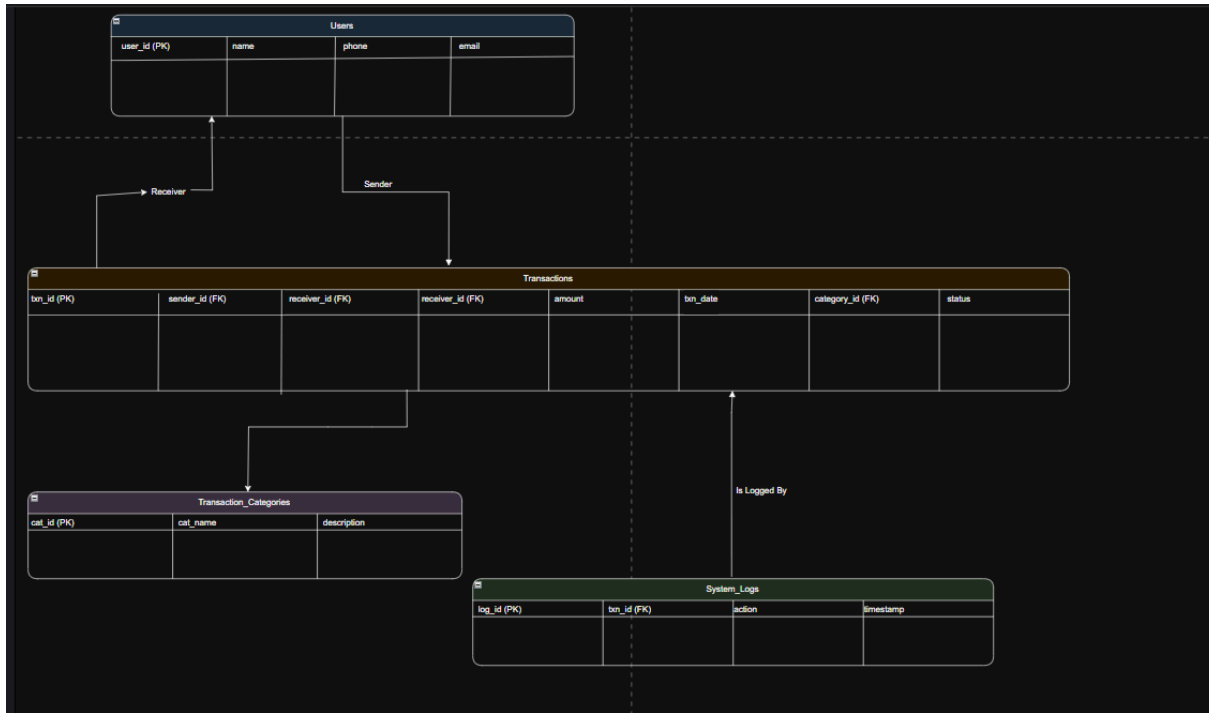


DATABASE DESIGN JUSTIFICATION



For our SMS Financial Tracker project, we defined the database to record the key aspects of mobile money transactions in a consistent, query-friendly, and straightforward manner so that it becomes user-friendly. The key entities are Users, Transactions, Transaction_Categories, and System_Logs as suggested.

We utilized a Users table because every transaction must have at least two people: a sender and a receiver. Instead of using two sender and receiver tables, both roles have a primary key referencing the same Users table. This avoids duplication and ensures that everyone's data (name, phone, email) is stored only once.

The Transactions table is the key table in the system. It holds transactional data like amount, date, status, and foreign keys to sender, receiver, and category. We capture that users can be in many roles by linking sender_id and receiver_id back to Users. That's where a many-to-many relationship naturally exists: a user can send many transactions and receive many. We worked around this by storing sender and receiver IDs directly in the Transactions table, instead of requiring a second junction table.

The Transaction_Categories table includes the functionality of grouping transactions (i.e., airtime, deposit). This is for future reporting and analysis. Finally, the System_Logs table logs actions and errors, providing administrators with feedback regarding how data is being processed.

In total, this design eliminates redundancy, imposes relationships using primary and foreign keys, and offers flexibility for future added functionality such as analytics or fraud detection.

Sample Queries Demonstrating Our Database Functionality & Unique Rules Added to Enhance Security and Accuracy of the DB

The screenshots demonstrate the database's functionality by showing live query results. The tables for Users, Transactions, Categories, and Logs are displayed with sample data, proving that the database schema is correctly implemented and can store information. For example, the JOIN query from the table who sent money to whom shows the relationship between the Users and Transaction tables correctly.

The unique rules show that security and accuracy rules have been implemented. For example, the error message for attempting to add a negative amount (6). This demonstrates a CHECK constraint in action, a vital rule that prevents the insertion of invalid data and ensures the integrity of financial records.

```
● gedeon@pop-os:~/Class/MoMo_SMS_Financial_Tracker$ ./demo_script.sh
=== MoMo SMS Financial Tracker Demo ===
This shows my database system working

1. Showing all users in my database:
[sudo] password for gedeon:
+-----+-----+-----+
| user_id | name       | phone   |
+-----+-----+-----+
| 1 | Alice Johnson | 0788000001 |
| 2 | Bob Smith    | 0788000002 |
| 3 | Clara Davis  | 0788000003 |
| 4 | Daniel Wilson | 0788000004 |
| 5 | Eve Brown    | 0788000005 |
| 6 | Frank Miller | 0788000006 |
+-----+-----+-----+

2. Showing transaction categories:
+-----+-----+
| cat_id | cat_name |
+-----+-----+
| 1 | Send Money |
| 2 | Withdraw  |
| 3 | Deposit   |
| 4 | Airtime   |
| 5 | Merchant  |
| 6 | Bill Payment |
+-----+-----+
```

3. Showing recent transactions:

txn_id	amount	txn_date	status
6	8500.00	2025-09-19 15:10:00	pending
5	12000.00	2025-09-19 14:20:00	completed
4	7500.00	2025-09-19 13:45:00	completed
3	20000.00	2025-09-19 12:00:00	completed
2	5000.00	2025-09-19 11:15:00	completed

4. Showing how tables are connected (who sent money to whom):

Sender	Receiver	Amount	Category
Frank Miller	Bob Smith	8500.00	Send Money
Eve Brown	Alice Johnson	12000.00	Withdraw
Daniel Wilson	Eve Brown	7500.00	Merchant
Clara Davis	Alice Johnson	20000.00	Deposit
Bob Smith	Clara Davis	5000.00	Airtime

5. Showing system logs (audit trail):

log_id	txn_id	action	timestamp
6	6	INSERT	2025-09-19 18:00:14
5	5	INSERT	2025-09-19 18:00:14
4	4	INSERT	2025-09-19 18:00:14
3	3	INSERT	2025-09-19 18:00:14
2	2	INSERT	2025-09-19 18:00:14

6. Testing data validation (trying to add invalid data):

Trying to add negative amount (should fail):

ERROR 3819 (HY000) at line 1: Check constraint 'chk_amount_positive' is violated.

7. Showing database statistics:

Table	Records
Users	6
Transactions	6
Categories	6
Logs	6

=== Demo Complete ===

This shows my database is working correctly with all the features I built!