

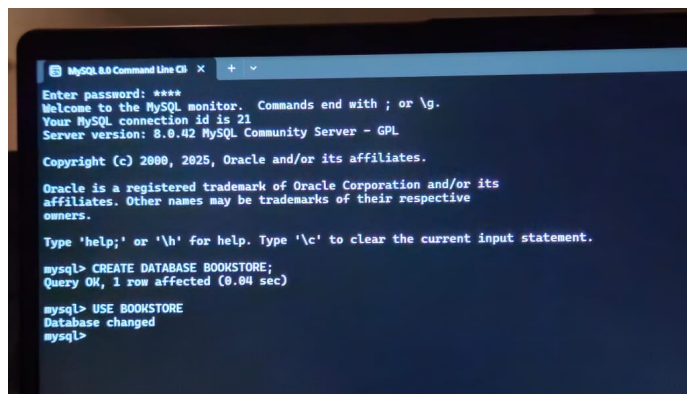
# SQL ASSIGNMENT

Shah Manan

28 August 2025

## Problem 1: ONLINE BOOKSTORE MANAGEMENT

{STEP 0 : CREATE DATABASE AND USE DATABASE}



```
MySQL 8.0 Command Line Client
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 21
Server version: 8.0.42 MySQL Community Server - GPL

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

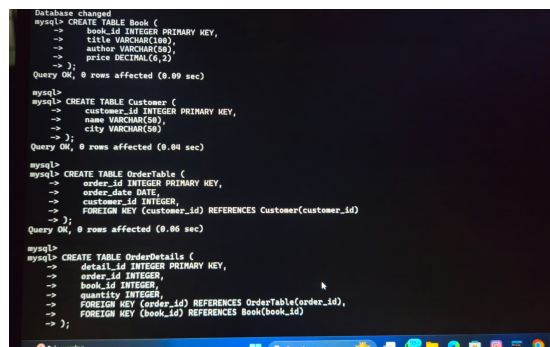
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE DATABASE BOOKSTORE;
Query OK, 1 row affected (0.04 sec)

mysql> USE BOOKSTORE
Database changed
mysql>
```

{step 1 : CREATE TABLES}



```
Database changed
mysql> CREATE TABLE Book (
-> book_id INTEGER PRIMARY KEY,
-> title VARCHAR(100),
-> author VARCHAR(50),
-> price DECIMAL(6,2)
-> );
Query OK, 0 rows affected (0.09 sec)

mysql> CREATE TABLE Customer (
-> customer_id INTEGER PRIMARY KEY,
-> name VARCHAR(50),
-> city VARCHAR(50)
-> );
Query OK, 0 rows affected (0.04 sec)

mysql> CREATE TABLE OrderTable (
-> order_id INTEGER PRIMARY KEY,
-> order_date DATE,
-> customer_id INTEGER,
-> FOREIGN KEY (customer_id) REFERENCES Customer(customer_id)
-> );
Query OK, 0 rows affected (0.06 sec)

mysql> CREATE TABLE OrderDetails (
-> detail_id INTEGER PRIMARY KEY,
-> order_id INTEGER,
-> book_id INTEGER,
-> quantity INTEGER,
-> FOREIGN KEY (order_id) REFERENCES OrderTable(order_id),
-> FOREIGN KEY (book_id) REFERENCES Book(book_id)
-> );
```

{STEP 2 : INSERT DATA}

```
mysql> INSERT INTO Book VALUES (1, 'SQL Guide', 'Jane Doe', 100.00);
-- (1, 'SQL Guide', 'Jane Doe', 100.00)
mysql> INSERT INTO Book VALUES (2, 'Python 101', 'Chetan Bhagat', 400.50);
-- (2, 'Python 101', 'Chetan Bhagat', 400.50)
mysql> INSERT INTO Book VALUES (3, 'Learn Java', 'Unknown', 287.50);
-- (3, 'Learn Java', 'Unknown', 287.50)
mysql> INSERT INTO Book VALUES (4, 'Advanced Guide to AI', 'Ada Lovelace', 573.85);
-- (4, 'Advanced Guide to AI', 'Ada Lovelace', 573.85)
mysql> INSERT INTO Book VALUES (5, 'Guide to Databases', 'Sam Knight', 143.75);
-- (5, 'Guide to Databases', 'Sam Knight', 143.75)
mysql> INSERT INTO Book VALUES (6, 'Mystery Tales', 'Chetan Bhagat', 368.00);
-- (6, 'Mystery Tales', 'Chetan Bhagat', 368.00)
Query OK, 6 rows affected (0.01 sec)
Records: 6 Duplicates: 0 Warnings: 0

mysql>
mysql> INSERT INTO Customer VALUES (1, 'Ravi Kumar', 'Mumbai');
-- (1, 'Ravi Kumar', 'Mumbai')
mysql> INSERT INTO Customer VALUES (2, 'Sara Ali', 'Aurangabad');
-- (2, 'Sara Ali', 'Aurangabad')
mysql> INSERT INTO Customer VALUES (3, 'Arjun Chopra', 'Agra');
-- (3, 'Arjun Chopra', 'Agra')
Query OK, 3 rows affected (0.01 sec)
Records: 3 Duplicates: 0 Warnings: 0

mysql>
mysql> INSERT INTO OrderDetails VALUES (1, '2020-09-01', 1);
-- (1, '2020-09-01', 1)
mysql> INSERT INTO OrderDetails VALUES (2, '2020-09-01', 2);
-- (2, '2020-09-01', 2)
Query OK, 2 rows affected (0.01 sec)
Records: 2 Duplicates: 0 Warnings: 0

mysql>
mysql> INSERT INTO OrderDetails VALUES (3, 1, 2, 1);
-- (3, 1, 2, 1)
mysql> INSERT INTO OrderDetails VALUES (4, 1, 2, 1);
-- (4, 1, 2, 1)
mysql> INSERT INTO OrderDetails VALUES (5, 4, 6, 2);
-- (5, 4, 6, 2);
```

{STEP 3 : UPDATE PRICES OF BOOKS BELOW 200}

```
mysql> UPDATE Book SET price = price * 1.2;
-- (1, 'SQL Guide', 'Jane Doe', 120.00)
Query OK, 6 rows affected (0.01 sec)
Records: 6 Duplicates: 0 Warnings: 0

mysql> SELECT * FROM Book WHERE price < 200;
+----+-----+-----+-----+
| book_id | title          | author | price |
+----+-----+-----+-----+
| 1       | SQL Guide      | Jane Doe | 120.00 |
| 2       | Python 101     | Chetan Bhagat | 482.50 |
| 3       | Learn Java     | Unknown | 287.50 |
| 4       | Advanced Guide to AI | Ada Lovelace | 573.85 |
| 5       | Guide to Databases | Sam Knight | 143.75 |
| 6       | Mystery Tales  | Chetan Bhagat | 368.00 |
+----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> UPDATE Book SET price = price * 1.2;
-- (1, 'SQL Guide', 'Jane Doe', 120.00)
Query OK, 6 rows affected (0.01 sec)
Records: 6 Duplicates: 0 Warnings: 0

mysql> SELECT * FROM Book WHERE price < 200;
+----+-----+-----+-----+
| book_id | title          | author | price |
+----+-----+-----+-----+
| 1       | SQL Guide      | Jane Doe | 120.00 |
| 2       | Python 101     | Chetan Bhagat | 482.50 |
| 3       | Learn Java     | Unknown | 287.50 |
| 4       | Advanced Guide to AI | Ada Lovelace | 573.85 |
| 5       | Guide to Databases | Sam Knight | 143.75 |
| 6       | Mystery Tales  | Chetan Bhagat | 368.00 |
+----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql>
```

{STEP 4 DELETE ALL BOOK WRITTEN BY AN AUTHOR NAMED UNKNOWN}

```
mysql> SELECT * FROM Book;
+----+-----+-----+-----+
| book_id | title          | author | price |
+----+-----+-----+-----+
| 1       | SQL Guide      | Jane Doe | 207.00 |
| 2       | Python 101     | Chetan Bhagat | 482.50 |
| 3       | Learn Java     | Unknown | 287.50 |
| 4       | Advanced Guide to AI | Ada Lovelace | 573.85 |
| 5       | Guide to Databases | Sam Knight | 143.75 |
| 6       | Mystery Tales  | Chetan Bhagat | 368.00 |
+----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> DELETE FROM Book
--> WHERE author = 'Unknown';
Query OK, 0 rows affected (0.00 sec)
```

{STEP 5 Display all customers whose city name starts with A}

```
--> where City Like 'A%';
ERROR 1146 (42S02): Table 'bookstore.customer' doesn't exist
mysql> SELECT * FROM Customer
--> where city like 'A%';
ERROR 1146 (42S02): Table 'bookstore.customer' doesn't exist
mysql> SELECT * FROM Customer
--> where city like 'A%';
+----+-----+-----+
| customer_id | name       | city |
+----+-----+-----+
| 2           | Anil Sharma | Ahmedabad |
| 3           | Sara Ali   | Aurangabad |
| 5           | Arjun Chopra | Agra |
+----+-----+-----+
3 rows in set (0.00 sec)

mysql>
```

```

-> where city like 'A%';
ERROR 1146 (42S02): Table 'bookstore.customer' doesn't exist
mysql> SELECT * FROM Customer
-> where city like 'A%';
ERROR 1146 (42S02): Table 'bookstore.customer' doesn't exist
mysql> SELECT * FROM Customer
-> where city like 'A%';
+-----+-----+-----+
| customer_id | name       | city      |
+-----+-----+-----+
| 2           | Anil Sharma | Ahmedabad |
| 3           | Sara Ali    | Aurangabad |
| 5           | Arjun Chopra | Agra      |
+-----+-----+-----+
3 rows in set (0.00 sec)

mysql>

```

```

-> where city like 'A%';
ERROR 1146 (42S02): Table 'bookstore.customer' doesn't exist
mysql> SELECT * FROM Customer
-> where city like 'A%';
ERROR 1146 (42S02): Table 'bookstore.customer' doesn't exist
mysql> SELECT * FROM Customer
-> where city like 'A%';
+-----+-----+-----+
| customer_id | name       | city      |
+-----+-----+-----+
| 2           | Anil Sharma | Ahmedabad |
| 3           | Sara Ali    | Aurangabad |
| 5           | Arjun Chopra | Agra      |
+-----+-----+-----+
3 rows in set (0.00 sec)

mysql>

```

H

{STEP 6 Find all books where the title contains the word Guide.}

STEP 7 Change the city of customer Ravi Kumar to Hyderabad

```

-> WHERE TITLE LIKE 'Guide';
Empty set (0.00 sec)

mysql> UPDATE Customer
-> SET city = 'Hyderabad'
-> WHERE name = 'Ravi Kumar';
Query OK, 1 row affected (0.23 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> SELECT * FROM Book
-> WHERE price BETWEEN 300 AND 500;
+-----+-----+-----+-----+
| book_id | title      | author      | price |
+-----+-----+-----+-----+

```

STEP 8 List all books with price between 300 and 500

STEP 9 : Add a new column stock quantity to the Book table

```

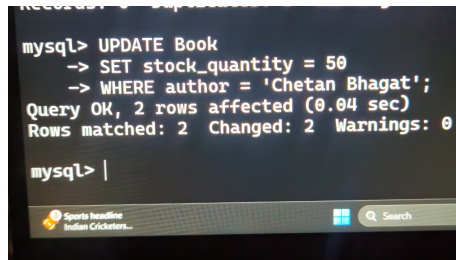
+-----+-----+-----+-----+
| 2 | Python 101 | Chetan Bhagat | 402 |
| 6 | Mystery Tales | Chetan Bhagat | 368 |
+-----+-----+-----+-----+
2 rows in set (0.02 sec)

mysql> ALTER TABLE Book
-> ADD COLUMN stock_quantity INTEGER;
Query OK, 0 rows affected (0.07 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> UPDATE Book
-> SET stock_quantity = 50
-> WHERE author = 'Chetan Bhagat';
Query OK, 2 rows affected (0.04 sec)

```

STEP 10 : Update stock quantity to 50 for all books by author Chetan Bhagat

A screenshot of a MySQL terminal window. The terminal shows the following commands and output: 

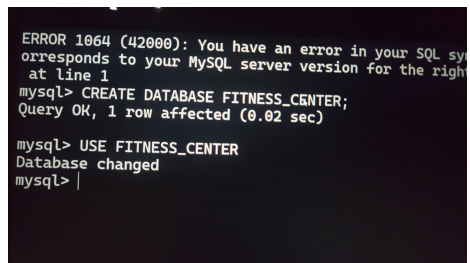
```
mysql> UPDATE Book  
-> SET stock_quantity = 50  
-> WHERE author = 'Chetan Bhagat';  
Query OK, 2 rows affected (0.04 sec)  
Rows matched: 2 Changed: 2 Warnings: 0  
  
mysql> |
```

 At the bottom of the terminal, there is a taskbar with a search bar and some application icons.

## PROBLEM 2 FITNESS CENTER MEMBERSHIP

STEP 0 : CREATE DATABASE FITNESS\_CENTER

0.1 : USE DATABASE

A screenshot of a MySQL terminal window. The terminal shows the following commands and output: 

```
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that  
corresponds to your MySQL server version for the right syntax to use  
at line 1  
mysql> CREATE DATABASE FITNESS_CENTER;  
Query OK, 1 row affected (0.02 sec)  
  
mysql> USE FITNESS_CENTER  
Database changed  
mysql> |
```

STEP 1 : CREATE TABLES

```
mysql> CREATE DATABASE FITNESS_CENTRE;
Query OK, 1 row affected (0.04 sec)

mysql> USE FITNESS_CENTRE;
Database changed
mysql> CREATE TABLE Member (
  -> member_id INTEGER PRIMARY KEY,
  -> name VARCHAR(50),
  -> join_date DATE,
  -> city VARCHAR(50)
  -> );
Query OK, 0 rows affected (0.28 sec)

mysql> CREATE TABLE Trainer (
  -> trainer_id INTEGER PRIMARY KEY,
  -> name VARCHAR(50),
  -> specialization VARCHAR(50)
  -> );
Query OK, 0 rows affected (0.04 sec)

mysql> CREATE TABLE Class (
  -> class_id INTEGER PRIMARY KEY,
  -> class_name VARCHAR(50),
  -> trainer_id INTEGER,
  -> FOREIGN KEY (trainer_id) REFERENCES Trainer(trainer_id)
  -> );
Query OK, 0 rows affected (0.06 sec)

mysql> CREATE TABLE Membership (
  -> membership_id INTEGER PRIMARY KEY,
  -> member_id INTEGER, FOREIGN KEY REFERENCES Trainer (member_id)
  -> class_id INTEGER, FOREIGN KEY REFERENCES Class (class_id)
  -> start_date (DATE),
  -> end_date (DATE),
  -> );
```

STEP 3 Update the specialization of trainers whose name starts with S to General

```
mysql> UPDATE Trainer
  -> SET specialization = 'General Fitness'
  -> WHERE name LIKE 'S';
Query OK, 0 rows affected (0.03 sec)
Rows matched: 0 Changed: 0 Warnings: 0
```

STEP 4 Delete all members whose city is Unknown

```
mysql> DELETE FROM Member
  -> WHERE city = 'Unknown';
Query OK, 0 rows affected (0.03 sec)
```

STEP 5 Display all classes where the class name contains Yoga

```
mysql> SELECT * FROM Class
-> WHERE class_name LIKE 'Yoga';
Empty set (0.02 sec)
```

STEP 6 Change the city of member Anjali Sharma to Pune

```
mysql> UPDATE Member
-> SET city = 'Pune'
-> WHERE name = 'Anjali Sharma';
Query OK, 1 row affected (0.04 sec)
Rows matched: 1 Changed: 1 Warnings: 0
```

STEP 7 List all members who joined after 2024-01-01

```
mysql> SELECT * FROM Member
-> WHERE join_date > '2024-01-01';
```

member_id	name	join_date	city
1	Anjali Sharma	2024-03-15	Pune
3	Neha Patel	2024-05-22	Ahmedabad
4	Amit Kumar	2025-01-03	Chennai
6	Vikram Verma	2024-07-30	Hyderabad

```
4 rows in set (0.02 sec)
```

STEP 8 Add a new column monthly fee to the Class table

```
mysql> ALTER TABLE Class
-> ADD COLUMN monthly_fee INTEGER;
Query OK, 0 rows affected (0.06 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

STEP 9 Set monthly fee to 2000 for all classes with Pilates in their name

```
mysql> UPDATE Class
-> SET monthly_fee = 2000
-> WHERE class_name LIKE 'Pilates';
Query OK, 0 rows affected (0.00 sec)
Rows matched: 0 Changed: 0 Warnings: 0
```

STEP 10 Display all trainers whose specialization is exactly 10 characters long

```
mysql> SELECT * FROM Trainer
-> WHERE LENGTH(specialization) = 10;
Empty set (0.02 sec)
```

### PROBLEM 3 : MOVIE RENTAL STORE

STEP 0 : CREATE DATABASE MOVIE\_STORE  
STEP 0.1:USE DATABASE

```
mysql> WHERE LENGTH(specialization) = 10
Empty set (0.02 sec)

mysql> CREATE DATABASE MOVIE_STORE;
Query OK, 1 row affected (0.14 sec)

mysql> USE MOVIE_STORE
Database changed
mysql>
```

STEP 1 : CREATE TABLE

```
mysql> CREATE DATABASE MOVIE_STORE;
Query OK, 1 row affected (0.14 sec)

mysql> USE MOVIE_STORE
Database changed
mysql> CREATE TABLE Movie (
  -> movie_id INTEGER PRIMARY KEY,
  -> title VARCHAR(100),
  -> genre VARCHAR(30),
  -> release_year INTEGER
  -> );
Query OK, 0 rows affected (0.08 sec)

mysql> CREATE TABLE Customer (
  -> customer_id INTEGER PRIMARY KEY,
  -> name VARCHAR(50),
  -> phone VARCHAR(15)
  -> );
Query OK, 0 rows affected (0.05 sec)

mysql> CREATE TABLE Rental (
  -> rental_id INTEGER PRIMARY KEY,
  -> rental_date DATE,
  -> customer_id INTEGER,
  -> FOREIGN KEY (customer_id) REFERENCES Customer(customer_id)
  -> );
Query OK, 0 rows affected (0.06 sec)

mysql> CREATE TABLE RentalDetails (
  -> rental_detail_id INTEGER PRIMARY KEY,
  -> rental_id INTEGER,
  -> movie_id INTEGER,
  -> return_date DATE,
  -> FOREIGN KEY (rental_id) REFERENCES Rental(rental_id),
  -> FOREIGN KEY (movie_id) REFERENCES Movie(movie_id)
  -> );
```

STEP 3 Update the genre of all movies released before 2000 to Classic

```
mysql> UPDATE Movie
-> SET genre = 'Classic'
-> WHERE release_year < 2000;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0
```

STEP 4 Delete movies whose title contains the word Test

```
mysql> DELETE FROM Movie
-> WHERE title LIKE 'Test';
Query OK, 0 rows affected (0.00 sec)
```

STEP 5 Display all customers whose name ends with Patel

```
mysql> SELECT * FROM Customer
-> WHERE name LIKE '%Patel';
```

customer_id	name	phone
2	Pooja Patel	9876543210
4	Neha Patel	7788996655

2 rows in set (0.00 sec)

STEP 6 Find all movies released between 2010 and 2020

```
mysql> SELECT * FROM Customer
-> WHERE name LIKE 'Patel';
Empty set (0.00 sec)

mysql> SELECT * FROM Movie
-> WHERE release_year BETWEEN 2010 AND 2020;
```

movie_id	title	genre	release_year
1	Fast Action	Action	2015
2	Comedy Nights	Comedy	2018
4	Thrill Action	Action	2020
6	Romantic Comedy	Comedy	2012

STEP 8 List all movies where the genre contains Action

```
mysql> SELECT * FROM Movie
-> WHERE genre LIKE 'Action';
```

movie_id	title	genre	release_year
1	Fast Action	Action	2015
4	Thrill Action	Action	2020

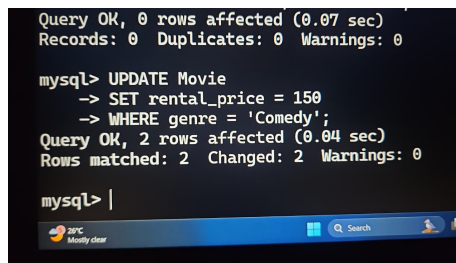


STEP 9 Add a new column rental price to the Movie table

```
\end
\begin{figure}[H]
  \centering
  \includegraphics[width=0.5\linewidth]{3.9.jpg}
  \label{fig:placeholder}
\end{figure}
```

```
\begin{lstlisting}
```

STEP 10 Set rental price to 150 for all Comedy movies

A screenshot of a terminal window showing a MySQL command-line interface. The prompt is 'mysql>'. The user has entered the command 'UPDATE Movie' followed by two indented lines: '-> SET rental\_price = 150' and '-> WHERE genre = 'Comedy';'. The output shows 'Query OK, 2 rows affected (0.04 sec)' and 'Rows matched: 2 Changed: 2 Warnings: 0'. The prompt 'mysql>' is shown again with a cursor. At the bottom of the terminal, there is a taskbar with a search bar and some icons.

```
Query OK, 2 rows affected (0.04 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> UPDATE Movie
      -> SET rental_price = 150
      -> WHERE genre = 'Comedy';
Query OK, 2 rows affected (0.04 sec)
Rows matched: 2 Changed: 2 Warnings: 0

mysql> |
```