

```

1  -- Create database
2  • CREATE DATABASE library;
3
4  -- Use the database
5  • USE library;
6
7  -- Create Branch table
8  • CREATE TABLE Branch (
9      Branch_no INT PRIMARY KEY,
10     Manager_Id INT,
11     Branch_address VARCHAR(255),
12     Contact_no VARCHAR(15)
13 );
14
15

```

Output ::::

Action Output

#	Time	Action	Message
✓ 1	11:00:14	CREATE DATABASE library	1 row(s) affected
✓ 2	11:00:17	USE library	0 row(s) affected
✓ 3	11:22:14	CREATE TABLE Branch (Branch_no INT PRIMARY KEY, Manager_Id INT, B...	0 row(s) affected

```

15  -- Create Employee table
16  • CREATE TABLE Employee (
17      Emp_Id INT PRIMARY KEY,
18      Emp_name VARCHAR(255),
19      Position VARCHAR(50),
20      Salary DECIMAL(10, 2),
21      Branch_no INT,
22      FOREIGN KEY (Branch_no) REFERENCES Branch(Branch_no)
23 );

```

Output ::::

Action Output

#	Time	Action	Message
✓ 1	11:00:14	CREATE DATABASE library	1 row(s) affected
✓ 2	11:00:17	USE library	0 row(s) affected
✓ 3	11:22:14	CREATE TABLE Branch (Branch_no INT PRIMARY KEY, Manager_Id INT, B...	0 row(s) affected
✓ 4	11:23:55	CREATE TABLE Employee (Emp_Id INT PRIMARY KEY, Emp_name VARCHA...	0 row(s) affected

```

25  -- Create Books table
26  ● CREATE TABLE Books (
27      ISBN INT PRIMARY KEY,
28      Book_title VARCHAR(255),
29      Category VARCHAR(50),
30      Rental_Price DECIMAL(10, 2),
31      Status ENUM('yes', 'no'),
32      Author VARCHAR(255),
33      Publisher VARCHAR(255)
34  );
35

```

Output

Action Output

#	Time	Action	Message
✓ 2	11:00:17	USE library	0 row(s) affected
✓ 3	11:22:14	CREATE TABLE Branch (Branch_no INT PRIMARY KEY, Manager_Id INT, ...	0 row(s) affected
✓ 4	11:23:55	CREATE TABLE Employee (Emp_Id INT PRIMARY KEY, Emp_name VARCH...	0 row(s) affected
✓ 5	11:24:20	CREATE TABLE Books (ISBN INT PRIMARY KEY, Book_title VARCHAR(255)...	0 row(s) affected

```

36  -- Create Customer table
37  ● CREATE TABLE Customer (
38      Customer_Id INT PRIMARY KEY,
39      Customer_name VARCHAR(255),
40      Customer_address VARCHAR(255),
41      Reg_date DATE
42  );
43
44
45
46
47

```

Output

Action Output

#	Time	Action	Message
✓ 3	11:22:14	CREATE TABLE Branch (Branch_no INT PRIMARY KEY, Manager_Id INT, ...	0 row(s) affected
✓ 4	11:23:55	CREATE TABLE Employee (Emp_Id INT PRIMARY KEY, Emp_name VARCH...	0 row(s) affected
✓ 5	11:24:20	CREATE TABLE Books (ISBN INT PRIMARY KEY, Book_title VARCHAR(255)...	0 row(s) affected
✓ 6	11:24:55	CREATE TABLE Customer (Customer_Id INT PRIMARY KEY, Customer_name...	0 row(s) affected

```

44  -- Create IssueStatus table
45  CREATE TABLE IssueStatus (
46      Issue_Id INT PRIMARY KEY,
47      Issued_cust INT,
48      Issued_book_name VARCHAR(255),
49      Issue_date DATE,
50      Isbn_book INT,
51      FOREIGN KEY (Issued_cust) REFERENCES Customer(Customer_Id),
52      FOREIGN KEY (Isbn_book) REFERENCES Books(ISBN)
53  );
54

```

Output

Action Output

#	Time	Action	Message
✓ 4	11:23:55	CREATE TABLE Employee (Emp_Id INT PRIMARY KEY, Emp_name VARCH...	0 row(s) affected
✓ 5	11:24:20	CREATE TABLE Books (ISBN INT PRIMARY KEY, Book_title VARCHAR(255)...	0 row(s) affected
✓ 6	11:24:55	CREATE TABLE Customer (Customer_Id INT PRIMARY KEY, Customer_name...	0 row(s) affected
✓ 7	11:25:35	CREATE TABLE IssueStatus (Issue_Id INT PRIMARY KEY, Issued_cust INT, ...	0 row(s) affected

```

55  -- Create ReturnStatus table
56  CREATE TABLE ReturnStatus (
57      Return_Id INT PRIMARY KEY,
58      Return_cust INT,
59      Return_book_name VARCHAR(255),
60      Return_date DATE,
61      Isbn_book2 INT,
62      FOREIGN KEY (Isbn_book2) REFERENCES Books(ISBN)
63  );
64

```

Output

Action Output

#	Time	Action	Message
✓ 5	11:24:20	CREATE TABLE Books (ISBN INT PRIMARY KEY, Book_title VARCHAR(255)...	0 row(s) affected
✓ 6	11:24:55	CREATE TABLE Customer (Customer_Id INT PRIMARY KEY, Customer_name...	0 row(s) affected
✓ 7	11:25:35	CREATE TABLE IssueStatus (Issue_Id INT PRIMARY KEY, Issued_cust INT, ...	0 row(s) affected
✓ 8	11:26:06	CREATE TABLE ReturnStatus (Return_Id INT PRIMARY KEY, Return_cust IN...	0 row(s) affected

```

65  -- Insert sample data into Branch table
66  • INSERT INTO Branch (Branch_no, Manager_Id, Branch_address, Contact_no)
67    VALUES
68      (1, 101, 'Thrissur', '1234567891'),
69      (2, 102, 'Ernakulam', '2345895432');
70  • SELECT * FROM Branch;

```

Result Grid	Filter Rows:	Edit:	Export/Import:	Wrap Cell Content:	Result Grid
Branch_no	Manager_Id	Branch_address	Contact_no		
1	101	Thrissur	1234567891		
2	102	Ernakulam	2345895432		
NULL	NULL	NULL	NULL		

Branch 1 x Apply Revert

#	Time	Action	Message
7	11:25:35	CREATE TABLE IssueStatus (Issue_Id INT PRIMARY KEY, Issued_cust INT...	0 row(s) affected
8	11:26:06	CREATE TABLE ReturnStatus (Return_Id INT PRIMARY KEY, Return_cust I...	0 row(s) affected
9	11:26:47	INSERT INTO Branch (Branch_no, Manager_Id, Branch_address, Contact_no) VA...	2 row(s) affected Records: 2 Duplicates: 0 Warnings: 0
10	11:27:22	SELECT * FROM Branch LIMIT 0, 1000	2 row(s) returned

```

72  -- Insert sample data into Employee table
73  • INSERT INTO Employee (Emp_Id, Emp_name, Position, Salary, Branch_no)
74    VALUES
75      (101, 'Seetha Swaminathan', 'Manager', 60000.00, 1),
76      (102, 'Fayas Abdul Wahab', 'Manager', 70000.00, 2),
77      (103, 'Jose K Alex', 'Clerk', 40000.00, 1),
78      (104, 'Aneena Chacko', 'Clerk', 45000.00, 2);
79  • SELECT * FROM Employee;

```

Result Grid	Filter Rows:	Edit:	Export/Import:	Wrap Cell Content:	Result Grid
Emp_Id	Emp_name	Position	Salary	Branch_no	
101	Seetha Swaminathan	Manager	60000.00	1	
102	Fayas Abdul Wahab	Manager	70000.00	2	
103	Jose K Alex	Clerk	40000.00	1	
104	Aneena Chacko	Clerk	45000.00	2	
NULL	NULL	NULL	NULL	NULL	

Employee 2 x Apply Revert

#	Time	Action	Message
9	11:26:47	INSERT INTO Branch (Branch_no, Manager_Id, Branch_address, Contact_no) VA...	2 row(s) affected Records: 2 Duplicates: 0 Warnings: 0
10	11:27:22	SELECT * FROM Branch LIMIT 0, 1000	2 row(s) returned
11	11:28:05	INSERT INTO Employee (Emp_Id, Emp_name, Position, Salary, Branch_no) VALU...	4 row(s) affected Records: 4 Duplicates: 0 Warnings: 0
12	11:28:21	SELECT * FROM Employee LIMIT 0, 1000	4 row(s) returned

```

81  -- Insert sample data into Books table
82  • INSERT INTO Books (ISBN, Book_title, Category, Rental_Price, Status, Author, Publisher)
83  VALUES
84      (1, 'Life Is Beautiful', 'Fiction', 20.00, 'yes', 'Christy Jose', 'ABC Publications'),
85      (2, 'How To Impress Yourself', 'Non-Fiction', 22.50, 'yes', 'Keerthana Menon', 'BayLeaf Publications'),
86      (3, '1947', 'History', 15.00, 'no', 'Ramji Rao', 'XYZ Publications'),
87      (4, 'Solar System - An Overview', 'Science', 11.00, 'yes', 'Franklin James', 'Something New Publicatio
88      (5, 'The Little Butterfly', 'Fiction', 20.00, 'yes', 'Hari Prasad', 'ABC Publications');
89  • SELECT * FROM Books;

```

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content:

	ISBN	Book_title	Category	Rental_Price	Status	Author	Publisher
▶	1	Life Is Beautiful	Fiction	20.00	yes	Christy Jose	ABC Publications
	2	How To Impress Yourself	Non-Fiction	22.50	yes	Keerthana Menon	BayLeaf Publications
	3	1947	History	15.00	no	Ramji Rao	XYZ Publications
	4	Solar System - An Overview	Science	11.00	yes	Franklin James	Something New Publications
	5	The Little Butterfly	Fiction	20.00	yes	Hari Prasad	ABC Publications
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Books 4

Apply

Revert

Output

Action Output

#	Time	Action	Message
✓ 1	11:30:17	SELECT * FROM Books LIMIT 0, 1000	5 row(s) returned

```

91  -- Insert sample data into Customer table
92  • INSERT INTO Customer (Customer_Id, Customer_name, Customer_address, Reg_date)
93  VALUES
94      (201, 'Sanjay Sankar', '789 Mg Road', '2021-12-15'),
95      (202, 'Lakshmi Rajagopal', '567 Gandhi Nagar', '2022-02-20');
96  • SELECT * FROM Customer;
97

```

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content:

	Customer_Id	Customer_name	Customer_address	Reg_date
▶	201	Sanjay Sankar	789 Mg Road	2021-12-15
	202	Lakshmi Rajagopal	567 Gandhi Nagar	2022-02-20
*	NULL	NULL	NULL	NULL

Customer 5

Apply

Revert

Output

Action Output

#	Time	Action	Message
✓ 1	11:30:17	SELECT * FROM Books LIMIT 0, 1000	5 row(s) returned
✓ 2	11:31:02	INSERT INTO Customer (Customer_Id, Customer_name, Customer_address, Reg_da...	2 row(s) affected Records: 2 Duplicates: 0 Warnings: 0
✓ 3	11:31:15	SELECT * FROM Customer LIMIT 0, 1000	2 row(s) returned

```

99  -- Insert sample data into IssueStatus table
100 • INSERT INTO IssueStatus (Issue_Id, Issued_cust, Issued_book_name, Issue_date, Isbn_book)
101  VALUES
102      (301, 201, 'Life Is Beautiful', '2023-06-05', 1),
103      (302, 202, 'How To Impress Yourself', '2023-06-10', 2);
104 • SELECT * FROM IssueStatus;
105

```

Result Grid					
	Issue_Id	Issued_cust	Issued_book_name	Issue_date	Isbn_book
▶	301	201	Life Is Beautiful	2023-06-05	1
	302	202	How To Impress Yourself	2023-06-10	2
•	NULL	NULL	NULL	NULL	NULL

IssueStatus 6 x Apply Revert

Output :.....

#	Time	Action	Message
✓ 3	11:31:15	SELECT * FROM Customer LIMIT 0, 1000	2 row(s) returned
✓ 4	11:32:00	INSERT INTO IssueStatus (Issue_Id, Issued_cust, Issued_book_name, Issue_date...	2 row(s) affected Records: 2 Duplicates: 0 Warnings: 0
✓ 5	11:32:17	SELECT * FROM IssueStatus LIMIT 0, 1000	2 row(s) returned

```

106  -- Insert sample data into ReturnStatus table
107 • INSERT INTO ReturnStatus (Return_Id, Return_cust, Return_book_name, Return_date, Isbn_book2)
108  VALUES
109      (401, 201, 'Life Is Beautiful', '2023-06-25', 1),
110      (402, 202, 'How To Impress Yourself', '2023-06-30', 2);
111 • SELECT * FROM ReturnStatus;

```

Result Grid					
	Return_Id	Return_cust	Return_book_name	Return_date	Isbn_book2
▶	401	201	Life Is Beautiful	2023-06-25	1
	402	202	How To Impress Yourself	2023-06-30	2
•	NULL	NULL	NULL	NULL	NULL

ReturnStatus 7 x Apply Revert

Output :.....

#	Time	Action	Message
✓ 5	11:32:17	SELECT * FROM IssueStatus LIMIT 0, 1000	2 row(s) returned
✓ 6	11:33:00	INSERT INTO ReturnStatus (Return_Id, Return_cust, Return_book_name, Return...	2 row(s) affected Records: 2 Duplicates: 0 Warnings: 0
✓ 7	11:33:17	SELECT * FROM ReturnStatus LIMIT 0, 1000	2 row(s) returned

```

112
113 -- 1. Retrieve the book title, category, and rental price of all available books.
114 • SELECT Book_title, Category, Rental_Price FROM Books
115 WHERE Status = 'yes';
116

```

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

	Book_title	Category	Rental_Price
▶	Life Is Beautiful	Fiction	20.00
	How To Impress Yourself	Non-Fiction	22.50
	Solar System - An Overview	Science	11.00
	The Little Butterfly	Fiction	20.00

Result Grid
Form Editor

```

116
117 -- 2. List the employee names and their respective salaries in descending order of salary.
118 • SELECT Emp_name, Salary FROM Employee
119 ORDER BY Salary DESC;
120

```

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

	Emp_name	Salary
▶	Fayas Abdul Wahab	70000.00
	Seetha Swaminathan	60000.00
	Aneena Chacko	45000.00
	Jose K Alex	40000.00

Result Grid
Form Editor

```

121 -- 3. Retrieve the book titles and the corresponding customers who have issued those books.
122 • SELECT Issued_book_name, Customer_name
123 FROM IssueStatus INNER JOIN Customer ON IssueStatus.Issued_cust = Customer.Customer_Id;
124

```

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

	Issued_book_name	Customer_name
▶	Life Is Beautiful	Sanjay Sankar
	How To Impress Yourself	Lakshmi Rajagopal

Result Grid
Form

```

125 -- 4. Display the total count of books in each category.
126 • SELECT Category, COUNT(*) AS Total_Count FROM Books
127 GROUP BY Category;
128

```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
Category	Total_Count		
Fiction	2		
Non-Fiction	1		
History	1		
Science	1		

```

130 -- 5. Retrieve the employee names and their positions for the employees
131 -- whose salaries are above Rs.50,000.
132 • SELECT Emp_name, Position FROM Employee
133 WHERE Salary > 50000;
134

```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
Emp_name	Position		
Seetha Swaminathan	Manager		
Fayas Abdul Wahab	Manager		

```

134
135 -- 6. List the customer names who registered before 2022-01-01 and have not issued any books yet.
136 -- List the customer names who registered before 2022-01-01 and have not issued any books yet
137 • SELECT c.Customer_name
138 FROM Customer c LEFT JOIN IssueStatus i
139 ON c.Customer_Id = i.Issued_cust
140 WHERE c.Reg_date < '2022-01-01' AND i.Issue_Id IS NULL;
141
142

```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
Customer_name			


```
141
142 -- 7. Display the branch numbers and the total count of employees in each branch.
143 • SELECT Branch_no, COUNT(*) AS Total_Employees FROM Employee
144 GROUP BY Branch_no;
145
```

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

	Branch_no	Total_Employees
▶	1	2
	2	2

 Result Grid

 Form Editor

```
145
146 -- 8. Display the names of customers who have issued books in the month of June 2023.
147 • SELECT DISTINCT c.Customer_name
148 FROM Customer c INNER JOIN IssueStatus i
149 ON c.Customer_Id = i.Issued_cust
150 WHERE YEAR(i.Issue_date) = 2023 AND MONTH(i.Issue_date) = 6;
151
152
```




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

	Customer_name
▶	Sanjay Sankar
	Lakshmi Rajagopal

 Result Grid

 Form Editor

```
152 -- 9. Retrieve book_title from book table containing history.
153 • SELECT Book_title FROM Books
154 WHERE Category = 'History';
155
```

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

	Book_title
▶	1947

 Result Grid

 Form Editor

```

155
156 -- 9. Retrieve book_title from book table containing the word "history" in the Book_title
157 • SELECT Book_title FROM Books
158 WHERE Book_title LIKE '%history%';
159

```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
Book_title			

```

160 -- 10. Retrieve the branch numbers along with the count of employees for branches having
161 -- more than 5 employees
162 • SELECT Branch_no, COUNT(*) AS Total_Employees FROM Employee
163 GROUP BY Branch_no
164 HAVING COUNT(*) > 5;
165

```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
Branch_no	Total_Employees		



```

165
166 -- 11. Retrieve the names of employees who manage branches and their respective branch addresses.
167 • SELECT e.Emp_name, b.Branch_address
168 FROM Employee e INNER JOIN Branch b
169 ON e.Branch_no = b.Branch_no
170 WHERE e.Position = 'Manager';
171
172

```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
Emp_name	Branch_address		
Seetha Swaminathan	Thrissur		
Fayas Abdul Wahab	Ernakulam		

```
172 -- 12. Display the names of customers who have issued books with a rental price higher than Rs. 25.  
173 -- Display the names of customers who have issued books with a rental price higher than Rs. 25  
174 • SELECT DISTINCT c.Customer_name FROM Customer c  
175 WHERE (  
176     SELECT i.Issue_Id  
177     FROM IssueStatus i INNER JOIN Books b ON i.Isbn_book = b.ISBN  
178     WHERE i.Issued_cust = c.Customer_Id AND b.Rental_Price > 25  
179 );  
180
```

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

Customer_name


Result
Grid


Form