



SCHEMAS

Filter objects

- example
 - Tables
 - employee
 - employee_department
 - employee_salary
 - Views
 - Stored Procedures
 - Functions
- mydb
- sales
- sys

Administration Schemas

Information

No object selected



```
1
2 • create table employee(
3     EID int,
4     FirstName varchar(50),
5     LastName varchar(50),
6     DOB date,
7     Gender varchar(10)
8 );
9
10 • INSERT INTO employee (EID, FirstName, LastName, DOB, Gender) VALUES
11 (121, 'John', 'Doe', '1990-05-15', 'Male'),
12 (244, 'Jane', 'Smith', '1988-09-23', 'Female'),
13 (123, 'Michael', 'Johnson', '1995-02-10', 'Male'),
14 (490, 'Emily', 'Brown', '1993-11-08', 'Female'),
15 (544, 'Christopher', 'Davis', '1992-07-20', 'Male'),
16 (926, 'Jessica', 'Wilson', '1987-04-30', 'Female'),
17 (733, 'Matthew', 'Martinez', '1994-12-18', 'Male'),
18 (508, 'Sarah', 'Anderson', '1991-08-12', 'Female'),
19 (889, 'Daniel', 'Taylor', '1989-03-25', 'Male'),
20 (1001, 'Amanda', 'Thomas', '1996-06-05', 'Female'),
21 (191, 'James', 'Jackson', '1986-01-17', 'Male'),
22 (112, 'Lauren', 'White', '1997-10-11', 'Female'),
23 (1163, 'Ryan', 'Harris', '1998-09-03', 'Male'),
24 (400, 'Rachel', 'Miller', '1990-07-28', 'Female'),
25 (905, 'Andrew', 'Wilson', '1985-03-14', 'Male'),
26 (886, 'Jennifer', 'Lee', '1992-11-25', 'Female'),
27 (185, 'David', 'Thompson', '1984-06-19', 'Male'),
28 (345, 'Melissa', 'Clark', '1993-04-02', 'Female'),
```



SCHEMAS

Filter objects

- example
 - Tables
 - employee
 - employee_department
 - employee_salary
 - Views
 - Stored Procedures
 - Functions
- mydb
- sales
- sys

No object selected

```
1  -- select all rows from tables
2
3  •  select * from employee;
4
5  •  select * from employee_salary;
6
7  •  select * from employee_department;
8
9  -- select employees who are born after 1990 and arrange them in descending order of age
10 •  select * from employee
11     WHERE Year(DOB) > 1990
12     ORDER BY DOB;
13
14 -- show the count of employees of each gender
15 •  select distinct(Gender), count(Gender) as total from employee
16     GROUP BY Gender;
17
18 -- returns first five EID and name in ascending order of EID
19 •  select EID , Firstname from employee
20     order by EID
21     limit 5 ;
22
23
24 -- limit shows the next 3 rows after offset omits the first 5 rows
25 •  select EID , Firstname from employee
26     order by EID
27     limit 3
28     offset 5 ;
```



SCHEMAS

Filter objects

- example
 - Tables
 - employee
 - employee_department
 - employee_salary
 - Views
 - Stored Procedures
 - Functions
- mydb
- sales
- sys

Administration Schemas

Information

No object selected

```
8
9  -- perform right outer join to return all values of right table(employee_salary) and return null if it doesn't occur in right table(employee)
10 • select * from employee e
11 right outer join employee_salary es on e.EID = es.EID;
12
13 -- perform full outer join as a union of left and right outer joins
14 -- using aliasing and selecting specific columns from both tables as per requirement
15 • select es.EID, concat(FirstName," ",LastName)as FullName,Salary from employee e
16 left outer join employee_salary es on e.EID = es.EID
17 union
18 select es.EID, concat(FirstName," ",LastName) as Fullname,Salary from employee e
19 right outer join employee_salary es on e.EID = es.EID;
20
21 -- select specific columns and joining all 3 tables
22 • select es.EID,concat(FirstName," ",LastName)as FullName, Salary , DepartmentName from employee e
23 join employee_salary es on e.EID = es.EID
24 join employee_department ed on e.EID = ed.EID;
25
26 -- using where and order by clause in join tables
27 • select es.EID,concat(FirstName," ",LastName)as FullName, Salary , DepartmentName from employee e
28 join employee_salary es on e.EID = es.EID
29 join employee_department ed on e.EID = ed.EID
30 where salary > 50000 and DepartmentName is NOT NULL
31 order by salary;
32
33
34
35
```

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

example

- Tables
 - employee
 - employee_department
 - employee_salary
- Views
- Stored Procedures
- Functions

mydb

sales

sys

Administration Schemas

Information

No object selected

creating tables Basic SQL joins groupby_having_aggregate x subquery insert_update_delete

Limit to 1000 rows

```
1
2  -- shows number of employees in each unique department
3  • select distinct(DepartmentName) , count(DepartmentName) as total from employee_department
4  where DepartmentName is NOT NULL
5  group by DepartmentName;
6
7  -- shows number of employees in each unique department but only if number of employees is more than 1
8  -- we cannot use where clause with aggregate functions hence we use having clause
9  • select distinct(DepartmentName) , count(DepartmentName) as total from employee_department
10 where DepartmentName is NOT NULL
11 group by DepartmentName
12 having total > 1;
13
14 -- aggregate functions (sum, count, avg, max, min)
15 • select max(salary) as Maximum_Salary, min(salary) as Minimum_Salary, avg(salary) as Average_Salary, sum(salary) as Total_Salary from employee_salary;
16
17 • select gender , count(gender)
18 from employee
19 group by gender;
20
21
22
23
24
```




SCHEMAS

Filter objects

- example
 - Tables
 - employee
 - employee_department
 - employee_salary
 - Views
 - Stored Procedures
 - Functions
- mydb
- sales
- sys

Administration Schemas

Information

No object selected



Limit to 1000 rows

```
1  -- Subquery in From
2
3  • Select a.EID, AllAvgSalary
4  From
5  (Select EID, Salary, AVG(Salary) over () as AllAvgSalary
6   From employee_salary) a
7  Order by a.Salary;
8
9  -- to return eid and salary of people born after 1990
10 • Select EID , Salary
11 From employee_salary
12 where EID in (
13     Select EID
14     From employee
15     where Year(DOB) > 1990);
16
17
18 -- retrieve all employees who do not have a salary record in the employee_salary table
19 • SELECT *
20 FROM employee e
21 WHERE NOT EXISTS (
22     SELECT 1
23     FROM employee_salary es
24     WHERE e.EID = es.EID
25 );
26
27
28 -- retrieve all employees who have a salary record in the employee_salary table
```



SCHEMAS

Filter objects

- example
 - Tables
 - employee
 - employee_department
 - employee_salary
 - Views
 - Stored Procedures
 - Functions
- mydb
- sales
- sys

Administration Schemas

Information

No object selected

```
24     WHERE e.EID = es.EID
25   );
26
27
28   -- retrieve all employees who have a salary record in the employee_salary table
29   SELECT *
30   FROM employee e
31   WHERE EXISTS (
32     SELECT 1
33     FROM employee_salary es
34     WHERE e.EID = es.EID
35   );
36
37   -- using case to find out the salary increment according to the department name of the employee
38   select e.EID , FirstName , Salary , DepartmentName ,
39   case
40     when DepartmentName = "Marketing" then Salary + (Salary*0.10)
41     when DepartmentName = "Finance" then Salary + (Salary*0.07)
42     when DepartmentName = "Operations" then Salary + (Salary*0.03)
43     when DepartmentName = "Human Resources" then Salary + (Salary*0.0001)
44     else Salary + (Salary*0.05)
45
46   end
47   as incremented_salary
48   from employee e
49   join employee_salary es on e.EID = es.EID
50   join employee_department ed on e.EID = ed.EID
51   where DepartmentName is NOT NULL;
```



SCHEMAS

Filter objects

- example
 - Tables
 - employee
 - employee_department
 - employee_salary
 - Views
 - Stored Procedures
 - Functions
- mydb
- sales
- sys

Administration Schemas

Information

No object selected



```
1 • INSERT INTO employee (EID, FirstName, LastName, DOB, Gender)
2   VALUES (125, 'Ruby', 'Doe', '1980-05-10', 'Female');
3
4 • select * from employee_department
5   order by EID ;
6
7 • UPDATE employee_department
8   SET DepartmentName = "New Department"
9   WHERE DepartmentName is null ;
10
11 • Delete from employee
12   where EID = 121;
13
14   -- to check for duplicates in the tables
15 • SELECT EID, COUNT(*)
16   FROM employee
17   GROUP BY EID
18   HAVING COUNT(*) > 1;
19
20 • SELECT EID, COUNT(*)
21   FROM employee_department
22   GROUP BY EID
23   HAVING COUNT(*) > 1;
24
25
26
27
28
```