credit card fraud prediction typically revolve around the following
  1. Risk Mitigation
  2. Enhanced Security
  3. Customer Trust
  4. Regulatory Compliance
  5. Operational Efficiency
  6. Data Analysis and Insights
- Overall, the objectives of credit card fraud prediction are centred around reducing financial losses, enhancing security, maintaining trust, ensuring regulatory compliance, improving operational efficiency, and deriving actionable insights from transactional data.

- Calculate the average transaction amount from the transactions table.



The screenshot shows a data grid interface. At the top, there is a status bar with '100%' zoom, a refresh icon, and a timer '19:73'. Below this is a toolbar with the text 'Result Grid', a grid icon, a refresh icon, the text 'Filter Rows:', a search input field with a magnifying glass icon and the text 'Search', and an 'Export:' button with a grid and save icon. The main area is a table with one column. The header row contains the text 'average\_transaction\_amo...'. The first data row contains the value '131.6448'. There are four more empty data rows below it.

average_transaction_amo...
131.6448

- Display the details of all the transactions having an amount more than the average.

```
77 • select *
78   from transactions
79   where amt >
80     (select avg(amt)
81      from transactions);
82
```

100% 1:82

**Result Grid** Filter Rows: Search Edit: Export/Import:

	trans_num	trans_date_trans_ti...	cc_num	amt	unix_time	merchant_id	is_fraud	customer_id
▶	02e57dd458293314b34608be3...	2020-06-21 19:04	3.58364E+15	156	1371841463	645	0	110475
	041e573c99f5f86a5110a99a22...	2020-06-28 23:48	3.58113E+15	1197	1372463331	1634	1	110764
	0456433ca67345e07e4cd005a...	2020-06-21 18:57	2.71402E+15	143	1371841022	1273	0	110693
	05668abc68d0c0f435bf53247e...	2020-07-05 02:44	4.98985E+18	137	1372992298	1756	1	110576
	067962ea69e16012169bddbc5...	2020-06-21 16:28	4.70787E+15	197	1371832127	460	0	110478
	078da9a1497abd069d041aec3...	2020-06-30 23:45	3.52823E+15	359	1372635952	774	1	110766
	08fbb1e3406ceaeb36e59f18c0...	2020-06-21 19:43	3.7493E+14	159	1371843786	197	0	110202
	0940da2e91607ffb8961e562b0...	2020-06-21 18:55	3.0235E+13	147	1371840903	1267	0	110180
	0a08fed1b670277ce0e1aed3b7...	2020-07-11 03:58	3.85443E+13	731	1373515139	1824	1	110662
	0af045d8a5b4ae876b90f72ea4...	2020-06-21 20:07	3.51867E+15	377	1371845241	296	0	110247
	0b0ef68bdb62d19caae3fb33e5...	2020-06-21 18:10	2.72083E+15	144	1371838230	300	0	110396
	0da884d5079ef2919db75faeb3...	2020-06-21 12:42	3.77234E+14	136	1371818549	191	0	110196
	0deca8c42edbdbc8dca65b69af...	2020-07-12 23:40	3.5149E+15	1111	1373672457	428	1	110479
	0e735bba4550c84d0cf3ba2b3c...	2020-07-11 00:52	3.85443E+13	347	1373503937	1141	1	110662

- Display the value of client 110262. Order the data by amt in descending order.

```
85 • Select *
86 from transactions
87 where customer_id=110262
88 order by amt desc;
89
```

100% 19:88

**Result Grid** Filter Rows: Search Edit: Export/Import:

trans_num	trans_date_trans_ti...	cc_num	amt	unix_time	merchant_id	is_fraud	customer_id
2567ddb79772901c49f33ce71c...	2020-06-21 13:10	4.73685E+18	185	1371820211	270	0	110262
a73b532566688d52d443b6e51...	2020-06-21 14:42	4.73685E+18	104	1371825756	524	0	110262
001c8cce16fc14b91ab9301c1d...	2020-06-21 19:19	4.73685E+18	90	1371842341	177	0	110262
253daa27ca591b4a26176b7b9...	2020-06-21 16:40	4.73685E+18	26	1371832811	877	0	110262
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL



- Display the average of amount. order the data by amt in descending order.

```

92 • SELECT * FROM Merchant
93     INNER JOIN Transactions ON transactions.Merchant_id = Merchant.Merchant_id
94     where (select avg(amt)
95            from transactions)
96     order by amt desc;
97

```

100% 19:96

**Result Grid** Filter Rows: Search Export:

	merchant_id	merchant	category	lat	lon	trans_num	trans_...	cc_num	amt	unix_time	merchant_id	is_fraud	customer_id
▶	1085	fraud_Pollich LLC	home	35	-99	000245bf66e...	2020-06-21...	4.31958E+18	106	1371837096	1085	0	110645
	322	fraud_Gerhold LLC	home	45	-89	0016e6ec2ed...	2020-06-21...	3.80575E+13	61	1371823413	322	0	110368
	344	fraud_Thiel Ltd	travel	41	-74	009df0a601fb...	2020-06-21...	2.13157E+14	8	1371831706	344	0	110169
	115	fraud_Welch, Rath an...	entertainment	29	-81	00da7249535...	2020-06-21...	4.90663E+18	2	1371817134	115	0	110125
	641	fraud_Watsica LLC	personal_care	40	-93	0143364c291...	2020-06-21...	4.90063E+12	7	1371845270	641	0	110228
	569	fraud_Boyer PLC	misc_pos	41	-84	014fc2cec14...	2020-06-21...	3.51237E+15	39	1371840623	569	0	110295
	338	fraud_Streich Ltd	home	42	-74	0183bebc44f...	2020-06-21...	4.82237E+15	11	1371821701	338	0	110314
	755	fraud_Auer LLC	personal_care	31	-85	019bacbe617...	2020-06-21...	3.43746E+14	68	1371846901	755	0	110275
	633	fraud_Christiansen-G...	misc_pos	46	-101	01b3d07c664...	2020-06-21...	3.82956E+13	31	1371846915	633	0	110675
	428	fraud_Kilback LLC	food_dining	41	-74	01bbb7a906...	2020-06-26...	2.13161E+14	8	1372226213	428	1	110395
	117	fraud_Lang, Towne a...	kids_pets	32	-97	01ce36bf41f8...	2020-06-21...	6.75946E+11	26	1371835599	117	0	110622
	140	fraud_Reilly LLC	home	30	-91	01de2fec50...	2020-06-21...	3.76445E+14	27	1371819319	140	0	110234
	838	fraud_Haley, Batz an...	health_fitness	41	-103	02279c6f930...	2020-06-21...	6.01137E+15	6	1371831878	838	0	110307
	307	fraud_Renner Ltd	home	34	-118	023848771c5...	2020-06-21...	2.30534E+15	22	1371821144	307	0	110293

- Display the transactions happening from personal care, entertainment, kids pets and travel categories

```
101 • SELECT *
102 FROM merchant
103 WHERE category IN ('personal_care', 'entertainment', 'kids_pets', 'travel');
104
```

100% 77:103

Result Grid Filter Rows: Search Edit: Export/Import:

	merchant_id	merchant	category	lat	lon
▶	101	fraud_Sporer-Keebler	personal_care	40	-110
	104	fraud_Johnston-Casper	travel	44	-85
	105	fraud_Daugherty LLC	kids_pets	42	-77
	107	fraud_Reichel LLC	personal_care	44	-98
	111	fraud_Gottlieb Group	kids_pets	43	-109
	114	fraud_Lubowitz-Walter	kids_pets	33	-87
	115	fraud_Welch, Rath and Koepp	entertainment	29	-81
	117	fraud_Lang, Towne and Schuppe	kids_pets	32	-97
	118	fraud_Morissette LLC	entertainment	30	-91
	119	fraud_Prosacco LLC	personal_care	41	-80
	120	fraud_Corwin-Romaguera	travel	34	-84
	121	fraud_Gottlieb Group	kids_pets	39	-105
	122	fraud_Tillman LLC	travel	40	-85
	123	fraud_Vaum-Koelnin	travel	45	-90



- Display the details of the average amt of merchants where the amount of merchant by comparing each transaction with the current within the merchant id which is the highest.

```

109 • Select * From Merchant
110     INNER JOIN Transactions ON Transactions.Merchant_id = Merchant.Merchant_id
111     Where Transactions.amt = (Select MAX(m2.amt)
112                               From Transactions AS m2
113                               Where m2.Merchant_id = Transactions.Merchant_id);
114

```

100% 50:113

Result Grid Filter Rows: Search Export:

	merchant_id	merchant	category	lat	lon	trans_num	trans_date...	cc_num	amt	unix_time	merchant_id	is_fraud	customer_
▶	1085	fraud_Pollich LLC	home	35	-99	000245bf66e...	2020-06-21...	4.31958E+18	106	1371837096	1085	0	110645
	161	fraud_White and Sons	home	34	-118	0245fb8523d...	2020-06-21...	3.40103E+14	35	1371825932	161	0	110441
	535	fraud_Schoen Ltd	kids_pets	44	-89	027bff6ecc28...	2020-06-21...	3.0273E+13	64	1371839300	535	0	110322
	357	fraud_Zboncak Ltd	food_dining	43	-90	02b71d98b7b...	2020-06-21...	3.75248E+14	101	1371822177	357	0	110186
	645	fraud_Jakubowski Group	food_dining	40	-89	02e57dd4582...	2020-06-21...	3.58364E+15	156	1371841463	645	0	110475
	1634	fraud_Miller-Harris	misc_net	44	-68	041e573c99f...	2020-06-28...	3.58113E+15	1197	1372463331	1634	1	110764
	935	fraud_Langworth LLC	personal_care	32	-99	04c93c46fc3...	2020-06-21...	3.56032E+15	29	1371834221	935	0	110532
	458	fraud_Parker-Kunde	personal_care	43	-109	053add4e5ab...	2020-06-21...	1.80014E+14	36	1371841906	458	0	110497
	1756	fraud_Bradtke PLC	grocery_pos	34	-82	05668abc68d...	2020-07-05...	4.98985E+18	137	1372992298	1756	1	110576
	460	fraud_Yost-Rogahn	personal_care	37	-87	067962ea69e...	2020-06-21...	4.70787E+15	197	1371832127	460	0	110478
	134	fraud_Lynch-Wisozk	home	40	-77	077282675bd...	2020-06-21...	1.80043E+14	112	1371828074	134	0	110181
	774	fraud_Jakubowski Inc	home	41	-89	078da9a1497...	2020-06-30...	3.52823E+15	359	1372635952	774	1	110766
	649	fraud_Reichel Inc	entertainment	41	-85	07b83334404...	2020-06-21...	3.50097E+15	122	1371827946	649	0	110301
	1421	fraud_Hermann and Sons	shopping_pos	38	-80	08168f8f22d7...	2020-06-21...	6.53889E+15	85	1371843973	1421	0	110617

- Display the details of the transactions where the total amt is more than the average amt of that customer

```
118 • select * from transactions as t1
119     where amt > (select avg(amt)
120                  from transactions as t2
121                  where t1.customer_id= t2.customer_id);|
122
```

100% 39:121

**Result Grid** Filter Rows: Search Edit: Export/Import:

	trans_num	trans_date...	cc_num	amt	unix_time	merchant_id	is_fraud	customer_id
▶	000245bf66e...	2020-06-21...	4.31958E+18	106	1371837096	1085	0	110645
	0143364c291...	2020-06-21...	4.90063E+12	7	1371845270	641	0	110228
	014fc2cec14...	2020-06-21...	3.51237E+15	39	1371840623	569	0	110295
	019bacbe617...	2020-06-21...	3.43746E+14	68	1371846901	755	0	110275
	01de2fec50...	2020-06-21...	3.76445E+14	27	1371819319	140	0	110234
	02b71d98b7b...	2020-06-21...	3.75248E+14	101	1371822177	357	0	110186
	02e57dd4582...	2020-06-21...	3.58364E+15	156	1371841463	645	0	110475
	041e573c99f...	2020-06-28...	3.58113E+15	1197	1372463331	1634	1	110764
	04c93c46fc3...	2020-06-21...	3.56032E+15	29	1371834221	935	0	110532
	06250075f7b...	2020-06-21...	4.58513E+12	42	1371831093	492	0	110421
	067962ea69e...	2020-06-21...	4.70787E+15	197	1371832127	460	0	110478
	069013b8389...	2020-06-21...	4.14924E+15	75	1371818302	175	0	110182
	07105504fc0...	2020-06-21...	4.16976E+15	47	1371834567	142	0	110439
	077282675bd...	2020-06-21...	1.80043E+14	112	1371828074	134	0	110181

- Can you create a common table expression to calculate the total transaction amount for each customer?

```
126 • With TotalTransactionAmount As (  
127     Select  
128         customer_id,  
129         SUM(amt) AS total_amount  
130     From  
131         Transactions  
132     Group By  
133         customer_id )  
134     Select  
135         customer_id,  
136         total_amount  
137     From TotalTransactionAmount;
```

100% 29:137

**Result Grid** Filter Rows: Search Export:

	customer_id	total_amount
▶	110645	183
▶	110580	6852
▶	110368	61
▶	110262	405
▶	110169	192
▶	110125	458
▶	110228	24
▶	110295	239
▶	110113	44

- Can you create a user-defined function to calculate the age of a customer based on their date of birth?

```
150 DELIMITER //
```

```
151 • Create Function getAge(dob date)
```

```
152     Returns Date
```

```
153     Deterministic
```

```
154     Begin
```

```
155         return year(curdate()) - year(dob);
```

```
156     End//
```




```
157 DELIMITER ;
```

```
158
```

```
159 • select getAge(dob) from customer;
```

```
160
```

12:157

Result Grid   Filter Rows:  Export: 

	getA...	
▶ 34		
53		
36		
68		
32		
73		
50		

- Are there any geographic regions or merchant categories with a higher risk of fraud?

```
221 • Select c.state, m.category,  
222 COUNT(*) As num_fraudulent_transactions,  
223 COUNT(*) * 100.0 / (SELECT COUNT(*) From transactions Where is_fraud = 1) As fraud_percentage  
224 From transactions t  
225 Join merchant m On t.Merchant_id = m.Merchant_id  
226 Join customer c On t.customer_id = c.customer_id  
227 Where t.is_fraud = 1  
228 Group by c.state, m.category  
229 Order by fraud_percentage Desc;
```

100% 32:229

Result Grid Filter Rows: Search Export:

	state	category	num_fraudulent_transacti...	fraud_percenta...
▶	SD	grocery_pos	6	2.23881
▶	NC	grocery_pos	5	1.86567
▶	WA	grocery_pos	5	1.86567
▶	NC	shopping_net	5	1.86567
▶	NJ	grocery_pos	5	1.86567
▶	NY	grocery_pos	4	1.49254
▶	MD	grocery_pos	4	1.49254

- Explain how you would use a trigger to update the total transaction count for a customer whenever a new transaction is inserted.

```
259 DELIMITER //
```

```
260
```

```
261 • Create Trigger update_trans_num
```

```
262 After Insert On Transactions
```

```
263 For Each Row
```

```
264 Begin
```

```
265     Update Customer_id
```

```
266     Set trans_num = trans_num + 1
```

```
267     Where customer_id = New.customer_id;
```

```
268 END;
```

```
269 //
```

```
270
```

```
271 DELIMITER ;
```



- Creating an index to know the maximum amt of transactions

```
274
275     Create Index idx_max_amt On transactions(amt);
276
277 •   Select MAX(amt) From transactions;
278
279 •   Drop index idx_max_amt On transactions;
```

- Are there any unusual patterns in the time of day or day of the week For fraudulent transactions?

```
184 • Select Hour(trans_date_trans_time) As hour_of_day,
185      Dayofweek(trans_date_trans_time) As day_of_week,
186      COUNT(*) As num_fraudulent_transactions
187      From transactions
188      Where is_fraud = 1
189      Group by Hour(trans_date_trans_time), Dayofweek(trans_date_trans_time)
190      Order by num_fraudulent_transactions Desc;
```

100% 1:183








Result Grid Filter Rows: Search Export:

	hour_of_day	day_of_week	num_fraudulent_transacti...
▶	23	1	19
	23	2	13
	22	1	13
	23	6	13
	23	7	11
	3	7	9
	3	3	9

- Display the customers where the fraud happened

```
67 • select * from transactions
68   where Is_fraud = 1 ;
69
```

21:68

**Result Grid**   Filter Rows:  Edit:    Export/Import:  

	trans_num	trans_date...	cc_num	amt	unix_time	merchant_id	is_fraud	customer_id	
▶	00085870bed...	2020-06-30...	4.7614E+18	19	1372631545	438	1	110580	
◀	01bbbf7a906...	2020-06-26...	2.13161E+14	8	1372226213	428	1	110395	
	041e573c99f...	2020-06-28...	3.58113E+15	1197	1372463331	1634	1	110764	
◀	05668abc68d...	2020-07-05...	4.98985E+18	137	1372992298	1756	1	110576	
	078da9a1497...	2020-06-30...	3.52823E+15	359	1372635952	774	1	110766	
◀	0a08fed1b67...	2020-07-11...	3.85443E+13	731	1373515139	1824	1	110662	
	0bac4292002...	2020-06-21...	3.52457E+15	23	1371856572	1421	1	110762	