UTM

UNIVERSITI TEKNOLOGI MALAYSIA

SCHOOL OF COMPUTING, FACULTY ENGINEERING

SESSION 2021/2022 SEMESTER 2

GROUP PROJECT: PART 1 (RDBMS)

SCSP5023

BIG DATA MANAGEMENT

NAME	MATRIC NO
ALYA NASUHA BINTI MOHAMMAD NASIRUDDIN	B19EC3001
CHONG XIAN JUN	MCS211047
TAN FEI ZHI	B19EC0041

1) DATA SELECTION

Dataset used in this project is retrieved from the relational website. The dataset describes PKDD'99 Financial that was collected from the issuance bank. There are a total of 8 entities that consist of 55 columns with 1,090,086 rows of data. The analysis is to help the issuance bank, Bank A to generate deeper insights about its clients to optimize the bank services provided.

Data Source: https://relational.fit.cvut.cz/dataset/Financial

2) CREATE SCENARIO FOR THE DATA

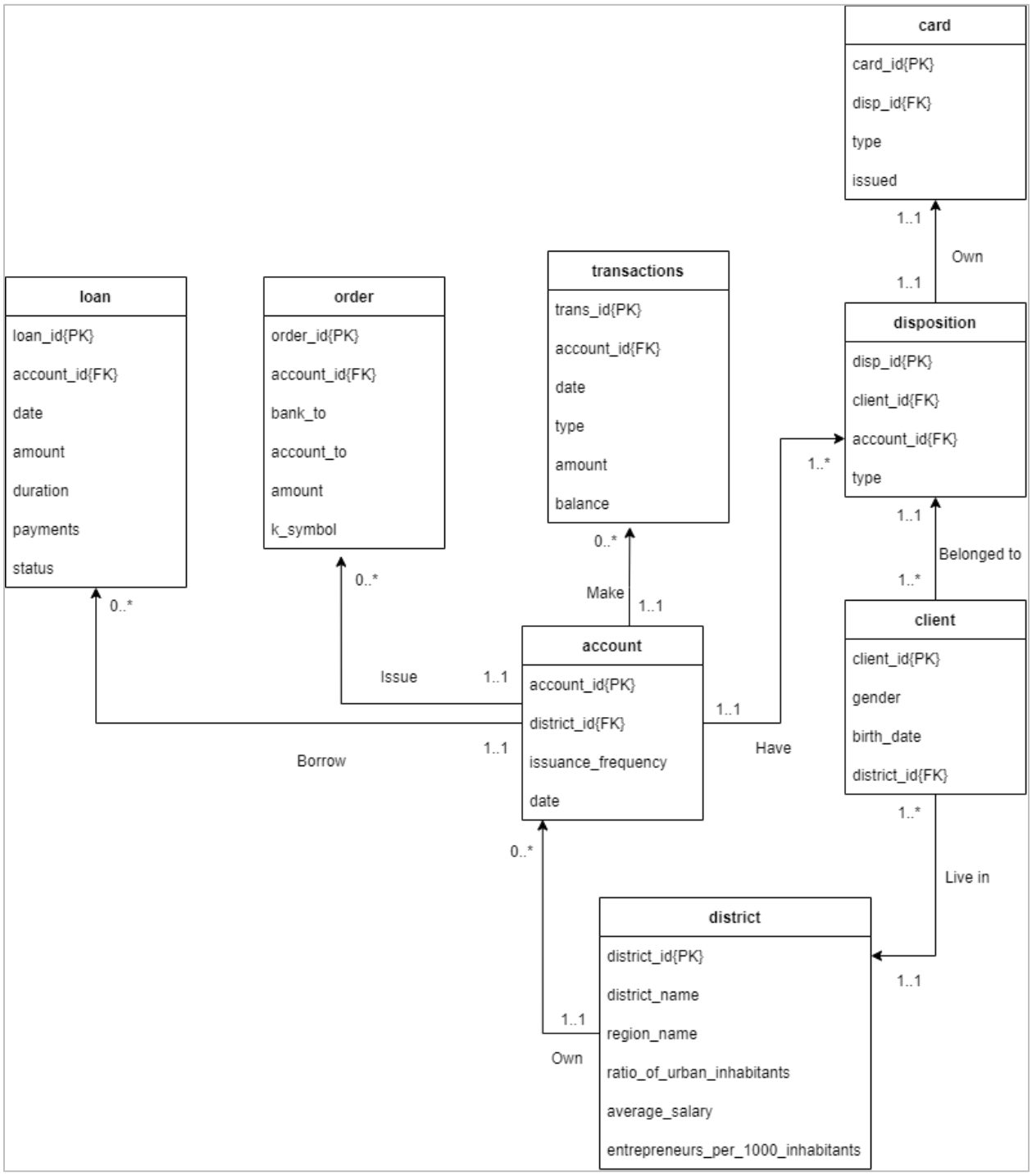
There are a total of 10 scenarios designed to analyze the data. The scenarios are listed in questions 4.

3) DEFINE POSSIBLE RELATIONS

Table Name	Primary Key	Foreign Key
loan	load_id	account_id
order	order_id	account_id
transactions	trans_id	account_id
account	account_id	district_id
card	card_id	disp_id
disposition	disp_id	client_id, account_id
client	client_id	district _id
district	district_id	-

Relation	**Keys**	**Participation in relation with**
loan (loan_id, account_id, date, amount, duration, payments, status)	PK: loan_id FK: account_id	account (optional)
Order (order_id, account_id, bank_to account_to, amount, k_symbol)	PK: order_id FK: account_id	account (optional)
Transactions (trans_id, account_id, date, type, amount, balance)	PK: trans_id FK: account_id	account (optional)
Account (account_id, district_id, issuance_frequency, date)	PK: account_id FK: district_id	Loan (optional) Order (optional) Transactions (optional) District (mandatory) Disposition (mandatory)
District (district_id, district_name region_name, ratio_of_urban_inhabitants, average_salary, entrepreneurs_per_1000_inhabitants)	PK: district_id	Account (optional) Client (mandatory)
Client (client_id, gender, birth_date, district_id)	PK: client_id FK: district_id	District (mandatory) Disposition (mandatory)

Age_mapping (client_id, age_range)	FK: client_id	Client (mandatory)
Disposition (disp_id, account_id, client_id, type)	PK: disp_id FK: account_id, client_id	Account (mandatory) Client (mandatory) Card (optional)
Card (card_id, disp_id, type, issued)	PK: card_id FK: disp_id	Disposition (optional)


card
card_id{PK}
disp_id{FK}
type
issued
1..1
Own
1..1
transactions
trans_id{PK}
account_id{FK}
date
type
amount
balance
disposition
disp_id{PK}
client_id{FK}
account_id{FK}
type
1..*
1..1
Belonged to
1..*
loan
loan_id{PK}
account_id{FK}
date
amount
duration
payments
status
order
order_id{PK}
account_id{FK}
bank_to
account_to
amount
k_symbol
0..*
Make
1..1
0..*
0..*
Issue
1..1
client
client_id{PK}
gender
birth_date
district_id{FK}
account
account_id{PK}
district_id{FK}
issuance_frequency
date
1..1
Have
1..1
Borrow
1..1
1..*
Live in
0..*
district
district_id{PK}
district_name
region_name
ratio_of_urban_inhabitants
average_salary
entrepreneurs_per_1000_inhabitants
1..1
1..1
Own

4) SQL queries

Dataset from PKDD '99 disclosing information of a bank, Bank A. Analysis scenario and its respective SQL codes are provided below

Basic overview for Bank A

A basic overview is conducted to investigate the basic service and client information for Bank A. The dataset consists of part of the public facing services provided by Bank A, i.e. registration of accounts which can be used to order financial transactions, borrow loans and use credit card services. An analysis is conducted to study the client profiles for Bank A for Bank A to understand its clients better and make more informed decisions in business.

Knowing the Clients of Bank A

A general picture of the clients of Bank A is extracted and portrayed. It covers very basic information: Are the clients male or female? How old are they? Where are they from?

1. <u>The distribution of male and female clients.</u>

```
SELECT gender,
       COUNT(gender) AS number_of_clients,
       100 * COUNT(gender) /SUM(COUNT(*)) OVER () AS client_percentage
FROM client
GROUP BY gender;
```

OUTPUT:

gender	number_of_clients	client_percentage
F	2645	49.2643
M	2724	50.7357

2. The distribution of clients by age.

```
CREATE TABLE age_group AS
SELECT
CASE
       WHEN age_1999 <= 20 THEN 'age <=20'
       WHEN age_1999 >20 and age_1999 <= 25 THEN '20 < age <= 25'
       WHEN age_1999 >25 and age_1999 <= 30 THEN '25 < age <= 30'
       WHEN age_1999 >30 and age_1999 <= 35 THEN '30 < age <= 35'
       WHEN age_1999 >35 and age_1999 <= 40 THEN '35 < age <= 40'
       WHEN age_1999 >40 and age_1999 <= 45 THEN '40 < age <= 45'
       WHEN age_1999 >45 and age_1999 <= 50 THEN '45 < age <= 50'
       WHEN age_1999 >50 and age_1999 <= 55 THEN '50 < age <= 55'
       WHEN age_1999 >55 and age_1999 <= 60 THEN '55 < age <= 60'
    ELSE 'age > 60'
END AS age_range,
COUNT(*) AS number_of_clients
FROM (SELECT timestampdiff(Year, birth_date,'1999-01-01') AS age_1999
FROM client) AS age
GROUP BY age_range
ORDER BY age_range;

SELECT * FROM age_group;
```

OUTPUT:

age_range	number_of_clients	client_percentage
age <=20	390	7.2639
20 < age <= 25	497	9.2568
25 < age <= 30	495	9.2196
30 < age <= 35	482	8.9775
35 < age <= 40	485	9.0333

40 < age <= 45	472	8.7912
45 < age <= 50	490	9.1265
50 < age <= 55	488	9.0892
55 < age <= 60	491	9.1451
age > 60	1079	20.0969

3. The distribution of clients by district in descending order.

```
SELECT district.district_id, district_name, region_name,
COUNT(account_id) AS number_of_accounts
FROM district
INNER JOIN account
       ON district.district_id = account.district_id
GROUP BY account.district_id
ORDER BY number_of_accounts DESC;
```

OUTPUT:

district_id	district_name	region_name	number_of_accounts	accounts_percentage
1	Hl.m. Praha	Prague	554	12.4438
70	Karvina	north Moravia	152	3.4142
74	Ostrava - mesto	north Moravia	135	3.0323
54	Brno - mesto	south Moravia	128	2.8751
64	Zlin	south Moravia	92	2.0665
72	Olomouc	north Moravia	88	1.9766
68	Frydek - Mistek	north Moravia	83	1.8643
5	Kolin	central Bohemia	65	1.46
46	Nachod	east Bohemia	59	1.3252
52	Usti nad Orlici	east Bohemia	57	1.2803
36	Liberec	north Bohemia	57	1.2803
…	…	…	…	…

[refer to table 3-number_of_acc_by_district.csv]

Demographic profiles of loan clients

4. Profiling loan borrowers by age and gender.
    a. Both the sum and average male and female borrowings in each age range and order by mean_loan_amount from the highest to lowest.

```
CREATE TABLE age_mapping AS
SELECT client_id, timestampdiff(Year, birth_date, '1999-01-01') AS age_1999,
CASE
       WHEN timestampdiff(Year, birth_date, '1999-01-01') <= 20 THEN 'age <=20'
       WHEN timestampdiff(Year, birth_date, '1999-01-01') >20 and
timestampdiff(Year, birth_date, '1999-01-01') <= 25 THEN '20 < age <= 25'
       WHEN timestampdiff(Year, birth_date, '1999-01-01') and
timestampdiff(Year, birth_date, '1999-01-01') <= 30 THEN '25 < age <= 30'
       WHEN timestampdiff(Year, birth_date, '1999-01-01') and
timestampdiff(Year, birth_date, '1999-01-01') <= 35 THEN '30 < age <= 35'
       WHEN timestampdiff(Year, birth_date, '1999-01-01') >35 and
timestampdiff(Year, birth_date, '1999-01-01') <= 40 THEN '35 < age <= 40'
       WHEN timestampdiff(Year, birth_date, '1999-01-01') >40 and
timestampdiff(Year, birth_date, '1999-01-01') <= 45 THEN '40 < age <= 45'
       WHEN timestampdiff(Year, birth_date, '1999-01-01') >45 and
timestampdiff(Year, birth_date, '1999-01-01') <= 50 THEN '45 < age <= 50'
       WHEN timestampdiff(Year, birth_date, '1999-01-01') >50 and
timestampdiff(Year, birth_date, '1999-01-01') <= 55 THEN '50 < age <= 55'
       WHEN timestampdiff(Year, birth_date, '1999-01-01') >55 and
timestampdiff(Year, birth_date, '1999-01-01') <= 60 THEN '55 < age <= 60'
    ELSE 'age > 60'
END AS age_range
FROM client;

SELECT * FROM age_mapping;
SELECT age_mapping.age_range, gender, SUM(amount) AS total_loan_amount,
AVG(amount) AS mean_loan_amount
FROM loan
JOIN disposition ON loan.account_id = disposition.account_id
JOIN client ON disposition.client_id = client.client_id
```

```
JOIN age_mapping ON age_mapping.client_id = client.client_id
GROUP BY gender, age_mapping.age_range
ORDER BY mean_loan_amount DESC;
```

OUTPUT:

age_range	gender	total_loan_amount	mean_loan_amount	number_of_loans
age <=20	M	6,126,528.00	185,652.36	33
age > 60	F	2,325,828.00	178,909.85	13
age <=20	F	7,389,192.00	175,933.14	42
30 < age <= 35	M	9,430,044.00	174,630.44	54
20 < age <= 25	M	6,732,108.00	168,302.70	40
35 < age <= 40	F	8,125,560.00	165,827.76	49
45 < age <= 50	M	10,111,596.00	165,763.87	61
45 < age <= 50	F	10,616,772.00	163,334.95	65
age > 60	M	6,506,856.00	162,671.40	40
30 < age <= 35	F	12,990,216.00	162,377.70	80
20 < age <= 25	F	9,723,576.00	159,402.89	61
50 < age <= 55	F	10,814,280.00	156,728.70	69
25 < age <= 30	F	7,122,384.00	148,383.00	48
40 < age <= 45	M	8,991,900.00	147,408.20	61
25 < age <= 30	M	9,802,392.00	146,304.36	67
50 < age <= 55	M	8,559,696.00	142,661.60	60
55 < age <= 60	F	10,071,672.00	139,884.33	72
35 < age <= 40	M	9,394,452.00	138,153.71	68
40 < age <= 45	F	8,709,864.00	133,997.91	65
55 < age <= 60	M	6,593,112.00	106,340.52	62

b. Gender differences: what are the statistics on borrowings for males and females?

```
SELECT gender, AVG(timestampdiff(Year, birth_date,'1999-01-01')) AS mean_age,
COUNT(*) AS number_of_loans, SUM(amount) AS total_loan_amount, AVG(amount) AS
mean_loan_amount
FROM loan
JOIN disposition ON loan.account_id = disposition.account_id
JOIN client ON disposition.client_id = client.client_id
GROUP BY gender;
```

OUTPUT:

gender	mean_age	number_of_loans	total_loan_amount	mean_loan_amount
M	41	546	82,248,684.00	150,638.62
F	40	564	87,889,344.00	155,832.17

c. Which are the groups of people who borrow the most?

```
SELECT age_mapping.age_range, gender, SUM(amount) AS total_loan_amount,
AVG(amount) AS mean_loan_amount, COUNT(*) AS number_of_loans
FROM loan
JOIN disposition ON loan.account_id = disposition.account_id
JOIN client ON disposition.client_id = client.client_id
JOIN age_mapping ON age_mapping.client_id = client.client_id
GROUP BY age_mapping.age_range, gender
ORDER BY mean_loan_amount DESC
LIMIT 3;
```

OUTPUT:

age_range	gender	total_loan_amount	mean_loan_amount	number_of_loans
age <=20	M	6,126,528.00	185,652.36	33
age > 60	F	2,325,828.00	178,909.85	13
age <=20	F	7,389,192.00	175,933.14	42

d. By mean loan amount of the group

```
SELECT age_mapping.age_range, gender, SUM(amount) AS total_loan_amount,
AVG(amount) AS mean_loan_amount, COUNT(*) AS number_of_loans
FROM loan
JOIN disposition ON loan.account_id = disposition.account_id
JOIN client ON disposition.client_id = client.client_id
JOIN age_mapping ON age_mapping.client_id = client.client_id
GROUP BY age_mapping.age_range, gender
ORDER BY total_loan_amount DESC
LIMIT 3;
```

OUTPUT:

age_range	gender	total_loan_amount	mean_loan_amount	number_of_loans
30 < age <= 35	F	12,990,216.00	162,377.70	80
50 < age <= 55	F	10,814,280.00	156,728.70	69
45 < age <= 50	F	10,616,772.00	163,334.95	65

e. Which are the groups that have the most status ‘A’ loans

```
SELECT age_mapping.age_range, gender, loan.status, SUM(amount) AS
total_loan_amount, AVG(amount) AS mean_loan_amount, COUNT(*) AS
number_of_loans
FROM loan
JOIN disposition ON loan.account_id = disposition.account_id
JOIN client ON disposition.client_id = client.client_id
JOIN age_mapping ON age_mapping.client_id = client.client_id
GROUP BY gender, age_mapping.age_range
HAVING loan.status = 'A'
ORDER BY total_loan_amount DESC;
```

OUTPUT:

age_range	gender	status	total_loan_amount	mean_loan_amount	number_of_loans
55 < age <= 60	F	A	10,071,672.00	139,884.33	72
50 < age <= 55	F	A	10,814,280.00	156,728.70	69
35 < age <= 40	M	A	9,394,452.00	138,153.71	68
55 < age <= 60	M	A	6,593,112.00	106,340.52	62
50 < age <= 55	M	A	8,559,696.00	142,661.60	60
age > 60	F	A	2,325,828.00	178,909.85	13

5. The amount of loans for different districts
    a. The amount of loans for the 5 districts with the highest total amount of loan

```
SELECT district.district_id, district_name, region_name, SUM(amount) AS
total_loan_amount, AVG(amount) AS mean_loan_amount, COUNT(*) AS
number_of_loans
FROM loan
JOIN account ON loan.account_id = account.account_id
JOIN district ON account.district_id = district.district_id
GROUP BY district_id
ORDER BY total_loan_amount DESC
LIMIT 5;
```

OUTPUT:

district_id	district_name	region_name	total_loan_amount	mean_loan_amount	number_of_loans
46	Nachod	east Bohemia	1,768,380.00	294,730.00	6
14	Ceske Budejovice	south Bohemia	2,010,924.00	251,365.50	8
3	Beroun	central Bohemia	1,460,796.00	243,466.00	6
6	Kutna Hora	central Bohemia	2,095,980.00	232,886.67	9
67	Bruntal	north Moravia	1,277,796.00	212,966.00	6

b. The amount of loans for the 5 districts with the lowest total amount of loan

```
SELECT district.district_id, district_name, region_name, SUM(amount) AS
total_loan_amount, AVG(amount) AS mean_loan_amount, COUNT(*) AS
number_of_loans
FROM loan
JOIN account ON loan.account_id = account.account_id
JOIN district ON account.district_id = district.district_id
GROUP BY district_id
ORDER BY total_loan_amount
LIMIT 5;
```

OUTPUT:

district_id	district_name	region_name	total_loan_amount	mean_loan_amount	number_of_loans
30	Sokolov	west Bohemia	148,524.00	74,262.00	2
32	Ceska Lipa	north Bohemia	462,684.00	77,114.00	6
77	Vsetin	north Moravia	490,980.00	81,830.00	6
15	Cesky Krumlov	south Bohemia	584,328.00	83,475.43	7
25	Klatovy	west Bohemia	270,876.00	90,292.00	3

6. Is the difference between districts big? If so, what are the factors that contribute to it?

    a. Do people from places with higher or lower salaries tend to borrow more?

```
SELECT CASE
       WHEN average_salary >8000 and average_salary <9000 THEN '8000 < salary
<9000'
    WHEN average_salary >=9000 and average_salary <10000 THEN '9000 <= salary
<9000'
    WHEN average_salary >=10000 and average_salary <11000 THEN '10000 <=
salary <11000'
    WHEN average_salary >=11000 and average_salary <12000 THEN '11000 <=
salary <12000'
    WHEN average_salary >=12000 THEN 'salary >= 12000'
END AS district_salary_range,
AVG(average_salary) AS mean_salary,
AVG(amount) AS mean_loan_amount
FROM district
JOIN account ON account.district_id = district.district_id
JOIN loan ON loan.account_id = account.account_id
GROUP BY district_salary_range;
```

OUTPUT:

district_salary_range	mean_salary	mean_loan_amount
salary >= 12000	12,541.00	153,957.29
9000 <= salary <9000	9,549.89	140,600.75
8000 < salary <9000	8,589.47	153,479.66
10000 <= salary <11000	10,377.06	155,327.82
11000 <= salary <12000	11,277.00	123,504.00

Credit Cards

7. What is the client_id, account_id and credit transactions for clients who own the gold-type card?

```
SELECT DISTINCT
disposition.disp_id,disposition.client_id,disposition.account_id,card.type AS
card_type, SUM(transactions.amount) AS transaction_amount
FROM disposition
INNER JOIN card ON disposition.disp_id=card.disp_id
INNER JOIN transactions ON disposition.account_id=transactions.account_id
WHERE card.type="gold" AND transactions.type="credit"
GROUP BY transactions.account_id;
```

OUTPUT:

disp_id	client_id	account_id	card_type	transaction_amount
9	9	7	gold	623,775
41	41	33	gold	1,652,341
79	79	68	gold	1,857,968
326	326	270	gold	1,215,611
548	548	456	gold	3,683,681
562	562	468	gold	821,280

8. Detect potential gold-card holders.

```
SELECT AVG(transactions.amount) AS AVG,card.type,transactions.account_id
FROM transactions
INNER JOIN disposition ON disposition.account_id=transactions.account_id
INNER JOIN card ON disposition.disp_id=card.disp_id
WHERE transactions.type="credit" AND card.type="gold";

SELECT transactions.account_id,card.type,SUM(transactions.amount) AS
transaction_amount
FROM transactions
JOIN disposition ON disposition.account_id=transactions.account_id
JOIN card ON disposition.disp_id=card.disp_id
WHERE transactions.type="credit"
GROUP BY transactions.account_id
HAVING transaction_amount > 32958.7157;
```

OUTPUT:

account_id	type	transaction_amount
7	gold	623,775
14	classic	408,445
33	gold	1,652,341
34	classic	2,181,988
43	junior	1,436,880
48	classic	628,885
51	junior	702,325
65	classic	701,415
66	classic	2,973,820
68	gold	1,857,968

71	junior	1,369,576
73	classic	1,301,198
95	classic	1,172,919
96	classic	3,397,102
97	classic	635,480
105	classic	252,833
106	classic	950,932
108	classic	784,651
109	classic	119,887
…	…	…

[refer to table 8 - potential gold card holders.csv]

9. Client profiling for clients who use credit cards.

    a. Age

```
CREATE TABLE age_mapping AS
SELECT client_id, timestampdiff(Year, birth_date, '1999-01-01') AS age_1999,
CASE
       WHEN timestampdiff(Year, birth_date, '1999-01-01') <= 20 THEN 'age <=20'
       WHEN timestampdiff(Year, birth_date, '1999-01-01') >20 and
timestampdiff(Year, birth_date, '1999-01-01') <= 25 THEN '20 < age <= 25'
       WHEN timestampdiff(Year, birth_date, '1999-01-01') and
timestampdiff(Year, birth_date, '1999-01-01') <= 30 THEN '25 < age <= 30'
       WHEN timestampdiff(Year, birth_date, '1999-01-01') and
timestampdiff(Year, birth_date, '1999-01-01') <= 35 THEN '30 < age <= 35'
       WHEN timestampdiff(Year, birth_date, '1999-01-01') >35 and
timestampdiff(Year, birth_date, '1999-01-01') <= 40 THEN '35 < age <= 40'
       WHEN timestampdiff(Year, birth_date, '1999-01-01') >40 and
timestampdiff(Year, birth_date, '1999-01-01') <= 45 THEN '40 < age <= 45'
       WHEN timestampdiff(Year, birth_date, '1999-01-01') >45 and
timestampdiff(Year, birth_date, '1999-01-01') <= 50 THEN '45 < age <= 50'
       WHEN timestampdiff(Year, birth_date, '1999-01-01') >50 and
timestampdiff(Year, birth_date, '1999-01-01') <= 55 THEN '50 < age <= 55'
       WHEN timestampdiff(Year, birth_date, '1999-01-01') >55 and
timestampdiff(Year, birth_date, '1999-01-01') <= 60 THEN '55 < age <= 60'
    ELSE 'age > 60'
END AS age_range
FROM client;

SELECT * FROM age_mapping;

SELECT age_mapping.age_range AS age_range,
             card.type AS card_type,
             SUM(transactions.amount) AS transaction_amount
FROM disposition
       JOIN client ON disposition.client_id = client.client_id
    JOIN age_mapping ON age_mapping.client_id = client.client_id
```

```
        INNER JOIN card ON disposition.disp_id=card.disp_id
        INNER JOIN transactions ON
disposition.account_id=transactions.account_id
GROUP BY card.type, age_mapping.age_range;
```

OUTPUT:

age_range	card_type	transaction_amount
age > 60	gold	8,505,633
55 < age <= 60	classic	18,375,004
25 < age <= 30	gold	6,844,328
age > 60	classic	12,446,499
age <=20	junior	24,112,896
35 < age <= 40	classic	21,384,000
30 < age <= 35	classic	18,082,755
40 < age <= 45	classic	14,778,808
50 < age <= 55	classic	24,630,880
25 < age <= 30	classic	32,302,317
45 < age <= 50	classic	23,082,844
20 < age <= 25	classic	13,212,835
20 < age <= 25	junior	8,688,636
30 < age <= 35	gold	2,354,445
20 < age <= 25	gold	1,565,232

b. Gender

```
SELECT gender,
            card.type AS card_type,
            SUM(transactions.amount) AS transaction_amount
FROM disposition
      JOIN client ON disposition.client_id = client.client_id
      INNER JOIN card ON disposition.disp_id=card.disp_id
      INNER JOIN transactions ON
disposition.account_id=transactions.account_id
GROUP BY card.type, gender;
```

OUTPUT:

gender	card_type	transaction_amount
M	gold	11,717,787
M	classic	84,970,338
F	junior	12,190,803
M	junior	20,610,729
F	classic	93,325,604
F	gold	7,551,851

c. District

```
SELECT district.district_id,
            card.type AS card_type,
            SUM(transactions.amount) AS transaction_amount
FROM  district
      JOIN account ON account.district_id = district.district_id
      JOIN transactions ON transactions.account_id=account.account_id
    JOIN disposition ON disposition.account_id=transactions.account_id
      JOIN card ON disposition.disp_id=card.disp_id
GROUP BY district_id,card.type;
```

OUTPUT:

district_id	card_type	transaction_amount
60	gold	1,185,555
47	classic	7,740,701
22	gold	3,212,154
67	classic	6,071,113
36	junior	4,625,434
21	classic	1,683,150
67	junior	1,314,395
36	classic	2,775,739
48	classic	5,863,352
37	gold	3,632,174
1	junior	4,291,649
72	classic	3,233,843
1	classic	13,210,765
68	classic	8,984,733
74	classic	3,447,974
63	classic	1,845,867
53	classic	4,504,017
31	classic	1,238,672
2	junior	802,051
…	…	…

[refer to table 9c - client who use creditcard by district.csv]

Account

10. Detect inactive accounts by querying account_id where last transfer is more than 1 year ago.

```
CREATE INDEX trans_date ON transactions(date);

SELECT client.client_id, disposition.account_id, MAX(transactions.date) AS
last_transaction_date
FROM transactions
JOIN disposition ON disposition.account_id = transactions.account_id
JOIN client ON client.client_id = disposition.client_id
GROUP BY disposition.account_id
HAVING datediff('1999-01-01', last_transaction_date) > 365;
```

OUTPUT:

client_id	account_id	last_transaction_date
551	459	1997-03-01

REFERENCES

- https://relational.fit.cvut.cz/dataset/Financial