

≡ OneCompiler

queries.sql 446pjw3ea

Pricing Learn ▾ Code ▾ Deploy ▾ More ▾ LOGIN

\* AI NEW SQLSERVER ▾ RUN ▶ :

```
14 store_id VARCHAR(10),
15 item_id VARCHAR(10),
16 item_category VARCHAR(50),
17 item_name VARCHAR(100)
18 );
19
20
21 INSERT INTO transactions VALUES
22 (3, '2019-09-19 21:19:06', NULL, 'a', 'a1', 58.00),
23 (12, '2019-12-10 20:10:14', '2019-12-15 23:19:06', 'b', 'b2', 475.00),
24 (3, '2020-09-01 23:59:46', '2020-09-02 21:22:06', 'f', 'f9', 33.00),
25 (2, '2020-04-30 21:19:06', NULL, 'd', 'd3', 250.00),
26 (1, '2020-10-22 22:20:06', NULL, 'f', 'f2', 91.00),
27 (8, '2020-04-16 21:10:22', NULL, 'e', 'e7', 24.00),
28 (5, '2019-09-23 12:09:35', '2019-09-27 02:55:02', 'g', 'g6', 61.00);
29
30 INSERT INTO items VALUES
31 ('a', 'a1', 'pants', 'denim pants'),
32 ('a', 'a2', 'tops', 'blouse'),
33 ('f', 'f1', 'table', 'coffee table'),
34 ('f', 'f5', 'chair', 'lounge chair'),
35 ('f', 'f6', 'chair', 'armchair'),
36 ('d', 'd2', 'jewelry', 'bracelet'),
37 ('b', 'b4', 'earphone', 'airpods');
38
39
40 --1
41 SELECT
42     DATEFROMPARTS(YEAR(purchase_time), MONTH(purchase_time), 1) AS purchase_month,
43     COUNT(*) AS purchase_count
44 FROM transactions
45 WHERE refund_item IS NULL
46 GROUP BY DATEFROMPARTS(YEAR(purchase_time), MONTH(purchase_time), 1)
47 ORDER BY purchase_month;
```

STDIN

Input for the program (Optional)

Output:

purchase_month	purchase_count
2019-09-01	1
2020-04-01	2
2020-10-01	1

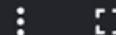


queries.sql 446pjw3ea 🖊

+ AI NEW

SQLSERVER ▾

RUN ➡



```
20
21 INSERT INTO transactions VALUES
22 (3, '2019-09-19 21:19:06', NULL, 'a', 'a1', 58.00),
23 (12, '2019-12-10 20:10:14', '2019-12-15 23:19:06', 'b', 'b2', 475.00),
24 (3, '2020-09-01 23:59:46', '2020-09-02 21:22:06', 'f', 'f9', 33.00),
25 (2, '2020-04-30 21:19:06', NULL, 'd', 'd3', 250.00),
26 (1, '2020-10-22 22:20:06', NULL, 'f', 'f2', 91.00),
27 (8, '2020-04-16 21:10:22', NULL, 'e', 'e7', 24.00),
28 (5, '2019-09-23 12:09:35', '2019-09-27 02:55:02', 'g', 'g6', 61.00);
29
30 INSERT INTO items VALUES
31 ('a', 'a1', 'pants', 'denim pants'),
32 ('a', 'a2', 'tops', 'blouse'),
33 ('f', 'f1', 'table', 'coffee table'),
34 ('f', 'f5', 'chair', 'lounge chair'),
35 ('f', 'f6', 'chair', 'armchair'),
36 ('d', 'd2', 'jewelry', 'bracelet'),
37 ('b', 'b4', 'earphone', 'airpods');
38
39
40 WITH store_counts AS (
41     SELECT
42         store_id,
43         COUNT(*) AS num_transactions
44     FROM transactions
45     WHERE purchase_time >= '2020-10-01' -- Removed DATE keyword
46     AND purchase_time < '2020-11-01' -- Removed DATE keyword
47     GROUP BY store_id
48 )
49 SELECT COUNT(*) AS num_stores_with_5_plus_orders
50 FROM store_counts
51 WHERE num_transactions >= 5;
52
53
54
```

STDIN

Input for the program (Optional)

Output:

num\_stores\_with\_5\_plus\_orders

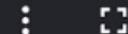
-----  
0

queries.sql 446pjw3ea 🖊

+ AI NEW

SQLSERVER ▾

RUN ➡



```
16     item_category VARCHAR(50),
17     item_name VARCHAR(100)
18 );
19
20
21 INSERT INTO transactions VALUES
22 (3, '2019-09-19 21:19:06', NULL, 'a', 'a1', 58.00),
23 (12, '2019-12-10 20:10:14', '2019-12-15 23:19:06', 'b', 'b2', 475.00),
24 (3, '2020-09-01 23:59:46', '2020-09-02 21:22:06', 'f', 'f9', 33.00),
25 (2, '2020-04-30 21:19:06', NULL, 'd', 'd3', 250.00),
26 (1, '2020-10-22 22:20:06', NULL, 'f', 'f2', 91.00),
27 (8, '2020-04-16 21:10:22', NULL, 'e', 'e7', 24.00),
28 (5, '2019-09-23 12:09:35', '2019-09-27 02:55:02', 'g', 'g6', 61.00);
29
30 INSERT INTO items VALUES
31 ('a', 'a1', 'pants', 'denim pants'),
32 ('a', 'a2', 'tops', 'blouse'),
33 ('f', 'f1', 'table', 'coffee table'),
34 ('f', 'f5', 'chair', 'lounge chair'),
35 ('f', 'f6', 'chair', 'armchair'),
36 ('d', 'd2', 'jewelry', 'bracelet'),
37 ('b', 'b4', 'earphone', 'airpods');
38
39
40 SELECT
41     store_id,
42     MIN(DATEDIFF(MINUTE, purchase_time, refund_item)) AS shortest_refund_i
43 FROM
44     transactions
45 WHERE
46     refund_item IS NOT NULL
47 GROUP BY
48     store_id;
```

STDIN

Input for the program (Optional)

Output:

store_id	shortest_refund_interval_min
b	7389
f	1283
g	5206



queries.sql

446pjw3ea

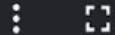
```
21 INSERT INTO transactions VALUES
22 (3, '2019-09-19 21:19:06', NULL, 'a', 'a1', 58.00),
23 (12,'2019-12-10 20:10:14','2019-12-15 23:19:06','b','b2',475.00),
24 (3, '2020-09-01 23:59:46', '2020-09-02 21:22:06', 'f', 'f9', 33.00),
25 (2, '2020-04-30 21:19:06',NULL,'d','d3',250.00),
26 (1, '2020-10-22 22:20:06',NULL,'f','f2',91.00),
27 (8, '2020-04-16 21:10:22',NULL,'e','e7',24.00),
28 (5, '2019-09-23 12:09:35','2019-09-27 02:55:02','g','g6',61.00);
29
30 INSERT INTO items VALUES
31 ('a','a1','pants','denim pants'),
32 ('a','a2','tops','blouse'),
33 ('f','f1','table','coffee table'),
34 ('f','f5','chair','lounge chair'),
35 ('f','f6','chair','armchair'),
36 ('d','d2','jewelry','bracelet'),
37 ('b','b4','earphone','airpods');
38
39
40 SELECT
41     store_id,
42     gross_transaction_value
43 FROM
44 (
45     SELECT
46         store_id,
47         gross_transaction_value,
48         ROW_NUMBER() OVER (PARTITION BY store_id ORDER BY purchase_time AS
49     FROM
50         transactions
51 ) AS ranked_transactions
52 WHERE
53     rn = 1;
54
```

↑ AI

NEW

SQLSERVER ▾

RUN ▶



STDIN

Input for the program (Optional)

Output:

store_id	gross_transaction_value
a	58.00
b	475.00
d	250.00
e	24.00
f	33.00
g	61.00



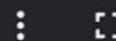
queries.sql

446pjw3ea

AI NEW

SQLSERVER ▾

RUN ▶



```
37 ('b','b4','earphone','airpods');
38
39
40 WITH first_purchase AS (
41     SELECT
42         t.*,
43         ROW_NUMBER() OVER (
44             PARTITION BY buyer_id
45             ORDER BY purchase_time
46         ) AS rn
47     FROM transactions t
48     -- If you want only non-refunded first purchases, use:
49     -- WHERE refund_item IS NULL
50 ),
51 first_purchase_items AS (
52     SELECT
53         fp.buyer_id,
54         fp.store_id,
55         fp.item_id,
56         i.item_name
57     FROM first_purchase fp
58     JOIN items i
59     ON fp.store_id = i.store_id
60     AND fp.item_id = i.item_id
61     WHERE fp.rn = 1
62 )
63 SELECT TOP 1
64     item_name,
65     COUNT(*) AS num_buyers
66 FROM first_purchase_items
67 GROUP BY item_name
68 ORDER BY num_buyers DESC, item_name;
69
70
71
```

STDIN

Input for the program (Optional)

Output:

item\_name

-----  
denim pants

queries.sql

446pjw3ea

+ AI

NEW

SQLSERVER

RUN ▶

⋮

```
20
21 INSERT INTO transactions VALUES
22 (3, '2019-09-19 21:19:06', NULL, 'a', 'a1', 58.00),
23 (12,'2019-12-10 20:10:14','2019-12-15 23:19:06','b','b2',475.00),
24 (3, '2020-09-01 23:59:46', '2020-09-02 21:22:06', 'f', 'f9', 33.00),
25 (2, '2020-04-30 21:19:06',NULL,'d','d3',250.00),
26 (1, '2020-10-22 22:20:06',NULL,'f','f2',91.00),
27 (8, '2020-04-16 21:10:22',NULL,'e','e7',24.00),
28 (5, '2019-09-23 12:09:35','2019-09-27 02:55:02','g','g6',61.00);
29
30 INSERT INTO items VALUES
31 ('a','a1','pants','denim pants'),
32 ('a','a2','tops','blouse'),
33 ('f','f1','table','coffee table'),
34 ('f','f5','chair','lounge chair'),
35 ('f','f6','chair','armchair'),
36 ('d','d2','jewelry','bracelet'),
37 ('b','b4','earphone','airpods');
38
39
40 SELECT
41     t.*,
42     i.item_category,
43     i.item_name,
44     CASE
45         WHEN t.refund_item IS NOT NULL
46             AND DATEDIFF(HOUR, t.purchase_time, t.refund_item) <= 72
47         THEN 1
48         ELSE 0
49     END AS refund_processable_flag
50 FROM transactions t
51 LEFT JOIN items i
52     ON t.store_id = i.store_id
53     AND t.item_id = i.item_id;
```

STDIN

Input for the program (Optional)

Output:

buyer_id	purchase_time	refund_item	store_id	item_id	gross_transac
3	2019-09-19 21:19:06.000		NULL	a	a1
12	2019-12-10 20:10:14.000	2019-12-15 23:19:06.000	b	b2	
3	2020-09-01 23:59:46.000	2020-09-02 21:22:06.000	f	f9	
2	2020-04-30 21:19:06.000		NULL	d	d3
1	2020-10-22 22:20:06.000		NULL	f	f2
8	2020-04-16 21:10:22.000		NULL	e	e7
5	2019-09-23 12:09:35.000	2019-09-27 02:55:02.000	g	g6	



queries.sql

446pjw3ea

+ AI

NEW

SQLSERVER ▾

RUN ▶

⋮ [ ]

```
20
21 INSERT INTO transactions VALUES
22 (3, '2019-09-19 21:19:06', NULL, 'a', 'a1', 50.00)
23 (12, '2019-12-10 20:10:14', '2019-12-15 23:19:00', 'b', 'b1', 100.00)
24 (3, '2020-09-01 23:59:46', '2020-09-02 21:22:00', 'c', 'c1', 200.00)
25 (2, '2020-04-30 21:19:06', NULL, 'd', 'd3', 250.00)
26 (1, '2020-10-22 22:20:06', NULL, 'f', 'f2', 91.00)
27 (8, '2020-04-16 21:10:22', NULL, 'e', 'e7', 24.00)
28 (5, '2019-09-23 12:09:35', '2019-09-27 02:55:00', 'g', 'g1', 150.00)
29
30 INSERT INTO items VALUES
31 ('a', 'a1', 'pants', 'denim pants'),
32 ('a', 'a2', 'tops', 'blouse'),
33 ('f', 'f1', 'table', 'coffee table'),
34 ('f', 'f5', 'chair', 'lounge chair'),
35 ('f', 'f6', 'chair', 'armchair'),
36 ('d', 'd2', 'jewelry', 'bracelet'),
37 ('b', 'b4', 'earphone', 'airpods');
38
39
40 WITH ranked_purchases AS (
41     SELECT
42         t.*,
43         ROW_NUMBER() OVER (
44             PARTITION BY buyer_id
45             ORDER BY purchase_time
46         ) AS purchase_rank
47     FROM transactions t
48 )
49 SELECT *
50 FROM ranked_purchases
51 WHERE purchase_rank = 2;
```

STDIN

Input for the program ( Optional )

Output:

buyer_id	purchase_time	refund_item	store_id	item_id	gross_transaction_value	purchase_rank
3	2020-09-01 23:59:46.000	2020-09-02 21:22:06.000	f	f9	33.00	



queries.sql

446pjw3ea ✍

+ AI NEW

SQLSERVER ▾

RUN ▶

⋮ [ ]

```
21 INSERT INTO transactions VALUES
22 (3, '2019-09-19 21:19:06', NULL, 'a', 'a1', 58.00),
23 (12,'2019-12-10 20:10:14','2019-12-15 23:19:06','b','b2',475.00),
24 (3, '2020-09-01 23:59:46','2020-09-02 21:22:06','f','f9',33.00),
25 (2, '2020-04-30 21:19:06',NULL,'d','d3',250.00),
26 (1, '2020-10-22 22:20:06',NULL,'f','f2',91.00),
27 (8, '2020-04-16 21:10:22',NULL,'e','e7',24.00),
28 (5, '2019-09-23 12:09:35','2019-09-27 02:55:02','g','g6',61.00);
29
30 INSERT INTO items VALUES
31 ('a','a1','pants','denim pants'),
32 ('a','a2','tops','blouse'),
33 ('f','f1','table','coffee table'),
34 ('f','f5','chair','lounge chair'),
35 ('f','f6','chair','armchair'),
36 ('d','d2','jewelry','bracelet'),
37 ('b','b4','earphone','airpods');
38
39
40 WITH ranked_transactions AS (
41     SELECT
42         buyer_id,
43         purchase_time,
44         ROW_NUMBER() OVER (
45             PARTITION BY buyer_id
46             ORDER BY purchase_time
47         ) AS rn
48     FROM transactions
49 )
50 SELECT
51     buyer_id,
52     purchase_time AS second_transaction_time
53 FROM ranked_transactions
54 WHERE rn = 2
55 ORDER BY buyer_id;
```

STDIN

Input for the program (Optional)

Output:

buyer\_id second\_transaction\_time

-----  
3 2020-09-01 23:59:46.000