Credit Fraud / Credit Score Analysis

Objective:
One of our clients, a credit lender in the USA, has shared some CSV files containing historical data as well as customer information with us. Our team is required to work on this data to extract some meaning full insights from the shared data so that the bank can understand further about the credit scores of their customers as well as the credit fraud being committed.
You'll create an automated dashboard to monitor fraud trends and generate customer credit score reports.

SQL: Write complex SQL queries to:

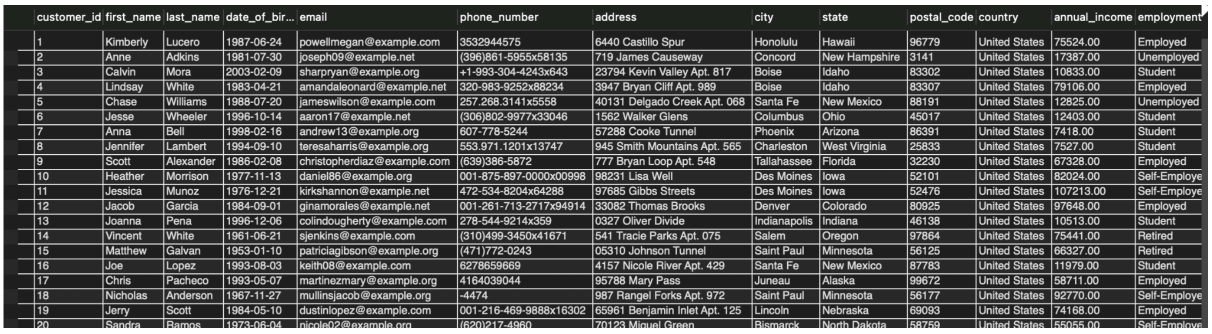
- Extract customer and transaction data based on defined features (e.g., monthly transaction totals, outlier detection on spending, etc.).
- Perform joins between customers, transactions, accounts, and credit_history to gather complete data per customer.
- Generate new features such as:
    - Average transaction amounts by time periods (daily/weekly).
    - Transaction location consistency (using geographical data).
    - Delinquency rates from credit_history.

Initial setting up

-- Create Customers table
```
CREATE TABLE Customers (
    customer_id INT PRIMARY KEY,
    first_name VARCHAR(50),
    last_name VARCHAR(50),
    date_of_birth DATE,
    email VARCHAR(100),
    phone_number VARCHAR(100),
    address VARCHAR(255),
    city VARCHAR(50),
    state VARCHAR(50),
    postal_code VARCHAR(10),
    country VARCHAR(50),
    annual_income DECIMAL(20, 2),
    employment_status VARCHAR(50),
    account_open_date DATE,
    credit_score INT
);
```

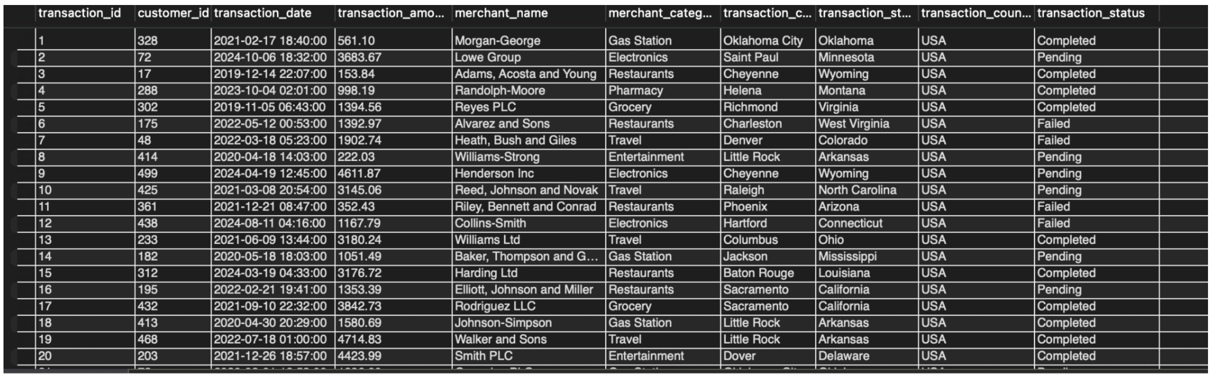
-- Corrected the date format in excel using 'Text to Columns' option and imported data to the excel table

```
select  * from Customers;
```



customer_id	first_name	last_name	date_of_bir...	email	phone_number	address	city	state	postal_code	country	annual_income	employment
1	Kimberly	Lucero	1987-06-24	powellmegan@example.com	3532944575	6440 Castillo Spur	Honolulu	Hawaii	96779	United States	75524.00	Employed
2	Anne	Adkins	1981-07-30	joseph09@example.net	(396)861-5955x58135	719 James Causeway	Concord	New Hampshire	3141	United States	17387.00	Unemployed
3	Calvin	Mora	2003-02-09	sharpryan@example.org	+1-993-304-4243x643	23794 Kevin Valley Apt. 817	Boise	Idaho	83302	United States	10833.00	Student
4	Lindsay	White	1983-04-21	amandaleonard@example.net	320-983-9252x88234	3947 Bryan Cliff Apt. 989	Boise	Idaho	83307	United States	79106.00	Employed
5	Chase	Williams	1988-07-20	jameswilson@example.com	257.268.3141x5558	40131 Delgado Creek Apt. 068	Santa Fe	New Mexico	88191	United States	12825.00	Unemployed
6	Jesse	Wheeler	1996-10-14	aaron17@example.net	(306)802-9977x33046	1562 Walker Glens	Columbus	Ohio	45017	United States	12403.00	Student
7	Anna	Bell	1998-02-16	andrew13@example.org	607-778-5244	57288 Cooke Tunnel	Phoenix	Arizona	86391	United States	7418.00	Student
8	Jennifer	Lambert	1994-09-10	teresaharris@example.org	553.971.1201x13747	945 Smith Mountains Apt. 565	Charleston	West Virginia	25833	United States	7527.00	Student
9	Scott	Alexander	1986-02-08	christopherdiaz@example.com	(639)386-5872	777 Bryan Loop Apt. 548	Tallahassee	Florida	32230	United States	67328.00	Employed
10	Heather	Morrison	1977-11-13	daniel86@example.org	001-875-897-0000x00998	98231 Lisa Well	Des Moines	Iowa	52101	United States	82024.00	Self-Employe
11	Jessica	Munoz	1976-12-21	kirkshannon@example.net	472-534-8204x64288	97685 Gibbs Streets	Des Moines	Iowa	52476	United States	107213.00	Self-Employe
12	Jacob	Garcia	1984-09-01	ginamorales@example.net	001-261-713-2717x94914	33082 Thomas Brooks	Denver	Colorado	80925	United States	97648.00	Employed
13	Joanna	Pena	1996-12-06	colindougherty@example.com	278-544-9214x359	0327 Oliver Divide	Indianapolis	Indiana	46138	United States	10513.00	Student
14	Vincent	White	1961-06-21	sjenkins@example.com	(310)499-3450x41671	541 Tracie Parks Apt. 075	Salem	Oregon	97864	United States	75441.00	Retired
15	Matthew	Galvan	1953-01-10	patriciagibson@example.org	(471)772-0243	05310 Johnson Tunnel	Saint Paul	Minnesota	56125	United States	66327.00	Retired
16	Joe	Lopez	1993-08-03	keith08@example.com	6278659669	4157 Nicole River Apt. 429	Santa Fe	New Mexico	87783	United States	11979.00	Student
17	Chris	Pacheco	1993-05-07	martinezmary@example.org	4164039044	95788 Mary Pass	Juneau	Alaska	99672	United States	58711.00	Employed
18	Nicholas	Anderson	1967-11-27	mullinsjacob@example.org	-4474	987 Rangel Forks Apt. 972	Saint Paul	Minnesota	56177	United States	92770.00	Self-Employe
19	Jerry	Scott	1984-05-10	dustinlopez@example.com	001-216-469-9888x16302	65961 Benjamin Inlet Apt. 125	Lincoln	Nebraska	69093	United States	74168.00	Employed
20	Sandra	Ramos	1973-06-04	nicole02@example.org	(620)217-4960	70123 Miguel Green	Bismarck	North Dakota	58759	United States	55055.00	Self-Employe

```
-- Create Transactions table
CREATE TABLE Transactions (
    transaction_id INT PRIMARY KEY,
    customer_id INT,
    transaction_date DATETIME,
    transaction_amount DECIMAL(15, 2),
    merchant_name VARCHAR(100),
    merchant_category VARCHAR(50),
    transaction_city VARCHAR(50),
    transaction_state VARCHAR(50),
    transaction_country VARCHAR(50),
    transaction_status VARCHAR(20)
);
-- corrected date and time values in Excel and imported
select  * from transactions;
```

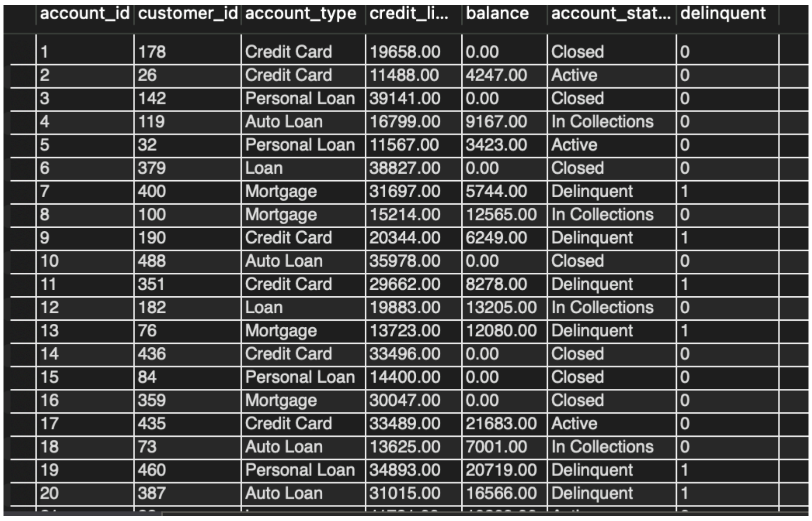


transaction_id	customer_id	transaction_date	transaction_amo...	merchant_name	merchant_categ...	transaction_c...	transaction_st...	transaction_coun...	transaction_status
1	328	2021-02-17 18:40:00	561.10	Morgan-George	Gas Station	Oklahoma City	Oklahoma	USA	Completed
2	72	2024-10-06 18:32:00	3683.67	Lowe Group	Electronics	Saint Paul	Minnesota	USA	Pending
3	17	2019-12-14 22:07:00	153.84	Adams, Acosta and Young	Restaurants	Cheyenne	Wyoming	USA	Completed
4	288	2023-10-04 02:01:00	998.19	Randolph-Moore	Pharmacy	Helena	Montana	USA	Completed
5	302	2019-11-05 06:43:00	1394.56	Reyes PLC	Grocery	Richmond	Virginia	USA	Completed
6	175	2022-05-12 00:53:00	1392.97	Alvarez and Sons	Restaurants	Charleston	West Virginia	USA	Failed
7	48	2022-03-18 05:23:00	1902.74	Heath, Bush and Giles	Travel	Denver	Colorado	USA	Failed
8	414	2020-04-18 14:03:00	222.03	Williams-Strong	Entertainment	Little Rock	Arkansas	USA	Pending
9	499	2024-04-19 12:45:00	4611.87	Henderson Inc	Electronics	Cheyenne	Wyoming	USA	Pending
10	425	2021-03-08 20:54:00	3145.06	Reed, Johnson and Novak	Travel	Raleigh	North Carolina	USA	Pending
11	361	2021-12-21 08:47:00	352.43	Riley, Bennett and Conrad	Restaurants	Phoenix	Arizona	USA	Failed
12	438	2024-08-11 04:16:00	1167.79	Collins-Smith	Electronics	Hartford	Connecticut	USA	Failed
13	233	2021-06-09 13:44:00	3180.24	Williams Ltd	Travel	Columbus	Ohio	USA	Completed
14	182	2020-05-18 18:03:00	1051.49	Baker, Thompson and G…	Gas Station	Jackson	Mississippi	USA	Pending
15	312	2024-03-19 04:33:00	3176.72	Harding Ltd	Restaurants	Baton Rouge	Louisiana	USA	Completed
16	195	2022-02-21 19:41:00	1353.39	Elliott, Johnson and Miller	Restaurants	Sacramento	California	USA	Pending
17	432	2021-09-10 22:32:00	3842.73	Rodriguez LLC	Grocery	Sacramento	California	USA	Completed
18	413	2020-04-30 20:29:00	1580.69	Johnson-Simpson	Gas Station	Little Rock	Arkansas	USA	Completed
19	468	2022-07-18 01:00:00	4714.83	Walker and Sons	Travel	Little Rock	Arkansas	USA	Completed
20	203	2021-12-26 18:57:00	4423.99	Smith PLC	Entertainment	Dover	Delaware	USA	Completed

```
-- Create Accounts table
CREATE TABLE Accounts (
    account_id INT PRIMARY KEY,
    customer_id INT,
    account_type VARCHAR(50),
    credit_limit DECIMAL(15, 2),
    balance DECIMAL(15, 2),
    account_status VARCHAR(20),
    delinquent BOOLEAN
);
```

-- converted delinquent boolean values to 0 & 1 in excel and imported

```
select * from accounts;
```



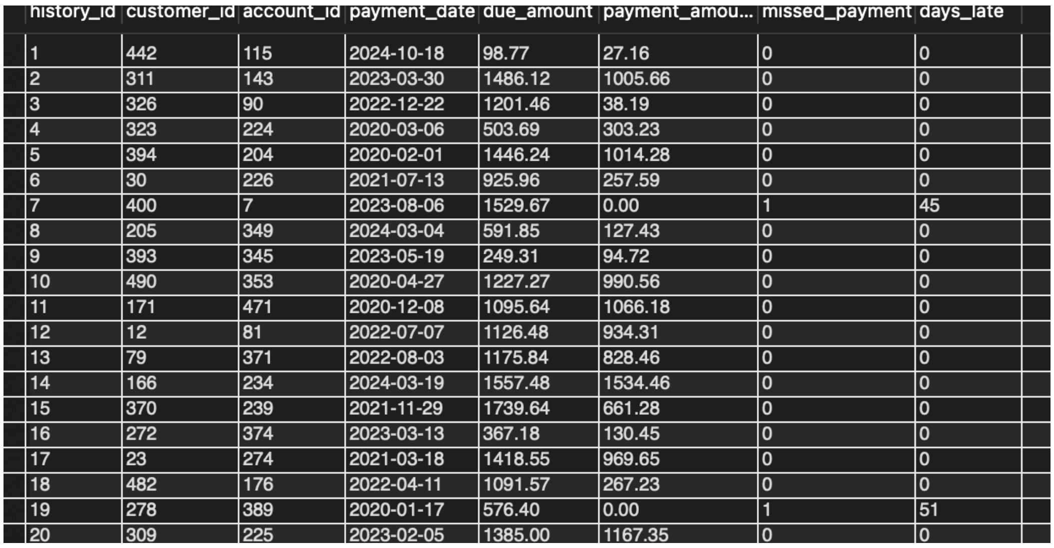
account_id	customer_id	account_type	credit_li...	balance	account_stat...	delinquent
1	178	Credit Card	19658.00	0.00	Closed	0
2	26	Credit Card	11488.00	4247.00	Active	0
3	142	Personal Loan	39141.00	0.00	Closed	0
4	119	Auto Loan	16799.00	9167.00	In Collections	0
5	32	Personal Loan	11567.00	3423.00	Active	0
6	379	Loan	38827.00	0.00	Closed	0
7	400	Mortgage	31697.00	5744.00	Delinquent	1
8	100	Mortgage	15214.00	12565.00	In Collections	0
9	190	Credit Card	20344.00	6249.00	Delinquent	1
10	488	Auto Loan	35978.00	0.00	Closed	0
11	351	Credit Card	29662.00	8278.00	Delinquent	1
12	182	Loan	19883.00	13205.00	In Collections	0
13	76	Mortgage	13723.00	12080.00	Delinquent	1
14	436	Credit Card	33496.00	0.00	Closed	0
15	84	Personal Loan	14400.00	0.00	Closed	0
16	359	Mortgage	30047.00	0.00	Closed	0
17	435	Credit Card	33489.00	21683.00	Active	0
18	73	Auto Loan	13625.00	7001.00	In Collections	0
19	460	Personal Loan	34893.00	20719.00	Delinquent	1
20	387	Auto Loan	31015.00	16566.00	Delinquent	1

-- Create Credit_History table

```
CREATE TABLE Credit_History (
    history_id INT PRIMARY KEY,
    customer_id INT,
    account_id INT,
    payment_date DATE,
    due_amount DECIMAL(15, 2),
    payment_amount DECIMAL(15, 2),
    missed_payment BOOLEAN,
    days_late INT
);
```

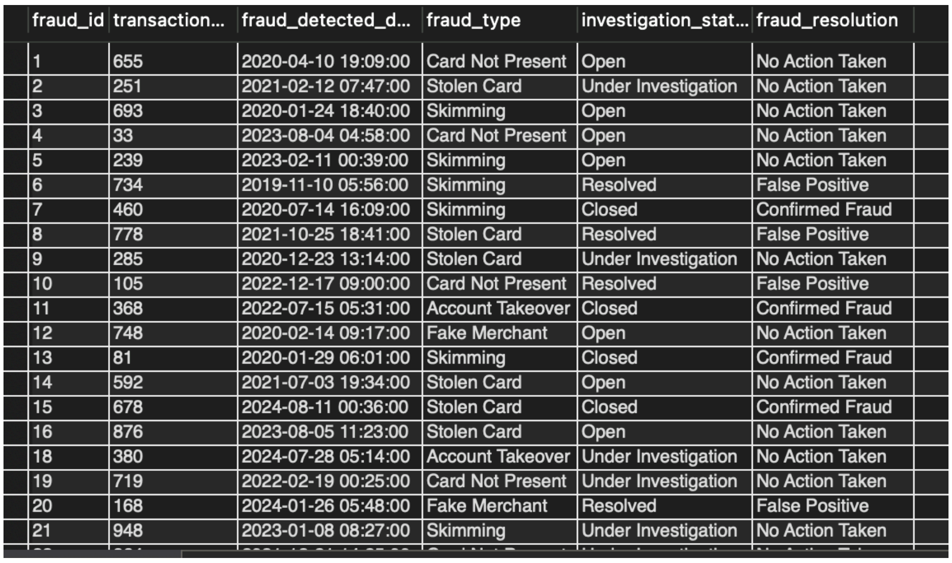
-- Converted payment_date and missed_payment column to correct formats in excel and imported

```
select * from credit_history;
```



history_id	customer_id	account_id	payment_date	due_amount	payment_amou...	missed_payment	days_late
1	442	115	2024-10-18	98.77	27.16	0	0
2	311	143	2023-03-30	1486.12	1005.66	0	0
3	326	90	2022-12-22	1201.46	38.19	0	0
4	323	224	2020-03-06	503.69	303.23	0	0
5	394	204	2020-02-01	1446.24	1014.28	0	0
6	30	226	2021-07-13	925.96	257.59	0	0
7	400	7	2023-08-06	1529.67	0.00	1	45
8	205	349	2024-03-04	591.85	127.43	0	0
9	393	345	2023-05-19	249.31	94.72	0	0
10	490	353	2020-04-27	1227.27	990.56	0	0
11	171	471	2020-12-08	1095.64	1066.18	0	0
12	12	81	2022-07-07	1126.48	934.31	0	0
13	79	371	2022-08-03	1175.84	828.46	0	0
14	166	234	2024-03-19	1557.48	1534.46	0	0
15	370	239	2021-11-29	1739.64	661.28	0	0
16	272	374	2023-03-13	367.18	130.45	0	0
17	23	274	2021-03-18	1418.55	969.65	0	0
18	482	176	2022-04-11	1091.57	267.23	0	0
19	278	389	2020-01-17	576.40	0.00	1	51
20	309	225	2023-02-05	1385.00	1167.35	0	0

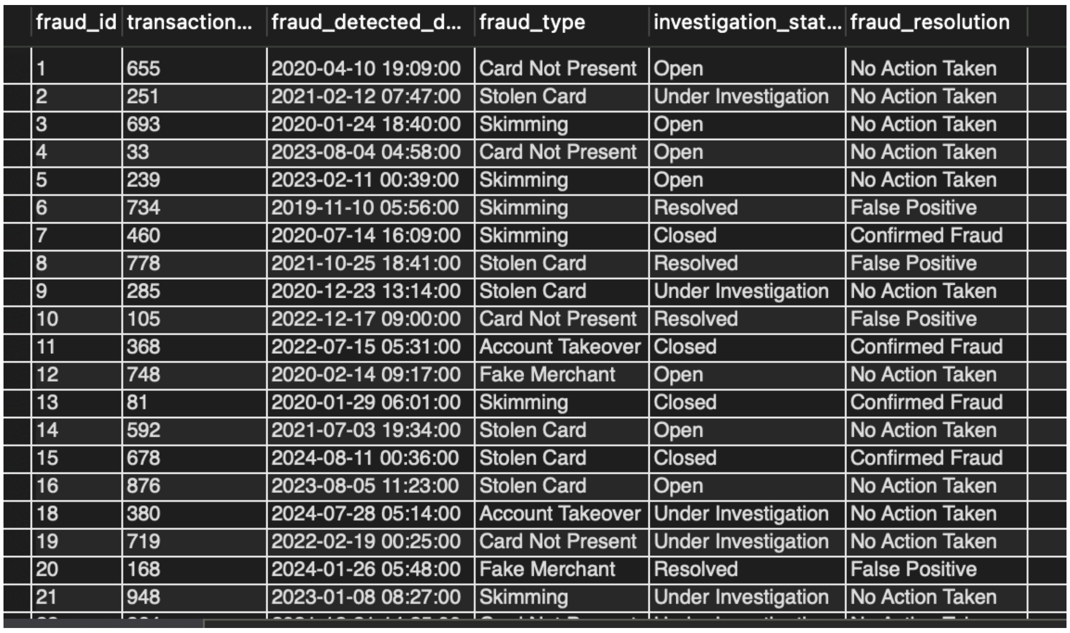
```
-- Create Fraud_Records table
CREATE TABLE Fraud_Records (
    fraud_id INT PRIMARY KEY,
    transaction_id INT,
    fraud_detected_date DATETIME,
    fraud_type VARCHAR(50),
    investigation_status VARCHAR(50),
    fraud_resolution VARCHAR(50)
);

-- converted date and time values in excel and imported
select * from fraud_records;
```



fraud_id	transaction...	fraud_detected_d...	fraud_type	investigation_stat...	fraud_resolution
1	655	2020-04-10 19:09:00	Card Not Present	Open	No Action Taken
2	251	2021-02-12 07:47:00	Stolen Card	Under Investigation	No Action Taken
3	693	2020-01-24 18:40:00	Skimming	Open	No Action Taken
4	33	2023-08-04 04:58:00	Card Not Present	Open	No Action Taken
5	239	2023-02-11 00:39:00	Skimming	Open	No Action Taken
6	734	2019-11-10 05:56:00	Skimming	Resolved	False Positive
7	460	2020-07-14 16:09:00	Skimming	Closed	Confirmed Fraud
8	778	2021-10-25 18:41:00	Stolen Card	Resolved	False Positive
9	285	2020-12-23 13:14:00	Stolen Card	Under Investigation	No Action Taken
10	105	2022-12-17 09:00:00	Card Not Present	Resolved	False Positive
11	368	2022-07-15 05:31:00	Account Takeover	Closed	Confirmed Fraud
12	748	2020-02-14 09:17:00	Fake Merchant	Open	No Action Taken
13	81	2020-01-29 06:01:00	Skimming	Closed	Confirmed Fraud
14	592	2021-07-03 19:34:00	Stolen Card	Open	No Action Taken
15	678	2024-08-11 00:36:00	Stolen Card	Closed	Confirmed Fraud
16	876	2023-08-05 11:23:00	Stolen Card	Open	No Action Taken
18	380	2024-07-28 05:14:00	Account Takeover	Under Investigation	No Action Taken
19	719	2022-02-19 00:25:00	Card Not Present	Under Investigation	No Action Taken
20	168	2024-01-26 05:48:00	Fake Merchant	Resolved	False Positive
21	948	2023-01-08 08:27:00	Skimming	Under Investigation	No Action Taken

1. Extract customer and transaction data based on defined features (e.g., monthly transaction totals, outlier detection on spending, etc.)

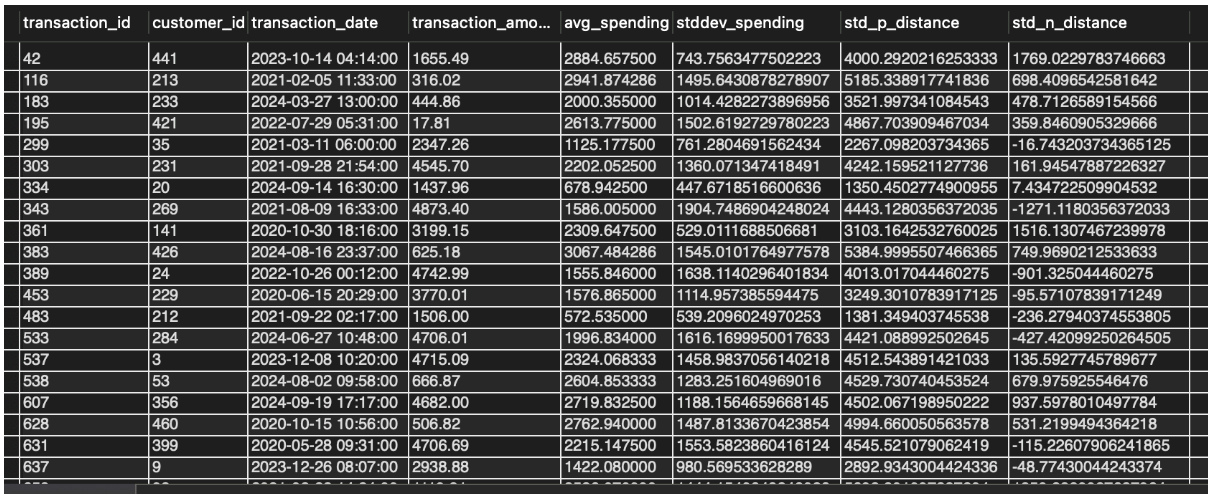
```
-- monthly transaction totals
select
    c.customer_id,
    c.first_name,
    c.last_name,
    year(t.transaction_date) as transaction_year,
    monthname(t.transaction_date) as transaction_month,
    sum(t.transaction_amount) as total_transaction
from
    customers c
join
    transactions t on c.customer_id = t.customer_id
group by
    c.customer_id, transaction_year, transaction_month
order by
    transaction_year, transaction_month, c.customer_id;
```



fraud_id	transaction...	fraud_detected_d...	fraud_type	investigation_stat...	fraud_resolution
1	655	2020-04-10 19:09:00	Card Not Present	Open	No Action Taken
2	251	2021-02-12 07:47:00	Stolen Card	Under Investigation	No Action Taken
3	693	2020-01-24 18:40:00	Skimming	Open	No Action Taken
4	33	2023-08-04 04:58:00	Card Not Present	Open	No Action Taken
5	239	2023-02-11 00:39:00	Skimming	Open	No Action Taken
6	734	2019-11-10 05:56:00	Skimming	Resolved	False Positive
7	460	2020-07-14 16:09:00	Skimming	Closed	Confirmed Fraud
8	778	2021-10-25 18:41:00	Stolen Card	Resolved	False Positive
9	285	2020-12-23 13:14:00	Stolen Card	Under Investigation	No Action Taken
10	105	2022-12-17 09:00:00	Card Not Present	Resolved	False Positive
11	368	2022-07-15 05:31:00	Account Takeover	Closed	Confirmed Fraud
12	748	2020-02-14 09:17:00	Fake Merchant	Open	No Action Taken
13	81	2020-01-29 06:01:00	Skimming	Closed	Confirmed Fraud
14	592	2021-07-03 19:34:00	Stolen Card	Open	No Action Taken
15	678	2024-08-11 00:36:00	Stolen Card	Closed	Confirmed Fraud
16	876	2023-08-05 11:23:00	Stolen Card	Open	No Action Taken
18	380	2024-07-28 05:14:00	Account Takeover	Under Investigation	No Action Taken
19	719	2022-02-19 00:25:00	Card Not Present	Under Investigation	No Action Taken
20	168	2024-01-26 05:48:00	Fake Merchant	Resolved	False Positive
21	948	2023-01-08 08:27:00	Skimming	Under Investigation	No Action Taken

```
-- outlier detection on spending
with TransactionStats as(
     select customer_id,
          avg(transaction_amount) as avg_spending,
          stddev(transaction_amount) as stddev_spending
    from
          Transactions
    group by
          customer_id
)
select t.transaction_id,
     t.customer_id,
    t.transaction_date,
    t.transaction_amount,
    ts.avg_spending,
    ts.stddev_spending,
    ts.avg_spending + (1.5 * ts.stddev_spending) as
std_p_distance,
    ts.avg_spending - (1.5 * ts.stddev_spending) as std_n_distance
from
     Transactions as t
join
     TransactionStats as ts
on
     t.customer_id = ts.customer_id
where
     t.transaction_amount > (ts.avg_spending + (1.5 *
ts.stddev_spending))
```

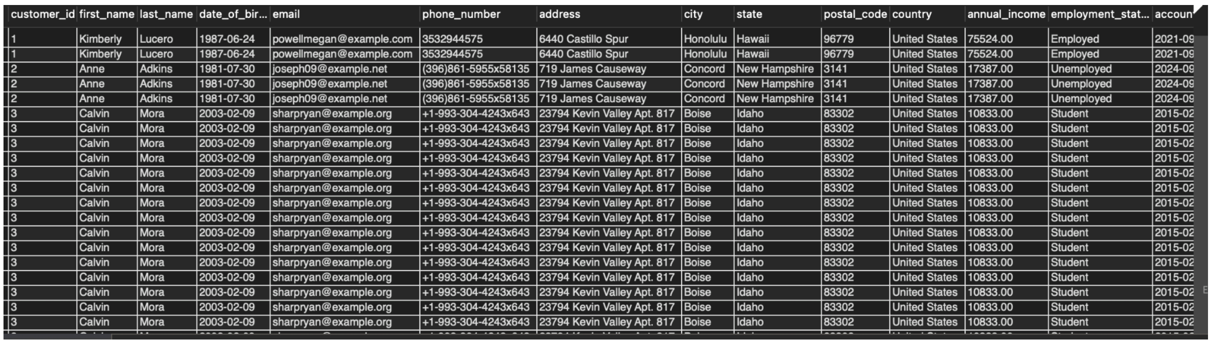
```
    or t.transaction_amount < (ts.avg_spending - (1.5 *
ts.stddev_spending));
```



transaction_id	customer_id	transaction_date	transaction_amo...	avg_spending	stddev_spending	std_p_distance	std_n_distance
42	441	2023-10-14 04:14:00	1655.49	2884.657500	743.7563477502223	4000.2920216253333	1769.0229783746663
116	213	2021-02-05 11:33:00	316.02	2941.874286	1495.6430878278907	5185.338917741836	698.4096542581642
183	233	2024-03-27 13:00:00	444.86	2000.355000	1014.4282273896956	3521.997341084543	478.7126589154566
195	421	2022-07-29 05:31:00	17.81	2613.775000	1502.6192729780223	4867.703909467034	359.8460905329666
299	35	2021-03-11 06:00:00	2347.26	1125.177500	761.2804691562434	2267.098203734365	-16.743203734365125
303	231	2021-09-28 21:54:00	4545.70	2202.052500	1360.071347418491	4242.159521127736	161.94547887226327
334	20	2024-09-14 16:30:00	1437.96	678.942500	447.6718516600636	1350.4502774900955	7.434722509904532
343	269	2021-08-09 16:33:00	4873.40	1586.005000	1904.7486904248024	4443.1280356372035	-1271.1180356372033
361	141	2020-10-30 18:16:00	3199.15	2309.647500	529.0111688506681	3103.1642532760025	1516.1307467239978
383	426	2024-08-16 23:37:00	625.18	3067.484286	1545.0101764977578	5384.9995507466365	749.9690212533633
389	24	2022-10-26 00:12:00	4742.99	1555.846000	1638.1140296401834	4013.017044460275	-901.325044460275
453	229	2020-06-15 20:29:00	3770.01	1576.865000	1114.957385594475	3249.3010783917125	-95.57107839171249
483	212	2021-09-22 02:17:00	1506.00	572.535000	539.2096024970253	1381.349403745538	-236.27940374553805
533	284	2024-06-27 10:48:00	4706.01	1996.834000	1616.1699950017633	4421.088992502645	-427.42099250264505
537	3	2023-12-08 10:20:00	4715.09	2324.068333	1458.9837056140218	4512.543891421033	135.5927745789677
538	53	2024-08-02 09:58:00	666.87	2604.853333	1283.251604969016	4529.730740453524	679.975925546476
607	356	2024-09-19 17:17:00	4682.00	2719.832500	1188.1564659668145	4502.067198950222	937.5978010497784
628	460	2020-10-15 10:56:00	506.82	2762.940000	1487.8133670423854	4994.660050563578	531.2199494364218
631	399	2020-05-28 09:31:00	4706.69	2215.147500	1553.5823860416124	4545.521079062419	-115.22607906241865
637	9	2023-12-26 08:07:00	2938.88	1422.080000	980.569533628289	2892.9343004424336	-48.77430044243374

2. Perform joins between customers, transactions, accounts, and credit_history to gather complete data per customer.

```
select

     c.customer_id,c.first_name,c.last_name,c.date_of_birth,c.emai
     l,c.phone_number,c.address,c.city,c.state,c.postal_code,c.cou
     ntry,c.annual_income,c.employment_status,c.account_open_date,
     c.credit_score,
     a.account_id,a.account_type,a.credit_limit,a.balance,a.accoun
     t_status,a.delinquent,
     t.transaction_id,t.transaction_date,t.transaction_amount,t.me
     rchant_name,t.merchant_category,t.transaction_city,t.transact
     ion_state,t.transaction_country,t.transaction_status,
     ch.history_id,ch.payment_date,ch.due_amount,ch.payment_amount
     ,ch.missed_payment,ch.days_late
from
     customers as c
left join
     accounts as a on c.customer_id=a.customer_id
left join
     transactions as t on c.customer_id=t.customer_id
left join
     credit_history as ch on c.customer_id=ch.customer_id and
a.account_id=ch.account_id
order by
     c.customer_id, t.transaction_date, ch.payment_date;
```

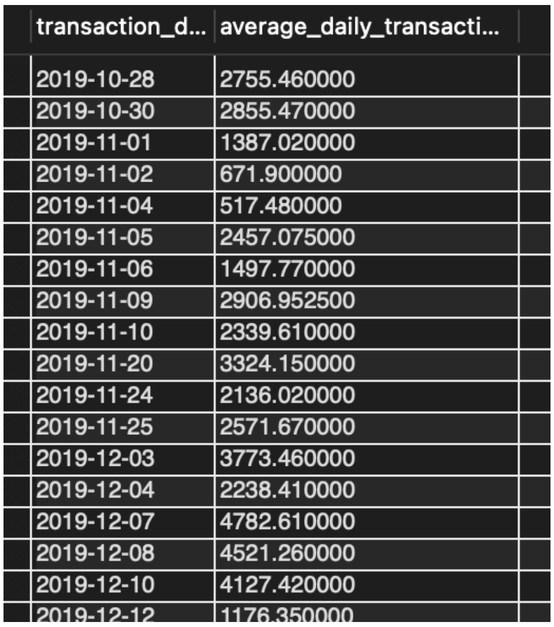


customer_id	first_name	last_name	date_of_bir...	email	phone_number	address	city	state	postal_code	country	annual_income	employment_stat...	accoun
1	Kimberly	Lucero	1987-06-24	powellmegan@example.com	3532944575	6440 Castillo Spur	Honolulu	Hawaii	96779	United States	75524.00	Employed	2021-09
1	Kimberly	Lucero	1987-06-24	powellmegan@example.com	3532944575	6440 Castillo Spur	Honolulu	Hawaii	96779	United States	75524.00	Employed	2021-09
2	Anne	Adkins	1981-07-30	joseph09@example.net	(396)861-5955x58135	719 James Causeway	Concord	New Hampshire	3141	United States	17387.00	Unemployed	2024-09
2	Anne	Adkins	1981-07-30	joseph09@example.net	(396)861-5955x58135	719 James Causeway	Concord	New Hampshire	3141	United States	17387.00	Unemployed	2024-09
2	Anne	Adkins	1981-07-30	joseph09@example.net	(396)861-5955x58135	719 James Causeway	Concord	New Hampshire	3141	United States	17387.00	Unemployed	2024-09
3	Calvin	Mora	2003-02-09	sharpryan@example.org	+1-993-304-4243x643	23794 Kevin Valley Apt. 817	Boise	Idaho	83302	United States	10833.00	Student	2015-02
3	Calvin	Mora	2003-02-09	sharpryan@example.org	+1-993-304-4243x643	23794 Kevin Valley Apt. 817	Boise	Idaho	83302	United States	10833.00	Student	2015-02
3	Calvin	Mora	2003-02-09	sharpryan@example.org	+1-993-304-4243x643	23794 Kevin Valley Apt. 817	Boise	Idaho	83302	United States	10833.00	Student	2015-02
3	Calvin	Mora	2003-02-09	sharpryan@example.org	+1-993-304-4243x643	23794 Kevin Valley Apt. 817	Boise	Idaho	83302	United States	10833.00	Student	2015-02
3	Calvin	Mora	2003-02-09	sharpryan@example.org	+1-993-304-4243x643	23794 Kevin Valley Apt. 817	Boise	Idaho	83302	United States	10833.00	Student	2015-02
3	Calvin	Mora	2003-02-09	sharpryan@example.org	+1-993-304-4243x643	23794 Kevin Valley Apt. 817	Boise	Idaho	83302	United States	10833.00	Student	2015-02
3	Calvin	Mora	2003-02-09	sharpryan@example.org	+1-993-304-4243x643	23794 Kevin Valley Apt. 817	Boise	Idaho	83302	United States	10833.00	Student	2015-02
3	Calvin	Mora	2003-02-09	sharpryan@example.org	+1-993-304-4243x643	23794 Kevin Valley Apt. 817	Boise	Idaho	83302	United States	10833.00	Student	2015-02
3	Calvin	Mora	2003-02-09	sharpryan@example.org	+1-993-304-4243x643	23794 Kevin Valley Apt. 817	Boise	Idaho	83302	United States	10833.00	Student	2015-02
3	Calvin	Mora	2003-02-09	sharpryan@example.org	+1-993-304-4243x643	23794 Kevin Valley Apt. 817	Boise	Idaho	83302	United States	10833.00	Student	2015-02
3	Calvin	Mora	2003-02-09	sharpryan@example.org	+1-993-304-4243x643	23794 Kevin Valley Apt. 817	Boise	Idaho	83302	United States	10833.00	Student	2015-02
3	Calvin	Mora	2003-02-09	sharpryan@example.org	+1-993-304-4243x643	23794 Kevin Valley Apt. 817	Boise	Idaho	83302	United States	10833.00	Student	2015-02
3	Calvin	Mora	2003-02-09	sharpryan@example.org	+1-993-304-4243x643	23794 Kevin Valley Apt. 817	Boise	Idaho	83302	United States	10833.00	Student	2015-02
3	Calvin	Mora	2003-02-09	sharpryan@example.org	+1-993-304-4243x643	23794 Kevin Valley Apt. 817	Boise	Idaho	83302	United States	10833.00	Student	2015-02
3	Calvin	Mora	2003-02-09	sharpryan@example.org	+1-993-304-4243x643	23794 Kevin Valley Apt. 817	Boise	Idaho	83302	United States	10833.00	Student	2015-02

3. Generate new features such as:

-- Average transaction amounts by time periods - Daily

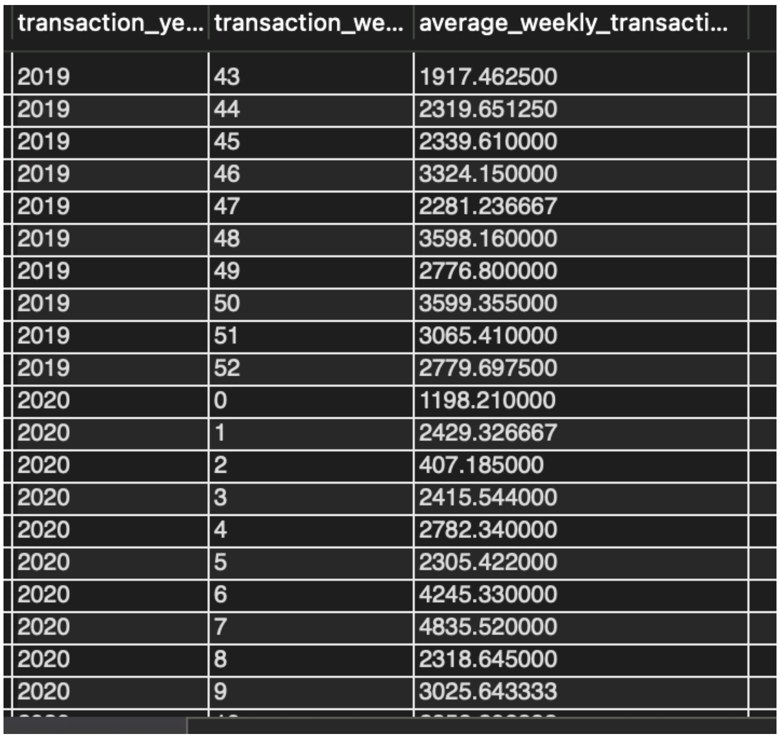
```
select
    date(transaction_date) as transaction_day,
    avg(transaction_amount) as average_daily_transaction
from
    transactions
group by
    transaction_day
order by
    Transaction_day;
```



transaction_d...	average_daily_transacti...
2019-10-28	2755.460000
2019-10-30	2855.470000
2019-11-01	1387.020000
2019-11-02	671.900000
2019-11-04	517.480000
2019-11-05	2457.075000
2019-11-06	1497.770000
2019-11-09	2906.952500
2019-11-10	2339.610000
2019-11-20	3324.150000
2019-11-24	2136.020000
2019-11-25	2571.670000
2019-12-03	3773.460000
2019-12-04	2238.410000
2019-12-07	4782.610000
2019-12-08	4521.260000
2019-12-10	4127.420000
2019-12-12	1176.350000

-- Average transaction amounts by time periods - Weekly

```
select
     year(transaction_date) as transaction_year,
    week(transaction_date) as transaction_week,
    avg(transaction_amount) as average_weekly_transaction
from
     transactions
group by
     transaction_year, transaction_week
order by
     transaction_year, transaction_week;
```

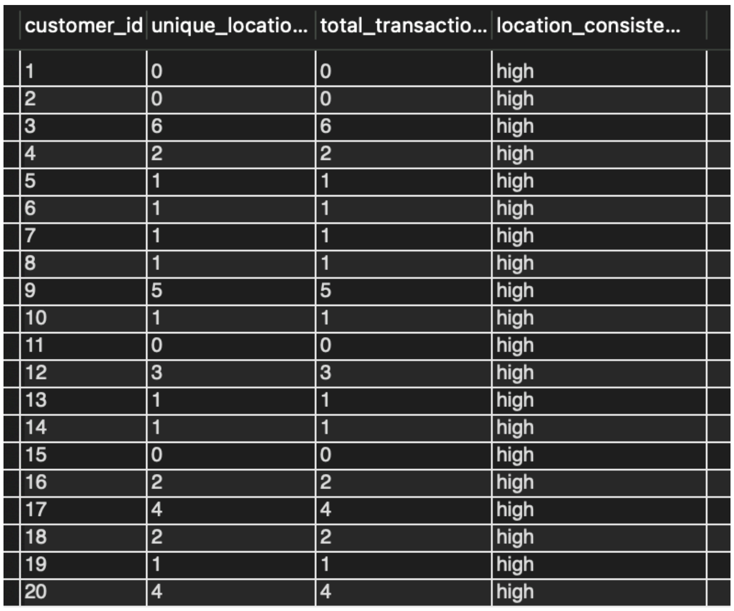


transaction_ye...	transaction_we...	average_weekly_transacti...
2019	43	1917.462500
2019	44	2319.651250
2019	45	2339.610000
2019	46	3324.150000
2019	47	2281.236667
2019	48	3598.160000
2019	49	2776.800000
2019	50	3599.355000
2019	51	3065.410000
2019	52	2779.697500
2020	0	1198.210000
2020	1	2429.326667
2020	2	407.185000
2020	3	2415.544000
2020	4	2782.340000
2020	5	2305.422000
2020	6	4245.330000
2020	7	4835.520000
2020	8	2318.645000
2020	9	3025.643333

-- Transaction location consistency

```
select
    c.customer_id,
    count(distinct concat(t.transaction_city, ', ',
t.transaction_state)) as unique_locations,
    count(t.transaction_id) as total_transactions,
    case
        when count(distinct concat(t.transaction_city, ', ',
t.transaction_state)) / count(t.transaction_id) < 0.5 then 'low'
        when count(distinct concat(t.transaction_city, ', ',
t.transaction_state)) / count(t.transaction_id) < 1 then 'medium'
        else 'high'
    end as location_consistency
```

```
from
    customers as c
left join
    transactions as t on c.customer_id = t.customer_id
group by
    c.customer_id
order by
    c.customer_id;
```

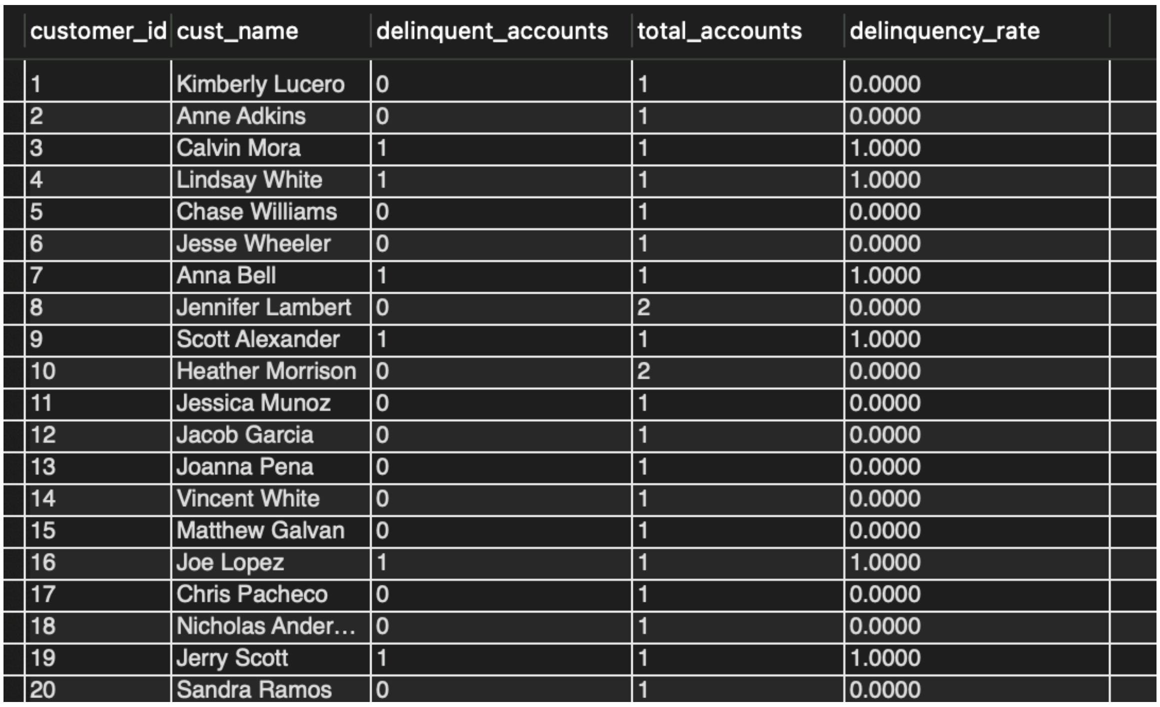


customer_id	unique_locatio...	total_transactio...	location_consiste...
1	0	0	high
2	0	0	high
3	6	6	high
4	2	2	high
5	1	1	high
6	1	1	high
7	1	1	high
8	1	1	high
9	5	5	high
10	1	1	high
11	0	0	high
12	3	3	high
13	1	1	high
14	1	1	high
15	0	0	high
16	2	2	high
17	4	4	high
18	2	2	high
19	1	1	high
20	4	4	high

-- Delinquency Rates from Credit History

```
select
    c.customer_id,
    concat(c.first_name, ' ', c.last_name) as cust_name,
    count(case when a.delinquent = 1 then 1 end) as
delinquent_accounts,
    count(a.account_id) as total_accounts,
    case
        when count(a.account_id) = 0 then 0
        else count(case when a.delinquent = 1 then 1 end) /
count(a.account_id)
    end as delinquency_rate
from
    customers c
left join
    accounts a on c.customer_id = a.customer_id
group by
    c.customer_id
order by
```

```
c.customer_id;
```



customer_id	cust_name	delinquent_accounts	total_accounts	delinquency_rate
1	Kimberly Lucero	0	1	0.0000
2	Anne Adkins	0	1	0.0000
3	Calvin Mora	1	1	1.0000
4	Lindsay White	1	1	1.0000
5	Chase Williams	0	1	0.0000
6	Jesse Wheeler	0	1	0.0000
7	Anna Bell	1	1	1.0000
8	Jennifer Lambert	0	2	0.0000
9	Scott Alexander	1	1	1.0000
10	Heather Morrison	0	2	0.0000
11	Jessica Munoz	0	1	0.0000
12	Jacob Garcia	0	1	0.0000
13	Joanna Pena	0	1	0.0000
14	Vincent White	0	1	0.0000
15	Matthew Galvan	0	1	0.0000
16	Joe Lopez	1	1	1.0000
17	Chris Pacheco	0	1	0.0000
18	Nicholas Ander…	0	1	0.0000
19	Jerry Scott	1	1	1.0000
20	Sandra Ramos	0	1	0.0000