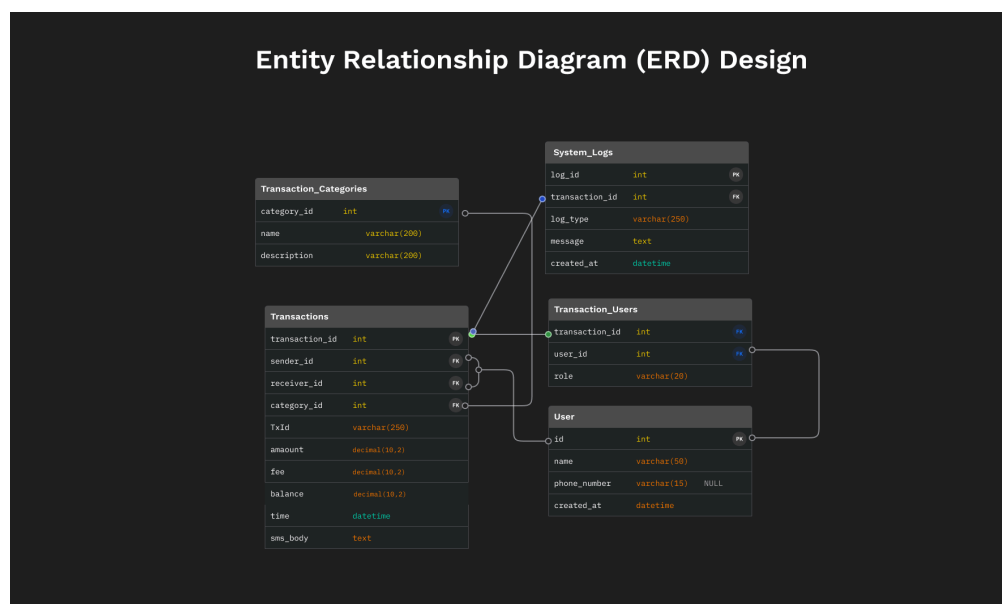


Entity Relationship Diagram (ERD) Design & Justification (Documentation)

Entity Relationship Diagram (ERD)

Below are ERD diagrams that illustrate the database schema for the MoMo SMS data processing system. They capture the entities, attributes, relationships, and keys used to ensure data integrity and scalability.



ERD Design Justification

- The Entity Relationship Diagram (ERD), we designed was to reshape mobile money (MoMo) transactions extracted from SMS data given in xml file. The design captures core entities and relationships necessary for robust financial data processing.
- The **Transactions** entity is central, storing details such as transaction amount, fee, balance, timestamp as time in our table's field, and raw SMS body. It links directly to **Transaction_categories**, which classifies transactions into types such as transfers, deposits, and payments. This separation supports flexibility in

categorization and reporting.

- The **User** entity manages customer details, including names and phone numbers, while sender and receiver roles in a transaction are represented using foreign keys. To support more complex associations where multiple users may be linked to a single transaction, the **Transaction_users** junction table resolves the many-to-many relationship and specifies user roles (e.g., sender, receiver, agent).
- To ensure accountability, **System_logs** records processing metadata such as log type, messages, and timestamps. This entity provides auditability and helps track errors or processing activities
- Generally, ERD is assured of integrity of information, scalability and trace-ability, similar to the actual MoMo SMS data structure. The design can do both transactional and analytical applications and thus it can be implemented in a financial system..

Data Dictionary

TABLE	COLUMN	TYPE	DESCRIPTION
users	user_id	INT (PK)	Unique user identifier
users	full_name	VARCHAR(150)	Full name of the user
users	phone	VARCHAR(20)	Unique phone number in E.164 format
users	email	VARCHAR(255)	Optional email address
users	is_kyc_done	TINYINT(1)	1 if KYC completed
users	user_role	ENUM	Role of user (customer, merchant, agent, system)
transactions	transaction_id	BIGINT (PK)	Internal transaction identifier
transactions	momo_reference	VARCHAR(100)	Provider transaction reference
transactions	amount	DECIMAL(13,2)	Transaction amount (>= 0)
transactions	currency	CHAR(3)	ISO currency code (e.g., RWF)
transactions	occurred_at	DATETIME	Time the transaction occurred
transactions	sender_id	INT (FK)	Foreign key to users (sender)

transactions	receiver_id	INT (FK)	Foreign key to users (receiver)
transactions	direction	ENUM	Transaction direction: IN/OUT
transactions	status	ENUM	Transaction status (PENDING, COMPLETED)
transactions	raw_payload	JSON	Original SMS/XML payload for audit
transaction_categorie	scategory_id	INT (PK)	Unique category identifier
transaction_categorie	scode	VARCHAR(50)	Short category code (e.g., P2P, BILL)
transaction_categorie	sname	VARCHAR(120)	Human-readable category name
transaction_category	map_id	INT (PK)	Junction table PK
transaction_category	maptransaction_id	BIGINT (FK)	FK to transactions
transaction_category	mapcategory_id	INT (FK)	FK to categories
system_logs	log_id	BIGINT (PK)	Unique log identifier
system_logs	processing_stage	VARCHAR(80)	Stage of data pipeline
system_logs	severity	ENUM	Log severity (DEBUG, INFO, WARNING, ERROR)
system_logs	message	TEXT	Log message details
system_logs	transaction_id	BIGINT (FK)	Optional FK to related transaction
system_logs	meta	JSON	Structured metadata

- Sample queries demonstrating your database functionality(with screenshots).

1. Database Setup and sample data in the Transactions table

The screenshot below shows the DB structure(set up) and schema creation, and also provides proof showing all databases we have with {SHOW DATABASES}, and also shows a sample of inserted records into the transactions table

```

mysql> source C:/Users/methu/Downloads/My_Project/momo-sms-data-analysis/database_setup.sql;
Query OK, 1 row affected (0.01 sec)

Database changed
Query OK, 0 rows affected (0.04 sec)

Query OK, 0 rows affected (0.02 sec)

Query OK, 0 rows affected (0.04 sec)

Query OK, 0 rows affected (0.02 sec)
Records: 5 Duplicates: 0 Warnings: 0

Query OK, 0 rows affected (0.01 sec)
Records: 0 Duplicates: 0 Warnings: 0

Query OK, 0 rows affected (0.04 sec)

Query OK, 0 rows affected (0.02 sec)

Query OK, 5 rows affected (0.00 sec)
Records: 5 Duplicates: 0 Warnings: 0

Query OK, 5 rows affected (0.00 sec)
Records: 5 Duplicates: 0 Warnings: 0

Query OK, 5 rows affected (0.00 sec)
Records: 5 Duplicates: 0 Warnings: 0

Query OK, 10 rows affected (0.00 sec)
Records: 10 Duplicates: 0 Warnings: 0

Query OK, 5 rows affected (0.00 sec)
Records: 5 Duplicates: 0 Warnings: 0

+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| transaction_id | sender_id | receiver_id | category_id | TxId | amount | fee | balance | time | sms_body |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | 1 | 2 | 2 | 76603221706 | 2000.00 | 0.00 | 2000.00 | 2024-05-10 16:30:51 | You have Received 2000 RwF... |
| 2 | 1 | 2 | 2 | 73214040427 | 1000.00 | 0.00 | 1000.00 | 2024-05-10 16:31:39 | Your payment of 1,000 RwF... |
| 3 | 1 | 2 | 2 | 51732411227 | 600.00 | 0.00 | 600.00 | 2024-05-10 21:32:32 | Your payment of 600 RwF... |
| 4 | 4 | 1 | 1 | 17618952211 | 40000.00 | 0.00 | 40000.00 | 2024-05-11 18:43:09 | A bank deposit of 40000 RwF... |
| 5 | 1 | 3 | 3 | 82113964658 | 3500.00 | 0.00 | 10000.00 | 2024-05-12 13:34:25 | Your payment of 3,500 RwF... |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0

Query OK, 1 row affected (0.00 sec)

Query OK, 1 row affected (0.00 sec)

mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| momo_sms |
| mysql |
| performance_schema |
| sys |
| testttt |
+-----+
0 rows in set (0.00 sec)

```

2.Us (Insert and Update)er CRUD just testing

- Firstly, we started by running dabase_set.sql script , we navigated to that Database using(USE momo_sms)
- Secondly, we showed that all tables are in our database
- Thirdly, we show the table users before inserting the new data(user)
- The User table enforces NOT NULL on the name column. This ensures that every user must have a valid name
- Then, after we try to insert a new user using the insert and values keywords. Later, we selected all the table users we could see that new users(Enzo and Delphine were added)
- Lastly we updated the user called Steven Kayitare's Phone number

4.Unique rules added to enhance security and accuracy of the DB

```
Windows PowerShell
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| momo_sms |
| mysql |
| performance_schema |
| sys |
| testttt |
+-----+
6 rows in set (0.00 sec)

mysql> use momo_sms;
Database changed
mysql> show tables;
+-----+
| Tables_in_momo_sms |
+-----+
| system_logs |
| transaction_categories |
| transaction_users |
| transactions |
| user |
+-----+
5 rows in set (0.00 sec)

mysql> describe user;
+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+
| id | int | NO | PRI | NULL | auto_increment |
| name | varchar(50) | NO | | NULL | |
| phone_number | varchar(15) | YES | UNI | NULL | |
| created_at | datetime | YES | | CURRENT_TIMESTAMP | DEFAULT_GENERATED |
+-----+
4 rows in set (0.01 sec)

mysql> insert into user (name, phone_number)
-> ^C
mysql> select * from user;
+-----+
| id | name | phone_number | created_at |
+-----+
| 1 | Steven Kayitare | 250790265770 | 2025-09-18 18:40:38 |
| 2 | Ishimwe Diane | 250790777734 | 2025-09-18 18:40:38 |
| 3 | Alex Robert | 250799200583 | 2025-09-18 18:40:38 |
| 4 | Cash Deposit Agent | 250799963036 | 2025-09-18 18:40:38 |
| 6 | Kenneth Enzo | 250791265656 | 2025-09-18 18:44:47 |
| 7 | Iriho Delphine | 250790375634 | 2025-09-18 18:44:47 |
+-----+
6 rows in set (0.00 sec)

mysql> insert into user (name, phone_number)
-> values ("Dinah Turikumwe", "250790375634")
-> ;
ERROR 1062 (23000): Duplicate entry '250790375634' for key 'user.phone_number'
mysql>
```

From this, you can see we started by describing the field for the Table user(as you can see, the field ID is a primary key, and also the phone_number field has to be unique). We also selected all from the table to show how the data of the table user, and we tried to insert a new user called Dinah Turikumwe, but it just threw an error since in our database we also had another user with the same phone_number, and the phone_number attribute was set to be unique