1. Identify the Most Popular Product by Order Count

```
SELECT p.product_id, p.name AS product_name, COUNT(oi.order_id) AS order_count
FROM products p
JOIN order_items oi ON p.product_id = oi.product_id
GROUP BY p.product_id, p.name
ORDER BY order_count DESC
LIMIT 1;
```

product_id	product_name	order_count
5	Whole Wheat Bread	1081

---

2. Find Users Who Have Never Placed an Order

```
SELECT u.user_id, u.full_name, u.email
FROM users u
LEFT JOIN orders o ON u.user_id = o.user_id
WHERE o.order_id IS NULL;
```

user_id	full_name	email
7	John Brown	user7@ecommerce.com
36	Daniel Brown	user36@ecommerce.com
48	Charlotte Martinez	user48@ecommerce.com
80	Emma Garcia	user80@ecommerce.com
107	Amelia Anderson	user107@ecommerce.com
126	Amelia Smith	user126@ecommerce.com
134	Benjamin Martin	user134@ecommerce.com
196	Ava Johnson	user196@ecommerce.com
197	Ava Jackson	user197@ecommerce.com
219	Emma Harris	user219@ecommerce.com
256	James Anderson	user256@ecommerce.com
258	Olivia Rodriguez	user258@ecommerce.com
276	Benjamin Thomas	user276@ecommerce.com
366	Daniel Allen	user366@ecommerce.com
390	William Lewis	user390@ecommerce.com

---

3. Calculate User's Average Order Value

```
SELECT o.user_id, u.full_name, AVG(o.total_amount) AS avg_order_value
FROM orders o
JOIN users u ON o.user_id = u.user_id
GROUP BY o.user_id, u.full_name
ORDER BY avg_order_value DESC;
```

user_id	full_name	avg_order_value
1673	Mia Harris	1496.940000
162	Harper Hall	1496.630000
2927	Benjamin Johnson	1496.600000
6296	Daniel Harris	1495.120000
997	Harper Allen	1494.660000
1755	John Martinez	1494.190000
3268	James Thompson	1493.670000
7826	Harper Allen	1493.440000
7668	Sarah Rodriguez	1492.810000
6176	Emma Johnson	1491.910000
2242	Isabella Robinson	1491.510000
9459	Amelia Martin	1490.550000
1703	Benjamin Smith	1489.740000
9012	Benjamin Anderson	1488.930000
4787	Benjamin Hall	1488.710000

---

4. Find Active Users Based on Page Views

```
SELECT p.user_id, u.full_name, COUNT(p.view_id) AS total_views
FROM page_views p
JOIN users u ON p.user_id = u.user_id
GROUP BY p.user_id, u.full_name
ORDER BY total_views DESC
LIMIT 10;
```

user_id	full_name	total_views
3960	Emma Anderson	9
7693	James Hall	9
3827	Amelia Robinson	8
673	Sarah Brown	8
822	Harper Anderson	8
2453	Sophia Harris	8
1073	William Martin	8
3355	Olivia Clark	8
7762	Harper Thomas	8
3474	James Robinson	8

---

25. Rank Users by Total Spending

```
SELECT u.user_id, u.full_name, SUM(o.total_amount) AS total_spent,
RANK() OVER (ORDER BY SUM(o.total_amount) DESC) AS spending_rank
FROM orders o
JOIN users  u ON o.user_id = u.user_id
GROUP BY u.user_id, u.full_name;
```

	user_id	full_name	total_spent	spending_rank
▶	398	Michael Walker	10181.53	1
	1627	James Lewis	9243.51	2
	5977	Sophia Jackson	9197.17	3
	1195	Sarah Lewis	8965.67	4
	1697	Sophia White	8874.65	5
	1802	Olivia Thomas	8845.39	6
	928	Harper Robinson	8800.36	7
	9901	Charlotte Hall	8768.20	8
	9395	Charlotte White	8708.06	9
	3758	Isabella Garcia	8529.62	10
	648	Isabella Clark	8512.65	11
	4311	Sophia Robinson	8495.04	12
	2569	Charlotte Taylor	8339.11	13
	4138	Benjamin Thom…	8337.14	14
	9324	Ava Hall	8266.54	15

---

26. Find the Top 5 Revenue Generating Products

```
SELECT p.name, SUM(oi.quantity * oi.price) AS revenue
FROM order_items oi
JOIN products p ON oi.product_id = p.product_id
GROUP BY p.name
ORDER BY revenue DESC
LIMIT 5;
```

	name	revenue
▶	Macaroni	812236.11
	Baby Wipes	810864.14
	Whole Wheat Bread	805790.49
	Vodka	802807.96
	Laundry Detergent	802185.40

---

27. Retrieve Orders That Contain More Than 5 Items

```
SELECT order_id, COUNT(*) AS item_count
FROM order_items
GROUP BY order_id
HAVING COUNT(*) > 5;
```

order_id	item_count
28	7
39	6
68	6
104	6
118	7
127	7
177	8
178	6
184	6
213	6
236	6
245	7
274	6
289	6
323	8

---

28. Calculate Percentage of Orders in Each Status

```
SELECT order_status, COUNT(*) * 100.0 / SUM(COUNT(*)) OVER() AS percentage
FROM orders GROUP BY order_status;
```

order_id	item_count
28	7
39	6
68	6
104	6
118	7
127	7
177	8
178	6
184	6
213	6
236	6
245	7
274	6
289	6
323	8

---

29. Find Customers Who Have Used More Than One Payment Method

```
SELECT user_id, COUNT(DISTINCT payment_method) AS payment_methods
FROM payments
GROUP BY user_id
HAVING COUNT(DISTINCT payment_method) > 1;
```

	user_id	payment_methods
▶	1	3
	3	3
	4	3
	5	4
	6	3
	7	3
	8	3
	12	2
	14	2
	15	4
	17	2
	18	3
	20	2
	21	2
	22	3

---

30. Identify Users with the Highest Number of Reviews

```
SELECT user_id, COUNT(review_id) AS total_reviews
FROM reviews
GROUP BY user_id
ORDER BY total_reviews DESC
LIMIT 5;
```

	user_id	total_reviews
▶	7177	9
	9735	9
	7462	9
	961	8
	4115	8

---

31. Determine the Most Common Product Size Ordered

```
SELECT pv.size, COUNT(oi.product_id) AS total_orders
FROM order_items oi
JOIN product_variants pv ON oi.product_id = pv.product_id
GROUP BY pv.size
ORDER BY total_orders DESC;
```

	size	total_orders
▶	Large	39753
	Small	35902
	Medium	34177
	Family Pack	32929

---

32. Retrieve the Most Frequently Searched Terms

```
SELECT search_query, COUNT(*) AS search_count
FROM search_logs
GROUP BY search_query
ORDER BY search_count DESC
LIMIT 10;
```

search_query	search_count
if	41
recognize	40
dark	40
until	40
sit	39
customer	39
physical	39
forward	38
movie	38
that	38

---

33. Find the Orders That Used Coupons

```
SELECT DISTINCT o.order_id, c.coupon_code
FROM orders o
JOIN payments p ON o.order_id = p.order_id
JOIN coupons c ON p.order_id = c.coupon_id;
```

order_id	coupon_code
2133	COUPON-100006
1560	COUPON-100793
2651	COUPON-100852
4097	COUPON-101510
2994	COUPON-101734
1395	COUPON-101859
2940	COUPON-101869
380	COUPON-102000
2259	COUPON-102919
3535	COUPON-103831
343	COUPON-104729
3210	COUPON-105440
1490	COUPON-106213
1369	COUPON-106600
3413	COUPON-106628

---

37. Identify Users Who Have Repeatedly Contacted Customer Support

```
SELECT user_id, COUNT(*) AS ticket_count
FROM customer_support_tickets
GROUP BY user_id
HAVING COUNT(*) > 1;
```

user_id	ticket_count
2	4
5	4
6	2
13	2
15	2
16	2
19	2
20	2
21	2
26	3
31	2
32	2
37	2
41	3
43	2