

# SQL Task

---

1. Create the database

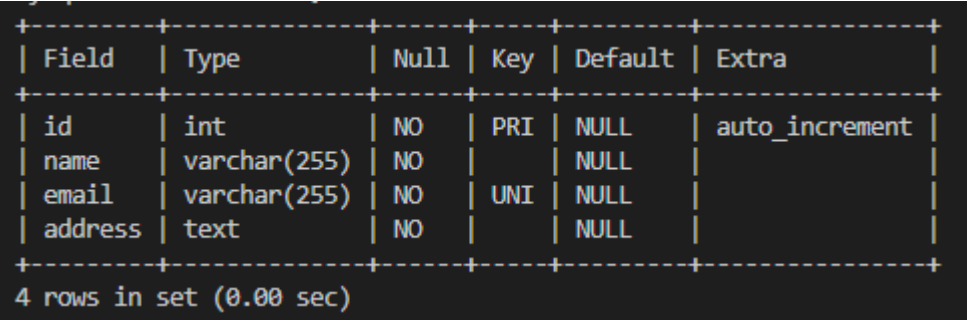
```
CREATE DATABASE ecommerce;
```

2. Use the database

```
USE ecommerce;
```

3. Create the customers table

```
CREATE TABLE customers (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  name VARCHAR(255) NOT NULL,  
  email VARCHAR(255) UNIQUE NOT NULL,  
  address TEXT NOT NULL  
);
```



Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
name	varchar(255)	NO		NULL	
email	varchar(255)	NO	UNI	NULL	
address	text	NO		NULL	

4 rows in set (0.00 sec)

4. Create the orders table

```
CREATE TABLE orders (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  customer_id INT NOT NULL,  
  order_date DATE NOT NULL,  
  total_amount DECIMAL(10, 2) NOT NULL,  
  FOREIGN KEY (customer_id) REFERENCES customers(id)  
);
```

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
customer_id	int	NO	MUL	NULL	
order_date	date	NO		NULL	
total_amount	decimal(10,2)	NO		NULL	

4 rows in set (0.00 sec)

5. Create the products table

```
CREATE TABLE products (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  name VARCHAR(255) NOT NULL,  
  price DECIMAL(10, 2) NOT NULL,  
  description TEXT  
);
```

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
name	varchar(255)	NO		NULL	
price	decimal(10,2)	NO		NULL	
description	text	YES		NULL	

4 rows in set (0.00 sec)

6. Insert sample data into the customers table

```
INSERT INTO customers (name, email, address)  
VALUES  
(  
'John Doe', 'john.doe@example.com', '123 Elm Street'),  
(  
'Jane Smith', 'jane.smith@example.com', '456 Oak Avenue'),  
(  
'Alice Johnson', 'alice.johnson@example.com', '789 Pine Road'),  
(  
'Bob Brown', 'bob.brown@example.com', '321 Maple Lane'),  
(  
'Carol White', 'carol.white@example.com', '654 Cedar Street'),  
(  
'David Black', 'david.black@example.com', '987 Birch Boulevard'),  
(  
'Emma Green', 'emma.green@example.com', '159 Spruce Drive'),  
(  
'Frank Gray', 'frank.gray@example.com', '753 Willow Way'),  
(  
'Grace Blue', 'grace.blue@example.com', '852 Aspen Court'),  
(  
'Hank Purple', 'hank.purple@example.com', '369 Sycamore Square');
```

name	email	address
John Doe	john.doe@example.com	123 Elm Street'
Jane Smith	jane.smith@example.com	456 Oak Avenue

name	email	address
Alice Johnson	alice.johnson@example.com	789 Pine Road
Bob Brown	bob.brown@example.com	321 Maple Lane
Carol White	carol.white@example.com	654 Cedar Street
David Black	david.black@example.com	987 Birch Boulevard
Emma Green	emma.green@example.com	159 Spruce Drive
Frank Gray	frank.gray@example.com'	753 Willow Way
Grace Blue	grace.blue@example.com	852 Aspen Court
Hank Purple	hank.purple@example.com	369 Sycamore Square

7. Insert sample data into the products table

```
INSERT INTO products (name, price, description)
VALUES
('Product A', 25.00, 'Description for Product A'),
('Product B', 30.00, 'Description for Product B'),
('Product C', 40.00, 'Description for Product C'),
('Product D', 20.00, 'Description for Product D'),
('Product E', 50.00, 'Description for Product E'),
('Product F', 15.00, 'Description for Product F'),
('Product G', 60.00, 'Description for Product G'),
('Product H', 35.00, 'Description for Product H'),
('Product I', 45.00, 'Description for Product I'),
('Product J', 55.00, 'Description for Product J');
```

name	price	description
Product A	25.00	Description for Product A
Product B	30.00	Description for Product B
Product C	40.00	Description for Product C
Product D	20.00	Description for Product D
Product E	50.00	Description for Product E
Product F	15.00	Description for Product F
Product G	60.00	Description for Product G
Product H	35.00	Description for Product H
Product I	45.00	Description for Product I
Product J	55.00	Description for Product J

8. Insert sample data into the orders table

```
INSERT INTO orders (customer_id, order_date, total_amount)
VALUES
(1, '2024-12-01', 50.00),
(1, '2024-12-15', 150.00),
(2, '2024-12-10', 75.00),
(3, '2024-12-20', 100.00),
(4, '2024-12-05', 120.00),
(5, '2024-11-15', 60.00),
(5, '2024-12-15', 120.00),
(6, '2024-12-18', 80.00),
(6, '2024-12-28', 50.00),
(7, '2024-7-19', 150.00),
(8, '2024-8-22', 200.00),
(8, '2024-1-22', 200.00),
(9, '2024-12-24', 175.00),
(10, '2024-12-15', 90.00);
(10, '2024-12-25', 180.00);
```

customer_id	order_date	total_amount
1	2024-12-01	50.00
1	2024-12-15	150.00
2	2024-12-10	75.00
3	2024-12-20	100.00
4	2024-12-05	120.00
5	2024-11-15	60.00
5	2024-12-15	120.00
6	2024-12-18	80.00
6	2024-12-28	50.00
7	2024-7-19	150.00
8	2024-8-22	200.00
8	2024-1-22	200.00
9	2024-12-24	175.00
10	2024-12-15	90.00
10	2024-12-25	180.00

Queries

- 1. Retrieve all customers who have placed an order in the last 30 days

```
SELECT DISTINCT c.*
FROM customers c
JOIN orders o ON c.id = o.customer_id
WHERE o.order_date >= CURDATE() - INTERVAL 30 DAY;
```

```
mysql> select * from orders;
+-----+-----+-----+-----+
| id | customer_id | order_date | total_amount |
+-----+-----+-----+-----+
| 1 | 1 | 2024-12-01 | 50.00 |
| 2 | 1 | 2024-12-15 | 150.00 |
| 3 | 2 | 2024-12-10 | 75.00 |
| 4 | 3 | 2024-12-20 | 100.00 |
| 5 | 4 | 2024-12-05 | 120.00 |
| 6 | 5 | 2024-11-15 | 60.00 |
| 7 | 5 | 2024-12-15 | 120.00 |
| 8 | 6 | 2024-12-18 | 80.00 |
| 9 | 6 | 2024-12-28 | 50.00 |
| 10 | 7 | 2024-07-19 | 150.00 |
| 11 | 8 | 2024-08-22 | 200.00 |
| 12 | 8 | 2024-01-22 | 200.00 |
| 13 | 9 | 2024-12-24 | 175.00 |
| 14 | 10 | 2024-12-15 | 90.00 |
+-----+-----+-----+-----+
14 rows in set (0.00 sec)

mysql> SELECT DISTINCT c.*
-> FROM customers c
-> JOIN orders o ON c.id = o.customer_id
-> WHERE o.order_date >= CURDATE() - INTERVAL 30 DAY;
+-----+-----+-----+-----+
| id | name | email | address |
+-----+-----+-----+-----+
| 1 | John Doe | john.doe@example.com | 123 Elm Street |
| 2 | Jane Smith | jane.smith@example.com | 456 Oak Avenue |
| 3 | Alice Johnson | alice.johnson@example.com | 789 Pine Road |
| 4 | Bob Brown | bob.brown@example.com | 321 Maple Lane |
| 5 | Carol White | carol.white@example.com | 654 Cedar Street |
| 6 | David Black | david.black@example.com | 987 Birch Boulevard |
| 9 | Grace Blue | grace.blue@example.com | 852 Aspen Court |
| 10 | Hank Purple | hank.purple@example.com | 369 Sycamore Square |
+-----+-----+-----+-----+
8 rows in set (0.00 sec)
```

2. Get the total amount of all orders placed by each customer

```
SELECT c.name, SUM(o.total_amount) AS total_spent
FROM customers c
JOIN orders o ON c.id = o.customer_id
GROUP BY c.id;
```

```
mysql> SELECT c.name, SUM(o.total_amount) AS total_spent
-> FROM customers c
-> JOIN orders o ON c.id = o.customer_id
-> GROUP BY c.id;
```

name	total_spent
John Doe	200.00
Jane Smith	75.00
Alice Johnson	100.00
Bob Brown	120.00
Carol White	180.00
David Black	130.00
Emma Green	150.00
Frank Gray	400.00
Grace Blue	175.00
Hank Purple	90.00

10 rows in set (0.00 sec)

3. Update the price of Product C to 45.00

```
UPDATE products
SET price = 45.00
WHERE name = 'Product C';
```

```
mysql> select * from products;
+----+-----+-----+-----+
| id | name      | price | description                |
+----+-----+-----+-----+
| 1  | Product A | 25.00 | Description for Product A |
| 2  | Product B | 30.00 | Description for Product B |
| 3  | Product C | 40.00 | Description for Product C |
| 4  | Product D | 20.00 | Description for Product D |
| 5  | Product E | 50.00 | Description for Product E |
| 6  | Product F | 15.00 | Description for Product F |
| 7  | Product G | 60.00 | Description for Product G |
| 8  | Product H | 35.00 | Description for Product H |
| 9  | Product I | 45.00 | Description for Product I |
| 10 | Product J | 55.00 | Description for Product J |
+----+-----+-----+-----+
10 rows in set (0.00 sec)

mysql> UPDATE products
  -> SET price = 45.00
  -> WHERE name = 'Product C';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from products;
+----+-----+-----+-----+
| id | name      | price | description                |
+----+-----+-----+-----+
| 1  | Product A | 25.00 | Description for Product A |
| 2  | Product B | 30.00 | Description for Product B |
| 3  | Product C | 45.00 | Description for Product C |
| 4  | Product D | 20.00 | Description for Product D |
| 5  | Product E | 50.00 | Description for Product E |
| 6  | Product F | 15.00 | Description for Product F |
| 7  | Product G | 60.00 | Description for Product G |
| 8  | Product H | 35.00 | Description for Product H |
| 9  | Product I | 45.00 | Description for Product I |
| 10 | Product J | 55.00 | Description for Product J |
+----+-----+-----+-----+
10 rows in set (0.00 sec)
```

4. Add a new column discount to the products table

```
ALTER TABLE products
ADD COLUMN discount DECIMAL(10, 2) DEFAULT 0.00;
```

```
mysql> ALTER TABLE products
-> ADD COLUMN discount DECIMAL(10, 2) DEFAULT 0.00;
Query OK, 0 rows affected (0.03 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> select * from products;
```

id	name	price	description	discount
1	Product A	25.00	Description for Product A	0.00
2	Product B	30.00	Description for Product B	0.00
3	Product C	45.00	Description for Product C	0.00
4	Product D	20.00	Description for Product D	0.00
5	Product E	50.00	Description for Product E	0.00
6	Product F	15.00	Description for Product F	0.00
7	Product G	60.00	Description for Product G	0.00
8	Product H	35.00	Description for Product H	0.00
9	Product I	45.00	Description for Product I	0.00
10	Product J	55.00	Description for Product J	0.00

10 rows in set (0.00 sec)

5. Retrieve the top 3 products with the highest price

```
SELECT *
FROM products
ORDER BY price DESC
LIMIT 3;
```

```
mysql> SELECT *
-> FROM products
-> ORDER BY price DESC
-> LIMIT 3;
```

id	name	price	description	discount
7	Product G	60.00	Description for Product G	0.00
10	Product J	55.00	Description for Product J	0.00
5	Product E	50.00	Description for Product E	0.00

3 rows in set (0.00 sec)

6. Get the names of customers who have ordered Product A

```
SELECT DISTINCT c.name
FROM customers c
JOIN orders o ON c.id = o.customer_id
JOIN order_items oi ON o.id = oi.order_id
JOIN products p ON oi.product_id = p.id
WHERE p.name = 'Product A';
```



```
mysql> SELECT DISTINCT c.name
-> FROM customers c
-> JOIN orders o ON c.id = o.customer_id
-> JOIN order_items oi ON o.id = oi.order_id
-> JOIN products p ON oi.product_id = p.id
-> WHERE p.name = 'Product A';
+-----+
| name |
+-----+
| John Doe |
| Emma Green |
+-----+
2 rows in set (0.00 sec)
```

7. Join the orders and customers tables to retrieve the customer's name and order date for each order

```
SELECT c.name, o.order_date
FROM customers c
JOIN orders o ON c.id = o.customer_id;
```

```
mysql> SELECT c.name, o.order_date
-> FROM customers c
-> JOIN orders o ON c.id = o.customer_id;
+-----+-----+
| name | order_date |
+-----+-----+
| John Doe | 2024-12-01 |
| John Doe | 2024-12-15 |
| Jane Smith | 2024-12-10 |
| Alice Johnson | 2024-12-20 |
| Bob Brown | 2024-12-05 |
| Carol White | 2024-11-15 |
| Carol White | 2024-12-15 |
| David Black | 2024-12-18 |
| David Black | 2024-12-28 |
| Emma Green | 2024-07-19 |
| Frank Gray | 2024-08-22 |
| Frank Gray | 2024-01-22 |
| Grace Blue | 2024-12-24 |
| Hank Purple | 2024-12-15 |
+-----+-----+
14 rows in set (0.00 sec)
```

8. Retrieve the orders with a total amount greater than 150.00

```
SELECT *
FROM orders
WHERE total_amount > 150.00;
```

```
mysql> SELECT *
-> FROM orders
-> WHERE total_amount > 150.00;
+-----+-----+-----+-----+
| id | customer_id | order_date | total_amount |
+-----+-----+-----+-----+
| 11 |          8 | 2024-08-22 |        200.00 |
| 12 |          8 | 2024-01-22 |        200.00 |
| 13 |          9 | 2024-12-24 |        175.00 |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

9. Normalize the database Create a separate table for order items

```
CREATE TABLE order_items (
  id INT AUTO_INCREMENT PRIMARY KEY,
  order_id INT NOT NULL,
  product_id INT NOT NULL,
  quantity INT NOT NULL,
  FOREIGN KEY (order_id) REFERENCES orders(id),
  FOREIGN KEY (product_id) REFERENCES products(id)
);
```

```
mysql> CREATE TABLE order_items (
-> id INT AUTO_INCREMENT PRIMARY KEY,
-> order_id INT NOT NULL,
-> product_id INT NOT NULL,
-> quantity INT NOT NULL,
-> FOREIGN KEY (order_id) REFERENCES orders(id),
-> FOREIGN KEY (product_id) REFERENCES products(id)
-> );
Query OK, 0 rows affected (0.08 sec)
```

```
mysql> desc order_items;
+-----+-----+-----+-----+-----+-----+
| Field      | Type | Null | Key | Default | Extra           |
+-----+-----+-----+-----+-----+-----+
| id         | int  | NO   | PRI | NULL    | auto_increment |
| order_id   | int  | NO   | MUL | NULL    |                 |
| product_id | int  | NO   | MUL | NULL    |                 |
| quantity   | int  | NO   |     | NULL    |                 |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.01 sec)
```

10. Retrieve the average total of all orders

```
SELECT AVG(total_amount) AS average_order_total
FROM orders;
```

```
mysql> SELECT AVG(total_amount) AS average_order_total  
-> FROM orders;  
+-----+  
| average_order_total |  
+-----+  
|          115.714286 |  
+-----+  
1 row in set (0.00 sec)
```