

Employee Database with tables department and employees

department table ↓

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'employee' selected, showing 'department' and 'employees' tables. The main editor shows a query window with the following SQL code:

```
4  dept_location VARCHAR(100)
5  );
6  explain department;
7
8  CREATE TABLE employees(
```

The 'Result Grid' at the bottom shows the structure of the 'department' table:

Field	Type	Null	Key	Default	Extra
dept_id	int	NO	PRI	NULL	
dept_location	varchar(100)	YES		NULL	
dept_name	varchar(45)	YES		NULL	

Below the result grid, the 'Table: department' information is displayed:

Columns:

- dept_id int PK
- dept_location varchar(100)
- dept_name varchar(45)

employees table ↓

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'employee' selected, showing 'department' and 'employees' tables. The main editor shows a query window with the following SQL code:

```
12 address VARCHAR(100),
13 salary DECIMAL(10,2),
14 dep_id INT
15 );
16 explain employees;
```

The 'Result Grid' at the bottom shows the structure of the 'employees' table:

Field	Type	Null	Key	Default	Extra
emp_id	int	NO	PRI	NULL	
emp_name	varchar(100)	NO		NULL	
job_name	varchar(100)	NO		NULL	
address	varchar(100)	NO		NULL	
salary	decimal(10,2)	NO		NULL	
dep_id	int	NO		NULL	

Below the result grid, the 'Table: department' information is displayed (this is a visual artifact from the previous screenshot):

Columns:

- dept_id int PK
- dept_location varchar(100)
- dept_name varchar(45)

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator: loyee.department kk* employee.department cars - Schema cars Administration - Data Export ordering_system customers

SCHEMAS

Filter objects

- Stored Procedures
- Functions
- employee**
 - Tables
 - department
 - employees
 - Views
 - Stored Procedures
 - Functions
- flask_db
- ordering_system
 - Tables
 - customers
 - orders

Administration Schemas

Information

Table: department

Columns:

- dept_id int PK
- dept_location varchar(100)
- dept_name varchar(45)

```
16 • explain employees;
17 • INSERT INTO department VALUES (1,'london','software development'),(2,'birmingham','testing'),(3,'manchester','administ
18 |
19 • SELECT * FROM department;
20
```

Result Grid

dept_id	dept_location	dept_name
1	london	software development
2	birmingham	testing
3	manchester	administration
4	Nottingham	Computer Network
5	glasgow	Information Security
6	Hull	Cloud Computing
7	leeds	Database Administrator
NULL	NULL	NULL

Record Update

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator: qq* x

SCHEMAS

Filter objects

- Stored Procedures
- Functions
- employee**
 - Tables
 - department
 - Columns
 - Indexes
 - Foreign Keys
 - Triggers
 - employees
 - Views
 - Stored Procedures
 - Functions
- flask_db

Administration Schemas

Information

Table: department

Columns:

- dept_id int PK
- dept_location varchar(100)
- dept_name varchar(45)

```
22
23 • UPDATE department
24 SET dept_location = 'Brighton'
25 WHERE dept_id = 6;
26 • SELECT * FROM department;
```

Result Grid

dept_id	dept_location	dept_name
1	london	software development
2	birmingham	testing
3	manchester	administration
4	Nottingham	Computer Network
5	glasgow	Information Security
6	Brighton	Cloud Computing
7	leeds	Database Administrator
NULL	NULL	NULL

department 10 x

Apply Revert Co

Deleting a record ↓

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'flask_db' selected. The main query editor contains the following SQL script:

```
25 WHERE dept_id = 6;  
26 • SELECT * FROM department;  
27 • DELETE FROM department  
28 WHERE dept_id = '3';  
29 • SELECT * FROM department;
```

The 'Result Grid' shows the output of the query, displaying a table with 3 columns: dept_id, dept_location, and dept_name. The data is as follows:

dept_id	dept_location	dept_name
1	london	software development
2	birmingham	testing
4	Nottingham	Computer Network
5	glasgow	Information Security
6	Brighton	Cloud Computing
7	leeds	Database Administrator

The bottom status bar indicates 'Table: department' and 'Columns: dept_id (int PK), dept_location (varchar(100)), dept_name (varchar(45))'.

Inner Join ↓

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'flask_db' selected. The main query editor contains the following SQL script:

```
26 • SELECT * FROM department;  
27 • DELETE FROM department  
28 WHERE dept_id = '3';  
29 • SELECT * FROM department;  
30 • SELECT * FROM employees;  
31  
32 • SELECT *  
33 FROM employees  
34 INNER JOIN department  
35 WHERE department.dept_id = employees.dept_id AND employees.salary = 4500.00;  
36
```

The 'Result Grid' shows the output of the query, displaying a table with 9 columns: emp_id, emp_name, job_name, address, salary, dept_id, dept_id, dept_location, and dept_name. The data is as follows:

emp_id	emp_name	job_name	address	salary	dept_id	dept_id	dept_location	dept_name
3	abc	developer	nottingham	4500.00	1	1	london	software development
6	Frank	Analyst	Boston	4500.00	1	1	london	software development

The bottom status bar indicates 'Table: department' and 'Columns: dept_id (int PK), dept_location (varchar(100)), dept_name (varchar(45))'.

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

- cars
 - Tables
 - Views
 - Stored Procedures
 - Functions
- employee
 - Tables
 - department
 - employees
 - Views
 - Stored Procedures
 - Functions
- flask_db
 - ordering_system

Administration Schemas

Information

No object selected

employees employees department

Limit to 1000 rows

```

32 • SELECT *
33 FROM employees
34 INNER JOIN department
35 WHERE department.dept_id = employees.dept_id AND employees.salary = 4500.00;
36 • use employee;
37 • select * from employees,department where department.dept_id = employees.dept_id;
38
39

```

Result Grid

emp_id	emp_name	job_name	address	salary	dept_id	dept_id	dept_location	dept_name
1	kk	Data Analyst	London	5000.00	1	1	london	software development
2	emma	tester	Birmingham	4000.00	2	2	birmingham	testing
3	abc	developer	nottingham	4500.00	1	1	london	software development
4	xyz	HR	leeds	3500.00	7	7	leeds	Database Administrator
5	Clare	Manager	Dudley	3000.00	4	4	Nottingham	Computer Network
6	Frank	Analyst	Boston	4500.00	1	1	london	software development
7	emp1	analyst	leeds	5000.00	1	1	london	software development

Sorting data in ascending order ↓

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

- cars
 - Tables
 - Views
 - Stored Procedures
 - Functions
- employee
 - Tables
 - department
 - employees
 - Views
 - Stored Procedures
 - Functions
- flask_db
 - ordering_system

Administration Schemas

Information

No object selected

employees employees department

Limit to 1000 rows

```

34 INNER JOIN department
35 WHERE department.dept_id = employees.dept_id AND employees.salary = 4500.00;
36 • use employee;
37 • select * from employees,department where department.dept_id = employees.dept_id;
38 • SELECT emp_name,salary,job_name FROM employees ORDER BY salary asc;
39

```

Result Grid

emp_name	salary	job_name
Clare	3000.00	Manager
xyz	3500.00	HR
emma	4000.00	tester
abc	4500.00	developer
Frank	4500.00	Analyst
kk	5000.00	Data Analyst
emp1	5000.00	analyst

employees 8 x

Read Only

Conte

Using comparison operator ↓

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

cars

- Tables
- Views
- Stored Procedures
- Functions

employee

- department
- employees
- Views
- Stored Procedures
- Functions

flask_db

- ordering_system

Administration Schemas

Information

No object selected

employees employees department

Limit to 1000 rows

```
34 INNER JOIN department
35 WHERE department.dept_id = employees.dept_id AND employees.salary = 4500.00;
36 use employee;
37 select * from employees,department where department.dept_id = employees.dept_id;
38 SELECT emp_name,salary,job_name FROM employees ORDER BY salary asc;
39 select * from employees where salary between 3500 and 4500;
```

Result Grid

emp_id	emp_name	job_name	address	salary	dept_id
2	emma	tester	Birmingham	4000.00	2
3	abc	developer	nottingham	4500.00	1
4	xyz	HR	leeds	3500.00	7
6	Frank	Analyst	Boston	4500.00	1
*	NULL	NULL	NULL	NULL	NULL

employees 9 x

Apply