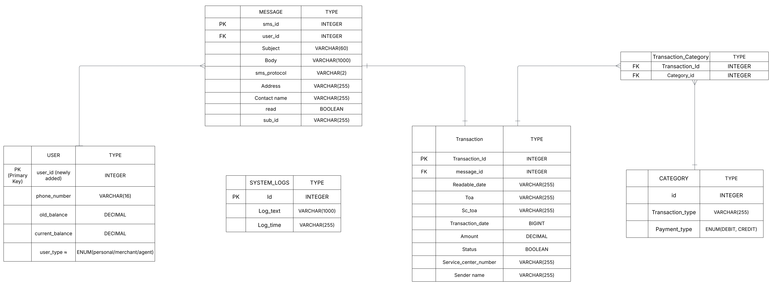
Database Design Document

ERD



REASONING

In designing this schema, we focused on balancing **data integrity, flexibility, and scalability** for processing MoMo SMS data. We began by separating raw messages from parsed transactions. The **MESSAGE** table preserves the full SMS content (subject, body, protocol, address, etc.), ensuring that we always retain the original record for verification or reprocessing. Each message links to a **USER**, allowing us to track phone numbers, balances, and user roles (personal, merchant, agent) while isolating sensitive information.

We then created the **Transaction** table to store structured details extracted from SMS messages, such as transaction date, amount, status, sender, and service center. By referencing the originating message through a foreign key, we maintain traceability between raw input and parsed data. Balances are managed in the USER table to support reconciliation and fraud detection.

For categorization, we introduced a many-to-many relationship via **Transaction_Category** and **CATEGORY** tables. This design gives us the flexibility to tag a transaction with multiple categories (e.g. CREDIT and DEBIT). This makes future reporting and analysis use cases easier without altering the schema.

Finally, the **SYSTEM_LOGS** table provides observability by recording parser actions and ETL events. This supports debugging, compliance, and auditability. Overall, our design separates concerns: raw ingestion, structured transactions, categorization, and monitoring. This ensures efficiency in queries, maintains integrity across relationships, and positions the system for future scaling as transaction volumes grow.

Data Dictionary

USER
- user_id *(PK)* – Unique identifier for each user.
- phone_number – User's registered mobile number.
- old_balance – Previous balance before last transaction.
- current_balance – Current balance after last transaction.
- user_type – ENUM: personal / merchant / agent.

MESSAGE
- sms_id *(PK)* – Unique SMS record.
- user_id *(FK → USER)* – User who received/sent the SMS.
- subject – SMS subject (optional).
- body – Full SMS text content.
- sms_protocol – SMS protocol identifier.
- address – Sender address (e.g., "M-Money").
- contact_name – Name if available in phone contacts.
- read – Boolean, true if SMS was read.
- sub_id – SIM subscription identifier.

TRANSACTION
- transaction_id *(PK)* – Unique transaction.
- message_id *(FK → MESSAGE)* – Source SMS.
- readable_date – Human-readable timestamp.
- toa, sc_toa – SMS technical metadata.
- transaction_date – Epoch time for sorting.
- amount – Transaction amount (DECIMAL).
- status – Boolean (success/failure).
- service_center_number – SMS service center number.
- sender_name – Sender label.

CATEGORY
- id *(PK)* – Unique category.
- transaction_type – Category name (e.g., P2P_SEND).
- payment_type – ENUM: DEBIT / CREDIT.

TRANSACTION_CATEGORY
- transaction_id *(FK)* – Links to TRANSACTION.
- category_id *(FK)* – Links to CATEGORY.

SYSTEM_LOGS
- id *(PK)* – Unique log entry.
- log_text – Message about system/parsing actions.
- log_time – Timestamp of log entry.

Sample Queries

a) Get all transactions for a user by phone number

```
SELECT t.transaction_id, t.amount, t.status, t.readable_date
FROM transaction AS t
JOIN user AS u ON t.user_id = u.user_id
WHERE u.phone_number = '+250788110381';
```

b) Calculate user's total credits and debits

```
SELECT c.transaction_type AS flow, SUM(t.amount) AS total_amount
FROM transaction AS t
JOIN transaction_category AS tc ON tc.transaction_id = t.transaction_id
JOIN category AS c ON c.category_id = tc.category_id
JOIN user AS u ON u.user_id = t.user_id
WHERE u.user_id = 42
  AND t.status = 'COMPLETED'
GROUP BY c.transaction_type;
```

c) Audit: Find failed transactions

```
SELECT t.transaction_id, t.amount, u.phone_number
FROM transaction AS t
JOIN user AS u ON u.user_id = t.user_id
WHERE t.status = 'FAILED';
```

```
mysql> -- find failed transactions
Query OK, 0 rows affected (0.000 sec)

mysql> SELECT t.transaction_id, t.amount, u.phone_number
    -> FROM transaction AS t
    -> JOIN user AS u ON u.user_id = t.user_id
    -> WHERE t.status = 'FAILED';
ERROR 1046 (3D000): No database selected
mysql> USE momo_data_visualizer;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SELECT t.transaction_id, t.amount, u.phone_number
    -> FROM transaction AS t
    -> JOIN user AS u ON u.user_id = t.user_id
    -> WHERE t.status = 'FAILED';
+----------------+--------+--------------+
| transaction_id | amount | phone_number |
+----------------+--------+--------------+
|              3 | 500.00 | 0781110003   |
+----------------+--------+--------------+
1 row in set (0.000 sec)

mysql> -- get all transactions for a user by phone
Query OK, 0 rows affected (0.000 sec)

mysql> SELECT t.transaction_id, t.amount, t.status, t.readable_date
    -> FROM transaction AS t
    -> JOIN user AS u ON t.user_id = u.user_id
    -> WHERE u.phone_number = '0781110003';
+----------------+--------+--------+---------------+
| transaction_id | amount | status | readable_date |
+----------------+--------+--------+---------------+
|              3 | 500.00 | FAILED | 2024-05-12    |
+----------------+--------+--------+---------------+
1 row in set (0.001 sec)

mysql>
```

Sample Queries - CRUD ops

```
Database changed
mysql> SHOW TABLES;
+-----------------------------------+
| Tables_in_momo_data_visualizer    |
+-----------------------------------+
| category                          |
| message                           |
| system_logs                       |
| transaction_category              |
| transaction_record                |
| user                              |
+-----------------------------------+
6 rows in set (0.001 sec)

mysql> SELECT * FROM user;
+---------+--------------+-------------+-----------------+-----------+
| user_id | phone_number | old_balance | current_balance | user_type |
+---------+--------------+-------------+-----------------+-----------+
|       1 | 0780000001   |    12000.00 |        41000.00 | personal  |
|       2 | 0780000006   |    12000.00 |        11000.00 | personal  |
|       3 | 0780000005   |    11000.00 |        11000.00 | personal  |
|       4 | 0780000003   |    13000.00 |        21000.00 | personal  |
|       5 | 0780000002   |    16000.00 |        31000.00 | personal  |
+---------+--------------+-------------+-----------------+-----------+
5 rows in set (0.006 sec)

mysql> UPDATE user SET user_type TO "agent" WHERE user_id = 1;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds
ax to use near 'TO "agent" WHERE user_id = 1' at line 1
mysql> UPDATE user SET user_type = "agent" WHERE user_id = 1;
Query OK, 1 row affected (0.004 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> DELETE FROM user WHERE user_id = 5;
Query OK, 1 row affected (0.046 sec)

mysql>
```

Unique Rules for Accuracy & Security

Unique constraints:
- USER.phone_number must be unique (no duplicates).
- TRANSACTION.transaction_id globally unique (no double imports)

Referential integrity:
- Foreign keys with ON DELETE CASCADE to avoid orphan records.