Gift Card Management Platfrom using Mysql

Core Features:-

1. Gift Card Generation: Ability to generate unique gift cards with customizable initial balances and expiration dates. For this i have make a procedure name "**generate_gift_cards**" to generate a single card and "**bulk_generate_gift_cards**" to generate a bulk gift cards and after generating bulk gift card I have created another procedure name **"assign_gift_card_to_user"** which will help to assign card to the user_id.

2. Gift Card Redemption & Fraud Detection: Mechanism for redeeming gift cards, deducting the redeemed amount from the balance. For this i have make procedure name **"redeem_gift_card".** While redeeming the gift card it will also simultaneously check that it is fraud transaction or not.

3. Balance Tracking: Real-time tracking and querying of gift card balances. For this I have make a table name **"gift_cards"** in which will help to check the real time balance.

4. Transaction Logging: Maintain a complete transaction history for each gift card, including redemptions and recharges. For this I have make a table name **"gift_card_transaction_table",** this table will help us to maintain transaction of cards;

5. Gift Card Expiry & automatic expiry handling: Automatic handling of expired cards, preventing further usage. For this I have created a procedure name **"expire_gift_card"** which will help us to maintain records of expiry cards and it will also handle the automatic expiry.

6. User Association: Optionally associate gift cards with users (support both as signed and unassigned cards). This feature is included in the procedure name **"bulk_generate_gift_cards"** it will generate a bulk cards without assigning to the user and its status by default will be **"inactive"** to prevent wrong redemption of the card after generating bulk cards we can assign one of the card to the user.

7. Gift Card Status Management: Support for activating, deactivating, and blocking gift cards. For this I have created a procedure name **"update_gift_card_status"** which will help to manage the status of the cards like **"active","inactive","blocked","expired",** which will help us to sort out the cards on the basis on its status.

8. Reporting:

Gift Card Summary:

- Total number of cards.
- Number of cards by status (active, inactive, blocked, expired).
- Total initial balance and total current balance (available funds).
- Cards per user.

Transaction Summary:

- Total transaction count by type (redemption, recharge, transfer_in, transfer_out).
- Total amounts per transaction type.
- Transactions by user or card.
- Transaction history per card.

Activity and Usage:

- Redemption amounts and frequency over time (daily, monthly).
- Recharge activity over time.
- Transfers between users.
- Expired cards count.

Bonus Features:-

Partial Redemption Support: Allow partial usage of gift card balance in multiple transactions. This feature is used in procedure name **"redeem_gift_card"** which checks if amount <= current_balance.

Gift Card Recharge: Enable recharging of existing gift cards. This feature is used in procedure name **"recharge_gift_card"** which allows to make recharge .

Bulk Gift Card Generation: Efficiently generate multiple gift cards at once. This feature is used in procedure name **"bulk_generate_gift_cards"** to generate bulk gift card.

Gift Card Transfer: Allow transferring a gift card from one user to another. This feature is used in procedure name **"transfer_gift_card"** which allows to transfer card to another user_id.

Fraud Detection: Implement basic checks to detect and prevent suspicious activities. This feature is used in procedure name **"redeem_gift_card"** which redeem gift card as well as simultaneously also check fraud detection and unless the updated user_id doesn't match to the card the redemption doesn't occur.

Assignning Card: I have made an extra procedure name **"assign_gift_card_to_user"** which help us to assign card to the user from the bulk cards to save our time.

Source Code

First of all we create a Database name Gift Card Platform and we will use it

```
create database gift_card_platform;
use gift_card_platform;
```

Then we will create a table of users,gift_cards,gift_card_transactions

- **User Table**

```
CREATE TABLE users (
  user_id INT unique AUTO_INCREMENT PRIMARY KEY,
  username VARCHAR(100) UNIQUE NOT NULL,
  email VARCHAR(255) UNIQUE NOT NULL,
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

- **Gift card table**

```
CREATE TABLE gift_cards (
  card_id CHAR(7) PRIMARY KEY,
  initial_balance DECIMAL(10,2) NOT NULL,
  current_balance DECIMAL(10,2) NOT NULL,
  expiration_date DATE NOT NULL,
  status ENUM('active', 'inactive', 'blocked', 'expired') DEFAULT 'active',
  user_id INT NULL,
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
  CONSTRAINT fk_user FOREIGN KEY (user_id)
    REFERENCES users(user_id)
    ON DELETE SET NULL,
  CHECK (current_balance >= 0)
);
```

- **Transaction table to check redemption,recharge,transfer**

```
CREATE TABLE gift_card_transactions (
  transaction_id BIGINT AUTO_INCREMENT PRIMARY KEY,
  card_id CHAR(7) NOT NULL,
  user_id INT NULL,
  transaction_type ENUM('redemption', 'recharge', 'transfer_out', 'transfer_in') NOT NULL,
  amount DECIMAL(10,2) NOT NULL,
  transaction_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  notes VARCHAR(255),
  CONSTRAINT fk_card FOREIGN KEY (card_id)
    REFERENCES gift_cards(card_id)
    ON DELETE CASCADE,
  CONSTRAINT fk_user_tx FOREIGN KEY (user_id)
    REFERENCES users(user_id)
    ON DELETE SET NULL
);
```

After creating table we will create a function and procedures

- **Function to generate unique card_id**

```
DELIMITER //

CREATE FUNCTION generate_card_id()
RETURNS CHAR(7)
DETERMINISTIC
BEGIN
  DECLARE card CHAR(7);
  DECLARE done INT DEFAULT 0;

  REPEAT
    SET card = SUBSTRING(MD5(RAND()), 1, 7);
    -- Check if card exists
    IF (SELECT COUNT(*) FROM gift_cards WHERE card_id = card) = 0 THEN
      SET done = 1;
    END IF;
  UNTIL done = 1 END REPEAT;

  RETURN card;
END //
```

- **Generate single gift card**

```
DELIMITTER //
CREATE PROCEDURE generate_gift_card (
  IN init_balance DECIMAL(10,2),
  IN expiry DATE,
  IN associated_user INT,
  OUT out_card_id CHAR(7)
)
BEGIN
  DECLARE new_card CHAR(7);
  SET new_card = generate_card_id();

  INSERT INTO gift_cards(card_id, initial_balance, current_balance, expiration_date, user_id)
  VALUES (new_card, init_balance, init_balance, expiry, associated_user);

  SET out_card_id = new_card;
END //
```

- **Bulk Gift Card Generate**

```
DELIMITER //

CREATE PROCEDURE bulk_generate_gift_cards (
  IN count INT,
  IN init_balance DECIMAL(10,2),
  IN expiry DATE,
  IN associated_user INT
)
BEGIN
  DECLARE i INT DEFAULT 0;
  DECLARE new_card CHAR(7);

  -- Validate input parameters
  IF count <= 0 THEN
      SIGNAL SQLSTATE '45000'
      SET MESSAGE_TEXT = 'Count must be a positive integer';
  END IF;

  IF init_balance <= 0 THEN
      SIGNAL SQLSTATE '45000'
      SET MESSAGE_TEXT = 'Initial balance must be positive';
  END IF;

  IF expiry <= CURDATE() THEN
      SIGNAL SQLSTATE '45000'
      SET MESSAGE_TEXT = 'Expiration date must be in the future';
  END IF;

  -- Start transaction for bulk operation
  START TRANSACTION;

  -- Generate the specified number of gift cards
  WHILE i < count DO
      SET new_card = generate_card_id();

      -- Insert with explicit 'inactive' status
      INSERT INTO gift_cards(
          card_id,
          initial_balance,
          current_balance,
          expiration_date,
          user_id,
          status
      )
      VALUES (
          new_card,
          init_balance,
          init_balance,
          expiry,
          associated_user,
          'inactive'  -- Explicitly set status to inactive
      );

      SET i = i + 1;
  END WHILE;
```

```
   COMMIT;

   -- Return success message
   SELECT CONCAT('Successfully generated ', count, ' inactive gift cards') AS message;
END //

DELIMITER ;
```

■ Redeem Gift Card

```
CREATE PROCEDURE redeem_gift_card (
   IN p_card_id CHAR(7),
   IN p_user_id INT,
   IN p_amount DECIMAL(10,2),
   OUT p_success BOOLEAN,
   OUT p_message VARCHAR(255)
)
BEGIN
   DECLARE v_balance DECIMAL(10,2);
   DECLARE v_status ENUM('active', 'inactive', 'blocked', 'expired');
   DECLARE v_expired BOOL;
   DECLARE v_user_id INT;

   proc_end: BEGIN
      -- Card existence and lock the row for concurrency safety
      SELECT current_balance, status, (expiration_date < CURDATE()), user_id
      INTO v_balance, v_status, v_expired, v_user_id
      FROM gift_cards
      WHERE card_id = p_card_id
      FOR UPDATE;

      -- If no row found (card doesn't exist)
      IF v_user_id IS NULL AND v_balance IS NULL THEN
         SET p_success = FALSE;
         SET p_message = 'Gift card not found';
         LEAVE proc_end;
      END IF;

      -- Check ownership: user must match assigned user_id
      IF v_user_id IS NULL OR v_user_id != p_user_id THEN
         SET p_success = FALSE;
         SET p_message = 'User does not own this gift card or card is unassigned';
         LEAVE proc_end;
      END IF;

      -- Update status if expired
      IF v_expired AND v_status != 'expired' THEN
UPDATE gift_cards SET status = 'expired', updated_at = NOW() WHERE card_id = p_card_id;
         SET p_success = FALSE;
         SET p_message = 'Gift card is expired';
         LEAVE proc_end;
```

```
        END IF;

        -- Check card status
        IF v_status != 'active' THEN
           SET p_success = FALSE;
           SET p_message = CONCAT('Gift card status is ', v_status);
           LEAVE proc_end;
        END IF;

        -- Balance check
        IF v_balance < p_amount THEN
           SET p_success = FALSE;
           SET p_message = 'Insufficient balance';
           LEAVE proc_end;
        END IF;

        -- Basic fraud detection
        IF p_amount > 1000 THEN
           SET p_success = FALSE;
           SET p_message = 'Transaction exceeds allowed limit';
           LEAVE proc_end;
        END IF;

        -- Update balance
        UPDATE gift_cards
        SET current_balance = current_balance - p_amount,
           updated_at = NOW()
        WHERE card_id = p_card_id;

        -- Log transaction
        INSERT INTO gift_card_transactions(card_id, user_id, transaction_type, amount, notes)
VALUES (p_card_id, p_user_id, 'redemption', p_amount, 'Redemption');

SET p_success = TRUE;
SET p_message = 'Redemption successful';
END proc_end;
END //

DELIMITER ;
```

- **Recharge Gift Card**

```
DELIMITER //

CREATE PROCEDURE recharge_gift_card (
   IN p_card_id CHAR(7),
   IN p_user_id INT,
   IN p_amount DECIMAL(10,2),
   OUT p_success BOOLEAN,
   OUT p_message VARCHAR(255)
)
BEGIN
```

```
   DECLARE v_status ENUM('active', 'inactive', 'blocked', 'expired');

 proc_end:Begin
   -- Check if card exists
   IF NOT EXISTS (SELECT 1 FROM gift_cards WHERE card_id = p_card_id) THEN
      SET p_success = FALSE;
      SET p_message = 'Gift card not found';
      LEAVE proc_end;
   END IF;

   SELECT status INTO v_status FROM gift_cards WHERE card_id = p_card_id;

   IF v_status != 'active' THEN
      SET p_success = FALSE;
      SET p_message = CONCAT('Gift card status is ', v_status);
      LEAVE proc_end;
   END IF;

   IF p_amount <= 0 THEN
      SET p_success = FALSE;
      SET p_message = 'Recharge amount must be positive';
      LEAVE proc_end;
   END IF;

   -- Update balance
   UPDATE gift_cards SET current_balance = current_balance + p_amount, updated_at = NOW()
   WHERE card_id = p_card_id;

   -- Log transaction
   INSERT INTO gift_card_transactions(card_id, user_id, transaction_type, amount, notes)
   VALUES (p_card_id, p_user_id, 'recharge', p_amount, 'Recharge');

   SET p_success = TRUE;
   SET p_message = 'Recharge successful';

   End proc_end;
END //

DELIMITER ;
```

- **Automatic expiry Handling**

```
DELIMITER //

CREATE PROCEDURE expire_gift_cards()
BEGIN
   UPDATE gift_cards
   SET status = 'expired', updated_at = NOW()
   WHERE expiration_date < CURDATE() AND status = 'active';
END //
DELIMITER ;
```

■ Redeem gift card & also Check for Fraud Detection

```
DELIMITER //

CREATE PROCEDURE redeem_gift_card (
   IN p_card_id CHAR(7),
   IN p_user_id INT,
   IN p_amount DECIMAL(10,2),
   OUT p_success BOOLEAN,
   OUT p_message VARCHAR(255)
)
BEGIN
   DECLARE v_balance DECIMAL(10,2);
   DECLARE v_status ENUM('active', 'inactive', 'blocked', 'expired');
   DECLARE v_expired BOOL;

   proc_end: BEGIN
      -- Card existence check
      IF NOT EXISTS (SELECT 1 FROM gift_cards WHERE card_id = p_card_id) THEN
         SET p_success = FALSE;
         SET p_message = 'Gift card not found';
         LEAVE proc_end;
      END IF;

      -- Fetch balance, status, and expiry status
      SELECT current_balance, status, (expiration_date < CURDATE())
      INTO v_balance, v_status, v_expired
      FROM gift_cards
      WHERE card_id = p_card_id;

      -- Update status if expired
      IF v_expired AND v_status != 'expired' THEN
         UPDATE gift_cards SET status = 'expired' WHERE card_id = p_card_id;
         SET p_success = FALSE;
         SET p_message = 'Gift card is expired';
         LEAVE proc_end;
      END IF;

      -- Check card status
      IF v_status != 'active' THEN
         SET p_success = FALSE;
         SET p_message = CONCAT('Gift card status is ', v_status);
         LEAVE proc_end;
      END IF;

      -- Balance check
      IF v_balance < p_amount THEN
         SET p_success = FALSE;
         SET p_message = 'Insufficient balance';
         LEAVE proc_end;
      END IF;
```

```
        -- Basic fraud detection
        IF p_amount > 1000 THEN
            SET p_success = FALSE;
            SET p_message = 'Transaction exceeds allowed limit';
            LEAVE proc_end;
        END IF;

        -- Update balance
        UPDATE gift_cards
        SET current_balance = current_balance - p_amount,
            updated_at = NOW()
        WHERE card_id = p_card_id;

        -- Log transaction
        INSERT INTO gift_card_transactions(card_id, user_id, transaction_type, amount, notes)
        VALUES (p_card_id, p_user_id, 'redemption', p_amount, 'Redemption');

        SET p_success = TRUE;
        SET p_message = 'Redemption successful';
    END proc_end;
END //

DELIMITER ;
```

■ Transfer Gift Card

```
DELIMITER //

CREATE PROCEDURE transfer_gift_card (
    IN p_card_id CHAR(7),
    IN p_from_user INT,
    IN p_to_user INT,
    OUT p_success BOOLEAN,
    OUT p_message VARCHAR(255)
)
BEGIN
    DECLARE v_current_user INT;
    DECLARE v_status ENUM('active', 'inactive', 'blocked', 'expired');

 proc_end:Begin
    -- Check if card exists and belongs to from_user
    SELECT user_id, status INTO v_current_user, v_status FROM gift_cards WHERE card_id =
p_card_id;

    IF v_current_user IS NULL THEN
        SET p_success = FALSE;
        SET p_message = 'Gift card is unassigned and cannot be transferred';
        LEAVE proc_end;
    END IF;
```

```
   IF v_current_user != p_from_user THEN
      SET p_success = FALSE;
      SET p_message = 'Gift card does not belong to the transferring user';
      LEAVE proc_end;
   END IF;

   IF v_status != 'active' THEN
      SET p_success = FALSE;
      SET p_message = CONCAT('Gift card status is ', v_status);
      LEAVE proc_end;
   END IF;

   -- Transfer card
   UPDATE gift_cards SET user_id = p_to_user, updated_at = NOW() WHERE card_id = p_card_id;

   -- Log transactions: transfer_out and transfer_in
   INSERT INTO gift_card_transactions(card_id, user_id, transaction_type, amount, notes)
   VALUES (p_card_id, p_from_user, 'transfer_out', 0, CONCAT('Transferred to user ', p_to_user));

   INSERT INTO gift_card_transactions(card_id, user_id, transaction_type, amount, notes)
   VALUES (p_card_id, p_to_user, 'transfer_in', 0, CONCAT('Received from user ', p_from_user));

   SET p_success = TRUE;
   SET p_message = 'Gift card transferred successfully';

   End proc_end;
END //

DELIMITER ;
```

■ Status Management (Active,Expired,Inactive,Blocked)

```
DELIMITER //

CREATE PROCEDURE update_gift_card_status (
   IN p_card_id CHAR(7),
   IN p_new_status ENUM('active', 'inactive', 'blocked', 'expired'),
   OUT p_success BOOLEAN,
   OUT p_message VARCHAR(255)
)
BEGIN
 proc_end:Begin
   IF NOT EXISTS (SELECT 1 FROM gift_cards WHERE card_id = p_card_id) THEN
      SET p_success = FALSE;
      SET p_message = 'Gift card not found';
      LEAVE proc_end;
   END IF;

   UPDATE gift_cards SET status = p_new_status, updated_at = NOW() WHERE card_id = p_card_id;
   SET p_success = TRUE;
   SET p_message = CONCAT('Gift card status updated to ', p_new_status);

   End proc_end;
END //

DELIMITER ;
```

■ Assign Gift Card to User

If we generate Bulk Cards then this procedure will help us to assign one of the card from Bulk Generated cards to the User.

```
DELIMITTER //

CREATE PROCEDURE assign_gift_card_to_user (
   IN p_card_id CHAR(7),
   IN p_user_id INT
)
BEGIN
   UPDATE gift_cards
   SET user_id = p_user_id
   WHERE card_id = p_card_id;
END //
DELIMITER ;
```

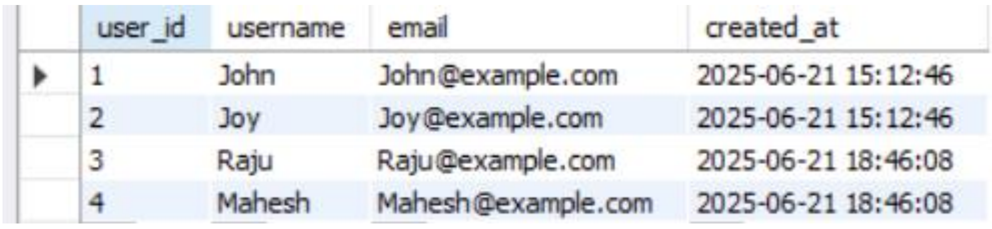
- **Reporting**

After Designing our database we will execute querries on it

To insert value into database:-

INSERT INTO users (username, email) VALUES ('John', 'John@example.com'),
('Joy', 'Joy@example.com'), ('Raju', 'Raju@example.com'),
('Mahesh', 'Mahesh@example.com');

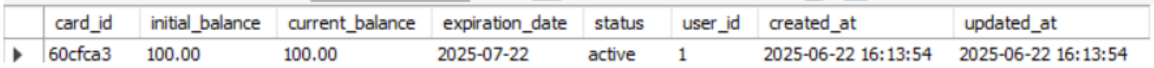
select * from users;



user_id	username	email	created_at
1	John	John@example.com	2025-06-21 15:12:46
2	Joy	Joy@example.com	2025-06-21 15:12:46
3	Raju	Raju@example.com	2025-06-21 18:46:08
4	Mahesh	Mahesh@example.com	2025-06-21 18:46:08

To generate a single gift card for specific user_id:-

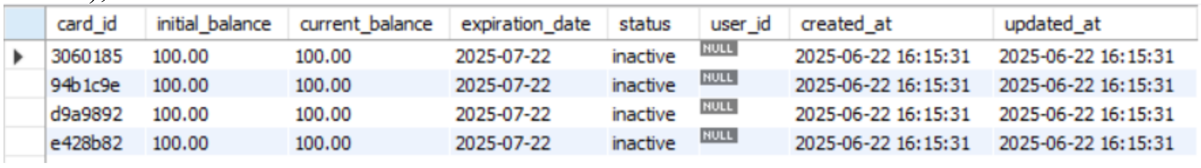
call generate_gift_card(100.00, CURDATE() + INTERVAL 30 DAY, 1, @card_id);



card_id	initial_balance	current_balance	expiration_date	status	user_id	created_at	updated_at
60cfca3	100.00	100.00	2025-07-22	active	1	2025-06-22 16:13:54	2025-06-22 16:13:54

To generate a bulk card without assign to a user_id:-

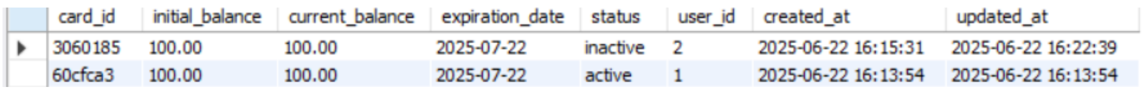
call bulk_generate_gift_cards(4, 100.00, DATE_ADD(CURDATE(), INTERVAL 30 DAY), NULL);



card_id	initial_balance	current_balance	expiration_date	status	user_id	created_at	updated_at
3060185	100.00	100.00	2025-07-22	inactive	NULL	2025-06-22 16:15:31	2025-06-22 16:15:31
94b1c9e	100.00	100.00	2025-07-22	inactive	NULL	2025-06-22 16:15:31	2025-06-22 16:15:31
d9a9892	100.00	100.00	2025-07-22	inactive	NULL	2025-06-22 16:15:31	2025-06-22 16:15:31
e428b82	100.00	100.00	2025-07-22	inactive	NULL	2025-06-22 16:15:31	2025-06-22 16:15:31

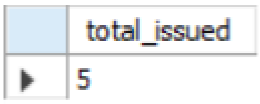
To assign a specific card_id to a user_id:-

call assign_gift_card_to_user('3060185', 2);-- This querry is applied for user_id 2



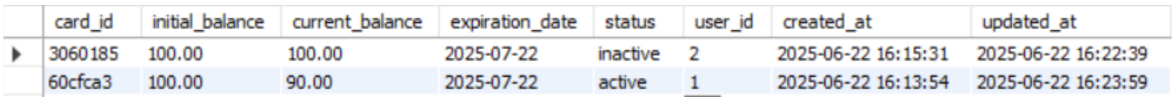
card_id	initial_balance	current_balance	expiration_date	status	user_id	created_at	updated_at
3060185	100.00	100.00	2025-07-22	inactive	2	2025-06-22 16:15:31	2025-06-22 16:22:39
60cfca3	100.00	100.00	2025-07-22	active	1	2025-06-22 16:13:54	2025-06-22 16:13:54

To check total issued-cards:-
select COUNT(*) AS total_issued FROM gift_cards;



total_issued
5

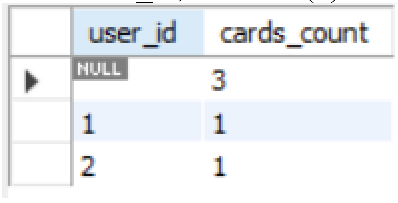
To redeem card:-
call redeem_gift_card('60cfca3',1,10,@success,@msg); -- this querry is applied for user id 1



card_id	initial_balance	current_balance	expiration_date	status	user_id	created_at	updated_at
3060185	100.00	100.00	2025-07-22	inactive	2	2025-06-22 16:15:31	2025-06-22 16:22:39
60cfca3	100.00	90.00	2025-07-22	active	1	2025-06-22 16:13:54	2025-06-22 16:23:59

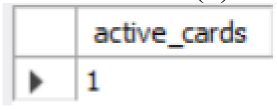
To check how many cards are generated and how many cards are assign to the user:-
-- First row is NULL of user_id because we generated a bulk cards without assign any user_id to it. The below screenshot means we have generated total 5 cards 2 cards assigned to the specific user_id and 3 remaining to assign.

select user_id, COUNT(*) AS cards_count FROM gift_cards GROUP BY user_id;



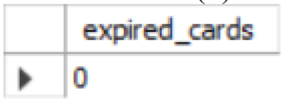
user_id	cards_count
NULL	3
1	1
2	1

To check number of active_cards:
select COUNT(*) AS active_cards FROM gift_cards WHERE status = 'active';



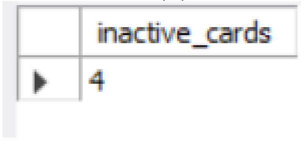
active_cards
1

To check expired_cards:-
select COUNT(*) AS expired_cards FROM gift_cards WHERE status = 'expired';



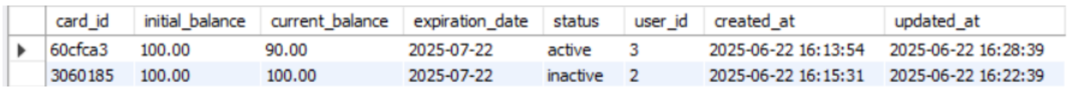
expired_cards
0

To check inactive_cards:-
select count(*) as inactive_cards from gift_cards where status = 'inactive';



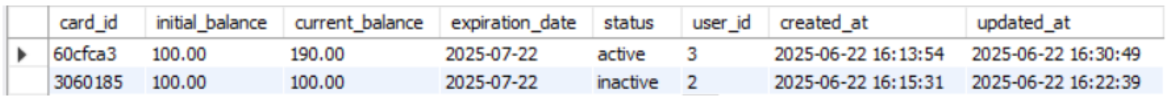
inactive_cards
4

To transfer one card_id to another user_id:
call transfer_gift_card('60cfca3', 1, 3, @success, @msg);-- Here card of user_id 1 is transfer to user_id 3



card_id	initial_balance	current_balance	expiration_date	status	user_id	created_at	updated_at
60cfca3	100.00	90.00	2025-07-22	active	3	2025-06-22 16:13:54	2025-06-22 16:28:39
3060185	100.00	100.00	2025-07-22	inactive	2	2025-06-22 16:15:31	2025-06-22 16:22:39

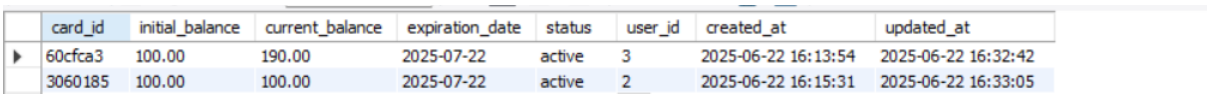
To recharge gift card:
call recharge_gift_card('60cfca3', 3, 100.00, @success, @message);



card_id	initial_balance	current_balance	expiration_date	status	user_id	created_at	updated_at
60cfca3	100.00	190.00	2025-07-22	active	3	2025-06-22 16:13:54	2025-06-22 16:30:49
3060185	100.00	100.00	2025-07-22	inactive	2	2025-06-22 16:15:31	2025-06-22 16:22:39

To update status of gift_card(blocked,inactive,active,expired):

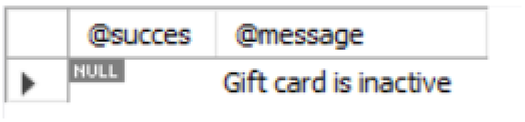
call update_gift_card_status('3060185','active',@success,@msg);



card_id	initial_balance	current_balance	expiration_date	status	user_id	created_at	updated_at
60cfca3	100.00	190.00	2025-07-22	active	3	2025-06-22 16:13:54	2025-06-22 16:32:42
3060185	100.00	100.00	2025-07-22	active	2	2025-06-22 16:15:31	2025-06-22 16:33:05

To recharge card like block,inactive,expired:
-- If we try to recharge card like (blocked,expired,inactive) then it will not work it will give message like. Here first i change status to inactive then i run this querry.

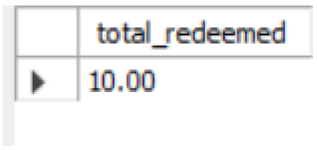
call recharge_gift_card('3060185', 2, 100.00, @success, @message);
select @succes,@message; -- After running this it will show message.



@succes	@message
NULL	Gift card is inactive

To check total redeem:

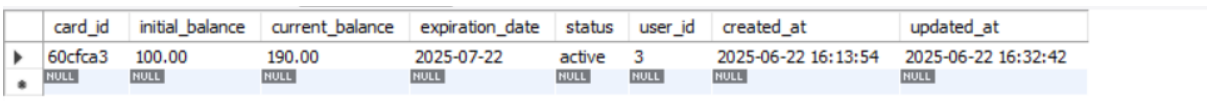
select IFNULL(SUM(amount),0) AS total_redeemed FROM gift_card_transactions WHERE transaction_type = 'redemption';



total_redeemed
10.00

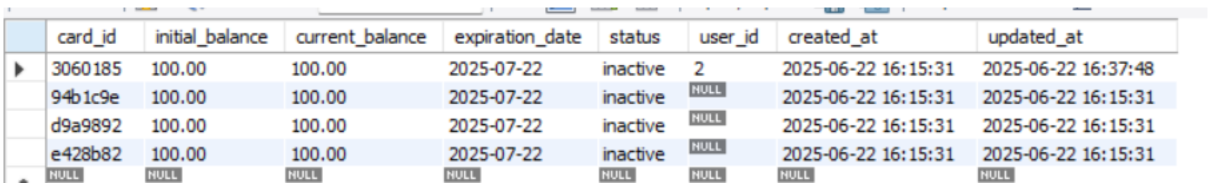
To check current_balance:

select * from gift_cards where current_balance > 100;



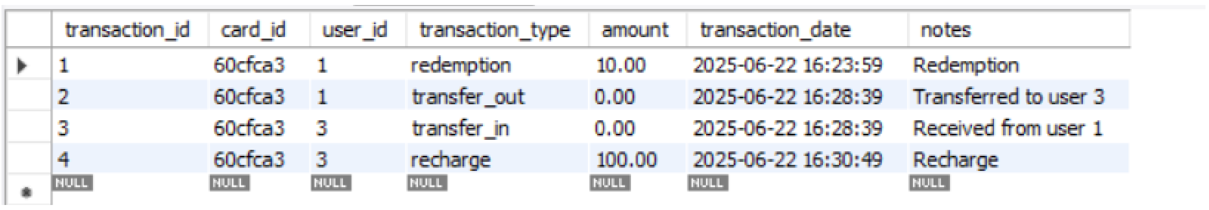
card_id	initial_balance	current_balance	expiration_date	status	user_id	created_at	updated_at
60cfca3	100.00	190.00	2025-07-22	active	3	2025-06-22 16:13:54	2025-06-22 16:32:42
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

select * from gift_cards where current_balance = 100;



card_id	initial_balance	current_balance	expiration_date	status	user_id	created_at	updated_at
3060185	100.00	100.00	2025-07-22	inactive	2	2025-06-22 16:15:31	2025-06-22 16:37:48
94b1c9e	100.00	100.00	2025-07-22	inactive	NULL	2025-06-22 16:15:31	2025-06-22 16:15:31
d9a9892	100.00	100.00	2025-07-22	inactive	NULL	2025-06-22 16:15:31	2025-06-22 16:15:31
e428b82	100.00	100.00	2025-07-22	inactive	NULL	2025-06-22 16:15:31	2025-06-22 16:15:31
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

To check gift_card_transactions:-
select * from gift_card_transactions;



transaction_id	card_id	user_id	transaction_type	amount	transaction_date	notes
1	60cfca3	1	redemption	10.00	2025-06-22 16:23:59	Redemption
2	60cfca3	1	transfer_out	0.00	2025-06-22 16:28:39	Transferred to user 3
3	60cfca3	3	transfer_in	0.00	2025-06-22 16:28:39	Received from user 1
4	60cfca3	3	recharge	100.00	2025-06-22 16:30:49	Recharge
NULL	NULL	NULL	NULL	NULL	NULL	NULL