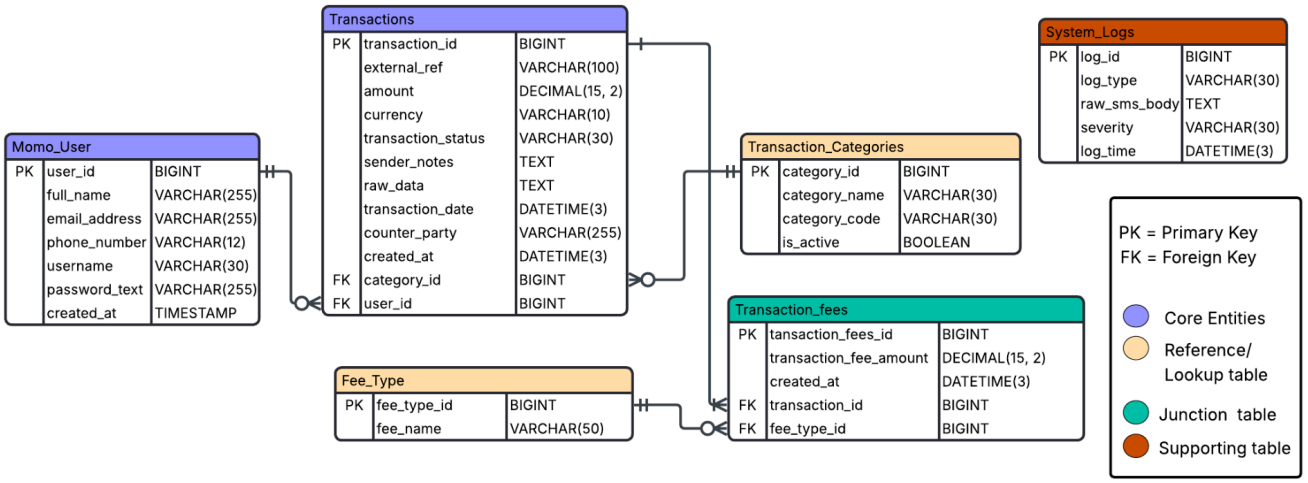
ERD Documentation



When we designed the entity relationship diagram, we first identified the core entities by analyzing the raw XML data that was given to us. This data included users and transactions, along with their specific attributes. From there, we decided to have two core entities and two lookup tables. We also added one junction table to resolve a many-to-many relationship and one independent table for system logs. This system log table helps us look for errors before transforming XML data into JSON or extracting transactions into the database.

Our core entities are Users and Transactions. For lookup tables, we have Transaction Categories and Fee Types. We found a many-to-many relationship between transactions and fees, because one transaction can have different fee types like tax, agent fees, or service fees, while one fee type can apply to many transactions. To solve this, we used a Transaction Fees junction table to track the specific fee for a single transaction.

As Momo user attributes, we included passwords(text) and usernames for Users to support basic authentication. We will not make the passwords hashed, as we are asked to implement basic/weak authentication in the coming phases of the project. Transactions had many attributes in the xml file, but we have reduced it to the main ones to accommodate the business logic primarily. We have a record to keep the amount, date, status, counterparty (anyone involved in the transaction), and sender notes. We originally had a table for currency, but since we are focusing on Momo (Mobile Money), which is operating in Rwanda, we reduced that and put it directly in the Transactions table to be efficient without over-engineering our design. However, we kept it future-considerate (easy to migrate it and create new tables) so we can add different currencies later if necessary. We chose Big Integer

for all primary keys because we expect a lot of users and transactions, and we didn't want to be limited by a standard integer range.

Data dictionary with table and column descriptions

Transactions table: Main transaction records from SMS data

```
mysql> SELECT      COLUMN_NAME AS 'Field',      COLUMN_TYPE AS 'Type',      IS_NULLABLE AS 'Allow Null',      COLUMN_KEY
 AS 'Key',      COLUMN_COMMENT AS 'Description' FROM information_schema.COLUMNS  WHERE TABLE_NAME = 'Transactions'  AND
TABLE_SCHEMA = 'momo_sms_db';
+--------------------+---------------+------------+-----+-------------------------------------------------+
| Field              | Type          | Allow Null | Key | Description                                     |
+--------------------+---------------+------------+-----+-------------------------------------------------+
| amount             | decimal(15,2) | NO         |     | Transaction amount                              |
| category_id        | bigint        | NO         | MUL | Transaction category                            |
| counter_party      | varchar(255)  | YES        |     | Name of other party in transaction              |
| created_at         | datetime(3)   | NO         |     | Database insertion timestamp                    |
| currency           | varchar(10)   | NO         |     | Currency code (RWF, ETB, etc.)                  |
| external_ref       | varchar(100)  | NO         | MUL | External transaction reference from SMS         |
| raw_data           | text          | NO         |     | Original SMS message body                       |
| sender_notes       | text          | YES        |     | Optional transaction notes or message from sender |
| transaction_date   | datetime(3)   | NO         | MUL | When transaction actually occurred              |
| transaction_id     | bigint        | NO         | PRI | Unique transaction identifier                   |
| transaction_status | varchar(30)   | NO         | MUL | Transaction status                              |
| user_id            | bigint        | NO         | MUL | Mobile money user                               |
+--------------------+---------------+------------+-----+-------------------------------------------------+
12 rows in set (0.03 sec)

mysql>
```

Momo User Table: tores mobile money user information

```
mysql> SELECT      COLUMN_NAME AS 'Field',      COLUMN_TYPE AS 'Type',      IS_NULLABLE AS 'Allow Null',      COLUMN_KEY
 AS 'Key',      COLUMN_COMMENT AS 'Description' FROM information_schema.COLUMNS  WHERE TABLE_NAME = 'Momo_User'  AND TAB
LE_SCHEMA = 'momo_sms_db';
+---------------+-------------+------------+-----+-------------------------------------------------------+
| Field         | Type        | Allow Null | Key | Description                                           |
+---------------+-------------+------------+-----+-------------------------------------------------------+
| created_at    | timestamp   | NO         |     | Account creation timestamp                            |
| email_address | varchar(255)| YES        |     | Optional email address                                |
| full_name     | varchar(255)| NO         |     | User full name                                        |
| password_text | varchar(255)| NO         |     | Text password (not hashed) for basic authentication   |
| phone_number  | varchar(12) | NO         | UNI | Mobile money phone number                             |
| user_id       | bigint      | NO         | PRI | Unique user identifier                                |
| username      | varchar(30) | YES        | UNI | Optional username for login                           |
+---------------+-------------+------------+-----+-------------------------------------------------------+
7 rows in set (0.01 sec)

mysql>
```

Transaction Categories table: Reference table for transaction types

```
mysql> SELECT      COLUMN_NAME AS 'Field',      COLUMN_TYPE AS 'Type',      IS_NULLABLE AS 'Allow Null',      COLUMN_KEY
 AS 'Key',      COLUMN_COMMENT AS 'Description' FROM information_schema.COLUMNS  WHERE TABLE_NAME = 'Transaction_Categor
ies'  AND TABLE_SCHEMA = 'momo_sms_db';
+---------------+-------------+------------+-----+-------------------------------------+
| Field         | Type        | Allow Null | Key | Description                         |
+---------------+-------------+------------+-----+-------------------------------------+
| category_code | varchar(30) | NO         | UNI | Internal code for application logic |
| category_id   | bigint      | NO         | PRI | Unique category identifier          |
| category_name | varchar(30) | NO         |     | Display name for category           |
| is_active     | tinyint(1)  | NO         |     | Whether category is currently active |
+---------------+-------------+------------+-----+-------------------------------------+
4 rows in set (0.01 sec)

mysql>
```

Fee Type table: Reference table for fee types

```
mysql> SELECT      COLUMN_NAME AS 'Field',      COLUMN_TYPE AS 'Type',      IS_NULLABLE AS 'Allow Null',      COLUMN_KEY
 AS 'Key',      COLUMN_COMMENT AS 'Description' FROM information_schema.COLUMNS  WHERE TABLE_NAME = 'Fee_Type'  AND TABL
E_SCHEMA = 'momo_sms_db';
+-------------+-------------+------------+-----+-------------------------------------------+
| Field       | Type        | Allow Null | Key | Description                               |
+-------------+-------------+------------+-----+-------------------------------------------+
| fee_name    | varchar(50) | NO         | MUL | Fee type name (e.g., Transaction Fee, Tax) |
| fee_type_id | bigint      | NO         | PRI | Unique fee type identifier                |
+-------------+-------------+------------+-----+-------------------------------------------+
2 rows in set (0.24 sec)

mysql>
```

Transaction fees table: Junction table resolving M:N relationship between transactions and fee types

```
mysql> SELECT      COLUMN_NAME AS 'Field',      COLUMN_TYPE AS 'Type',      IS_NULLABLE AS 'Allow Null',      COLUMN_KEY
 AS 'Key',      COLUMN_COMMENT AS 'Description' FROM information_schema.COLUMNS  WHERE TABLE_NAME = 'Transaction_fees'
AND TABLE_SCHEMA = 'momo_sms_db';
+------------------------+---------------+------------+-----+---------------------------------+
| Field                  | Type          | Allow Null | Key | Description                     |
+------------------------+---------------+------------+-----+---------------------------------+
| created_at             | datetime(3)   | NO         |     | Fee record creation timestamp   |
| fee_type_id            | bigint        | NO         | MUL | Type of fee applied             |
| transaction_fee_amount | decimal(15,2) | NO         |     | Actual fee amount charged       |
| transaction_fees_id    | bigint        | NO         | PRI | Unique fee record identifier    |
| transaction_id         | bigint        | NO         | MUL | Reference to parent transaction |
+------------------------+---------------+------------+-----+---------------------------------+
5 rows in set (0.01 sec)

mysql>
```

System logs table: System processing logs for debugging and monitoring

```
mysql> SELECT      COLUMN_NAME AS 'Field',      COLUMN_TYPE AS 'Type',      IS_NULLABLE AS 'Allow Null',      COLUMN_KEY
 AS 'Key',      COLUMN_COMMENT AS 'Description' FROM information_schema.COLUMNS  WHERE TABLE_NAME = 'System_Logs'  AND T
ABLE_SCHEMA = 'momo_sms_db';
+--------------+-------------+------------+-----+-------------------------------------------------+
| Field        | Type        | Allow Null | Key | Description                                     |
+--------------+-------------+------------+-----+-------------------------------------------------+
| log_id       | bigint      | NO         | PRI | Unique log entry identifier                     |
| log_time     | datetime(3) | NO         | MUL | When log entry was created                      |
| log_type     | varchar(30) | NO         | MUL | Type of log entry (PARSE_ERROR, DB_ERROR, etc.) |
| raw_sms_body | text        | YES        |     | Raw SMS content if parsing error occurred       |
| severity     | varchar(30) | NO         | MUL | Log severity level                              |
+--------------+-------------+------------+-----+-------------------------------------------------+
5 rows in set (0.00 sec)

mysql>
```

Sample queries (with screenshots)

```
INSERT INTO Transactions (external_ref, amount, currency, transaction_status,
sender_notes, raw_data, transaction_date, counter_party, category_id, user_id)
VALUES ('TEST001', 750.00, 'RWF', 'COMPLETED', 'Test transaction', 'Test SMS
body', NOW(), 'Test User', 1, 1);
```

```
mysql> INSERT INTO Transactions (external_ref, amount, currency, transaction_status, sender_notes, raw_data, transaction
_date, counter_party, category_id, user_id) VALUES ('TEST001', 750.00, 'RWF', 'COMPLETED', 'Test transaction', 'Test SMS
 body', NOW(), 'Test User', 1, 1);
Query OK, 1 row affected (0.03 sec)

mysql> SELECT * FROM Transactions WHERE external_ref = 'TEST001';
+----------------+--------------+--------+----------+--------------------+-----------------+--------------+-----------
-------------+--------------+-------------------------+-------------+---------+
| transaction_id | external_ref | amount | currency | transaction_status | sender_notes    | raw_data     | transactio
n_date        | counter_party | created_at              | category_id | user_id |
+----------------+--------------+--------+----------+--------------------+-----------------+--------------+-----------
-------------+--------------+-------------------------+-------------+---------+
|             10 | TEST001      | 750.00 | RWF      | COMPLETED          | Test transaction | Test SMS body | 2026-01-26
 01:35:56.000 | Test User     | 2026-01-26 01:35:56.936 |           1 |       1 |
|             11 | TEST001      | 750.00 | RWF      | COMPLETED          | Test transaction | Test SMS body | 2026-01-26
 01:37:10.000 | Test User     | 2026-01-26 01:37:10.209 |           1 |       1 |
+----------------+--------------+--------+----------+--------------------+-----------------+--------------+-----------
-------------+--------------+-------------------------+-------------+---------+
2 rows in set (0.00 sec)

mysql>
```

```
UPDATE Transactions
SET transaction_status = 'COMPLETED', sender_notes = 'Updated test
transaction'
WHERE external_ref = 'TEST001';
```

```
mysql> UPDATE Transactions  SET transaction_status = 'COMPLETED', sender_notes = 'Updated test transaction' WHERE extern
al_ref = 'TEST001';
Query OK, 2 rows affected (0.31 sec)
Rows matched: 2  Changed: 2  Warnings: 0

mysql> SELECT transaction_id, external_ref, transaction_status, sender_notes, FROM Transactions WHERE external_ref = 'TE
ST001';
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version
 for the right syntax to use near 'FROM Transactions WHERE external_ref = 'TEST001'' at line 1
mysql> SELECT transaction_id, external_ref, transaction_status, sender_notes FROM Transactions WHERE external_ref = 'TES
T001';
+----------------+--------------+--------------------+-------------------------+
| transaction_id | external_ref | transaction_status | sender_notes            |
+----------------+--------------+--------------------+-------------------------+
|             10 | TEST001      | COMPLETED          | Updated test transaction |
|             11 | TEST001      | COMPLETED          | Updated test transaction |
+----------------+--------------+--------------------+-------------------------+
2 rows in set (0.00 sec)

mysql>
```

```
DELETE FROM Transactions WHERE external_ref = 'TEST001';
```

```
mysql> DELETE FROM Transactions WHERE external_ref = 'TEST001';
Query OK, 2 rows affected (0.06 sec)

mysql> SELECT COUNT(*) as remaining_test_transactions FROM Transactions WHERE external_ref = 'TEST001';
+----------------------------+
| remaining_test_transactions |
+----------------------------+
|                          0 |
+----------------------------+
1 row in set (0.00 sec)

mysql>
```

```
SELECT COUNT(*) as remaining_test_transactions
FROM Transactions
WHERE external_ref = 'TEST001';
```

```
mysql> SELECT COUNT(*) as remaining_test_transactions
    -> FROM Transactions
    -> WHERE external_ref = 'TEST001';
+-----------------------------+
| remaining_test_transactions |
+-----------------------------+
|                           0 |
+-----------------------------+
1 row in set (0.02 sec)

mysql>
```

Unique rules added to enhance security and accuracy of the DB(with screenshots)

1 We have implemented the **CHECK** where we restrict adding some data's outside the specified one's. Example below:

```
CHECK (transaction_status IN ('COMPLETED', 'FAILED', 'PENDING')),
```

```
mysql> INSERT INTO Transactions (external_ref, amount, currency, transaction_status, sender_notes, raw_data, transaction
_date, counter_party, category_id, user_id) VALUES ('76662021700', -2000.00, 'RWF', 'COMPLETED', '', 'You have received
2000 RWF from Jane Smith (*********013) on your mobile money account at 2024-05-10 16:30:51. Message from sender: . Your
 new balance:2000 RWF. Financial Transaction Id: 76662021700.', '2024-05-10 16:30:51', 'Jane Smith', 1, 1);
ERROR 3819 (HY000): Check constraint 'Transactions_chk_1' is violated.
mysql>
```

2 Implemented **CONSTRAINTS** with custom names to keep data integrity (cascading delete and updates):

```
    CONSTRAINT fk_transaction_fees_fee_type FOREIGN KEY (fee_type_id)
REFERENCES Fee_Type(fee_type_id)
        ON DELETE RESTRICT
        ON UPDATE CASCADE,
```

```
mysql> DELETE FROM Momo_User WHERE user_id = 1;
ERROR 1451 (23000): Cannot delete or update a parent row: a foreign key constraint fails (`momo_sms_db`.`Transactions`,
CONSTRAINT `Transactions_ibfk_2` FOREIGN KEY (`user_id`) REFERENCES `Momo_User` (`user_id`) ON DELETE RESTRICT ON UPDATE
 CASCADE)
mysql>
```

3 Implemented **INDEX** for performance optimization:

```
    INDEX idx_fee_type_id (fee_type_id),
    INDEX idx_transaction_id (transaction_id),
```

4 NOT NULL we can not leave some fields empty.

```
 category_name VARCHAR(30) NOT NULL COMMENT 'Display name for category',
    category_code VARCHAR(30) NOT NULL UNIQUE COMMENT 'Internal code for
application logic',
```

```
mysql> INSERT INTO Momo_User (user_id, username, password_text, full_name)
 'existing_user'    -> VALUES (100, 'existing_user', 'password123', 'Test Name');
ERROR 1364 (HY000): Field 'phone_number' doesn't have a default value
mysql> INSERT INTO Transactions (transaction_id, external_ref, amount, currency, transaction_status, user_id)
    -> VALUES (999, 'TEST-REF-001', -500.00, 'ETB', 'Completed', 1);
ERROR 1364 (HY000): Field 'raw_data' doesn't have a default value
mysql>
```