**Data Generation Process for Fintech Database**

**1. Overview**

Synthetic data is crucial for testing the scalability, performance, and integrity of the fintech platform. This document outlines the data generation process to ensure reproducibility and clarity.

**2. Schema Explanation**

The fintech platform consists of the following key tables:

* **Users**: Stores user details.
* **Vendors**: Represents merchants providing services.
* **Accounts**: Maintains user and vendor account balances.
* **FinancialTransactions**: Logs all transactions between users and vendors.
* **MarketData**: Stores market-related pricing data.
* **Audits**: Tracks modifications across tables.
* **Disputes**: Captures transaction disputes raised by users.

**3. Data Generation Strategy**

**3.1 Randomization Methods**

* **User Data**: Generated using Faker (names, emails, phone numbers).
* **Transaction Data**: Random user-vendor pairs with varying amounts and transaction types.
* **Timestamps**: Randomized within realistic timeframes.

**3.2 Constraints Handling**

* **Unique Keys**: Ensured for UserID, VendorID, and primary keys.
* **Foreign Key Integrity**: All foreign keys reference valid parent records.
* **Valid Statuses**: ENUM fields (e.g., transaction status, dispute resolution) randomly assigned valid values.

**3.3 Volume & Distribution**

* **Users Table**: 100,000 users.
* **Vendors Table**: 5,000 vendors.
* **FinancialTransactions**: 10 million transactions.
* **MarketData**: 1 million records.
* **Audits & Disputes**: 100,000 audit logs, 50,000 disputes.

**4. SQL Scripts & Tools**

**Data Insertion Approach:**

* **SQL Scripts**: Bulk INSERT statements for synthetic data.
* **Python (Faker & Pandas)**: Used for generating realistic user and transaction data.
* **Stored Procedures**: Predefined procedures for controlled data generation.

**5. Execution Steps**

1. **Run Schema Scripts**: Execute create\_tables.sql to set up the database.
2. **Generate Data**:
   * Use populate\_users.sql for users.
   * Use populate\_vendors.sql for vendors.
   * Use Python script (generate\_transactions.py) for bulk transactions.
3. **Verify Data Integrity**:
   * Ensure all IDs exist in parent tables.
   * Check constraints using CHECK and FOREIGN KEY validations.

**6. Validation & Performance Testing**

**6.1 Data Verification**

* Run SELECT COUNT(\*) on each table to confirm expected row counts.
* Validate random samples with SELECT \* FROM table\_name LIMIT 10.

**6.2 Performance Testing**

* Execute EXPLAIN ANALYZE on critical queries.
* Benchmark execution times before and after indexing.
* Record CPU and memory usage for large-scale queries.

This document ensures a clear, structured approach to synthetic data generation, making it easy to reproduce the process across different environments.