Entity Relationship Diagram (ERD) – MoMo SMS Analytics

Core Entities and Attributes

1. Users

- **user_id (PK)** – INT, auto-increment
- phone_number – VARCHAR(20), UNIQUE
- full_name – VARCHAR(100)
- user_type – ENUM('SENDER','RECEIVER','AGENT','MERCHANT')
- created_at – DATETIME

2. Transactions

- **transaction_id (PK)** – INT, auto-increment
- transaction_reference – VARCHAR(50), UNIQUE
- sender_id (FK → Users.user_id)
- receiver_id (FK → Users.user_id)
- category_id (FK → Transaction_Categories.category_id)
- amount – DECIMAL(10,2)
- currency – VARCHAR(5)
- transaction_date – DATETIME
- status – ENUM('SUCCESS','FAILED','PENDING')

3. Transaction_Categories

- **category_id (PK)** – INT, auto-increment
- category_name – VARCHAR(50)
- description – TEXT

4. System_Logs

- **log_id (PK)** – INT, auto-increment
- transaction_id (FK → Transactions.transaction_id)
- log_message – VARCHAR(255)
- log_level – ENUM('INFO','WARNING','ERROR')
- created_at – DATETIME

5. Transaction_Tags (Junction Table)

- **transaction_id (FK → Transactions.transaction_id)**
- **tag_id (FK → Tags.tag_id)**

6. Tags

- **tag_id (PK)** – INT, auto-increment
- tag_name – VARCHAR(50)

Relationships & Cardinality

- Users **1:M** Transactions (sender)
- Users **1:M** Transactions (receiver)
- Transaction_Categories **1:M** Transactions
- Transactions **1:M** System_Logs
- Transactions **M:N** Tags (resolved via Transaction_Tags)

ERD Diagram (Draw.io / Lucidchart Guide)

1. Create rectangles for each entity above.
2. Place PKs at the top, FKs beneath.
3. Use crow's foot notation for 1:M relationships.
4. Connect Transactions ↔ Tags using Transaction_Tags.
5. Label relationships clearly (e.g., "initiates", "belongs to").

Design Justification (≈230 words)

This ERD was designed to support efficient storage, querying, and future scalability of MoMo SMS transaction data. The **Transactions** entity serves as the central table, capturing all core transaction details such as amount, date, status, and category. Separating **Users** into their own entity avoids data redundancy and supports reuse of sender and receiver information across multiple transactions, ensuring normalization up to Third Normal Form (3NF).

The **Transaction_Categories** table abstracts payment and transfer types (e.g., P2P transfer, bill payment), allowing easy extension as new transaction types are introduced without altering the Transactions table structure. **System_Logs** enables traceability and debugging of SMS processing and transaction lifecycle events, which is essential for auditability and system monitoring.

A many-to-many relationship between **Transactions** and **Tags** is resolved using the **Transaction_Tags** junction table. This design allows flexible tagging (e.g., "salary", "rent", "utilities") for analytics and reporting without complicating the core transaction schema. Referential integrity is enforced through foreign keys, ensuring consistent and reliable data.

Overall, the design balances normalization with performance, supports analytical queries, and aligns well with both relational database storage and JSON-based API serialization, making it suitable for enterprise-scale MoMo SMS analytics systems.