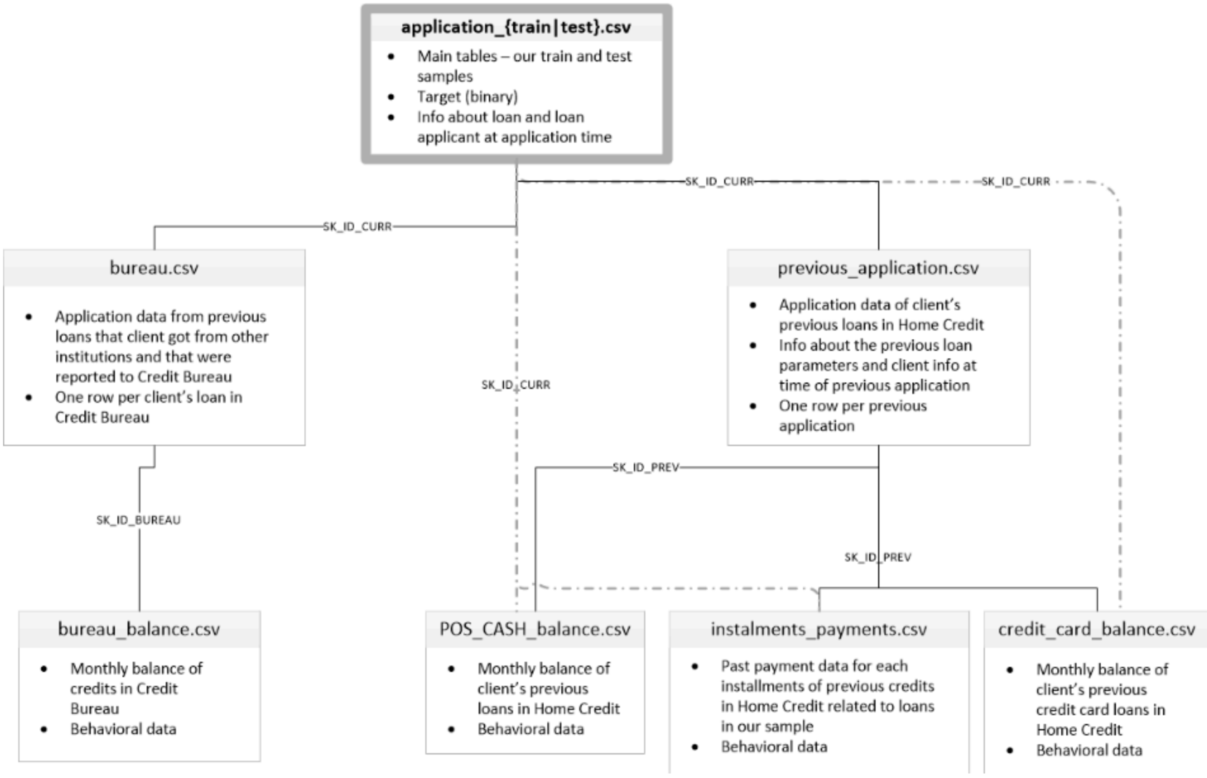
Home Credit Default Risk



Loan Process Mapped to Columns

Let's say you (the client) go to a shop and buy a washing machine on EMI.

Step-by-Step Flow:

Step	Real Life	Dataset Column
You apply for EMI	Client applies for POS loan	**SK_ID_CURR** (Client ID)
Loan is approved	Each loan has a unique ID	**SK_ID_PREV** (Loan ID)
Monthly tracking starts	Each month, the loan status is recorded	**MONTHS_BALANCE** (e.g., -1 = last month, 0 = now)
You had to pay 12 EMIs	Total planned payments	**CNT_INSTALMENT**
You've paid 6 so far	Remaining 6 EMIs left	**CNT_INSTALMENT_FUTURE**
Contract is still ongoing	Contract is active or closed	**NAME_CONTRACT_STATUS** (Active, Completed)
You delayed 2 payments	Days overdue for that month	**SK_DPD** (Days Past Due)

You had serious delays (default)	DPD but in default terms	**SK_DPD_DEF**

Example

SK_ID_CURR: 123456
SK_ID_PREV: 78910
MONTHS_BALANCE: -3
CNT_INSTALMENT: 12
CNT_INSTALMENT_FUTURE: 9
NAME_CONTRACT_STATUS: Active
SK_DPD: 15
SK_DPD_DEF: 0

It means:

- Client **123456** took a loan (**78910**) 3 months ago.
- It was a 12-month EMI plan; 9 months are still left.
- As of 3 months ago, they were **15 days late** in payment, but not yet in official default.

Datasets:-

1.POS_CASH_balance (df_pos_cash)

This dataset contains **monthly balance snapshots** for previous loans (both installment and revolving) from **POS** (Point of Sale) and cash loan channels. It **records a snapshot of every month** for each **previous loan** the client has taken.

Column	Description

SK_ID_PREV	ID of previous credit (loan)
SK_ID_CURR	ID of client
MONTHS_BALANCE	Month relative to the loan balance snapshot (e.g., 0 is current)
CNT_INSTALMENT	Number of installment payments initially agreed
CNT_INSTALMENT_FUTURE	Remaining installment payments at the snapshot time
NAME_CONTRACT_STATUS	Current status of the contract (e.g., Active, Completed)
SK_DPD	Days past due on the loan (at that snapshot)
SK_DPD_DEF	Days past due, considering default logic

****

Key Terms:

POS (Point of Sale) :-
This refers to loans taken at the point of purchase — e.g., you buy a phone on EMI in a store or online and get instant credit.
Think of it like swiping your card or using EMI on Amazon — you get the product now and pay monthly.

Revolving Loan :-
This is a **flexible credit line** — you can borrow, repay, and borrow again within a credit limit (like a **credit card**).

Snapshot:-
A **monthly snapshot** is like a monthly report of the loan:
“What was the loan status this month? How many payments are left? Any delay?”

SK_ID_CURR vs SK_ID_PREV

- **SK_ID_CURR** → This is the Client ID ("client" here means customer like you or me)
- **SK_ID_PREV** → This is the Loan ID for a specific loan the client took

<u>One client (SK_ID_CURR) can have many loans, and each loan has a different SK_ID_PREV</u>

EMI (Equated Monthly Installment)

EMI is a **fixed monthly payment** that a borrower makes to repay a **loan** — like for a car, home, or credit card bill.

Each EMI includes:

- **Principal** (the original loan amount)

- **Interest** (the cost charged by the lender)

****

2.Application_train (df_application_train)

This table contains the main information about loan applications made by customers. It acts as the **fact table** is central to any analysis or modeling. Each row represents one loan application.

Column Name	**Description**
SK_ID_CURR	Unique customer ID for current loan application
TARGET	Target label: 1 = client defaulted on loan, 0 = repaid
NAME_CONTRACT_TYPE	Type of loan contract (e.g., Cash loans, Revolving loans)
CODE_GENDER	Gender of the applicant
FLAG_OWN_CAR	Flag: Does the applicant own a car (Y/N)
FLAG_OWN_REALTY	Flag: Does the applicant own a house/apartment (Y/N)
CNT_CHILDREN	Number of children the applicant has
AMT_INCOME_TOTAL	Annual income of the applicant
AMT_CREDIT	Loan amount applied for

AMT_ANNUITY	Monthly installment amount to be paid
AMT_GOODS_PRICE	Price of the goods the loan is meant to buy
NAME_TYPE_SUITE	Who accompanied the client to the application (e.g., spouse, friend, etc.)
NAME_INCOME_TYPE	Type of applicant's income (e.g., working, pensioner, business)
NAME_EDUCATION_TYPE	Education level of the applicant
NAME_FAMILY_STATUS	Marital status
NAME_HOUSING_TYPE	Applicant's housing situation (e.g., own apartment, rented, etc.)
DAYS_BIRTH	Age of the applicant in days (negative number; convert to positive years)
DAYS_EMPLOYED	Number of days employed (can have large positive value for unemployed)
FLAG_MOBIL	Mobile phone provided flag
FLAG_WORK_PHONE	Work phone provided flag
FLAG_PHONE	Home phone provided flag
FLAG_EMAIL	Email provided flag
CNT_FAM_MEMBERS	Number of family members declared by the applicant
EXT_SOURCE_1/2/3	External risk scores from other sources (normalized values between 0 and 1)
DAYS_REGISTRATION	Days since registration with company
DAYS_ID_PUBLISH	Days since ID was changed

Key concepts:-

In a **data engineering** or **data warehouse** context, a **fact table** is:-
A central table that contains **measurable, quantitative data** (facts) about business processes, and typically references **dimension tables** via foreign keys

`application_train.csv` is the **fact table** because:

- Each row represents a **loan application** (a business event).

- It contains **measurable values** like:
    - **`AMT_CREDIT`**, **`AMT_INCOME_TOTAL`**, **`DAYS_BIRTH`**, etc.
- It can be **joined** with other tables like:
    - **`bureau.csv`** (past loans → dimension/fact)
    - **`previous_application.csv`** (previous apps → dimension/fact)
    - **`credit_card_balance.csv`** (monthly credit card behavior)

3.Bureau (df_bureau)

This table gives a comprehensive **credit history snapshot** for each client from external credit bureaus (like Experian, Equifax, etc.), at the time they applied for a loan with Home Credit.
This dataset includes **external credit history information** for each customer (SK_ID_CURR) from credit bureaus. It's used to understand the applicant's creditworthiness based on their **past and current loans** from other institutions.

Column Name	**Description**
SK_ID_CURR	ID of the current loan applicant (matches with application_train / application_test)
SK_BUREAU_ID	Unique ID for a reported credit line from the credit bureau
CREDIT_ACTIVE	Status of the credit line (Active / Closed / Sold)
CREDIT_CURRENCY	Currency used for the credit (mostly recoded)
DAYS_CREDIT	How many days before the loan application this credit was opened (e.g., -500 = 500 days ago)
CREDIT_DAY_OVERDUE	Number of days this credit was past due at time of application
DAYS_CREDIT_ENDDATE	Remaining duration of the credit (in days) at the time of application
DAYS_ENDDATE_FACT	Days since this credit actually ended at application time (only for closed credits)
AMT_CREDIT_MAX_OVERDUE	Max amount that was ever overdue on this credit
CNT_CREDIT_PROLONG	Number of times the credit was extended or restructured

AMT_CREDIT_SUM	Total amount of the credit
AMT_CREDIT_SUM_DEBT	Current unpaid debt for that credit line
AMT_CREDIT_SUM_LIMIT	Credit limit, typically for credit card lines
AMT_CREDIT_SUM_OVERDUE	Current overdue amount (if any)
CREDIT_TYPE	Type of the credit product (e.g., Credit card, Consumer credit, Mortgage)
DAYS_CREDIT_UPDATE	How many days before application the bureau last updated this record
AMT_ANNUITY	Monthly payment obligation for this credit line (if known)

Conceptual Understanding

- A single SK_ID_CURR (customer) can have **multiple** SK_BUREAU_IDs, each representing a separate **external loan or credit**.
- This data gives **historical behavior**: Was the client usually on time? Did they miss payments? How much debt are they carrying?
- Fields like DAYS_CREDIT, DAYS_CREDIT_ENDDATE, and DAYS_ENDDATE_FACT are crucial to **temporal analysis** of credit activity.

How to Use It ?

- Join bureau.csv with bureau_balance.csv on SK_BUREAU_ID
- Then join with application_train on SK_ID_CURR
- This gives you a **timeline and status** of each customer's credit activity **before** applying for the loan

4. Bureau_balance (df_bureau_balance)

Column Name	**Description**

SK_BUREAU_ID	A unique ID for each credit line reported by external credit bureaus. Use this to join with the main bureau table (df_bureau).
MONTHS_BALANCE	The month relative to the application when the snapshot was taken. For example, -1 means one month before the application, -6 means 6 months ago.
STATUS	Status of the bureau loan during that month. Values:

- "0" → No DPD (customer paid on time)
- "1" to "5" → Indicates increasing ranges of Days Past Due (e.g. "1" = 1–30 DPD)
- "C" → Closed
- "X" → Unknown status

Conceptual Notes for bureau_balance

- Think of **bureau_balance** as monthly **"status snapshots"** of each credit/loan the customer had *before* applying for the current loan.
- These loans are reported by **external credit bureaus**.
- You will get **one row per loan per month**. So it's a **time-series-like** table in **long format**..

Example Interpretation

Say a person had an external credit card with SK_BUREAU_ID = 100123:

- MONTHS_BALANCE = -1, STATUS = 0 → One month before loan application, payment was on time.
- MONTHS_BALANCE = -2, STATUS = 2 → Two months ago, they were **31–60 days late**.
- MONTHS_BALANCE = -5, STATUS = C → The loan got **closed** 5 months before application.

Joining Logic

To analyze customer behavior:

- **Join `bureau_balance` to `bureau` on `SK_BUREAU_ID`**
- Then **join `bureau` to `application_train` or `application_test` on `SK_ID_CURR`**

5. credit_card_balance (df_credit_card_balance)

This dataset provides month-wise details of a customer's credit card usage, payments, and outstanding amounts for their previous loans.
It **shows the monthly usage and repayment behavior of credit cards** related to previous loans taken by customer

Column	Meaning	Real-world Example
`MONTHS_BALANCE`	Month number relative to current application. -1 = most recent month.	-3 means 3 months before applying for the new loan
`AMT_BALANCE`	Credit card balance for the month	₹20,000 remaining to pay
`AMT_CREDIT_LIMIT_ACTUAL`	Credit card limit	₹1,00,000 limit set by bank
`AMT_DRAWINGS_ATM_CURRENT`	Amount withdrawn via ATM this month	Withdrew ₹5,000 cash
`AMT_DRAWINGS_POS_CURRENT`	Purchases via swipe/card at stores (Point of Sale)	Bought groceries worth ₹2,000
`AMT_DRAWINGS_CURRENT`	Total of all drawings (ATM + POS + others)	₹5,000 + ₹2,000 = ₹7,000

`AMT_INST_MIN_REGULARITY`	Minimum payment due this month	₹3,000 minimum due
`AMT_PAYMENT_CURRENT`	Actual payment made	Paid ₹5,000 this month
`AMT_TOTAL_RECEIVABLE`	Total remaining amount expected from customer	₹18,000 left to pay
`CNT_DRAWINGS_*`	How many times the card was used	2 ATM withdrawals, 5 store purchases
`CNT_INSTALMENT_MATURE_CUM`	How many installments have been paid till now	10 EMI payments made
`NAME_CONTRACT_STATUS`	Status of the credit	'Active' = credit card still running
`SK_DPD`	Days payment was delayed that month	5 = 5 days late
`SK_DPD_DEF`	Days late, but ignores low-amount delays	Same as **`SK_DPD`** if serious enough

Loan & Credit Card Process (Simplified)

1. **The customer applies for a loan or credit card.**
2. If approved, a **credit limit** is set (`AMT_CREDIT_LIMIT_ACTUAL`).
3. Customers make **drawings (spending)** via POS (Point of Sale), ATM, or others.
4. Monthly, a **minimum installment** is due (`AMT_INST_MIN_REGULARITY`).
5. Customers make **payments** (`AMT_PAYMENT_CURRENT`, etc.).
6. Remaining balance becomes **receivable** (`AMT_TOTAL_RECEIVABLE`).
7. Any delays show up in **Days Past Due (SK_DPD)**.

`AMT_BALANCE`	**Principal amount** the customer currently owes. This is the actual unpaid amount (think: total debt remaining).
`AMT_TOTAL_RECEIVABLE`	**Total amount to be received**, including **interest**, **fees**, and **other charges** (e.g. late payment penalties). This is what the lender *expects to collect*.

You borrowed ₹10,000 and missed some payments.
Now:

- You still owe ₹10,000 in principal → **AMT_BALANCE = 10,000**
- But you've accumulated ₹1,000 in interest and fees → A**MT_TOTAL_RECEIVABLE = 11,000**

6. installments_payments (df_installments_payments)

When someone takes out a loan (like for a house, car, or mobile phone), they agree to repay the amount (plus interest) in installments (monthly payments). Here's how this is reflected in the data:

Column	Description	Real-world Example
SK_ID_CURR	ID of the loan in the current dataset.(**Client ID** or **Current Loan ID**)	Suppose this is your main loan being analyzed (e.g., home loan).
SK_ID_PREV	A loan ID for a **previous credit** linked to the current customer.	You may have taken a mobile EMI earlier; this tracks history.
NUM_INSTALMENT_VERSION	Tracks changes to the **installment**	You agreed to pay ₹5000/month, but later renegotiated to ₹4000 –

	plan version (0 = credit card).	this change creates a new version.
NUM_INSTALMENT_NUMBER	Number of the installment (1st, 2nd, 3rd…)	First EMI = 1, Second EMI = 2, etc.
DAYS_INSTALMENT	**Scheduled date** of installment (days before application).	If it's -120, the installment was due 120 days before the main loan application.
DAYS_ENTRY_PAYMENT	**Actual payment date** (in days before application).	If -125, the person paid **early**. If -90, it's **late**.
AMT_INSTALMENT	Expected payment amount (installment)	For example, ₹5000.
AMT_PAYMENT	Actual amount paid by customer	Could be ₹5000 (on-time), ₹0 (missed), ₹3000 (partial).

What we Can Derive

- Consistency in payments = financially responsible customer.
- Early or over-payments = low risk.
- Missed or partial payments = red flags.
- Visualize delays (DAYS_INSTALMENT - DAYS_ENTRY_PAYMENT).

How This Links to the Loan Process

Let's say a customer took a personal loan a year ago and is now applying for a new home loan.

1. SK_ID_PREV tracks their previous loans.
2. NUM_INSTALMENT_NUMBER, DAYS_INSTALMENT, and AMT_INSTALMENT record what they were supposed to pay and when.
3. DAYS_ENTRY_PAYMENT and AMT_PAYMENT show if they paid on time, early, late, or missed.
4. Late or missed payments (or overpaid ones) give insights into their creditworthiness.

7. previous_application (df_previous_application)

In the real world, when a client (customer) applies for a loan, that application—**even if rejected**—is recorded. This dataset tracks **all past applications** before the **current loan under analysis**.

The dataset contains information about **past loan applications** made by each customer (`SK_ID_CURR`) to Home Credit. Each row corresponds to a **past application** (approved, refused, canceled, etc.), and is uniquely identified by `SK_ID_PREV`

Column	Description	Why Important
`SK_ID_CURR`	ID of client in current loan	Used to join with main dataset
`SK_ID_PREV`	ID of previous loan	Key to track multiple applications
`NAME_CONTRACT_TYPE`	Type of loan (Cash, Consumer, Revolving)	Helps analyze behavior by product
`AMT_APPLICATION`	Loan amount applied for	Key financial info
`AMT_CREDIT`	Credit approved by HC	Compare application vs. granted
`AMT_DOWN_PAYMENT`	Down payment made	Affects risk analysis
`AMT_ANNUITY`	Installment amount	Ties to repayment capacity

`NAME_CASH_LOAN_PURPOSE`	Purpose of loan (repairs, education...)	Useful for grouping behavior
`NAME_CLIENT_TYPE`	First-time or returning client	Risk profiling
`NAME_CONTRACT_STATUS`	Final status (Approved, Refused)	Helps model behavior of successful/failed applicants
`DAYS_DECISION`	Days relative to current application when this was decided	Time-lag to current application
`CNT_PAYMENT`	Number of payment installments planned	Duration of past loan

Use:-

1. **Join** `previous_application` to main application data via `SK_ID_CURR`.
2. **Aggregate** useful metrics (e.g., average past credit amount, number of refused loans, etc.) for each customer.
3. **Drop** redundant/sparse columns post aggregation.

Timeline Flow for a Previous Application

1. You apply for ₹500,000 (**AMT_APPLICATION**) on a **Tuesday morning** (**WEEKDAY_APPR_PROCESS_START**, **HOUR_APPR_PROCESS_START**).
2. The bank approves ₹450,000 (**AMT_CREDIT**) with 10% down (**RATE_DOWN_PAYMENT**), to buy a laptop (**NAME_GOODS_CATEGORY**).
3. You select 12-month EMI (**CNT_PAYMENT**) with a ₹5,000 EMI (**AMT_ANNUITY**).
4. The first EMI was due 30 days later (**DAYS_FIRST_DUE**), and the loan ended a year later (**DAYS_TERMINATION**).
5. This application may have been **approved** or **refused** (**NAME_CONTRACT_STATUS**).