```
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4
Server version: 5.1.50-community MySQL Community Server (GPL)

Copyright (c) 2000, 2010, Oracle and/or its affiliates. All rights reserved.
This software comes with ABSOLUTELY NO WARRANTY. This is free software,
and you are welcome to modify and redistribute it under the GPL v2 license

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create database ms;
Query OK, 1 row affected (0.00 sec)

mysql> use ms;
Database changed
mysql> CREATE TABLE employees (
    ->     employee_id INT PRIMARY KEY,
    ->     first_name VARCHAR(50),
    ->     last_name VARCHAR(50),
    ->     hire_date DATE,
    ->     department_id INT,
    ->     salary DECIMAL(10, 2)
    -> );
Query OK, 0 rows affected (0.02 sec)

mysql> CREATE TABLE departments (
    ->     department_id INT PRIMARY KEY,
    ->     department_name VARCHAR(100)
    -> );
Query OK, 0 rows affected (0.03 sec)
```

```
mysql>
mysql> INSERT INTO departments (department_id, department_name)
    -> VALUES
    -> (1, 'Sales'),
    -> (2, 'Marketing'),
    -> (3, 'Engineering'),
    -> (4, 'HR'),
    -> (5, 'Finance');
Query OK, 5 rows affected (0.00 sec)
Records: 5  Duplicates: 0  Warnings: 0

mysql>
mysql> INSERT INTO employees (employee_id, first_name, last_name, hire_date, department_id,
salary) VALUES
    -> (1, 'John', 'Doe', '2015-06-23', 1, 55000.00),
    -> (2, 'Jane', 'Smith', '2018-02-10', 2, 62000.00),
    -> (3, 'Samuel', 'Adams', '2012-11-04', 3, 90000.00),
    -> (4, 'Emily', 'Clark', '2020-03-15', 1, 45000.00),
    -> (5, 'Daniel', 'Harris', '2016-07-19', 4, 49000.00),
    -> (6, 'Rachel', 'Baker', '2019-10-01', 3, 95000.00),
    -> (7, 'Paul', 'Jones', '2017-09-13', 2, 55000.00),
    -> (8, 'Sophia', 'Taylor', '2014-12-21', 5, 73000.00),
    -> (9, 'Michael', 'Lee', '2011-08-14', 4, 47000.00),
    -> (10, 'Olivia', 'King', '2022-01-30', 3, 98000.00);
Query OK, 10 rows affected (0.02 sec)
Records: 10  Duplicates: 0  Warnings: 0

mysql> #1.To concat the first_name and last_name of employees into a single column named
full_name.
mysql>  select first_name,last_name,concat(first_name," ",last_name) as full_name from employees;
+------------+-----------+---------------+
```

| first_name | last_name | full_name |

+------------+-----------+---------------+

| John | Doe | John Doe |

| Jane | Smith | Jane Smith |

| Samuel | Adams | Samuel Adams |

| Emily | Clark | Emily Clark |

| Daniel | Harris | Daniel Harris |

| Rachel | Baker | Rachel Baker |

| Paul | Jones | Paul Jones |

| Sophia | Taylor | Sophia Taylor |

| Michael | Lee | Michael Lee |

| Olivia | King | Olivia King |

+------------+-----------+---------------+

10 rows in set (0.02 sec)

mysql> #2.Query that retrieves the first and last names of employees, as well as their department names by using a subquery inside the SELECT statement.

mysql> SELECT e.first_name, e.last_name, (SELECT d.department_name FROM departments d WHERE d.department_id = e.department_id) AS department_name FROM employees e;

+------------+-----------+-----------------+

| first_name | last_name | department_name |

+------------+-----------+-----------------+

| John | Doe | Sales |

| Jane | Smith | Marketing |

| Samuel | Adams | Engineering |

| Emily | Clark | Sales |

| Daniel | Harris | HR |

| Rachel | Baker | Engineering |

| Paul | Jones | Marketing |

| Sophia | Taylor | Finance |

| Michael | Lee | HR |

| Olivia | King | Engineering |

```
+------------+-----------+-----------------+
```

10 rows in set (0.00 sec)

mysql> #3.Query counts the number of employees in each department.

mysql> SELECT department_id, COUNT(*) AS employee_count FROM employees GROUP BY department_id;

```
+---------------+----------------+
| department_id | employee_count |
+---------------+----------------+
|             1 |              2 |
|             2 |              2 |
|             3 |              3 |
|             4 |              2 |
|             5 |              1 |
+---------------+----------------+
```

5 rows in set (0.00 sec)

mysql> #4.Query to lists all departments and counts the number of employees in each department .

mysql> SELECT d.department_name, COUNT(e.employee_id) AS employee_count FROM departments d LEFT JOIN employees e ON d.department_id = e.department_id GROUP BY d.department_name;

```
+-----------------+----------------+
| department_name | employee_count |
+-----------------+----------------+
| Engineering     |              3 |
| Finance         |              1 |
| HR              |              2 |
| Marketing       |              2 |
| Sales           |              2 |
+-----------------+----------------+
```

5 rows in set (0.00 sec)

mysql> #5.To find employees with salary greater than average salary of their department

mysql> SELECT e.first_name, e.last_name, e.salary, e.department_id FROM employees e WHERE e.salary > (SELECT AVG(salary) FROM employees WHERE department_id = e.department_id);

+------------+-----------+----------+---------------+

| first_name | last_name | salary | department_id |

+------------+-----------+----------+---------------+

| John | Doe | 55000.00 | 1 |

| Jane | Smith | 62000.00 | 2 |

| Daniel | Harris | 49000.00 | 4 |

| Rachel | Baker | 95000.00 | 3 |

| Olivia | King | 98000.00 | 3 |

+------------+-----------+----------+---------------+

5 rows in set (0.00 sec)

mysql> #6.To get all employee names in each department.

mysql> SELECT d.department_name, GROUP_CONCAT(e.first_name," ", e.last_name) AS employee_names FROM departments d JOIN

    -> employees e ON d.department_id = e.department_id GROUP BY d.department_name;

+-----------------+--------------------------------------+

| department_name | employee_names |

+-----------------+--------------------------------------+

| Engineering | Rachel Baker,Samuel Adams,Olivia King |

| Finance | Sophia Taylor |

| HR | Michael Lee,Daniel Harris |

| Marketing | Paul Jones,Jane Smith |

| Sales | Emily Clark,John Doe |

+-----------------+--------------------------------------+

5 rows in set (0.00 sec)

mysql> #7.Subquery to find the employee with the highest salary in each department.

mysql> SELECT e.department_id, e.first_name, e.last_name, e.salary FROM employees e WHERE (e.department_id, e.salary) IN (SELECT department_id, MAX(salary) FROM employees GROUP BY department_id);

+---------------+------------+-----------+----------+

| department_id | first_name | last_name | salary |

+---------------+------------+-----------+----------+

| 1 | John | Doe | 55000.00 |

| 2 | Jane | Smith | 62000.00 |

| 4 | Daniel | Harris | 49000.00 |

| 5 | Sophia | Taylor | 73000.00 |

| 3 | Olivia | King | 98000.00 |

+---------------+------------+-----------+----------+

5 rows in set (0.00 sec)

mysql> #8.Group employees by hire year and calculate the total salary for each year

mysql> SELECT DATE_FORMAT(hire_date,'%Y') AS hire_year, SUM(salary) AS total_salary FROM employees GROUP BY hire_year;

+-----------+--------------+

| hire_year | total_salary |

+-----------+--------------+

| 2011 | 47000.00 |

| 2012 | 90000.00 |

| 2014 | 73000.00 |

| 2015 | 55000.00 |

| 2016 | 49000.00 |

| 2017 | 55000.00 |

| 2018 | 62000.00 |

| 2019 | 95000.00 |

| 2020 | 45000.00 |

| 2022 | 98000.00 |

+-----------+--------------+

10 rows in set (0.00 sec)

mysql> #9.Group departments and count the number of employees in each department

mysql> SELECT d.department_name, COUNT(e.employee_id) AS employee_count FROM departments d LEFT JOIN employees e ON d.department_id = e.department_id GROUP BY d.department_name;

+-----------------+----------------+

| department_name | employee_count |

+-----------------+----------------+

| Engineering | 3 |

| Finance | 1 |

| HR | 2 |

| Marketing | 2 |

| Sales | 2 |

+-----------------+----------------+

5 rows in set (0.00 sec)

mysql> #10.Find the highest salary in each department

mysql> SELECT department_id, MAX(salary) AS max_salary FROM employees GROUP BY department_id;

+---------------+------------+

| department_id | max_salary |

+---------------+------------+

| 1 | 55000.00 |

| 2 | 62000.00 |

| 3 | 98000.00 |

| 4 | 49000.00 |

| 5 | 73000.00 |

+---------------+------------+

5 rows in set (0.01 sec)

mysql> #11.Group employees by department and show the average salary in each department.

```
mysql> SELECT department_id, AVG(salary) AS average_salary FROM employees GROUP BY department_id;
+---------------+----------------+
| department_id | average_salary |
+---------------+----------------+
|             1 |   50000.000000 |
|             2 |   58500.000000 |
|             3 |   94333.333333 |
|             4 |   48000.000000 |
|             5 |   73000.000000 |
+---------------+----------------+
5 rows in set (0.00 sec)

mysql> #12.Find departments with more than 5 employees
mysql>  SELECT department_id FROM employees GROUP BY department_id HAVING COUNT(*) > 5;
Empty set (0.00 sec)

mysql> #13.List departments where the average salary is greater than 50,000.
mysql>  SELECT department_id FROM employees GROUP BY department_id HAVING AVG(salary) > 50000;
+---------------+
| department_id |
+---------------+
|             2 |
|             3 |
|             5 |
+---------------+
3 rows in set (0.00 sec)

mysql> #14.List employees who earn more than 60,000 and belong to a department with more than 3 employees.
```

mysql> SELECT employee_id, first_name, last_name FROM employees WHERE salary > 60000 AND department_id IN (SELECT department_id FROMemployees GROUP BY department_id HAVING COUNT(*) > 3);

Empty set (0.00 sec)

mysql> #15.Show departments where there is more than one employee with a salary over 100,000.

mysql> SELECT department_id FROM employees WHERE salary > 100000 GROUP BY department_id HAVING COUNT(*) > 1;

Empty set (0.00 sec)

mysql> #16.Delete an employee with a specific employee_id (e.g., 5)

mysql> DELETE FROM employees WHERE employee_id = 5;

Query OK, 1 row affected (0.00 sec)

mysql> #17.Delete employees who have a salary lower than 30,000

mysql> DELETE FROM employees WHERE salary < 30000;

Query OK, 0 rows affected (0.01 sec)

mysql> Query OK, 0 rows affected (0.00 sec)