Credix Case: Data Discovery

Created by: Breno Ruy

Last updated: November 26, 2025

1. Executive Summary

The main goal is to build a scalable Cost of Risk engine. An initial profile of the raw data (assets and ratings) shows some cornercases. With the help of the Risk team, those points will be addressed to better prepare for this implementation.

2. Data Health & Standardization

1. "Time Travel" Anomaly

- **Technical Problem:** Some settled records have a settled_at timestamp that is *earlier* than the created_at timestamp.
- **Business Translation:** Data suggests loans were paid back before they technically existed in the system (“Timestamp of the asset creation”). This breaks any calculation related to "Time to Repayment" (resulting in negative days).
- **Example:** A loan appears to be **Created on Nov 11, 2025**, but **Settled on Jan 16, 2025** (10 months earlier).
- **Recommendation:** We assume created_at represents "Database Ingestion/Migration" timestamp, not the true "Loan Origination Date." We should rely on due_date and settled_at for financial logic and ignore created_at for duration metrics.
- **Decision:** (NOT A PROBLEM) They will update dataset with correct timestamps

2. Status Standardization (Repaid)

- **Technical Problem:** Raw data shows a status value Repaid which is not documented in the provided business rules (only Settled and Canceled were specified).
- **Business Translation:** We risk under-reporting our success metrics. If a team asks for "Settled Loans," and we exclude the "Repaid" ones, we are presenting an incomplete picture of the portfolio's health.
- **Example:** Some rows have the status Repaid but look very much like the Settled loans (balance paid, settlement date present).
- **Recommendation:** Map both Repaid and Settled to a single standardized status "Settled" in the Staging Layer to ensure consistency downstream.
- **Decision:** The same, just standardize everything to Settled

3. Missing Identity (*asset_id*)

- **Technical Problem:** Raw assets.csv file did not contain the asset_id column mentioned in the spec.
- **Business Translation:** Without a unique identifier for every loan, we risk duplicate counting and cannot audit specific transactions.
- **Example:** The schema expected a UUID (universal unique identifier) column, but it was physically missing from the file.
- **Recommendation:** To generate a Surrogate Key (hash based on Buyer ID + Amount + Timestamp). This ensures every loan is uniquely identifiable and traceable without blocking the pipeline.
- **Decision:** We may take the recommended approach

4. "Unrated" Policy

- **Technical Problem:** Active buyers in the Assets table do not have a corresponding record in the Ratings table.
- **Business Translation:** We have active risk exposure to clients we don't know. The current logic defaults these to "Rating F" (40% Provision), which assumes the worst-case scenario (conservative approach).
- **Example:** Buyer Rreno Buy has a $10k active loan but no rating row. Currently, we book $4k (40%) as Expected Loss.
- **Recommendation:** If we want to remain conservative, keep them as Rating F. However, for a more realistic view, perhaps we should create an "Unknown" category (perhaps mapped to the portfolio average risk or other dynamic parameter) to distinguish "Bad Payers" from simple "New Clients."
- **Decision:** We should not take into consideration when calculating the Cost of Risk, this will drag us down because of conservative F grade for new clients/unknown rating (just remove those unknown risk clients assets

5. Dynamic vs. Static Risk Definition

- **Technical Problem:** Should we join Assets to Ratings based purely on tax_id (Current View) or tax_id + created_at (Point-in-Time View)?
- **Business Translation:** Does the "Cost of Risk" metric represent the quality of the loan when we sold it or the risk we hold right now?
- **Example:** A buyer was Rating A (1%) when they took the loan, but is Rating D (20%) today.
- **Recommendation:** Current View. This ensures that if a client's health deteriorates today, our provision/reserves update immediately to reflect the new reality.
- **Decision:** Both history + present (deviation might be taken into account, they might use it in a machine learning model)

6. Architectural Scope (Snapshot vs. Trends)

- **Technical Problem:** Should the final data model be a simple Accumulating Snapshot (Current State) or a Periodic Snapshot Fact (Historical State)?
- **Business Translation:** Do you need your risk today or the risk evolution within a time window?

- **Example:** Do we expect questions such as "Did our Cost of Risk go up or down compared to last month?"
- **Recommendation:** To build a Monthly Periodic Snapshot is a good generic approach. This allows us to track the evolution of risk over time as well as an updated picture of the current risk. If ratings change often we may shorten the time window for the snapshot for more sensibility to changes.
- **Decision:** Amount measured at the beginning of the month and at the end of the month, update of cost of risk then, they follow the metric in smaller chunks instead of a continuous line

3. Additional comments

-