DBT MINI PROJECT ON LIBRARY MANAGEMENT SYSTEM (Assignment)

NAME: Abhishek Narayan Jagtap , Kapil Umakant Katte

PRN: 250840320005, 250840320085

EMAIL: abhishek.jagtap.cmaug25@gmail.com , kapil.katte.cmaug25@gmail.com

DBT Topic Name: Library Management System

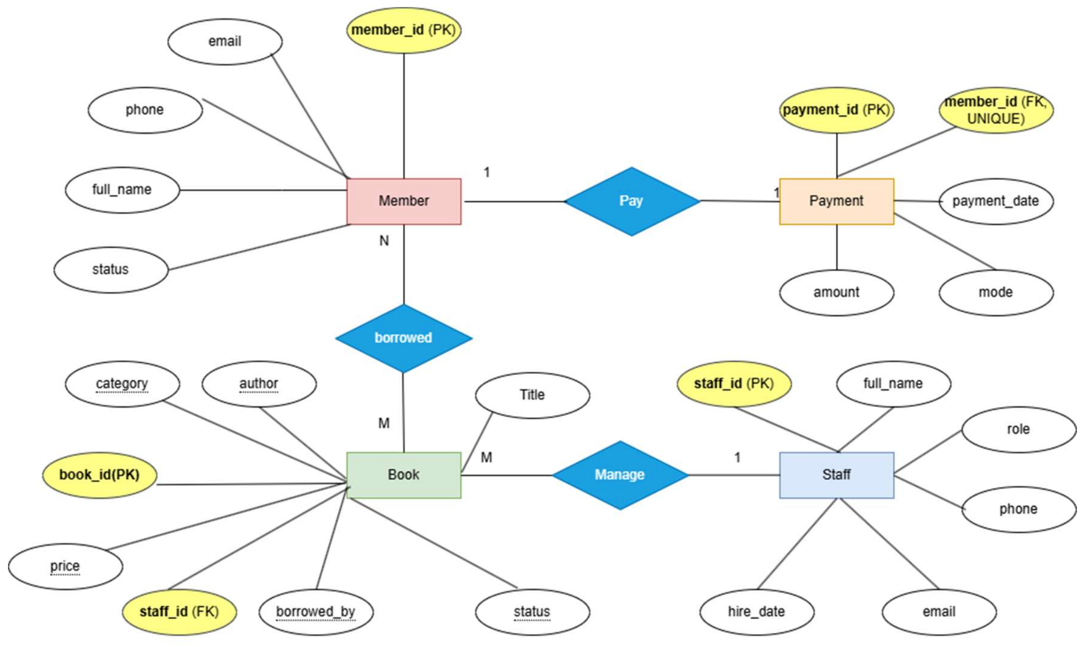
DBT Project Team No: Team No 5

---

1) Draw an ER Diagram in draw.io showing entities, attributes, and relationships.

- Identify all major entities, their attributes, and primary keys.
- Show relationships (1–M, M–N, 1–1) with clear cardinalities.
- Include associative entities wherever M:N relationships exist.
- Indicate foreign keys and participation constraints clearly.

E – R DIAGRAM :

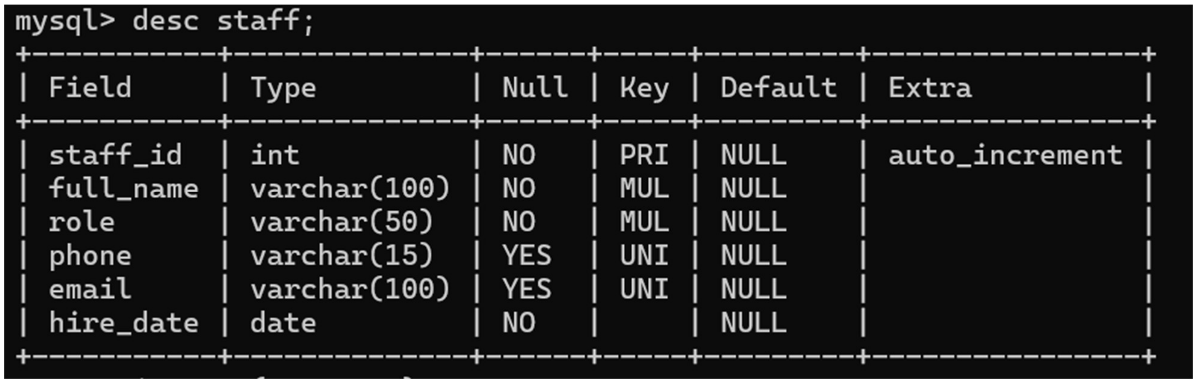


Library Management System

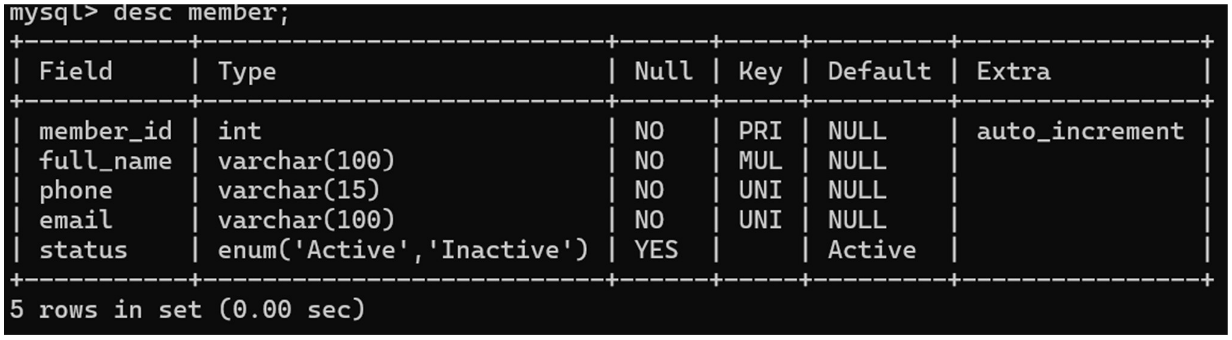
2) Create the database schema (DDL) with all required constraints and relationships.

- Appropriate data types and size definitions.
- Primary Keys and Foreign Keys for relationships.
- Unique, Check, and Not Null constraints.
- Use ENUM or SET data types where suitable (e.g., gender, status).
- Create indexes on key searchable fields.

```
CREATE TABLE Staff (
  staff_id INT AUTO_INCREMENT PRIMARY KEY,
  full_name VARCHAR(100) NOT NULL,
  role VARCHAR(50) NOT NULL,
  phone VARCHAR(15) UNIQUE,
  email VARCHAR(100) UNIQUE,
  hire_date DATE NOT NULL
  INDEX idx_staff_name (full_name),
  INDEX idx_staff_role (role)
);
```

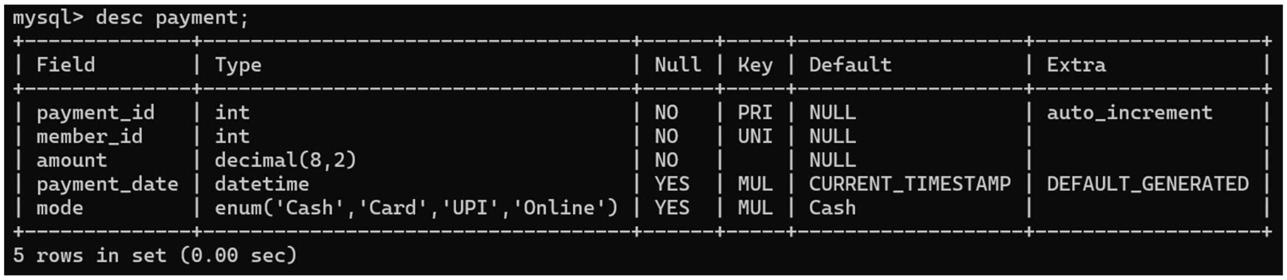


```
mysql> desc staff;
+-----------+--------------+------+-----+---------+----------------+
| Field     | Type         | Null | Key | Default | Extra          |
+-----------+--------------+------+-----+---------+----------------+
| staff_id  | int          | NO   | PRI | NULL    | auto_increment |
| full_name | varchar(100) | NO   | MUL | NULL    |                |
| role      | varchar(50)  | NO   | MUL | NULL    |                |
| phone     | varchar(15)  | YES  | UNI | NULL    |                |
| email     | varchar(100) | YES  | UNI | NULL    |                |
| hire_date | date         | NO   |     | NULL    |                |
+-----------+--------------+------+-----+---------+----------------+
CREATE TABLE Member (
  member_id INT AUTO_INCREMENT PRIMARY KEY,
  full_name VARCHAR(100) NOT NULL,
  phone VARCHAR(15) UNIQUE NOT NULL,
  email VARCHAR(100) UNIQUE NOT NULL,
  status ENUM('Active','Inactive') DEFAULT 'Active',
  INDEX idx_member_name (full_name),
  INDEX idx_member_email (email)
);
```



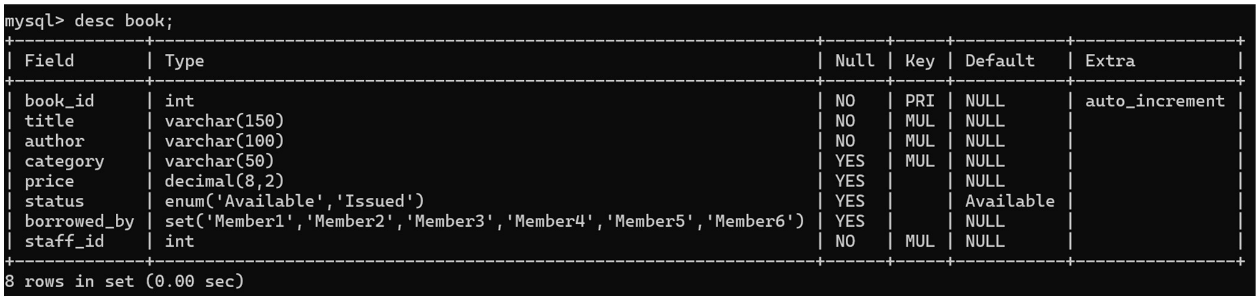
```
mysql> desc member;
+-----------+--------------------------+------+-----+---------+----------------+
| Field     | Type                     | Null | Key | Default | Extra          |
+-----------+--------------------------+------+-----+---------+----------------+
| member_id | int                      | NO   | PRI | NULL    | auto_increment |
| full_name | varchar(100)             | NO   | MUL | NULL    |                |
| phone     | varchar(15)              | NO   | UNI | NULL    |                |
| email     | varchar(100)             | NO   | UNI | NULL    |                |
| status    | enum('Active','Inactive') | YES  |     | Active  |                |
+-----------+--------------------------+------+-----+---------+----------------+
5 rows in set (0.00 sec)
```

```
CREATE TABLE Payment (
 payment_id INT AUTO_INCREMENT PRIMARY KEY,
 member_id INT NOT NULL UNIQUE,
 amount DECIMAL(8,2) NOT NULL CHECK (amount >= 0),
 payment_date DATETIME DEFAULT CURRENT_TIMESTAMP,
 mode ENUM('Cash','Card','UPI','Online') DEFAULT 'Cash',

 FOREIGN KEY (member_id) REFERENCES Member(member_id)
  ON DELETE CASCADE ON UPDATE CASCADE,

 INDEX idx_payment_mode (mode),
 INDEX idx_payment_date (payment_date)
);
```



```
mysql> desc payment;
+--------------+-----------------------------------+------+-----+-------------------+-------------------+
| Field        | Type                              | Null | Key | Default           | Extra             |
+--------------+-----------------------------------+------+-----+-------------------+-------------------+
| payment_id   | int                               | NO   | PRI | NULL              | auto_increment    |
| member_id    | int                               | NO   | UNI | NULL              |                   |
| amount       | decimal(8,2)                      | NO   |     | NULL              |                   |
| payment_date | datetime                          | YES  | MUL | CURRENT_TIMESTAMP | DEFAULT_GENERATED |
| mode         | enum('Cash','Card','UPI','Online') | YES  | MUL | Cash              |                   |
+--------------+-----------------------------------+------+-----+-------------------+-------------------+
5 rows in set (0.00 sec)
```

```
CREATE TABLE Book (
  book_id INT AUTO_INCREMENT PRIMARY KEY,
  title VARCHAR(150) NOT NULL,
  author VARCHAR(100) NOT NULL,
  category VARCHAR(50),
  price DECIMAL(8,2) CHECK (price >= 0),
  status ENUM('Available','Issued') DEFAULT 'Available',
  borrowed_by SET('Member1','Member2','Member3','Member4','Member5'),
  staff_id INT NOT NULL,

  FOREIGN KEY (staff_id) REFERENCES Staff(staff_id)
    ON DELETE RESTRICT ON UPDATE CASCADE,
  INDEX idx_book_title (title),
  INDEX idx_book_author (author),
  INDEX idx_book_category (category)
);
```



```
mysql> desc book;
+-------------+-----------------------------------------------------------------+------+-----+-----------+----------------+
| Field       | Type                                                            | Null | Key | Default   | Extra          |
+-------------+-----------------------------------------------------------------+------+-----+-----------+----------------+
| book_id     | int                                                             | NO   | PRI | NULL      | auto_increment |
| title       | varchar(150)                                                    | NO   | MUL | NULL      |                |
| author      | varchar(100)                                                    | NO   | MUL | NULL      |                |
| category    | varchar(50)                                                     | YES  | MUL | NULL      |                |
| price       | decimal(8,2)                                                    | YES  |     | NULL      |                |
| status      | enum('Available','Issued')                                      | YES  |     | Available |                |
| borrowed_by | set('Member1','Member2','Member3','Member4','Member5','Member6') | YES  |     | NULL      |                |
| staff_id    | int                                                             | NO   | MUL | NULL      |                |
+-------------+-----------------------------------------------------------------+------+-----+-----------+----------------+
8 rows in set (0.00 sec)
```

3) Perform DML operations (Insert, Update, Delete) to populate sample data.

- Insert at least 5–10 records in each main table.
- Update some attribute (e.g., change contact info, modify price, update status).
- Delete one or more records safely (with WHERE condition).

INSERT STATEMENT:

INSERT INTO Staff (full_name, role, phone, email, hire_date) VALUES

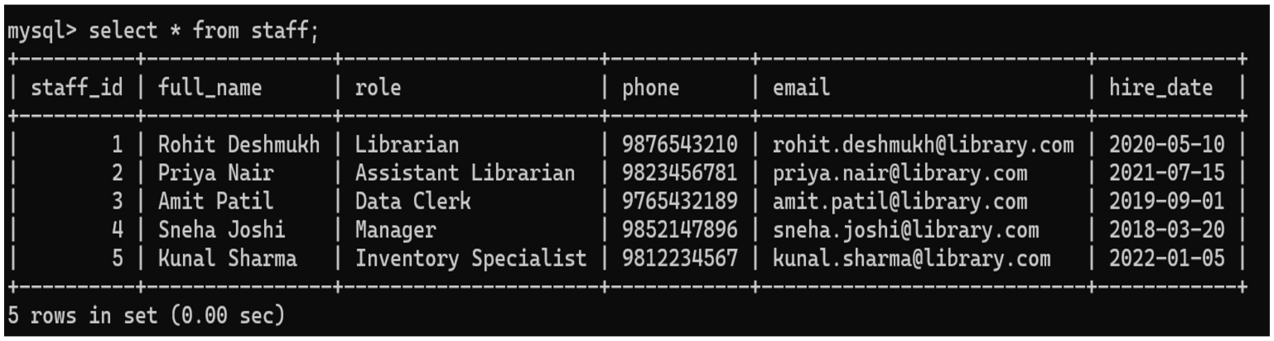
('Rohit Deshmukh', 'Librarian', '9876543210', 'rohit.deshmukh@library.com', '2020-05-10'),

('Priya Nair', 'Assistant Librarian', '9823456781', 'priya.nair@library.com', '2021-07-15'),

('Amit Patil', 'Data Clerk', '9765432189', 'amit.patil@library.com', '2019-09-01'),

('Sneha Joshi', 'Manager', '9852147896', 'sneha.joshi@library.com', '2018-03-20'),

('Kunal Sharma', 'Inventory Specialist', '9812234567', 'kunal.sharma@library.com', '2022-01-05');


```
mysql> select * from staff;
+----------+----------------+----------------------+------------+----------------------------+------------+
| staff_id | full_name      | role                 | phone      | email                      | hire_date  |
+----------+----------------+----------------------+------------+----------------------------+------------+
|        1 | Rohit Deshmukh | Librarian            | 9876543210 | rohit.deshmukh@library.com | 2020-05-10 |
|        2 | Priya Nair     | Assistant Librarian  | 9823456781 | priya.nair@library.com     | 2021-07-15 |
|        3 | Amit Patil     | Data Clerk           | 9765432189 | amit.patil@library.com     | 2019-09-01 |
|        4 | Sneha Joshi    | Manager              | 9852147896 | sneha.joshi@library.com    | 2018-03-20 |
|        5 | Kunal Sharma   | Inventory Specialist | 9812234567 | kunal.sharma@library.com   | 2022-01-05 |
+----------+----------------+----------------------+------------+----------------------------+------------+
5 rows in set (0.00 sec)
```

INSERT INTO Member (full_name, phone, email, status) VALUES

('Kapil Katte', '9865321470', 'kapil.katte@example.com', 'Active'),

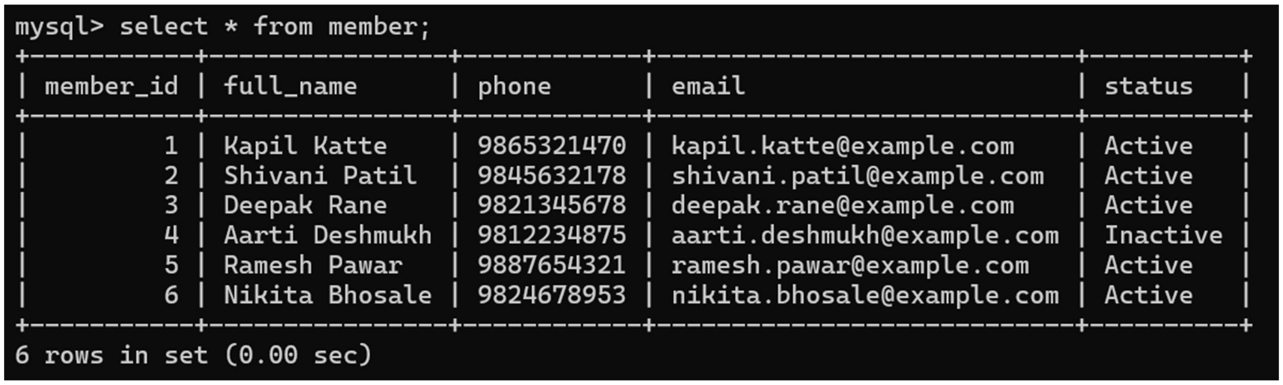
('Shivani Patil', '9845632178', 'shivani.patil@example.com', 'Active'),

('Deepak Rane', '9821345678', 'deepak.rane@example.com', 'Active'),

('Aarti Deshmukh', '9812234875', 'aarti.deshmukh@example.com', 'Inactive'),

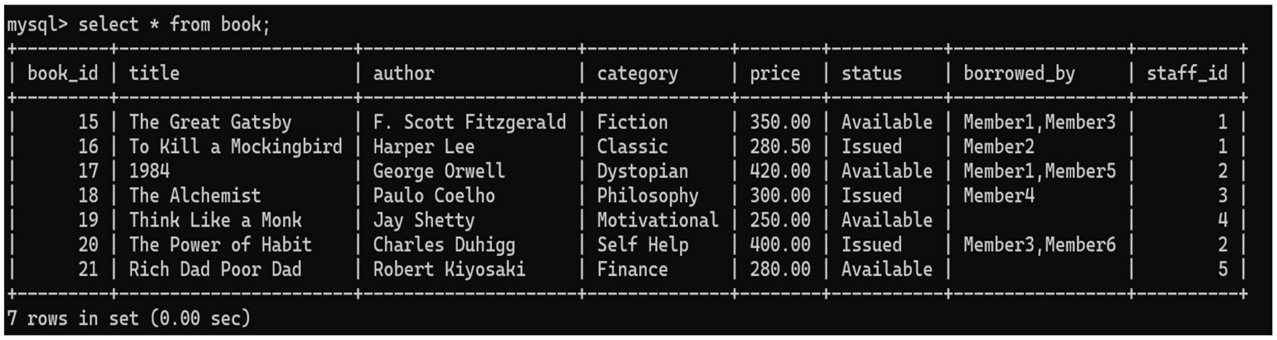
('Ramesh Pawar', '9887654321', 'ramesh.pawar@example.com', 'Active'),

('Nikita Bhosale', '9824678953', 'nikita.bhosale@example.com', 'Active');



```
mysql> select * from member;
+-----------+----------------+------------+-----------------------------+----------+
| member_id | full_name      | phone      | email                       | status   |
+-----------+----------------+------------+-----------------------------+----------+
|         1 | Kapil Katte    | 9865321470 | kapil.katte@example.com     | Active   |
|         2 | Shivani Patil  | 9845632178 | shivani.patil@example.com   | Active   |
|         3 | Deepak Rane    | 9821345678 | deepak.rane@example.com     | Active   |
|         4 | Aarti Deshmukh | 9812234875 | aarti.deshmukh@example.com  | Inactive |
|         5 | Ramesh Pawar   | 9887654321 | ramesh.pawar@example.com    | Active   |
|         6 | Nikita Bhosale | 9824678953 | nikita.bhosale@example.com  | Active   |
+-----------+----------------+------------+-----------------------------+----------+
6 rows in set (0.00 sec)
```

INSERT INTO Book (title, author, category, price, status, borrowed_by, staff_id) VALUES
  ('The Great Gatsby', 'F. Scott Fitzgerald', 'Fiction', 350.00, 'Available',
'Member1,Member3', 1),
  ('To Kill a Mockingbird', 'Harper Lee', 'Classic', 280.50, 'Issued', 'Member2', 1),
  ('1984', 'George Orwell', 'Dystopian', 420.00, 'Available', 'Member1,Member5', 2),
  ('The Alchemist', 'Paulo Coelho', 'Philosophy', 300.00, 'Issued', 'Member4', 3),
  ('Think Like a Monk', 'Jay Shetty', 'Motivational', 250.00, 'Available', '', 4),
  ('The Power of Habit', 'Charles Duhigg', 'Self Help', 400.00, 'Issued',
'Member3,Member6', 2),
  ('Rich Dad Poor Dad', 'Robert Kiyosaki', 'Finance', 280.00, 'Available', '', 5);



```
mysql> select * from book;
+---------+-----------------------+---------------------+--------------+--------+-----------+-----------------+----------+
| book_id | title                 | author              | category     | price  | status    | borrowed_by     | staff_id |
+---------+-----------------------+---------------------+--------------+--------+-----------+-----------------+----------+
|      15 | The Great Gatsby      | F. Scott Fitzgerald | Fiction      | 350.00 | Available | Member1,Member3 |        1 |
|      16 | To Kill a Mockingbird | Harper Lee          | Classic      | 280.50 | Issued    | Member2         |        1 |
|      17 | 1984                  | George Orwell       | Dystopian    | 420.00 | Available | Member1,Member5 |        2 |
|      18 | The Alchemist         | Paulo Coelho        | Philosophy   | 300.00 | Issued    | Member4         |        3 |
|      19 | Think Like a Monk     | Jay Shetty          | Motivational | 250.00 | Available |                 |        4 |
|      20 | The Power of Habit    | Charles Duhigg      | Self Help    | 400.00 | Issued    | Member3,Member6 |        2 |
|      21 | Rich Dad Poor Dad     | Robert Kiyosaki     | Finance      | 280.00 | Available |                 |        5 |
+---------+-----------------------+---------------------+--------------+--------+-----------+-----------------+----------+
7 rows in set (0.00 sec)
```

INSERT INTO Payment (member_id, amount, payment_date, mode) VALUES

(1, 500.00, '2024-12-15 10:30:00', 'UPI'),

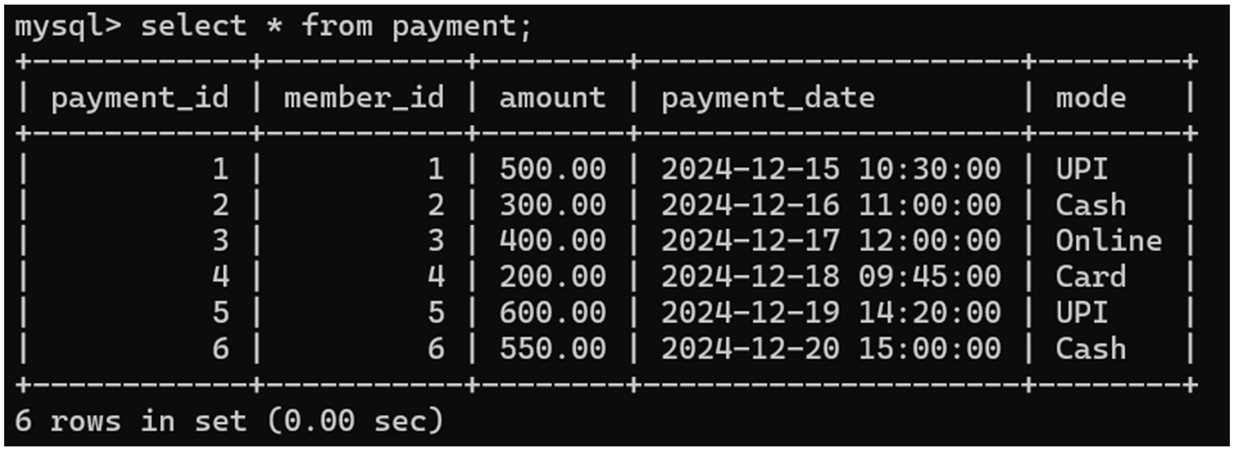
(2, 300.00, '2024-12-16 11:00:00', 'Cash'),

(3, 400.00, '2024-12-17 12:00:00', 'Online'),

(4, 200.00, '2024-12-18 09:45:00', 'Card'),

(5, 600.00, '2024-12-19 14:20:00', 'UPI'),

(6, 550.00, '2024-12-20 15:00:00', 'Cash');



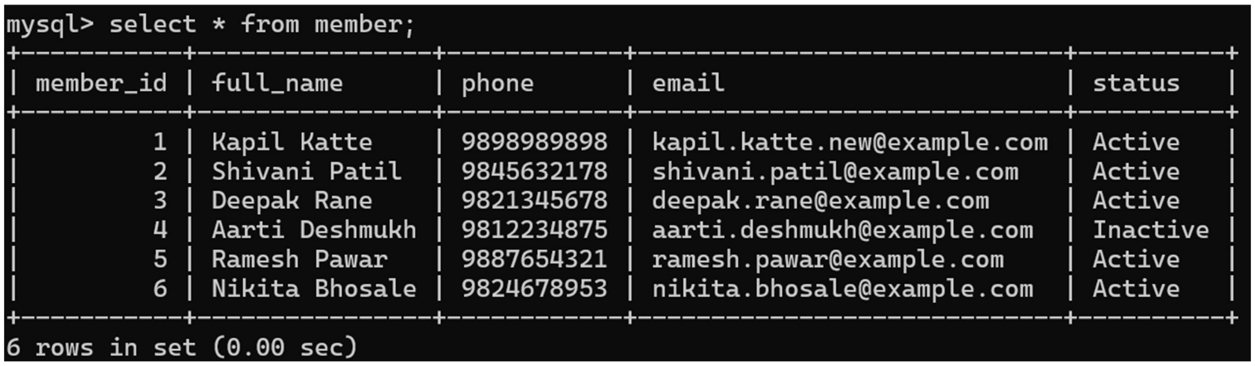
```
mysql> select * from payment;
+------------+-----------+--------+---------------------+--------+
| payment_id | member_id | amount | payment_date        | mode   |
+------------+-----------+--------+---------------------+--------+
|          1 |         1 | 500.00 | 2024-12-15 10:30:00 | UPI    |
|          2 |         2 | 300.00 | 2024-12-16 11:00:00 | Cash   |
|          3 |         3 | 400.00 | 2024-12-17 12:00:00 | Online |
|          4 |         4 | 200.00 | 2024-12-18 09:45:00 | Card   |
|          5 |         5 | 600.00 | 2024-12-19 14:20:00 | UPI    |
|          6 |         6 | 550.00 | 2024-12-20 15:00:00 | Cash   |
+------------+-----------+--------+---------------------+--------+
6 rows in set (0.00 sec)
```

UPDATE STATEMENTS :

UPDATE Member

SET phone = '9898989898', email = 'kapil.katte.new@example.com'

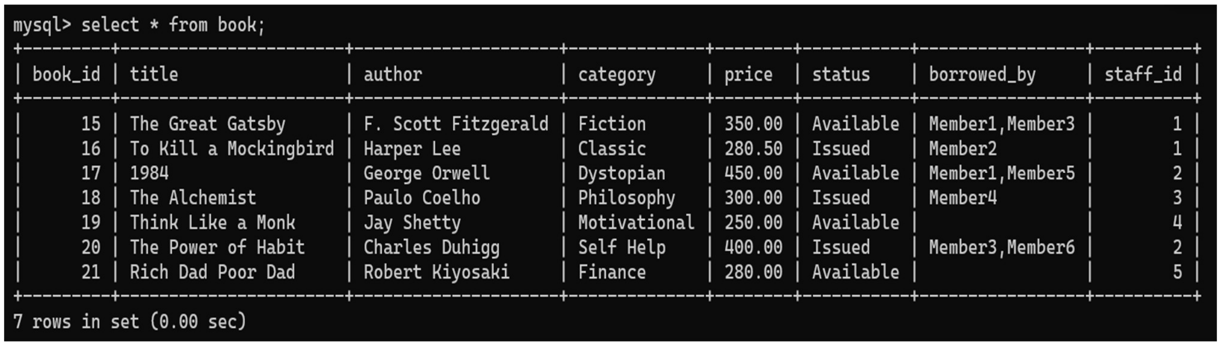
WHERE full_name = 'Kapil Katte';



```
mysql> select * from member;
+-----------+----------------+------------+-----------------------------+----------+
| member_id | full_name      | phone      | email                       | status   |
+-----------+----------------+------------+-----------------------------+----------+
|         1 | Kapil Katte    | 9898989898 | kapil.katte.new@example.com | Active   |
|         2 | Shivani Patil  | 9845632178 | shivani.patil@example.com   | Active   |
|         3 | Deepak Rane    | 9821345678 | deepak.rane@example.com     | Active   |
|         4 | Aarti Deshmukh | 9812234875 | aarti.deshmukh@example.com  | Inactive |
|         5 | Ramesh Pawar   | 9887654321 | ramesh.pawar@example.com    | Active   |
|         6 | Nikita Bhosale | 9824678953 | nikita.bhosale@example.com  | Active   |
+-----------+----------------+------------+-----------------------------+----------+
6 rows in set (0.00 sec)
```

UPDATE Book

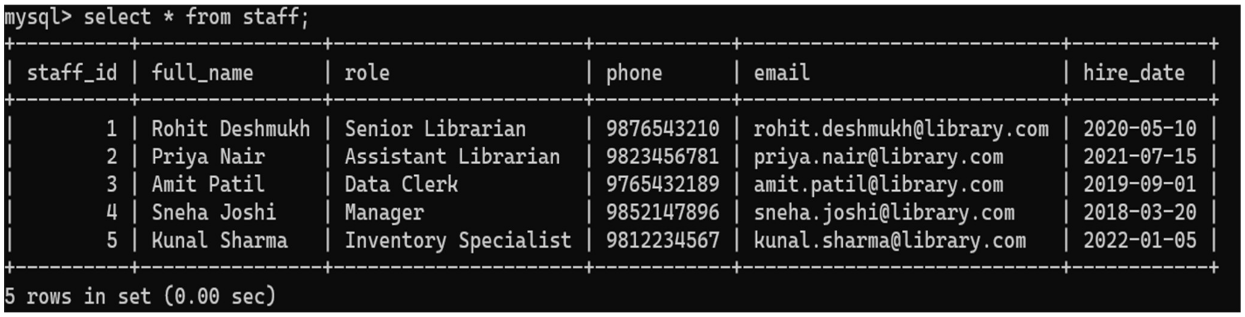
SET price = 450.00 WHERE title = '1984';



```
mysql> select * from book;
+---------+-----------------------+----------------------+--------------+--------+-----------+-----------------+----------+
| book_id | title                 | author               | category     | price  | status    | borrowed_by     | staff_id |
+---------+-----------------------+----------------------+--------------+--------+-----------+-----------------+----------+
|      15 | The Great Gatsby      | F. Scott Fitzgerald  | Fiction      | 350.00 | Available | Member1,Member3 |        1 |
|      16 | To Kill a Mockingbird | Harper Lee           | Classic      | 280.50 | Issued    | Member2         |        1 |
|      17 | 1984                  | George Orwell        | Dystopian    | 450.00 | Available | Member1,Member5 |        2 |
|      18 | The Alchemist         | Paulo Coelho         | Philosophy   | 300.00 | Issued    | Member4         |        3 |
|      19 | Think Like a Monk     | Jay Shetty           | Motivational | 250.00 | Available |                 |        4 |
|      20 | The Power of Habit    | Charles Duhigg       | Self Help    | 400.00 | Issued    | Member3,Member6 |        2 |
|      21 | Rich Dad Poor Dad     | Robert Kiyosaki      | Finance      | 280.00 | Available |                 |        5 |
+---------+-----------------------+----------------------+--------------+--------+-----------+-----------------+----------+
7 rows in set (0.00 sec)
```

UPDATE Staff

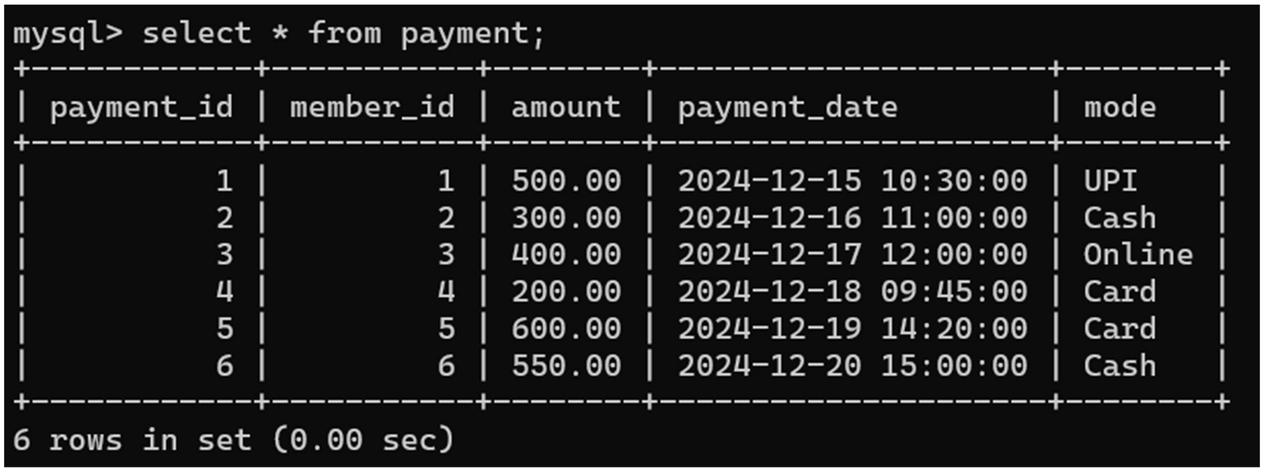
SET role = 'Senior Librarian' WHERE full_name = 'Rohit Deshmukh';



```
mysql> select * from staff;
+----------+----------------+---------------------+------------+----------------------------+------------+
| staff_id | full_name      | role                | phone      | email                      | hire_date  |
+----------+----------------+---------------------+------------+----------------------------+------------+
|        1 | Rohit Deshmukh | Senior Librarian    | 9876543210 | rohit.deshmukh@library.com | 2020-05-10 |
|        2 | Priya Nair     | Assistant Librarian | 9823456781 | priya.nair@library.com     | 2021-07-15 |
|        3 | Amit Patil     | Data Clerk          | 9765432189 | amit.patil@library.com     | 2019-09-01 |
|        4 | Sneha Joshi    | Manager             | 9852147896 | sneha.joshi@library.com    | 2018-03-20 |
|        5 | Kunal Sharma   | Inventory Specialist | 9812234567 | kunal.sharma@library.com   | 2022-01-05 |
+----------+----------------+---------------------+------------+----------------------------+------------+
5 rows in set (0.00 sec)
```

UPDATE Payment

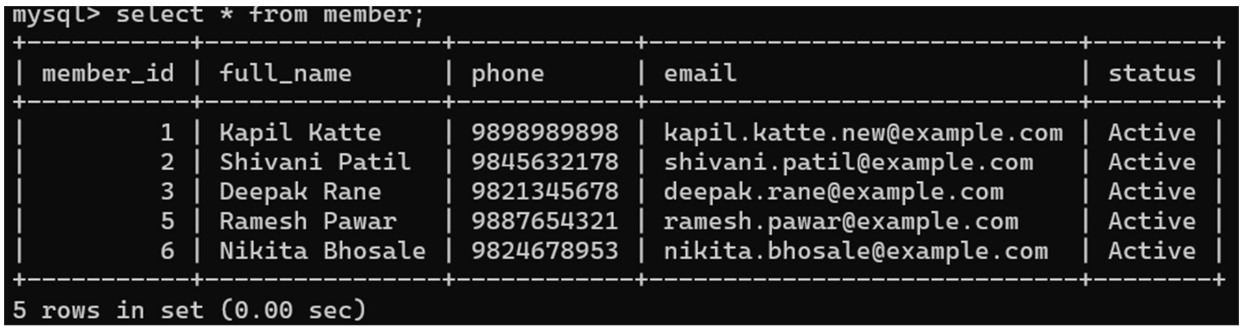
SET mode = 'Card' WHERE member_id = 5;



```
mysql> select * from payment;
+------------+-----------+--------+---------------------+--------+
| payment_id | member_id | amount | payment_date        | mode   |
+------------+-----------+--------+---------------------+--------+
|          1 |         1 | 500.00 | 2024-12-15 10:30:00 | UPI    |
|          2 |         2 | 300.00 | 2024-12-16 11:00:00 | Cash   |
|          3 |         3 | 400.00 | 2024-12-17 12:00:00 | Online |
|          4 |         4 | 200.00 | 2024-12-18 09:45:00 | Card   |
|          5 |         5 | 600.00 | 2024-12-19 14:20:00 | Card   |
|          6 |         6 | 550.00 | 2024-12-20 15:00:00 | Cash   |
+------------+-----------+--------+---------------------+--------+
6 rows in set (0.00 sec)
```

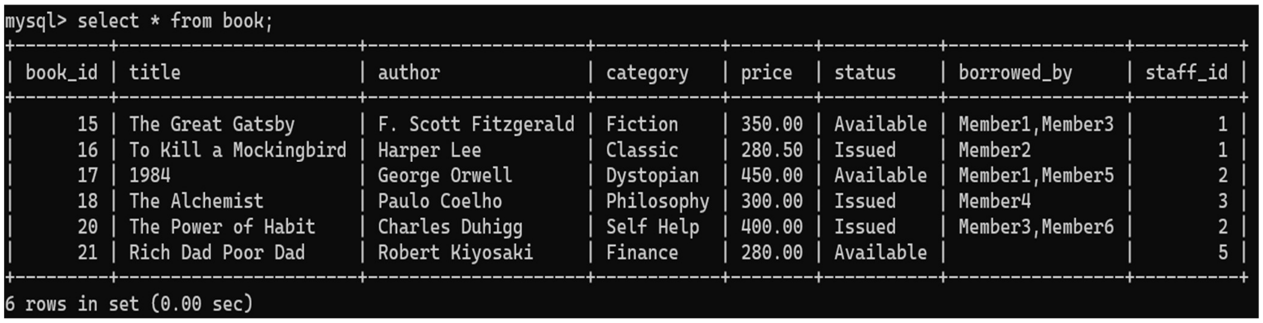
DELETE STATEMENTS:

DELETE FROM Member WHERE status = 'Inactive';



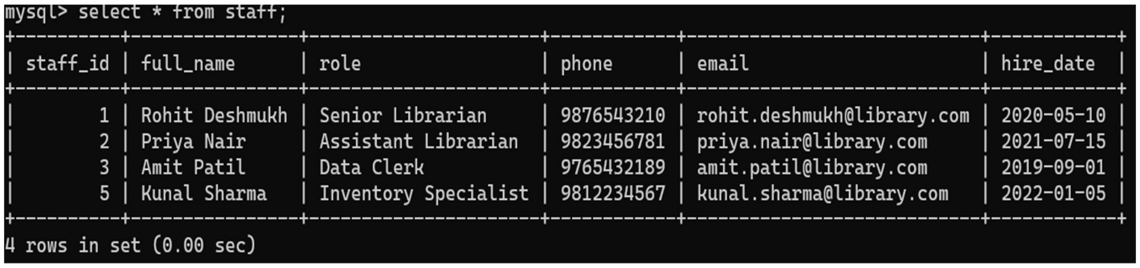
```
mysql> select * from member;
+-----------+-----------------+------------+------------------------------+--------+
| member_id | full_name       | phone      | email                        | status |
+-----------+-----------------+------------+------------------------------+--------+
|         1 | Kapil Katte     | 9898989898 | kapil.katte.new@example.com  | Active |
|         2 | Shivani Patil   | 9845632178 | shivani.patil@example.com    | Active |
|         3 | Deepak Rane     | 9821345678 | deepak.rane@example.com      | Active |
|         5 | Ramesh Pawar    | 9887654321 | ramesh.pawar@example.com     | Active |
|         6 | Nikita Bhosale  | 9824678953 | nikita.bhosale@example.com   | Active |
+-----------+-----------------+------------+------------------------------+--------+
5 rows in set (0.00 sec)
```

DELETE FROM Book WHERE price < 260.00;



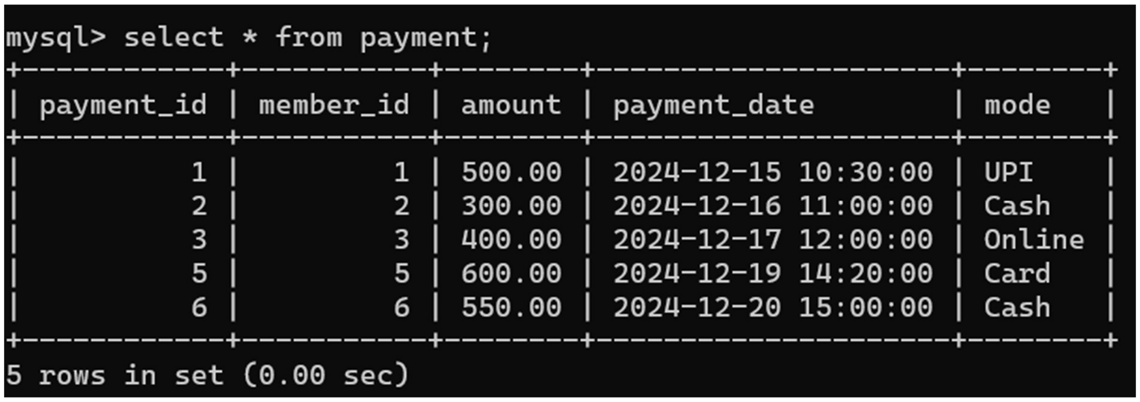
```
mysql> select * from book;
+---------+-----------------------+---------------------+------------+--------+-----------+-----------------+----------+
| book_id | title                 | author              | category   | price  | status    | borrowed_by     | staff_id |
+---------+-----------------------+---------------------+------------+--------+-----------+-----------------+----------+
|      15 | The Great Gatsby      | F. Scott Fitzgerald | Fiction    | 350.00 | Available | Member1,Member3 |        1 |
|      16 | To Kill a Mockingbird | Harper Lee          | Classic    | 280.50 | Issued    | Member2         |        1 |
|      17 | 1984                  | George Orwell       | Dystopian  | 450.00 | Available | Member1,Member5 |        2 |
|      18 | The Alchemist         | Paulo Coelho        | Philosophy | 300.00 | Issued    | Member4         |        3 |
|      20 | The Power of Habit    | Charles Duhigg      | Self Help  | 400.00 | Issued    | Member3,Member6 |        2 |
|      21 | Rich Dad Poor Dad     | Robert Kiyosaki     | Finance    | 280.00 | Available |                 |        5 |
+---------+-----------------------+---------------------+------------+--------+-----------+-----------------+----------+
6 rows in set (0.00 sec)
```

DELETE FROM Staff WHERE hire_date < '2019-01-01';



```
mysql> select * from staff;
+----------+----------------+----------------------+------------+----------------------------+------------+
| staff_id | full_name      | role                 | phone      | email                      | hire_date  |
+----------+----------------+----------------------+------------+----------------------------+------------+
|        1 | Rohit Deshmukh | Senior Librarian     | 9876543210 | rohit.deshmukh@library.com | 2020-05-10 |
|        2 | Priya Nair     | Assistant Librarian  | 9823456781 | priya.nair@library.com     | 2021-07-15 |
|        3 | Amit Patil     | Data Clerk           | 9765432189 | amit.patil@library.com     | 2019-09-01 |
|        5 | Kunal Sharma   | Inventory Specialist | 9812234567 | kunal.sharma@library.com   | 2022-01-05 |
+----------+----------------+----------------------+------------+----------------------------+------------+
4 rows in set (0.00 sec)
```

DELETE FROM Payment WHERE member_id = 4;

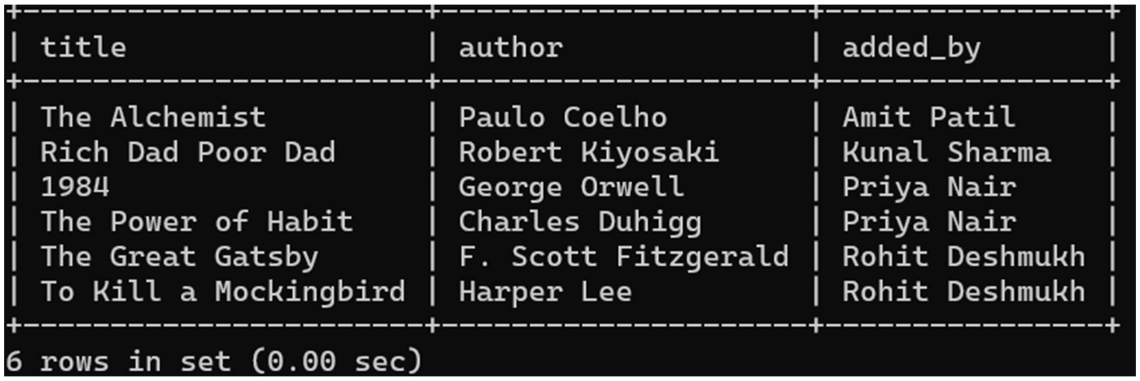


```
mysql> select * from payment;
+------------+-----------+--------+---------------------+--------+
| payment_id | member_id | amount | payment_date        | mode   |
+------------+-----------+--------+---------------------+--------+
|          1 |         1 | 500.00 | 2024-12-15 10:30:00 | UPI    |
|          2 |         2 | 300.00 | 2024-12-16 11:00:00 | Cash   |
|          3 |         3 | 400.00 | 2024-12-17 12:00:00 | Online |
|          5 |         5 | 600.00 | 2024-12-19 14:20:00 | Card   |
|          6 |         6 | 550.00 | 2024-12-20 15:00:00 | Cash   |
+------------+-----------+--------+---------------------+--------+
5 rows in set (0.00 sec)
```

4) Write SQL Queries using Joins, Aggregate functions, Grouping, and Subqueries to retrieve meaningful information.

JOIN EXAMPLE :

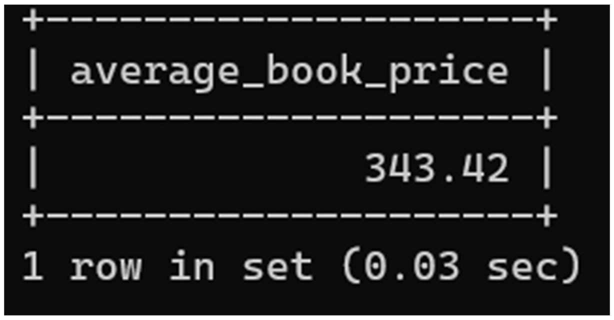
```
SELECT
   b.title,
   b.author,
   s.full_name AS added_by
FROM
   Book b
JOIN
   Staff s ON b.staff_id = s.staff_id;
```



```
+------------------------+---------------------+-----------------+
| title                  | author              | added_by        |
+------------------------+---------------------+-----------------+
| The Alchemist          | Paulo Coelho        | Amit Patil      |
| Rich Dad Poor Dad      | Robert Kiyosaki     | Kunal Sharma    |
| 1984                   | George Orwell       | Priya Nair      |
| The Power of Habit     | Charles Duhigg      | Priya Nair      |
| The Great Gatsby       | F. Scott Fitzgerald | Rohit Deshmukh  |
| To Kill a Mockingbird  | Harper Lee          | Rohit Deshmukh  |
+------------------------+---------------------+-----------------+
6 rows in set (0.00 sec)
```

AGGREGATE FUNCATION EXAMPLE :

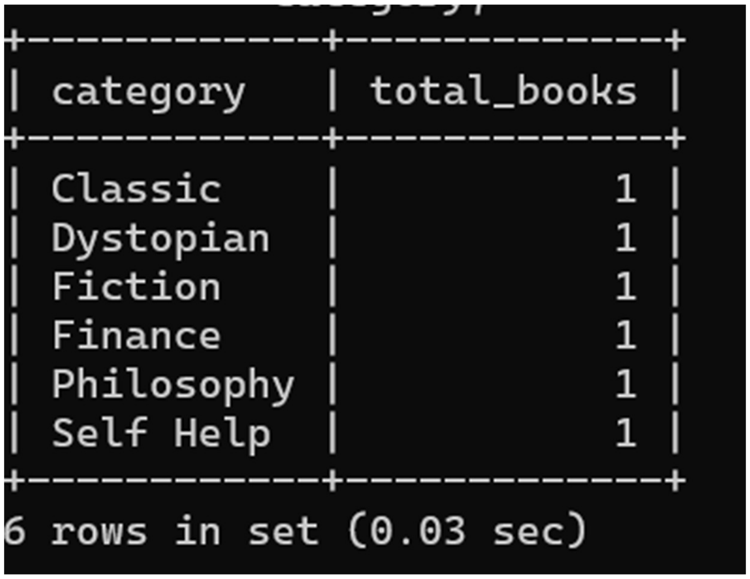
```
SELECT
   ROUND(AVG(price), 2) AS average_book_price
FROM
   Book;
```



```
+--------------------+
| average_book_price |
+--------------------+
|             343.42 |
+--------------------+
1 row in set (0.03 sec)
```

GROUPING EXAMPLE:

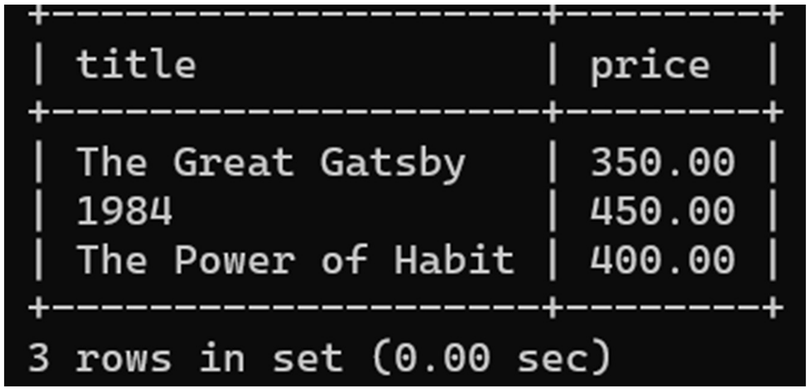
```
SELECT
   category,
   COUNT(*) AS total_books
FROM
   Book
GROUP BY
   category;
```



```
+------------+-------------+
| category   | total_books |
+------------+-------------+
| Classic    |           1 |
| Dystopian  |           1 |
| Fiction    |           1 |
| Finance    |           1 |
| Philosophy |           1 |
| Self Help  |           1 |
+------------+-------------+
6 rows in set (0.03 sec)
```

SUBQUERIES EXAMPLE:

```
SELECT
  title,
  price
FROM
  Book
WHERE
  price > (
    SELECT AVG(price) FROM Book
  );
```

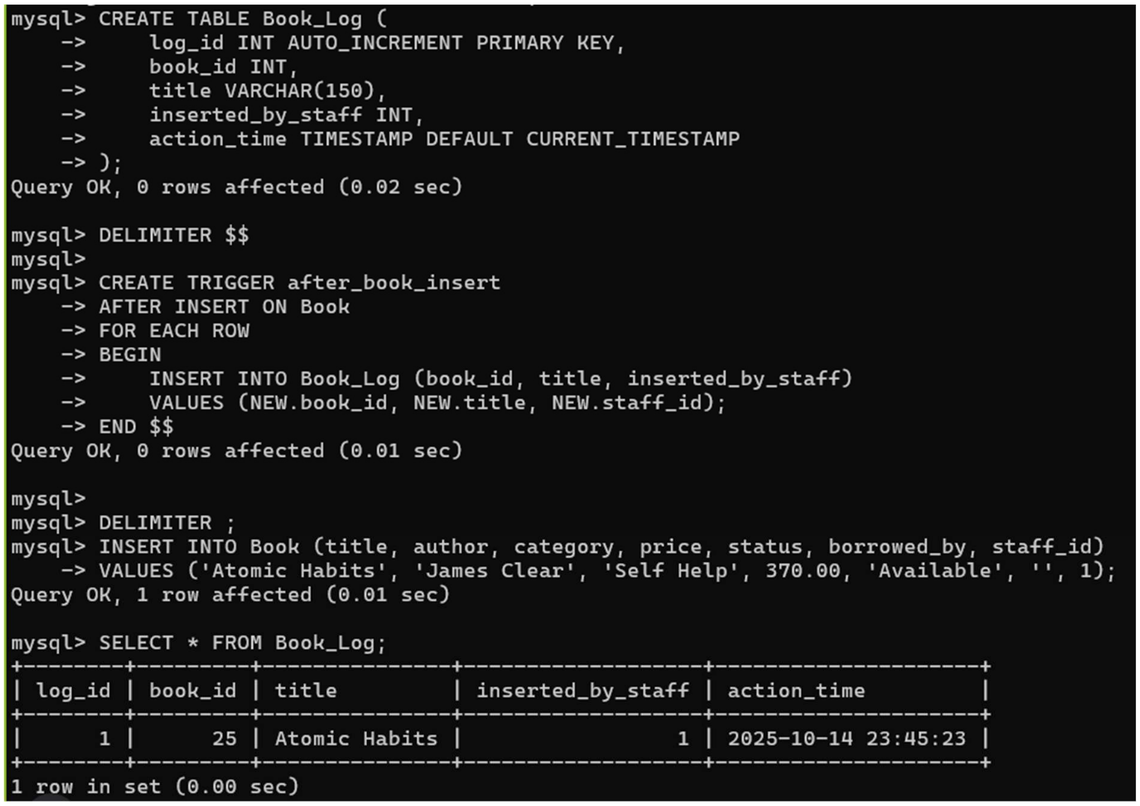


```
+--------------------+--------+
| title              | price  |
+--------------------+--------+
| The Great Gatsby   | 350.00 |
| 1984               | 450.00 |
| The Power of Habit | 400.00 |
+--------------------+--------+
3 rows in set (0.00 sec)
```

5) Implement a Trigger, a Function, and a Stored Procedure relevant to your system's logic.

TRIGGER :

```
CREATE TABLE Book_Log (
  log_id INT AUTO_INCREMENT PRIMARY KEY,
  book_id INT,
  title VARCHAR(150),
  inserted_by_staff INT,
  action_time TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
DELIMITER $$
CREATE TRIGGER after_book_insert
AFTER INSERT ON Book
FOR EACH ROW
BEGIN
  INSERT INTO Book_Log (book_id, title, inserted_by_staff)
  VALUES (NEW.book_id, NEW.title, NEW.staff_id);
END $$
DELIMITER ;

SELECT * FROM  Book_Log
```



```
mysql> CREATE TABLE Book_Log (
    ->     log_id INT AUTO_INCREMENT PRIMARY KEY,
    ->     book_id INT,
    ->     title VARCHAR(150),
    ->     inserted_by_staff INT,
    ->     action_time TIMESTAMP DEFAULT CURRENT_TIMESTAMP
    -> );
Query OK, 0 rows affected (0.02 sec)

mysql> DELIMITER $$
mysql>
mysql> CREATE TRIGGER after_book_insert
    -> AFTER INSERT ON Book
    -> FOR EACH ROW
    -> BEGIN
    ->     INSERT INTO Book_Log (book_id, title, inserted_by_staff)
    ->     VALUES (NEW.book_id, NEW.title, NEW.staff_id);
    -> END $$
Query OK, 0 rows affected (0.01 sec)

mysql>
mysql> DELIMITER ;
mysql> INSERT INTO Book (title, author, category, price, status, borrowed_by, staff_id)
    -> VALUES ('Atomic Habits', 'James Clear', 'Self Help', 370.00, 'Available', '', 1);
Query OK, 1 row affected (0.01 sec)

mysql> SELECT * FROM Book_Log;
+--------+---------+---------------+-------------------+---------------------+
| log_id | book_id | title         | inserted_by_staff | action_time         |
+--------+---------+---------------+-------------------+---------------------+
|      1 |      25 | Atomic Habits |                 1 | 2025-10-14 23:45:23 |
+--------+---------+---------------+-------------------+---------------------+
1 row in set (0.00 sec)
```

FUNCTION :

DELIMITER $$

CREATE FUNCTION GetTotalPayments(memberID INT)

RETURNS DECIMAL(10,2)

DETERMINISTIC

BEGIN

  DECLARE total DECIMAL(10,2);

  SELECT SUM(amount) INTO total

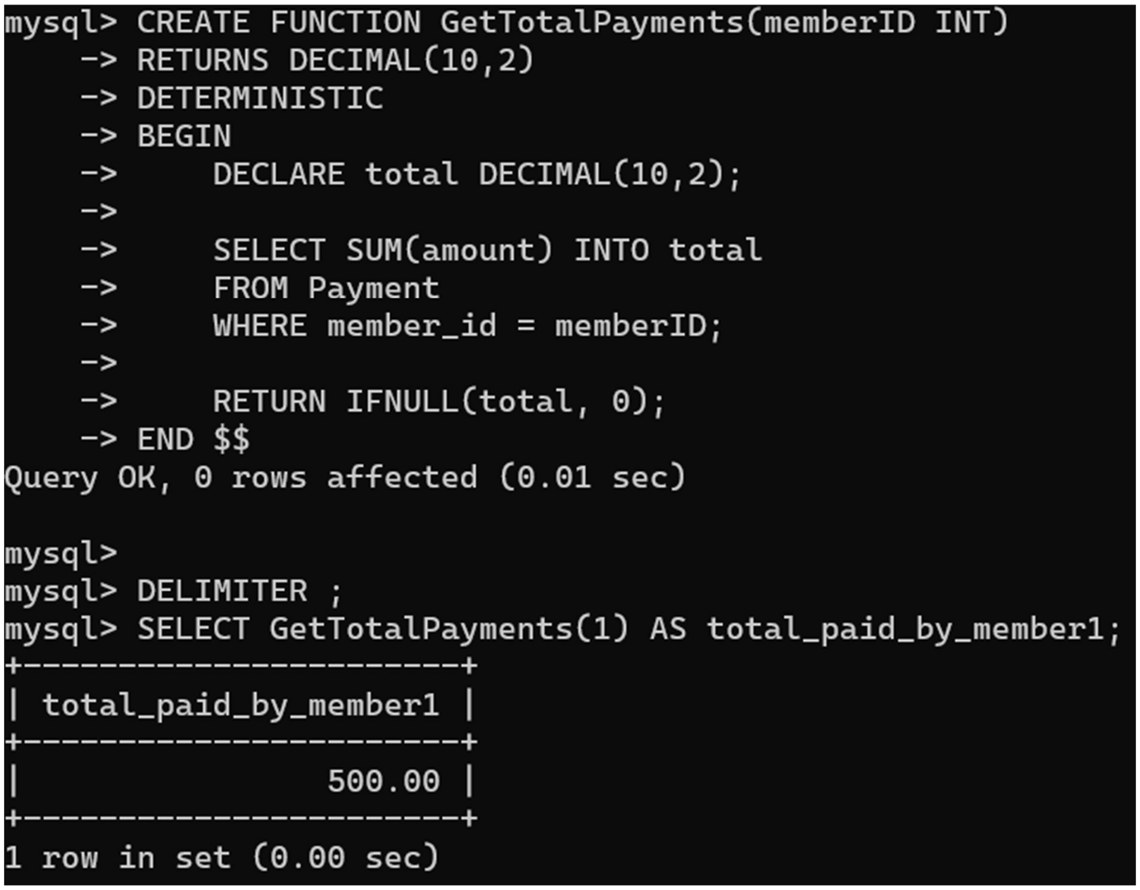
  FROM Payment

  WHERE member_id = memberID;

    RETURN IFNULL(total, 0);

END $$

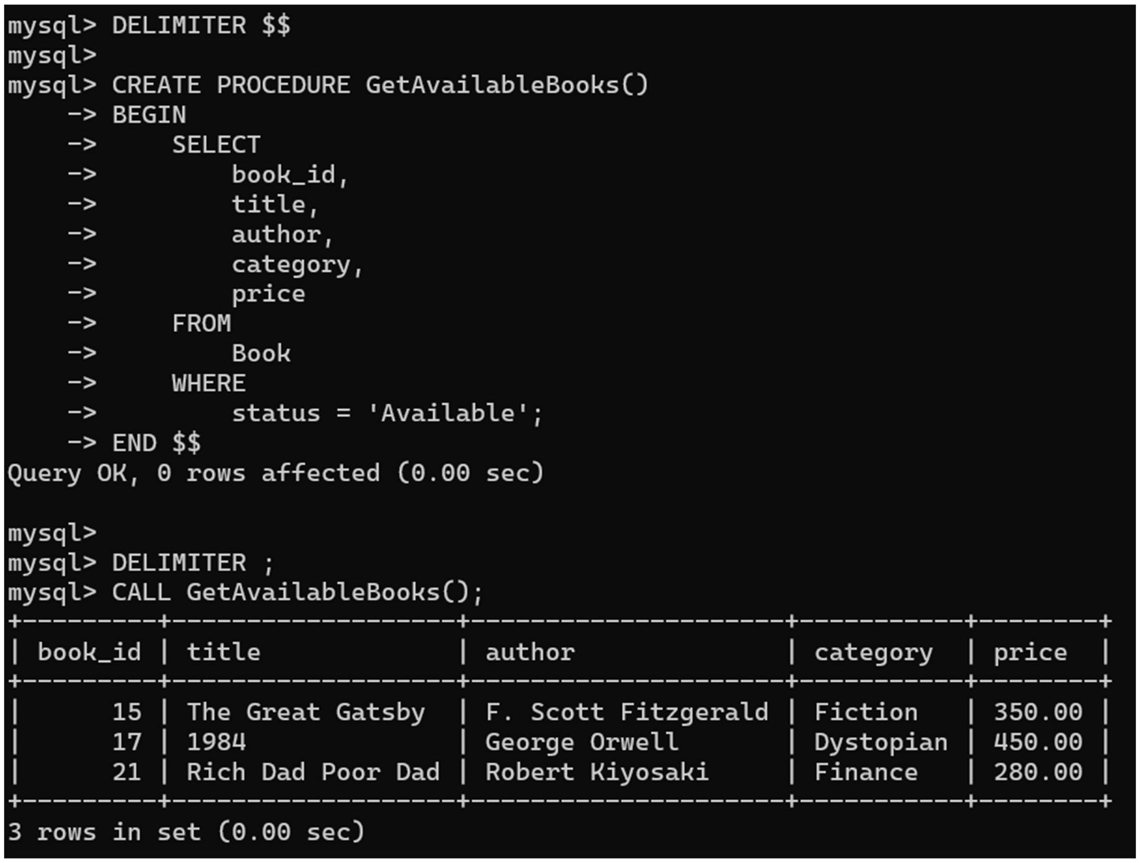
DELIMITER ;



```
mysql> CREATE FUNCTION GetTotalPayments(memberID INT)
    -> RETURNS DECIMAL(10,2)
    -> DETERMINISTIC
    -> BEGIN
    ->     DECLARE total DECIMAL(10,2);
    ->
    ->     SELECT SUM(amount) INTO total
    ->     FROM Payment
    ->     WHERE member_id = memberID;
    ->
    ->     RETURN IFNULL(total, 0);
    -> END $$
Query OK, 0 rows affected (0.01 sec)

mysql>
mysql> DELIMITER ;
mysql> SELECT GetTotalPayments(1) AS total_paid_by_member1;
+-----------------------+
| total_paid_by_member1 |
+-----------------------+
|                500.00 |
+-----------------------+
1 row in set (0.00 sec)
```

STORED PROCEDURE :

```
DELIMITER $$

CREATE PROCEDURE GetAvailableBooks()
BEGIN
  SELECT
    book_id,
    title,
    author,
    category,
    price
  FROM
    Book
  WHERE
    status = 'Available';
END $$

DELIMITER ;
```



```
mysql> DELIMITER $$
mysql>
mysql> CREATE PROCEDURE GetAvailableBooks()
    -> BEGIN
    ->     SELECT
    ->         book_id,
    ->         title,
    ->         author,
    ->         category,
    ->         price
    ->     FROM
    ->         Book
    ->     WHERE
    ->         status = 'Available';
    -> END $$
Query OK, 0 rows affected (0.00 sec)

mysql>
mysql> DELIMITER ;
mysql> CALL GetAvailableBooks();
+---------+-------------------+---------------------+-----------+--------+
| book_id | title             | author              | category  | price  |
+---------+-------------------+---------------------+-----------+--------+
|      15 | The Great Gatsby  | F. Scott Fitzgerald | Fiction   | 350.00 |
|      17 | 1984              | George Orwell       | Dystopian | 450.00 |
|      21 | Rich Dad Poor Dad | Robert Kiyosaki     | Finance   | 280.00 |
+---------+-------------------+---------------------+-----------+--------+
3 rows in set (0.00 sec)
```

6) Normalize your database up to Third Normal Form (3NF) and provide a short explanation.

- Identify repeating groups → convert to 1NF.
- Remove partial dependencies → convert to 2NF.
- Remove transitive dependencies → convert to 3NF.
- Clearly show the final normalized tables.
- Explain each step briefly

1NF (First Normal Form)

Goal: Eliminate repeating groups and ensure atomicity.

Action:

In the Book table, the borrowed_by column contains multiple member names (e.g., “Member1, Member3”), which is a repeating group.

Split this data into a separate table — Book_Borrower — so that each record represents one book borrowed by one member.

Remove borrowed_by from Book.

Create a new table: Book_Borrower(book_id, member_id).

2NF (Second Normal Form)

Goal: Remove partial dependencies (non-key attributes depending on only part of a composite key).

Action:

Each table should have a single-column primary key (no composite key).

Book_Borrower will have a composite key (book_id, member_id) — both are needed to uniquely identify a record, so no partial dependency exists.

Other tables like Book, Member, Staff, and Payment already have unique primary keys (book_id, member_id, staff_id, payment_id), so they are already in 2NF.

No partial dependency remains.

3NF (Third Normal Form)

Goal: Remove transitive dependencies — non-key attributes should depend only on the primary key.

Action:

Ensure each non-key field depends only on its table’s primary key.

Example:

In Book, the staff_id determines staff details — but that data (name, email, etc.) is stored separately in Staff, not in Book.

In Payment, member_id determines member info, which correctly resides in the Member table.

No transitive dependencies exist — every non-key attribute depends directly on its primary key.

Final Normalized Tables (Up to 3NF)

Book

(book_id, title, author, category, price, status, staff_id)

Member

(member_id, full_name, phone, email, status)

Staff

(staff_id, full_name, role, phone, email, hire_date)

Payment

(payment_id, member_id, amount, payment_date, mode)

Book_Borrower

(book_id, member_id) → Represents which member borrowed which book