Credit Risk Analysis Based on Client Data

Teofilus San Prasetya

Dataset :

Link : https://www.kaggle.com/datasets/rikdifos/credit-card-approval-prediction

Data :

1. **application_record.csv :**

Feature Name	Explanation	Remarks
ID	Client number	Unique identifier for each client
CODE_GENDER	Gender	Specifies the gender of the client
FLAG_OWN_CAR	Is there a car	Indicates if the client owns a car (1 = Yes, 0 = No)
FLAG_OWN_REALTY	Is there a property	Indicates if the client owns real estate property (1 = Yes, 0 = No)
CNT_CHILDREN	Number of children	Number of children dependent on the client
AMT_INCOME_TOTAL	Annual income	Total annual income of the client
NAME_INCOME_TYPE	Income category	Type of income source (e.g., Working, Pensioner, Commercial associate)
NAME_EDUCATION_TYPE	Education level	Highest level of education attained by the client
NAME_FAMILY_STATUS	Marital status	The client's marital status (e.g., Single, Married, Separated)
NAME_HOUSING_TYPE	Way of living	Type of housing the client resides in (e.g., House, Apartment)
DAYS_BIRTH	Birthday	Client's age, counted backward from the current day (e.g., -1 = yesterday)
DAYS_EMPLOYED	Start date of employment	Employment tenure, counted backward. Positive values indicate unemployment
FLAG_MOBIL	Is there a mobile phone	Whether the client owns a mobile phone (1 = Yes, 0 = No)

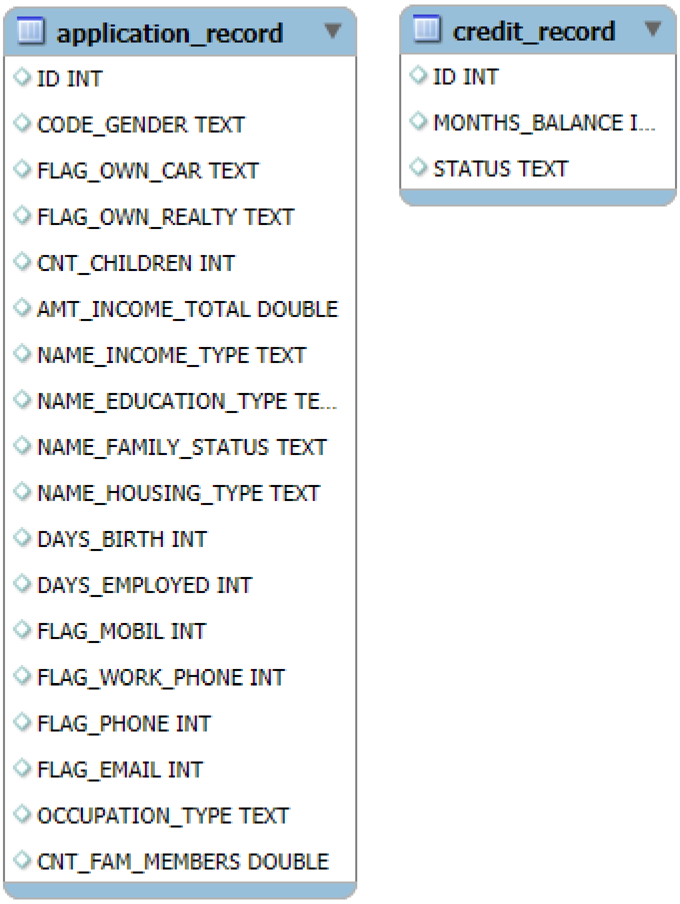
Feature Name	Explanation	Remarks
FLAG_WORK_PHONE	Is there a work phone	Whether the client has a work phone (1 = Yes, 0 = No)
FLAG_PHONE	Is there a phone	Whether the client has a phone (1 = Yes, 0 = No)
FLAG_EMAIL	Is there an email	Whether the client has an email address (1 = Yes, 0 = No)
OCCUPATION_TYPE	Occupation	The type of occupation of the client
CNT_FAM_MEMBERS	Family size	Total number of family members, including the client

2. **credit_record.csv :**

Feature Name	Explanation	Remarks
ID	Client number	Unique identifier for each client
MONTHS_BALANCE	Record month	The month relative to the current one (e.g., 0 = current, -1 = previous)
STATUS	Credit status	Status of the client's credit for that month: - 0: 1-29 days past due - 1: 30-59 days past due - 2: 60-89 days past due - 3: 90-119 days past due - 4: 120-149 days past due - 5: Overdue or bad debts for more than 150 days - C: Paid off that month - X: No loan for that month

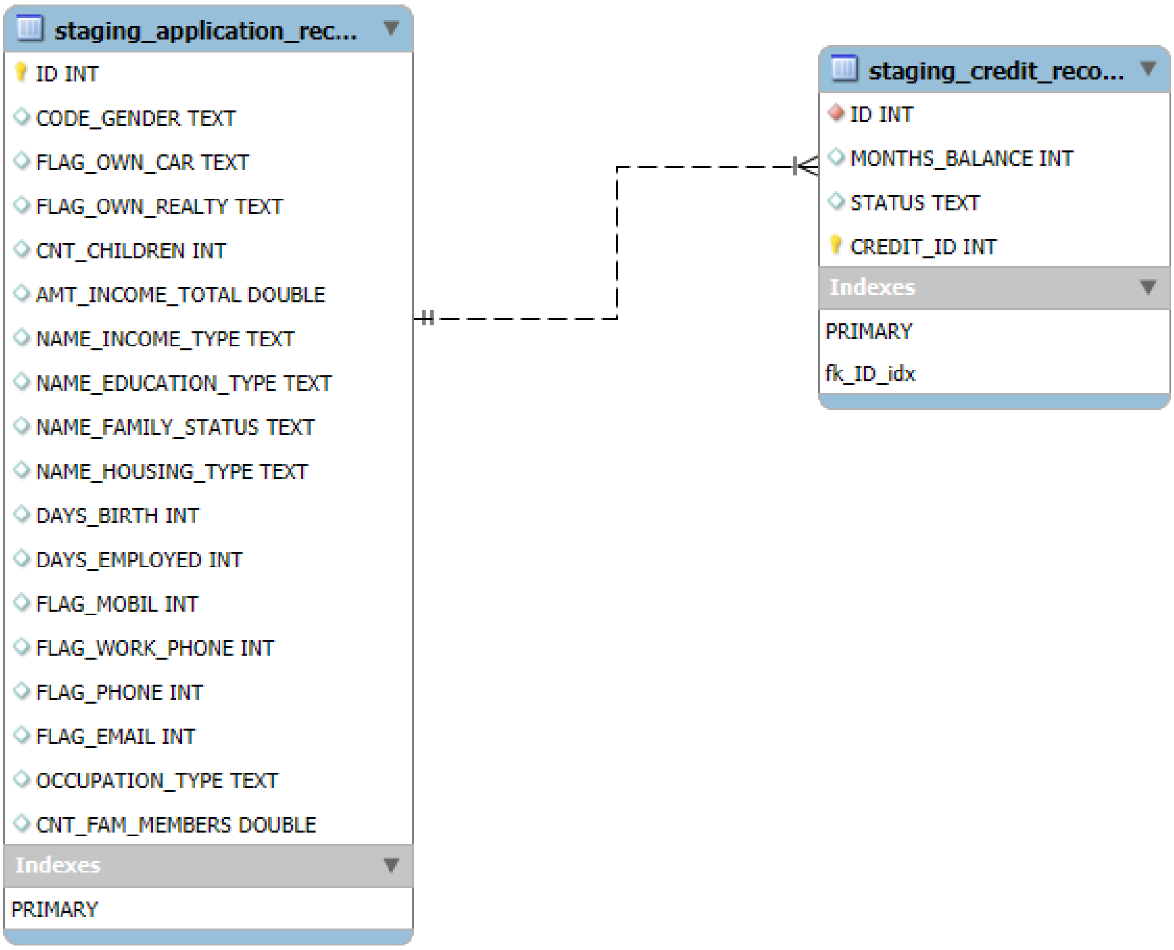
This dataset focuses on predicting credit risk by analyzing applicants demographic, financial, and credit history information. It consists of two tables: “application_record.csv”, which includes details like gender, income, employment, and family size, and “credit_record.csv”, which tracks monthly credit statuses, such as days past due or loan repayment.

The goal is to extract insights from these data to enhance credit risk predictions. By identifying patterns between client behavior and credit status, we aim to improve the classification of clients as “good” or “bad” based on their likelihood of default. This analysis supports building more accurate predictive models and addresses challenges such as data imbalance, ultimately improving credit approval decisions and risk management.



Initially, the staging_credit_record table only used the customer ID to identify credit records without any additional unique key, which could potentially lead to data duplication and difficulty in distinguishing each record, especially if a customer had multiple records with different statuses and months. To address this issue, credit_id was added as the primary key. This addition provides a unique identity to each credit record, facilitating data manipulation processes such as updates or deletions, and maintaining data integrity by preventing duplication and reference errors between tables.

ERD :



The one-to-many relationship between the "staging_application_record" and "staging_credit_record" tables in the "Credit Risk Analysis" ERD indicates that one customer (identified by "ID" as the Primary Key in "staging_application_record") can have multiple credit records (identified by "credit_id" as the Primary Key in "staging_credit_record"). In the "staging_credit_record" table, "ID" serves as the Foreign Key, linking each credit record back to the corresponding customer. This structure allows the system to store a comprehensive credit history for each customer, ensuring data integrity and preventing duplication. With this setup, data analysis becomes more efficient and accurate, supporting credit risk assessment based on customer behavior patterns.

Data Cleaning Project: Customer Bank Data

Project Description: In this project, I performed data cleaning on a dataset from a banking institution, aimed at preparing the data for further analysis. The dataset contains loan application information and credit records of customers, comprising two main tables: application_record and credit_record.

Steps Taken:

1. **Database and Table Creation:**

   Created the database customer_bank_data and defined two main tables (application_record and credit_record) with appropriate structure, including data type definitions and setting primary and foreign keys.

   Query :

```
CREATE DATABASE customer_bank_data;

USE customer_bank_data;


CREATE TABLE application_record (
  ID INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
  CODE_GENDER VARCHAR(5) NOT NULL,
  FLAG_OWN_CAR VARCHAR(5) NOT NULL,
  FLAG_OWN_REALTY VARCHAR(5) NOT NULL,
  CNT_CHILDREN INT NOT NULL,
  AMT_INCOME_TOTAL DECIMAL(15,2) NOT NULL,
  NAME_INCOME_TYPE VARCHAR(255) NOT NULL,
  NAME_EDUCATION_TYPE VARCHAR(255) NOT NULL,
  NAME_FAMILY_STATUS VARCHAR(255) NOT NULL,
  NAME_HOUSING_TYPE VARCHAR(255) NOT NULL,
  DAYS_BIRTH INT NOT NULL,
  DAYS_EMPLOYED INT NOT NULL,
  FLAG_MOBIL INT NOT NULL,
  FLAG_WORK_PHONE INT NOT NULL,
  FLAG_PHONE INT NOT NULL,
  FLAG_EMAIL INT NOT NULL,
  OCCUPATION_TYPE VARCHAR(255) NOT NULL,
  CNT_FAM_MEMBERS INT NOT NULL
```

```
);
```

```
CREATE TABLE credit_record (
    credit_id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    ID INT NOT NULL,
    MONTHS_BALANCE INT NOT NULL,
    STATUS VARCHAR(5) NOT NULL,
    FOREIGN KEY (ID) REFERENCES application_record(ID)
);
```

2. **Staging Table Creation:**

   Created staging tables to work with raw data, allowing me to perform operations without affecting the original dataset.

   Query :

```
CREATE TABLE customer_bank_data.staging_application_record AS
SELECT * FROM customer_bank_data.application_record;
```


ID	CODE_GENDER	FLAG_OWN_CAR	FLAG_OWN_REALTY	CNT_CHILDREN	AMT_INCOME_TOTAL	NAME_INCOME_TYPE	NAME_EDUCATION_TYPE	NAME_FAMILY_STATUS	NAME_HOUSING_TYPE	DAYS_BIRTH	DA
6563539	F	N	Y	1	157500	Working	Secondary / secondary special	Married	House / apartment	-11505	-36
6563540	F	N	Y	1	157500	Working	Secondary / secondary special	Married	House / apartment	-11505	-36
6563541	F	N	Y	1	157500	Working	Secondary / secondary special	Married	House / apartment	-11505	-36
6563542	F	N	Y	1	157500	Working	Secondary / secondary special	Married	House / apartment	-11505	-36
6563543	F	N	Y	1	157500	Working	Secondary / secondary special	Married	House / apartment	-11505	-36
6563544	F	N	Y	1	157500	Working	Secondary / secondary special	Married	House / apartment	-11505	-36
6563554	M	N	Y	0	180000	Working	Secondary / secondary special	Married	House / apartment	-13672	-23
6563555	M	N	Y	0	180000	Working	Secondary / secondary special	Married	House / apartment	-13672	-23
6563574	F	N	Y	1	157500	Working	Secondary / secondary special	Married	House / apartment	-17756	-50
6563592	F	N	Y	0	67500	Pensioner	Secondary / secondary special	Civil marriage	House / apartment	-23390	36[illegible]
6563594	F	N	Y	0	67500	Pensioner	Secondary / secondary special	Civil marriage	House / apartment	-23390	36[illegible]
6563597	M	Y	Y	0	292500	Commercial associate	Higher education	Married	House / apartment	-19546	-78
6563600	M	Y	Y	0	292500	Commercial associate	Higher education	Married	House / apartment	-19546	-78
6563601	M	Y	Y	0	292500	Commercial associate	Higher education	Married	House / apartment	-19546	-78
6563605	M	Y	Y	0	292500	Commercial associate	Higher education	Married	House / apartment	-19546	-78
6563606	F	N	N	2	157500	Working	Higher education	Separated	House / apartment	-12623	-67
6563607	F	Y	Y	2	157500	Commercial associate	Higher education	Married	House / apartment	-14240	-25
6563617	F	N	N	0	270000	State servant	Secondary / secondary special	Married	House / apartment	-19850	-79

```
CREATE TABLE customer_bank_data.staging_credit_record AS
SELECT * FROM customer_bank_data.credit_record;
```

	ID	MONTHS_BALANCE	STATUS
▶	5001711	0	X
	5001711	-1	0
	5001711	-2	0
	5001711	-3	0
	5001712	0	C
	5001712	-1	C
	5001712	-2	C
	5001712	-3	C
	5001712	-4	C
	5001712	-5	C
	5148918	-16	0
	5001712	-6	C
	5148918	-17	0
	5001712	-7	C
	5001712	-8	C
	5001712	-9	0
	5001712	-10	0
	5001712	-11	0

	Tables_in_customer_bank_data
▶	application_record
	credit_record
	staging_application_record
	staging_credit_record

3. **Add the credit_id column as PRIMARY KEY and AUTO_INCREMENT**
   Query :

   ALTER TABLE customer_bank_data.staging_credit_record

   ADD CREDIT_ID INT NOT NULL AUTO_INCREMENT PRIMARY KEY;

	ID	MONTHS_BALANCE	STATUS	credit_id
▶	5116524	-42	0	1
	5045868	-16	C	2
	5116524	-43	0	3
	5045868	-17	C	4
	5116524	-44	0	5
	5116524	-45	X	6
	5116525	0	X	7
	5116525	-1	X	8
	5116525	-2	X	9
	5045868	-18	0	10
	5116525	-3	X	11
	5045869	0	C	12
	5116525	-4	X	13
	5045869	-1	C	14
	5116525	-5	X	15

4. **Duplicate Checking and Removal:**

Checked for duplicates in both staging tables and removed unnecessary entries.

Query :

```sql
SELECT COUNT(*), ID, CODE_GENDER, FLAG_OWN_CAR, FLAG_OWN_REALTY, CNT_CHILDREN, AMT_INCOME_TOTAL, NAME_INCOME_TYPE, NAME_EDUCATION_TYPE, NAME_FAMILY_STATUS, NAME_HOUSING_TYPE, DAYS_BIRTH, DAYS_EMPLOYED, FLAG_MOBIL, FLAG_WORK_PHONE, FLAG_PHONE, FLAG_EMAIL, OCCUPATION_TYPE, CNT_FAM_MEMBERS

FROM customer_bank_data.staging_application_record

GROUP BY ID, CODE_GENDER, FLAG_OWN_CAR, FLAG_OWN_REALTY, CNT_CHILDREN, AMT_INCOME_TOTAL, NAME_INCOME_TYPE, NAME_EDUCATION_TYPE, NAME_FAMILY_STATUS, NAME_HOUSING_TYPE, DAYS_BIRTH, DAYS_EMPLOYED, FLAG_MOBIL, FLAG_WORK_PHONE, FLAG_PHONE, FLAG_EMAIL, OCCUPATION_TYPE, CNT_FAM_MEMBERS

HAVING COUNT(*) > 1;


-- Remove duplicates in application_record

DELETE t1 FROM customer_bank_data.staging_application_record t1

INNER JOIN customer_bank_data.staging_application_record t2

WHERE t1.ID > t2.ID

AND t1.ID = t2.ID

AND t1.CODE_GENDER = t2.CODE_GENDER

AND t1.FLAG_OWN_CAR = t2.FLAG_OWN_CAR

AND t1.FLAG_OWN_REALTY = t2.FLAG_OWN_REALTY

AND t1.CNT_CHILDREN = t2.CNT_CHILDREN

AND t1.AMT_INCOME_TOTAL = t2.AMT_INCOME_TOTAL

AND t1.NAME_INCOME_TYPE = t2.NAME_INCOME_TYPE

AND t1.NAME_EDUCATION_TYPE = t2.NAME_EDUCATION_TYPE

AND t1.NAME_FAMILY_STATUS = t2.NAME_FAMILY_STATUS

AND t1.NAME_HOUSING_TYPE = t2.NAME_HOUSING_TYPE

AND t1.DAYS_BIRTH = t2.DAYS_BIRTH
```

```
AND t1.DAYS_EMPLOYED = t2.DAYS_EMPLOYED

AND t1.FLAG_MOBIL = t2.FLAG_MOBIL

AND t1.FLAG_WORK_PHONE = t2.FLAG_WORK_PHONE

AND t1.FLAG_PHONE = t2.FLAG_PHONE

AND t1.FLAG_EMAIL = t2.FLAG_EMAIL

AND t1.OCCUPATION_TYPE = t2.OCCUPATION_TYPE

AND t1.CNT_FAM_MEMBERS = t2.CNT_FAM_MEMBERS;
```

5. **Cleaning Specific Columns:**

   Checked for hidden characters in the STATUS column from credit_record and cleaned up those entries.

   Query :

```
SELECT DISTINCT STATUS, LENGTH(STATUS) AS length_of_status

FROM customer_bank_data.credit_record

ORDER BY STATUS;
```


STATUS	length_of_status
0	1
0	2
1	1
1	2
2	2
3	2
4	2
5	2
C	1
C	2
X	1
X	2

```
-- Clean STATUS column values

UPDATE customer_bank_data.staging_credit_record

SET STATUS = REPLACE(REPLACE(REPLACE(REPLACE(STATUS, CHAR(9), ''), CHAR(10),
''), CHAR(13), ''), ' ', '');
```

	STATUS	length_of_status
▶	0	1
	1	1
	2	1
	3	1
	4	1
	5	1
	C	1
	X	1

6. **Standardization and Handling Null Values:**

   Standardized values for certain columns and checked for NULL values in various fields. Handled empty string values by assigning appropriate default values.

   Query :

```
UPDATE customer_bank_data.staging_application_record

SET OCCUPATION_TYPE = NULL

WHERE OCCUPATION_TYPE = '';
```


empty_occupation_type
20545

FLAG_PHONE	FLAG_EMAIL	OCCUPATION_TYPE	CNT_FAM_MEMBERS
1	1	Core staff	3
1	1	Core staff	3
0	0	Drivers	2
0	0	Drivers	2
0	0		3
0	0		2
0	0		2
0	0	Managers	2
0	0	Managers	2
0	0	Managers	2

```
-- Handle null values in the OCCUPATION_TYPE column

UPDATE customer_bank_data.staging_application_record

SET OCCUPATION_TYPE = COALESCE(OCCUPATION_TYPE, 'Unknown');
```

FLAG_PHONE	FLAG_EMAIL	OCCUPATION_TYPE	CNT_FAM_MEMBERS
1	1	Core staff	3
1	1	Core staff	3
0	0	Drivers	2
0	0	Drivers	2
0	0	Unknown	3
0	0	Unknown	2
0	0	Unknown	2
0	0	Managers	2
0	0	Managers	2
0	0	Managers	2

7. **Final Review of Cleaned Data:**

   Conducted a final review of the cleaned data to ensure all operations were successfully applied.

   Query :

   SELECT * FROM customer_bank_data.staging_application_record;

ID	CODE_GENDER	FLAG_OWN_CAR	FLAG_OWN_REALTY	CNT_CHILDREN	AMT_INCOME_TOTAL	NAME_INCOME_TYPE	NAME_EDUCATION_TYPE	NAME_FAMILY_STATUS	NAME_HOUSING_TYPE	DAYS_BIRTH	DA
6563539	F	N	Y	1	157500	Working	Secondary / secondary special	Married	House / apartment	-11505	-36
6563540	F	N	Y	1	157500	Working	Secondary / secondary special	Married	House / apartment	-11505	-36
6563541	F	N	Y	1	157500	Working	Secondary / secondary special	Married	House / apartment	-11505	-36
6563542	F	N	Y	1	157500	Working	Secondary / secondary special	Married	House / apartment	-11505	-36
6563543	F	N	Y	1	157500	Working	Secondary / secondary special	Married	House / apartment	-11505	-36
6563544	F	N	Y	1	157500	Working	Secondary / secondary special	Married	House / apartment	-11505	-36
6563554	M	N	Y	0	180000	Working	Secondary / secondary special	Married	House / apartment	-13672	-23
6563555	M	N	Y	0	180000	Working	Secondary / secondary special	Married	House / apartment	-13672	-23
6563574	F	N	Y	1	157500	Working	Secondary / secondary special	Married	House / apartment	-17756	-50
6563592	F	N	Y	0	67500	Pensioner	Secondary / secondary special	Civil marriage	House / apartment	-23390	365
6563594	F	N	Y	0	67500	Pensioner	Secondary / secondary special	Civil marriage	House / apartment	-23390	365
6563597	M	Y	Y	0	292500	Commercial associate	Higher education	Married	House / apartment	-19546	-78
6563600	M	Y	Y	0	292500	Commercial associate	Higher education	Married	House / apartment	-19546	-78
6563601	M	Y	Y	0	292500	Commercial associate	Higher education	Married	House / apartment	-19546	-78
6563605	M	Y	Y	0	292500	Commercial associate	Higher education	Married	House / apartment	-19546	-78
6563606	F	N	N	2	157500	Working	Higher education	Separated	House / apartment	-12623	-67
6563607	F	Y	Y	2	157500	Commercial associate	Higher education	Married	House / apartment	-14240	-25
6563617	F	N	N	0	270000	State servant	Secondary / secondary special	Married	House / apartment	-19850	-79

   SELECT * FROM customer_bank_data.staging_credit_record;

ID	MONTHS_BALANCE	STATUS	credit_id
5116524	-42	0	1
5045868	-16	C	2
5116524	-43	0	3
5045868	-17	C	4
5116524	-44	0	5
5116524	-45	X	6
5116525	0	X	7
5116525	-1	X	8
5116525	-2	X	9
5045868	-18	0	10
5116525	-3	X	11
5045869	0	C	12
5116525	-4	X	13
5045869	-1	C	14
5116525	-5	X	15

Data Analysis Project: Customer Bank Data

Project Description: This data analysis project involved examining various aspects of customer profiles from a banking institution, particularly focusing on their credit records and payment behaviors. Using SQL, I conducted exploratory data analysis to uncover relationships between customer characteristics (e.g., income, occupation, family size, education) and their credit payment statuses.

Steps and Insights:

1. **Customer Distribution by Payment Status:**

   Analyzed the number of customers in each payment delay category (status ranging from 0-5, C, X).

   This provides an understanding of the overall distribution of customers in terms of payment behavior, from those paying on time to those experiencing significant delays.

   Query :

```
SELECT
  STATUS,
  COUNT(*) AS num_customers
FROM customer_bank_data.staging_credit_record
GROUP BY STATUS;
```

STATUS	num_customers
0	383176
C	442089
X	209338
1	11092
2	868
5	1693
4	223
3	320

2. **Average Annual Income by Payment Status:**

   Examined the average income of customers based on their payment status to see if financial standing correlates with credit repayment behavior.

   Query :

```
SELECT
```

```
  cr.STATUS,
  AVG(ar.AMT_INCOME_TOTAL) AS avg_income
FROM customer_bank_data.staging_application_record ar
JOIN customer_bank_data.staging_credit_record cr
ON ar.ID = cr.ID
GROUP BY cr.STATUS;
```

STATUS	avg_income
0	185892.56462585033
2	186620.850931677
3	202260.3896103896
X	171591.7943107221
5	222126.10169491524
C	187653.32197614992
1	181425.3298969072
4	219465.7894736842

3. **Payment Delay by Income Category:**

   Explored the relationship between income type (e.g., working, business owners) and payment delays, revealing whether different income groups tend to experience varying degrees of payment delays.

   Query :

```
SELECT
  ar.NAME_INCOME_TYPE,
  cr.STATUS,
  COUNT(*) AS num_customers
FROM customer_bank_data.staging_application_record ar
JOIN customer_bank_data.staging_credit_record cr
ON ar.ID = cr.ID
GROUP BY ar.NAME_INCOME_TYPE, cr.STATUS;
```

	NAME_INCOME_TYPE	STATUS	num_customers
▶	Pensioner	X	215
	Pensioner	0	640
	Pensioner	2	108
	Commercial associate	5	210
	Commercial associate	4	33
	Commercial associate	3	45
	Commercial associate	2	110
	Commercial associate	1	199
	Commercial associate	X	104
	Working	2	243
	Working	1	477
	Working	0	1360
	Working	X	522
	Commercial associate	0	507
	Pensioner	C	818
	Pensioner	1	144
	Pensioner	5	125

4. **Occupation Type and Credit Payment Delays:**

   Investigated how customers from different occupations (e.g., managers, laborers) perform in terms of credit payments, identifying specific job roles associated with higher default risks.

   Query :

```
SELECT
  ar.OCCUPATION_TYPE,
  cr.STATUS,
  COUNT(*) AS num_customers
FROM customer_bank_data.staging_application_record ar
JOIN customer_bank_data.staging_credit_record cr
ON ar.ID = cr.ID
GROUP BY ar.OCCUPATION_TYPE, cr.STATUS;
```

OCCUPATION_TYPE	STATUS	num_customers
Unknown	X	313
Unknown	0	811
Unknown	2	151
Low-skill Laborers	5	44
Low-skill Laborers	4	2
Low-skill Laborers	3	2
Low-skill Laborers	2	11
Low-skill Laborers	1	8
Low-skill Laborers	X	15
Sales staff	2	42
Sales staff	1	72
Sales staff	0	197
Core staff	X	168
Core staff	0	350
Core staff	2	57
Core staff	1	125
Cleaning staff	0	67

5. **Influence of Family Size on Credit Delays:**

Analyzed whether family size influences payment delays, looking at trends in larger versus smaller families.

Query :

```
SELECT
  ar.CNT_FAM_MEMBERS,
  cr.STATUS,
  COUNT(*) AS num_customers
FROM customer_bank_data.staging_application_record ar
JOIN customer_bank_data.staging_credit_record cr
ON ar.ID = cr.ID
GROUP BY ar.CNT_FAM_MEMBERS, cr.STATUS;
```

	CNT_FAM_MEMBERS	STATUS	num_customers
▶	2	X	548
	2	0	1504
	2	2	276
	3	5	145
	3	4	20
	3	3	17
	3	2	71
	3	1	113
	3	X	77
	2	1	540
	1	X	220
	1	0	607
	1	2	101
	1	1	169
	1	C	841

6. **Analysis of Significant Delays (60+ Days):**

Focused on customers with severe delays (status 2-5) by occupation and family size to determine which customer segments are most prone to these payment issues.

Query :

```
SELECT
  ar.OCCUPATION_TYPE,
  ar.CNT_FAM_MEMBERS,
  COUNT(*) AS num_customers
FROM customer_bank_data.staging_application_record ar
JOIN customer_bank_data.staging_credit_record cr
ON ar.ID = cr.ID
WHERE cr.STATUS IN ('2', '3', '4', '5')
GROUP BY ar.OCCUPATION_TYPE, ar.CNT_FAM_MEMBERS;
```

	OCCUPATION_TYPE	CNT_FAM_MEMBERS	num_customers
▶	Sales staff	2	19
	Core staff	1	35
	Cleaning staff	2	15
	Unknown	1	96
	Core staff	5	10
	Laborers	1	27
	High skill tech staff	4	17
	Drivers	2	54
	Cooking staff	1	4
	Sales staff	1	33
	Laborers	2	156
	Core staff	3	48
	Managers	2	74
	Core staff	2	38
	Low-skill Laborers	2	9
	Unknown	2	260

7. **Employment Duration and Credit Payment Delays:**

Examined the impact of employment duration on credit payment delays, identifying whether longer employment corresponds to timely payments.

Query :

```
SELECT
  ABS(ar.DAYS_EMPLOYED) AS employment_duration,
  cr.STATUS,
  COUNT(*) AS num_customers
FROM customer_bank_data.staging_application_record ar
JOIN customer_bank_data.staging_credit_record cr
ON ar.ID = cr.ID
GROUP BY employment_duration, cr.STATUS;
```

employment_duration	STATUS	num_customers
365243	X	203
365243	0	529
365243	2	84
2269	5	44
2269	4	2
2269	3	2
2269	2	2
2269	1	2
2269	X	2
655	2	1
655	1	1
655	0	8
779	X	27
779	0	9
779	2	1
779	1	1

8. **Education Level and Payment Delays:**

Investigated how customer education levels (e.g., secondary, higher education) are related to their credit payment status.

Query :

```
SELECT
  ar.NAME_EDUCATION_TYPE,
  cr.STATUS,
  COUNT(*) AS num_customers
FROM customer_bank_data.staging_application_record ar
JOIN customer_bank_data.staging_credit_record cr
ON ar.ID = cr.ID
GROUP BY ar.NAME_EDUCATION_TYPE, cr.STATUS;
```

NAME_EDUCATION_TYPE	STATUS	num_customers
Secondary / secondary special	2	318
Secondary / secondary special	1	521
Secondary / secondary special	0	1893
Secondary / secondary special	X	596
Incomplete higher	C	103
Incomplete higher	0	100
Incomplete higher	2	20
Incomplete higher	1	84
Secondary / secondary special	C	2389
Secondary / secondary special	3	101
Higher education	C	1030
Higher education	2	140
Higher education	1	261
Higher education	0	598
Secondary / secondary special	5	367
Secondary / secondary special	4	71

9. **Car Ownership and Payment Delays:**

Analyzed whether customers owning cars were more or less likely to experience payment delays, providing insights into the relationship between asset ownership and credit behavior.

Query :

```
SELECT
  ar.FLAG_OWN_CAR,
  cr.STATUS,
  COUNT(*) AS num_customers
FROM customer_bank_data.staging_application_record ar
JOIN customer_bank_data.staging_credit_record cr
ON ar.ID = cr.ID
GROUP BY ar.FLAG_OWN_CAR, cr.STATUS;
```

	FLAG_OWN_CAR	STATUS	num_customers
▶	N	X	664
	N	0	1788
	N	2	289
	N	5	382
	N	4	72
	N	3	92
	N	1	536
	N	C	2305
	Y	5	208
	Y	4	42
	Y	3	62
	Y	2	194
	Y	1	337
	Y	0	858
	Y	C	1217
	Y	X	250

10. **Customer Age and Credit Payment Delays:**

    Explored the influence of customer age on payment behavior, identifying whether certain age groups are more likely to experience delays.

    Query :

```
SELECT
  ABS(ar.DAYS_BIRTH / 365) AS age,
  cr.STATUS,
  COUNT(*) AS num_customers
FROM customer_bank_data.staging_application_record ar
JOIN customer_bank_data.staging_credit_record cr
ON ar.ID = cr.ID
GROUP BY age, cr.STATUS;
```

age	STATUS	num_customers
58.3890	X	37
58.3890	0	49
58.3890	2	4
42.3726	5	44
42.3726	4	2
42.3726	3	2
42.3726	2	2
42.3726	1	2
42.3726	X	2
49.2000	2	1
49.2000	1	1
49.2000	0	8
33.2603	X	27
33.2603	0	9
33.2603	2	1
33.2603	1	1

11. Unemployed Customers and Payment Delays

Explored the correlation between unemployment and credit payment delays, identifying the distribution of payment statuses for currently unemployed customers.

Query :

```
SELECT
  cr.STATUS,
  COUNT(*) AS num_unemployed_customers
FROM customer_bank_data.staging_application_record ar
JOIN customer_bank_data.staging_credit_record cr
ON ar.ID = cr.ID
WHERE ar.DAYS_EMPLOYED > 0
GROUP BY cr.STATUS;
```

	STATUS	num_unemployed_customers
▶	0	529
	2	84
	3	36
	X	203
	5	78
	C	635
	1	115
	4	16

12. **Employment Duration and Payment Delays (30-59 Days)**

Analyzed the average number of days customers have been employed for those with 30-59 day payment delays, providing insights into whether employment duration affects this delay category.

Query :

```
SELECT
  AVG(ar.DAYS_EMPLOYED) AS avg_days_employed
FROM customer_bank_data.staging_application_record ar
JOIN customer_bank_data.staging_credit_record cr
ON ar.ID = cr.ID
WHERE cr.STATUS = '1';
```

	avg_days_employed
▶	46414.5750

13. **Children and Payment Delays Over 90 Days**

Investigated the relationship between the number of children customers have and the likelihood of experiencing payment delays over 90 days, assessing if family size impacts credit behavior.

Query :

```
SELECT
  ar.CNT_CHILDREN,
  COUNT(cr.STATUS) AS num_late_customers
```

```
FROM customer_bank_data.staging_application_record ar
JOIN customer_bank_data.staging_credit_record cr
ON ar.ID = cr.ID
WHERE cr.STATUS IN ('3', '4', '5')
GROUP BY ar.CNT_CHILDREN;
```

CNT_CHILDREN	num_late_customers
1	253
0	567
2	32
3	2
4	4

14. Family Size and Credit Payment Delays

Examined how the number of family members correlates with credit payment delays, identifying if larger family sizes increase the likelihood of delayed payments.

Query :

```
SELECT
  ar.CNT_FAM_MEMBERS,
  cr.STATUS,
  COUNT(*) AS num_customers
FROM customer_bank_data.staging_application_record ar
JOIN customer_bank_data.staging_credit_record cr
ON ar.ID = cr.ID
GROUP BY ar.CNT_FAM_MEMBERS, cr.STATUS;
```

CNT_FAM_MEMBERS	STATUS	num_customers
2	X	548
2	0	1504
2	2	276
3	5	145
3	4	20
3	3	17
3	2	71
3	1	113
3	X	77
2	1	540
1	X	220
1	0	607
1	2	101
1	1	169
1	C	841
5	C	56
5	X	27

15. Average Family Size for On-Time Payments

Calculated the average family size of customers who pay their credit on time, offering a baseline comparison against customers with delayed payments.

Query :

```
SELECT
  AVG(ar.CNT_FAM_MEMBERS) AS avg_family_size
FROM customer_bank_data.staging_application_record ar
JOIN customer_bank_data.staging_credit_record cr
ON ar.ID = cr.ID
WHERE cr.STATUS = 'C';
```

avg_family_size
2.0636002271436684

16. **Property Ownership and Payment Delays**

Analyzed how property ownership influences payment behavior, identifying whether customers who own property are more or less likely to experience credit payment delays.

Query :

```
SELECT
  ar.FLAG_OWN_REALTY,
  cr.STATUS,
  COUNT(*) AS num_customers
FROM customer_bank_data.staging_application_record ar
JOIN customer_bank_data.staging_credit_record cr
ON ar.ID = cr.ID
GROUP BY ar.FLAG_OWN_REALTY, cr.STATUS;
```

FLAG_OWN_REALTY	STATUS	num_customers
Y	X	549
Y	0	1568
Y	2	272
Y	5	330
Y	4	63
Y	3	76
Y	1	501
Y	C	1793
N	X	365
N	0	1078
N	5	260
N	C	1729
N	3	78
N	2	211
N	4	51
N	1	372

17. Monthly Trends in Payment Delays

Tracked trends in payment delays over time by analyzing monthly data, highlighting potential patterns in customer payment behaviors.

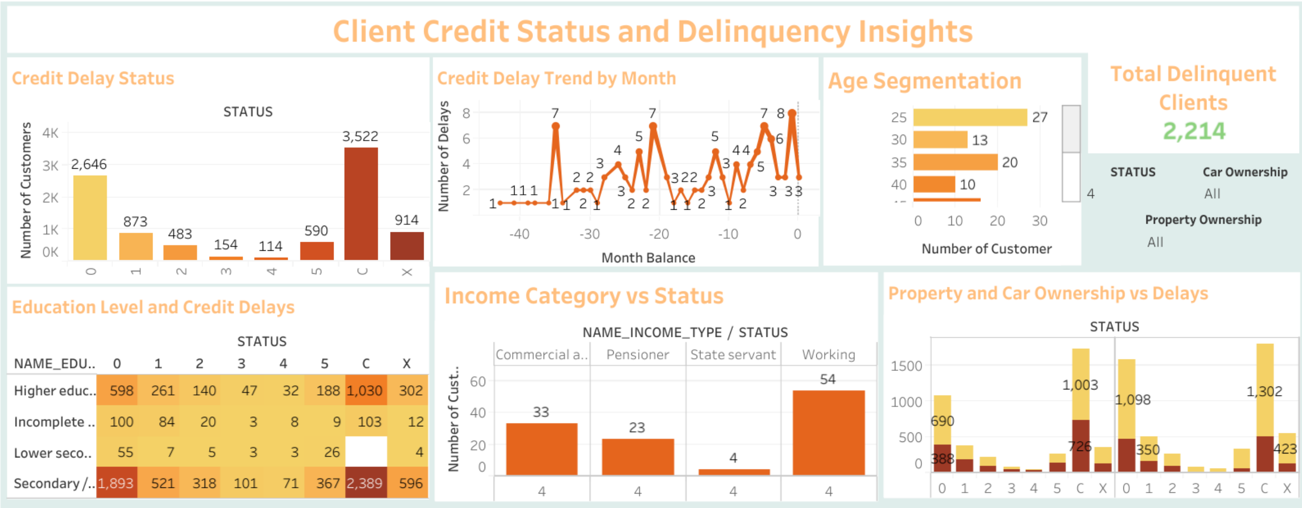
Query :

```sql
SELECT

  cr.MONTHS_BALANCE,

  COUNT(*) AS num_late_customers

FROM customer_bank_data.staging_credit_record cr

WHERE cr.STATUS IN ('2', '3', '4', '5')

GROUP BY cr.MONTHS_BALANCE

ORDER BY cr.MONTHS_BALANCE;
```

MONTHS_BALANCE	num_late_customers
-56	2
-55	3
-54	5
-53	10
-52	11
-51	17
-50	25
-49	28
-48	32
-47	35
-46	33
-45	33
-44	37
-43	39
-42	38
-41	50

Dashboard :

Link :
https://public.tableau.com/views/ClientCreditStatusandDelinquencyInsight/Dashboard1?:language=en-US&:sid=&:redirect=auth&:display_count=n&:origin=viz_share_link



1. **Credit Delay Status:**
   This bar chart breaks down the distribution of customers across different credit delay statuses. Most customers have no delays (status '0'), but there are significant numbers with delays (status '1' to '5'), particularly in statuses '5' and 'C' (those who have completely paid off their credit).
2. **Credit Delay Trend by Month:**
   This line graph shows how credit delays fluctuate over time, represented by the number of delays at different points in the month balance. The spikes suggest certain periods are more prone to credit delays, which could help in targeting interventions.
3. **Age Segmentation:**
   The age segmentation bar chart highlights the distribution of credit delay statuses across different age groups. This could help identify which age groups are more prone to experiencing delays and support age-targeted strategies.
4. **Total Delinquent Clients:**
   A clear indicator that 2,214 clients are delinquent in their payments. This metric allows for a quick understanding of the overall scope of delinquent clients.
5. **Education Level and Credit Delays:**
   This heatmap cross-references education levels with credit delay statuses, showing that customers with secondary and higher education have more delayed payments. The insights can guide financial institutions in tailoring services based on educational background.
6. **Income Category vs Status:**
   This bar chart compares income categories (commercial associates, pensioners, state servants, and working clients) with credit delay statuses. Notably, most delayed payments occur among working clients, indicating income category relevance in credit delay analysis.
7. **Property and Car Ownership vs Delays:**
   The stacked bar chart shows how property and car ownership influence credit delay statuses. It reveals that customers owning both property and cars are more likely to experience delays, suggesting that asset ownership might contribute to financial strain.