**Spaghetti Database Description**

The spaghetti database comprises of the following tables.

**ATM**

The ATM table is the central table containing details about each ATM, including its configuration, status, and location. Each ATM is uniquely identified by the ATM\_ID and has several attributes like TIME\_ZONE, ENTITY\_ID, CURRENT\_MODE, and ATM\_TYPE\_ID that describe its operational state and settings.

**CASH\_HANDLER**

The CASH\_HANDLER table stores information about the cash operations of each ATM, including the amount of notes dispensed and rejected. It links to the ATM table via the ATM\_ID to ensure that cash handling details are associated with the correct ATM.

**BNA (Bill Note Acceptor)**

The BNA table contains details about the bill note acceptor for each ATM, recording notes refunded, rejected, and encashed. This table is linked to the ATM table using ATM\_ID to track the cash handling efficiency and issues specific to each ATM.

**ATM\_DEVICE**

The ATM\_DEVICE table lists the various devices associated with each ATM, identified by the composite key of ATM\_ID and DEVICE\_ID. This table helps in managing and monitoring the different hardware components installed in each ATM.

**LAST\_TRAN\_STATUS**

The LAST\_TRAN\_STATUS table records the last transaction status for each ATM, providing details such as notes dispensed in different categories. It connects to the ATM table via ATM\_ID and references the LAST\_TRAN\_STATUS\_TYPE table to classify the transaction status.

**LAST\_TRAN\_STATUS\_TYPE**

The LAST\_TRAN\_STATUS\_TYPE table defines the different types of transaction statuses that an ATM can have. Each status type is uniquely identified by Last\_Tran\_Sts\_Type\_Id and described in the Last\_Tran\_Status\_Desc column.

**CURRENCY**

The CURRENCY table holds information about the different currencies handled by the ATMs, including their ISO codes and descriptions. It links to the CASH\_HANDLER table to provide context for the currency amounts dispensed or rejected by each ATM.

**INCOMING\_TRANSACTION**

The INCOMING\_TRANSACTION table records incoming transactions processed by the ATMs, including details like transaction serial number (TSN), amount, and PAN. It is linked to the ATM table via ATM\_ID to associate each transaction with the correct ATM.

**DECLINED\_TRAN\_LOG**

The DECLINED\_TRAN\_LOG table logs details of transactions that were declined by the ATM. This includes the amount, TSN, and the transaction code, and it links back to the ATM table via ATM\_ID to identify which ATM declined the transaction.

**ATM\_PROTOCOL**

The ATM\_PROTOCOL table describes the communication protocols used by the ATMs, identified by ATM\_PROTOCOL\_ID. It contains information such as protocol description and configuration paths, linking back to the ATM table to specify which protocols are used by each ATM.

**ATM\_TYPE**

The ATM\_TYPE table categorizes the types of ATMs, with each type being uniquely identified by ATM\_TYPE\_ID. It includes a description of each type and is linked to the ATM table to classify each ATM according to its type.

**TRAN\_SET**

The TRAN\_SET table stores sets of transactions, identified by SET\_ID, and linked to the PRE\_PROCESSING table. It also contains an ENTITY\_ID to relate it to a specific entity or organization.

**PRE\_PROCESSING**

The PRE\_PROCESSING table contains preprocessing details for transactions, uniquely identified by PRE\_PROCESSING\_ID. It links to TRAN\_SET and TRANSACTION\_CODE to describe preprocessing steps and related transaction codes.

**TRANSACTION\_CODE**

The TRANSACTION\_CODE table defines various transaction codes and their descriptions, identified by TRANSACTION\_CODE\_ID. It is linked to the PRE\_PROCESSING table to provide context for each preprocessing step.

**CONT\_RECEIPT**

The CONT\_RECEIPT table holds patterns and descriptions for receipts, identified by CONT\_RECEIPT\_ID. It links to the TRAN\_SET table via SET\_ID to specify the receipt content related to each transaction set.

### Assumptions in creation of tables

#### Normalization

The database schema is highly normalized, with each table focusing on specific aspects of ATM operations. This design ensures minimal data redundancy and optimal data integrity.

#### Foreign Keys

All relationships are maintained using foreign keys that link primary keys of related tables. This approach enforces referential integrity and ensures that all related data is consistent and properly connected.

#### Composite Keys

Some tables, such as ATM\_DEVICE, utilize composite keys to uniquely identify records by combining multiple columns. In the ATM\_DEVICE table, the composite key is a combination of ATM\_ID and DEVICE\_ID. This composite key uniquely identifies each record in the ATM\_DEVICE table, ensuring that each device associated with a particular ATM can be distinctly identified. This ensures precise identification of records in scenarios where a single column is not sufficient.

#### Entity-Relationship Integrity

The relationships between tables, such as those between ATM, CASH\_HANDLER, and BNA, are assumed to enforce referential integrity. This guarantees that all foreign key references are valid and consistent.

#### Operational Focus

The database schema is designed to track various aspects of ATM operations, including cash handling, transaction status, device management, and declined transactions. This comprehensive focus supports detailed monitoring and management of ATM functions.

#### Scalability

The structure of the database is assumed to be scalable, allowing for the addition of new ATMs, devices, and transaction records without compromising performance or data integrity. This ensures the system can grow and adapt to increasing demands.

### Summary of the ATM Database Schema

The ATM database schema is highly normalized, with a focus on different aspects of ATM operations, cash handling, and transaction processing. The ATM table serves as the central hub, connecting to various other tables that manage cash handler details, transaction logs, device management, and configuration protocols. Each table is designed to capture specific details, ensuring data integrity and scalability.

The relationships between these tables are defined using foreign keys, ensuring referential integrity and consistency. This structure supports comprehensive tracking and management of ATM operations, from handling multiple currencies to logging incoming and declined transactions. The schema is designed to be scalable, allowing for easy addition of new ATMs, devices, and transaction records, making it a robust framework for managing ATM networks.

**Process:**

* **Data Extraction from OLTP Database**:

SQL queries are executed to extract data from various OLTP tables related to ATM transactions and operations.

* **Data Transformation and Cleaning**:

Missing values are handled and data is cleaned by applying transformations such as filtering out irrelevant columns or rows.

* **Data Type Conversion**:

Data types are converted as needed to ensure consistency and compatibility with the target schema, such as converting strings to datetime objects.

* **Creation of Dimension Tables**:

Dimension tables are created in SQL Server to store descriptive attributes about entities like ATMs, currencies, and transaction types.

* **Population of Dimension Tables**:

Extracted and transformed data is loaded into the dimension tables, ensuring that each table is populated with the relevant attributes.

* **Creation of Fact Tables**:

Fact tables are created to store transactional data and measures such as declined transactions and notes dispensed.

* **Population of Fact Tables**:

Transactional data is loaded into the fact tables, associating each transaction with the appropriate dimensions and measures.

* **Creation of Star Schema**:

Primary and foreign key relationships are established between dimension and fact tables, forming a star schema that facilitates efficient querying and analysis.

* **Data Mapping to Star Schema**:

Data from the staging area is mapped to the appropriate dimension and fact tables in the star schema, ensuring accurate representation of the underlying business entities and transactions.

* **Power BI Dashboard Creation**:

Visualizations are created in Power BI based on the star schema to provide insights into ATM transactions and operations.

* **Dashboard Monitoring and Update**:

The dashboard is monitored for changes in metrics, and it is automatically updated with each execution of the pipeline to reflect the latest data.

**Design Diagram of Pipeline**

**OLTP Database (ATM)**

**Dashboard Monitoring and Automatic Update**

**Power BI Dashboard Creation**

**Data Extraction**

**Staging Area**

**Data** **Transformation** **and Cleaning**

**Fact Tables Creation**

**Dimension Tables Creation**

**Star Schema Creation and Data Mapping to Star Schema**

Population of Fact Tables

Population of Dimension Tables

Data Type Conversion