</talentlabs>

Credit Risk DA Project

Database Connection

Download the DBeaver SQL client to connect to the MySQL database:

- https://dbeaver.io/

Follow the documentation to set up a connection to the database:

- https://dbeaver.com/docs/wiki/Create-Connection/

The database is hosted on AWS, here are the connection details:

- Endpoint: home-credit-default-risk.c7rizeij2t53.ap-southeast-1.rds.amazonaws.com
- Port: 3306
- Database: credit
- Login User: student
- Login Password: student

Overview

Consider you are asked to review a list of loan applications. The given “credit” database contains data on the loan applicant and their historical loan behavior. There are many columns in the database, you **don’t need to use all the columns**, We will provide a list of useful column descriptions for you.

Cautions

Missing Values:

There are columns with missing values. You need to handle them during your analysis. There are multiple ways we can handle missing values: 4 Ways to Replace NULL with a Different Value in MySQL

Discretization:

Discretization means we want to convert numbers into bins, for example, age to age groups or income to income groups. There are mainly 2 reasons for this:

- It is easier to see patterns with a group of values. For example, it is better to say people older than 20 are richer than people younger than 20, instead of saying people aged 20 are richer than people aged 21.
- We want to avoid biased statistics. If we apply group by aggregation directly on a number column like age, the average statistics can be biased. For example, if there is only 1 person aged 59, then the average income of people aged 59 only represents that 1 person in the dataset.

We can do it with the CASE Function in MySQL:
MySQL CASE Function
During the analysis, you can consider converting some factors into groups.

Task 1 Run SQL via DBeaver

Follow the documentation to open the “SQL Editor”:

- https://dbeaver.com/docs/wiki/SQL-Editor/

Run SQL to examine the number of rows in each table:

Table	**Count**
application	307,511
bureau	1,716,428

Loan Applications

The “application” table stores the loan applications. This includes:

- The demographic of the loan applicants
- The loan size or purposes
- The applicant’s credit score
- Is the loan applicant has a payment difficulties with the loan.

SK_ID_CURR	ID of the loan in our sample
TARGET	Target variable, this is the **future information**. Will this loan applicant has payment difficulties? (1: client with payment difficulties: he/she had late payment more than X days, 0: no payment difficulties)
CODE_GENDER	Gender of the client
FLAG_OWN_CAR	Flag if the client owns a car
FLAG_OWN_REALTY	Flag if the client owns a house or flat
CNT_CHILDREN	Number of children the client has
AMT_INCOME_TOTAL	Income of the client
AMT_CREDIT	Credit amount of the loan
AMT_ANNUITY	Loan annuity

AMT_GOODS_PRICE	For consumer loans it is the price of the goods for which the loan is given
NAME_TYPE_SUITE	Who was accompanying client when he was applying for the loan
NAME_INCOME_TYPE	Clients income type (businessman, working, maternity leave,…)
NAME_EDUCATION_TYPE	Level of highest education the client achieved
NAME_FAMILY_STATUS	Family status of the client
NAME_HOUSING_TYPE	What is the housing situation of the client (renting, living with parents, ...)
DAYS_BIRTH	Client's age in days at the time of application
DAYS_EMPLOYED	How many days before the application the person started current employment
OCCUPATION_TYPE	What kind of occupation does the client have
EXT_SOURCE_1	Normalized credit score from an external data source
EXT_SOURCE_2	Normalized credit score from an external data source
EXT_SOURCE_3	Normalized credit score from an external data source

Task 2 What is a Credit Score

In the “application” table above there are 3 credit score columns. Research online to see what is a credit score and why we need it. (Note that the scores in the database are normalized, which means they are scaled to the 0 to 1 range)

A credit score is a numerical measure of a person's creditworthiness, used by lenders to assess the risk of lending money. It helps determine whether someone will be approved for a loan and what interest rate they will receive. A higher score indicates a lower risk of default. In the dataset, the EXT_SOURCE_1, EXT_SOURCE_2, and EXT_SOURCE_3 columns represent normalized credit scores, scaled between 0 and 1, which allows for easier comparison and analysis. Normalization standardizes scores across different models and sources, ensuring consistency in assessing an applicant's financial risk.

Task 3 Understand Credit Amount and Annuity

What are Credit Amount and Annuity? Fill in your answer below:

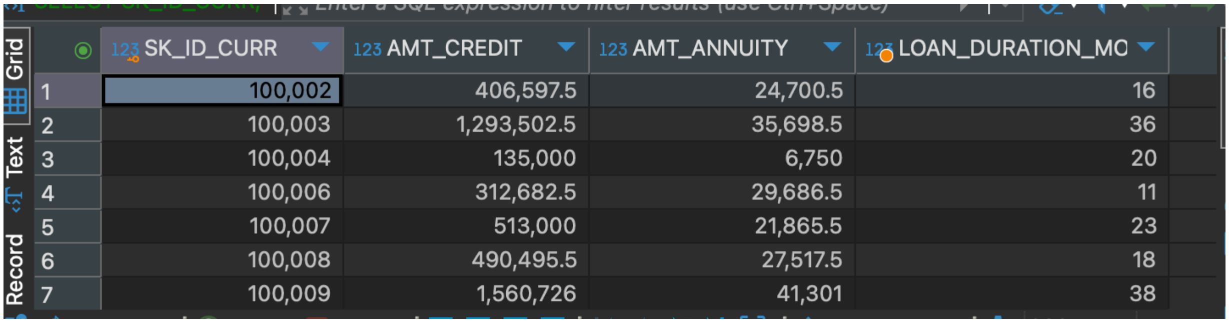
Credit Amount	The total amount of money that a borrower requests from a lender, which will be provided as a loan. It is the principal amount the borrower is obligated to repay, typically with interest, over a specified period.
Annuity	The fixed monthly payment that the borrower needs to make to repay the loan. It includes both principal and interest, ensuring the loan is fully paid off by the end of the term. The annuity amount is typically calculated based on the loan amount, interest rate, and loan term.

Task 4 Deduce the Loan Duration

Given the information from Task 4, we should be able to deduce the Loan Duration for each application. Loan duration describes how many periods (months) the applicant will need to pay back their loans.

Paste the SQL and part of the results below:

```
SELECT
   SK_ID_CURR,
   AMT_CREDIT,
   AMT_ANNUITY,
   ROUND(AMT_CREDIT / AMT_ANNUITY) AS LOAN_DURATION_MONTHS
FROM application;
```



	SK_ID_CURR	AMT_CREDIT	AMT_ANNUITY	LOAN_DURATION_MO
1	100,002	406,597.5	24,700.5	16
2	100,003	1,293,502.5	35,698.5	36
3	100,004	135,000	6,750	20
4	100,006	312,682.5	29,686.5	11
5	100,007	513,000	21,865.5	23
6	100,008	490,495.5	27,517.5	18
7	100,009	1,560,726	41,301	38

Task 5 Are there any factors in the application table affecting the Credit Scores?

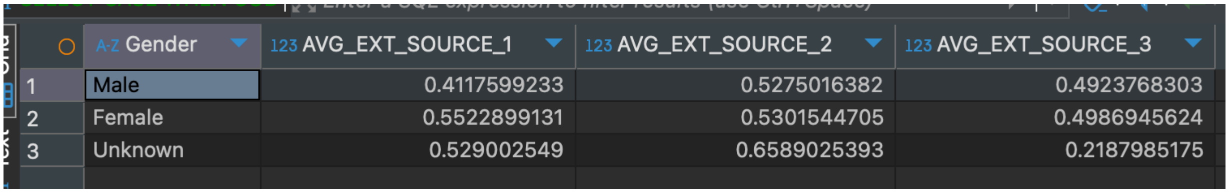
In the “application” table try to explore if there are any columns affecting the credit score. For example, is gender a factor?

Do the analysis of at least 3 factors for 3 different credit scores, it is expected to see different results for different credit scores, for example, a factor might affect EXT_SOURCE_1 but not EXT_SOURCE_3.

Please explain your findings with SQL statements and results:

1. Gender and Credit Scores

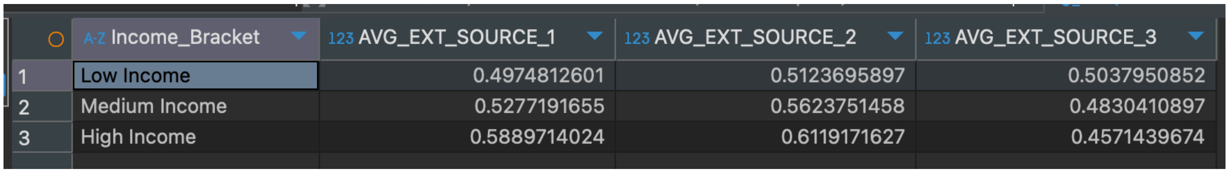
```
SELECT
   CASE
      WHEN CODE_GENDER = 'M' THEN 'Male'
      WHEN CODE_GENDER = 'F' THEN 'Female'
      ELSE 'Unknown'
   END AS Gender,
   AVG(EXT_SOURCE_1) AS AVG_EXT_SOURCE_1,
   AVG(EXT_SOURCE_2) AS AVG_EXT_SOURCE_2,
   AVG(EXT_SOURCE_3) AS AVG_EXT_SOURCE_3
FROM application
WHERE EXT_SOURCE_1 IS NOT NULL AND EXT_SOURCE_2 IS NOT NULL AND EXT_SOURCE_3
IS NOT NULL
GROUP BY Gender;
```



	Gender	AVG_EXT_SOURCE_1	AVG_EXT_SOURCE_2	AVG_EXT_SOURCE_3
1	Male	0.4117599233	0.5275016382	0.4923768303
2	Female	0.5522899131	0.5301544705	0.4986945624
3	Unknown	0.529002549	0.6589025393	0.2187985175

2. Income and Credit Scores

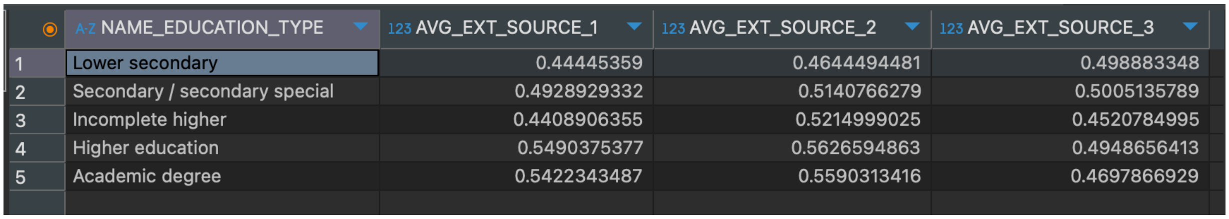
```
SELECT
   CASE
      WHEN AMT_INCOME_TOTAL < 200000 THEN 'Low Income'
      WHEN AMT_INCOME_TOTAL BETWEEN 200000 AND 500000 THEN 'Medium Income'
      ELSE 'High Income'
   END AS Income_Bracket,
   AVG(IFNULL(EXT_SOURCE_1, 0)) AS AVG_EXT_SOURCE_1,
   AVG(IFNULL(EXT_SOURCE_2, 0)) AS AVG_EXT_SOURCE_2,
   AVG(IFNULL(EXT_SOURCE_3, 0)) AS AVG_EXT_SOURCE_3
FROM application
WHERE EXT_SOURCE_1 IS NOT NULL AND EXT_SOURCE_2 IS NOT NULL AND EXT_SOURCE_3
IS NOT NULL
GROUP BY Income_Bracket
ORDER BY
   CASE
      WHEN Income_Bracket = 'Low Income' THEN 1
      WHEN Income_Bracket = 'Medium Income' THEN 2
      WHEN Income_Bracket = 'High Income' THEN 3 -- To ensure result is sorted by income bracket
   END;
```



	Income_Bracket	AVG_EXT_SOURCE_1	AVG_EXT_SOURCE_2	AVG_EXT_SOURCE_3
1	Low Income	0.4974812601	0.5123695897	0.5037950852
2	Medium Income	0.5277191655	0.5623751458	0.4830410897
3	High Income	0.5889714024	0.6119171627	0.4571439674

3. Education Levels and Credit Scores

```
SELECT
   NAME_EDUCATION_TYPE,
   AVG(IFNULL(EXT_SOURCE_1, 0)) AS AVG_EXT_SOURCE_1,
   AVG(IFNULL(EXT_SOURCE_2, 0)) AS AVG_EXT_SOURCE_2,
   AVG(IFNULL(EXT_SOURCE_3, 0)) AS AVG_EXT_SOURCE_3
FROM application
WHERE EXT_SOURCE_1 IS NOT NULL AND EXT_SOURCE_2 IS NOT NULL AND EXT_SOURCE_3
IS NOT NULL
GROUP BY NAME_EDUCATION_TYPE
ORDER BY
         CASE
                  WHEN NAME_EDUCATION_TYPE = 'Lower secondary' THEN 1
                  WHEN NAME_EDUCATION_TYPE = 'Secondary / secondary special' THEN 2
                  WHEN NAME_EDUCATION_TYPE = 'Incomplete higher' THEN 3
                  WHEN NAME_EDUCATION_TYPE = 'Higher education' THEN 4
                  WHEN NAME_EDUCATION_TYPE = 'Academic degree' THEN 5
         END;
```



	A-Z NAME_EDUCATION_TYPE	123 AVG_EXT_SOURCE_1	123 AVG_EXT_SOURCE_2	123 AVG_EXT_SOURCE_3
1	Lower secondary	0.44445359	0.4644494481	0.498883348
2	Secondary / secondary special	0.4928929332	0.5140766279	0.5005135789
3	Incomplete higher	0.4408906355	0.5214999025	0.4520784995
4	Higher education	0.5490375377	0.5626594863	0.4948656413
5	Academic degree	0.5422343487	0.5590313416	0.4697866929

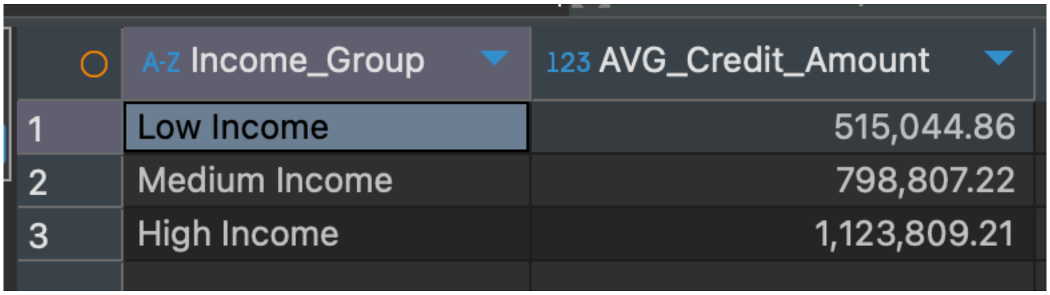
Task 6 Are there any factors in the application table affecting the Credit Amount?

Who is going to lend more money than others? In this task, we want to see are there any factors affecting the credit amount. **Do the analysis of at least 3 factors**

Please explain your findings with SQL statements and results:

1. **Income Group and Credit Amount**

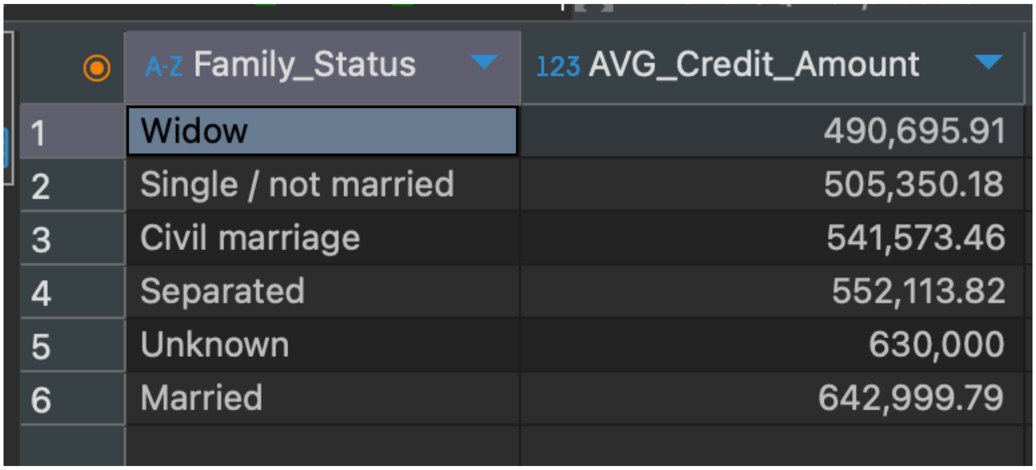
```
SELECT
   CASE
      WHEN AMT_INCOME_TOTAL <= 200000 THEN 'Low Income'
      WHEN AMT_INCOME_TOTAL > 200000 AND AMT_INCOME_TOTAL <= 500000
THEN 'Medium Income'
      ELSE 'High Income'
   END AS Income_Group,
   ROUND(AVG(AMT_CREDIT),2) AS AVG_Credit_Amount
FROM application
WHERE AMT_CREDIT IS NOT NULL
   AND AMT_INCOME_TOTAL IS NOT NULL
GROUP BY Income_Group
ORDER BY
   CASE
      WHEN Income_Group = 'Low Income' THEN 1
      WHEN Income_Group = 'Medium Income' THEN 2
      WHEN Income_Group = 'High Income' THEN 3
   END;
```



	Income_Group	AVG_Credit_Amount
1	Low Income	515,044.86
2	Medium Income	798,807.22
3	High Income	1,123,809.21

2. **Family Status and Credit Amount**

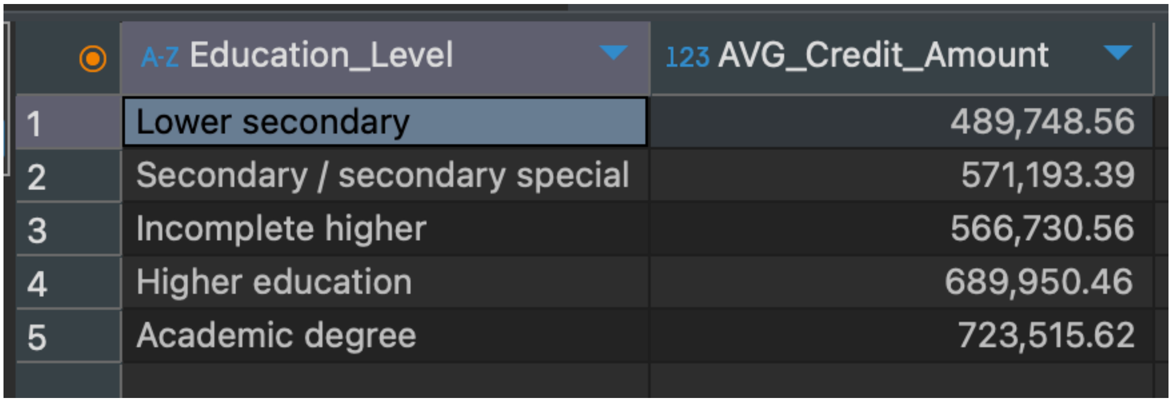
```
SELECT
   NAME_FAMILY_STATUS AS Family_Status,
   ROUND(AVG(AMT_CREDIT),2) AS AVG_Credit_Amount
FROM application
WHERE AMT_CREDIT IS NOT NULL
   AND NAME_FAMILY_STATUS IS NOT NULL
GROUP BY Family_Status
ORDER BY AVG_Credit_Amount ASC;
```



	Family_Status	AVG_Credit_Amount
1	Widow	490,695.91
2	Single / not married	505,350.18
3	Civil marriage	541,573.46
4	Separated	552,113.82
5	Unknown	630,000
6	Married	642,999.79

3. Education Level and Credit Amount

```
SELECT
   NAME_EDUCATION_TYPE AS Education_Level,
   ROUND(AVG(AMT_CREDIT),2) AS AVG_Credit_Amount
FROM application
WHERE AMT_CREDIT IS NOT NULL
   AND NAME_EDUCATION_TYPE IS NOT NULL
GROUP BY Education_Level
ORDER BY
   CASE
      WHEN Education_Level = 'Lower secondary' THEN 1
      WHEN Education_Level = 'Secondary / secondary special' THEN 2
      WHEN Education_Level = 'Incomplete higher' THEN 3
      WHEN Education_Level = 'Higher education' THEN 4
      WHEN Education_Level = 'Academic degree' THEN 5
   END;
```



	Education_Level	AVG_Credit_Amount
1	Lower secondary	489,748.56
2	Secondary / secondary special	571,193.39
3	Incomplete higher	566,730.56
4	Higher education	689,950.46
5	Academic degree	723,515.62

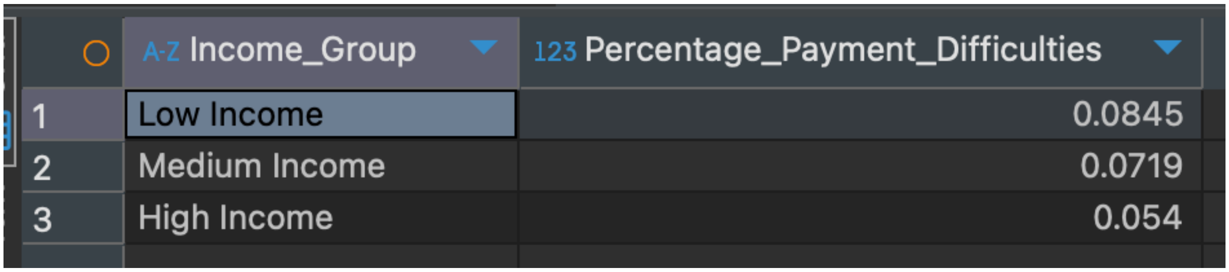
Task 7 Are there any factors in the application table affecting the Payment Difficulties?

In the database, the TARGET column describes will there be a payment difficulty for a loan. We want to see if there are any factors in the application table that can be used to predict this future information. **Do the analysis of at least 3 factors**

Please explain your findings with SQL statements and results:

1. **Income and Payment Difficulties**

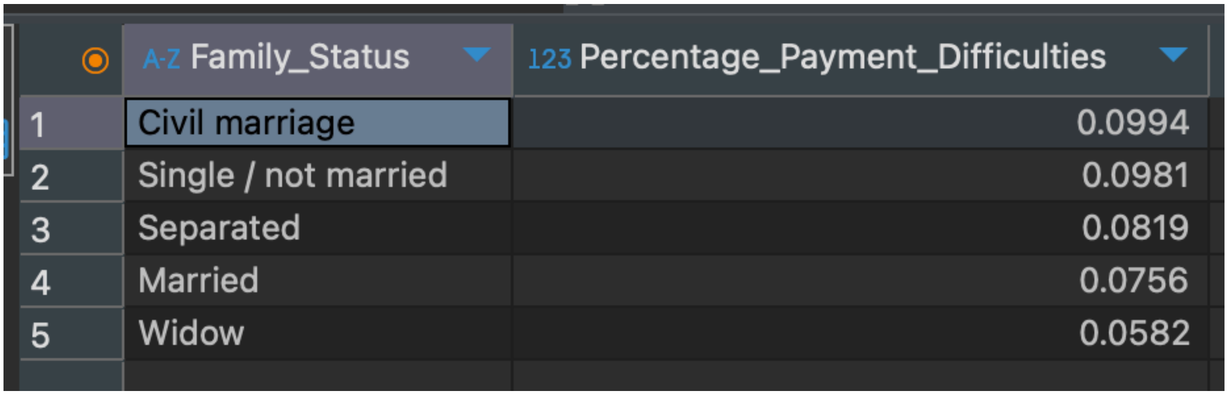
```
SELECT
  CASE
    WHEN AMT_INCOME_TOTAL <= 200000 THEN 'Low Income'
    WHEN AMT_INCOME_TOTAL > 200000 AND AMT_INCOME_TOTAL <= 500000 THEN
'Medium Income'
    ELSE 'High Income'
  END AS Income_Group,
  AVG(TARGET) AS Percentage_Payment_Difficulties
FROM application
WHERE AMT_INCOME_TOTAL IS NOT NULL
GROUP BY Income_Group
ORDER BY
  CASE
    WHEN Income_Group = 'Low Income' THEN 1
    WHEN Income_Group = 'Medium Income' THEN 2
    WHEN Income_Group = 'High Income' THEN 3
  END;
```



	Income_Group	Percentage_Payment_Difficulties
1	Low Income	0.0845
2	Medium Income	0.0719
3	High Income	0.054

2. **Family Status and Payment Difficulties**

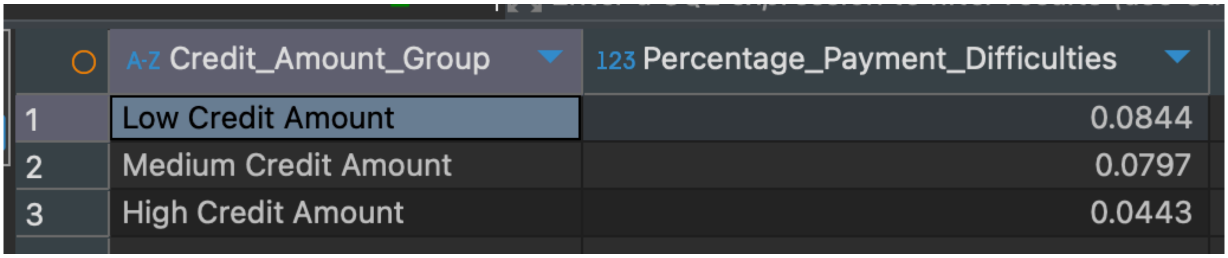
```
SELECT
  NAME_FAMILY_STATUS AS Family_Status,
  AVG(TARGET) AS Percentage_Payment_Difficulties
FROM application
WHERE NAME_FAMILY_STATUS IS NOT NULL AND NAME_FAMILY_STATUS != 'Unknown'
GROUP BY Family_Status
ORDER BY Percentage_Payment_Difficulties DESC;
```



	Family_Status	Percentage_Payment_Difficulties
1	Civil marriage	0.0994
2	Single / not married	0.0981
3	Separated	0.0819
4	Married	0.0756
5	Widow	0.0582

3. Credit Amount and Payment Difficulties

```
SELECT
   CASE
      WHEN AMT_CREDIT <= 500000 THEN 'Low Credit Amount'
      WHEN AMT_CREDIT > 500000 AND AMT_CREDIT <= 1500000 THEN 'Medium Credit Amount'
      ELSE 'High Credit Amount'
   END AS Credit_Amount_Group,
   AVG(TARGET) AS Percentage_Payment_Difficulties
FROM application
WHERE AMT_CREDIT IS NOT NULL
GROUP BY Credit_Amount_Group
ORDER BY
   CASE
      WHEN Credit_Amount_Group = 'Low Credit Amount' THEN 1
      WHEN Credit_Amount_Group = 'Medium Credit Amount' THEN 2
      WHEN Credit_Amount_Group = 'High Credit Amount' THEN 3
   END;
```



	Credit_Amount_Group	Percentage_Payment_Difficulties
1	Low Credit Amount	0.0844
2	Medium Credit Amount	0.0797
3	High Credit Amount	0.0443

Previous/Other Loan Applications

In the previous section, we explored if the demographic data related to payment difficulties, this section we want to see if **historical loan behavior** affecting the payment difficulties.

The “bureau” table stores the other loans of the applicants from the other lenders.

“bureau” table:

SK_ID_CURR	ID of loan in our sample - one loan in our sample can have 0,1,2 or more related previous credits in credit bureau
SK_BUREAU_ID	Recoded ID of previous Credit Bureau credit related to our loan (unique coding for each loan application), The IDs of the “other loans”
CREDIT_DAY_OVERDUE	Number of days past due on CB credit at the time of application for related loan in our sample
AMT_CREDIT_MAX_OVERDUE	Maximal amount overdue on the Credit Bureau credit so far (at application date of loan in our sample)
CNT_CREDIT_PROLONG	How many times was the Credit Bureau credit prolonged
AMT_CREDIT_SUM	Current credit amount for the Credit Bureau credit
AMT_CREDIT_SUM_DEBT	Current debt on Credit Bureau credit
AMT_CREDIT_SUM_LIMIT	Current credit limit of credit card reported in Credit Bureau
AMT_CREDIT_SUM_OVERDUE	Current amount overdue on Credit Bureau credit
CREDIT_TYPE	Type of Credit Bureau credit (Car, cash,...)
DAYS_CREDIT_UPDATE	How many days before loan application did last information about the Credit Bureau credit come
AMT_ANNUITY	Annuity of the Credit Bureau credit

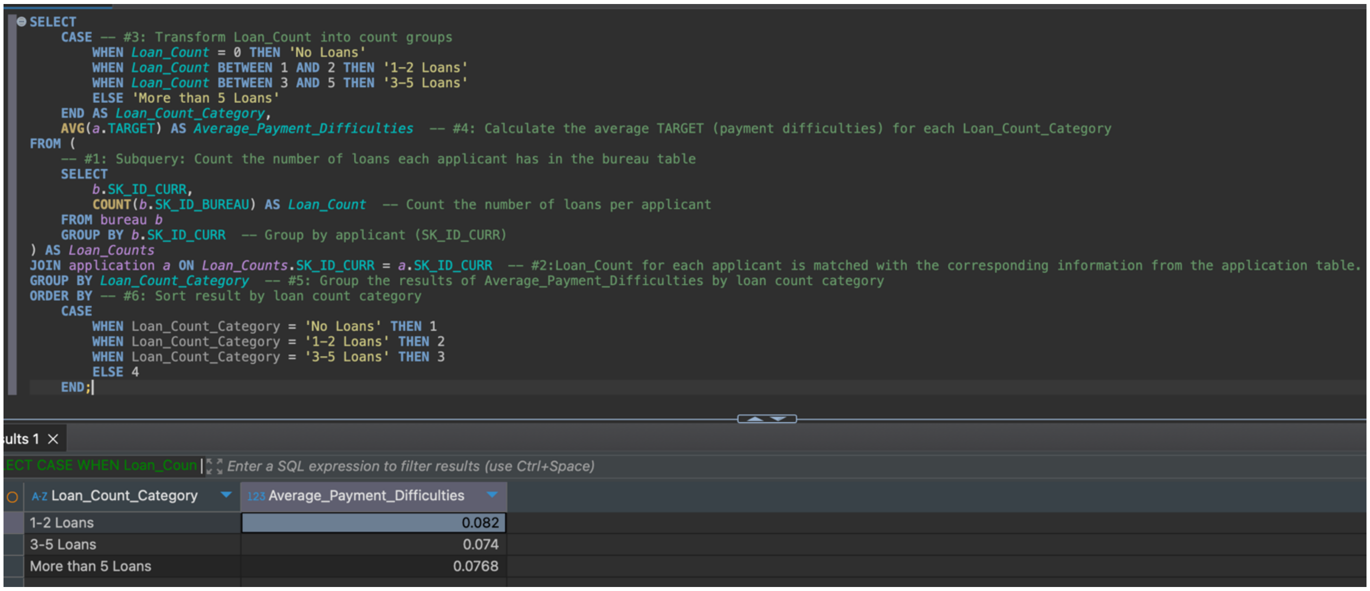
Task 7 Is the number of other loans affecting the payment difficulties?

We want to see if loan applicants have other historical loans affecting their payment abilities. Hints:

- You will need to count the number of loans for each SK_ID_CURR in the “bureau” table.
- Transform the counts into count groups (Discretization).
- Compute the relation between average other loan count to the TARGET

Paste the SQL and part of the results below:

```
SELECT
  CASE
    WHEN Loan_Count = 0 THEN 'No Loans'
    WHEN Loan_Count BETWEEN 1 AND 2 THEN '1-2 Loans'
    WHEN Loan_Count BETWEEN 3 AND 5 THEN '3-5 Loans'
    ELSE 'More than 5 Loans'
  END AS Loan_Count_Category,
  AVG(a.TARGET) AS Average_Payment_Difficulties
FROM (
  SELECT
    b.SK_ID_CURR,
    COUNT(b.SK_ID_BUREAU) AS Loan_Count
  FROM bureau b
  GROUP BY b.SK_ID_CURR
) AS Loan_Counts
JOIN application a ON Loan_Counts.SK_ID_CURR = a.SK_ID_CURR
GROUP BY Loan_Count_Category
ORDER BY
  CASE
    WHEN Loan_Count_Category = 'No Loans' THEN 1
    WHEN Loan_Count_Category = '1-2 Loans' THEN 2
    WHEN Loan_Count_Category = '3-5 Loans' THEN 3
    ELSE 4
  END;
```



```sql
SELECT
    CASE -- #3: Transform Loan_Count into count groups
        WHEN Loan_Count = 0 THEN 'No Loans'
        WHEN Loan_Count BETWEEN 1 AND 2 THEN '1-2 Loans'
        WHEN Loan_Count BETWEEN 3 AND 5 THEN '3-5 Loans'
        ELSE 'More than 5 Loans'
    END AS Loan_Count_Category,
    AVG(a.TARGET) AS Average_Payment_Difficulties  -- #4: Calculate the average TARGET (payment difficulties) for each Loan_Count_Category
FROM (
    -- #1: Subquery: Count the number of loans each applicant has in the bureau table
    SELECT
        b.SK_ID_CURR,
        COUNT(b.SK_ID_BUREAU) AS Loan_Count  -- Count the number of loans per applicant
    FROM bureau b
    GROUP BY b.SK_ID_CURR  -- Group by applicant (SK_ID_CURR)
) AS Loan_Counts
JOIN application a ON Loan_Counts.SK_ID_CURR = a.SK_ID_CURR  -- #2:Loan_Count for each applicant is matched with the corresponding information from the application table.
GROUP BY Loan_Count_Category  -- #5: Group the results of Average_Payment_Difficulties by loan count category
ORDER BY -- #6: Sort result by loan count category
    CASE
        WHEN Loan_Count_Category = 'No Loans' THEN 1
        WHEN Loan_Count_Category = '1-2 Loans' THEN 2
        WHEN Loan_Count_Category = '3-5 Loans' THEN 3
        ELSE 4
    END;
```

Loan_Count_Category	Average_Payment_Difficulties
1-2 Loans	0.082
3-5 Loans	0.074
More than 5 Loans	0.0768

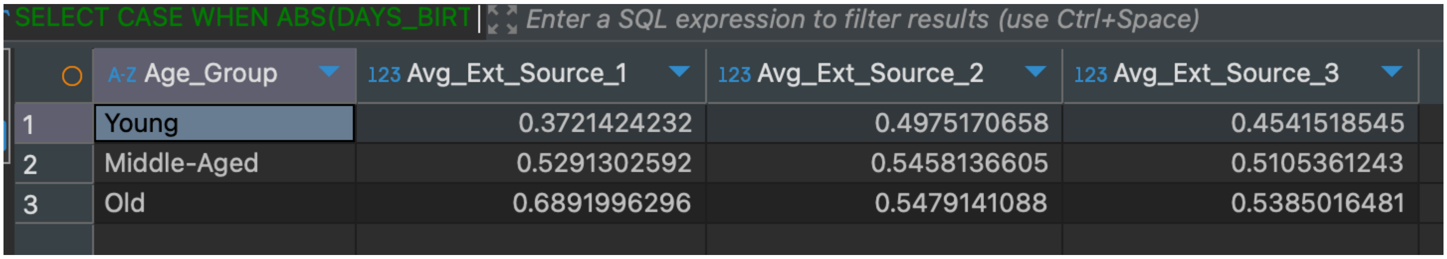
Task 8 FreeStyle

Now, conduct your own research and analysis to see what factors from the “application” and the “bureau” tables are affecting

- The Credit Scores
- The Payment Difficulty

1. Age and Credit Scores

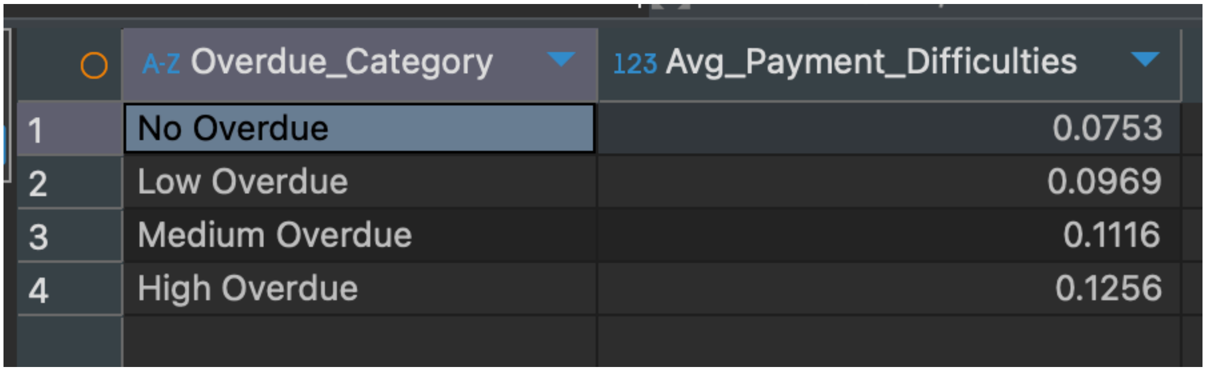
```
SELECT
    CASE
        WHEN ABS(DAYS_BIRTH) <= 35 * 365 THEN 'Young'
        WHEN ABS(DAYS_BIRTH) > 35 * 365 AND ABS(DAYS_BIRTH) <= 50 * 365 THEN 'Middle-Aged'
        ELSE 'Old'
    END AS Age_Group,
    AVG(EXT_SOURCE_1) AS Avg_Ext_Source_1,
    AVG(EXT_SOURCE_2) AS Avg_Ext_Source_2,
    AVG(EXT_SOURCE_3) AS Avg_Ext_Source_3
FROM application
WHERE EXT_SOURCE_1 IS NOT NULL
  AND EXT_SOURCE_2 IS NOT NULL
  AND EXT_SOURCE_3 IS NOT NULL
GROUP BY Age_Group
ORDER BY
    CASE
        WHEN Age_Group = 'Young' THEN 1
        WHEN Age_Group = 'Middle-Aged' THEN 2
        ELSE 3
    END;
```


SELECT CASE WHEN ABS(DAYS_BIRT
Enter a SQL expression to filter results (use Ctrl+Space)

	Age_Group	Avg_Ext_Source_1	Avg_Ext_Source_2	Avg_Ext_Source_3
1	Young	0.3721424232	0.4975170658	0.4541518545
2	Middle-Aged	0.5291302592	0.5458136605	0.5105361243
3	Old	0.6891996296	0.5479141088	0.5385016481

2. Max Overdue Amount and Payment Difficulties

```
SELECT
   CASE
            WHEN AMT_CREDIT_MAX_OVERDUE = 0 THEN 'No Overdue'
            WHEN AMT_CREDIT_MAX_OVERDUE > 0 and AMT_CREDIT_MAX_OVERDUE <= 50000 THEN
'Low Overdue'
      WHEN AMT_CREDIT_MAX_OVERDUE > 50000 AND AMT_CREDIT_MAX_OVERDUE <= 200000
THEN 'Medium Overdue'
      ELSE 'High Overdue'
   END AS Overdue_Category,
   AVG(a.TARGET) AS Avg_Payment_Difficulties
FROM bureau b
JOIN application a ON b.SK_ID_CURR = a.SK_ID_CURR
WHERE b.AMT_CREDIT_MAX_OVERDUE IS NOT NULL
GROUP BY Overdue_Category
ORDER BY
   CASE
      WHEN Overdue_Category = 'No Overdue' THEN 1
      WHEN Overdue_Category = 'Low Overdue' THEN 2
      WHEN Overdue_Category = 'Medium Overdue' THEN 3
      ELSE 4
   END;
```



	Overdue_Category	Avg_Payment_Difficulties
1	No Overdue	0.0753
2	Low Overdue	0.0969
3	Medium Overdue	0.1116
4	High Overdue	0.1256