```
create database customers;
use customers;
CREATE TABLE customers (
   c_id INT PRIMARY KEY,
   c_name VARCHAR(50),
   city VARCHAR(50)
);

CREATE TABLE orders (
   o_id INT PRIMARY KEY,
   c_id INT,
   amount INT,
   order_date DATE,
   FOREIGN KEY (c_id) REFERENCES customers(c_id)
);

INSERT INTO customers (c_id, c_name, city) VALUES
(1, 'Rajesh', 'Hyderabad'),
(2, 'Anitha', 'Chennai'),
(3, 'Kiran', 'Bangalore'),
(4, 'Priya', 'Mumbai');

INSERT INTO orders (o_id, c_id, amount, order_date) VALUES
(101, 1, 2000, '2024-05-01'),
(102, 2, 3500, '2024-05-02'),
```

(103, 1, 1500, '2024-05-03'),

(104, 3, 4000, '2024-05-04'),

(105, 4, 2500, '2024-05-05');

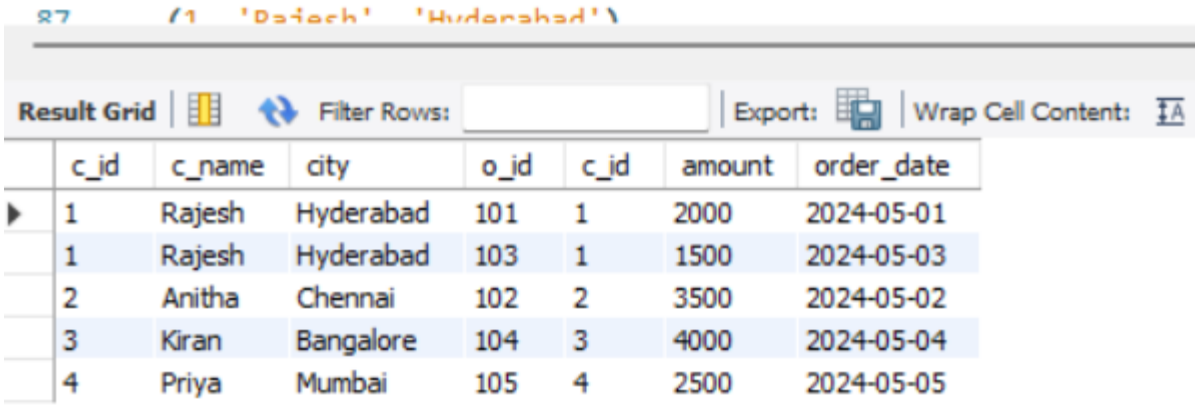
1.List all customers along with the details of their orders.

SELECT *FROM customers

INNER JOIN orders

ON customers.c_id = orders.c_id;

OUTPUT:



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

c_id	c_name	city	o_id	c_id	amount	order_date
1	Rajesh	Hyderabad	101	1	2000	2024-05-01
1	Rajesh	Hyderabad	103	1	1500	2024-05-03
2	Anitha	Chennai	102	2	3500	2024-05-02
3	Kiran	Bangalore	104	3	4000	2024-05-04
4	Priya	Mumbai	105	4	2500	2024-05-05

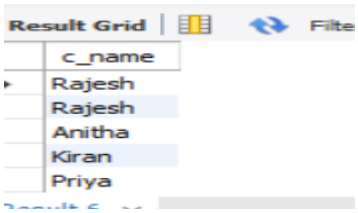
2. Show the names of customers who have placed at least one order.

SELECT c_name from customers

INNER JOIN orders

ON customers.c_id = orders.c_id;

OUTPUT:



Result Grid | Filte

c_name
Rajesh
Rajesh
Anitha
Kiran
Priya

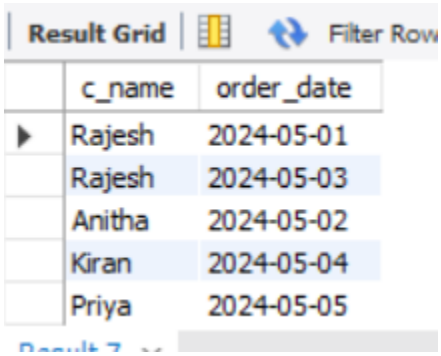
3. Display each customer's name and the order date of their purchase.

```
SELECT c_name,order_date from customers

INNER JOIN orders

ON customers.c_id = orders.c_id;
```

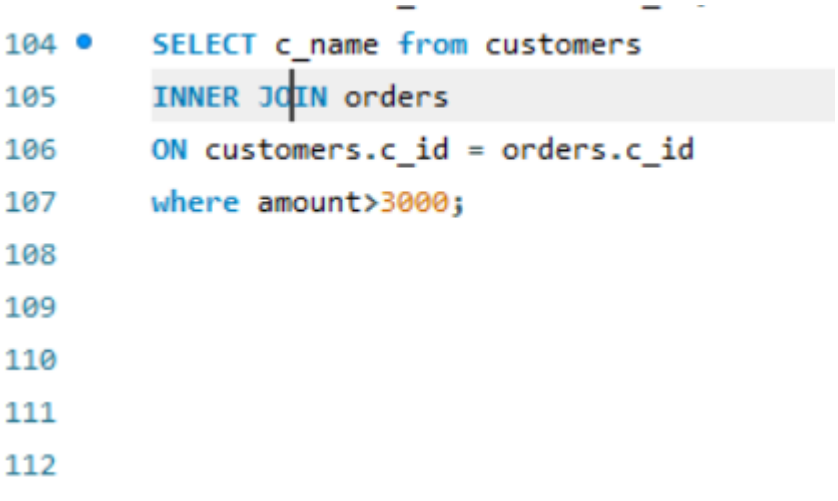
OUTPUT:



Result Grid | Filter Row

c_name	order_date
Rajesh	2024-05-01
Rajesh	2024-05-03
Anitha	2024-05-02
Kiran	2024-05-04
Priya	2024-05-05

4. Find customers who have placed orders with an amount greater than ₹3000



```
104 •   SELECT c_name from customers
105     INNER JOIN orders
106     ON customers.c_id = orders.c_id
107     where amount>3000;
108
109
110
111
112
```

Result Grid | Filter Rows: | Export:

c_name
Anitha
Kiran

5. Display the total amount spent by each customer.

```
107     where amount>3000;
108 •   SELECT c_name, SUM(orders.amount) AS total_amount
109     FROM customers
110     INNER JOIN orders
111     ON customers.c_id = orders.c_id
112     GROUP BY c_name;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

c_name	total_amount
Rajesh	3500
Anitha	3500
Kiran	4000
Priya	2500

6.List customer names along with their cities and order IDs.

```
3 •   SELECT c_name,city,o_id
4     FROM customers
5     INNER JOIN orders
6     ON customers.c_id = orders.c_id
7
8
9
0
1
2
```

sult Grid | Filter Rows:

c_name	city	o_id
Rajesh	Hyderabad	101
Rajesh	Hyderabad	103
Anitha	Chennai	102
Kiran	Bangalore	104
Priya	Mumbai	105

7.Show customers who placed more than one order.

```
117
118 •  SELECT c_name,count(o_id) from customers
119    INNER JOIN orders
120    ON customers.c_id = orders.c_id
121    group by c_name having count(o_id)>1;
122
123
124
125
```

Result Grid | Filter Rows: | Export: | Wrap

c_name	count(o_id)
Rajesh	2

8. Find all order IDs along with the customer names who placed them.

SELECT c_name,o_id from customers

INNER JOIN orders

ON customers.c_id = orders.c_id;

OUTPUT:

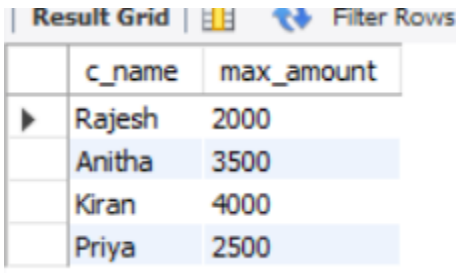
Result Grid | Fi

c_name	o_id
Rajesh	101
Rajesh	103
Anitha	102
Kiran	104
Priya	105

9. Display the names of customers and the highest order amount they made.

```
SELECT c_name,max(amount) as max_amount from customers
INNER JOIN orders
ON customers.c_id = orders.c_id
group by c_name;
```

OUTPUT:



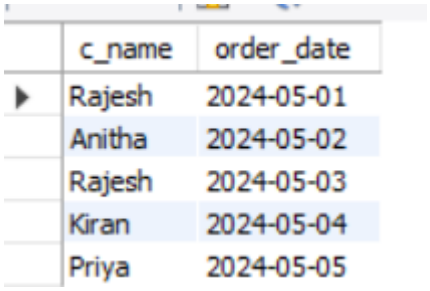
Result Grid | Filter Rows

c_name	max_amount
Rajesh	2000
Anitha	3500
Kiran	4000
Priya	2500

10. List all customers and their order details sorted by order date.

```
SELECT c_name,order_date from customers
INNER JOIN orders
ON customers.c_id = orders.c_id
order by order_date;
```

OUTPUT:



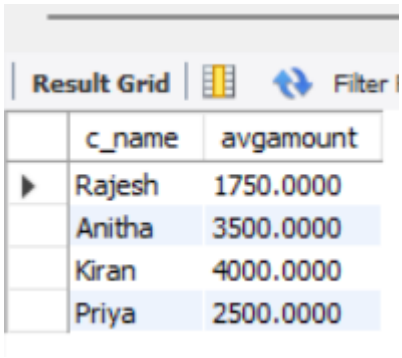
c_name	order_date
Rajesh	2024-05-01
Anitha	2024-05-02
Rajesh	2024-05-03
Kiran	2024-05-04
Priya	2024-05-05

11. Show customer names and the average amount of their orders.

```
SELECT c_name,avg(amount) as avgamount from customers
```

```
INNER JOIN orders
ON customers.c_id = orders.c_id
group by c_name;
```

OUTPUT:



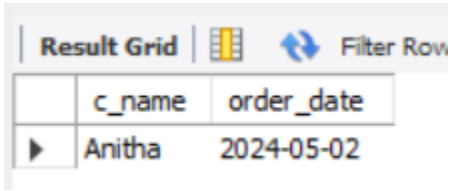
Result Grid | Filter

c_name	avgamount
Rajesh	1750.0000
Anitha	3500.0000
Kiran	4000.0000
Priya	2500.0000

12. Find customers who placed orders on a specific date (e.g., '2024-05-02').

```
SELECT c_name, order_date
FROM customers
INNER JOIN orders
ON customers.c_id = orders.c_id
WHERE order_date = '2024-05-02';
```

OUTPUT:



Result Grid | Filter Row

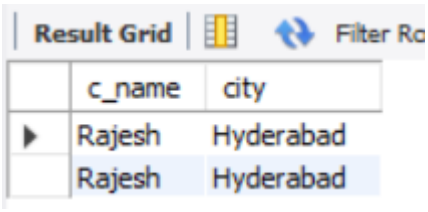
c_name	order_date
Anitha	2024-05-02

13. Display customers and order details for those living in ‘Hyderabad’.

```
SELECT c_name, city
FROM customers
```

```
INNER JOIN orders
ON customers.c_id = orders.c_id
WHERE city='Hyderabad';
```

OUTPUT:

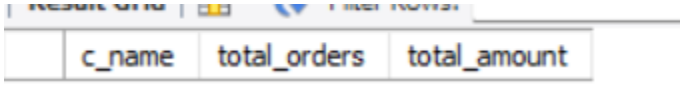


	c_name	city
▶	Rajesh	Hyderabad
	Rajesh	Hyderabad

14. Find customers who have ordered multiple times with total amount > ₹5000.

```
SELECT c_name,COUNT(o_id) AS total_orders,SUM(amount) AS total_amount
FROM customers
INNER JOIN orders
ON customers.c_id = orders.c_id
GROUP BY c_name having sum(amount)>5000;
```

OUTPUT:



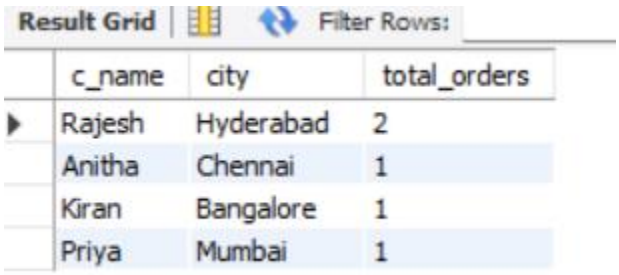
	c_name	total_orders	total_amount

15. Show each customer's name, city, and total number of orders placed.

```
SELECT c_name,city,COUNT(o_id) AS total_orders
FROM customers
LEFT JOIN orders
ON customers.c_id = orders.c_id
```

GROUP BY c_name,city;

OUTPUT:



Result Grid | Filter Rows:

c_name	city	total_orders
Rajesh	Hyderabad	2
Anitha	Chennai	1
Kiran	Bangalore	1
Priya	Mumbai	1