```
Microsoft Windows [Version 10.0.26100.4770]
(c) Microsoft Corporation. All rights reserved.

C:\Program Files\MySQL\MySQL Server 9.2\bin>mysql -u root -p
Enter password: ************
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 9.2.0 MySQL Community Server - GPL

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql> CREATE DATABASE college_db;
Query OK, 1 row affected (0.01 sec)

mysql> USE college_db;
Database changed
mysql> CREATE TABLE student (
    ->     student_id INT PRIMARY KEY,
    ->     name VARCHAR(50),
    ->     major VARCHAR(50),
    ->     cgpa DECIMAL(3,2)
    -> );
Query OK, 0 rows affected (0.06 sec)

mysql>
mysql> CREATE TABLE enrollment (
    ->     enrollment_id INT PRIMARY KEY,
    ->     student_id INT,
    ->     course VARCHAR(100),
    ->     grade VARCHAR(5),
    ->     credits INT
    ->
    -> );
Query OK, 0 rows affected (0.02 sec)

mysql> INSERT INTO student (student_id, name, major, cgpa) VALUES
    -> (1, 'Alice', 'Computer Science', 3.8),
    -> (2, 'Bob', 'Mathematics', 3.2),
    -> (3, 'Carol', 'Physics', 3.9),
    -> (4, 'David', 'Computer Science', 3.5),
    -> (5, 'Emma', 'Mathematics', 3.7);
Query OK, 5 rows affected (0.05 sec)
Records: 5  Duplicates: 0  Warnings: 0

mysql> INSERT INTO enrollment (enrollment_id, student_id, course, grade, credits) VALUES
    -> (101, 1, 'Database', 'A', 3),
    -> (102, 1, 'Programming', 'B+', 4),
    -> (103, 2, 'Calculus', 'A-', 4),
    -> (104, 3, 'Quantum Physics', 'A', 3),
    -> (105, 4, 'Data Structures', 'B', 3),
    -> (106, 5, 'Statistics', 'A', 3),
    -> (107, 2, 'Algebra', 'B+', 3);
Query OK, 7 rows affected (0.00 sec)
Records: 7  Duplicates: 0  Warnings: 0

mysql> SELECT s.name, e.enrollment_id, e.course, e.grade
```

```
    -> FROM student s
    -> INNER JOIN enrollment e ON s.student_id = e.student_id;
+-------+---------------+-----------------+-------+
| name  | enrollment_id | course          | grade |
+-------+---------------+-----------------+-------+
| Alice |           101 | Database        | A     |
| Alice |           102 | Programming     | B+    |
| Bob   |           103 | Calculus        | A-    |
| Carol |           104 | Quantum Physics | A     |
| David |           105 | Data Structures | B     |
| Emma  |           106 | Statistics      | A     |
| Bob   |           107 | Algebra         | B+    |
+-------+---------------+-----------------+-------+
7 rows in set (0.02 sec)

mysql> SELECT s.name, e.enrollment_id, e.course, e.grade
    -> FROM student s
    -> LEFT JOIN enrollment e ON s.student_id = e.student_id;
+-------+---------------+-----------------+-------+
| name  | enrollment_id | course          | grade |
+-------+---------------+-----------------+-------+
| Alice |           102 | Programming     | B+    |
| Alice |           101 | Database        | A     |
| Bob   |           107 | Algebra         | B+    |
| Bob   |           103 | Calculus        | A-    |
| Carol |           104 | Quantum Physics | A     |
| David |           105 | Data Structures | B     |
| Emma  |           106 | Statistics      | A     |
+-------+---------------+-----------------+-------+
7 rows in set (0.03 sec)
```

```
mysql> SELECT s.name, e.enrollment_id, e.course, e.grade
    -> FROM student s
    -> RIGHT JOIN enrollment e ON s.student_id = e.student_id;
+-------+---------------+-----------------+-------+
| name  | enrollment_id | course          | grade |
+-------+---------------+-----------------+-------+
| Alice |           101 | Database        | A     |
| Alice |           102 | Programming     | B+    |
| Bob   |           103 | Calculus        | A-    |
| Carol |           104 | Quantum Physics | A     |
| David |           105 | Data Structures | B     |
| Emma  |           106 | Statistics      | A     |
| Bob   |           107 | Algebra         | B+    |
+-------+---------------+-----------------+-------+
7 rows in set (0.00 sec)

mysql> SELECT s.student_id, s.name, e.enrollment_id, e.course, e.grade
    -> FROM student s
    -> LEFT JOIN enrollment e ON s.student_id = e.student_id
    ->
    -> UNION
    ->
    ->
    -> SELECT s.student_id, s.name, e.enrollment_id, e.course, e.grade
    -> FROM student s
    -> RIGHT JOIN enrollment e ON s.student_id = e.student_id;
+------------+-------+---------------+-----------------+-------+
| student_id | name  | enrollment_id | course          | grade |
+------------+-------+---------------+-----------------+-------+
|          1 | Alice |           102 | Programming     | B+    |
|          1 | Alice |           101 | Database        | A     |
|          2 | Bob   |        107 | Algebra          | B+    |
|          2 | Bob   |        103 | Calculus         | A-    |
|          3 | Carol |        104 | Quantum Physics  | A     |
|          4 | David |        105 | Data Structures  | B     |
|          5 | Emma  |        106 | Statistics       | A     |
+------------+-------+------------+------------------+-------+
7 rows in set (0.03 sec)

mysql> SELECT
    ->     s.name,
    ->     s.major,
    ->     SUM(e.credits) AS total_credits,
    ->     ROUND(AVG(
    ->         CASE e.grade
    ->             WHEN 'A' THEN 4.0
    ->             WHEN 'A-' THEN 3.7
    ->             WHEN 'B+' THEN 3.3
    ->             WHEN 'B' THEN 3.0
    ->             ELSE NULL
    ->         END
    ->     ), 2) AS avg_gpa
    -> FROM student s
    -> LEFT JOIN enrollment e ON s.student_id = e.student_id
    -> GROUP BY s.student_id, s.name, s.major;
+-------+------------------+---------------+---------+
| name  | major            | total_credits | avg_gpa |
+-------+------------------+---------------+---------+
| Alice | Computer Science |             7 |    3.65 |
| Bob   | Mathematics      |             7 |    3.50 |
| Carol | Physics          |             3 |    4.00 |
| David | Computer Science |             3 |    3.00 |
| Emma  | Mathematics      |             3 |    4.00 |
+-------+------------------+---------------+---------+
5 rows in set (0.04 sec)

mysql> SELECT * FROM student
    -> WHERE cgpa > (
    ->     SELECT cgpa FROM student WHERE name = 'Bob'
    -> );
+------------+-------+------------------+------+
| student_id | name  | major            | cgpa |
+------------+-------+------------------+------+
|          1 | Alice | Computer Science | 3.80 |
|          3 | Carol | Physics          | 3.90 |
|          4 | David | Computer Science | 3.50 |
|          5 | Emma  | Mathematics      | 3.70 |
+------------+-------+------------------+------+
4 rows in set (0.04 sec)

mysql> SELECT course FROM enrollment
    -> WHERE student_id IN (
    ->     SELECT student_id FROM student WHERE major = 'Computer Science'
    -> );
+-----------------+
| course          |
+-----------------+
| Database        |
| Programming     |
| Data Structures |
+-----------------+
3 rows in set (0.00 sec)
```

```
mysql> SELECT * FROM enrollment
    -> WHERE student_id = (
    ->     SELECT student_id FROM student
    ->     WHERE cgpa = (SELECT MAX(cgpa) FROM student)
    -> );
+---------------+------------+-----------------+-------+---------+
| enrollment_id | student_id | course          | grade | credits |
+---------------+------------+-----------------+-------+---------+
|           104 |          3 | Quantum Physics | A     |       3 |
+---------------+------------+-----------------+-------+---------+
1 row in set (0.03 sec)

mysql> SELECT * FROM student
    -> WHERE student_id IN (
    ->     SELECT student_id FROM enrollment
    ->     GROUP BY student_id
    ->     HAVING COUNT(*) > 1
    -> );
+------------+-------+------------------+------+
| student_id | name  | major            | cgpa |
+------------+-------+------------------+------+
|          1 | Alice | Computer Science | 3.80 |
|          2 | Bob   | Mathematics      | 3.20 |
+------------+-------+------------------+------+
2 rows in set (0.03 sec)

mysql> SELECT * FROM student s
    -> WHERE EXISTS (
    ->     SELECT 1 FROM enrollment e
    ->     WHERE e.student_id = s.student_id
    -> );
```

```
+------------+-------+------------------+------+
| student_id | name  | major            | cgpa |
+------------+-------+------------------+------+
|          1 | Alice | Computer Science | 3.80 |
|          2 | Bob   | Mathematics      | 3.20 |
|          3 | Carol | Physics          | 3.90 |
|          4 | David | Computer Science | 3.50 |
|          5 | Emma  | Mathematics      | 3.70 |
+------------+-------+------------------+------+
5 rows in set (0.00 sec)

mysql> SELECT * FROM student s
    -> WHERE NOT EXISTS (
    ->     SELECT 1 FROM enrollment e
    ->     WHERE e.student_id = s.student_id
    -> );
Empty set (0.00 sec)

mysql> SELECT name,
    ->     (SELECT COUNT(*) FROM enrollment e WHERE e.student_id = s.student_id) AS course_count
    -> FROM student s;
+-------+--------------+
| name  | course_count |
+-------+--------------+
| Alice |            2 |
| Bob   |            2 |
| Carol |            1 |
| David |            1 |
| Emma  |            1 |
+-------+--------------+
5 rows in set (0.00 sec)
```

```
mysql> SELECT * FROM student s
    -> WHERE cgpa > (
    ->     SELECT AVG(cgpa) FROM student
    ->     WHERE major = s.major
    -> );
+------------+-------+------------------+------+
| student_id | name  | major            | cgpa |
+------------+-------+------------------+------+
|          1 | Alice | Computer Science | 3.80 |
|          5 | Emma  | Mathematics      | 3.70 |
+------------+-------+------------------+------+
2 rows in set (0.00 sec)

mysql> SELECT * FROM student s
    -> WHERE (
    ->     SELECT SUM(credits) FROM enrollment e WHERE e.student_id = s.student_id
    -> ) > (
    ->     SELECT AVG(total_credits) FROM (
    ->         SELECT student_id, SUM(credits) AS total_credits
    ->         FROM enrollment
    ->         GROUP BY student_id
    ->     ) AS credit_totals
    -> );
+------------+-------+------------------+------+
| student_id | name  | major            | cgpa |
+------------+-------+------------------+------+
|          1 | Alice | Computer Science | 3.80 |
|          2 | Bob   | Mathematics      | 3.20 |
+------------+-------+------------------+------+
2 rows in set (0.03 sec)
```

```
mysql> SELECT * FROM student
    -> WHERE major = 'Computer Science'
    -> AND cgpa > (
    ->     SELECT AVG(cgpa) FROM student
    -> );
+------------+-------+------------------+------+
| student_id | name  | major            | cgpa |
+------------+-------+------------------+------+
|          1 | Alice | Computer Science | 3.80 |
+------------+-------+------------------+------+
1 row in set (0.00 sec)
```