

```
1 -- DROP if they already exist (optional, helps reset)
2 DROP TABLE IF EXISTS orders;
3 DROP TABLE IF EXISTS customers;
4
5 -- Create customers table
6 CREATE TABLE customers (
7     customer_id INT PRIMARY KEY,
8     name VARCHAR(100),
9     email VARCHAR(100)
10 );
11
12 -- Create orders table
13 CREATE TABLE orders (
14     order_id INT PRIMARY KEY,
15     customer_id INT,
16     order_date DATE,
17     total_amount DECIMAL(10,2),
18     FOREIGN KEY (customer_id) REFERENCES customers(customer_id)
19 );
20
21 -- Insert data into customers
22 INSERT INTO customers (customer_id, name, email) VALUES
23 (1, 'Alice', 'alice@mail.com'),
24 (2, 'Bob', 'bob@mail.com'),
25 (3, 'Charlie', 'charlie@mail.com');
26
27 -- Insert data into orders
28 INSERT INTO orders (order_id, customer_id, order_date, total_amount) VALUES
29 (101, 1, '2024-01-05', 250.00),
30 (102, 2, '2024-01-07', 750.00),
31 (103, 3, '2024-01-10', 180.00),
32 (104, 1, '2024-01-15', 550.00);
33
34
35 SELECT * FROM customers;
36 SELECT * FROM orders;
37
38
39
40 SELECT AVG(total_amount) AS avg_order_value FROM orders;
41
42
43
44 -- INNER JOIN
45 SELECT c.name, o.total_amount
46 FROM customers c
47 JOIN orders o ON c.customer_id = o.customer_id;
```

STDIN

Input for the program (Optional)

Output:

```
+-----+-----+-----+
| customer_id | name   | email           |
+-----+-----+-----+
|          1 | Alice  | alice@mail.com  |
|          2 | Bob    | bob@mail.com    |
|          3 | Charlie| charlie@mail.com|
+-----+-----+-----+

+-----+-----+-----+-----+
| order_id | customer_id | order_date | total_amount |
+-----+-----+-----+-----+
|       101 |           1 | 2024-01-05 |       250.00 |
|       102 |           2 | 2024-01-07 |       750.00 |
|       103 |           3 | 2024-01-10 |       180.00 |
|       104 |           1 | 2024-01-15 |       550.00 |
+-----+-----+-----+-----+

+-----+
| avg_order_value |
+-----+
|       432.500000 |
+-----+

+-----+-----+
| name   | total_amount |
+-----+-----+
| Alice  |       250.00 |
| Alice  |       550.00 |
| Bob    |       750.00 |
| Charlie|       180.00 |
+-----+-----+

+-----+
| avg_order_value |
+-----+
|       432.500000 |
+-----+
```

```

38
39
40 SELECT AVG(total_amount) AS avg_order_value FROM orders;
41
42
43 -- INNER JOIN
44 SELECT c.name, o.total_amount
45 FROM customers c
46 JOIN orders o ON c.customer_id = o.customer_id;
47
48 -- AVG
49 SELECT AVG(total_amount) AS avg_order_value FROM orders;
50
51 ---- Subqueries
52 SELECT name FROM customers
53 WHERE customer_id IN (
54     SELECT customer_id FROM orders WHERE total_amount > 500
55 );
56
57
58
59 SELECT customer_id, SUM(total_amount) AS total_spent
60 FROM orders
61 GROUP BY customer_id
62 HAVING total_spent > 1000;
63
64
65
66 SELECT customers.name, orders.total_amount
67 FROM customers
68 INNER JOIN orders ON customers.customer_id = orders.customer_id;
69
70
71 CREATE VIEW high_value_orders AS
72 SELECT * FROM orders WHERE total_amount > 500;
73
74
75 SELECT * FROM high_value_orders;
76
77
78
79
80
81

```

```

|      432.500000 |
+-----+
+-----+-----+
| name | total_amount |
+-----+-----+
| Alice |      250.00 |
| Alice |      550.00 |
| Bob   |      750.00 |
| Charlie |     180.00 |
+-----+-----+
+-----+
| avg_order_value |
+-----+
|      432.500000 |
+-----+
+-----+
| name |
+-----+
| Alice |
| Bob   |
+-----+
+-----+-----+
| name | total_amount |
+-----+-----+
| Alice |      250.00 |
| Alice |      550.00 |
| Bob   |      750.00 |
| Charlie |     180.00 |
+-----+-----+
+-----+-----+-----+-----+
| order_id | customer_id | order_date | total_amount |
+-----+-----+-----+-----+
|      102 |           2 | 2024-01-07 |      750.00 |
|      104 |           1 | 2024-01-15 |      550.00 |
+-----+-----+-----+-----+

```