Database Design Document — MoMo SMS Data Processing System

Course: Enterprise Web Development
Team: Frank Musiime, Olga IKIREZI, Placide Niyonizeye
Date: 19.09.2025

1. Overview

This document presents the relational database design for handling MoMo SMS transaction data. It covers entity relationships, design justification, data dictionary, queries, and database rules to ensure integrity and scalability.

2. ERD (Entity Relationship Diagram)

Entities:
- customer — senders and receivers of transactions
- transactions — main record of each MoMo transaction
- transaction categories — classification of transaction types
- System logs — logs of data processing data events

Relationships:
- One user → many sent or received transactions
- One transaction → many categories (via junction) - One transaction → many logs

(See attached ERD diagram in /docs/erd_diagram.pdf.)

3. Design Rationale

The schema was built around transactions as the central fact table with links to customers, categories, and system logs. This ensures that sender and receiver details are normalized, avoiding duplication and simplifying updates. Categorization is handled via a junction table to allow transactions to carry multiple tags (e.g., merchant + promotion), giving flexibility for business reporting without altering the schema.

System logs is included to capture processing events and errors for traceability, a crucial requirement for financial systems. Constraints such as CHECK (amount >= 0) and UNIQUE keys on transaction references ensure data integrity. Indexes on frequently queried fields (timestamps, status, phone numbers) improve performance for analytics and reporting.

The design balances normalization with practical needs: while customers and category data is normalized, the raw payload JSON field in transactions preserves the original SMS/XML snapshot for debugging and compliance. This dual approach supports both structured querying and raw-data validation. Overall, the design is scalable, secure, and suitable for extension (e.g., adding agents, fees, or audit trails).

4. Data Dictionary (Summary)

Table	Key Columns	Notes
customer	User id (PK), msisdn (UNIQUE)	Stores senders/receiver
transactions	Transaction id (PK), sender user id (FK), receiver user id (FK)	Main transaction data
Transaction categories	Category id (PK)	Category definitions
System logs	Log id (PK), transaction id (FK)	Logs of processing

5. Sample SQL Queries

```
mysql> describe Customer;
+--------------+-----------------------------------+------+-----+---------+-------+
| Field        | Type                              | Null | Key | Default | Extra |
+--------------+-----------------------------------+------+-----+---------+-------+
| customerId   | int                               | NO   | PRI | NULL    |       |
| first_name   | varchar(100)                      | NO   |     | NULL    |       |
| last_name    | varchar(100)                      | NO   |     | NULL    |       |
| phone_number | varchar(15)                       | NO   | UNI | NULL    |       |
| status       | enum('active','suspended','close') | YES  |     | close   |       |
| balance      | decimal(10,2)                     | YES  |     | 0.02    |       |
+--------------+-----------------------------------+------+-----+---------+-------+
6 rows in set (0.01 sec)

mysql> select * from Customer;
+------------+------------+-----------+--------------+--------+---------+
| customerId | first_name | last_name | phone_number | status | balance |
+------------+------------+-----------+--------------+--------+---------+
|       1001 | Kalisa     | John      | 250720006765 | active | 1000.00 |
+------------+------------+-----------+--------------+--------+---------+
1 row in set (0.00 sec)
```

```
mysql> describe Transaction_categories;
+---------------+--------------+------+-----+---------+-------+
| Field         | Type         | Null | Key | Default | Extra |
+---------------+--------------+------+-----+---------+-------+
| Category_id   | int          | NO   | PRI | NULL    |       |
| Category_name | varchar(50)  | NO   | UNI | NULL    |       |
| Description   | varchar(100) | YES  |     | NULL    |       |
+---------------+--------------+------+-----+---------+-------+
3 rows in set (0.04 sec)

mysql> INSERT INTO Transaction_categories VALUES(001, "Transfer", "transfer money from one customer to another");
Query OK, 1 row affected (0.05 sec)

mysql> SELECT * FROM Transaction_categories;
+-------------+---------------+-----------------------------------------+
| Category_id | Category_name | Description                             |
+-------------+---------------+-----------------------------------------+
|           1 | Transfer      | transfer money from one customer to another |
+-------------+---------------+-----------------------------------------+
1 row in set (0.00 sec)
```

6. Unique Rules & Security

```
mysql> SHOW CREATE TABLE transactions_denorm\G
*************************** 1. row ***************************
       Table: transactions_denorm
Create Table: CREATE TABLE `transactions_denorm` (
  `Transaction_id` int NOT NULL AUTO_INCREMENT,
  `external_txn_id` varchar(50) NOT NULL,
  `Amount` decimal(10,2) NOT NULL,
  `Sender_id` int NOT NULL,
  `Receiver_id` int NOT NULL,
  `Sender_name` varchar(201) NOT NULL,
  `Receiver_name` varchar(201) NOT NULL,
  `Timestamp` datetime NOT NULL,
  `Currency` char(3) NOT NULL,
  `Status` enum('active','suspended','close') DEFAULT 'close',
  `Category_id` int DEFAULT NULL,
  PRIMARY KEY (`Transaction_id`),
  UNIQUE KEY `external_txn_id` (`external_txn_id`),
  KEY `Sender_id` (`Sender_id`),
  KEY `Receiver_id` (`Receiver_id`),
  KEY `Category_id` (`Category_id`),
  CONSTRAINT `transactions_denorm_ibfk_1` FOREIGN KEY (`Sender_id`) REFERENCES `Customer` (`customerId`) ON DELETE CASCADE ON UPDATE
CASCADE,
  CONSTRAINT `transactions_denorm_ibfk_2` FOREIGN KEY (`Receiver_id`) REFERENCES `Customer` (`customerId`) ON DELETE CASCADE ON UPDAT
E CASCADE,
  CONSTRAINT `transactions_denorm_ibfk_3` FOREIGN KEY (`Category_id`) REFERENCES `Transaction_categories` (`Category_id`) ON DELETE S
ET NULL ON UPDATE CASCADE,
  CONSTRAINT `transactions_denorm_chk_1` CHECK ((`Amount` > 0)),
  CONSTRAINT `transactions_denorm_chk_2` CHECK ((char_length(`Currency`) = 3))
) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
1 row in set (0.00 sec)
```