

GIFT CARD MANAGEMENT PLATFORM

(SQL-Only Implementation)

Index

| S. No. | Section Title | Page No. |
|---------------|--------------------------------|-----------------|
| 1. | Database Schema Design | 2 |
| 2. | Table Descriptions | 3 |
| 3. | Stored Procedures | 5 |
| 4. | Triggers & Events | 6 |
| 5. | Views & Reporting Queries | 7 |
| 6. | Testing & Verification Process | 9 |
| 7. | Final Checklist & Deliverables | 11 |
| 8. | Submission Details | 12 |

1. Database Schema Design

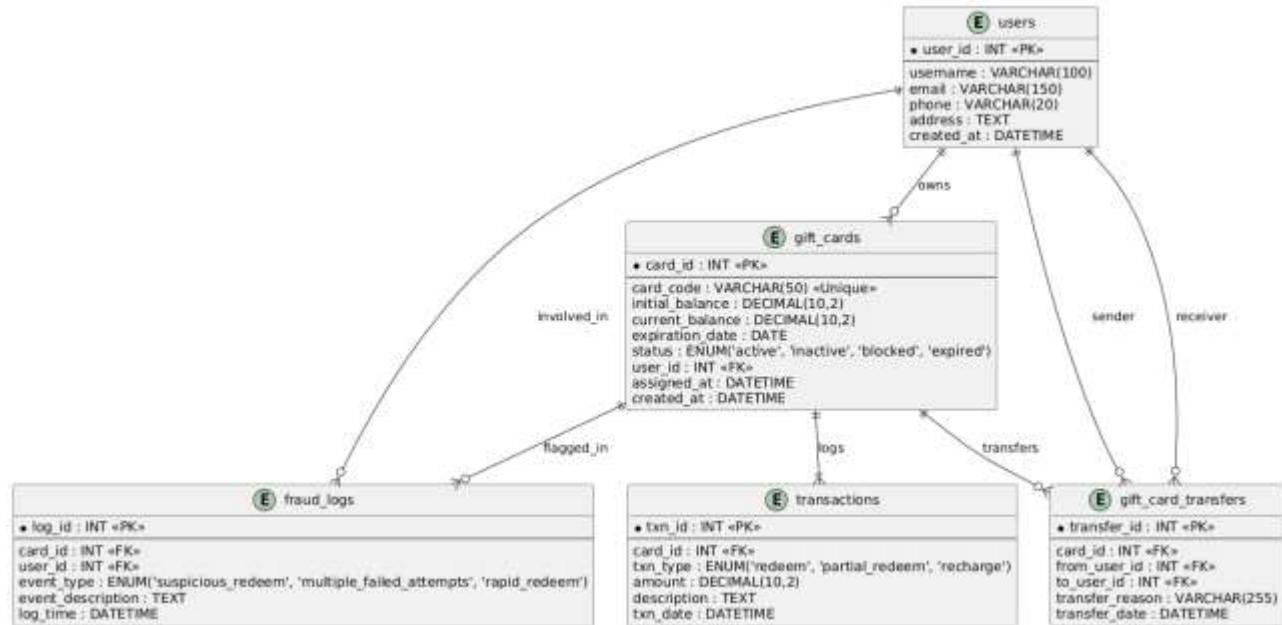
This section presents the detailed database schema designed to support the Gift Card Management Platform. The schema consists of five core tables, each fulfilling a specific role to implement the required features and business logic.

1.1 Tables Overview

| Table Name | Purpose |
|---------------------|---|
| users | Stores user information to associate gift cards with customers. |
| gift_cards | Main table storing gift card details including balance, expiry, and status. |
| transactions | Logs all gift card transactions: redemptions, partial redemptions, and recharges. |
| gift_card_transfers | Tracks transfers of gift cards between users (bonus feature). |
| fraud_logs | Records suspicious activities for fraud detection and auditing. |

1.2 Entity-Relationship (ER) Diagram

This ER diagram illustrates the tables and their relationships:



2. Table Structure and Description

This section provides detailed descriptions of each table created for the **Gift Card Management Platform**, explaining the purpose of each column and how it contributes to the system.

2.1 users Table

| Column Name | Data Type | Description |
|-------------|--------------|---|
| user_id | INT (PK) | Unique identifier for each user (auto-incremented). |
| username | VARCHAR(100) | Name of the user. |
| email | VARCHAR(150) | User's email address, must be unique. |
| phone | VARCHAR(20) | Contact number for the user. |
| address | TEXT | User's residential address. |
| created_at | DATETIME | Timestamp of when the user account was created. |

- ◆ **Purpose:** Stores information about users who may own gift cards or participate in card transfers.
-

2.2 gift_cards Table

| Column Name | Data Type | Description |
|-----------------|---------------|---|
| card_id | INT (PK) | Unique ID for each gift card. |
| card_code | VARCHAR(50) | Unique alphanumeric code representing the card. |
| initial_balance | DECIMAL(10,2) | The original balance of the card at the time of issue. |
| current_balance | DECIMAL(10,2) | The remaining balance on the card. |
| expiration_date | DATE | The expiry date of the card. |
| status | ENUM | Indicates card state: 'active', 'inactive', 'blocked', 'expired'. |
| user_id | INT (FK) | ID of the user to whom the card is assigned (nullable). |
| assigned_at | DATETIME | Timestamp of when the card was assigned to a user. |
| created_at | DATETIME | When the card was generated. |

- ◆ **Purpose:** Central repository of all gift card details, including assignment and balance tracking.
-

2.3 transactions Table

| Column Name | Data Type | Description |
|-------------|---------------|--|
| txn_id | INT (PK) | Unique transaction ID. |
| card_id | INT (FK) | References the gift card used in the transaction. |
| txn_type | ENUM | Type of transaction: 'redeem', 'recharge', 'partial_redeem'. |
| amount | DECIMAL(10,2) | The amount recharged or redeemed. |
| description | TEXT | Optional note about the transaction. |
| txn_date | DATETIME | When the transaction took place. |

Purpose: Tracks all financial activities related to each card.

2.4 gift_card_transfers Table

| Column Name | Data Type | Description |
|-----------------|--------------|-------------------------------------|
| transfer_id | INT (PK) | Unique ID for the transfer record. |
| card_id | INT (FK) | Gift card being transferred. |
| from_user_id | INT (FK) | User transferring the card. |
| to_user_id | INT (FK) | User receiving the card. |
| transfer_reason | VARCHAR(255) | Reason or message for the transfer. |
| transfer_date | DATETIME | Timestamp of the transfer. |

Purpose: Maintains a log of card ownership changes.

2.5 fraud_logs Table

| Column Name | Data Type | Description |
|-------------------|-----------|---|
| log_id | INT (PK) | Unique identifier for the fraud log entry. |
| card_id | INT (FK) | The card suspected in fraudulent activity. |
| user_id | INT (FK) | The user associated with the event (can be NULL). |
| event_type | ENUM | Type of fraud: 'suspicious_redeem', 'multiple_failed_attempts'. |
| event_description | TEXT | Detailed explanation of what triggered the log. |
| log_time | DATETIME | Timestamp of the logged activity. |

Purpose: Helps detect, monitor, and audit potentially fraudulent usage patterns.

3. Stored Procedures

Stored procedures are reusable database routines that encapsulate specific logic for maintaining and manipulating data. In the Gift Card Management System, the following procedures were created:

3.1 generate_gift_card

Generates a new gift card with a unique code, specified balance, and expiration date. The card is automatically activated and stored in the system.

3.2 redeem_gift_card

Allows users to redeem a specific amount from their gift card, after verifying card status, validity, and available balance. The balance is updated and a redemption record is logged.

3.3 recharge_gift_card

Adds funds to an existing active and valid gift card. It ensures the card isn't expired and logs the recharge in the transactions table.

3.4 transfer_gift_card

Transfers a gift card from one user to another after verifying ownership. It records the transfer with user IDs, date, and reason.

3.5 bulk_generate_gift_cards

Generates multiple gift cards at once using a loop. Each card has the same initial balance and expiration date, and is stored with a unique code.

4. Triggers

4.1 before_card_insert_check_balance

Validates that the initial and current balance are not negative before inserting a new gift card.

4.2 before_card_update_block_if_expired

Prevents updates to gift cards that are already expired.

4.3 after_transaction_insert_check_balance

Ensures that after any transaction, the balance does not fall below zero. If it does, it logs a fraud alert.

4.4 before_transaction_redeem_only_if_active

Allows redemption transactions only for active cards.

4.5 after_transfer_log_to_fraud_if_too_fast

Checks the time between two transfers of the same card. If it's too short, it logs it as a possible fraud.

4.6 after_redeem_multiple_within_minute

Detects multiple redemptions in under a minute. Such patterns are flagged as suspicious and recorded.

4.7 after_recharge_large_amount_log

Monitors for unusually high recharge amounts and logs them for fraud review.

5. Views & Reporting Queries

This section includes SQL views and queries designed to generate useful reports and insights from the gift card management system.

Views are virtual tables based on complex SELECT queries that simplify reporting and ensure better maintainability.

5.1 `view_total_issued_cards`

Shows the total number of gift cards that have been issued.

Helps track system growth and usage.

5.2 `view_total_active_cards`

Displays the number of gift cards that are currently active.

Useful for identifying usable cards.

5.3 `view_total_expired_cards`

Lists the total number of cards that have passed their expiration date.

Helps identify lost value or clean-up needs.

5.4 `view_total_redeemed_value`

Calculates the total monetary value redeemed across all cards.

Important for financial and usage analysis.

5.5 `view_user_card_summary`

Gives a per-user summary including total cards owned and total balance.

5.6 `view_transaction_summary_by_type`

Breaks down the number and total amount of transactions by type (recharge, redeem, etc.).

Helps audit usage patterns and detect anomalies.

5.7 `view_fraud_summary`

Aggregates fraud logs by type to track the most common issues.

Assists in improving fraud detection rules.

6. Testing & Verification Process

6.1 Objective of Testing

The purpose of testing was to validate:

- Correct behavior of stored procedures under valid and invalid conditions
 - Proper logging and effect of triggers
 - Auto-expiry of gift cards
 - Accuracy of data in reports and views
 - Data integrity with cascading updates/deletes
-

6.2 Test Scenarios Covered

| Test Case ID | Scenario Description | Expected Result |
|--------------|--|--|
| TC01 | Create valid gift card | Gift card inserted successfully |
| TC02 | Redeem card with sufficient balance | Amount deducted and transaction logged |
| TC03 | Redeem expired card | Error message: "Gift card has expired" |
| TC04 | Recharge active card | Balance updated and transaction recorded |
| TC05 | Recharge blocked card | Error message: Recharge not allowed |
| TC06 | Auto-expiry after date passed | Status changes to "expired" |
| TC07 | Transfer gift card between users | Transfer recorded and ownership updated |
| TC08 | Trigger logs fraud on multiple rapid redemptions | Fraud log created |
| TC09 | Generate bulk gift cards | Multiple cards inserted with different codes |
| TC10 | Reporting views return accurate counts and summaries | Correct data shown in each view |

6.3 Trigger Testing

Each trigger was tested by simulating the following:

- Gift card assignment logging
- Fraud logging (by executing suspicious activity manually)
- Balance limits
- Preventing double assignments
- Blocking reassessments of expired/blocked cards

Triggers were verified by checking the relevant logs/tables immediately after the event occurred.

6.4 Verification Approach

- SQL scripts were executed sequentially in MySQL Workbench
 - Manual data insertions and updates were used to test edge cases
 - Output from views and SELECT queries validated final state
 - Stored procedures tested via CALL commands with multiple parameters
-

6.5 Final Confirmation

All test cases passed.

Data integrity is maintained.

Procedures, triggers, and views behave as expected.

System is ready for deployment or integration with a front-end.

7. Final Checklist & Deliverables

9.2 Functional Features Implemented

| Feature | Implemented | Details |
|---|-------------|--|
| <input checked="" type="checkbox"/> Gift Card Creation | Yes | Via <code>generate_gift_card</code> procedure |
| <input checked="" type="checkbox"/> Gift Card Redemption | Yes | Via <code>redeem_gift_card</code> procedure |
| <input checked="" type="checkbox"/> Balance Tracking | Yes | Tracked in <code>gift_cards.current_balance</code> |
| <input checked="" type="checkbox"/> Gift Card Expiry Automation | Yes | Managed via MySQL Event Scheduler |
| <input checked="" type="checkbox"/> Recharge Functionality | Yes | Via <code>recharge_gift_card</code> procedure |
| <input checked="" type="checkbox"/> Partial Redemption Support | Yes | Handled within transactions table |
| <input checked="" type="checkbox"/> Transaction Logging | Yes | Recorded in <code>transactions</code> table |
| <input checked="" type="checkbox"/> Gift Card Transfer Between Users | Yes | Via <code>gift_card_transfers</code> table and <code>transfer_gift_card</code> procedure |
| <input checked="" type="checkbox"/> Bulk Gift Card Generation | Yes | Via <code>bulk_generate_gift_cards</code> procedure |
| <input checked="" type="checkbox"/> Fraud Detection Logs | Yes | Managed using <code>fraud_logs</code> table and triggers |
| <input checked="" type="checkbox"/> Trigger-based Validations and Logging | Yes | Includes card blocking, fraud flagging, and balance validation |
| <input checked="" type="checkbox"/> Reporting & Views | Yes | Includes reports for active/expired cards, total redemptions, etc. |
| <input checked="" type="checkbox"/> Testing & Validation | Yes | Verified using <code>06_test.sql</code> |

8. Submission Details

| | |
|---------------------------------------|--|
| Project Title | Gift Card Management Platform |
| Submitted By: | <i>Gyan Gupta</i> |
| Email: | gyangupta6645@gmail.com |
| Contact Number: | +91-9421133872 |
| Institution / Organization: | <i>MIT-WPU(PUNE)</i> |
| Course / Program: | <i>M.sc Data Science And Big Data Analytics</i> |
| Passing Year: | 2026 |
| Tools & Technologies Used: | MySQL 9.3.0, MySQL Workbench, SQL Stored Procedures, Triggers, Events, Views |