

SQL-COMPREHENSIVE ASSESSMENT (Topic : Library Management System)

Topic : Library Management System

Create a database named library and following TABLES in the database:

- 1. Branch**
- 2. Employee**
- 3. Books**
- 4. Customer**
- 5. IssueStatus**
- 6. ReturnStatus**

Attributes for the tables:

1. Branch

- **Branch_no** - Set as PRIMARY KEY
- **Manager_Id**
- **Branch_address**
- **Contact_no**

MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Navigator Schemas SQL File 1 SQL File 2* library - Schema SQL File 4* ×

Filter objects

Schemas

- company
- country
- countries
- customers
- global_store_db
- india
- library**
- Tables
- Views
- Stored Procedures
- Functions

- product
- sales
- store
- student
- sys
- teacher_log
- teachers
- worker

1 USE LIBRARY;

2 CREATE TABLE IF NOT EXISTS Branch (

3 Branch_no INT PRIMARY KEY,

4 Manager_Id INT,

5 Branch_address VARCHAR(255),

6 Contact_no VARCHAR(20)

7);

8

9

Administration Schemas

Information Output

Schema: library

#	Time	Action	Message
1	17:05:39	Apply changes to library	Changes applied
2	17:06:50	USE LIBRARY	0 row(s) affected
3	17:06:58	CREATE TABLE IF NOT EXISTS Branch (Branch_no INT PRIMARY KEY, Manager_Id INT, Branch_ad...)	0 row(s) affected

Insert Values :

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the database schema with the **library** schema selected, containing Tables, Views, Stored Procedures, and Functions.
- SQL File 4*** (active tab):


```

5     Branch_address VARCHAR(255),
6     Contact_no VARCHAR(20)
7   );
8
9 •   INSERT INTO Branch (Branch_no, Manager_Id, Branch_address, Contact_no) VALUES
10    (1, 101, '123 Main St', '123-456-7890'),
11    (2, 102, '456 Elm St', '987-654-3210'),
12    (3, 103, '789 Oak St', '321-654-0987');
13
14 • ○ CREATE TABLE IF NOT EXISTS Employee (
15     Emp_Id INT PRIMARY KEY,
16     Emp_name VARCHAR(255),
17     Position VARCHAR(100),
18     Salary DECIMAL(10, 2),
19     Branch_no INT,
20     FOREIGN KEY (Branch_no) REFERENCES Branch(Branch_no)
21 );
22
      
```
- Output:** Displays the execution log with the following entries:

#	Time	Action	Message
5	17:09:26	CREATE TABLE IF NOT EXISTS Books (ISBN VARCHAR(13) PRIMARY KEY, Book_title VARCHAR(2...	0 row(s) affected
6	17:12:42	CREATE TABLE IF NOT EXISTS Customer (Customer_Id INT PRIMARY KEY, Customer_name VARCHAR...	0 row(s) affected
7	17:13:38	CREATE TABLE IF NOT EXISTS IssueStatus (Issue_Id INT PRIMARY KEY, Issued_cust INT, Issued...	0 row(s) affected
8	17:14:27	CREATE TABLE IF NOT EXISTS ReturnStatus (Return_Id INT PRIMARY KEY, Return_cust INT, Ret...	0 row(s) affected
9	17:16:16	SELECT Book_title, Category, Rental_Price FROM Books WHERE Status = 'yes' LIMIT 0, 1000	0 row(s) returned
10	17:18:39	INSERT INTO Branch (Branch_no, Manager_Id, Branch_address, Contact_no) VALUES (1, 101, '123 Main St...', 3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0	6 row(s) affected Records: 6 Duplicates: 0 Warnings: 0
11	17:19:06	INSERT INTO Employee (Emp_Id, Emp_name, Position, Salary, Branch_no) VALUES (201, 'John Doe', 'Mana...	6 row(s) affected Records: 6 Duplicates: 0 Warnings: 0

2. Employee

- **Emp_Id** – Set as **PRIMARY KEY**
- **Emp_name**
- **Position**
- **Salary**
- **Branch_no** - Set as **FOREIGN KEY** and it refer **Branch_no** in **Branch** table

MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Navigator: Schemas SQL File 1 SQL File 2* library - Schema SQL File 4* ×

Schemas: company, country, countries, customers, global_store_db, india, library, product, sales, store, student, sys, teacher_log, teachers, worker

SQL File 1:

```
1 • USE LIBRARY;
2 • CREATE TABLE IF NOT EXISTS Branch (
3     Branch_no INT PRIMARY KEY,
4     Manager_Id INT,
5     Branch_address VARCHAR(255),
6     Contact_no VARCHAR(20)
7 );
8 • CREATE TABLE IF NOT EXISTS Employee (
9     Emp_Id INT PRIMARY KEY,
10    Emp_name VARCHAR(255),
11    Position VARCHAR(100),
12    Salary DECIMAL(10, 2),
13    Branch_no INT,
14    FOREIGN KEY (Branch_no) REFERENCES Branch(Branch_no)
15 );
16
```

Administration Schemas

Information: Schema: library

Action Output

#	Time	Action	Message
1	17:05:39	Apply changes to library	Changes applied
2	17:06:50	USE LIBRARY	0 row(s) affected
3	17:06:58	CREATE TABLE IF NOT EXISTS Branch (Branch_no INT PRIMARY KEY, Manager_Id INT, Branch_ad...)	0 row(s) affected
4	17:08:21	CREATE TABLE IF NOT EXISTS Employee (Emp_Id INT PRIMARY KEY, Emp_name VARCHAR(255), ...)	0 row(s) affected

Insert Values:

MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Navigator: Schemas SQL File 1 SQL File 2 library - Schema SQL File 4* ×

Schemas: company, country, countries, customers, global_store_db, india, library, product, sales, store, student, sys, teacher_log, teachers, worker

SQL File 1:

```

17     Position VARCHAR(100),
18     Salary DECIMAL(10, 2),
19     Branch_no INT,
20     FOREIGN KEY (Branch_no) REFERENCES Branch(Branch_no)
21   );
22
23 • INSERT INTO Employee (Emp_Id, Emp_name, Position, Salary, Branch_no) VALUES
24   (201, 'John Doe', 'Manager', 50000, 1),
25   (202, 'Jane Smith', 'Clerk', 35000, 1),
26   (203, 'Alice Johnson', 'Librarian', 40000, 2),
27   (204, 'Bob Williams', 'Assistant Librarian', 30000, 2),
28   (205, 'Michael Brown', 'Clerk', 32000, 3),
29   (206, 'Emily Jones', 'Assistant Librarian', 28000, 3);
30
31
32 • CREATE TABLE IF NOT EXISTS Books (
33   ISBN VARCHAR(13) PRIMARY KEY,
34   Book_title VARCHAR(255),
35   Category VARCHAR(100)

```

Output:

#	Time	Action	Message
5	17:09:26	CREATE TABLE IF NOT EXISTS Books (ISBN VARCHAR(13) PRIMARY KEY, Book_title VARCHAR(255), Category VARCHAR(100))	0 row(s) affected
6	17:12:42	CREATE TABLE IF NOT EXISTS Customer (Customer_Id INT PRIMARY KEY, Customer_name VARCHAR(100), Address VARCHAR(255))	0 row(s) affected
7	17:13:38	CREATE TABLE IF NOT EXISTS IssueStatus (Issue_Id INT PRIMARY KEY, Issued_cust INT, Issued_branch INT, Status CHAR(1))	0 row(s) affected
8	17:14:27	CREATE TABLE IF NOT EXISTS ReturnStatus (Return_Id INT PRIMARY KEY, Return_cust INT, Return_branch INT, Status CHAR(1))	0 row(s) affected
9	17:16:16	SELECT Book_title, Category, Rental_Price FROM Books WHERE Status = 'yes' LIMIT 0, 1000	0 row(s) returned
10	17:18:39	INSERT INTO Branch (Branch_no, Manager_Id, Branch_address, Contact_no) VALUES (1, 101, '123 Main St.', '555-1234')	3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0
11	17:19:06	INSERT INTO Employee (Emp_Id, Emp_name, Position, Salary, Branch_no) VALUES (201, 'John Doe', 'Manager', 50000, 1)	6 row(s) affected Records: 6 Duplicates: 0 Warnings: 0

3. Books

- ISBN - Set as PRIMARY KEY
- Book_title
- Category
- Rental_Price
- Status [Give yes if book available and no if book not available]
- Author
- Publisher

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator Schemas SQL File 1 SQL File 2* library - Schema SQL File 4* ×

Schemas

- company
- country
- countries
- customers
- global_store_db
- india
- library
- Tables
- Views
- Stored Procedures
- Functions
- product
- sales
- store
- student
- sys
- teacher_log
- teachers
- worker

SQL File 1 SQL File 2* library - Schema SQL File 4* ×

```
10     Emp_name VARCHAR(255),
11     Position VARCHAR(100),
12     Salary DECIMAL(10, 2),
13     Branch_no INT,
14     FOREIGN KEY (Branch_no) REFERENCES Branch(Branch_no)
15   );
16
17 • CREATE TABLE IF NOT EXISTS Books (
18     ISBN VARCHAR(13) PRIMARY KEY,
19     Book_title VARCHAR(255),
20     Category VARCHAR(100),
21     Rental_Price DECIMAL(10, 2),
22     Status ENUM('yes', 'no'),
23     Author VARCHAR(255),
24     Publisher VARCHAR(255)
25   );
26
27
```

Administration Schemas

Information Schema: library

Action Output

#	Time	Action	Message
1	17:05:39	Apply changes to library	Changes applied
2	17:06:50	USE LIBRARY	0 row(s) affected
3	17:06:58	CREATE TABLE IF NOT EXISTS Branch (Branch_no INT PRIMARY KEY, Manager_Id INT, Branch_ad...)	0 row(s) affected
4	17:08:21	CREATE TABLE IF NOT EXISTS Employee (Emp_id INT PRIMARY KEY, Emp_name VARCHAR(255), ...)	0 row(s) affected
5	17:09:26	CREATE TABLE IF NOT EXISTS Books (ISBN VARCHAR(13) PRIMARY KEY, Book_title VARCHAR(255), ...)	0 row(s) affected

Insert Values :

MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Navigator: Schemas SQL File 1 SQL File 2 library - Schema SQL File 4 ×

File Edit View Insert Query Database Server Tools Scripting Help

Schemas

company country countrys customers global_store_db india library Tables Views Stored Procedures Functions product sales store student sys teacher_log teachers worker

SQL File 1 SQL File 2* library - Schema SQL File 4* ×

```

29      (206, 'Emily Jones', 'Assistant Librarian', 28000, 3);
30
31
32 • CREATE TABLE IF NOT EXISTS Books (
33     ISBN VARCHAR(13) PRIMARY KEY,
34     Book_title VARCHAR(255),
35     Category VARCHAR(100),
36     Rental_Price DECIMAL(10, 2),
37     Status ENUM('yes', 'no'),
38     Author VARCHAR(255),
39     Publisher VARCHAR(255)
40 );
41
42 • INSERT INTO Books (ISBN, Book_title, Category, Rental_Price, Status, Author, Publisher) VALUES
43     ('978', 'To Kill a Mockingbird', 'Fiction', 10.99, 'yes', 'Harper Lee', 'HarperCollins'),
44     ('979', 'Harry Potter and the Philosopher's Stone', 'Fantasy', 12.99, 'yes', 'J.K. Rowling', 'Bloomsbury'),
45     ('980', 'The Catcher in the Rye', 'Fiction', 9.99, 'yes', 'J.D. Salinger', 'Little, Brown and Company');
46

```

Administration Schemas

Information Schema: library

Action Output

#	Time	Action	Message
7	17:13:38	CREATE TABLE IF NOT EXISTS IssueStatus (Issue_Id INT PRIMARY KEY, Issued_cust INT, Issued...)	0 row(s) affected
8	17:14:27	CREATE TABLE IF NOT EXISTS ReturnStatus (Return_Id INT PRIMARY KEY, Return_cust INT, Ret...)	0 row(s) affected
9	17:16:16	SELECT Book_title, Category, Rental_Price FROM Books WHERE Status = 'yes' LIMIT 0, 1000	0 row(s) returned
10	17:18:39	INSERT INTO Branch (Branch_no, Manager_Id, Branch_address, Contact_no) VALUES (1, 101, '123 Main St...', 3 rows) affected Records: 3 Duplicates: 0 Warnings: 0	
11	17:19:06	INSERT INTO Employee (Emp_Id, Emp_name, Position, Salary, Branch_no) VALUES (201, 'John Doe', 'Manager', 60000, 1) rows) affected Records: 6 Duplicates: 0 Warnings: 0	
12	17:20:48	INSERT INTO Books (ISBN, Book_title, Category, Rental_Price, Status, Author, Publisher) VALUES ('978-3-16-14841-9', 'The Catcher in the Rye', 'Fiction', 9.99, 'yes', 'J.D. Salinger', 'Little, Brown and Company')	Error Code: 1406. Data too long for column 'ISBN' at row 1
13	17:21:41	INSERT INTO Books (ISBN, Book_title, Category, Rental_Price, Status, Author, Publisher) VALUES ('978-3-16-14841-9', 'The Catcher in the Rye', 'Fiction', 9.99, 'yes', 'J.D. Salinger', 'Little, Brown and Company')	3 rows) affected Records: 3 Duplicates: 0 Warnings: 0

4. Customer

- **Customer_Id** - Set as PRIMARY KEY
- **Customer_name**
- **Customer_address**
- **Reg_date**

MySQL Workbench

Local instance MySQL800 ×

File Edit View Query Database Server Tools Scripting Help

Navigator SQL File 1 SQL File 2* library - Schema SQL File 4* ×

SCHEMAS

Filter objects

- company
- country
- countries
- customers
- global_store_db
- india
- library**
 - Tables
 - Views
 - Stored Procedures
 - Functions
- product
- sales
- store
- student
- sys
- teacher_log
- teachers
- worker

Administration Schemas

Information

Schema: library

Action Output

#	Time	Action	Message
1	17:05:39	Apply changes to library	Changes applied
2	17:06:50	USE LIBRARY	0 row(s) affected
3	17:06:58	CREATE TABLE IF NOT EXISTS Branch (Branch_no INT PRIMARY KEY, Manager_Id INT, Branch_ad...)	0 row(s) affected
4	17:08:21	CREATE TABLE IF NOT EXISTS Employee (Emp_Id INT PRIMARY KEY, Emp_name VARCHAR(255), ...)	0 row(s) affected
5	17:09:26	CREATE TABLE IF NOT EXISTS Books (ISBN VARCHAR(13) PRIMARY KEY, Book_title VARCHAR(255), ...)	0 row(s) affected
6	17:12:42	CREATE TABLE IF NOT EXISTS Customer (Customer_Id INT PRIMARY KEY, Customer_name VARCHAR(...))	0 row(s) affected

Object Info Session

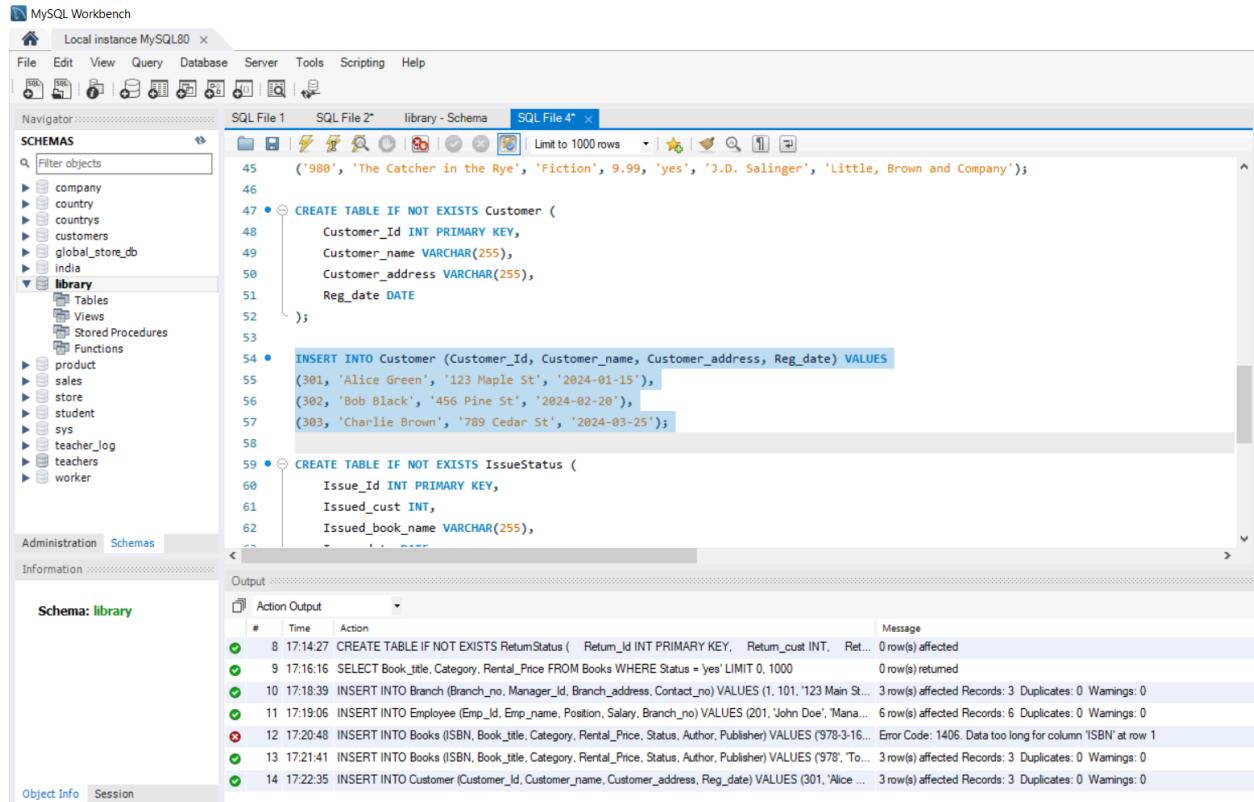
```

15 );
16
17 • CREATE TABLE IF NOT EXISTS Books (
18     ISBN VARCHAR(13) PRIMARY KEY,
19     Book_title VARCHAR(255),
20     Category VARCHAR(100),
21     Rental_Price DECIMAL(10, 2),
22     Status ENUM('yes', 'no'),
23     Author VARCHAR(255),
24     Publisher VARCHAR(255)
25 );
26
27 • CREATE TABLE IF NOT EXISTS Customer (
28     Customer_Id INT PRIMARY KEY,
29     Customer_name VARCHAR(255),
30     Customer_address VARCHAR(255),
31     Reg_date DATE
32 );

```

Output

Insert Values :



5. IssueStatus

- **Issue_Id** - Set as **PRIMARY KEY**
 - **Issued_cust** – Set as **FOREIGN KEY** and it refer **customer_id** in **CUSTOMER** table
 - Issued_book_name**
 - **Issue_date**
 - **Isbn_book** – Set as **FOREIGN KEY** and it should refer **isbn** in **BOOKS** table

MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Navigator Schemas SQL File 1 SQL File 2 library - Schema SQL File 4* ×

File Edit View Insert Query Database Server Tools Scripting Help

Schemas

company country countrys customers global_store_db india library Tables Views Stored Procedures Functions product sales store student sys teacher_log teachers worker

SQL File 1 SQL File 2* library - Schema SQL File 4* ×

```

25    );
26
27 • CREATE TABLE IF NOT EXISTS Customer (
28     Customer_Id INT PRIMARY KEY,
29     Customer_name VARCHAR(255),
30     Customer_address VARCHAR(255),
31     Reg_date DATE
32 );
33
34 • CREATE TABLE IF NOT EXISTS IssueStatus (
35     Issue_Id INT PRIMARY KEY,
36     Issued_cust INT,
37     Issued_book_name VARCHAR(255),
38     Issue_date DATE,
39     Isbn_book VARCHAR(13),
40     FOREIGN KEY (Issued_cust) REFERENCES Customer(Customer_Id),
41     FOREIGN KEY (Isbn_book) REFERENCES Books(ISBN)
42 );

```

Administration Schemas

Information

Schema: library

Action Output

#	Time	Action	Message
1	17:05:39	Apply changes to library	Changes applied
2	17:06:50	USE LIBRARY	0 row(s) affected
3	17:06:58	CREATE TABLE IF NOT EXISTS Branch (Branch_no INT PRIMARY KEY, Manager_Id INT, Branch_ad...	0 row(s) affected
4	17:08:21	CREATE TABLE IF NOT EXISTS Employee (Emp_Id INT PRIMARY KEY, Emp_name VARCHAR(255), ...	0 row(s) affected
5	17:09:26	CREATE TABLE IF NOT EXISTS Books (ISBN VARCHAR(13) PRIMARY KEY, Book_title VARCHAR(255), ...	0 row(s) affected
6	17:12:42	CREATE TABLE IF NOT EXISTS Customer (Customer_Id INT PRIMARY KEY, Customer_name VARCHAR(...)	0 row(s) affected
7	17:13:38	CREATE TABLE IF NOT EXISTS IssueStatus (Issue_Id INT PRIMARY KEY, Issued_cust INT, Issued_b...	0 row(s) affected

Object Info Session

Insert Values:

```

CREATE TABLE IssueStatus (
    Issue_Id INT,
    Issued_cust INT,
    Issued_book_name VARCHAR(255),
    Issue_date DATE,
    Isbn_book VARCHAR(13),
    FOREIGN KEY (Issued_cust) REFERENCES Customer(Customer_Id),
    FOREIGN KEY (Isbn_book) REFERENCES Books(ISBN)
);

INSERT INTO IssueStatus (Issue_Id, Issued_cust, Issued_book_name, Issue_date, Isbn_book) VALUES
(401, 301, 'To Kill a Mockingbird', '2024-05-01', '978'),
(402, 302, 'Harry Potter and the Philosopher\'s Stone', '2024-05-05', '979'),
(403, 303, 'The Catcher in the Rye', '2024-05-10', '980');

CREATE TABLE IF NOT EXISTS ReturnStatus (
    Return_Id INT PRIMARY KEY,
    Return_cust INT,
    Return_book_name VARCHAR(255),
    Return_date DATE,
    FOREIGN KEY (Return_cust) REFERENCES Customer(Customer_Id),
    FOREIGN KEY (Return_book_name) REFERENCES Books(ISBN)
);

```

Action Output

#	Time	Action	Message
10	17:18:39	INSERT INTO Branch (Branch_no, Manager_Id, Branch_address, Contact_no) VALUES (1, 101, '123 Main St...', '3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0')	
11	17:19:06	INSERT INTO Employee (Emp_Id, Emp_name, Position, Salary, Branch_no) VALUES (201, 'John Doe', 'Manager', 50000, 1) 6 row(s) affected Records: 6 Duplicates: 0 Warnings: 0	
12	17:20:48	INSERT INTO Books (ISBN, Book_title, Category, Rental_Price, Status, Author, Publisher) VALUES ('978-3-16-14840-0', 'The Catcher in the Rye', 'Fiction', 12.99, 'Available', 'J.D. Salinger', 'Random House') Error Code: 1406. Data too long for column 'ISBN' at row 1	
13	17:21:41	INSERT INTO Books (ISBN, Book_title, Category, Rental_Price, Status, Author, Publisher) VALUES ('978-0-553-20954-4', 'Harry Potter and the Philosopher\'s Stone', 'Fantasy', 9.99, 'Available', 'J.K. Rowling', 'Scholastic') 3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0	
14	17:22:35	INSERT INTO Customer (Customer_Id, Customer_name, Customer_address, Reg_date) VALUES (301, 'Alice Smith', '123 Elm Street', '2024-05-01') 3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0	
15	17:23:50	INSERT INTO IssueStatus (Issue_Id, Issued_cust, Issued_book_name, Issue_date, Isbn_book) VALUES (401, 301, 'To Kill a Mockingbird', '2024-05-01', '978') Error Code: 1406. Data too long for column 'Isbn_book' at row 1	
16	17:24:23	INSERT INTO IssueStatus (Issue_Id, Issued_cust, Issued_book_name, Issue_date, Isbn_book) VALUES (401, 301, 'To Kill a Mockingbird', '2024-05-01', '978) 3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0	

6. ReturnStatus

- **Return_Id** - Set as **PRIMARY KEY**
- **Return_cust**
- **Return_book_name**
- **Return_date**
- **Isbn_book2** - Set as **FOREIGN KEY** and it should refer isbn in BOOKS table

MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Navigator: Schemas SQL File 1 SQL File 2 library - Schema SQL File 4* ×

Schemas: company, country, countrys, customers, global_store_db, india, library, product, sales, store, student, sys, teacher_log, teachers, worker.

SQL File 4*:

```

35 Issue_Id INT PRIMARY KEY,
36 Issued_cust INT,
37 Issued_book_name VARCHAR(255),
38 Issue_date DATE,
39 Isbn_book VARCHAR(13),
40 FOREIGN KEY (Issued_cust) REFERENCES Customer(Customer_Id),
41 FOREIGN KEY (Isbn_book) REFERENCES Books(ISBN)
42 );
43
44 CREATE TABLE IF NOT EXISTS ReturnStatus (
45     Return_Id INT PRIMARY KEY,
46     Return_cust INT,
47     Return_book_name VARCHAR(255),
48     Return_date DATE,
49     Isbn_book2 VARCHAR(13),
50     FOREIGN KEY (Return_cust) REFERENCES Customer(Customer_Id),
51     FOREIGN KEY (Isbn_book2) REFERENCES Books(ISBN)
52 );

```

Administration Schemas

Information: Schema: library

Action Output:

#	Time	Action	Message
2	17:06:50	USE LIBRARY	0 row(s) affected
3	17:06:58	CREATE TABLE IF NOT EXISTS Branch (Branch_no INT PRIMARY KEY, Manager_Id INT, Branch_a...)	0 row(s) affected
4	17:08:21	CREATE TABLE IF NOT EXISTS Employee (Emp_Id INT PRIMARY KEY, Emp_name VARCHAR(255), ...)	0 row(s) affected
5	17:09:26	CREATE TABLE IF NOT EXISTS Books (ISBN VARCHAR(13) PRIMARY KEY, Book_title VARCHAR(25...))	0 row(s) affected
6	17:12:42	CREATE TABLE IF NOT EXISTS Customer (Customer_Id INT PRIMARY KEY, Customer_name VARCHA...)	0 row(s) affected
7	17:13:38	CREATE TABLE IF NOT EXISTS IssueStatus (Issue_Id INT PRIMARY KEY, Issued_cust INT, Issued_...)	0 row(s) affected
8	17:14:27	CREATE TABLE IF NOT EXISTS ReturnStatus (Return_Id INT PRIMARY KEY, Return_cust INT, Retu...)	0 row(s) affected

Object Info Session

Insert Values :

The screenshot shows the MySQL Workbench interface. In the top navigation bar, the database is set to "Local instance MySQL80". The main area has four tabs: "SQL File 1", "SQL File 2*", "library - Schema", and "SQL File 4*". The "library - Schema" tab is active. The left sidebar lists databases and their schemas. Under the "library" schema, there are tables, views, stored procedures, and functions. The SQL editor contains the following code:

```

15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84 • INSERT INTO ReturnStatus (Return_Id, Return_cust, Return_book_name, Return_date, Isbn_book2) VALUES
85 (501, 301, 'To Kill a Mockingbird', '2024-06-01', '978'),
86 (502, 302, 'Harry Potter and the Philosopher\'s Stone', '2024-06-05', '979'),
87 (503, 303, 'The Catcher in the Rye', '2024-06-10', '980');
88
89 • SELECT Book_title, Category, Rental_Price
90 FROM Books
91 WHERE Status = 'yes';
92

```

The "Output" pane at the bottom shows the results of the executed queries, including log entries and message details.

1. Retrieve the book title, category, and rental price of all available books.

QUERY :

```
SELECT Book_title, Category, Rental_Price
```

```
FROM Books
```

```
WHERE Status = 'yes';
```

OUTPUT :

The screenshot shows the MySQL Workbench interface. The top menu bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. Below the menu is a toolbar with various icons. The left sidebar is titled 'Navigator' and shows a tree view of 'SCHEMAS' containing databases like company, country, countrys, customers, global_store_db, india, library, product, sales, store, student, sys, teacher_log, teachers, and worker. The main area has tabs for 'SQL File 2*', 'SQL File 4*' (which is active), 'SQL File 4*' (disabled), and 'SQL File 5'. The SQL tab contains several lines of SQL code. The results tab shows a grid of data from the 'library' schema, specifically from the 'Books' table, with columns Book_Title, Category, and Rental_Price. The data includes titles like 'To Kill a Mockingbird', 'Harry Potter and the Philosopher's Stone', 'The Catcher in the Rye', etc., categorized by Fiction, Fantasy, History, etc., with rental prices ranging from 9.99 to 30.00.

```
139
140 • SELECT Book_title, Category, Rental_Price
141   FROM Books
142   WHERE Status = 'yes';
143
144 • SELECT Emp_name, Salary
145   FROM Employee
146   ORDER BY Salary DESC;
147
148 • SELECT Books.Book_title, Customer.Customer_name
149   FROM IssueStatus
150   INNER JOIN Books ON IssueStatus.Ibsn_book = Books.ISBN
151   INNER JOIN Customer ON IssueStatus.Issued_cust = Customer.Customer_Id;
152
153 • SELECT Category, COUNT(*) AS Total_Count
154   FROM Books
155   GROUP BY Category;
156
```

Book_Title	Category	Rental_Price
To Kill a Mockingbird	Fiction	10.99
Harry Potter and the Philosopher's Stone	Fantasy	12.99
The Catcher in the Rye	Fiction	9.99
Modern History	History	11.09
Ancient History	History	15.12
Talkative Man	Novel	30.00
I Had a Love Story	Novel	26.02
Alchemist	Fantasy Fiction	79.70

2. List the employee names and their respective salaries in descending order of salary.

QUERY :

```
SELECT Emp_name, Salary
```

```
FROM Employee
```

```
ORDER BY Salary DESC;
```

OUTPUT :

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree, which includes 'company', 'country', 'countries', 'customers', 'global_store_db', 'india', 'library', 'product', 'sales', 'store', 'student', 'sys', 'teacher_log', 'teachers', and 'worker'. The 'library' schema is selected. The main area contains several SQL queries in the 'SQL File 2*' tab:

```

139
140 •   SELECT Book_title, Category, Rental_Price
141     FROM Books
142     WHERE Status = 'yes';
143
144 •   SELECT Emp_name, Salary
145     FROM Employee
146     ORDER BY Salary DESC;
147
148 •   SELECT Books.Book_title, Customer.Customer_name
149     FROM IssueStatus
150     INNER JOIN Books ON IssueStatus.Issn_book = Books.ISBN
151     INNER JOIN Customer ON IssueStatus.Issued_cust = Customer.Customer_Id;
152
153 •   SELECT Category, COUNT(*) AS Total_Count
154     FROM Books

```

The 'Result Grid' shows the output of the second query:

Emp_name	Salary
Azin	95000.00
Sudha	90000.00
Sudha	90000.00
Veena	80000.00
John Doe	50000.00
Geetha	40000.00
Alic Johnson	40000.00

The 'Output' section shows the history of actions:

#	Time	Action	Message
47	19:06:20	SELECT Emp_name, Position FROM Employee WHERE Salary > 50000 LIMIT 0, 1000	4 row(s) returned
48	19:08:01	SELECT Category, COUNT(*) AS Total_Count FROM Books GROUP BY Category LIMIT 0, 1000	5 row(s) returned

3. Retrieve the book titles and the corresponding customers who have issued those books.

QUERY :

```

SELECT Books.Book_title, Customer.Customer_name
FROM IssueStatus
INNER JOIN Books ON IssueStatus.Issn_book = Books.ISBN
INNER JOIN Customer ON IssueStatus.Issued_cust = Customer.Customer_Id;

```

OUTPUT :

The screenshot shows the MySQL Workbench interface. In the top navigation bar, 'File', 'Edit', 'View', 'Query', 'Database', 'Server', 'Tools', 'Scripting', and 'Help' are visible. Below the menu is a toolbar with icons for file operations like Open, Save, and Print. The main area has tabs for 'SQL File 2*', 'SQL File 4*' (which is active), and 'SQL File 5'. The SQL editor contains the following code:

```
139
140 • SELECT Book_title, Category, Rental_Price
141   FROM Books
142   WHERE Status = 'yes';
143
144 • SELECT Emp_name, Salary
145   FROM Employee
146   ORDER BY Salary DESC;
147
148 • SELECT Books.Book_title, Customer.Customer_name
149   FROM IssueStatus
150   INNER JOIN Books ON IssueStatus.Issn_book = Books.ISBN
151   INNER JOIN Customer ON IssueStatus.Issued_cust = Customer.Customer_Id;
152
153 • SELECT Category, COUNT(*) AS Total_Count
```

The results grid shows the following data:

Book_title	Customer_name
To Kill a Mockingbird	Alice Green
Harry Potter and the Philosopher's Stone	Bob Black
The Catcher in the Rye	Charlie Brown
Talkative Man	Veena

The 'Output' pane at the bottom shows the following log entries:

#	Time	Action	Message
46	19:04:11	SELECT Customer_name FROM Customer WHERE Reg_date < '2022-01-01' AND Customer_Id NOT IN (SEL...	1 row(s) returned
47	19:06:20	SELECT Emp_name, Position FROM Employee WHERE Salary > 50000 LIMIT 0, 1000	4 row(s) returned
48	19:08:01	SELECT Category, COUNT(*) AS Total_Count FROM Books GROUP BY Category LIMIT 0, 1000	5 row(s) returned
49	19:10:00	SELECT Books.Book_title, Customer.Customer_name FROM IssueStatus INNER JOIN Books ON IssueStatus...	4 row(s) returned

4. Display the total count of books in each category.

QUERY :

```
SELECT Category, COUNT(*) AS Total_Count
```

FROM Books

GROUP BY Category;

OUTPUT:

The screenshot shows the MySQL Workbench interface with the following details:

- File**, **Edit**, **View**, **Query**, **Database**, **Server**, **Tools**, **Scripting**, **Help** menu items.
- Navigator** pane on the left showing the schema tree with nodes like company, country, countries, customers, global_store_db, india, library, product, sales, store, student, sys, teacher_log, teachers, worker.
- SQL File 4*** tab active, containing the following SQL code:

```
152 •    SELECT Category, COUNT(*) AS Total_Count
153   FROM Books
154   GROUP BY Category;
155
156
157 •    SELECT Emp_name, Position
158   FROM Employee
159   WHERE Salary > 50000;
160
161 •    SELECT Customer_name
162   FROM Customer
163   WHERE Reg_date < '2022-01-01'
164   AND Customer_Id NOT IN (SELECT Issued_cust FROM IssueStatus);
165
166 •    INSERT INTO Customer (Customer_Id,Customer_name, Customer_address, Reg_date)
```

- Result Grid** pane showing the results of the first query:

Category	Total_Count
Fiction	2
Fantasy	1
History	2
Novel	2
Fantasy Fiction	1

- Administration**, **Schemas**, **Information** tabs at the bottom, with **Schemas** selected.
- Schema: library** dropdown at the bottom left.
- Action Output** pane at the bottom right showing the history of queries:

#	Time	Action	Message
45	19.01.28	SELECT Branch_no, COUNT(*) AS Total_Employees FROM Employee GROUP BY Branch_no LIMIT 0, 1000	6 row(s) returned
46	19.04.11	SELECT Customer_name FROM Customer WHERE Reg_date < '2022-01-01' AND Customer_Id NOT IN (SEL...	1 row(s) returned
47	19.06.20	SELECT Emp_name, Position FROM Employee WHERE Salary > 50000 LIMIT 0, 1000	4 row(s) returned
48	19.08.01	SELECT Category, COUNT(*) AS Total_Count FROM Books GROUP BY Category LIMIT 0, 1000	5 row(s) returned

5. Retrieve the employee names and their positions for the employees whose salaries are above Rs.50,000.

QUERY :

SELECT Emp_name, Position

FROM Employee

WHERE Salary > 50000;

OUTPUT :

The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** Shows the current schema is "library". Other schemas listed include: company, country, countrys, customers, global_store_db, india, product, sales, store, student, sys, teacher_log, teachers, worker.
- SQL Editor:** Contains the following SQL code:

```
152
153 •  SELECT Category, COUNT(*) AS Total_Count
154   FROM Books
155   GROUP BY Category;
156
157 •  SELECT Emp_name, Position
158   FROM Employee
159   WHERE Salary > 50000;
160
161 •  SELECT Customer_name
162   FROM Customer
163   WHERE Reg_date < '2022-01-01'
164   AND Customer_Id NOT IN (SELECT Issued_cust FROM IssueStatus);
165
166 •  INSERT INTO Customer (Customer_Id,Customer_name, Customer_address, Reg_date)
```
- Result Grid:** Displays the results of the query in the SQL editor.

Emp_name	Position
Sudha	Auditor
Veeva	Engineer
Azin	Engineer
Sudha	Auditor
- Output Window:** Shows the log of actions taken during the session.

#	Time	Action	Message
44	19:01:01	INSERT INTO Customer (Customer_Id,Customer_name, Customer_address, Reg_date) SELECT DISTINCT ...	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
45	19:01:28	SELECT Branch_no, COUNT(*) AS Total_Employees FROM Employee GROUP BY Branch_no LIMIT 0, 1000	6 row(s) returned
46	19:04:11	SELECT Customer_name FROM Customer WHERE Reg_date < '2022-01-01' AND Customer_Id NOT IN (SEL...	1 row(s) returned
47	19:06:20	SELECT Emp_name, Position FROM Employee WHERE Salary > 50000 LIMIT 0, 1000	4 row(s) returned

6. List the customer names who registered before 2022-01-01 and have not issued any books yet.

QUERY :

```
SELECT Customer_name  
FROM Customer  
WHERE Reg_date < '2022-01-01'  
AND Customer_Id NOT IN (SELECT Issued_cust FROM IssueStatus);
```

OUTPUT :

The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** Shows the database schema with the **library** schema selected.
- SQL Editor:** Contains the following SQL code:

```
152
153 •  SELECT Category, COUNT(*) AS Total_Count
154   FROM Books
155   GROUP BY Category;
156
157 •  SELECT Emp_name, Position
158   FROM Employee
159   WHERE Salary > 50000;
160
161 •  SELECT Customer_name
162   FROM Customer
163   WHERE Reg_date < '2022-01-01'
164   AND Customer_Id NOT IN (SELECT Issued_cust FROM IssueStatus);
165
166 •  INSERT INTO Customer (Customer_Id,Customer_name, Customer_address, Reg_date)
```
- Result Grid:** Displays the result of the last query, showing a single row: **Lipisha**.
- Output Window:** Shows the execution log for the session:

#	Time	Action	Message
43	18:58:58	SELECT DISTINCT Customer.Customer_name FROM Customer JOIN IssueStatus ON Customer.Customer_Id ...	1 row(s) returned
44	19:01:01	INSERT INTO Customer (Customer_Id, Customer_name, Customer_address, Reg_date) SELECT DISTINCT ...	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
45	19:01:28	SELECT Branch_no, COUNT(*) AS Total_Employees FROM Employee GROUP BY Branch_no LIMIT 0, 1000	6 row(s) returned
46	19:04:11	SELECT Customer_name FROM Customer WHERE Reg_date < '2022-01-01' AND Customer_Id NOT IN (SEL...	1 row(s) returned

7. Display the branch numbers and the total count of employees in each branch.

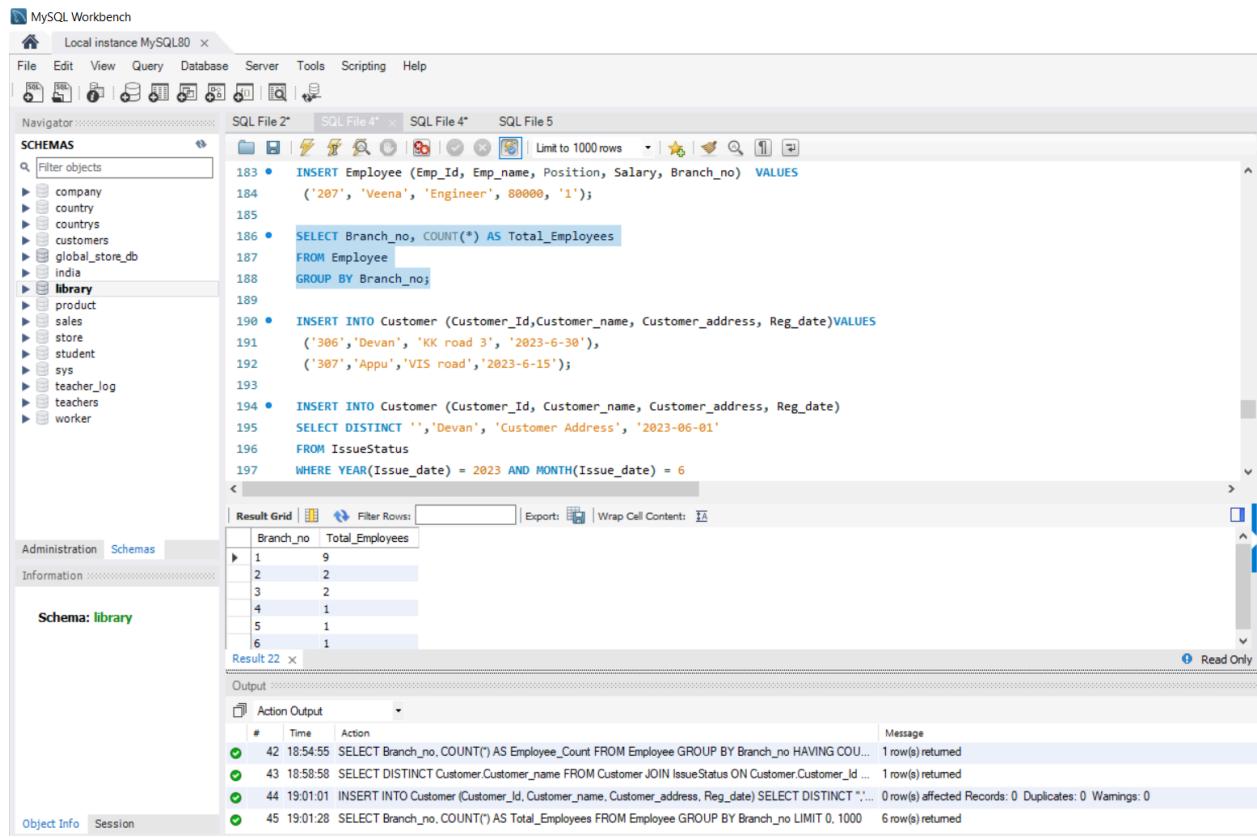
QUERY :

```

SELECT Branch_no, COUNT(*) AS Total_Employees
FROM Employee
GROUP BY Branch_no;

```

OUTPUT :



The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** Shows the database schema with the following tables: company, country, countrys, customers, global_store_db, india, library, product, sales, store, student, sys, teacher_log, teachers, worker.
- SQL Editor:** Contains the following SQL code:

```

183 • INSERT Employee (Emp_Id, Emp_name, Position, Salary, Branch_no) VALUES
184     ('207', 'Veeena', 'Engineer', 80000, '1');
185
186 • SELECT Branch_no, COUNT(*) AS Total_Employees
187     FROM Employee
188     GROUP BY Branch_no;
189
190 • INSERT INTO Customer (Customer_Id,Customer_name, Customer_address, Reg_date)VALUES
191     ('306','Devan', 'KK road 3', '2023-6-30'),
192     ('307','Appu', 'VIS road','2023-6-15');
193
194 • INSERT INTO Customer (Customer_Id, Customer_name, Customer_address, Reg_date)
195     SELECT DISTINCT '','Devan', 'Customer Address', '2023-06-01'
196     FROM IssueStatus
197     WHERE YEAR(Issue_date) = 2023 AND MONTH(Issue_date) = 6

```
- Result Grid:** Displays the results of the query in the SQL editor:

Branch_no	Total_Employees
1	9
2	2
3	2
4	1
5	1
6	1
- Output:** Shows the log of actions taken during the session:
 - 42 18:54:55 SELECT Branch_no, COUNT(*) AS Employee_Count FROM Employee GROUP BY Branch_no HAVING COU... 1 row(s) returned
 - 43 18:58:58 SELECT DISTINCT Customer.Customer_name FROM Customer JOIN IssueStatus ON Customer.Customer_Id ... 1 row(s) returned
 - 44 19:01:01 INSERT INTO Customer (Customer_Id, Customer_name, Customer_address, Reg_date) SELECT DISTINCT "... 0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
 - 45 19:01:28 SELECT Branch_no, COUNT(*) AS Total_Employees FROM Employee GROUP BY Branch_no LIMIT 0, 1000 6 row(s) returned

8. Display the names of customers who have issued books in the month of June 2023.

QUERY :

```
SELECT DISTINCT Customer.Customer_name  
FROM Customer  
JOIN IssueStatus ON Customer.Customer_Id = IssueStatus.Issued_cust  
WHERE YEAR(IssueStatus.Issue_date) = 2023 AND MONTH(IssueStatus.Issue_date) = 6;
```

OUTPUT :

```

165 • INSERT INTO IssueStatus (Issue_Id, Issued_cust, Issued_book_name, Issue_date, ISBN_book) VALUES
166     (404,306,'Buried Thoughts','2023-06-01','981');
167
168
169 • SELECT * FROM IssueStatus;
170
171 • SELECT DISTINCT Customer.Customer_name
172   FROM Customer
173   JOIN IssueStatus ON Customer.Customer_Id = IssueStatus.Issued_cust
174   WHERE YEAR(IssueStatus.Issue_date) = 2023 AND MONTH(IssueStatus.Issue_date) = 6;
175

```

Customer_name
Charlie Brown

Result Grid | Filter Rows: [] | Export: [] | Wrap Cell Content: []

Result 24 x

Output

#	Time	Action	Message
63	18:58:12	INSERT INTO IssueStatus (Issue_Id, Issued_cust, Issued_book_name, Issue_date, ISBN_book) VALUES (40...	Error Code: 1452. Cannot add or update a child row: a foreign key constraint fa...
64	18:58:52	SELECT DISTINCT Customer.Customer_name FROM Customer JOIN IssueStatus ON Customer.Customer_Id ...	0 row(s) returned
65	19:00:56	INSERT INTO IssueStatus (Issue_Id, Issued_cust, Issued_book_name, Issue_date, ISBN_book) VALUES (40...	Error Code: 1452. Cannot add or update a child row: a foreign key constraint fa...
66	19:01:46	INSERT INTO IssueStatus (Issue_Id, Issued_cust, Issued_book_name, Issue_date, ISBN_book) VALUES (404...	Error Code: 1452. Cannot add or update a child row: a foreign key constraint fa...
67	19:02:34	INSERT INTO IssueStatus (Issue_Id, Issued_cust, Issued_book_name, Issue_date, ISBN_book) VALUES (40...	Error Code: 1452. Cannot add or update a child row: a foreign key constraint fa...
68	19:04:10	UPDATE IssueStatus SET Issue_date = '2023-06-09' WHERE Issue_Id = '403'	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0
69	19:04:22	SELECT DISTINCT Customer.Customer_name FROM Customer JOIN IssueStatus ON Customer.Customer_Id ...	1 row(s) returned

9. Retrieve book_title from book table containing history.

QUERY :

SELECT Book_title

FROM Books

WHERE Category = 'History';

OUTPUT :

```

177 WHERE YEAR(IssueStatus.Issue_date) = 2023 AND MONTH(IssueStatus.Issue_date) = 6;
178
179 • SELECT Book_title
180   FROM Books
181   WHERE Category = 'History';
182
183
184 •   SELECT * FROM Customer;
185
186
187

```

Book_title
Modern History
Ancient History

Books 26 x

Action Output

#	Time	Action	Message
67	19:02:34	INSERT INTO IssueStatus (Issue_Id, Issued_cust, Issued_book_name, Issue_date, ISBN_book) VALUES (40...,	Error Code: 1452. Cannot add or update a child row: a foreign key constraint failed.
68	19:04:10	UPDATE IssueStatus SET Issue_date = '2023-06-09' WHERE Issue_Id = '403'	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0
69	19:04:22	SELECT DISTINCT Customer.Customer_name FROM Customer JOIN IssueStatus ON Customer.Customer_Id ...	1 row(s) returned
70	19:05:46	SELECT Book_title FROM Books WHERE Category = 'History' LIMIT 0, 1000	0 row(s) returned
71	19:09:27	INSERT INTO Books (ISBN, Book_title, Category, Rental_Price, Status, Author, Publisher) VALUES ('981', 'Mo...', 1 row(s) affected	
72	19:12:37	INSERT INTO Books (ISBN, Book_title, Category, Rental_Price, Status, Author, Publisher) VALUES ('982', 'An...', 1 row(s) affected	
73	19:12:46	SELECT Book_title FROM Books WHERE Category = 'History' LIMIT 0, 1000	2 row(s) returned

10. Retrieve the branch numbers along with the count of employees for branches having more than 5 employees

QUERY :

```
SELECT Branch_no, COUNT(*) AS Employee_Count
```

```
FROM Employee
```

```
GROUP BY Branch_no
```

```
HAVING COUNT(*) > 5;
```

OUTPUT :

The screenshot shows the MySQL Workbench interface. In the top navigation bar, 'Local instance MySQL80' is selected. The 'Schemas' pane on the left shows the 'library' schema is currently selected. The main area displays a SQL editor with the following code:

```
208     JOIN IssueStatus ON Customer.Customer_Id = IssueStatus.Issued_cust
209     WHERE YEAR(IssueStatus.Issue_date) = 2023 AND MONTH(IssueStatus.Issue_date) = 6;
210
211 •   SELECT Book_title
212     FROM Books
213     WHERE Category = 'History';
214
215 •   SELECT Branch_no, COUNT(*) AS Employee_Count
216     FROM Employee
217     GROUP BY Branch_no
218     HAVING COUNT(*) > 5;
219
220 •   SELECT Employee.Emp_name, Branch.Branch_address
221     FROM Employee
222     JOIN Branch ON Employee.Fan_Id = Branch.Manager_Id;
```

The 'Result Grid' tab is active, showing the output of the second query:

Branch_no	Employee_Count
1	9

The 'Result 20' tab shows the history of actions:

#	Time	Action	Message
39	18:47:09	SELECT * FROM BRANCH LIMIT 0, 1000	13 row(s) returned
40	18:48:18	SELECT Employee.Emp_name, Branch.Branch_address FROM Employee JOIN Branch ON Employee.Emp_Id = Branch.Manager_Id;	0 row(s) returned
41	18:48:52	SELECT Employee.Emp_name, Branch.Branch_address FROM Employee JOIN Branch ON Employee.Emp_Id = Branch.Manager_Id;	13 row(s) returned
42	18:54:55	SELECT Branch_no, COUNT(*) AS Employee_Count FROM Employee GROUP BY Branch_no HAVING COU...	1 row(s) returned

11. Retrieve the names of employees who manage branches and their respective branch addresses.

QUERY :

```
SELECT Employee.Emp_name, Branch.Branch_address
FROM Employee
JOIN Branch ON Employee.Emp_Id = Branch.Manager_Id;
```

OUTPUT :

The screenshot shows the MySQL Workbench interface with three tabs open: SQL File 2*, SQL File 4*, and SQL File 5. The SQL File 4* tab contains the following query:

```
215 •    SELECT Branch_no, COUNT(*) AS Employee_Count
216     FROM Employee
217     GROUP BY Branch_no
218     HAVING COUNT(*) > 5;
219
220 •    SELECT Employee.Emp_name, Branch.Branch_address
221     FROM Employee
222     JOIN Branch ON Employee.Emp_Id = Branch.Manager_Id;
223
224 •    SELECT DISTINCT Customer.Customer_name
225     FROM Customer
226     JOIN IssueStatus ON Customer.Customer_Id = IssueStatus.Issued_cust
227     JOIN Books ON IssueStatus.Issue_Status = Books.ISBN
228     WHERE Books.Rental_Price > 25;
229
```

The Result Grid shows the output of the second query:

Emp_name	Branch_address
John Doe	123 Main St
Jane Smith	456 Elm St
Alice Johnson	789 Oak St
Bob Williams	989 Vis gh
Michael Brown	565 JKJ
Freda Turner	R441 Fn

The Output pane shows the log of actions:

#	Time	Action	Message
37	18:42:28	UPDATE Branch SET Manager_Id = 212 WHERE Branch_no = 12	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0
38	18:46:39	UPDATE Branch SET Manager_Id = 213 WHERE Branch_no = 13	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0
39	18:47:09	SELECT * FROM BRANCH LIMIT 0, 1000	13 row(s) returned
40	18:48:18	SELECT Employee.Emp_name, Branch.Branch_address FROM Employee JOIN Branch ON Employee.Emp_Id = Branch.Manager_Id;	0 row(s) returned
41	18:48:52	SELECT Employee.Emp_name, Branch.Branch_address FROM Employee JOIN Branch ON Employee.Emp_Id = Branch.Manager_Id;	13 row(s) returned

12. Display the names of customers who have issued books with a rental price higher than Rs. 25.

QUERY :

```
SELECT DISTINCT Customer.Customer_name
```

FROM Customer

JOIN IssueStatus ON Customer.Customer_Id = IssueStatus.Issued_cust

JOIN Books ON IssueStatus.Isbn_book = Books.ISBN

WHERE Books.Rental_Price > 25;

OUTPUT :

The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** Shows the database schema with tables like Employee, Branch, Customer, IssueStatus, Books, and Returnstatus, along with their foreign keys and triggers.
- SQL Editor:** Contains the following SQL code:

```
223 • SELECT Employee.Emp_name, Branch.Branch_address
224   FROM Employee
225   JOIN Branch ON Employee.Emp_Id = Branch.Manager_Id;
226
227
228
229 • SELECT DISTINCT Customer.Customer_name
230   FROM Customer
231   JOIN IssueStatus ON Customer.Customer_Id = IssueStatus.Issued_cust
232   JOIN Books ON IssueStatus.Isbn_book = Books.ISBN
233   WHERE Books.Rental_Price > 25;
234
```
- Result Grid:** Displays the result of the query, showing a single row:

Customer_name
Veena
- Result 6:** Shows the message "1 row(s) returned".
- Information Panel:** Shows the structure of the "returnstatus" table, which has columns: Return_Id (int), Return_cust (int), Return_book_name (varchar), Return_date (date), and ISBN_book2 (varchar).

