**🧩 Data Model Rationale**

Your demo involves **6 core tables**, each representing a business domain component common in NBFC lending:

**1. loan\_accounts**

* Captures master data of each loan
* Fields: loan\_id, customer\_id, product\_type (e.g., 2W, 4W, CD), start\_date, end\_date, status
* Primary key: loan\_id

💡 This is your central table — almost all other tables reference it.

**2. repayment\_schedule**

* Holds EMI schedule and payment history
* Fields: loan\_id, due\_date, amount\_due, amount\_paid, payment\_date

💡 Used to calculate repayment gaps and derive DPD (Days Past Due).

**3. collections**

* Tracks field collection activities
* Fields: agent\_id, customer\_id, visit\_date, amount\_collected, remarks

💡 Used for operational tracking and analyzing recovery efficiency.

**4. disbursal\_data**

* Captures loan disbursal volume by type/branch/date
* Fields: loan\_type, disbursal\_date, disbursal\_amount

💡 Used to show volume trends, productivity, and drop in disbursals.

**5. interest\_rates**

* Historical interest rates per loan product
* Fields: loan\_type, rate, revised\_date

💡 Supports IRR calculations, policy analysis, or select AI queries.

**6. delinquency\_metrics**

* Contains calculated delinquency indicators per loan
* Fields: loan\_id, dpd, bucket, npa\_flag

💡 This is the **derived table** that feeds Select AI with delinquency classification.

**📊 How Delinquency Metrics Are Derived**

Let’s break it down:

**🔢 1. DPD (Days Past Due)**

Calculated as:

DPD = Current Date - Due Date of Last Unpaid Installment

**Rules:**

* If amount\_due > amount\_paid → installment unpaid → DPD applies
* If fully paid → DPD = 0

**🗂️ 2. Bucket Assignment**

Based on DPD, loans are grouped into **buckets**:

| **DPD Range** | **Bucket** |
| --- | --- |
| 1–30 | 30+ |
| 31–60 | 60+ |
| 61–90 | 90+ |
| 91–180 | 180+ |
| >180 | 180++ |
| 0 or null | Current |

CASE

WHEN dpd BETWEEN 1 AND 30 THEN '30+'

WHEN dpd BETWEEN 31 AND 60 THEN '60+'

WHEN dpd BETWEEN 61 AND 90 THEN '90+'

WHEN dpd BETWEEN 91 AND 180 THEN '180+'

WHEN dpd > 180 THEN '180++'

ELSE 'Current'

END AS bucket

**🚩 3. NPA Flag**

Flagged as Y or N:

* Y → NPA (Non-Performing Asset), usually if DPD > 90 or more than 3 missed EMIs
* Based on regulatory logic or business rules

Example logic:

CASE WHEN dpd > 90 THEN 'Y' ELSE 'N' END AS npa\_flag

**⚙️ How to Derive delinquency\_metrics Table from Raw Data**

Here’s a simplified SQL example:

INSERT INTO delinquency\_metrics (loan\_id, dpd, bucket, npa\_flag)

SELECT

r.loan\_id,

CASE

WHEN r.amount\_paid < r.amount\_due THEN TRUNC(SYSDATE) - r.due\_date

ELSE 0

END AS dpd,

CASE

WHEN r.amount\_paid < r.amount\_due AND TRUNC(SYSDATE) - r.due\_date BETWEEN 1 AND 30 THEN '30+'

WHEN r.amount\_paid < r.amount\_due AND TRUNC(SYSDATE) - r.due\_date BETWEEN 31 AND 60 THEN '60+'

WHEN r.amount\_paid < r.amount\_due AND TRUNC(SYSDATE) - r.due\_date BETWEEN 61 AND 90 THEN '90+'

WHEN r.amount\_paid < r.amount\_due AND TRUNC(SYSDATE) - r.due\_date > 90 THEN '180+'

ELSE 'Current'

END AS bucket,

CASE

WHEN r.amount\_paid < r.amount\_due AND TRUNC(SYSDATE) - r.due\_date > 90 THEN 'Y'

ELSE 'N'

END AS npa\_flag

FROM repayment\_schedule r

JOIN (

SELECT loan\_id, MAX(due\_date) AS last\_due

FROM repayment\_schedule

GROUP BY loan\_id

) last\_due\_info ON r.loan\_id = last\_due\_info.loan\_id AND r.due\_date = last\_due\_info.last\_due;

**🎯 Summary**

* **loan\_accounts** is the core
* **repayment\_schedule** feeds DPD calculations
* **delinquency\_metrics** is a derived summary for analytics
* **collections** helps correlate efforts vs. recovery
* **disbursal\_data** helps assess business impact

Would you like:

* A **materialized view** or procedure to auto-refresh delinquency\_metrics daily?
* Sample visualizations using **APEX** or **Oracle Data Studio**?

Let me know how far you want to extend the demo.