

1.Branch table

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

- Project:** assignment8_subquery_view
- Tab:** assignment8_subquery_view
- Code:**

```
4
5
6 -- 1. Branch Branch_no - Set as PRIMARY KEY Manager_Id Branch_address Contact_no
7 • Create table Branch(Branch_no Int PRIMARY KEY,
8   Manager_Id Int,
9   Branch_address varchar(30),
10  Contact_no bigint);
11 • INSERT INTO Branch (Branch_no, Manager_Id, Branch_address, Contact_no)
12    VALUES
13    (1, 101, 'Downtown Library', 9876543210),
14    (2, 102, 'Uptown Library', 8765432109),
15    (3, 103, 'Central City Library', 7654321098),
16    (4, 104, 'Westside Library', 6543210987),
17    (5, 105, 'Eastside Library', 5432109876);
18 • select * from Branch;
```
- Result Grid:** Shows the data inserted into the Branch table.

Branch_no	Manager_Id	Branch_address	Contact_no
1	101	Downtown Library	9876543210
2	102	Uptown Library	8765432109
3	103	Central City Library	7654321098
4	104	Westside Library	6543210987
5	105	Eastside Library	5432109876

2.Employee table

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

- Project:** assignment8_subquery_view
- Tab:** assignment8_subquery_view
- Code:**

```
28 VALUES
29  (101, 'John Doe', 'Manager', 75000, 1),
30  (102, 'Jane Smith', 'Manager', 50000, 2),
31  (103, 'Robert Brown', 'Manager', 60000, 3),
32  (104, 'Emily Davis', 'Technician', 55000, 4),
33  (105, 'Michael Wilson', 'Clerk', 45000, 5),
34  (106, 'Varun K', 'Clerk', 70000, 3),
35  (107, 'Simi K', 'Assistant', 70000, 5),
36  (108, 'Kiran K', 'Technician', 50000, 3),
37  (109, 'Baiji K', 'Technician', 60000, 3),
38  (110, 'Adam K', 'Technician', 70000, 3),
39  (111, 'Nooh K', 'Manager', 70000, 3);
40 • select * from Employee;
```
- Result Grid:** Shows the data inserted into the Employee table.

Emp_Id	Emp_name	Position	Salary	Branch_no
101	John Doe	Manager	75000	1
102	Jane Smith	Manager	50000	2
103	Robert Brown	Manager	60000	3
104	Emily Davis	Technician	55000	4
105	Michael Wilson	Clerk	45000	5
106	Varun K	Clerk	70000	3
107	Simi K	Assistant	70000	5
108	Kiran K	Technician	50000	3
109	Baiji K	Technician	60000	3
110	Adam K	Technician	70000	3
111	Nooh K	Manager	70000	3

3.Books table

```

Category varchar(30),
Rental_Price int,
Status varchar(10),
Author VARCHAR(50),
Publisher VARCHAR(50));
50 • INSERT INTO Books (ISBN, Book_title, Category, Rental_Price, Status, Author, Publisher)
VALUES
(1110, 'The Great Gatsby', 'Fiction', 300, 'yes', 'F. Scott Fitzgerald', 'Scribner'),
(1111, 'To Kill a Mockingbird', 'Fiction', 250, 'no', 'Harper Lee', 'J.B. Lippincott & Co.'),
(1112, 'Fahrenheit 451', 'Dystopian', 350, 'yes', 'Ray Bradbury', 'Simon & Schuster'),
(1113, '1984', 'Dystopian', 280, 'no', 'George Orwell', 'Houghton Mifflin Harcourt'),
(1114, 'Pride and Prejudice', 'Romance', 200, 'yes', 'Jane Austen', 'Modern Library'),
(1115, 'History forever', 'Romance', 200, 'yes', 'Jane Austen', 'Modern Library');
58 • select * from Books;
59 -- 4. Customer Customer_Id - Set as PRIMARY KEY Customer_name Customer_address Reg_date

```

Result Grid

ISBN	Book_title	Category	Rental_Price	Status	Author	Publisher
1110	The Great Gatsby	Fiction	300	yes	F. Scott Fitzgerald	Scribner
1111	To Kill a Mockingbird	Fiction	250	no	Harper Lee	J.B. Lippincott & Co.
1112	Fahrenheit 451	Dystopian	350	yes	Ray Bradbury	Simon & Schuster
1113	1984	Dystopian	280	no	George Orwell	Houghton Mifflin Harcourt
1114	Pride and Prejudice	Romance	200	yes	Jane Austen	Modern Library
1115	History forever	Romance	200	yes	Jane Austen	Modern Library
HULL	HULL	HULL	HULL	HULL	HULL	HULL

4.Customer table

```

Reg_date date);
65 • INSERT INTO Customer (Customer_Id, Customer_name, Customer_address, Reg_date)
VALUES
(1, 'John Doe', '123 Elm St, Springfield', '2024-01-15'),
(2, 'Jane Smith', '456 Oak St, Springfield', '2024-02-22'),
(3, 'Emily Johnson', '789 Pine St, Springfield', '2024-03-10'),
(4, 'Michael Brown', '101 Maple St, Springfield', '2024-04-05'),
(5, 'Sarah Davis', '202 Birch St, Springfield', '2024-05-18'),
(6, 'Rincy Davis', '202 Birch St, Springfield', '2021-05-18'),
(7, 'Natalie Portman', '501 Oak St, Springfield', '2021-11-15'),
(8, 'Chris Hemsworth', '601 Pine St, Springfield', '2021-12-10');
75 • select * from Customer;
76
77 -- 5. IssueStatus Issue_Id - Set as PRIMARY KEY Issued_cust - Set as FOREIGN KEY and it refer customer_id in CUSTOMER table Issued_by
78 • create table IssueStatus(Issue_Id int PRIMARY KEY,

```

Result Grid

Customer_Id	Customer_name	Customer_address	Reg_date
1	John Doe	123 Elm St, Springfield	2024-01-15
2	Jane Smith	456 Oak St, Springfield	2024-02-22
3	Emily Johnson	789 Pine St, Springfield	2024-03-10
4	Michael Brown	101 Maple St, Springfield	2024-04-05
5	Sarah Davis	202 Birch St, Springfield	2024-05-18
6	Rincy Davis	202 Birch St, Springfield	2021-05-18
7	Natalie Portman	501 Oak St, Springfield	2021-11-15
8	Chris Hemsworth	601 Pine St, Springfield	2021-12-10
HULL	HULL	HULL	HULL

5.Issue Status table

```

81  Issued_book_name varchar(30),
82  Issue_date date,
83  Isbn_book bigint,
84  FOREIGN KEY (Isbn_book) REFERENCES BOOKS(ISBN));
85 • INSERT INTO IssueStatus (Issue_Id, Issued_cust, Issued_book_name, Issue_date, Isbn_book)
86  VALUES
87      (1, 1, 'The Great Gatsby', '2024-08-01', 1110),
88      (2, 2, 'To Kill a Mockingbird', '2024-08-05', 1111),
89      (3, 3, '1984', '2024-08-10', 1113),
90      (4, 4, 'Fahrenheit 451', '2024-08-15', 1112),
91      (5, 5, 'Pride and Prejudice', '2024-08-20', 1114),
92      (6, 6, '202 Birch St, Springfield', '2023-06-04', 1115);
93 • select * from IssueStatus;
94  -- 6. ReturnStatus Return_Id – Set as PRIMARY KEY Return_cust Return_book_name Return_date Isbn_book2 – Set as FOREIGN KEY and it
95 • create table ReturnStatus(Return_Id int PRIMARY KEY,
100% 31:93 |
```

Result Grid Filter Rows: Search Edit: Export/Import: Result Grid Form Editor

Issue_Id	Issued_cust	Issued_book_name	Issue_date	Isbn_book
1	1	The Great Gatsby	2024-08-01	1110
2	2	To Kill a Mockingbird	2024-08-05	1111
3	3	1984	2024-08-10	1113
4	4	Fahrenheit 451	2024-08-15	1112
5	5	Pride and Prejudice	2024-08-20	1114
6	6	202 Birch St, Springfield	2023-06-04	1115
HULL	HULL	HULL	HULL	HULL

6.Return Status table

```

95 • create table ReturnStatus(Return_Id int PRIMARY KEY,
96  Return_cust varchar(20),
97  Return_book_name varchar(30),
98  Return_date date,
99  Isbn_book2 bigint,
100  FOREIGN KEY (Isbn_book2) REFERENCES BOOKS(ISBN));
101 • INSERT INTO ReturnStatus (Return_Id, Return_cust, Return_book_name, Return_date, Isbn_book2)
102  VALUES
103      (1, 1, 'The Great Gatsby', '2024-08-15', 1110),
104      (2, 2, 'To Kill a Mockingbird', '2024-08-20', 1111),
105      (3, 3, '1984', '2024-08-25', 1113),
106      (4, 4, 'Fahrenheit 451', '2024-09-01', 1112),
107      (5, 5, 'Pride and Prejudice', '2024-09-10', 1114);
108 • select * from ReturnStatus;
109  -- 1. Retrieve the book title, category, and rental price of all available books.
100% 32:108 |
```

Result Grid Filter Rows: Search Edit: Export/Import: Result Grid Form Editor

Return_Id	Return_cust	Return_book_name	Return_date	Isbn_book2
1	1	The Great Gatsby	2024-08-15	1110
2	2	To Kill a Mockingbird	2024-08-20	1111
3	3	1984	2024-08-25	1113
4	4	Fahrenheit 451	2024-09-01	1112
5	5	Pride and Prejudice	2024-09-10	1114
HULL	HULL	HULL	HULL	HULL

1. Q1_Result

Limit to 1000 rows

```

98 | Return_date date,
99 | Isbn_book2 bigint,
100| FOREIGN KEY (Isbn_book2) REFERENCES BOOKS(ISBN));
101 • INSERT INTO ReturnStatus (Return_Id, Return_cust, Return_book_name, Return_date, Isbn_book2)
102 VALUES
103     (1, 1, 'The Great Gatsby', '2024-08-15', 1110),
104     (2, 2, 'To Kill a Mockingbird', '2024-08-20', 1111),
105     (3, 3, '1984', '2024-08-25', 1113),
106     (4, 4, 'Fahrenheit 451', '2024-09-01', 1112),
107     (5, 5, 'Pride and Prejudice', '2024-09-10', 1114);
108 • select * from ReturnStatus;
109 -- 1. Retrieve the book title, category, and rental price of all available books.
110 • select Book_title,Category,Rental_Price from Books;
111
112 -- 2. List the employee names and their respective salaries in descending order of salary.

```

Result Grid Filter Rows: Search Export:

Book_title	Category	Rental_Price
The Great Gatsby	Fiction	300
To Kill a Mockingbird	Fiction	250
Fahrenheit 451	Dystopian	350
1984	Dystopian	280
Pride and Prejudice	Romance	200
History forever	Romance	200

2. Q2_Result

Limit to 1000 rows

```

105      (3, 3, '1984', '2024-08-25', 1113),
106      (4, 4, 'Fahrenheit 451', '2024-09-01', 1112),
107      (5, 5, 'Pride and Prejudice', '2024-09-10', 1114);
108 •   select * from ReturnStatus;
109
110      -- 1. Retrieve the book title, category, and rental price of all available books.
111 •   select Book_title,Category,Rental_Price from Books;
112
113      -- 2. List the employee names and their respective salaries in descending order of salary.
114 •   select Emp_name,Salary from Employee order by salary desc;
115
116      -- 3. Retrieve the book titles and the corresponding customers who have issued those books.
117 •   Select Issued_book_name,Issued_cust from IssueStatus;

```

Result Grid

Emp_name	Salary
John Doe	75000
Varun K	70000
Simi K	70000
Adam K	70000
Nooh K	70000
Robert Brown	60000
Emily Davis	55000
Jane Smith	50000
Michael Wilson	45000
Baiji K	6000
Kiran K	5000

Employee 30

Read

3. Q3_Result

Limit to 1000 rows

```

107      (5, 5, 'Pride and Prejudice', '2024-09-10', 1114);
108 •   select * from ReturnStatus;
109
110      -- 1. Retrieve the book title, category, and rental price of all available books.
111 •   select Book_title,Category,Rental_Price from Books;
112
113      -- 2. List the employee names and their respective salaries in descending order of salary.
114 •   select Emp_name,Salary from Employee order by salary desc;
115
116      -- 3. Retrieve the book titles and the corresponding customers who have issued those books.
117 •   Select Issued_book_name,Issued_cust from IssueStatus;
118
119      -- 4. Display the total count of books in each category.

```

Result Grid

Issued_book_name	Issued_cust
The Great Gatsby	1
To Kill a Mockingbird	2
1984	3
Fahrenheit 451	4
Pride and Prejudice	5
202 Birch St, Springfield	6

4. Q4_Result

Limit to 1000 rows

```

111 •     select Book_title,Category,Rental_Price from Books;
112
113     -- 2. List the employee names and their respective salaries in descending order of salary.
114 •     select Emp_name,Salary from Employee order by salary desc;
115
116     -- 3. Retrieve the book titles and the corresponding customers who have issued those books.
117 •     Select Issued_book_name,Issued_cust from IssueStatus;
118
119     -- 4. Display the total count of books in each category.
120 •         select category,count(*) as 'Count of books' from books group by category;
121
122     -- 5. Retrieve the employee names and their positions for the employees whose salaries are above Rs.50,000.
123 •         select Emp_name,Position from Employee where Salary > 50000;

```

Result Grid

category	Count of boo...
Fiction	2
Dystopian	2
Romance	2

5. Q5_Result

Limit to 1000 rows

```

-- 4. Display the total count of books in each category.
select category,count(*) as 'Count of books' from books group by category;

-- 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;

-- 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);

-- 7. Display the branch numbers and the total count of employees in each branch.
select Branch_no, count(*) as Employee_Count from Employee group by(Branch_no);

```

Result Grid

Emp_name	Position
John Doe	Manager
Robert Brown	Manager
Emily Davis	Technician
Varun K	Clerk
Simi K	Assistant
Adam K	Technician
Nooh K	Manager

6. Q6_Result

119 -- 4. Display the total count of books in each category.
120 • select category,count(*) as 'Count of books' from books group by category;
121
122 -- 5. Retrieve the employee names and their positions for the employees whose salaries are above Rs.50,000.
123 • select Emp_name,Position from Employee where Salary > 50000;
124
125 -- 6. List the customer names who registered before 2022-01-01 and have not issued any books yet.
126 • select Customer_name from Customer where Reg_date < '2022-01-01' and Customer_Id NOT IN (select Issued_cust from IssueStatus);
127
128 -- 7. Display the branch numbers and the total count of employees in each branch.
129 • select Branch_no, count(*) as Employee_Count from Employee group by(Branch_no);
130
131 -- 8. Display the names of customers who have issued books in the month of June 2023.

Customer 34 Read Only

7. Q7_Result

119 -- 4. Display the total count of books in each category.
120 • select category,count(*) as 'Count of books' from books group by category;
121
122 -- 5. Retrieve the employee names and their positions for the employees whose salaries are above Rs.50,000.
123 • select Emp_name,Position from Employee where Salary > 50000;
124
125 -- 6. List the customer names who registered before 2022-01-01 and have not issued any books yet.
126 • select Customer_name from Customer where Reg_date < '2022-01-01' and Customer_Id NOT IN (select Issued_cust from IssueStatus);
127
128 -- 7. Display the branch numbers and the total count of employees in each branch.
129 • select Branch_no, count(*) as Employee_Count from Employee group by(Branch_no);
130
131 -- 8. Display the names of customers who have issued books in the month of June 2023.

Branch_no Employee_Count

Branch_no	Employee_Count
1	1
2	1
3	6
4	1
5	2

8. Q8_Result

```

assignment8_subquery_view Project* storedprocedure assignment9_storedprocedures before trigger aftertrigger assignment10_triggers >
122
123 • 6. List the customer names who registered before 2022-01-01 and have not issued any books yet.
124
125 -- 6. List the customer names who registered before 2022-01-01 and have not issued any books yet.
126 • select Customer_name from Customer where Reg_date < '2022-01-01' and Customer_Id NOT IN (select Issued_cust from IssueStatus);
127
128 -- 7. Display the branch numbers and the total count of employees in each branch.
129 • select Branch_no, count(*) as Employee_Count from Employee group by(Branch_no);
130
131 -- 8. Display the names of customers who have issued books in the month of June 2023.
132 • select C.Customer_name from Customer C JOIN IssueStatus I ON C.Customer_Id = I.Issued_cust where I.Issue_date between '2023-06-01'
133
134 -- 9. Retrieve book_title from book table containing history.
135 • select Book_title from Books where Book_title LIKE '%history%';
136
100% 153:132
Result Grid Filter Rows: Search Export: 
Customer_name
Rincy Davis

```

9. Q9_Result

```

assignment8_subquery_view Project* storedprocedure assignment9_storedprocedures before trigger aftertrigger assignment10_triggers >
124
125 -- 6. List the customer names who registered before 2022-01-01 and have not issued any books yet.
126 • select Customer_name from Customer where Reg_date < '2022-01-01' and Customer_Id NOT IN (select Issued_cust from IssueStatus);
127
128 -- 7. Display the branch numbers and the total count of employees in each branch.
129 • select Branch_no, count(*) as Employee_Count from Employee group by(Branch_no);
130
131 -- 8. Display the names of customers who have issued books in the month of June 2023.
132 • select C.Customer_name from Customer C JOIN IssueStatus I ON C.Customer_Id = I.Issued_cust where I.Issue_date between '2023-06-01'
133
134 -- 9. Retrieve book_title from book table containing history.
135 • select Book_title from Books where Book_title LIKE '%history%';
136
137 -- 10. Retrieve the branch numbers along with the count of employees for branches having more than 5 employees
100% 68:135
Result Grid Filter Rows: Search Export: 
Book_title
History forever

```

10. Q10_Result

```

127
128 -- 7. Display the branch numbers and the total count of employees in each branch.
129 • select Branch_no, count(*) as Employee_Count from Employee group by(Branch_no);
130
131 -- 8. Display the names of customers who have issued books in the month of June 2023.
132 • select C.Customer_name from Customer C JOIN IssueStatus I ON C.Customer_Id = I.Issued_cust where I.Issue_date between '2023-06-01'
133
134 -- 9. Retrieve book_title from book table containing history.
135 • select Book_title from Books where Book_title LIKE '%history%';
136
137 -- 10. Retrieve the branch numbers along with the count of employees for branches having more than 5 employees
138 • SELECT Branch_no, COUNT(Emp_Id) AS Employee_Count FROM Employee GROUP BY Branch_no HAVING COUNT(Emp_Id) > 5;
139
140 -- 11. Retrieve the names of employees who manage branches and their respective branch addresses.
100% 113:138 |
```

Result Grid Filter Rows: Search Export:

Branch_no	Employee_Count
3	6

11. Q11_Result

```

131
132 • 
133
134 -- 9. Retrieve book_title from book table containing history.
135 • select Book_title from Books where Book_title LIKE '%history%';
136
137 -- 10. Retrieve the branch numbers along with the count of employees for branches having more than 5 employees
138 • SELECT Branch_no, COUNT(Emp_Id) AS Employee_Count FROM Employee GROUP BY Branch_no HAVING COUNT(Emp_Id) > 5;
139
140 -- 11. Retrieve the names of employees who manage branches and their respective branch addresses.
141 • SELECT E.Emp_name, B.Branch_address FROM Employee E JOIN Branch B ON E.Emp_Id = B.Manager_Id AND E.Branch_no = B.Branch_no;
142
143 -- 12. Display the names of customers who have issued books with a rental price higher than Rs. 25.
144 • select C.Customer_name from Customer C JOIN IssueStatus I ON C.Customer_Id = I.Issued_cust JOIN Books B ON B.Isbn = I.Issn_book wh
145
100% 128:141 |
```

Result Grid Filter Rows: Search Export:

Emp_name	Branch_address
John Doe	Downtown Library
Jane Smith	Uptown Library
Robert Brown	Central City Library
Emily Davis	Westside Library
Michael Wilson	Eastside Library

12. Q12_Result

```
133
134      -- 9. Retrieve book_title from book table containing history.
135 •     select Book_title from Books where Book_title LIKE '%history%';
136
137      -- 10. Retrieve the branch numbers along with the count of employees for branches having more than 5 employees
138 •     SELECT Branch_no, COUNT(Emp_Id) AS Employee_Count FROM Employee GROUP BY Branch_no HAVING COUNT(Emp_Id) > 5;
139
140      -- 11. Retrieve the names of employees who manage branches and their respective branch addresses.
141 •     SELECT E.Emp_name, B.Branch_address FROM Employee E JOIN Branch B ON E.Emp_Id = B.Manager_Id AND E.Branch_no = B.Branch_no;
142
143      -- 12. Display the names of customers who have issued books with a rental price higher than Rs. 25.
144 •     select C.Customer_name from Customer C JOIN IssueStatus I ON C.Customer_Id = I.Issued_cust JOIN Books B ON B.Isbn = I.Isbn_book w
145
```

100% 159:144

Result Grid Filter Rows: Search Export:

Customer_name
John Doe
Jane Smith
Michael Brown
Emily Johnson
Sarah Davis
Rincy Davis

Result Grid Form Editor Field Types