```
mysql> SELECT student.name
    -> FROM student
    -> JOIN department ON student.dept_name = department.dept_name
    -> WHERE department.building = 'Taylor'
    -> AND student.tot_cred > 50
    -> LIMIT 5;
+-----------+
| name      |
+-----------+
| Yamashita |
| Zander    |
| Langer    |
| Palmer    |
| Kanata    |
+-----------+
5 rows in set (0.00 sec)

mysql> SELECT student.ID, student.name, COUNT(takes.course_id) AS num_courses
    -> FROM student
    -> JOIN takes ON student.ID = takes.ID
    -> GROUP BY student.ID, student.name
    -> HAVING COUNT(takes.course_id) >= 2
    -> LIMIT 5;
+-------+-----------+-------------+
| ID    | name      | num_courses |
+-------+-----------+-------------+
| 1000  | Manber    |          13 |
| 10033 | Zelty     |          22 |
| 10076 | Duan      |          14 |
| 1018  | Colin     |          22 |
| 10204 | Mediratta |          12 |
+-------+-----------+-------------+
5 rows in set (0.09 sec)

mysql> SELECT DISTINCT s1.ID
    -> FROM takes s1
    -> WHERE NOT EXISTS (
    ->     SELECT course_id
    ->     FROM takes s2
    ->     WHERE s2.ID = '10204'
    ->     AND s2.course_id NOT IN (SELECT s1.course_id)
    -> )
    -> LIMIT 5;
Empty set (0.20 sec)
```

```
mysql> SELECT course.course_id, course.title
    -> FROM course
    -> JOIN takes ON course.course_id = takes.course_id
    -> WHERE takes.ID = '12683'
    -> LIMIT 5;
+-----------+--------------------------+
| course_id | title                    |
+-----------+--------------------------+
| 105       | Image Processing         |
| 200       | The Music of the Ramones |
| 274       | Corporate Law            |
| 319       | World History            |
| 338       | Graph Theory             |
+-----------+--------------------------+
5 rows in set (0.00 sec)

mysql> SELECT student.name
    -> FROM student
    -> WHERE student.ID NOT IN (
    ->     SELECT DISTINCT takes.ID
    ->     FROM takes
    ->     JOIN course ON takes.course_id = course.course_id
    ->     WHERE course.dept_name = 'Elec. Eng.'
    -> )
    -> LIMIT 5;
+--------+
| name   |
+--------+
| Manber |
| Zelty  |
| Duan   |
| Colin  |
| Rzecz  |
+--------+
5 rows in set (0.00 sec)
```

```
mysql> SELECT DISTINCT student.name
    -> FROM student
    -> JOIN takes ON student.ID = takes.ID
    -> JOIN course ON takes.course_id = course.course_id
    -> WHERE course.dept_name = 'Comp. Sci.'
    -> LIMIT 5;
+-----------+
| name      |
+-----------+
| Colin     |
| Mediratta |
| Shabuno   |
| Jr        |
| Saito     |
+-----------+
5 rows in set (0.00 sec)

mysql> SELECT student.ID
    -> FROM student
    -> JOIN takes ON student.ID = takes.ID
    -> JOIN course ON takes.course_id = course.course_id
    -> WHERE course.dept_name = 'Comp. Sci.'
    -> GROUP BY student.ID
    -> HAVING COUNT(takes.course_id) <= 1
    -> LIMIT 5;
+-------+
| ID    |
+-------+
| 1018  |
| 10204 |
| 11422 |
| 11510 |
| 12563 |
+-------+
5 rows in set (0.00 sec)
```

```
mysql> SELECT student.name
    -> FROM student
    -> WHERE student.tot_cred > (
    ->     SELECT MIN(s.tot_cred)
    ->     FROM student s
    ->     WHERE s.dept_name = 'Comp. Sci.'
    -> )
    -> LIMIT 5;
+------------+
| name       |
+------------+
| Manber     |
| Zelty      |
| Duan       |
| Colin      |
| Mediratta  |
+------------+
5 rows in set (0.00 sec)
```

```
mysql> WITH dept_tot_cred AS (
    ->     SELECT dept_name, SUM(course.credits) AS total_credit
    ->     FROM course
    ->     GROUP BY dept_name
    -> )
    -> SELECT dept_name
    -> FROM dept_tot_cred
    -> WHERE total_credit = (
    ->     SELECT MAX(total_credit) FROM dept_tot_cred
    -> )
    -> LIMIT 1;
+-------------+
| dept_name   |
+-------------+
| Cybernetics |
+-------------+
1 row in set (0.00 sec)
```

```
mysql> CREATE TEMPORARY TABLE to_delete AS
    -> SELECT DISTINCT s.ID
    -> FROM student s
    -> JOIN takes t ON s.ID = t.ID
    -> JOIN course c ON t.course_id = c.course_id
    -> WHERE t.grade = 'F'
    -> AND c.dept_name = 'Comp. Sci.';
Query OK, 0 rows affected (0.01 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> DELETE FROM student
    -> WHERE ID IN (SELECT ID FROM to_delete);
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> CREATE TABLE Customer (
    ->     customer_id INT PRIMARY KEY,
    ->     name VARCHAR(100),
    ->     email VARCHAR(100),
    ->     address VARCHAR(255)
    -> );
Query OK, 0 rows affected (0.18 sec)

mysql> CREATE TABLE Category (
    ->     category_id INT PRIMARY KEY,
    ->     category_name VARCHAR(100)
    -> );
Query OK, 0 rows affected (0.01 sec)

mysql> CREATE TABLE Product (
    ->     product_id INT PRIMARY KEY,
    ->     name VARCHAR(100),
    ->     price DECIMAL(10, 2),
    ->     description TEXT,
    ->     category_id INT,
    ->     FOREIGN KEY (category_id) REFERENCES Category(category_id)
    -> );
Query OK, 0 rows affected (0.03 sec)

mysql> CREATE TABLE `Order` (
    ->     order_id INT PRIMARY KEY,
    ->     customer_id INT,
    ->     order_date DATE,
    ->     total_amount DECIMAL(10, 2),
    ->     FOREIGN KEY (customer_id) REFERENCES Customer(customer_id)
    -> );
Query OK, 0 rows affected (0.02 sec)
```

```
mysql> DESCRIBE Category;
+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
| category_id   | int         | NO   | PRI | NULL    |       |
| category_name | varchar(100) | YES  |     | NULL    |       |
+---------------+-------------+------+-----+---------+-------+
2 rows in set (0.04 sec)

mysql> DESCRIBE Product;
+-------------+--------------+------+-----+---------+-------+
| Field       | Type         | Null | Key | Default | Extra |
+-------------+--------------+------+-----+---------+-------+
| product_id  | int          | NO   | PRI | NULL    |       |
| name        | varchar(100) | YES  |     | NULL    |       |
| price       | decimal(10,2) | YES  |     | NULL    |       |
| description | text         | YES  |     | NULL    |       |
| category_id | int          | YES  | MUL | NULL    |       |
+-------------+--------------+------+-----+---------+-------+
5 rows in set (0.00 sec)

mysql> DESCRIBE `Order`;
+--------------+--------------+------+-----+---------+-------+
| Field        | Type         | Null | Key | Default | Extra |
+--------------+--------------+------+-----+---------+-------+
| order_id     | int          | NO   | PRI | NULL    |       |
| customer_id  | int          | YES  | MUL | NULL    |       |
| order_date   | date         | YES  |     | NULL    |       |
| total_amount | decimal(10,2) | YES  |     | NULL    |       |
+--------------+--------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

```
mysql> INSERT INTO Customer (customer_id, name, email, address)
    -> VALUES (1, 'John Doe', 'john@example.com', '123 Main St'),
    ->        (2, 'Jane Smith', 'jane@example.com', '456 Oak Ave');
Query OK, 2 rows affected (0.01 sec)
Records: 2  Duplicates: 0  Warnings: 0

mysql> INSERT INTO Category (category_id, category_name)
    -> VALUES (1, 'Electronics'),
    ->        (2, 'Clothing');
Query OK, 2 rows affected (0.01 sec)
Records: 2  Duplicates: 0  Warnings: 0

mysql> INSERT INTO Product (product_id, name, price, description, category_id)
    -> VALUES (1, 'Smartphone', 699.99, 'Latest smartphone model', 1),
    ->        (2, 'T-shirt', 19.99, 'Cotton t-shirt', 2);
Query OK, 2 rows affected (0.00 sec)
Records: 2  Duplicates: 0  Warnings: 0

mysql> INSERT INTO `Order` (order_id, customer_id, order_date, total_amount)
    -> VALUES (101, 1, '2024-10-22', 719.98),
    ->        (102, 2, '2024-10-23', 19.99);
Query OK, 2 rows affected (0.01 sec)
Records: 2  Duplicates: 0  Warnings: 0
```

```
mysql> SELECT * FROM Product;
+------------+------------+--------+------------------------+-------------+
| product_id | name       | price  | description            | category_id |
+------------+------------+--------+------------------------+-------------+
|          1 | Smartphone | 699.99 | Latest smartphone model |           1 |
|          2 | T-shirt    |  19.99 | Cotton t-shirt         |           2 |
+------------+------------+--------+------------------------+-------------+
2 rows in set (0.00 sec)

mysql> SELECT * FROM `Order`;
+----------+-------------+------------+--------------+
| order_id | customer_id | order_date | total_amount |
+----------+-------------+------------+--------------+
|      101 |           1 | 2024-10-22 |       719.98 |
|      102 |           2 | 2024-10-23 |        19.99 |
+----------+-------------+------------+--------------+
2 rows in set (0.00 sec)
```

```
mysql> -- Retrieve the customer names and the corresponding order IDs.
mysql> SELECT Customer.name, `Order`.order_id
    -> FROM Customer
    -> JOIN `Order` ON Customer.customer_id = `Order`.customer_id;
+------------+----------+
| name       | order_id |
+------------+----------+
| John Doe   |      101 |
| Jane Smith |      102 |
+------------+----------+
2 rows in set (0.00 sec)

mysql> -- Retrieve the product names and their respective categories.
mysql> SELECT Product.name, Category.category_name
    -> FROM Product
    -> JOIN Category ON Product.category_id = Category.category_id;
+------------+---------------+
| name       | category_name |
+------------+---------------+
| Smartphone | Electronics   |
| T-shirt    | Clothing      |
+------------+---------------+
2 rows in set (0.00 sec)

mysql> -- Retrieve all distinct customer names and product names.
mysql> SELECT name FROM Customer
    -> UNION
    -> SELECT name FROM Product;
+------------+
| name       |
+------------+
| John Doe   |
| Jane Smith |
| Smartphone |
| T-shirt    |
+------------+
4 rows in set (0.00 sec)

mysql> -- Retrieve names that are both customer names and product names.
mysql> SELECT name FROM Customer
    -> INTERSECT
    -> SELECT name FROM Product;
Empty set (0.00 sec)
```

```
mysql> -- Retrieve the total amount of orders placed by each customer.
mysql> SELECT Customer.name, SUM(`Order`.total_amount) AS total_spent
    -> FROM Customer
    -> JOIN `Order` ON Customer.customer_id = `Order`.customer_id
    -> GROUP BY Customer.name;
+------------+-------------+
| name       | total_spent |
+------------+-------------+
| John Doe   |      719.98 |
| Jane Smith |       19.99 |
+-----------+-------------+
2 rows in set (0.01 sec)

mysql> -- Count the number of products in each category.
mysql> SELECT Category.category_name, COUNT(Product.product_id) AS num_products
    -> FROM Product
    -> JOIN Category ON Product.category_id = Category.category_id
    -> GROUP BY Category.category_name;
+--------------+--------------+
| category_name | num_products |
+--------------+--------------+
| Electronics   |            1 |
| Clothing      |            1 |
+--------------+--------------+
2 rows in set (0.00 sec)

mysql> -- Retrieve all orders placed by the customer whose name is 'John Doe'.
mysql> SELECT `Order`.order_id, `Order`.total_amount
    -> FROM `Order`
    -> WHERE `Order`.customer_id = (
    ->     SELECT customer_id FROM Customer WHERE name = 'John Doe'
    -> );
+----------+--------------+
| order_id | total_amount |
+----------+--------------+
|      101 |       719.98 |
+----------+--------------+
1 row in set (0.01 sec)

mysql> -- Retrieve all products that belong to the 'Electronics' category.
mysql> SELECT name
    -> FROM Product
    -> WHERE category_id IN (
    ->     SELECT category_id FROM Category WHERE category_name = 'Electronics'
    -> );
+------------+
| name       |
+------------+
| Smartphone |
+------------+
1 row in set (0.00 sec)
```

```
mysql> -- Retrieve customer names who have placed at least one order.
mysql> SELECT name
    -> FROM Customer
    -> WHERE EXISTS (
    ->     SELECT 1 FROM `Order` WHERE Customer.customer_id = `Order`.customer_id
    -> );
+------------+
| name       |
+------------+
| John Doe   |
| Jane Smith |
+------------+
2 rows in set (0.00 sec)

mysql> -- Retrieve category names that have at least one product.
mysql> SELECT category_name
    -> FROM Category
    -> WHERE EXISTS (
    ->     SELECT 1 FROM Product WHERE Category.category_id = Product.category_id
    -> );
+---------------+
| category_name |
+---------------+
| Electronics   |
| Clothing      |
+---------------+
2 rows in set (0.00 sec)
```

```
mysql> -- Update John Doe's email to 'john.doe@newemail.com'.
mysql> UPDATE Customer
    -> SET email = 'john.doe@newemail.com'
    -> WHERE name = 'John Doe';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> -- Increase the price of all products in the 'Clothing' category by 10%.
mysql> UPDATE Product
    -> SET price = price * 1.10
    -> WHERE category_id = (SELECT category_id FROM Category WHERE category_name = 'Clot
hing');
Query OK, 1 row affected, 1 warning (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 1

mysql> -- Retrieve all orders placed on '2024-10-22'.
mysql> SELECT * FROM `Order` WHERE order_date = '2024-10-22';
+----------+-------------+------------+--------------+
| order_id | customer_id | order_date | total_amount |
+----------+-------------+------------+--------------+
|      101 |           1 | 2024-10-22 |       719.98 |
+----------+-------------+------------+--------------+
1 row in set (0.00 sec)

mysql> -- Retrieve product names and prices for products priced over $100.
mysql> SELECT name, price FROM Product WHERE price > 100;
+------------+--------+
| name       | price  |
+------------+--------+
| Smartphone | 699.99 |
+------------+--------+
1 row in set (0.00 sec)
```

```
mysql> -- Insert a new product 'Laptop' into the 'Electronics' category.
mysql> INSERT INTO Product (product_id, name, price, description, category_id)
    -> VALUES (3, 'Laptop', 999.99, 'High-performance laptop', 1);
Query OK, 1 row affected (0.00 sec)

mysql> -- Retrieve the total number of orders in the 'Order' table.
mysql> SELECT COUNT(order_id) FROM `Order`;
+-----------------+
| COUNT(order_id) |
+-----------------+
|               2 |
+-----------------+
1 row in set (0.00 sec)
```

```
mysql> -- Retrieve all products with prices between $50 and $500.
mysql> SELECT name, price
    -> FROM Product
    -> WHERE price BETWEEN 50 AND 500;
Empty set (0.00 sec)

mysql> -- Retrieve the total number of products in each category.
mysql> SELECT Category.category_name, COUNT(Product.product_id) AS total_products
    -> FROM Product
    -> JOIN Category ON Product.category_id = Category.category_id
    -> GROUP BY Category.category_name;
+---------------+----------------+
| category_name | total_products |
+---------------+----------------+
| Electronics   |              2 |
| Clothing      |              1 |
+---------------+----------------+
2 rows in set (0.00 sec)

mysql> -- Apply a 5% discount to the total amount of orders placed on '2024-10-22'.
mysql> UPDATE `Order`
    -> SET total_amount = total_amount * 0.95
    -> WHERE order_date = '2024-10-22';
Query OK, 1 row affected, 1 warning (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 1

mysql> -- Delete all products in the 'Clothing' category that are priced above $50.
mysql> DELETE FROM Product
    -> WHERE price > 50
    -> AND category_id = (SELECT category_id FROM Category WHERE category_name = 'Clothi
ng');
Query OK, 0 rows affected (0.01 sec)

mysql> -- Retrieve the names of customers who have not placed any orders.
mysql> SELECT Customer.name
    -> FROM Customer
    -> LEFT JOIN `Order` ON Customer.customer_id = `Order`.customer_id
    -> WHERE `Order`.order_id IS NULL;
Empty set (0.00 sec)

mysql>
```

```
mysql> -- Retrieve customer names and their corresponding orders, including customers wi
thout any orders (LEFT OUTER JOIN).
mysql> SELECT Customer.name, `Order`.order_id
    -> FROM Customer
    -> LEFT OUTER JOIN `Order` ON Customer.customer_id = `Order`.customer_id;
+------------+----------+
| name       | order_id |
+------------+----------+
| John Doe   |      101 |
| Jane Smith |      102 |
+------------+----------+
2 rows in set (0.00 sec)
```

```
mysql> -- Create a view 'OrderSummary' to show the customer name, order ID, and total am
ount.
mysql> CREATE VIEW OrderSummary AS
    -> SELECT Customer.name, `Order`.order_id, `Order`.total_amount
    -> FROM Customer
    -> JOIN `Order` ON Customer.customer_id = `Order`.customer_id;
Query OK, 0 rows affected (0.01 sec)

mysql>
mysql> -- Select from the created view 'OrderSummary'.
mysql> SELECT * FROM OrderSummary;
+------------+----------+--------------+
| name       | order_id | total_amount |
+------------+----------+--------------+
| John Doe   |      101 |       683.98 |
| Jane Smith |      102 |        19.99 |
+------------+----------+--------------+
2 rows in set (0.00 sec)
```

```
mysql> -- Create a table 'DiscountedProduct' with a CHECK constraint that ensures the di
scount cannot exceed 50%.
mysql> CREATE TABLE DiscountedProduct (
    ->     product_id INT PRIMARY KEY,
    ->     discount DECIMAL(3,2) CHECK (discount <= 0.50)
    -> );
Query OK, 0 rows affected (0.01 sec)
```