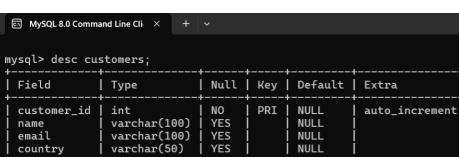
```
mysql> create database ecommerce;
Query OK, 1 row affected (0.07 sec)
mysql> use ecommerce;
Database changed
mysql> CREATE TABLE Customers (
          customer_id INT PRIMARY KEY AUTO_INCREMENT,
          name VARCHAR(100),
   ->
   ->
          email VARCHAR(100)
          country VARCHAR(50)
   ->
   -> ):
Query OK, 0 rows affected (0.10 sec)
mysql> CREATE TABLE Products (
   ->
          product_id INT PRIMARY KEY AUTO_INCREMENT,
   ->
          product_name VARCHAR(100),
   ->
          category VARCHAR(50),
   ->
          price DECIMAL(10,2)
   -> );
Query OK, 0 rows affected (0.05 sec)
nvsql>
mysql> CREATE TABLE Orders (
          order_id INT PRIMARY KEY AUTO_INCREMENT,
   ->
   ->
          customer_id INT,
   ->
          order_date DATE,
          total_amount DECIMAL(10,2),
   ->
   ->
          FOREIGN KEY (customer_id) REFERENCES Customers(customer_id)
   -> );
Query OK, 0 rows affected (0.06 sec)
nysql>
mysql> CREATE TABLE Orders (
          order_id INT PRIMARY KEY AUTO_INCREMENT,
   ->
   ->
          customer_id INT,
          order_date DATE,
   ->
          total_amount DECIMAL(10,2),
   ->
   ->
          FOREIGN KEY (customer_id) REFERENCES Customers(customer_id)
```



4 rows in set (0.08 sec)

mysql> desc products;

+	 Type	Null	Key	Default	Extra
product_id product_name category price	int varchar(100) varchar(50) decimal(10,2)	NO YES YES YES	PRI	NULL NULL NULL NULL	auto_increment

4 rows in set (0.00 sec)

mysql> desc orders;

L	L			L	L
Field	Туре	Null	Key	Default	Extra
customer_id order_date	int int date decimal(10,2)	YES YES			auto_increment

4 rows in set (0.00 sec)

mysql> desc orderdetails;

Field	Туре	Null	Key	Default	Extra
order_detail_id order_id product_id	int	NO YES YES	MUL	NULL NULL NULL	auto_increment auto_increment



mysql> select * from products;

product_id	product_name	category	price
2 3		Electronics Electronics Fashion Fashion	1200.00 700.00 80.00 150.00

4 rows in set (0.00 sec)

mysql> select * from customers;

customer_id		email	country
2	Bob	alice@email.com bob@email.com charlie@email.com	USA UK India

3 rows in set (0.00 sec)

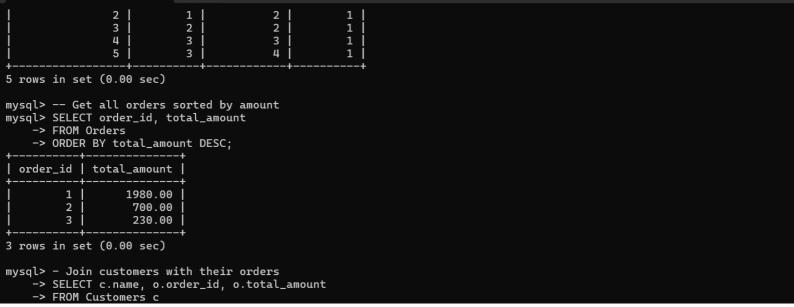
mysql> select * from orders;

order_id	customer_id	order_date	total_amount
			1000 00
1 2		2025-09-01 2025-09-02	1980.00 700.00
3	3	2025-09-03	230.00

3 rows in set (0.00 sec)

mysql> select * from orderdetails;

			
order_detail_id	order_id	product_id	quantity
1	1	1	1
2	1	2	1
3	2	2	1
4	3	3	1
l 5	3	4	1



lacktriangledown MySQL 8.0 Command Line Cli imes

```
mysql> -- Join customers with their orders
mysql> SELECT c.name, o.order_id, o.total_amount
-> FROM Customers c
    -> INNER JOIN Orders o ON c.customer_id = o.customer_id;
name
             order_id | total_amount |
  Alice
                               1980.00
  Bob
                                700.00
 Charlie
                     3
                                230.00
3 rows in set (0.01 sec)
mysql> -- Total revenue
mysql> SELECT SUM(total_amount) AS total_revenue
    -> FROM Orders;
| total_revenue
         2910.00
1 row in set (0.02 sec)
mysql> -- Average order value
mysql> SELECT AVG(total_amount) AS avg_order_value
    -> FROM Orders;
 avg_order_value
       970.000000
1 row in set (0.00 sec)
mysql> -- Revenue by country
mysql> SELECT c.country, SUM(o.total_amount) AS country_revenue
    -> FROM Customers c
    -> JOIN Orders o ON c.customer_id = o.customer_id
    -> GROUP BY c.country;
| country | country_revenue |
```

 \blacksquare MySQL 8.0 Command Line Cli imes + imes

```
-> JOIN Orders o ON c.customer_id = o.customer_id
-> GROUP BY c.country;
| country | country_revenue |
   USA
                                      1980.00
                                        700.00
230.00
    UK
   India
3 rows in set (0.00 sec)
mysql> -- Customers who spent more than average
mysql> SELECT name
-> FROM Customers
       -> WHERE customer_id IN (
-> SELECT customer_id
-> FROM Orders
-> GROUP BY customer_id
-> HAVING SUM(total_amount) > (SELECT AVG(total_amount) FROM Orders)
       -> );
| name
| Alice |
1 row in set (0.02 sec)
mysql> -- Create a view for order summary
mysql> CREATE VIEW OrderSummary AS
    -> SELECT o.order_id, c.name, o.order_date, o.total_amount
-> FROM Orders o
-> JOIN Customers c ON o.customer_id = c.customer_id;
Query OK, 0 rows affected (0.03 sec)
mysql>
mysql> -- Create index for faster joins
mysql> -- Create index for faster joins
mysql> CREATE INDEX idx_customer_id ON Orders(customer_id);
Query OK, 0 rows affected (0.13 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

MySQL 8.0 Command Line Cli × + ∨

```
-> JOIN Customers c ON o.customer_id = c.customer_id;
Query OK, 0 rows affected (0.03 sec)
mysql>
mysql> -- Create index for faster joins
mysql> CREATE INDEX idx_customer_id ON Orders(customer_id);
Query OK, 0 rows affected (0.13 sec)
Records: 0 Duplicates: 0 Warnings: 0
mysql> -- Show all customers, even if they have no orders mysql> SELECT c.name, o.order_id, o.total_amount
     -> FROM Customers c
     -> LEFT JOIN Orders o ON c.customer_id = o.customer_id;
              order_id |
                            total_amount
  name
  Alice
                                  1980.00
                       1
  Bob
                                   700.00
  Charlie
                       3
                                   230.00
3 rows in set (0.00 sec)
mysql> -- Show all orders, even if customer info is missing mysql> SELECT o.order_id, o.total_amount, c.name
     -> FROM Customers c
     -> RIGHT JOIN Orders o ON c.customer_id = o.customer_id;
  order_id
                total_amount
                                  name
                      1980.00
                                  Alice
           2
                       700.00
                                  Bob
           3
                       230.00
                                  Charlie
3 rows in set (0.01 sec)
```

MySQL 8.0 Command Line Cli ×
-> FROM Orders o

```
mysql> CREATE TABLE OrderDetails (

-> order_detail_id INT PRIMARY KEY AUTO_INCREMENT,
order_id INT,
-> product_id INT,
-> quantity INT,
-> FOREIGN KEY (order_id) REFERENCES Orders(order_id),
-> FOREIGN KEY (product_id) REFERENCES Products(product_id)
-> ;
Query OK, 0 rows affected (0.09 sec)

mysql> INSERT INTO Customers (name, email, country) VALUES
-> ('Ablice' 'alice@email.com', 'USA'),
-> ('Charlie', 'charlie@email.com', 'UNA'),
-> ('Charlie', 'charlie@email.com', 'India');
Query OK, 3 rows affected (0.03 sec)
Records: 3 Duplicates: 0 Warnings: 0

mysql> INSERT INTO Products (product_name, category, price) VALUES
-> ('Laptop', 'Electronics', '200.00),
-> ('Phone', 'Electronics', '700.00),
-> ('Natch', 'Fashion', 80.00),
-> ('Natch', 'Fashion', 80.00),
-> ('Watch', 'Fashion', 80.00),
-> ('Watch', 'Fashion', 80.00),
-> ('1, '2025-09-01', 1900.00),
-> (2, '2025-09-01', 1900.00),
-> (2, '2025-09-02', 700.00),
-> (2, '2025-09-02', 700.00),
-> (2, '2025-09-02', 700.00),
-> (2, '2025-09-02', 700.00),
-> (2, '2025-09-02', 700.00),
-> (2, '2025-09-02', 700.00),
-> (2, '2025-09-02', 700.00),
-> (2, '2025-09-02', 700.00),
-> (2, '2025-09-02', 700.00),
-> (2, '2025-09-02', 700.00),
-> (2, '2025-09-02', 700.00),
-> (2, '2025-09-02', 700.00),
-> (2, '2025-09-02', 700.00),
-> (2, '2025-09-02', 700.00),
-> (2, '2025-09-02', 700.00),
-> (2, '2025-09-02', 700.00),
-> (2, '2025-09-02', 700.00),
-> (2, '2025-09-02', 700.00),
-> (2, '2025-09-02', 700.00),
-> (2, '2025-09-02', 700.00),
-> (2, '2025-09-02', 700.00),
-> (2, '2025-09-02', 700.00),
-> (2, '2025-09-02', 700.00),
-> (2, '2025-09-02', 700.00),
-> (2, '2025-09-02', 700.00),
-> (2, '2025-09-02', 700.00),
-> (2, '2025-09-02', 700.00),
-> (2, '2025-09-02', 700.00),
-> (2, '2025-09-02', 700.00),
-> (3, '2025-09-02', 700.00),
-> (3, '2025-09-02', 700.00),
-> (3, '2025-09-02', 700.00),
-> (3, '2025-09-02', 700.00),
-> (3, '2025-09-02', 700.00),
-> (3, '2025-09-02', 700.00),
-> (3, '2025-09-02', 700.00),
-> (3, '2025-09-02', 700.00),
-> (3, '2025-09-02', 700.00),
-> (3, '2025
```

mysql>