

Module 2: 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

- a. Branch_no - Set as PRIMARY KEY
- b. Manager_Id
- c. Branch_address
- d. Contact_no

```
CREATE TABLE Branch (  
  Branch_no INT PRIMARY KEY,  
  Manager_Id INT,  
  Branch_address VARCHAR(255),  
  Contact_no VARCHAR(15)  
);
```

The screenshot displays a database management interface. The left sidebar shows a tree view of databases, including 'library'. The main window shows the SQL editor with the following script:

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

The 'Result Grid' shows the table structure for 'Branch':

Field	Type	Null	Key	Default	Extra
Branch_no	int	NO	PRI		
Manager_Id	int	YES			
Branch_address	varchar(255)	YES			
Contact_no	varchar(15)	YES			

The 'Output' pane shows the execution results of the SQL script:

#	Time	Action	Message	Duration / Fetch
60	11:04:30	DELETE FROM teachers WHERE name = 'Sbin Xavier'	0 row(s) affected	0.000 sec
61	11:04:37	SELECT * FROM teacher_log LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
62	11:10:14	USE library	0 row(s) affected	0.000 sec
63	11:26:34	CREATE TABLE Branch (Branch_no INT PRIMARY KEY, Manager_Id INT, Bran...	0 row(s) affected	0.016 sec
64	11:26:37	CREATE TABLE Employee (Emp_Id INT PRIMARY KEY, Emp_name VARCHAR(25...	0 row(s) affected	0.016 sec
65	11:26:41	CREATE TABLE Books (ISBN INT PRIMARY KEY, Book_title VARCHAR(255), C...	0 row(s) affected	0.016 sec
66	11:26:45	CREATE TABLE Customer (Customer_Id INT PRIMARY KEY, Customer_name VAR...	0 row(s) affected	0.016 sec
67	11:26:48	CREATE TABLE IssueStatus (Issue_Id INT PRIMARY KEY, Issued_cust INT, Iss...	0 row(s) affected	0.031 sec
68	11:26:51	CREATE TABLE ReturnStatus (Return_Id INT PRIMARY KEY, Return_cust INT, ...	0 row(s) affected	0.031 sec
69	11:27:13	desc ReturnStatus	5 row(s) returned	0.016 sec / 0.000 sec
70	11:27:30	desc ReturnStatus	5 row(s) returned	0.000 sec / 0.000 sec
71	11:32:08	desc Branch	4 row(s) returned	0.000 sec / 0.000 sec

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

```
CREATE TABLE Employee (
    Emp_Id INT PRIMARY KEY,
    Emp_name VARCHAR(255),
    Position VARCHAR(255),
    Salary DECIMAL(10, 2),
    Branch_no INT,
    FOREIGN KEY (Branch_no) REFERENCES Branch(Branch_no)
);
```

The screenshot shows a database management tool interface. On the left, a 'SCHEMAS' pane lists various databases including 'd32', 'employee', 'library', 'product', 'sales', 'school', 'sys', 'testdb', and 'testdb32'. The 'library' database is selected, showing its internal structure with 'Tables', 'Views', 'Stored Procedures', and 'Functions'. The main editor displays SQL code for creating the 'Employee' table and a command to describe it. Below the editor, the 'Result Grid' shows the table's structure with columns: Field, Type, Null, Key, Default, and Extra. The 'Output' pane at the bottom shows a log of database actions, including the successful creation of the 'Employee' table and its description.

```
-- Employee table
15
16 CREATE TABLE Employee (
17     Emp_Id INT PRIMARY KEY,
18     Emp_name VARCHAR(255),
19     Position VARCHAR(255),
20     Salary DECIMAL(10, 2),
21     Branch_no INT,
22     FOREIGN KEY (Branch_no) REFERENCES Branch(Branch_no)
23 );
24
25 desc Employee;
```

Field	Type	Null	Key	Default	Extra
Emp_Id	int	NO	PRI		
Emp_name	varchar(255)	YES			
Position	varchar(255)	YES			
Salary	decimal(10,2)	YES			
Branch_no	int	YES	MUL		

Result 4 x

#	Time	Action	Message	Duration / Fe
61	11:04:37	SELECT * FROM teacher_log LIMIT 0, 1000	1 row(s) returned	0.000 sec / C
62	11:10:14	USE library	0 row(s) affected	0.000 sec
63	11:26:34	CREATE TABLE Branch (Branch_no INT PRIMARY KEY, Manager_Id INT, Bran...	0 row(s) affected	0.016 sec
64	11:26:37	CREATE TABLE Employee (Emp_Id INT PRIMARY KEY, Emp_name VARCHAR(25...	0 row(s) affected	0.016 sec
65	11:26:41	CREATE TABLE Books (ISBN INT PRIMARY KEY, Book_title VARCHAR(255), C...	0 row(s) affected	0.016 sec
66	11:26:45	CREATE TABLE Customer (Customer_Id INT PRIMARY KEY, Customer_name VAR...	0 row(s) affected	0.016 sec
67	11:26:48	CREATE TABLE IssueStatus (Issue_Id INT PRIMARY KEY, Issued_cust INT, Iss...	0 row(s) affected	0.031 sec
68	11:26:51	CREATE TABLE ReturnStatus (Return_Id INT PRIMARY KEY, Return_cust INT, ...	0 row(s) affected	0.031 sec
69	11:27:13	desc ReturnStatus	5 row(s) returned	0.016 sec / C
70	11:27:30	desc ReturnStatus	5 row(s) returned	0.000 sec / C
71	11:32:08	desc Branch	4 row(s) returned	0.000 sec / C
72	11:35:55	desc Employee	5 row(s) returned	0.000 sec / C

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

```
CREATE TABLE Books (
    ISBN INT PRIMARY KEY,
    Book_title VARCHAR(255),
    Category VARCHAR(255),
    Rental_Price DECIMAL(10, 2),
    Status ENUM('yes', 'no'),
    Author VARCHAR(255),
    Publisher VARCHAR(255)
);
```

The screenshot shows the SQL Enterprise Manager interface. The left pane displays the 'SCHEMAS' tree with a filter on 'library'. The main pane shows the SQL script for creating the 'Books' table. Below the script, the 'Result Grid' shows the table's schema, and the 'Output' pane shows the execution results.

SQL Script:

```
-- Books table
CREATE TABLE Books (
    ISBN INT PRIMARY KEY,
    Book_title VARCHAR(255),
    Category VARCHAR(255),
    Rental_Price DECIMAL(10, 2),
    Status ENUM('yes', 'no'),
    Author VARCHAR(255),
    Publisher VARCHAR(255)
);
```

Result Grid:

Field	Type	Null	Key	Default	Extra
ISBN	int	NO	PRI	NULL	
Book_title	varchar(255)	YES		NULL	
Category	varchar(255)	YES		NULL	
Rental_Price	decimal(10,2)	YES		NULL	
Status	enum('yes','no')	YES		NULL	

Output:

#	Time	Action	Message	Duration / Fetch
62	11:10:14	USE library	0 row(s) affected	0.000 sec
63	11:26:34	CREATE TABLE Branch (Branch_no INT PRIMARY KEY, Manager_Id INT, Bran...	0 row(s) affected	0.016 sec
64	11:26:37	CREATE TABLE Employee (Emp_Id INT PRIMARY KEY, Emp_name VARCHAR(25...	0 row(s) affected	0.016 sec
65	11:26:41	CREATE TABLE Books (ISBN INT PRIMARY KEY, Book_title VARCHAR(255), C...	0 row(s) affected	0.016 sec
66	11:26:45	CREATE TABLE Customer (Customer_Id INT PRIMARY KEY, Customer_name VAR...	0 row(s) affected	0.016 sec
67	11:26:48	CREATE TABLE IssueStatus (Issue_Id INT PRIMARY KEY, Issued_cust INT, Iss...	0 row(s) affected	0.031 sec
68	11:26:51	CREATE TABLE ReturnStatus (Return_Id INT PRIMARY KEY, Return_cust INT, ...	0 row(s) affected	0.031 sec
69	11:27:13	desc ReturnStatus	5 row(s) returned	0.016 sec / 0.000 sec
70	11:27:30	desc ReturnStatus	5 row(s) returned	0.000 sec / 0.000 sec
71	11:32:08	desc Branch	4 row(s) returned	0.000 sec / 0.000 sec
72	11:35:55	desc Employee	5 row(s) returned	0.000 sec / 0.000 sec
73	11:37:44	Desc Books	7 row(s) returned	0.016 sec / 0.000 sec

4. Customer

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

```

CREATE TABLE Customer (
    Customer_Id INT PRIMARY KEY,
    Customer_name VARCHAR(255),
    Customer_address VARCHAR(255),
    Reg_date DATE
);

```

The screenshot shows a SQL IDE with the following components:

- Navigator:** A tree view on the left showing a database schema with tables, views, stored procedures, and functions. The 'library' table is selected.
- SQL File 4:** The main editor window showing the SQL code for creating the 'Customer' table and a comment for the 'IssueStatus' table.
- Result Grid:** A table showing the structure of the 'Customer' table:

Field	Type	Null	Key	Default	Extra
Customer_Id	int	NO	PRI	NULL	
Customer_name	varchar(255)	YES	NO	NULL	
Customer_address	varchar(255)	YES	NO	NULL	
Reg_date	date	YES	NO	NULL	
- Output:** A log of SQL execution results showing 14 commands and their outcomes, including table creation and description commands.

5. IssueStatus

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

```

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

```

The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the 'testdb32' database structure. The central pane shows the SQL script for creating the 'IssueStatus' table. The right pane displays the 'Result Grid' for the 'desc IssueStatus' command, showing the table's schema. Below the result grid, the 'Output' pane shows the execution log, including the creation of several tables and the execution of 'desc' commands for various tables.

```

50 -- IssueStatus table
51 CREATE TABLE IssueStatus (
52     Issue_Id INT PRIMARY KEY,
53     Issued_cust INT,
54     Issued_book_name VARCHAR(255),
55     Issue_date DATE,
56     Isbn_book INT,
57     FOREIGN KEY (Issued_cust) REFERENCES Customer(Customer_Id),
58     FOREIGN KEY (Isbn_book) REFERENCES Books(ISBN)
59 );
60 desc IssueStatus;
61
62 -- ReturnStatus table

```

Field	Type	Null	Key	Default	Extra
Issue_Id	int	NO	PRI	NULL	
Issued_cust	int	YES	MUL	NULL	
Issued_book_name	varchar(255)	YES		NULL	
Issue_date	date	YES		NULL	
Isbn_book	int	YES	MUL	NULL	

#	Time	Action	Message	Duration / Fetch
65	11:26:41	CREATE TABLE Books (ISBN INT PRIMARY KEY, Book_title VARCHAR(255), C...	0 row(s) affected	0.016 sec
66	11:26:45	CREATE TABLE Customer (Customer_Id INT PRIMARY KEY, Customer_name VAR...	0 row(s) affected	0.016 sec
67	11:26:48	CREATE TABLE IssueStatus (Issue_Id INT PRIMARY KEY, Issued_cust INT, Iss...	0 row(s) affected	0.031 sec
68	11:26:51	CREATE TABLE ReturnStatus (Return_Id INT PRIMARY KEY, Return_cust INT, ...	0 row(s) affected	0.031 sec
69	11:27:13	desc ReturnStatus	5 row(s) returned	0.016 sec / 0.000 sec
70	11:27:30	desc ReturnStatus	5 row(s) returned	0.000 sec / 0.000 sec
71	11:32:08	desc Branch	4 row(s) returned	0.000 sec / 0.000 sec
72	11:35:55	desc Employee	5 row(s) returned	0.000 sec / 0.000 sec
73	11:37:44	Desc Books	7 row(s) returned	0.016 sec / 0.000 sec
74	11:39:52	desc Customer	4 row(s) returned	0.000 sec / 0.000 sec
75	11:41:40	desc IssueStatus	5 row(s) returned	0.000 sec / 0.000 sec
76	11:42:11	desc IssueStatus	5 row(s) returned	0.000 sec / 0.000 sec

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

```

CREATE TABLE ReturnStatus (
    Return_Id INT PRIMARY KEY,
    Return_cust INT,
    Return_book_name VARCHAR(255),
    Return_date DATE,
    Isbn_book2 INT,
    FOREIGN KEY (Return_cust) REFERENCES Customer(Customer_Id),
    FOREIGN KEY (Isbn_book2) REFERENCES Books(ISBN)
);

```

Navigation pane on the left shows a database schema with tables like `employee`, `library`, `product`, `sales`, `school`, `sys`, `testdb`, and `testdb32`.

```

62 -- ReturnStatus table
63 CREATE TABLE ReturnStatus (
64     Return_Id INT PRIMARY KEY,
65     Return_cust INT,
66     Return_book_name VARCHAR(255),
67     Return_date DATE,
68     isbn_book2 INT,
69     FOREIGN KEY (Return_cust) REFERENCES Customer(Customer_Id),
70     FOREIGN KEY (isbn_book2) REFERENCES Books(ISBN)
71 );
72
73 desc ReturnStatus;
74

```

Below the code editor, the **Result Grid** shows the structure of the `ReturnStatus` table:

Field	Type	Null	Key	Default	Extra
Return_Id	int	NO	PRI		INDEX
Return_cust	int	YES	MUL		INDEX
Return_book_name	varchar(255)	YES			INDEX
Return_date	date	YES			
isbn_book2	int	YES	MUL		INDEX

The **Output** pane shows the execution log of several SQL statements, including the creation and description of the `ReturnStatus` table.

Display all the tables and Write the queries for the following :

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

```

SELECT Book_title, Category, Rental_Price
FROM Books
WHERE Status = 'yes';

```

The screenshot shows the same SQL IDE with the query executed. The **Result Grid** displays the following data:

Book_title	Category	Rental_Price
Kurukkan	Novel	50.00
Randam Oozham	Epic	55.00
Mannu	Drama	60.00

The **Output** pane shows the execution log for the query, indicating that 3 rows were returned.

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

```
SELECT Emp_name, Salary
FROM Employee
ORDER BY Salary DESC;
```

The screenshot shows a SQL IDE interface with a Navigator on the left, a central query editor, and a bottom output pane. The Navigator shows a tree of schemas including 'library', 'product', 'sales', 'school', 'sys', 'testdb', and 'testdb32'. The central query editor contains the following SQL code:

```
122
123
124 -- Retrieve the book title category and rental price of all available books
125
126 • SELECT Book_title, Category, Rental_Price
127 FROM Books
128 WHERE Status = 'yes';
129
130 -- List the employee names and their respective salaries in descending order of salary.
131 • SELECT Emp_name, Salary
132 FROM Employee
133 ORDER BY Salary DESC;
134
```

The bottom pane shows the 'Result Grid' for the second query, displaying a table with two columns: 'Emp_name' and 'Salary'. The data is as follows:

Emp_name	Salary
Rudra Kumar	63000.00
Nirmal Mani	62000.00
Sijumon S	61000.00
Judit Thomas	60000.00
Jobin Jose	58000.00
Sajini Pillai	44000.00
Athira PS	43000.00
Soya Joseph	42000.00
Betty Sbin	41000.00
Sbin Xavier	40000.00

The bottom pane also shows the 'Output' section with a table of action messages:

#	Time	Action	Message	D
77	11:43:12	desc ReturnStatus	5 row(s) returned	0
78	11:46:13	INSERT INTO Branch (Branch_no, Manager_Id, Branch_address, Contact_no) VALUES (...	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0
79	11:48:46	INSERT INTO Employee (Emp_Id, Emp_name, Position, Salary, Branch_no) VALUES (101...	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0
80	11:51:49	INSERT INTO Books (ISBN, Book_title, Category, Rental_Price, Status, Author, Publisher)...	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0
81	11:53:55	INSERT INTO Customer (Customer_Id, Customer_name, Customer_address, Reg_date) V...	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0
82	11:54:52	INSERT INTO IssueStatus (Issue_Id, Issued_cust, Issued_book_name, Issue_date, Ichn...	Error Code: 1452 Cannot add or update a child row: a foreign key constraint fails ('library'...	0

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

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

SCHEMAS

Filter objects

- d32
- employe
- library
 - Tables
 - Views
 - Stored Procedure
 - Functions
- product
- sales
 - Tables
 - Views
 - Stored Procedure
 - Functions
- school
- sys
- testdb
- testdb32

Administration

Information

No object selected

Limit to 1000 rows

```

129
130 -- List the employee names and their respective salaries in descending order of salary.
131 • SELECT Emp_name, Salary
132 FROM Employee
133 ORDER BY Salary DESC;
134
135 -- Retrieve the book titles and the corresponding customers who have issued those books.
136 • SELECT Books.Book_title, Customer.Customer_name
137 FROM Books
138 JOIN IssueStatus ON Books.ISBN = IssueStatus.Isbn_book
139 JOIN Customer ON IssueStatus.Issued_cust = Customer.Customer_Id;
140
141

```

Result Grid

Book_title	Customer_name
Kurukkan	Subin K Joseph
Kurukkan	Tesimol Joseph
Randam Oozham	Nidhin Mathew
Randam Oozham	Nimiha Ratheesh
Mannu	Christy Joy

Output

#	Time	Action	Message	Duration / Fetch
84	11:59:02	INSERT INTO ReturnStatus (Return_Id, Return_cust, Return_book_name, Return_date, ...)	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.000 sec
85	12:00:21	-Retrieve the book title, category, and rental price of all available books. SELECT Book_t...	Error Code: 1064. You have an error in your SQL syntax; check the manual that correspon...	0.000 sec
86	12:00:34	SELECT Book_title, Category, Rental_Price FROM Books WHERE Status = 'yes' LIMIT 0...	3 row(s) returned	0.000 sec / 0.000 sec
87	12:02:52	-Retrieve the book title, category, and rental price of all available books. SELECT Book_t...	Error Code: 1064. You have an error in your SQL syntax; check the manual that correspon...	0.000 sec
88	12:03:44	SELECT Emp_name, Salary FROM Employee ORDER BY Salary DESC LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec
89	12:05:06	SELECT Books.Book_title, Customer.Customer_name FROM Books JOIN IssueStatus ON ...	5 row(s) returned	0.000 sec / 0.000 sec

Object Info

Session

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

SELECT Category, COUNT(*) AS Total_Books
FROM Books
GROUP BY Category;

CHEMAS

Filter objects

- d32
- employe
- library
 - Tables
 - Views
 - Stored Procedure
 - Functions
- product
- sales
 - Tables
 - Views
 - Stored Procedure
 - Functions
- school
- sys
- testdb
- testdb32

Administration

Information

No object selected

Limit to 1000 rows

```

135 -- Retrieve the book titles and the corresponding customers who have issued those books.
136 • SELECT Books.Book_title, Customer.Customer_name
137 FROM Books
138 JOIN IssueStatus ON Books.ISBN = IssueStatus.Isbn_book
139 JOIN Customer ON IssueStatus.Issued_cust = Customer.Customer_Id;
140
141 -- Display the total count of books in each category.
142
143 • SELECT Category, COUNT(*) AS Total_Books
144 FROM Books
145 GROUP BY Category;
146
147

```

Result Grid

Category	Total_Books
Novel	1
Epic	1
Poetry	1
Drama	1
Fiction	1

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

```
SELECT Emp_name, Position
FROM Employee
WHERE Salary > 50000;
```

The screenshot shows a database management tool interface. On the left, a tree view displays the database structure, including tables, views, stored procedures, and functions. The main area displays a SQL query: `SELECT Emp_name, Position FROM Employee WHERE Salary > 50000;`. Below the query, a 'Result Grid' shows the results of the query, which are five rows of employee names and positions: Judit Thomas (Manager), Jobin Jose (Manager), Nirmal Mani (Manager), Sijumon S (Manager), and Rudra Kumar (Manager). The interface also includes a 'Filter Rows' section and an 'Export' button.

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

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

The screenshot shows a database management tool interface. On the left, a tree view displays the database structure, including tables, views, stored procedures, and functions. The main area displays a SQL query: `SELECT Customer_name FROM Customer WHERE Reg_date < '2022-01-01' AND Customer_Id NOT IN (SELECT Issued_cust FROM IssueStatus);`. Below the query, a 'Result Grid' shows the results of the query, which are five rows of customer names: Customer 1, Customer 2, Customer 3, Customer 4, and Customer 5. The interface also includes a 'Filter Rows' section and an 'Export' button.

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

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

The screenshot shows a database management tool interface. On the left, a 'SCHEMAS' pane lists various databases including 'd32', 'employe', 'library', 'product', 'sales', 'school', 'sys', 'testdb', and 'testdb32'. The 'library' database is selected, showing its internal structure. The main area displays a SQL query in a text editor, with line numbers 154 to 166. The query is:
154 FROM Customer
155 WHERE Reg_date < '2022-01-01'
156 AND Customer_Id NOT IN (SELECT Issued_cust FROM IssueStatus);
157
158
159 -- Display the branch numbers and the total count of employees in each branch.
160 • SELECT Branch_no, COUNT(*) AS Total_Employees
161 FROM Employee
162 GROUP BY Branch_no;
163
164
165
166
Below the query editor, a 'Result Grid' shows the output of the query. It has two columns: 'Branch_no' and 'Total_Employees'. The data is as follows:

Branch_no	Total_Employees
1	2
2	2
3	2
4	2
5	2

 The bottom status bar indicates 'Result 16'.

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

```
SELECT Customer.Customer_name
FROM Customer
JOIN IssueStatus ON Customer.Customer_Id = IssueStatus.Issued_cust
WHERE Issue_date BETWEEN '2023-06-01' AND '2023-06-30';
```

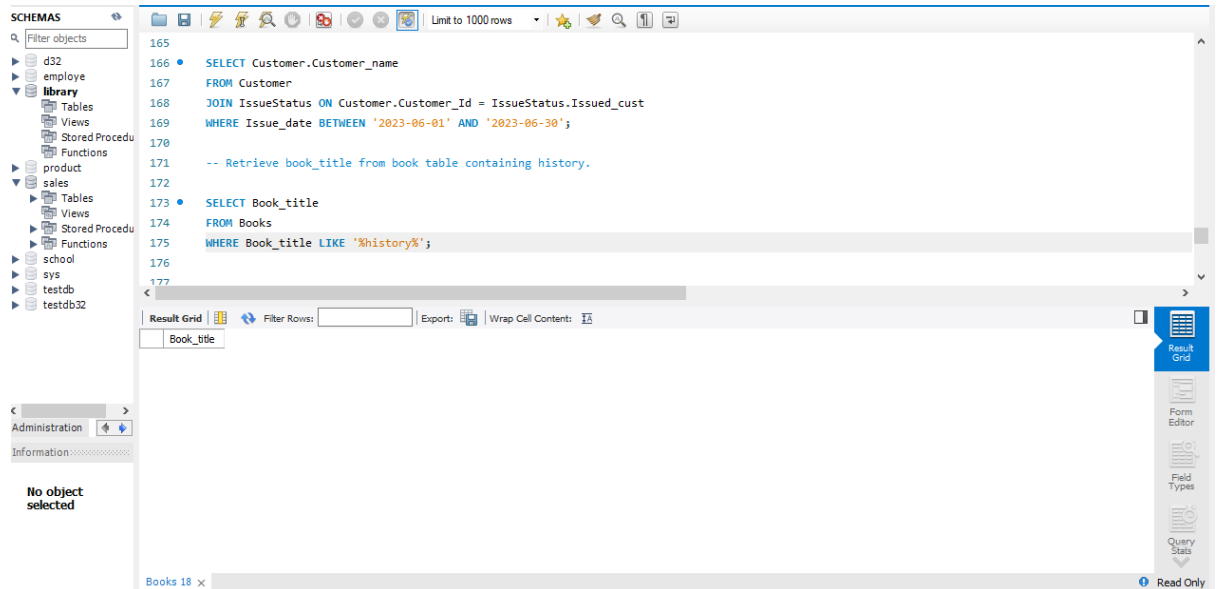
The screenshot shows the same database management tool interface. The 'SCHEMAS' pane is on the left. The main area displays a SQL query in a text editor, with line numbers 164 to 176. The query is:
164 -- Display the names of customers who have issued books in the month of June 2023.
165
166 • SELECT Customer.Customer_name
167 FROM Customer
168 JOIN IssueStatus ON Customer.Customer_Id = IssueStatus.Issued_cust
169 WHERE Issue_date BETWEEN '2023-06-01' AND '2023-06-30';
170
171
172
173
174
175
176
Below the query editor, a 'Result Grid' shows the output of the query. It has one column: 'Customer_name'. The data is as follows:

Customer_name
Subin K Joseph
Nidhin Mathew
Christy Joy
Tesimol Joseph
Nimitha Ratheesh

 The bottom status bar indicates 'Result 17'.

9. Retrieve book_title from book table containing history.

```
SELECT Book_title
FROM Books
WHERE Book_title LIKE '%history%';
```



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

```
SELECT Branch_no, COUNT(*) AS Total_Employees
FROM Employee
GROUP BY Branch_no
HAVING COUNT(*) > 5;
```

The screenshot shows the SQL Developer interface. The left pane displays the 'SCHEMAS' tree with a filter on 'library'. The main editor contains the following SQL query:

```

174 FROM Books
175 WHERE Book_title LIKE '%history%';
176
177 -- Retrieve the branch numbers along with the count of employees for branches having more than 5 employees.
178
179 • SELECT Branch_no, COUNT(*) AS Total_Employees
180 FROM Employee
181 GROUP BY Branch_no
182 HAVING COUNT(*) > 5;
183
184
185
186

```

Below the query editor, the 'Result Grid' shows the following columns: Branch_no, Total_Employees. The 'Output' pane shows the following message:

```

91 12:07:02 SELECT Emp_name, Position FROM Employee WHERE Salary > 50000 LIMIT 0. 1000
5 row(s) returned

```

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

```

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

```

The screenshot shows the SQL Developer interface. The left pane displays the 'SCHEMAS' tree with a filter on 'library'. The main editor contains the following SQL query:

```

181 GROUP BY Branch_no
182 HAVING COUNT(*) > 5;
183
184 -- Retrieve the names of employees who manage branches and their respective branch addresses.
185
186 • SELECT e.Emp_name, b.Branch_address
187 FROM Employee e
188 JOIN Branch b ON e.Emp_Id = b.Manager_Id;
189
190
191
192
193

```

Below the query editor, the 'Result Grid' shows the following columns: Emp_name, Branch_address. The 'Output' pane shows the following message:

```

91 12:07:02 SELECT Emp_name, Position FROM Employee WHERE Salary > 50000 LIMIT 0. 1000
5 row(s) returned

```

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

```

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;

```

The screenshot shows a database management tool interface. On the left, a 'SCHEMAS' pane lists various database objects including 'd32', 'employee', 'library', 'product', 'sales', 'school', 'sys', 'testdb', and 'testdb32'. The main area displays two SQL queries. The first query (lines 186-189) selects employee names and branch addresses. The second query (lines 192-196) is the one shown in the text block, selecting distinct customer names who have issued books with a rental price greater than Rs. 25. Below the queries, a 'Result Grid' shows the output of the second query, listing customer names: Subin K Joseph, Tesimol Joseph, Nidhin Mathew, Nimitha Ratheesh, and Christy Joy. At the bottom, an 'Output' pane shows a log of database actions, including the execution of the query shown in the text block, which returned 5 rows.

```

186 • SELECT e.Emp_name, b.Branch_address
187 FROM Employee e
188 JOIN Branch b ON e.Emp_Id = b.Manager_Id;
189
190 -- Display the names of customers who have issued books with a rental price higher than Rs. 25.
191
192 • SELECT DISTINCT Customer.Customer_name
193 FROM Customer
194 JOIN IssueStatus ON Customer.Customer_Id = IssueStatus.Issued_cust
195 JOIN Books ON IssueStatus.Isbn_book = Books.ISBN
196 WHERE Books.Rental_Price > 25;
197

```

Customer_name
Subin K Joseph
Tesimol Joseph
Nidhin Mathew
Nimitha Ratheesh
Christy Joy

#	Time	Action	Message	Duration / Fetch
93	12:09:58	SELECT Branch_no, COUNT(*) AS Total_Employees FROM Employee GROUP BY Branch_no	5 row(s) returned	0.016 sec / 0.000 sec
94	12:10:54	SELECT Customer.Customer_name FROM Customer JOIN IssueStatus ON Customer.Customer_Id = IssueStatus.Issued_cust	5 row(s) returned	0.000 sec / 0.000 sec
95	12:11:34	SELECT Book_title FROM Books WHERE Book_title LIKE "%history%"; LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec
96	12:12:11	SELECT Branch_no, COUNT(*) AS Total_Employees FROM Employee GROUP BY Branch_no	5 row(s) returned	0.000 sec / 0.000 sec