

Navigator

SCHEMAS

Filter objects

- bankdatabase
- colleges
- company
- db
- education
- employee
- employee1
- global_store_db
- india
- manager_details
- places
- product
- sales
- school
- school2
- store
- studentjoin
- sys
- testdatabaseofmysql
- triggers
- workemployee

Administration Schemas

Information

No object selected

Object Info Session

SQL File 1 task9 triggers SQL File 4* sql comprehension

Limit to 1000 rows

```

1 -- Create the library database
2 CREATE DATABASE library;
3 -- Use the library database
4 USE library;
5 -- Create Branch table
6 CREATE TABLE Branch (
7     Branch_no INT PRIMARY KEY,
8     Manager_Id INT,
9     Branch_address VARCHAR(255),
10    Contact_no VARCHAR(15)
11 );
12 -- Insert values into Branch table
13 INSERT INTO Branch (Branch_no, Manager_Id, Branch_address, Contact_no) VALUES
14 (1, 101, '123 Main St, Cityville', '123-456-7890'),
15 (2, 102, '456 Elm St, Townsville', '234-567-8901'),
16 (3, 103, '789 Oak St, Villageville', '345-678-9012');
17 select *from Branch;

```

Result Grid

Branch_no	Manager_Id	Branch_address	Contact_no
1	101	123 Main St, Cityville	123-456-7890
2	102	456 Elm St, Townsville	234-567-8901
3	103	789 Oak St, Villageville	345-678-9012

Branch 1 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
4	18:37:24	INSERT INTO Branch (Branch_no, Manager_Id, Branch_address, Contact_no) VALUES	(1, 101, '123 Main St, Cityville', '123-456-7890');	0.109 sec
5	18:37:49	select *from Branch LIMIT 0, 1000	(2, 102, '456 Elm St, Townsville', '234-567-8901'), (3, 103, '789 Oak St, Villageville', '345-678-9012');	0.015 sec / 0.000 sec

Navigator

SCHEMAS

Filter objects

- bankdatabase
- colleges
- company
- db
- education
- employee
- employee1
- global_store_db
- india
- manager_details
- places
- product
- sales
- school
- school2
- store
- studentjoin
- sys
- testdatabaseofmysql
- triggers
- workemployee

Administration Schemas

Information

No object selected

Object Info Session

SQL File 1 task9 triggers SQL File 4* sql comprehension

Limit to 1000 rows

```

19 -- Create Employee table
20 CREATE TABLE Employee (
21     Emp_Id INT PRIMARY KEY,
22     Emp_name VARCHAR(100),
23     Position VARCHAR(50),
24     Salary DECIMAL(10, 2),
25     Branch_no INT,
26     FOREIGN KEY (Branch_no) REFERENCES Branch(Branch_no)
27 );
28 -- Insert values into Employee table
29 INSERT INTO Employee (Emp_Id, Emp_name, Position, Salary, Branch_no) VALUES
30 (1, 'Alice Johnson', 'Manager', 60000.00, 1),
31 (2, 'Bob Smith', 'Librarian', 40000.00, 1),
32 (3, 'Charlie Brown', 'Assistant', 30000.00, 2),
33 (4, 'Diana Prince', 'Manager', 65000.00, 2),
34 (5, 'Eve Adams', 'Librarian', 42000.00, 3);
35 select *from Employee;

```

Result Grid

Emp_Id	Emp_name	Position	Salary	Branch_no
1	Alice Johnson	Manager	60000.00	1
2	Bob Smith	Librarian	40000.00	1
3	Charlie Brown	Assistant	30000.00	2
4	Diana Prince	Manager	65000.00	2
5	Eve Adams	Librarian	42000.00	3

Employee 2 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
7	18:38:54	INSERT INTO Employee (Emp_Id, Emp_name, Position, Salary, Branch_no) VALUES	(1, 'Alice Johnson', 'Manager', 60000.00, 1), (2, 'Bob Smith', 'Librarian', 40000.00, 1), (3, 'Charlie Brown', 'Assistant', 30000.00, 2), (4, 'Diana Prince', 'Manager', 65000.00, 2), (5, 'Eve Adams', 'Librarian', 42000.00, 3);	0.078 sec
8	18:39:10	select *from Employee LIMIT 0, 1000		0.000 sec / 0.000 sec

SCHEMAS

Filter objects

- bankdatabase
- colleges
- company
- db
- education
- employee
- employee1
- global_store_db
- india
- manager_details
- places
- product
- sales
- school
- school2
- store
- studentjoin
- sys
- testdatabaseofmysql
- triggers
- workemployee

Administration Schemas

Information

No object selected

```

37 CREATE TABLE Books (
38     ISBN VARCHAR(20) PRIMARY KEY,
39     Book_title VARCHAR(255),
40     Category VARCHAR(100),
41     Rental_Price DECIMAL(10, 2),
42     Status VARCHAR(3) CHECK (Status IN ('yes', 'no')),
43     Author VARCHAR(100),
44     Publisher VARCHAR(100)
45 );
46 -- Insert values into Books table
47 INSERT INTO Books (ISBN, Book_title, Category, Rental_Price, Status, Author, Publisher) VALUES
48 ('978-3-16-148410-0', 'The Great Gatsby', 'Fiction', 5.99, 'yes', 'F. Scott Fitzgerald', 'Scribner'),
49 ('978-0-14-017739-0', 'To Kill a Mockingbird', 'Fiction', 6.99, 'no', 'Harper Lee', 'J.B. Lippincott & Co.'),
50 ('978-0-452-28423-4', '1984', 'Dystopian', 7.99, 'yes', 'George Orwell', 'Secker & Warburg'),
51 ('978-0-7432-7356-5', 'The Da Vinci Code', 'Mystery', 8.99, 'no', 'Dan Brown', 'Doubleday'),
52 ('978-1-4516-7321-8', 'Steve Jobs', 'Biography', 9.99, 'yes', 'Walter Isaacson', 'Simon & Schuster'),
53 ('978-3-16-148410-0', 'The Great Gatsby', 'Fiction', 5.99, 'yes', 'F. Scott Fitzgerald', 'Scribner');
54 select *from Books;

```

Result Grid

ISBN	Book_title	Category	Rental_Price	Status	Author	Publisher
978-0-14-017739-0	To Kill a Mockingbird	Fiction	6.99	no	Harper Lee	J.B. Lippincott & Co.
978-0-452-28423-4	1984	Dystopian	7.99	yes	George Orwell	Secker & Warburg
978-0-7432-7356-5	The Da Vinci Code	Mystery	8.99	no	Dan Brown	Doubleday
978-1-4516-7321-8	Steve Jobs	Biography	9.99	yes	Walter Isaacson	Simon & Schuster
978-3-16-148410-0	The Great Gatsby	Fiction	5.99	yes	F. Scott Fitzgerald	Scribner

Books 3 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
10	18:44:01	INSERT INTO Books (ISBN, Book_title, Category, Rental_Price, Status, Author, Publisher) VALUES (17/8/3...	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.031 sec
11	18:44:15	select *from Books LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec

SQL File 1 task9 triggers SQL File 4* sql compression*

SCHEMAS

Filter objects

- bankdatabase
- colleges
- company
- db
- education
- employee
- employee1
- global_store_db
- india
- manager_details
- places
- product
- sales
- school
- school2
- store
- studentjoin
- sys
- testdatabaseofmysql
- triggers
- workemployee

Administration Schemas

Information

No object selected

```

54 -- Create Customer table
55 CREATE TABLE Customer (
56     Customer_id INT PRIMARY KEY,
57     Customer_name VARCHAR(100),
58     Customer_address VARCHAR(255),
59     Reg_date DATE
60 );
61 -- Insert values into Customer table
62 INSERT INTO Customer (Customer_id, Customer_name, Customer_address, Reg_date) VALUES
63 (1, 'John Doe', '789 Maple St, Suburbia', '2022-01-15'),
64 (2, 'Jane Smith', '456 Pine St, Urbania', '2022-03-20'),
65 (3, 'Tom Hanks', '123 Birch St, Countryville', '2022-05-25');
66 select *from customer;
67 -- Create IssueStatus table
68 CREATE TABLE IssueStatus (
69     Issue_id INT PRIMARY KEY,
70     Issued_cust_id INT,

```

Result Grid

Customer_id	Customer_name	Customer_address	Reg_date
1	John Doe	789 Maple St, Suburbia	2022-01-15
2	Jane Smith	456 Pine St, Urbania	2022-03-20
3	Tom Hanks	123 Birch St, Countryville	2022-05-25

customer 4 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
13	18:45:10	INSERT INTO Customer (Customer_id, Customer_name, Customer_address, Reg_date) VALUES (1, 30971 U...	3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0	0.015 sec
14	18:45:13	select *from customer LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

SQL File 1 task9 triggers SQL File 4* sql comprehension

```

67 -- Create IssueStatus table
68 CREATE TABLE IssueStatus (
69     Issue_Id INT PRIMARY KEY,
70     Issued_cust_id INT,
71     Issued_book_name VARCHAR(255),
72     Issue_date DATE,
73     Isbn_book VARCHAR(20),
74     FOREIGN KEY (Issued_cust_id) REFERENCES Customer(Customer_Id),
75     FOREIGN KEY (Isbn_book) REFERENCES Books(ISBN)
76 );
77 -- Insert values into IssueStatus table
78 INSERT INTO IssueStatus (Issue_Id, Issued_cust_id, Issued_book_name, Issue_date, Isbn_book) VALUES
79 (1, 1, 'To Kill a Mockingbird', '2023-07-01', '978-0-14-017739-8'),
80 (2, 2, 'The Da Vinci Code', '2023-07-02', '978-0-7432-7356-5');
81 select * from IssueStatus;
82 -- Create ReturnStatus table
83 CREATE TABLE ReturnStatus (

```

Result Grid

Issue_Id	Issued_cust_id	Issued_book_name	Issue_date	Isbn_book
1	1	To Kill a Mockingbird	2023-07-01	978-0-14-017739-8
2	2	The Da Vinci Code	2023-07-02	978-0-7432-7356-5

IssueStatus 5 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
16	18:46:09	INSERT INTO IssueStatus (Issue_Id, Issued_cust_id, Issued_book_name, Issue_date, Isbn_book) VALUES	2 row(s) affected Records: 2 Duplicates: 0 Warnings: 0	0.016 sec
17	18:46:11	select * from IssueStatus LIMIT 0, 1000	INSERT INTO IssueStatus (Issue_Id, Issued_cust_id, Issued_book_name, Issue_date, Isbn_book) VALUES (1, 1, 'To Kill a Mockingbird', '2023-07-01', '978-0-14-017739-8'), (2, 2, 'The Da Vinci Code', '2023-07-02', '978-0-7432-7356-5');	0.000 sec / 0.000 sec

SQL File 1 task9 triggers SQL File 4* sql comprehension

```

82 -- Create ReturnStatus table
83 CREATE TABLE ReturnStatus (
84     Return_Id INT PRIMARY KEY,
85     Return_cust INT,
86     Return_book_name VARCHAR(255),
87     Return_date DATE,
88     Isbn_book2 VARCHAR(20),
89     FOREIGN KEY (Return_cust) REFERENCES Customer(Customer_Id),
90     FOREIGN KEY (Isbn_book2) REFERENCES Books(ISBN)
91 );
92 -- Insert values into ReturnStatus table
93 INSERT INTO ReturnStatus (Return_Id, Return_cust, Return_book_name, Return_date, Isbn_book2) VALUES
94 (1, 1, 'To Kill a Mockingbird', '2023-07-10', '978-0-14-017739-8'),
95 (2, 2, 'The Da Vinci Code', '2023-07-12', '978-0-7432-7356-5');
96
97 select * from ReturnStatus;
98 -- 1. Retrieve the book title, category, and rental price of all available books.

```

Result Grid

Return_Id	Return_cust	Return_book_name	Return_date	Isbn_book2
1	1	To Kill a Mockingbird	2023-07-10	978-0-14-017739-8
2	2	The Da Vinci Code	2023-07-12	978-0-7432-7356-5

ReturnStatus 6 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
19	18:46:48	INSERT INTO ReturnStatus (Return_Id, Return_cust, Return_book_name, Return_date, Isbn_book2) VALUES	2 row(s) affected Records: 2 Duplicates: 0 Warnings: 0	0.031 sec
20	18:47:06	select from ReturnStatus LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec

SQL File 1 task9 triggers SQL File 4* sql comprehension

```

98 -- 1. Retrieve the book title, category, and rental price of all available books.
99
100 SELECT Book_title, Category, Rental_Price
101 FROM Books
102 WHERE Status = 'yes';

```

Result Grid

Book_title	Category	Rental_Price
1984	Dystopian	7.99
Steve Jobs	Biography	9.99
The Great Gatsby	Fiction	5.99

SQL File 1 task9 triggers SQL File 4* sql comprehension*

Limit to 1000 rows

```

104 -- 2. List the employee names and their respective salaries in descending order of salary.
105 SELECT Emp_name, Salary
106 FROM Employee
107 ORDER BY Salary DESC;
108

```

Result Grid

Emp_name	Salary
Diana Prince	65000.00
Alice Johnson	60000.00
Eve Adams	42000.00
Bob Smith	40000.00
Charlie Brown	30000.00

Navigation: Schemas

- bankdatabase
- colleges
- company
- db
- education
- employee
- employee1
- global_store_db
- india
- manager_details
- places
- product
- sales
- school
- school2
- store
- studentjoin
- sys

SQL File 1 task9 triggers SQL File 4* sql comprehension*

Limit to 1000 rows

```

109 -- 3. Retrieve the book titles and the corresponding customers who have issued those books.
110 SELECT Books.Book_title, Customer.Customer_name
111 FROM IssueStatus
112 JOIN Books ON IssueStatus.Isbn_book = Books.ISBN
113 JOIN Customer ON IssueStatus.Issued_cust_id = Customer.Customer_Id;

```

Result Grid

Book_title	Customer_name
To Kill a Mockingbird	John Doe
The Da Vinci Code	Jane Smith

Navigation: Schemas

- bankdatabase
- colleges
- company
- db
- education
- employee
- employee1
- global_store_db
- india
- manager_details
- places
- product
- sales
- school
- school2
- store
- studentjoin
- sys

SQL File 1 task9 triggers SQL File 4* sql comprehension*

Limit to 1000 rows

```

116 -- 4. Display the total count of books in each category.
117 SELECT Category, COUNT(*) AS Total_Books
118 FROM Books
119 GROUP BY Category;
120

```

Result Grid

Category	Total_Books
Fiction	2
Dystopian	1
Mystery	1
Biography	1

Navigation: Schemas

- bankdatabase
- colleges
- company
- db
- education
- employee
- employee1
- global_store_db
- india
- manager_details
- places
- product
- sales
- school
- school2
- store
- studentjoin
- sys
- testdatabaseformysql
- triggers
- workemployee

SQL File 1 task9 triggers SQL File 4* sql comprehension*

Limit to 1000 rows

```

121 -- 5. Retrieve the employee names and their positions for the employees whose salaries are above Rs.50,000.
122 SELECT Emp_name, Position
123 FROM Employee
124 WHERE Salary > 50000;
125

```

Result Grid

Emp_name	Position
Alice Johnson	Manager
Diana Prince	Manager

Navigation: Schemas

- bankdatabase
- colleges
- company
- db
- education
- employee
- employee1
- global_store_db
- india
- manager_details
- places
- product
- sales
- school
- school2
- store
- studentjoin
- sys
- testdatabaseformysql
- triggers
- workemployee

Information: No object selected

Navigator: SQL File 1 task9 triggers SQL File 4* sql comprehension

SCHMAS

Filter objects

- bankdatabase
- colleges
- company
- db
- education
- employee
- employee1
- global_store_db
- india
- manager_details
- places
- product

```

126 -- 6. List the customer names who registered before 2022-01-01 and have not issued any books yet.
127 * SELECT Customer.Customer_name
128 FROM Customer
129 LEFT JOIN IssueStatus ON Customer.Customer_Id = IssueStatus.Issued_cust_id
130 WHERE Customer.Reg_date < '2022-01-01' AND IssueStatus.Issue_Id IS NULL;

```

Result Grid | Filter Rows: | Exports | Wrap Cell Contents: |

Customer_name

Navigator: SQL File 1 task9 triggers SQL File 4* sql comprehension

SCHMAS

Filter objects

- bankdatabase
- colleges
- company
- db
- education
- employee
- employee1
- global_store_db
- india
- manager_details
- places
- product
- sales
- school
- school2
- store
- studentjoin
- sys
- testdatabaseofmysd
- triggers
- workemployee

```

132 -- 7. Display the branch numbers and the total count of employees in each branch.
133 * SELECT Branch_no, COUNT(*) AS Total_Employees
134 FROM Employee
135 GROUP BY Branch_no;
136

```

Result Grid | Filter Rows: | Exports | Wrap Cell Contents: |

Branch_no	Total_Employees
1	2
2	2
3	1

No object selected

Navigator: SQL File 1 task9 triggers SQL File 4* sql comprehension

SCHMAS

Filter objects

- bankdatabase
- colleges
- company
- db
- education
- employee
- employee1
- global_store_db
- india
- manager_details
- places
- product
- sales
- school
- school2
- store
- studentjoin
- sys

```

137 -- 8. Display the names of customers who have issued books in the month of June 2023.
138 * SELECT DISTINCT Customer.Customer_name
139 FROM Customer
140 JOIN IssueStatus ON Customer.Customer_Id = IssueStatus.Issued_cust_id
141 WHERE IssueStatus.Issue_date BETWEEN '2023-06-01' AND '2023-06-30';
142
143

```

Result Grid | Filter Rows: | Exports | Wrap Cell Contents: |

Customer_name

Navigator: SQL File 1 task9 triggers SQL File 4* sql comprehension

SCHMAS

Filter objects

- bankdatabase
- colleges
- company
- db
- education
- employee
- employee1
- global_store_db
- india
- manager_details
- places
- product
- sales
- school

```

143
144 -- 9. Retrieve book_title from book table containing history.
145 * SELECT Book_title
146 FROM Books
147 WHERE Book_title LIKE '%history%';
148
149

```

Result Grid | Filter Rows: | Exports | Wrap Cell Contents: |

Book_title

Navigator SQL File 1 task9 triggers SQL File 4* sql comprehension

149

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

151 • SELECT Branch_no, COUNT(*) AS Employee_Count

152 FROM Employee

153 GROUP BY Branch_no

154 HAVING COUNT(*) > 5;

155

Result Grid Filter Rows: Export: Wrap Cell Contents: 15

Branch_no	Employee_Count
-----------	----------------

Result Grid Form Editor Field Types

152 FROM Employee

153 GROUP BY Branch_no

154 HAVING COUNT(*) > 5;

155

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

157 • SELECT Employee.Emp_name, Branch.Branch_address

158 FROM Employee

159 JOIN Branch ON Employee.Branch_no = Branch.Branch_no

160 WHERE Employee.Position = 'Manager';

161

Result Grid Filter Rows: Export: Wrap Cell Contents: 15

Emp_name	Branch_address
Alice Johnson	123 Main St, Cityville
Diana Prince	456 Elm St, Townsville

Result Grid Form Editor

152 FROM Employee

153 GROUP BY Branch_no

154 HAVING COUNT(*) > 5;

155

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

157 • SELECT Employee.Emp_name, Branch.Branch_address

158 FROM Employee

159 JOIN Branch ON Employee.Branch_no = Branch.Branch_no

160 WHERE Employee.Position = 'Manager';

161

Result Grid Filter Rows: Export: Wrap Cell Contents: 15

Emp_name	Branch_address
Alice Johnson	123 Main St, Cityville
Diana Prince	456 Elm St, Townsville

Result Grid Form Editor