SQL PROJECT

E-commerce Sales Analysis

E-commerce Sales Analysis

SQL PROJECT

Objective:

- The primary objective of this project is to achieve sustainable business growth while addressing existing challenges.
- Objective is to analyzed key things, which overcomes current challenges and improve future growth
- Analyze customer purchasing behavior by tracking order history, total spending, and repeat purchases using SQL queries.
- Measure sales and revenue metrics such as total sales per customer, highest value orders, and average order value

Question Modes:

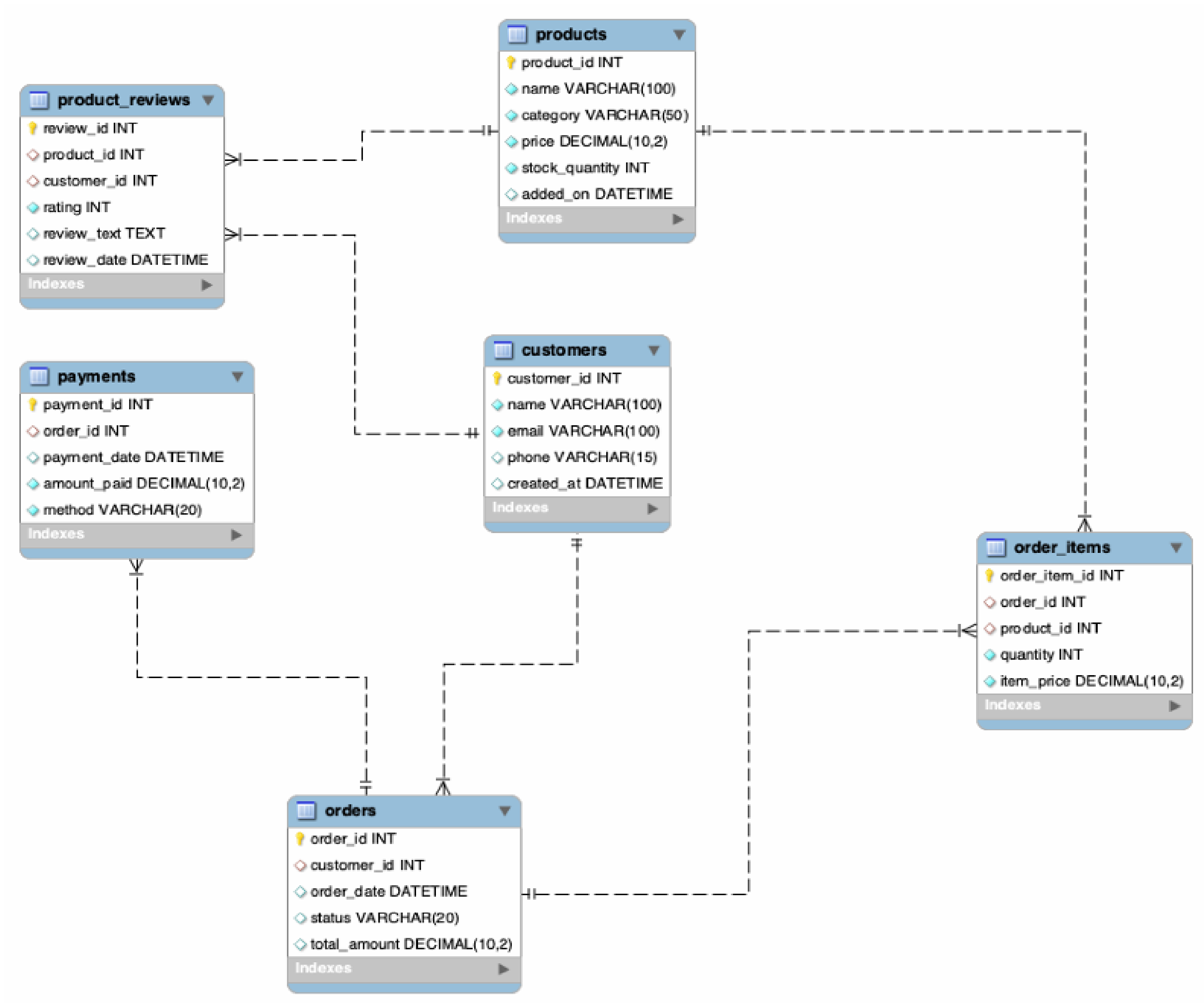
- Level 1:- Select, Distinct, Where, Order by, Logic Operator
- Level 2:- Column concatenation, IS NULL, IS NOT NULL, Formatting date
- Level 3:- Aggregate functions, Group by, HAVING Vs WHERE,
- Level 4:- Joins
- Level 5:- Subquery
- Level 6:- Union, AND, OR, INTERSECT(If Support)

E-commerce Sales Analysis

SQL PROJECT

E-Commerce DataBase:

ER Diagram:-

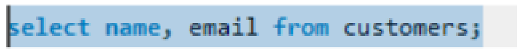


E-commerce Sales Analysis

Level 1: Basics

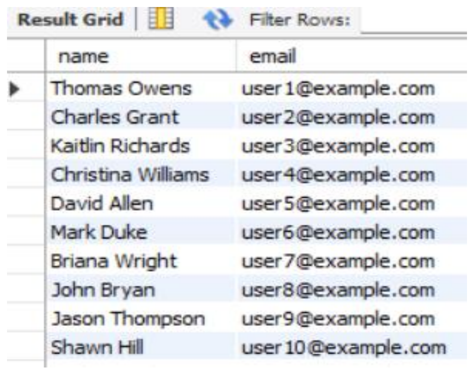
Q1. Retrieve customer names and emails for email marketing

INPUT



```
select name, email from customers;
```

OUTPUT

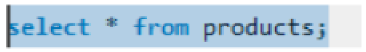


Result Grid | Filter Rows:

name	email
Thomas Owens	user1@example.com
Charles Grant	user2@example.com
Kaitlin Richards	user3@example.com
Christina Williams	user4@example.com
David Allen	user5@example.com
Mark Duke	user6@example.com
Briana Wright	user7@example.com
John Bryan	user8@example.com
Jason Thompson	user9@example.com
Shawn Hill	user10@example.com

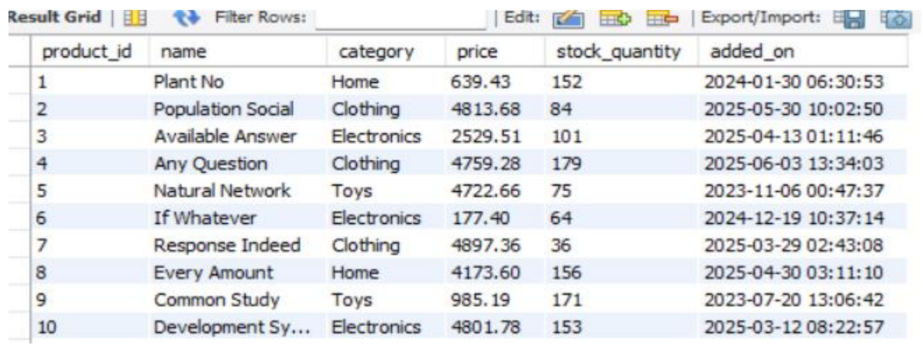
Q2. View complete product catalog with all available details

INPUT



```
select * from products;
```

OUTPUT



Result Grid | Filter Rows: | Edit: | Export/Import:

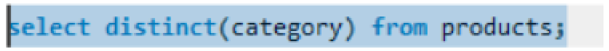
product_id	name	category	price	stock_quantity	added_on
1	Plant No	Home	639.43	152	2024-01-30 06:30:53
2	Population Social	Clothing	4813.68	84	2025-05-30 10:02:50
3	Available Answer	Electronics	2529.51	101	2025-04-13 01:11:46
4	Any Question	Clothing	4759.28	179	2025-06-03 13:34:03
5	Natural Network	Toys	4722.66	75	2023-11-06 00:47:37
6	If Whatever	Electronics	177.40	64	2024-12-19 10:37:14
7	Response Indeed	Clothing	4897.36	36	2025-03-29 02:43:08
8	Every Amount	Home	4173.60	156	2025-04-30 03:11:10
9	Common Study	Toys	985.19	171	2023-07-20 13:06:42
10	Development Sy...	Electronics	4801.78	153	2025-03-12 08:22:57

E-commerce Sales Analysis

Level 1: Basics

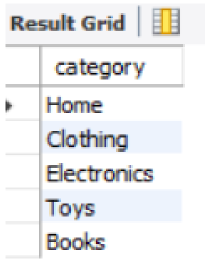
Q3. List all unique product categories

INPUT



```
select distinct(category) from products;
```

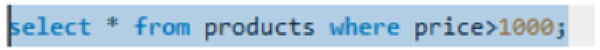
OUTPUT



category
Home
Clothing
Electronics
Toys
Books

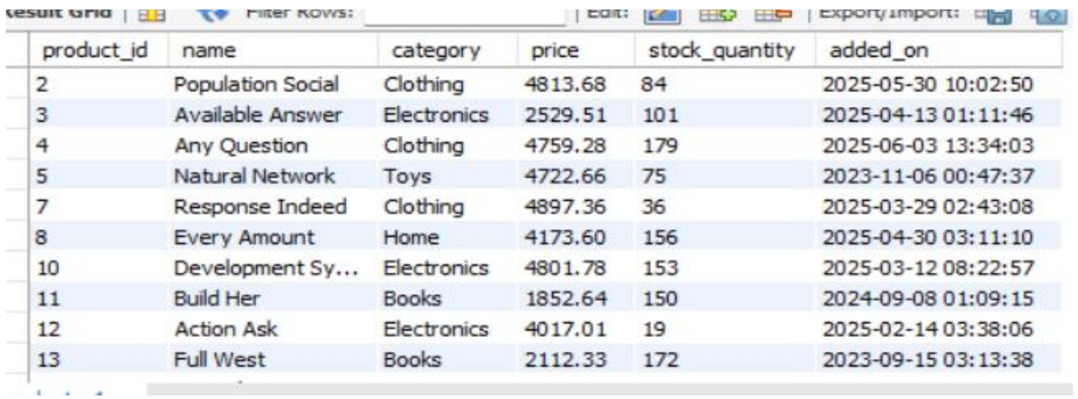
Q4. Show all products priced above ₹1,000

INPUT



```
select * from products where price>1000;
```

OUTPUT



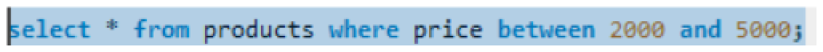
product_id	name	category	price	stock_quantity	added_on
2	Population Social	Clothing	4813.68	84	2025-05-30 10:02:50
3	Available Answer	Electronics	2529.51	101	2025-04-13 01:11:46
4	Any Question	Clothing	4759.28	179	2025-06-03 13:34:03
5	Natural Network	Toys	4722.66	75	2023-11-06 00:47:37
7	Response Indeed	Clothing	4897.36	36	2025-03-29 02:43:08
8	Every Amount	Home	4173.60	156	2025-04-30 03:11:10
10	Development Sy...	Electronics	4801.78	153	2025-03-12 08:22:57
11	Build Her	Books	1852.64	150	2024-09-08 01:09:15
12	Action Ask	Electronics	4017.01	19	2025-02-14 03:38:06
13	Full West	Books	2112.33	172	2023-09-15 03:13:38

E-commerce Sales Analysis

Level 1: Basics

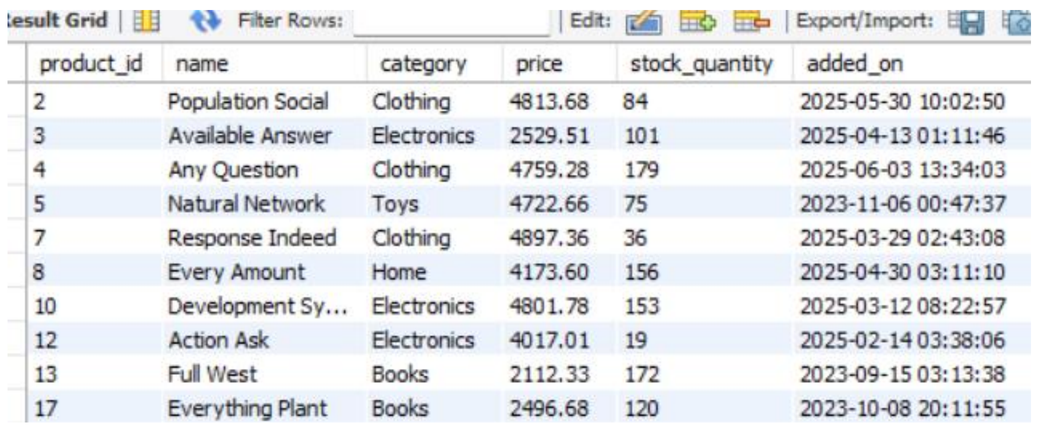
Q5. Display products within a mid-range price bracket(₹2,000 to₹5,000)

INPUT



```
select * from products where price between 2000 and 5000;
```

OUTPUT

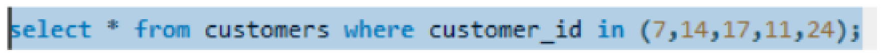


Result Grid | Filter Rows: | Edit: | Export/Import:

product_id	name	category	price	stock_quantity	added_on
2	Population Social	Clothing	4813.68	84	2025-05-30 10:02:50
3	Available Answer	Electronics	2529.51	101	2025-04-13 01:11:46
4	Any Question	Clothing	4759.28	179	2025-06-03 13:34:03
5	Natural Network	Toys	4722.66	75	2023-11-06 00:47:37
7	Response Indeed	Clothing	4897.36	36	2025-03-29 02:43:08
8	Every Amount	Home	4173.60	156	2025-04-30 03:11:10
10	Development Sy...	Electronics	4801.78	153	2025-03-12 08:22:57
12	Action Ask	Electronics	4017.01	19	2025-02-14 03:38:06
13	Full West	Books	2112.33	172	2023-09-15 03:13:38
17	Everything Plant	Books	2496.68	120	2023-10-08 20:11:55

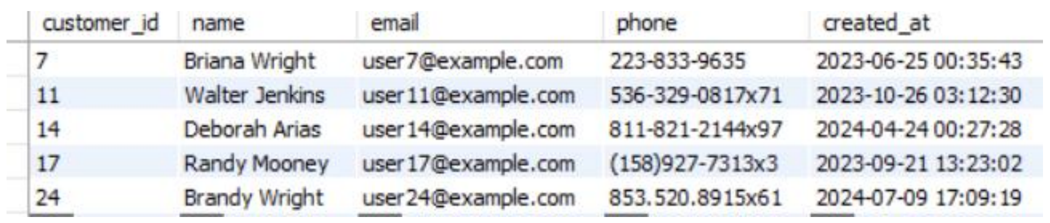
Q6. Fetch data for specific customer IDs (e.g., from loyalty program list)

INPUT



```
select * from customers where customer_id in (7,14,17,11,24);
```

OUTPUT



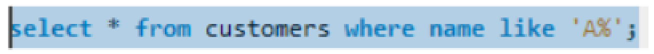
customer_id	name	email	phone	created_at
7	Briana Wright	user7@example.com	223-833-9635	2023-06-25 00:35:43
11	Walter Jenkins	user11@example.com	536-329-0817x71	2023-10-26 03:12:30
14	Deborah Arias	user14@example.com	811-821-2144x97	2024-04-24 00:27:28
17	Randy Mooney	user17@example.com	(158)927-7313x3	2023-09-21 13:23:02
24	Brandy Wright	user24@example.com	853.520.8915x61	2024-07-09 17:09:19

E-commerce Sales Analysis

Level 1: Basics

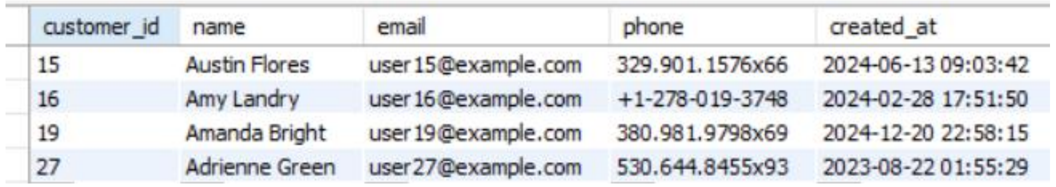
Q7. Identify customers whose names start with the letter 'A'

INPUT



```
select * from customers where name like 'A%';
```

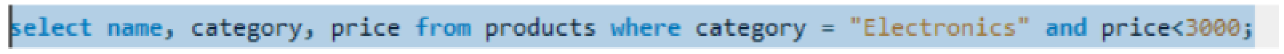
OUTPUT



customer_id	name	email	phone	created_at
15	Austin Flores	user15@example.com	329.901.1576x66	2024-06-13 09:03:42
16	Amy Landry	user16@example.com	+1-278-019-3748	2024-02-28 17:51:50
19	Amanda Bright	user19@example.com	380.981.9798x69	2024-12-20 22:58:15
27	Adrienne Green	user27@example.com	530.644.8455x93	2023-08-22 01:55:29

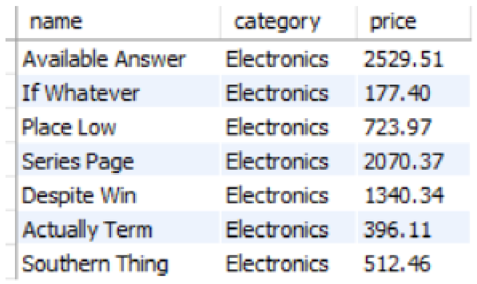
Q8. List electronics products priced under ₹3,000

INPUT



```
select name, category, price from products where category = "Electronics" and price<3000;
```

OUTPUT



name	category	price
Available Answer	Electronics	2529.51
If Whatever	Electronics	177.40
Place Low	Electronics	723.97
Series Page	Electronics	2070.37
Despite Win	Electronics	1340.34
Actually Term	Electronics	396.11
Southern Thing	Electronics	512.46

E-commerce Sales Analysis

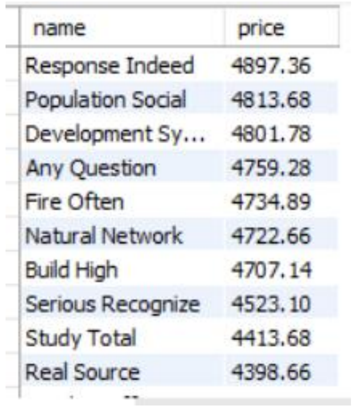
Level 1: Basics

Q9. Display product names and prices in descending order of price

INPUT

```
select name, price from products order by price desc;
```

OUTPUT



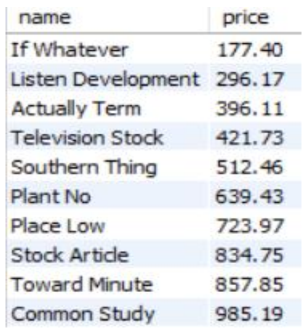
name	price
Response Indeed	4897.36
Population Social	4813.68
Development Sy...	4801.78
Any Question	4759.28
Fire Often	4734.89
Natural Network	4722.66
Build High	4707.14
Serious Recognize	4523.10
Study Total	4413.68
Real Source	4398.66

Q10. Display product names and prices, sorted by price and then by name

INPUT

```
select name, price from products order by price , name ;
```

OUTPUT



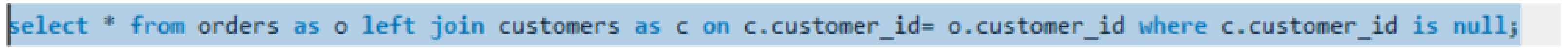
name	price
If Whatever	177.40
Listen Development	296.17
Actually Term	396.11
Television Stock	421.73
Southern Thing	512.46
Plant No	639.43
Place Low	723.97
Stock Article	834.75
Toward Minute	857.85
Common Study	985.19

E-commerce Sales Analysis

Level 2: Filtering and Formatting

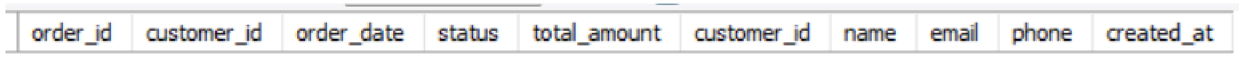
Q1. Retrieve orders where customer information is missing (possibly due to data migration or deletion)

INPUT



```
select * from orders as o left join customers as c on c.customer_id= o.customer_id where c.customer_id is null;
```

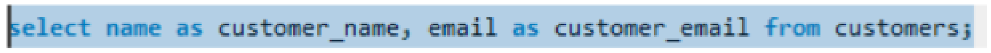
OUTPUT



order_id	customer_id	order_date	status	total_amount	customer_id	name	email	phone	created_at

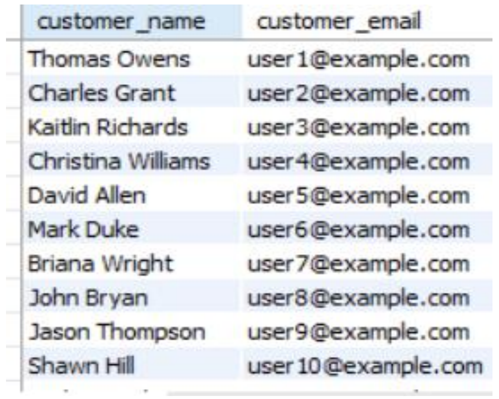
Q2. Display customer names and emails using column aliases for frontend readability

INPUT



```
select name as customer_name, email as customer_email from customers;
```

OUTPUT



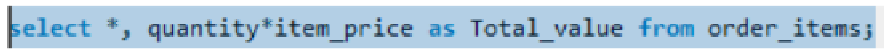
customer_name	customer_email
Thomas Owens	user1@example.com
Charles Grant	user2@example.com
Kaitlin Richards	user3@example.com
Christina Williams	user4@example.com
David Allen	user5@example.com
Mark Duke	user6@example.com
Briana Wright	user7@example.com
John Bryan	user8@example.com
Jason Thompson	user9@example.com
Shawn Hill	user10@example.com

E-commerce Sales Analysis

Level 2: Filtering and Formatting

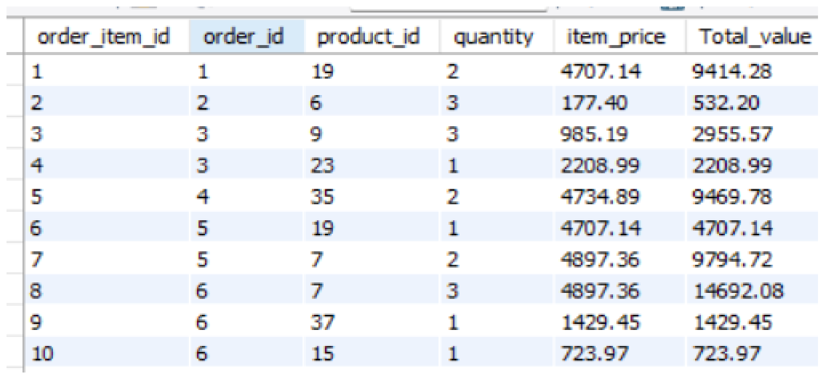
Q3. Calculate total value per item ordered by multiplying quantity and item price

INPUT



```
select *, quantity*item_price as Total_value from order_items;
```

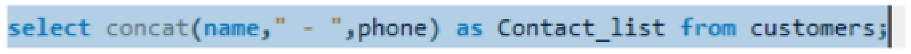
OUTPUT



order_item_id	order_id	product_id	quantity	item_price	Total_value
1	1	19	2	4707.14	9414.28
2	2	6	3	177.40	532.20
3	3	9	3	985.19	2955.57
4	3	23	1	2208.99	2208.99
5	4	35	2	4734.89	9469.78
6	5	19	1	4707.14	4707.14
7	5	7	2	4897.36	9794.72
8	6	7	3	4897.36	14692.08
9	6	37	1	1429.45	1429.45
10	6	15	1	723.97	723.97

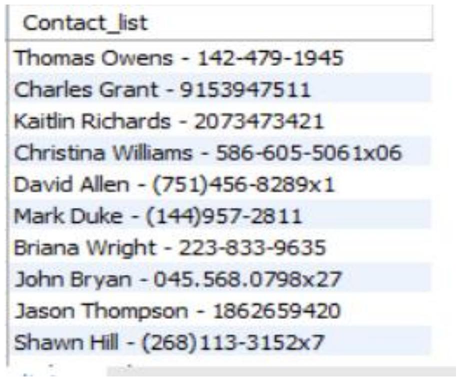
Q4. Combine customer name and phone number in a single column

INPUT



```
select concat(name," - ",phone) as Contact_list from customers;
```

OUTPUT



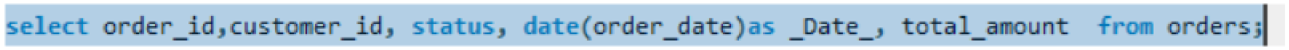
Contact_list
Thomas Owens - 142-479-1945
Charles Grant - 9153947511
Kaitlin Richards - 2073473421
Christina Williams - 586-605-5061x06
David Allen - (751)456-8289x1
Mark Duke - (144)957-2811
Briana Wright - 223-833-9635
John Bryan - 045.568.0798x27
Jason Thompson - 1862659420
Shawn Hill - (268)113-3152x7

E-commerce Sales Analysis

Level 2: Filtering and Formatting

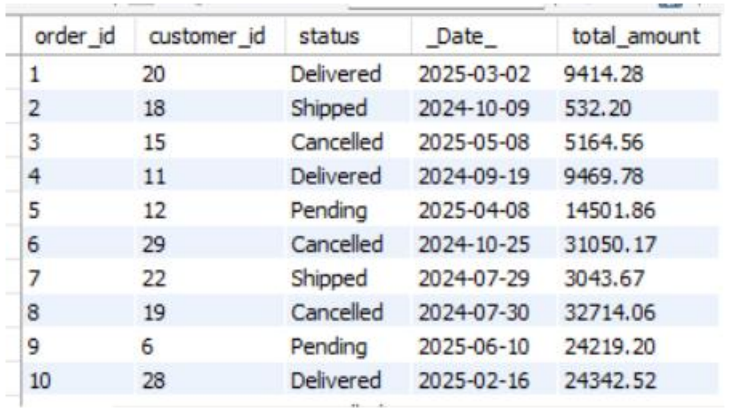
Q5. Extract only the date part from order timestamps for date-wise reporting

INPUT



```
select order_id,customer_id, status, date(order_date)as _Date_, total_amount  from orders;
```

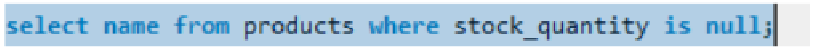
OUTPUT



order_id	customer_id	status	_Date_	total_amount
1	20	Delivered	2025-03-02	9414.28
2	18	Shipped	2024-10-09	532.20
3	15	Cancelled	2025-05-08	5164.56
4	11	Delivered	2024-09-19	9469.78
5	12	Pending	2025-04-08	14501.86
6	29	Cancelled	2024-10-25	31050.17
7	22	Shipped	2024-07-29	3043.67
8	19	Cancelled	2024-07-30	32714.06
9	6	Pending	2025-06-10	24219.20
10	28	Delivered	2025-02-16	24342.52

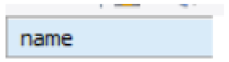
Q6. List products that do not have any stock left

INPUT



```
select name from products where stock_quantity is null;
```

OUTPUT



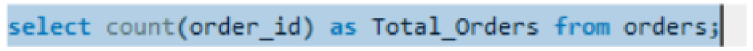
name

E-commerce Sales Analysis

Level 3: Aggregations

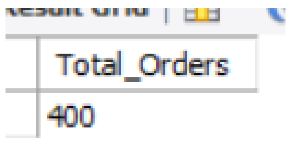
Q1. Count the total number of orders placed

INPUT



```
select count(order_id) as Total_Orders from orders;
```

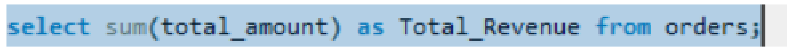
OUTPUT



```
Total_Orders
400
```

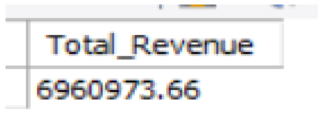
Q2. Calculate the total revenue collected from all orders

INPUT



```
select sum(total_amount) as Total_Revenue from orders;
```

OUTPUT



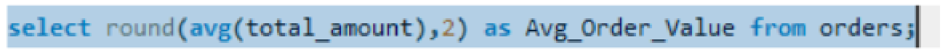
```
Total_Revenue
6960973.66
```

E-commerce Sales Analysis

Level 3: Aggregations

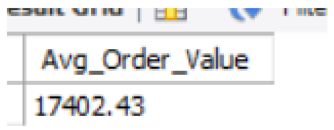
Q3. Calculate the average order value

INPUT



```
select round(avg(total_amount),2) as Avg_Order_Value from orders;
```

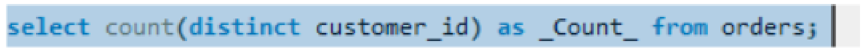
OUTPUT



```
Avg_Order_Value
17402.43
```

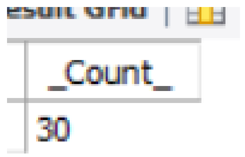
Q4. Count the number of customers who have placed at least one order

INPUT



```
select count(distinct customer_id) as _Count_ from orders;
```

OUTPUT



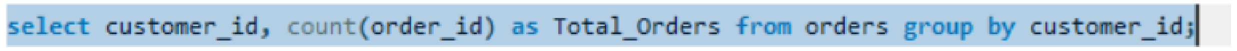
```
_Count_
30
```

E-commerce Sales Analysis

Level 3: Aggregations

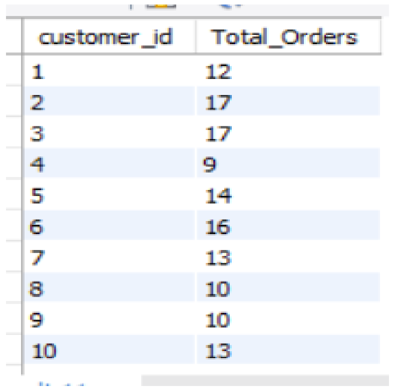
Q5. Find the number of orders placed by each customer

INPUT



```
select customer_id, count(order_id) as Total_Orders from orders group by customer_id;
```

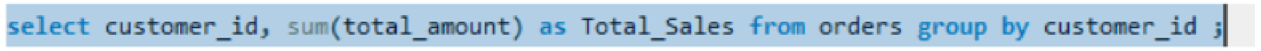
OUTPUT



customer_id	Total_Orders
1	12
2	17
3	17
4	9
5	14
6	16
7	13
8	10
9	10
10	13

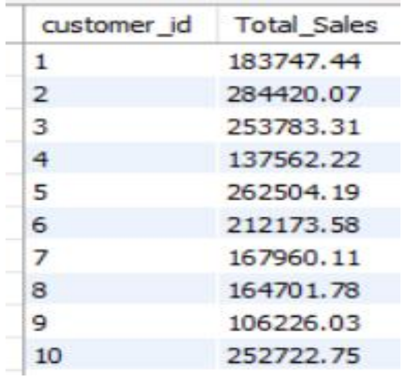
Q6. Find total sales amount made by each customer

INPUT



```
select customer_id, sum(total_amount) as Total_Sales from orders group by customer_id ;
```

OUTPUT



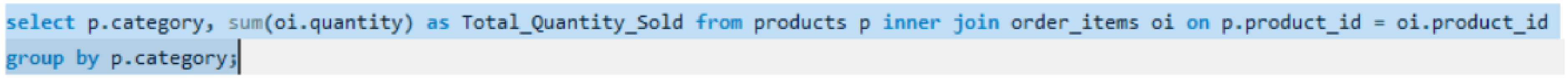
customer_id	Total_Sales
1	183747.44
2	284420.07
3	253783.31
4	137562.22
5	262504.19
6	212173.58
7	167960.11
8	164701.78
9	106226.03
10	252722.75

E-commerce Sales Analysis

Level 3: Aggregations

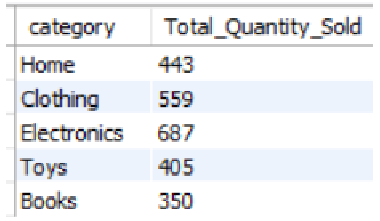
Q7. List the number of products sold per category

INPUT



```
select p.category, sum(oi.quantity) as Total_Quantity_Sold from products p inner join order_items oi on p.product_id = oi.product_id
group by p.category;
```

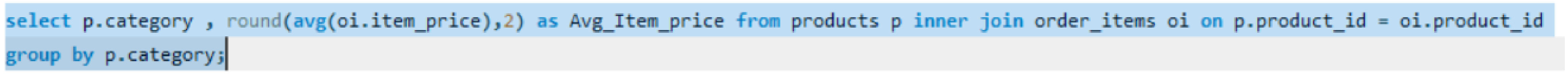
OUTPUT



category	Total_Quantity_Sold
Home	443
Clothing	559
Electronics	687
Toys	405
Books	350

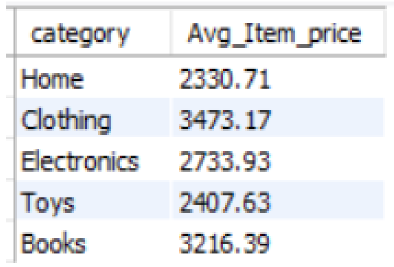
Q8. Find the average item price per category

INPUT



```
select p.category , round(avg(oi.item_price),2) as Avg_Item_price from products p inner join order_items oi on p.product_id = oi.product_id
group by p.category;
```

OUTPUT



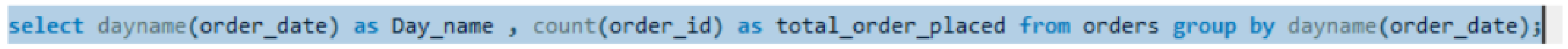
category	Avg_Item_price
Home	2330.71
Clothing	3473.17
Electronics	2733.93
Toys	2407.63
Books	3216.39

E-commerce Sales Analysis

Level 3: Aggregations

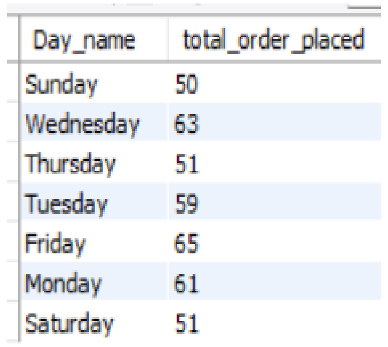
Q9. Show number of orders placed per day

INPUT



```sql
select dayname(order_date) as Day_name , count(order_id) as total_order_placed from orders group by dayname(order_date);
```

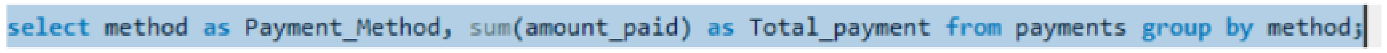
OUTPUT



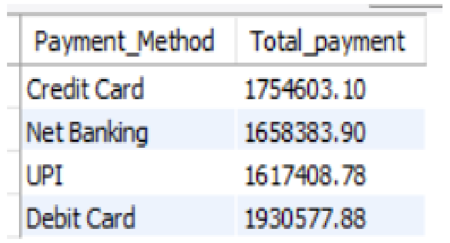
Day_name	total_order_placed
Sunday	50
Wednesday	63
Thursday	51
Tuesday	59
Friday	65
Monday	61
Saturday	51

Q10. List total payments received per payment method

INPUT



```sql
select method as Payment_Method, sum(amount_paid) as Total_payment from payments group by method;
```



Payment_Method	Total_payment
Credit Card	1754603.10
Net Banking	1658383.90
UPI	1617408.78
Debit Card	1930577.88

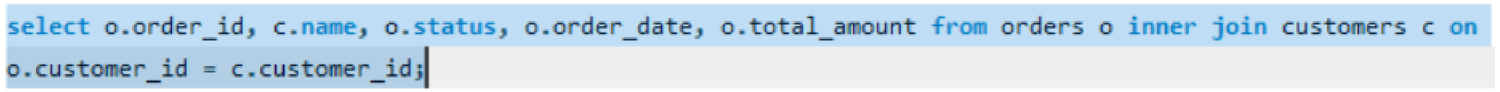
OUTPUT

E-commerce Sales Analysis

Level 4: Multi-Table Queries(Joins)

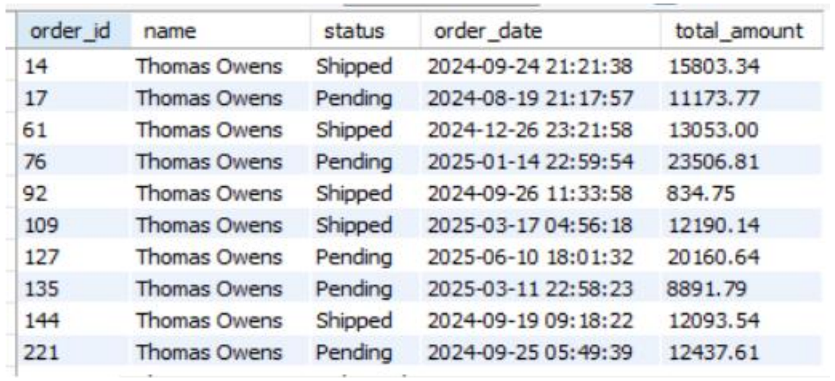
Q1. Retrieve order details along with the customer name (INNER JOIN)

INPUT



```sql
select o.order_id, c.name, o.status, o.order_date, o.total_amount from orders o inner join customers c on
o.customer_id = c.customer_id;
```

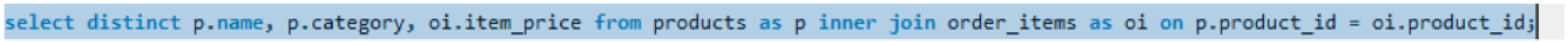
OUTPUT



order_id	name	status	order_date	total_amount
14	Thomas Owens	Shipped	2024-09-24 21:21:38	15803.34
17	Thomas Owens	Pending	2024-08-19 21:17:57	11173.77
61	Thomas Owens	Shipped	2024-12-26 23:21:58	13053.00
76	Thomas Owens	Pending	2025-01-14 22:59:54	23506.81
92	Thomas Owens	Shipped	2024-09-26 11:33:58	834.75
109	Thomas Owens	Shipped	2025-03-17 04:56:18	12190.14
127	Thomas Owens	Pending	2025-06-10 18:01:32	20160.64
135	Thomas Owens	Pending	2025-03-11 22:58:23	8891.79
144	Thomas Owens	Shipped	2024-09-19 09:18:22	12093.54
221	Thomas Owens	Pending	2024-09-25 05:49:39	12437.61

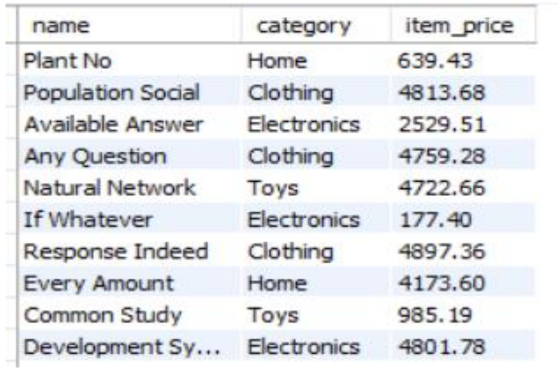
Q2. Get list of products that have been sold (INNER JOIN)

INPUT



```sql
select distinct p.name, p.category, oi.item_price from products as p inner join order_items as oi on p.product_id = oi.product_id;
```

OUTPUT



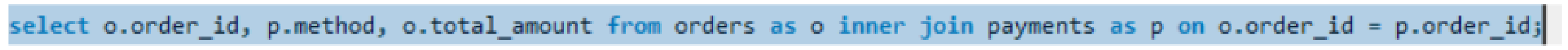
name	category	item_price
Plant No	Home	639.43
Population Social	Clothing	4813.68
Available Answer	Electronics	2529.51
Any Question	Clothing	4759.28
Natural Network	Toys	4722.66
If Whatever	Electronics	177.40
Response Indeed	Clothing	4897.36
Every Amount	Home	4173.60
Common Study	Toys	985.19
Development Sy...	Electronics	4801.78

E-commerce Sales Analysis

Level 4: Multi-Table Queries(Joins)

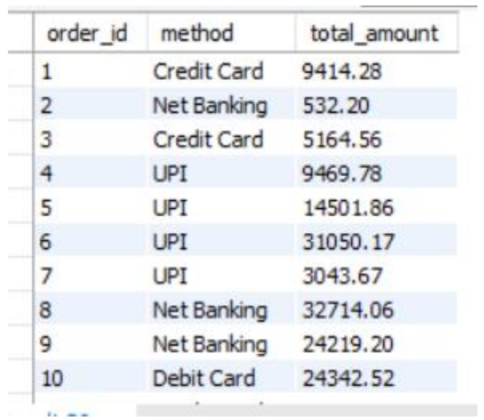
Q3. List all orders with their payment method (INNER JOIN)

INPUT



```sql
select o.order_id, p.method, o.total_amount from orders as o inner join payments as p on o.order_id = p.order_id;
```

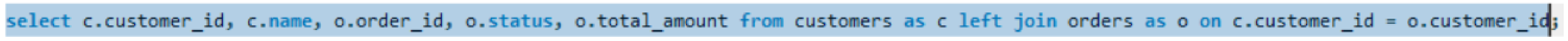
OUTPUT



order_id	method	total_amount
1	Credit Card	9414.28
2	Net Banking	532.20
3	Credit Card	5164.56
4	UPI	9469.78
5	UPI	14501.86
6	UPI	31050.17
7	UPI	3043.67
8	Net Banking	32714.06
9	Net Banking	24219.20
10	Debit Card	24342.52

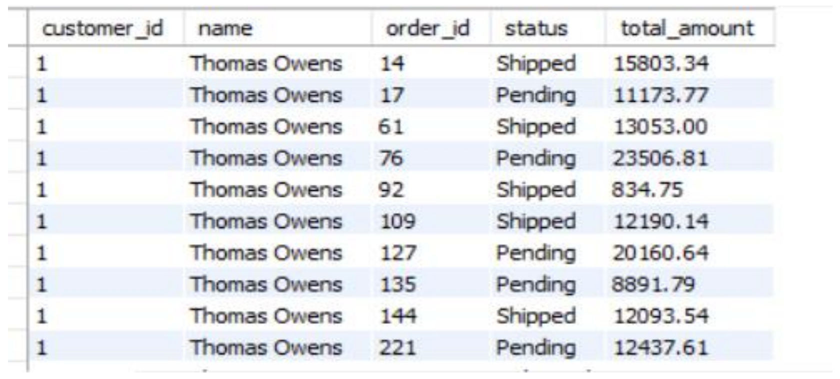
Q4. Get list of customers and their orders (LEFT JOIN)

INPUT



```sql
select c.customer_id, c.name, o.order_id, o.status, o.total_amount from customers as c left join orders as o on c.customer_id = o.customer_id;
```

OUTPUT



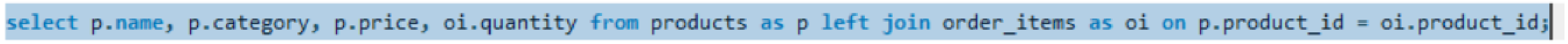
customer_id	name	order_id	status	total_amount
1	Thomas Owens	14	Shipped	15803.34
1	Thomas Owens	17	Pending	11173.77
1	Thomas Owens	61	Shipped	13053.00
1	Thomas Owens	76	Pending	23506.81
1	Thomas Owens	92	Shipped	834.75
1	Thomas Owens	109	Shipped	12190.14
1	Thomas Owens	127	Pending	20160.64
1	Thomas Owens	135	Pending	8891.79
1	Thomas Owens	144	Shipped	12093.54
1	Thomas Owens	221	Pending	12437.61

E-commerce Sales Analysis

Level 4: Multi-Table Queries(Joins)

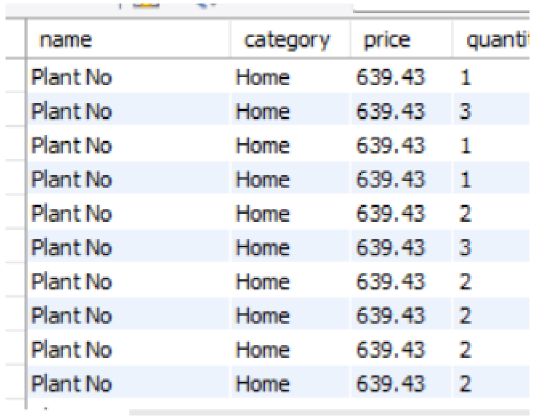
Q5. List all products along with order item quantity (LEFT JOIN)

INPUT



```
select p.name, p.category, p.price, oi.quantity from products as p left join order_items as oi on p.product_id = oi.product_id;
```

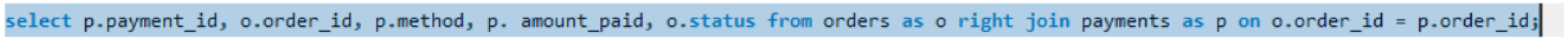
OUTPUT



name	category	price	quanti
Plant No	Home	639.43	1
Plant No	Home	639.43	3
Plant No	Home	639.43	1
Plant No	Home	639.43	1
Plant No	Home	639.43	2
Plant No	Home	639.43	3
Plant No	Home	639.43	2
Plant No	Home	639.43	2
Plant No	Home	639.43	2
Plant No	Home	639.43	2

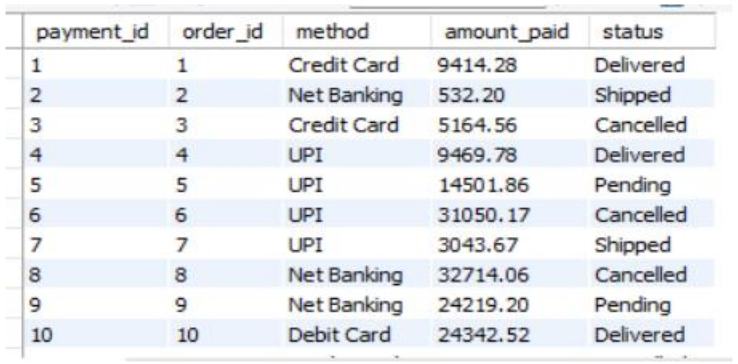
Q6. List all payments including those with no matching orders (RIGHT)

INPUT



```
select p.payment_id, o.order_id, p.method, p. amount_paid, o.status from orders as o right join payments as p on o.order_id = p.order_id;
```

OUTPUT



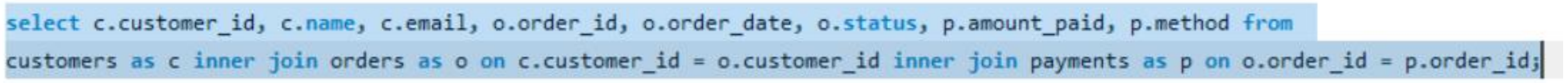
payment_id	order_id	method	amount_paid	status
1	1	Credit Card	9414.28	Delivered
2	2	Net Banking	532.20	Shipped
3	3	Credit Card	5164.56	Cancelled
4	4	UPI	9469.78	Delivered
5	5	UPI	14501.86	Pending
6	6	UPI	31050.17	Cancelled
7	7	UPI	3043.67	Shipped
8	8	Net Banking	32714.06	Cancelled
9	9	Net Banking	24219.20	Pending
10	10	Debit Card	24342.52	Delivered

E-commerce Sales Analysis

Level 4: Multi-Table Queries(Joins)

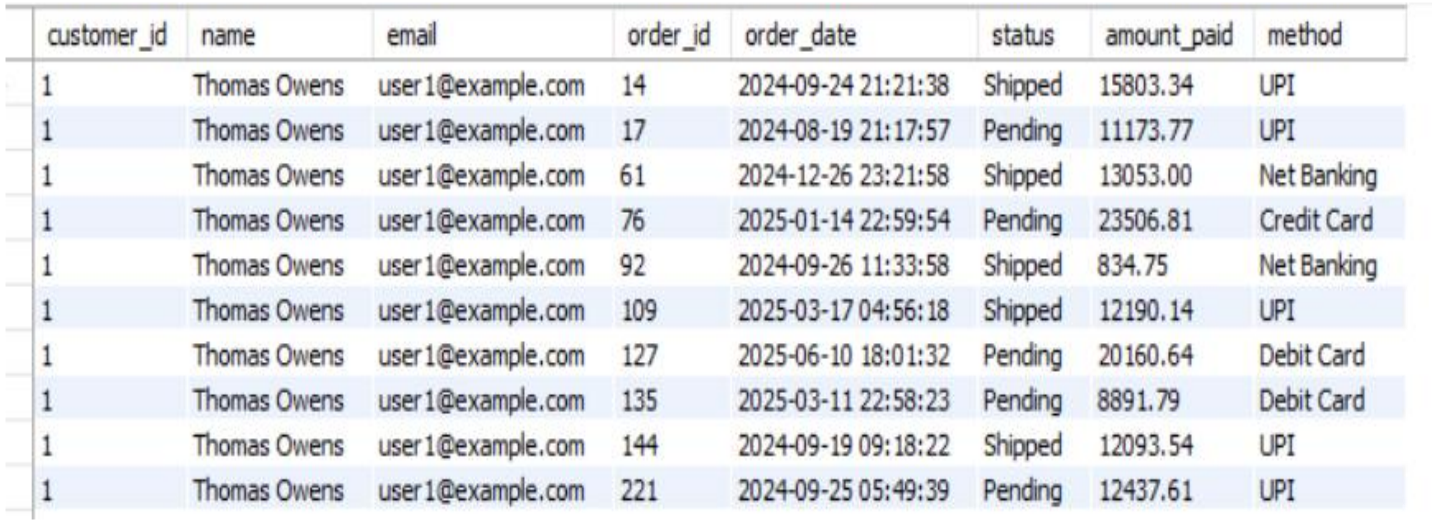
Q7. Combine data from three tables: customer, order, and payment

INPUT



```
select c.customer_id, c.name, c.email, o.order_id, o.order_date, o.status, p.amount_paid, p.method from
customers as c inner join orders as o on c.customer_id = o.customer_id inner join payments as p on o.order_id = p.order_id;
```

OUTPUT



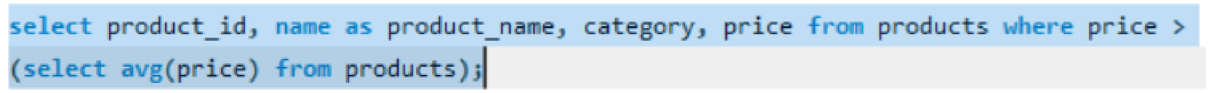
customer_id	name	email	order_id	order_date	status	amount_paid	method
1	Thomas Owens	user1@example.com	14	2024-09-24 21:21:38	Shipped	15803.34	UPI
1	Thomas Owens	user1@example.com	17	2024-08-19 21:17:57	Pending	11173.77	UPI
1	Thomas Owens	user1@example.com	61	2024-12-26 23:21:58	Shipped	13053.00	Net Banking
1	Thomas Owens	user1@example.com	76	2025-01-14 22:59:54	Pending	23506.81	Credit Card
1	Thomas Owens	user1@example.com	92	2024-09-26 11:33:58	Shipped	834.75	Net Banking
1	Thomas Owens	user1@example.com	109	2025-03-17 04:56:18	Shipped	12190.14	UPI
1	Thomas Owens	user1@example.com	127	2025-06-10 18:01:32	Pending	20160.64	Debit Card
1	Thomas Owens	user1@example.com	135	2025-03-11 22:58:23	Pending	8891.79	Debit Card
1	Thomas Owens	user1@example.com	144	2024-09-19 09:18:22	Shipped	12093.54	UPI
1	Thomas Owens	user1@example.com	221	2024-09-25 05:49:39	Pending	12437.61	UPI

E-commerce Sales Analysis

Level 5: Subqueries (Inner Queries)

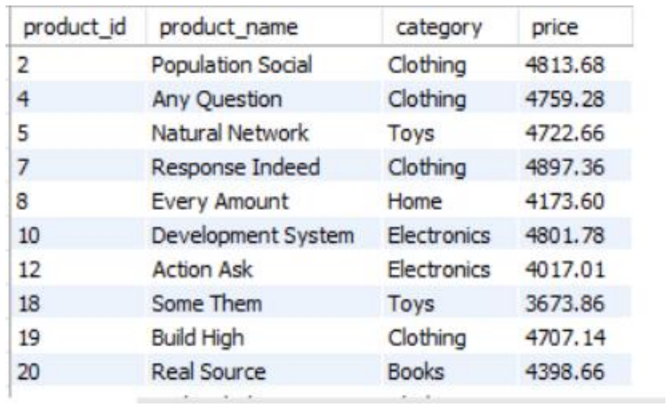
Q1. List all products priced above the average product price

INPUT



```sql
select product_id, name as product_name, category, price from products where price >
(select avg(price) from products);
```

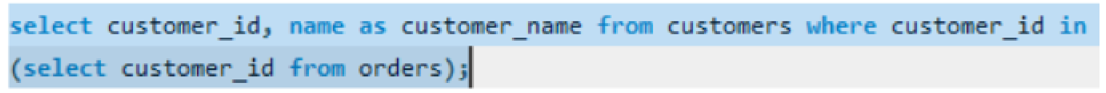
OUTPUT



product_id	product_name	category	price
2	Population Social	Clothing	4813.68
4	Any Question	Clothing	4759.28
5	Natural Network	Toys	4722.66
7	Response Indeed	Clothing	4897.36
8	Every Amount	Home	4173.60
10	Development System	Electronics	4801.78
12	Action Ask	Electronics	4017.01
18	Some Them	Toys	3673.86
19	Build High	Clothing	4707.14
20	Real Source	Books	4398.66

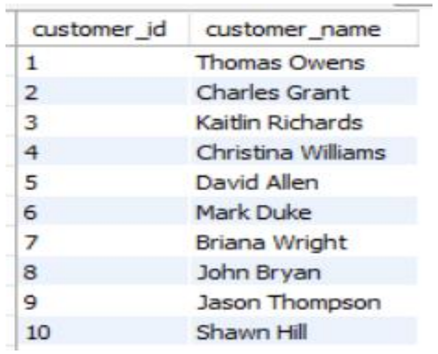
Q2. Find customers who have placed at least one order

INPUT



```sql
select customer_id, name as customer_name from customers where customer_id in
(select customer_id from orders);
```

OUTPUT



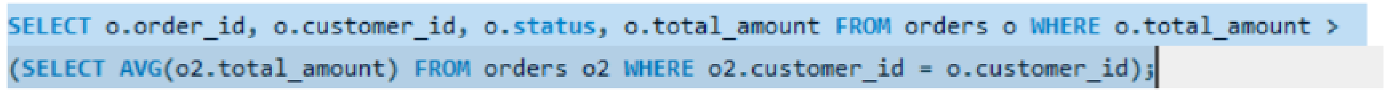
customer_id	customer_name
1	Thomas Owens
2	Charles Grant
3	Kaitlin Richards
4	Christina Williams
5	David Allen
6	Mark Duke
7	Briana Wright
8	John Bryan
9	Jason Thompson
10	Shawn Hill

E-commerce Sales Analysis

Level 5: Subqueries (Inner Queries)

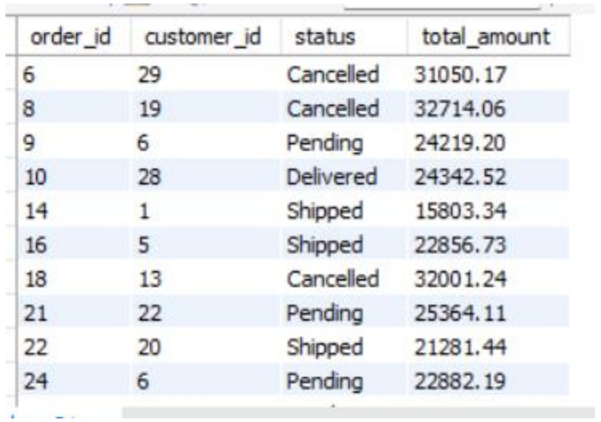
Q3. Show orders whose total amount is above the average for that customer

INPUT



```
SELECT o.order_id, o.customer_id, o.status, o.total_amount FROM orders o WHERE o.total_amount >
(SELECT AVG(o2.total_amount) FROM orders o2 WHERE o2.customer_id = o.customer_id);
```

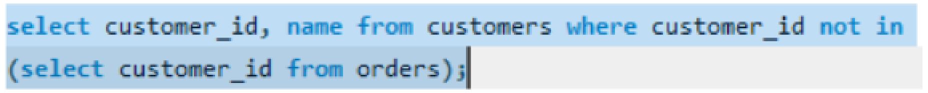
OUTPUT



order_id	customer_id	status	total_amount
6	29	Cancelled	31050.17
8	19	Cancelled	32714.06
9	6	Pending	24219.20
10	28	Delivered	24342.52
14	1	Shipped	15803.34
16	5	Shipped	22856.73
18	13	Cancelled	32001.24
21	22	Pending	25364.11
22	20	Shipped	21281.44
24	6	Pending	22882.19

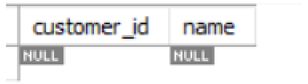
Q4. Display customers who haven't placed any orders

INPUT



```
select customer_id, name from customers where customer_id not in
(select customer_id from orders);
```

OUTPUT



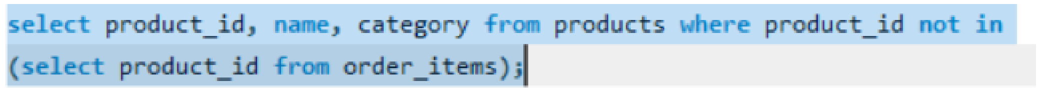
customer_id	name
NULL	NULL

E-commerce Sales Analysis

Level 5: Subqueries (Inner Queries)

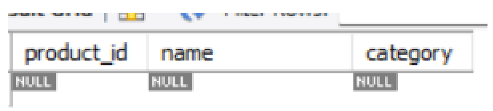
Q5. Show products that were never ordered

INPUT



```
select product_id, name, category from products where product_id not in
(select product_id from order_items);
```

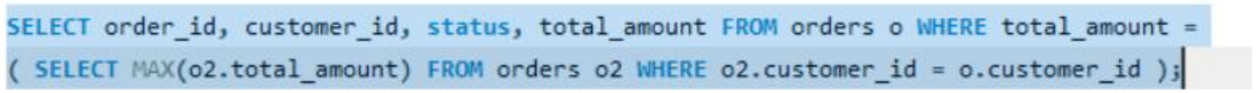
OUTPUT



product_id	name	category
NULL	NULL	NULL

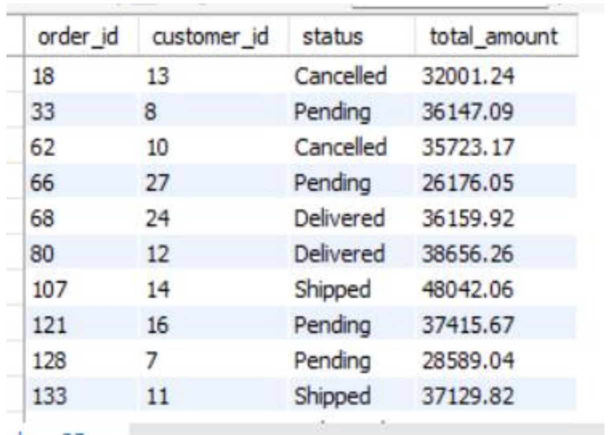
Q6. Show highest value order per customer

INPUT



```
SELECT order_id, customer_id, status, total_amount FROM orders o WHERE total_amount =
( SELECT MAX(o2.total_amount) FROM orders o2 WHERE o2.customer_id = o.customer_id );
```

OUTPUT



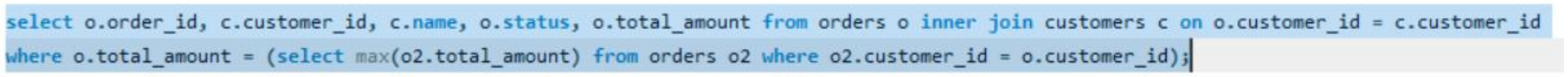
order_id	customer_id	status	total_amount
18	13	Cancelled	32001.24
33	8	Pending	36147.09
62	10	Cancelled	35723.17
66	27	Pending	26176.05
68	24	Delivered	36159.92
80	12	Delivered	38656.26
107	14	Shipped	48042.06
121	16	Pending	37415.67
128	7	Pending	28589.04
133	11	Shipped	37129.82

E-commerce Sales Analysis

Level 5: Subqueries (Inner Queries)

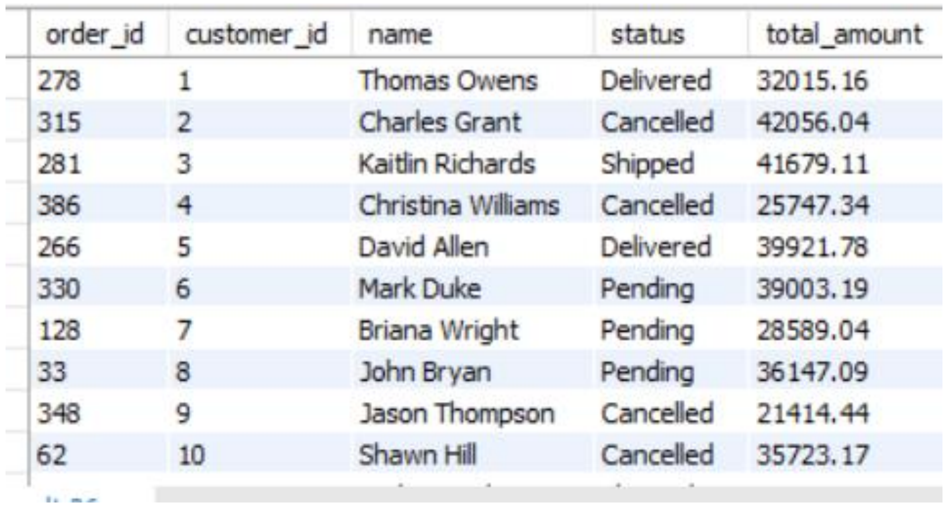
Q7. Highest Order Per Customer (Including Names)

INPUT



```
select o.order_id, c.customer_id, c.name, o.status, o.total_amount from orders o inner join customers c on o.customer_id = c.customer_id
where o.total_amount = (select max(o2.total_amount) from orders o2 where o2.customer_id = o.customer_id);
```

OUTPUT



order_id	customer_id	name	status	total_amount
278	1	Thomas Owens	Delivered	32015.16
315	2	Charles Grant	Cancelled	42056.04
281	3	Kaitlin Richards	Shipped	41679.11
386	4	Christina Williams	Cancelled	25747.34
266	5	David Allen	Delivered	39921.78
330	6	Mark Duke	Pending	39003.19
128	7	Briana Wright	Pending	28589.04
33	8	John Bryan	Pending	36147.09
348	9	Jason Thompson	Cancelled	21414.44
62	10	Shawn Hill	Cancelled	35723.17

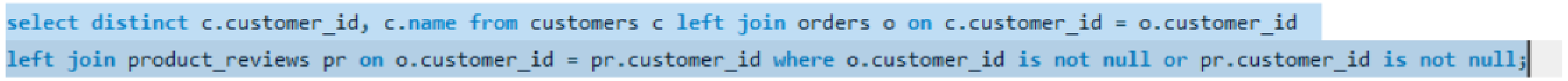
E-commerce Sales Analysis

Level 6: Set Operations

Q1. List all customers who have either placed an order or written a product review

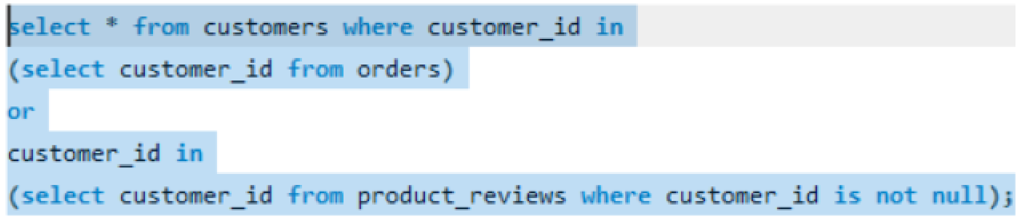
INPUT

Using Double join



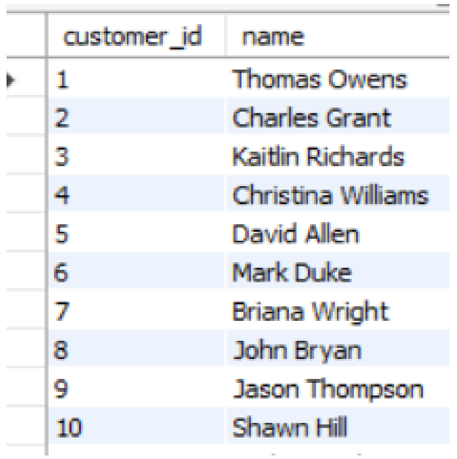
```
select distinct c.customer_id, c.name from customers c left join orders o on c.customer_id = o.customer_id
left join product_reviews pr on o.customer_id = pr.customer_id where o.customer_id is not null or pr.customer_id is not null;
```

Using union



```
select * from customers where customer_id in
(select customer_id from orders)
or
customer_id in
(select customer_id from product_reviews where customer_id is not null);
```

OUTPUT



customer_id	name
1	Thomas Owens
2	Charles Grant
3	Kaitlin Richards
4	Christina Williams
5	David Allen
6	Mark Duke
7	Briana Wright
8	John Bryan
9	Jason Thompson
10	Shawn Hill

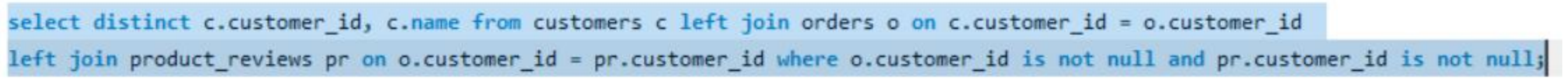
E-commerce Sales Analysis

Level 6: Set Operations

Q2. List all customers who have placed an order as well as reviewed a product

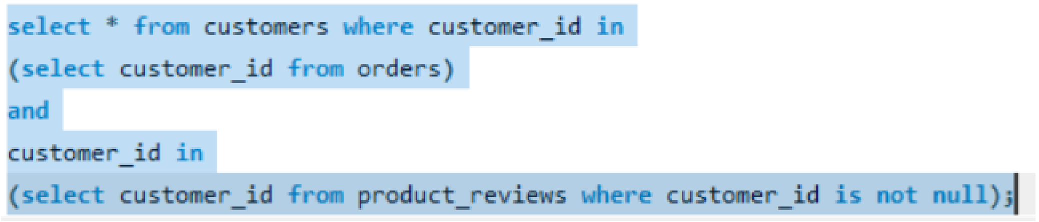
INPUT

Using Double join



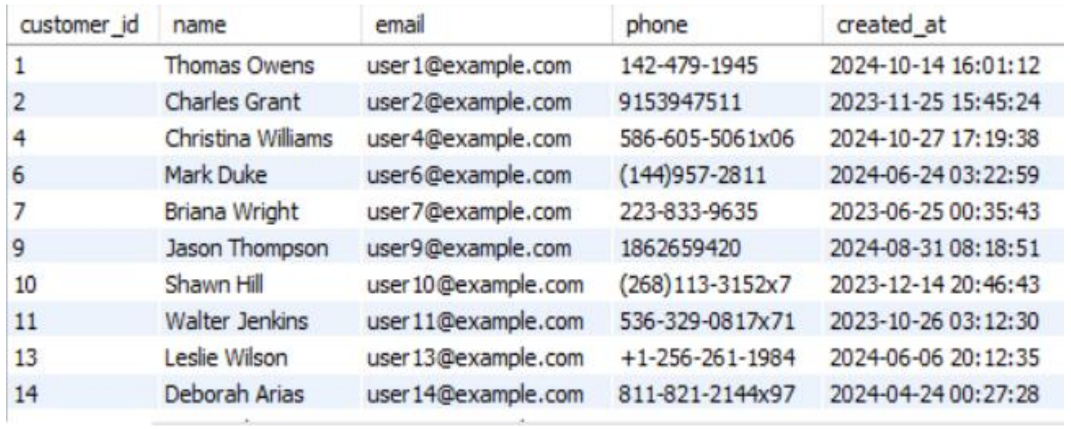
```
select distinct c.customer_id, c.name from customers c left join orders o on c.customer_id = o.customer_id
left join product_reviews pr on o.customer_id = pr.customer_id where o.customer_id is not null and pr.customer_id is not null;
```

Using SubQuery



```
select * from customers where customer_id in
(select customer_id from orders)
and
customer_id in
(select customer_id from product_reviews where customer_id is not null);
```

OUTPUT



customer_id	name	email	phone	created_at
1	Thomas Owens	user1@example.com	142-479-1945	2024-10-14 16:01:12
2	Charles Grant	user2@example.com	9153947511	2023-11-25 15:45:24
4	Christina Williams	user4@example.com	586-605-5061x06	2024-10-27 17:19:38
6	Mark Duke	user6@example.com	(144)957-2811	2024-06-24 03:22:59
7	Briana Wright	user7@example.com	223-833-9635	2023-06-25 00:35:43
9	Jason Thompson	user9@example.com	1862659420	2024-08-31 08:18:51
10	Shawn Hill	user10@example.com	(268)113-3152x7	2023-12-14 20:46:43
11	Walter Jenkins	user11@example.com	536-329-0817x71	2023-10-26 03:12:30
13	Leslie Wilson	user13@example.com	+1-256-261-1984	2024-06-06 20:12:35
14	Deborah Arias	user14@example.com	811-821-2144x97	2024-04-24 00:27:28