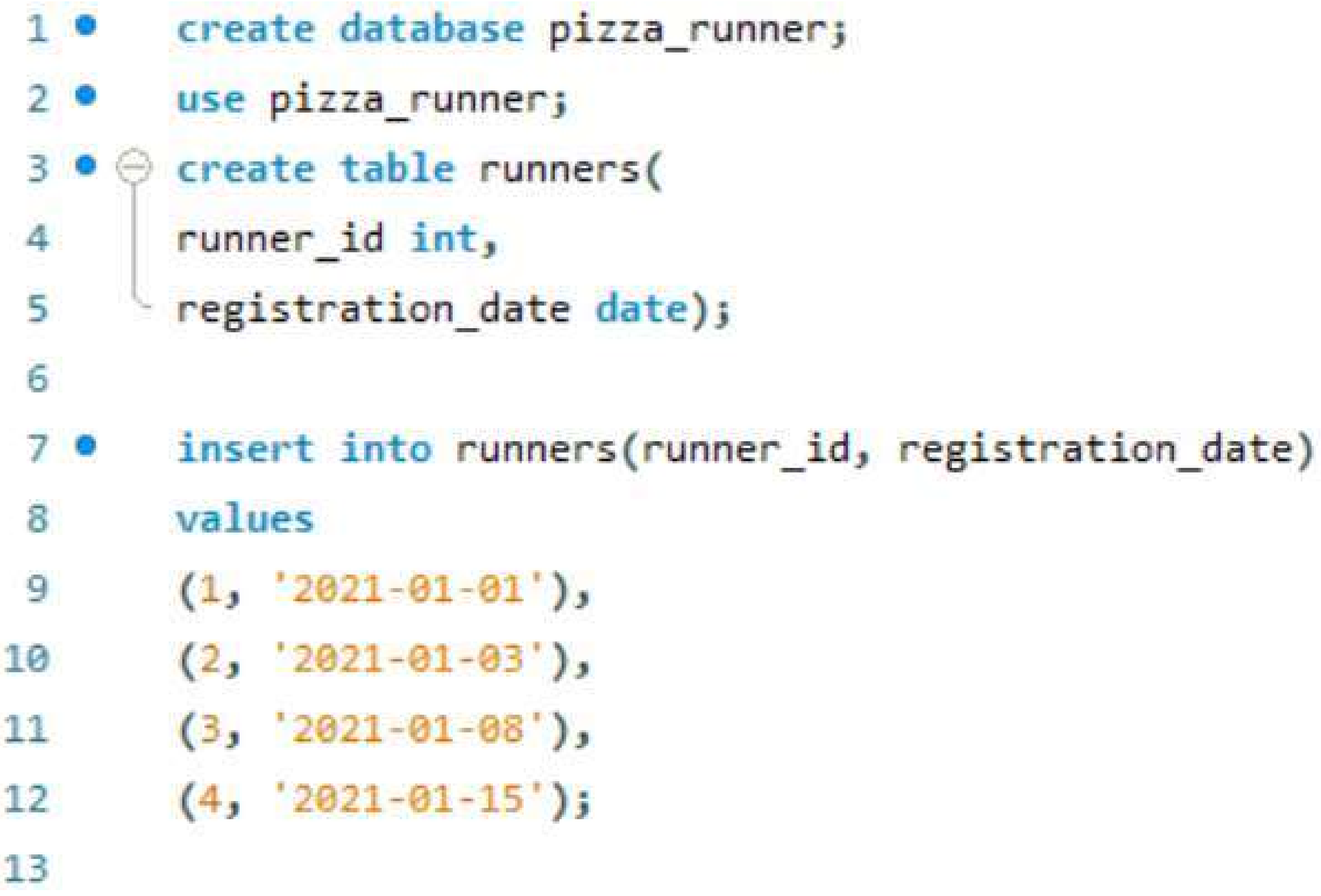
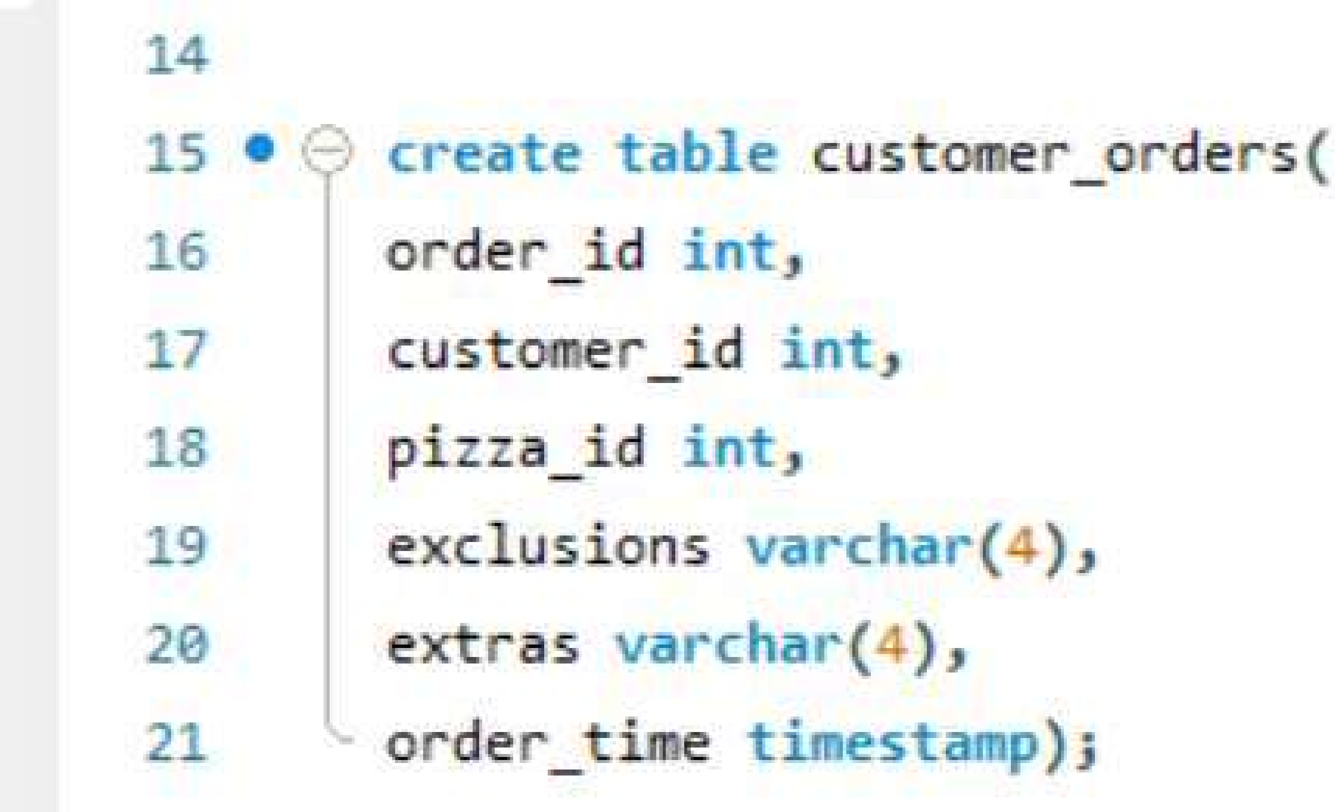
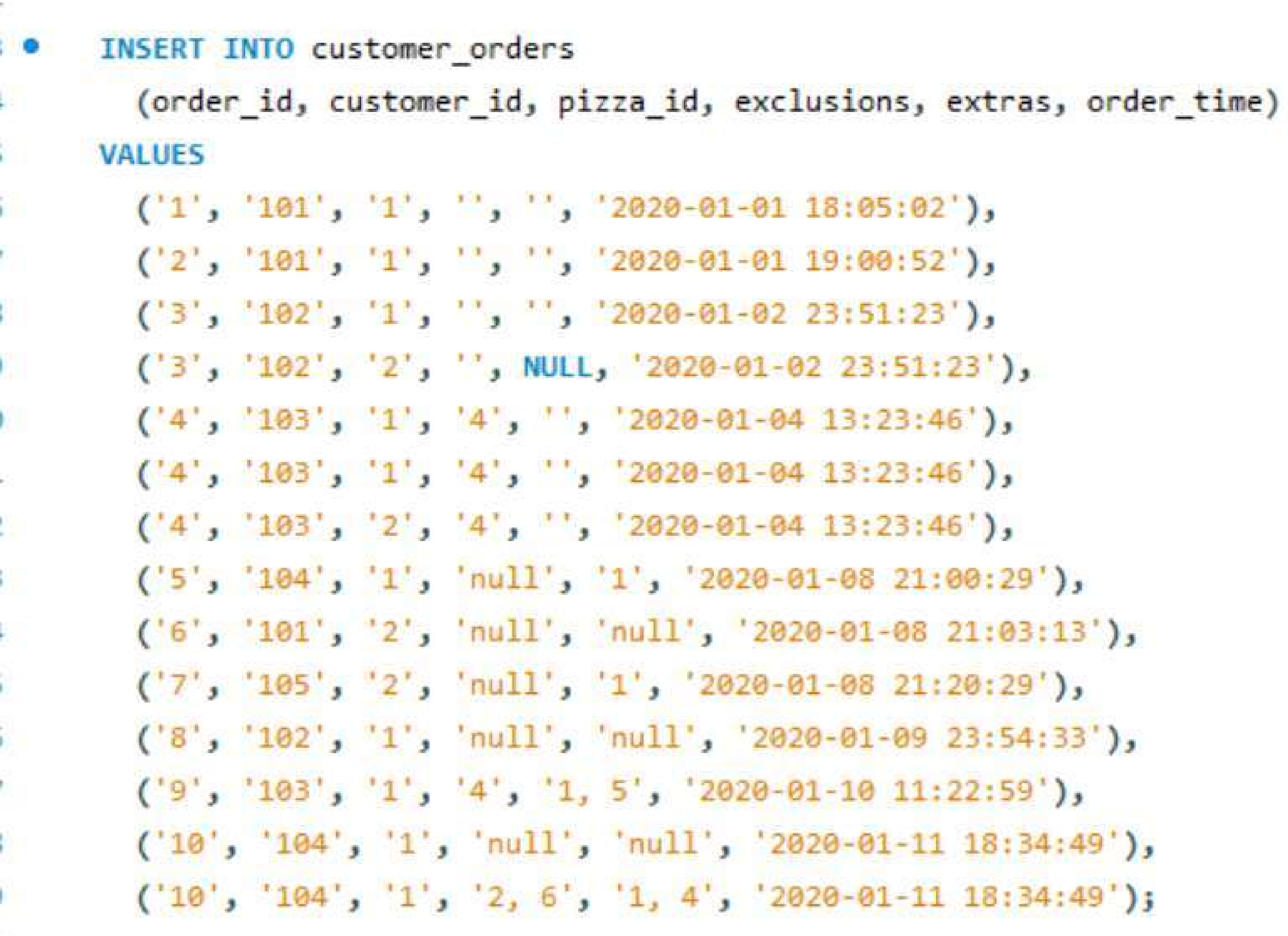


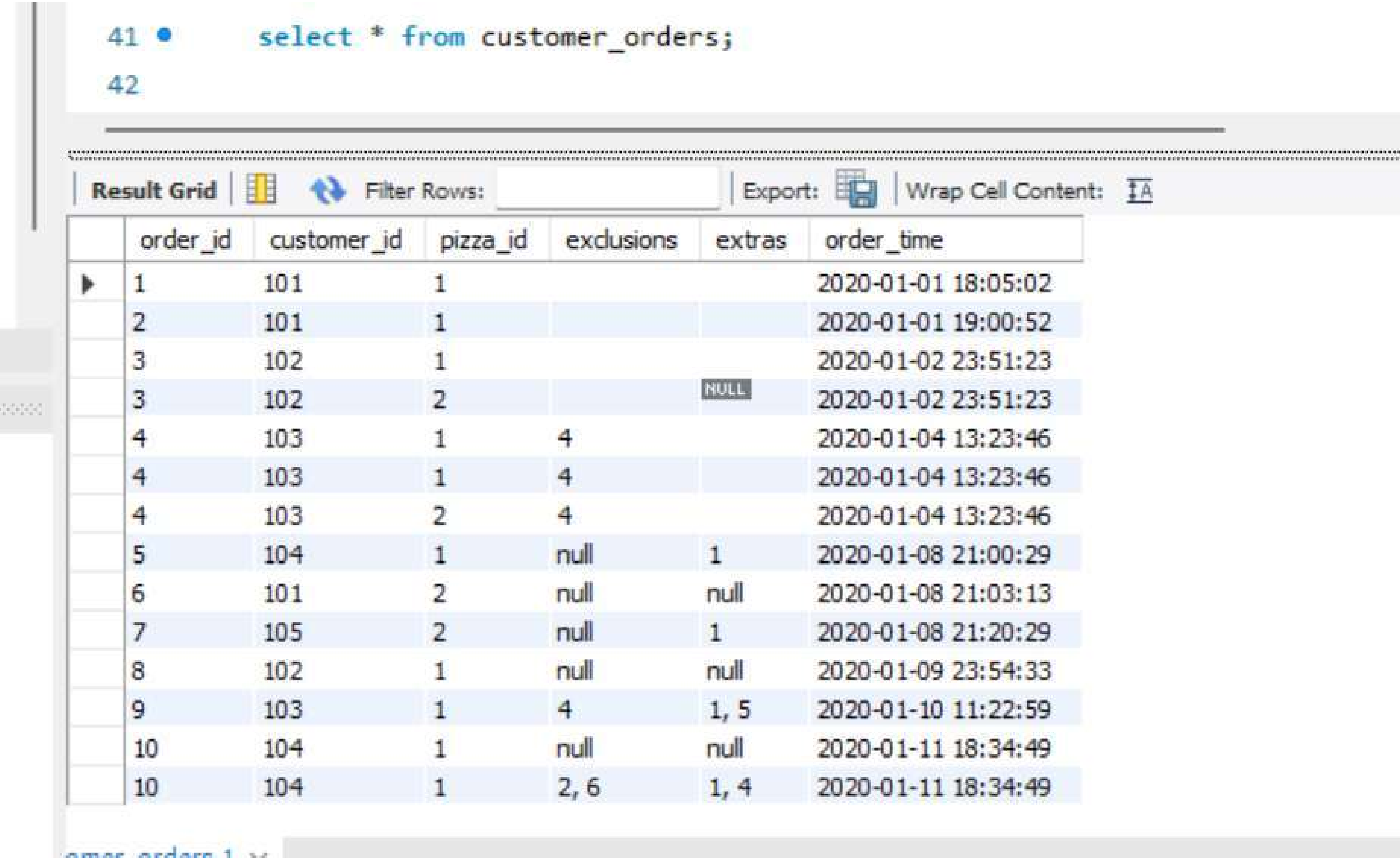
```sql
create database pizza_runner;
use pizza_runner;
create table runners(
runner_id int,
registration_date date);

insert into runners(runner_id, registration_date)
values
(1, '2021-01-01'),
(2, '2021-01-03'),
(3, '2021-01-08'),
(4, '2021-01-15');

```



```
create table customer_orders(
order_id int,
customer_id int,
pizza_id int,
exclusions varchar(4),
extras varchar(4),
order_time timestamp);
```

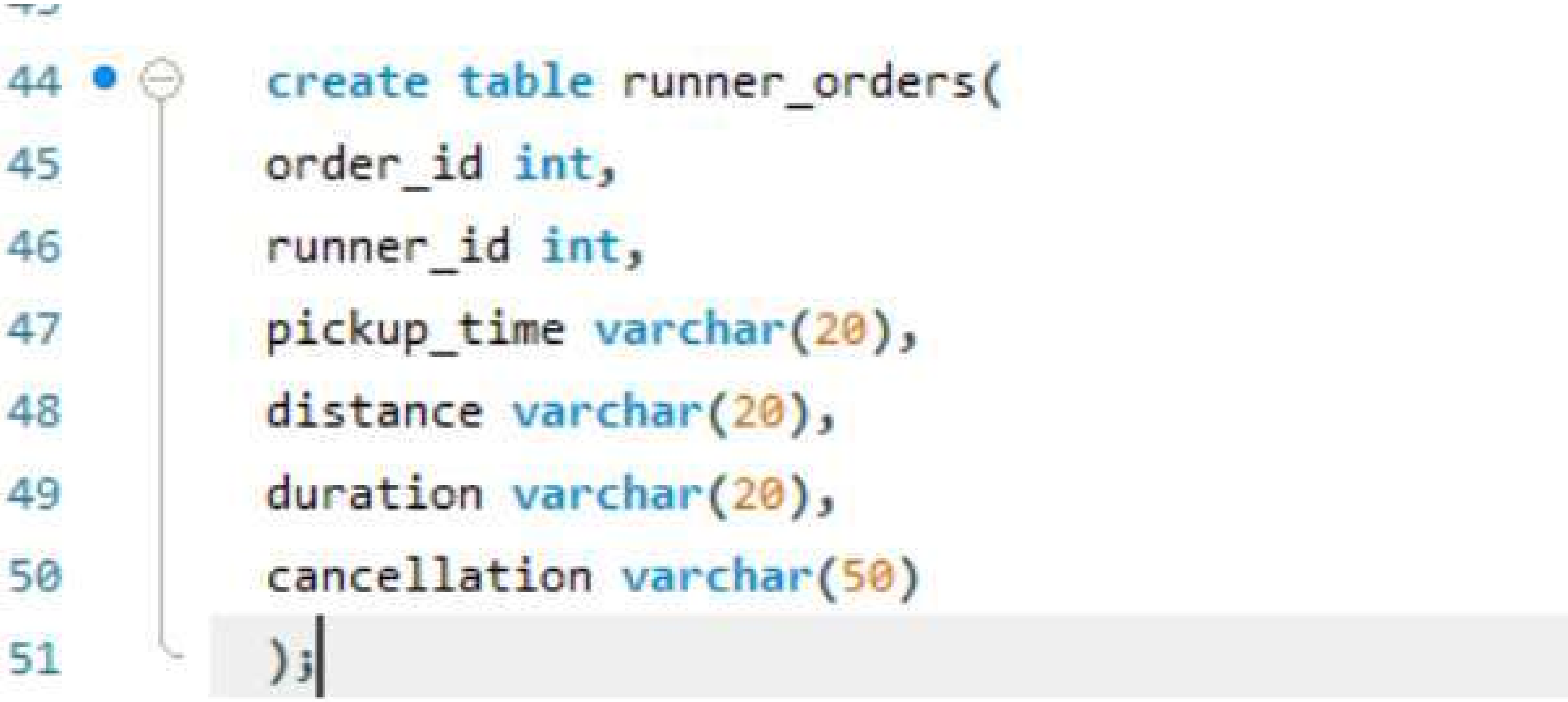


```sql
INSERT INTO customer_orders
  (order_id, customer_id, pizza_id, exclusions, extras, order_time)
VALUES
  ('1', '101', '1', '', '', '2020-01-01 18:05:02'),
  ('2', '101', '1', '', '', '2020-01-01 19:00:52'),
  ('3', '102', '1', '', '', '2020-01-02 23:51:23'),
  ('3', '102', '2', '', NULL, '2020-01-02 23:51:23'),
  ('4', '103', '1', '4', '', '2020-01-04 13:23:46'),
  ('4', '103', '1', '4', '', '2020-01-04 13:23:46'),
  ('4', '103', '2', '4', '', '2020-01-04 13:23:46'),
  ('5', '104', '1', 'null', '1', '2020-01-08 21:00:29'),
  ('6', '101', '2', 'null', 'null', '2020-01-08 21:03:13'),
  ('7', '105', '2', 'null', '1', '2020-01-08 21:20:29'),
  ('8', '102', '1', 'null', 'null', '2020-01-09 23:54:33'),
  ('9', '103', '1', '4', '1, 5', '2020-01-10 11:22:59'),
  ('10', '104', '1', 'null', 'null', '2020-01-11 18:34:49'),
  ('10', '104', '1', '2, 6', '1, 4', '2020-01-11 18:34:49');
```

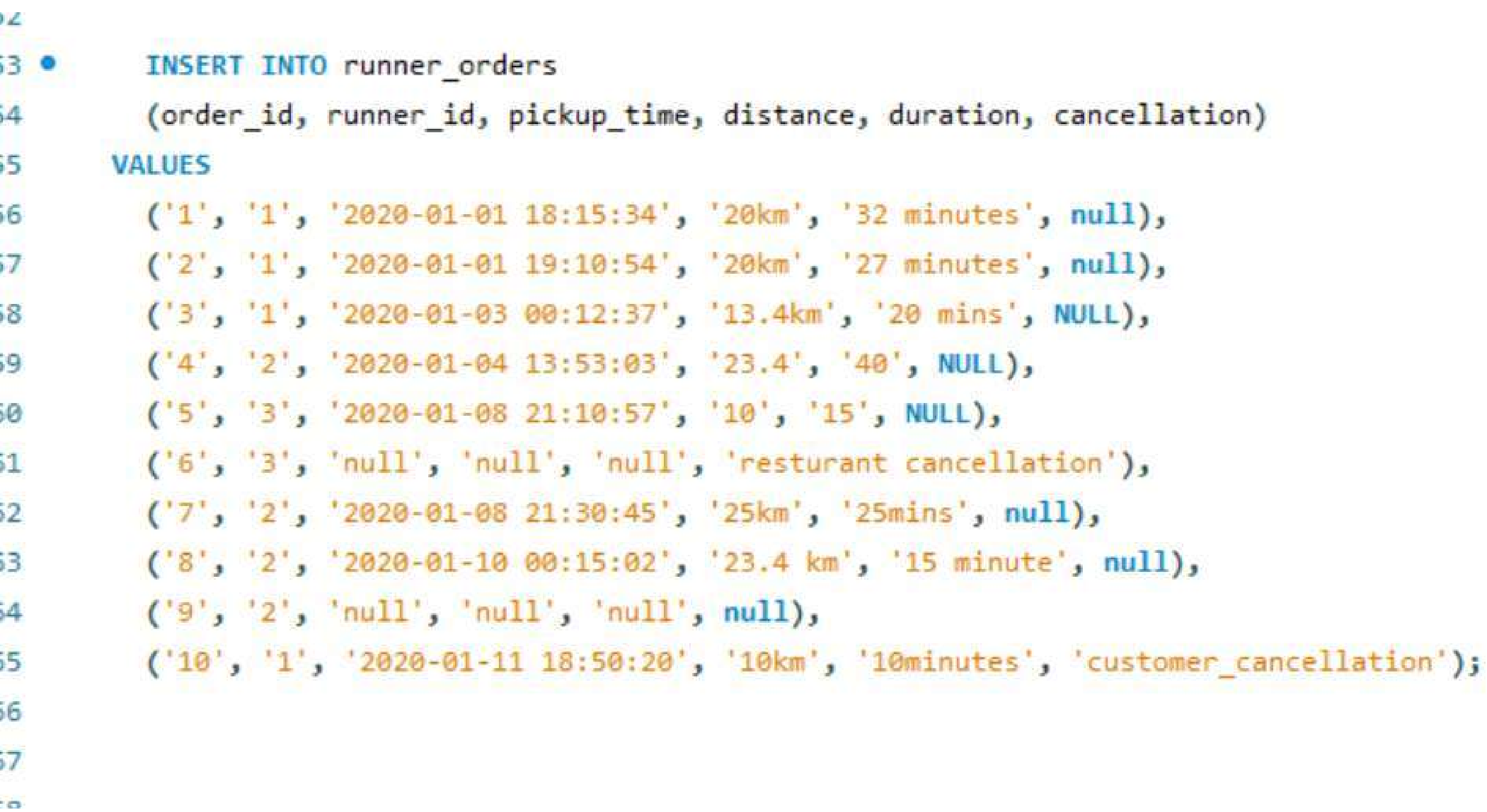


```sql
select * from customer_orders;
```

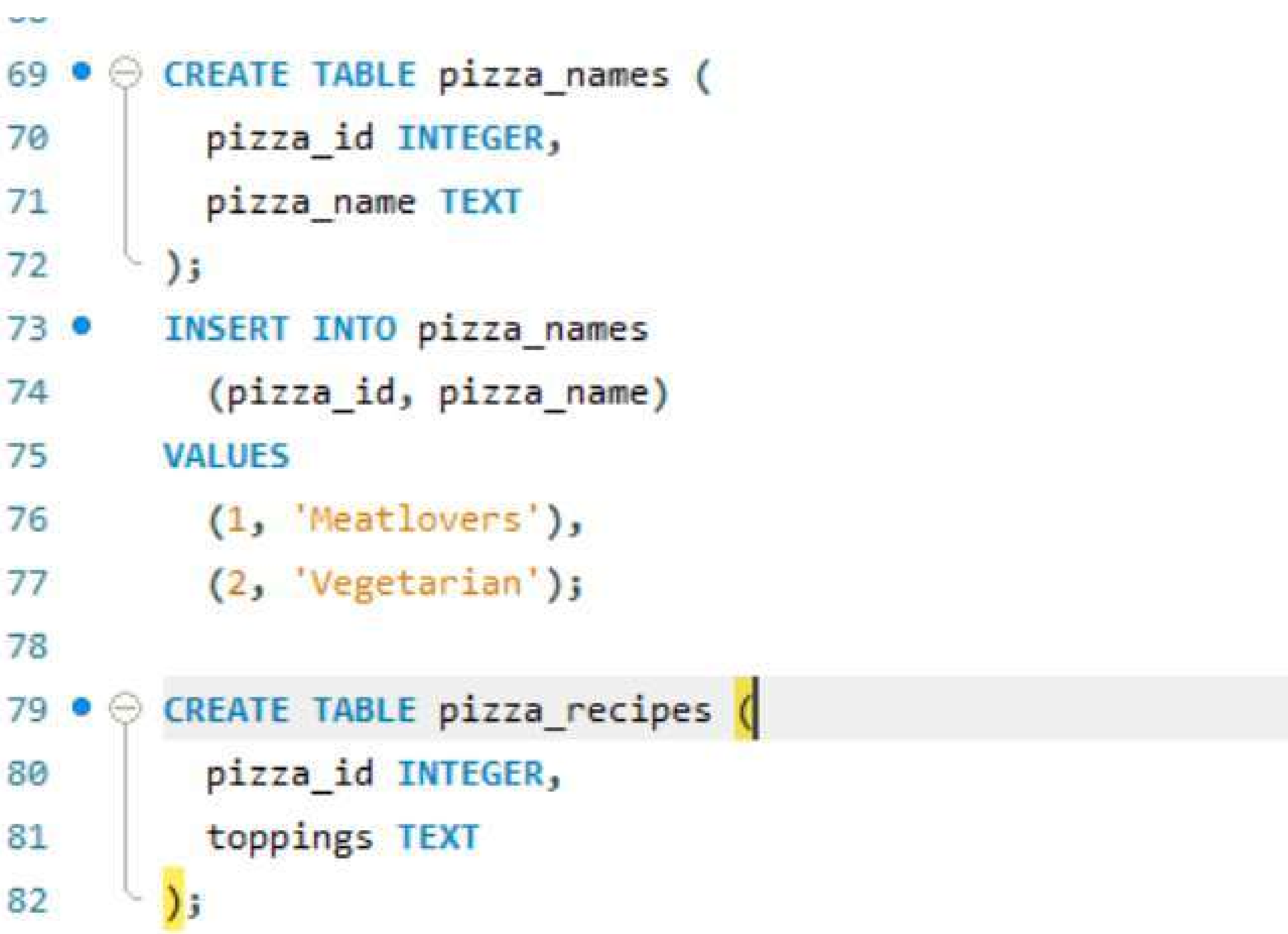
order_id	customer_id	pizza_id	exclusions	extras	order_time
1	101	1			2020-01-01 18:05:02
2	101	1			2020-01-01 19:00:52
3	102	1			2020-01-02 23:51:23
3	102	2		NULL	2020-01-02 23:51:23
4	103	1	4		2020-01-04 13:23:46
4	103	1	4		2020-01-04 13:23:46
4	103	2	4		2020-01-04 13:23:46
5	104	1	null	1	2020-01-08 21:00:29
6	101	2	null	null	2020-01-08 21:03:13
7	105	2	null	1	2020-01-08 21:20:29
8	102	1	null	null	2020-01-09 23:54:33
9	103	1	4	1, 5	2020-01-10 11:22:59
10	104	1	null	null	2020-01-11 18:34:49
10	104	1	2, 6	1, 4	2020-01-11 18:34:49



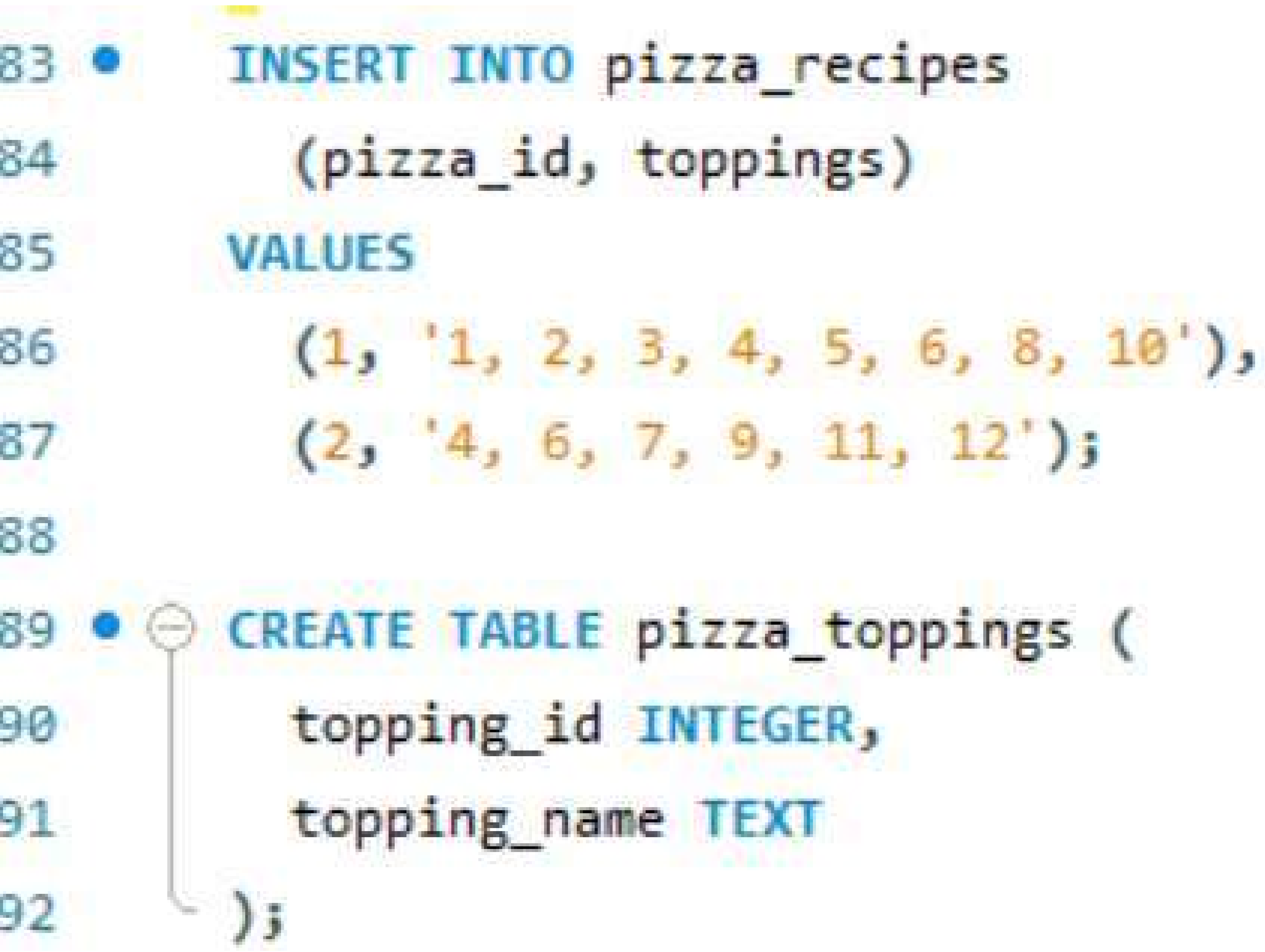
```sql
create table runner_orders(
order_id int,
runner_id int,
pickup_time varchar(20),
distance varchar(20),
duration varchar(20),
cancellation varchar(50)
);
```



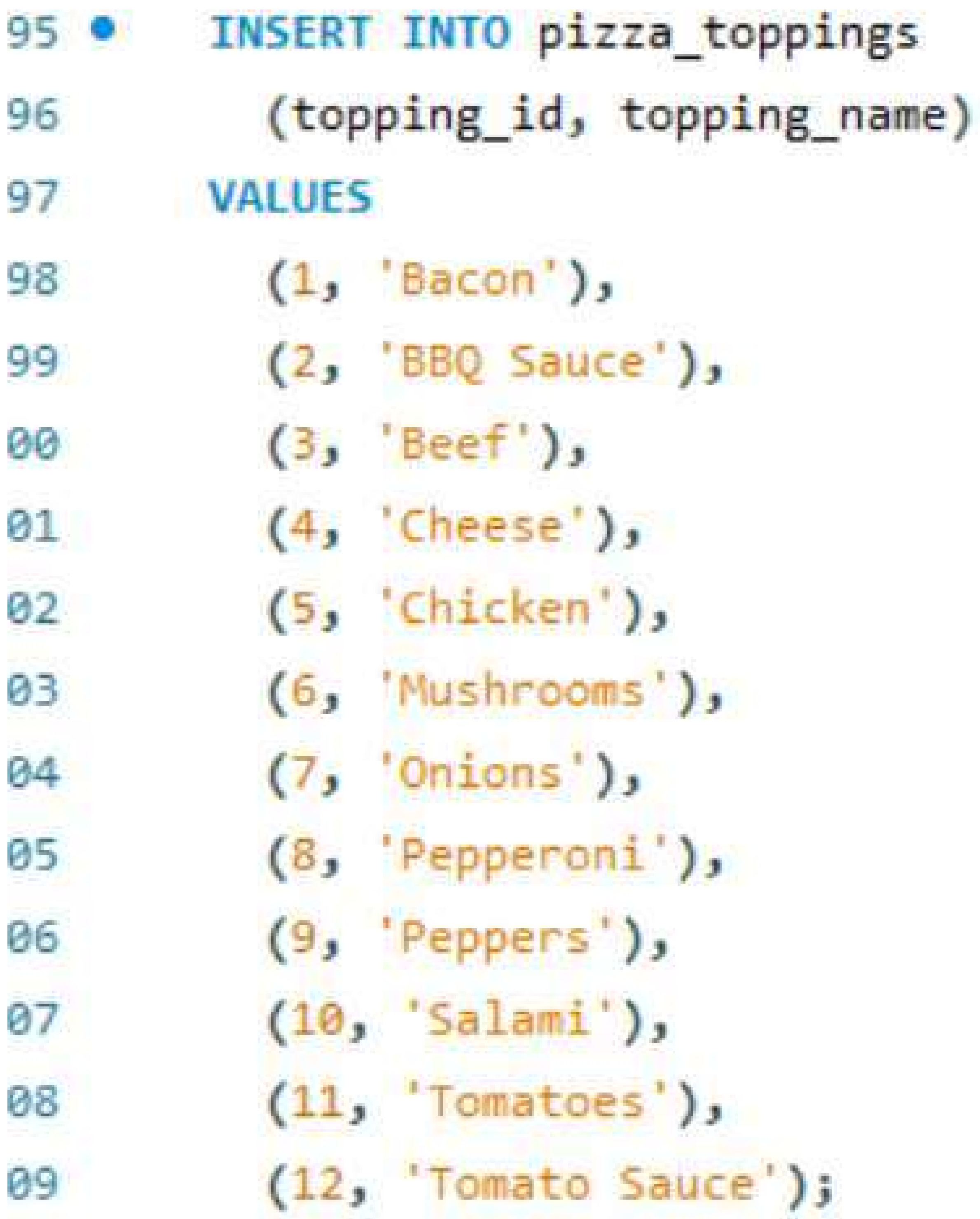
```sql
  INSERT INTO runner_orders
  (order_id, runner_id, pickup_time, distance, duration, cancellation)
VALUES
  ('1', '1', '2020-01-01 18:15:34', '20km', '32 minutes', null),
  ('2', '1', '2020-01-01 19:10:54', '20km', '27 minutes', null),
  ('3', '1', '2020-01-03 00:12:37', '13.4km', '20 mins', NULL),
  ('4', '2', '2020-01-04 13:53:03', '23.4', '40', NULL),
  ('5', '3', '2020-01-08 21:10:57', '10', '15', NULL),
  ('6', '3', 'null', 'null', 'null', 'resturant cancellation'),
  ('7', '2', '2020-01-08 21:30:45', '25km', '25mins', null),
  ('8', '2', '2020-01-10 00:15:02', '23.4 km', '15 minute', null),
  ('9', '2', 'null', 'null', 'null', null),
  ('10', '1', '2020-01-11 18:50:20', '10km', '10minutes', 'customer_cancellation');
```



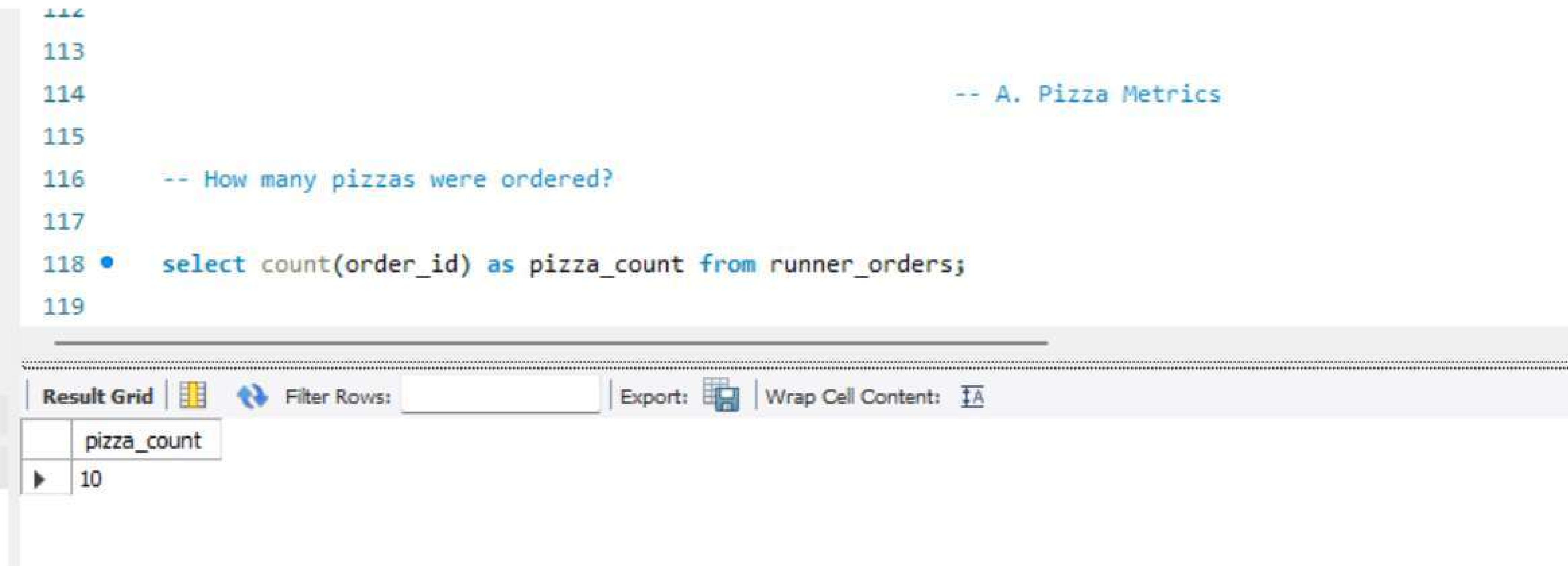
```sql
CREATE TABLE pizza_names (
  pizza_id INTEGER,
  pizza_name TEXT
);
INSERT INTO pizza_names
  (pizza_id, pizza_name)
VALUES
  (1, 'Meatlovers'),
  (2, 'Vegetarian');

CREATE TABLE pizza_recipes (
  pizza_id INTEGER,
  toppings TEXT
);
```



```
INSERT INTO pizza_recipes
  (pizza_id, toppings)
VALUES
  (1, '1, 2, 3, 4, 5, 6, 8, 10'),
  (2, '4, 6, 7, 9, 11, 12');

CREATE TABLE pizza_toppings (
  topping_id INTEGER,
  topping_name TEXT
);
```

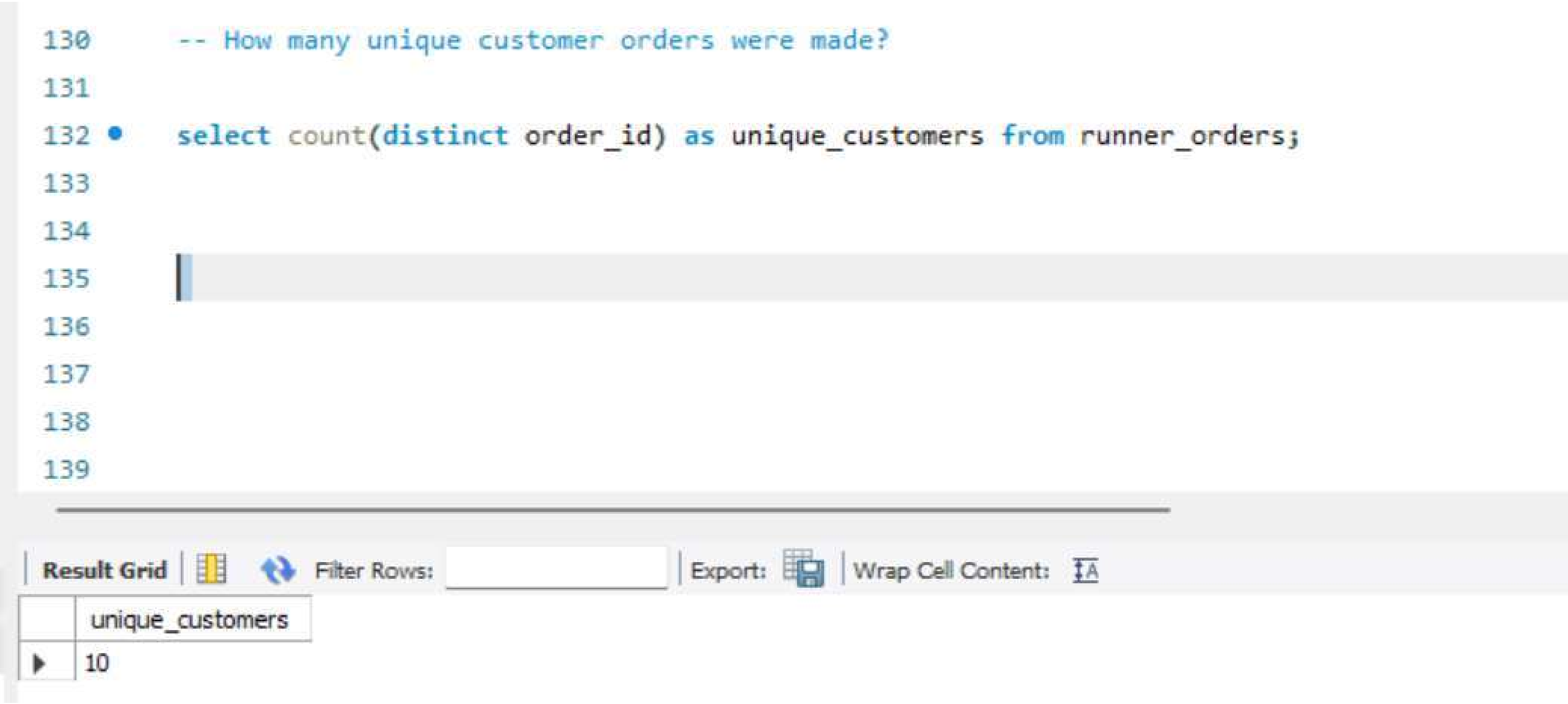


```sql
INSERT INTO pizza_toppings
  (topping_id, topping_name)
VALUES
  (1, 'Bacon'),
  (2, 'BBQ Sauce'),
  (3, 'Beef'),
  (4, 'Cheese'),
  (5, 'Chicken'),
  (6, 'Mushrooms'),
  (7, 'Onions'),
  (8, 'Pepperoni'),
  (9, 'Peppers'),
  (10, 'Salami'),
  (11, 'Tomatoes'),
  (12, 'Tomato Sauce');
```



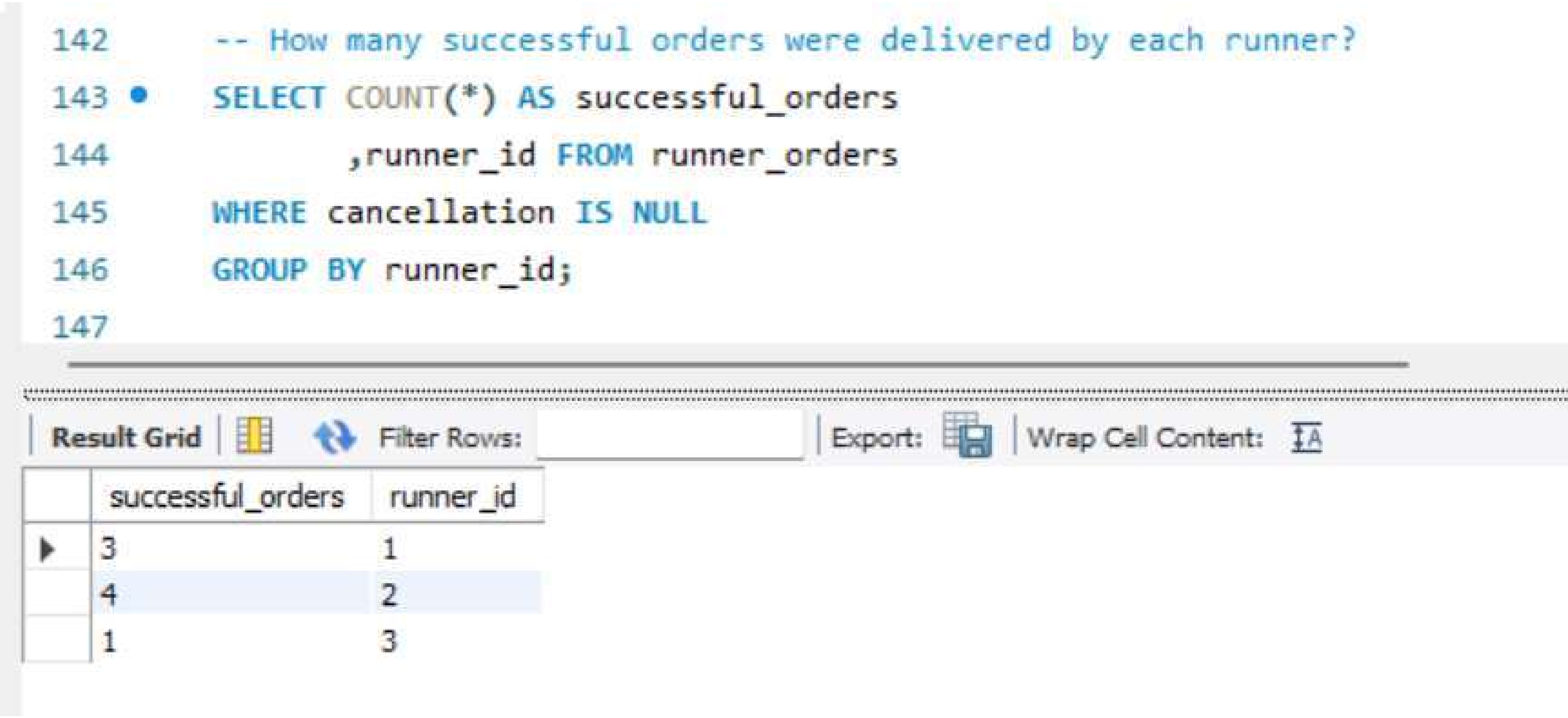
```
                                                          -- A. Pizza Metrics

-- How many pizzas were ordered?

select count(order_id) as pizza_count from runner_orders;
```

pizza_count
10



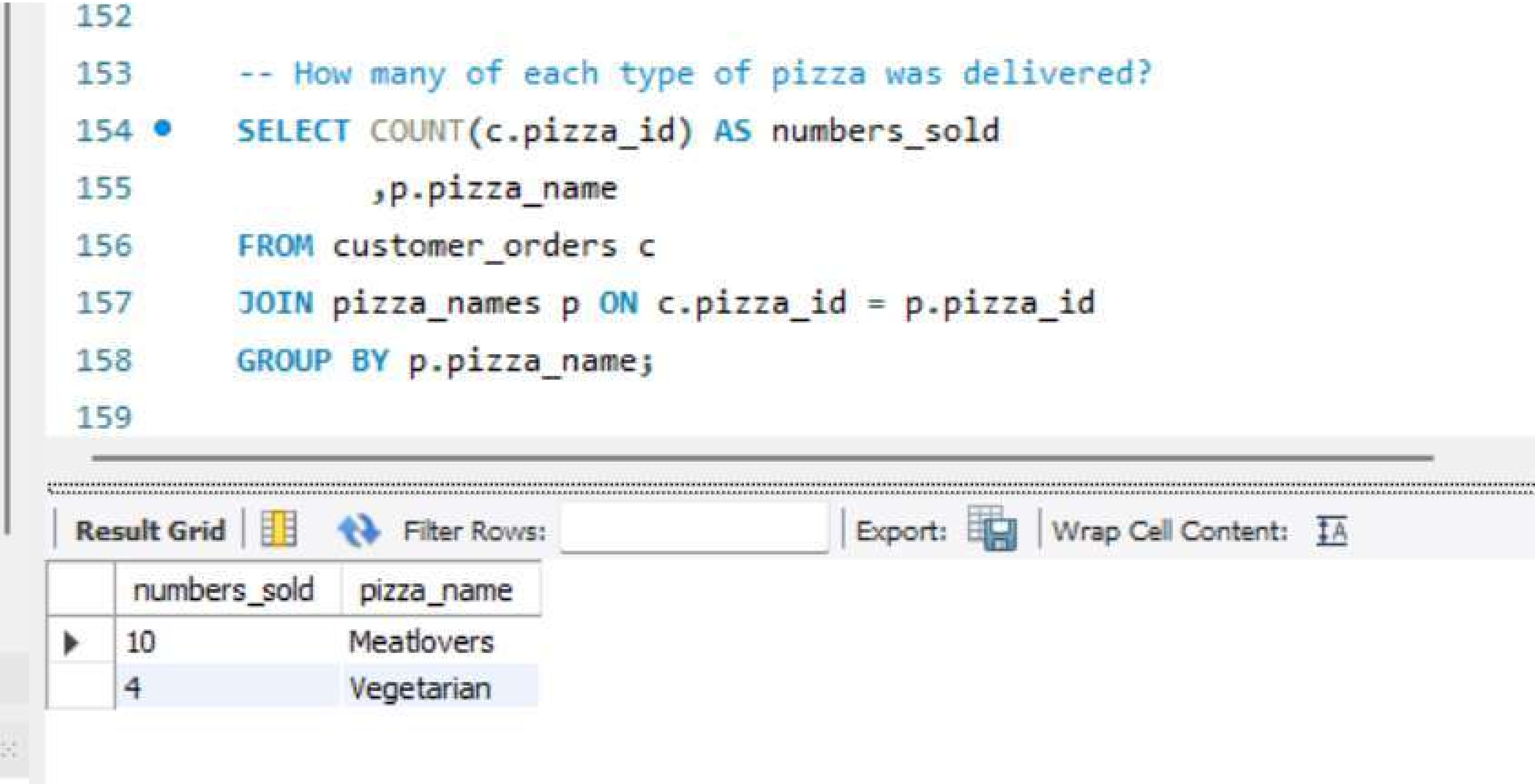
```sql
-- How many unique customer orders were made?

select count(distinct order_id) as unique_customers from runner_orders;
```

```
unique_customers
10
```



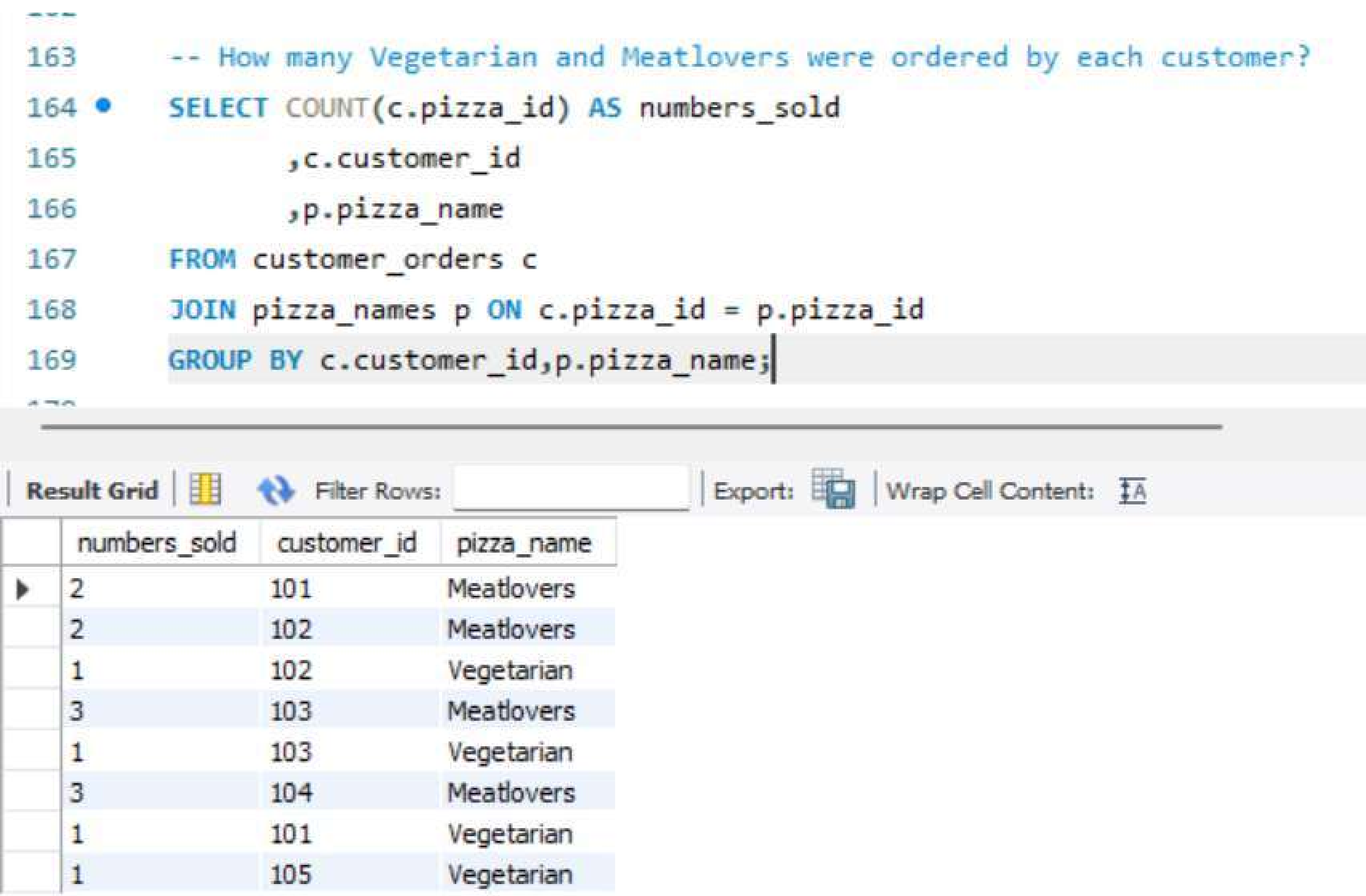
```sql
-- How many successful orders were delivered by each runner?
SELECT COUNT(*) AS successful_orders
       ,runner_id FROM runner_orders
WHERE cancellation IS NULL
GROUP BY runner_id;
```

successful_orders	runner_id
3	1
4	2
1	3



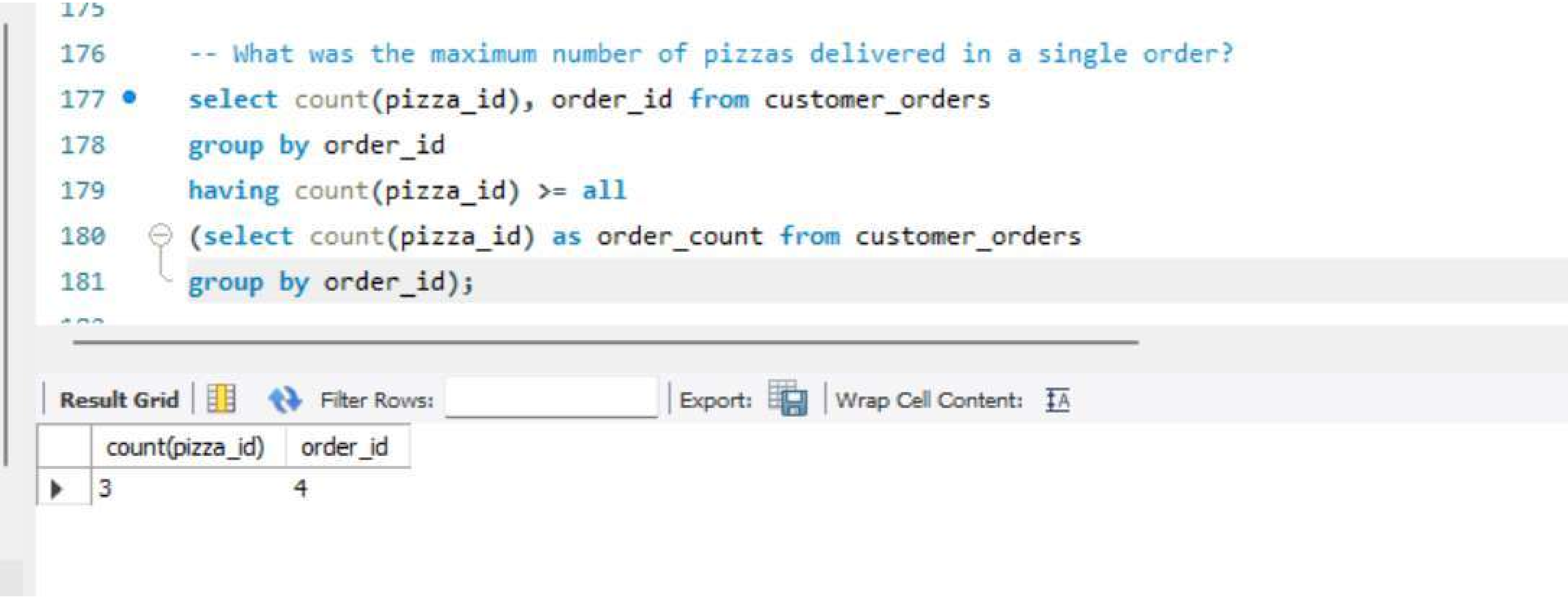
```sql

-- How many of each type of pizza was delivered?
SELECT COUNT(c.pizza_id) AS numbers_sold
       ,p.pizza_name
FROM customer_orders c
JOIN pizza_names p ON c.pizza_id = p.pizza_id
GROUP BY p.pizza_name;

```

numbers_sold	pizza_name
10	Meatlovers
4	Vegetarian



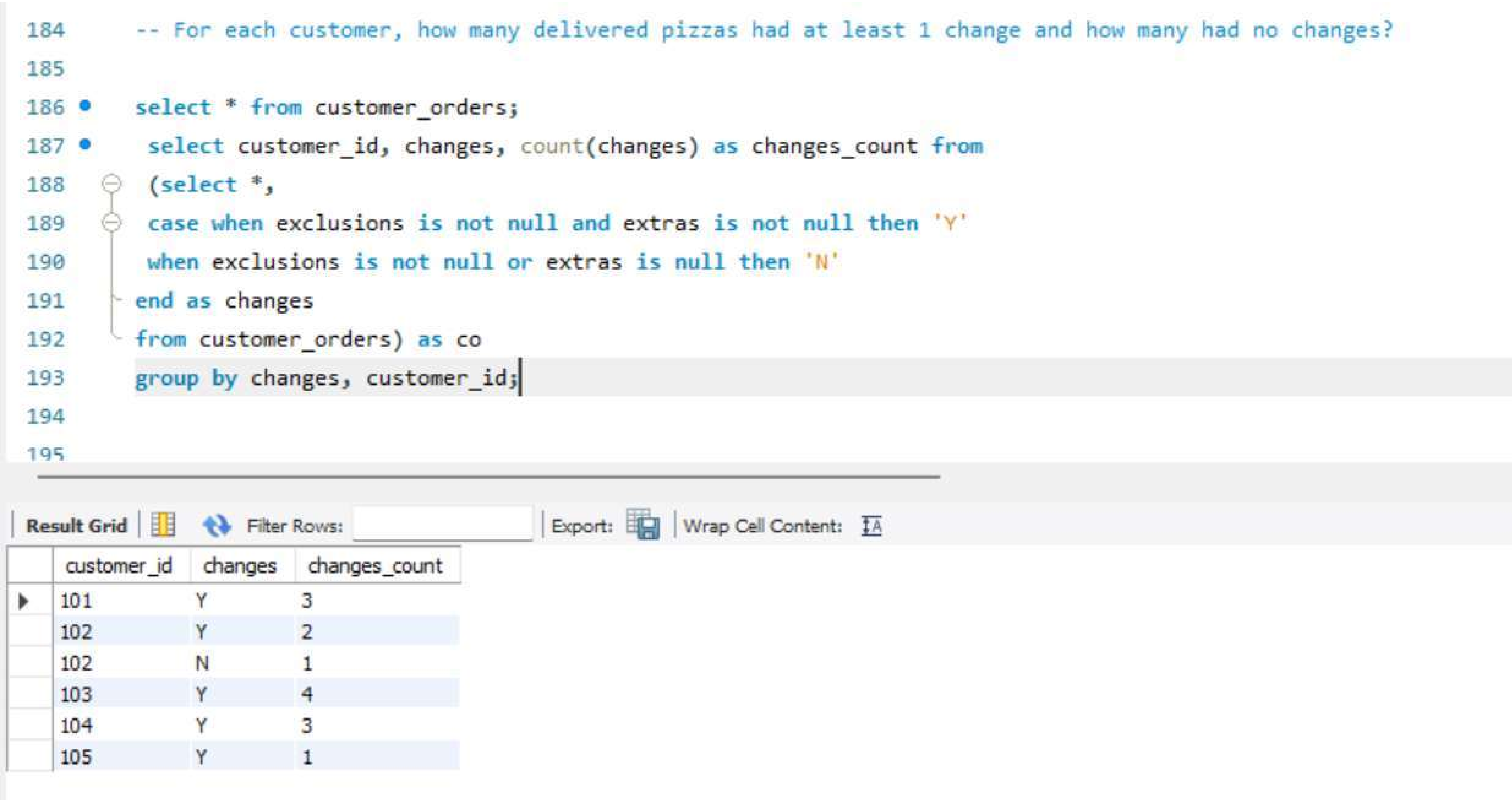
```sql
-- How many Vegetarian and Meatlovers were ordered by each customer?
SELECT COUNT(c.pizza_id) AS numbers_sold
       ,c.customer_id
       ,p.pizza_name
FROM customer_orders c
JOIN pizza_names p ON c.pizza_id = p.pizza_id
GROUP BY c.customer_id,p.pizza_name;
```

numbers_sold	customer_id	pizza_name
2	101	Meatlovers
2	102	Meatlovers
1	102	Vegetarian
3	103	Meatlovers
1	103	Vegetarian
3	104	Meatlovers
1	101	Vegetarian
1	105	Vegetarian



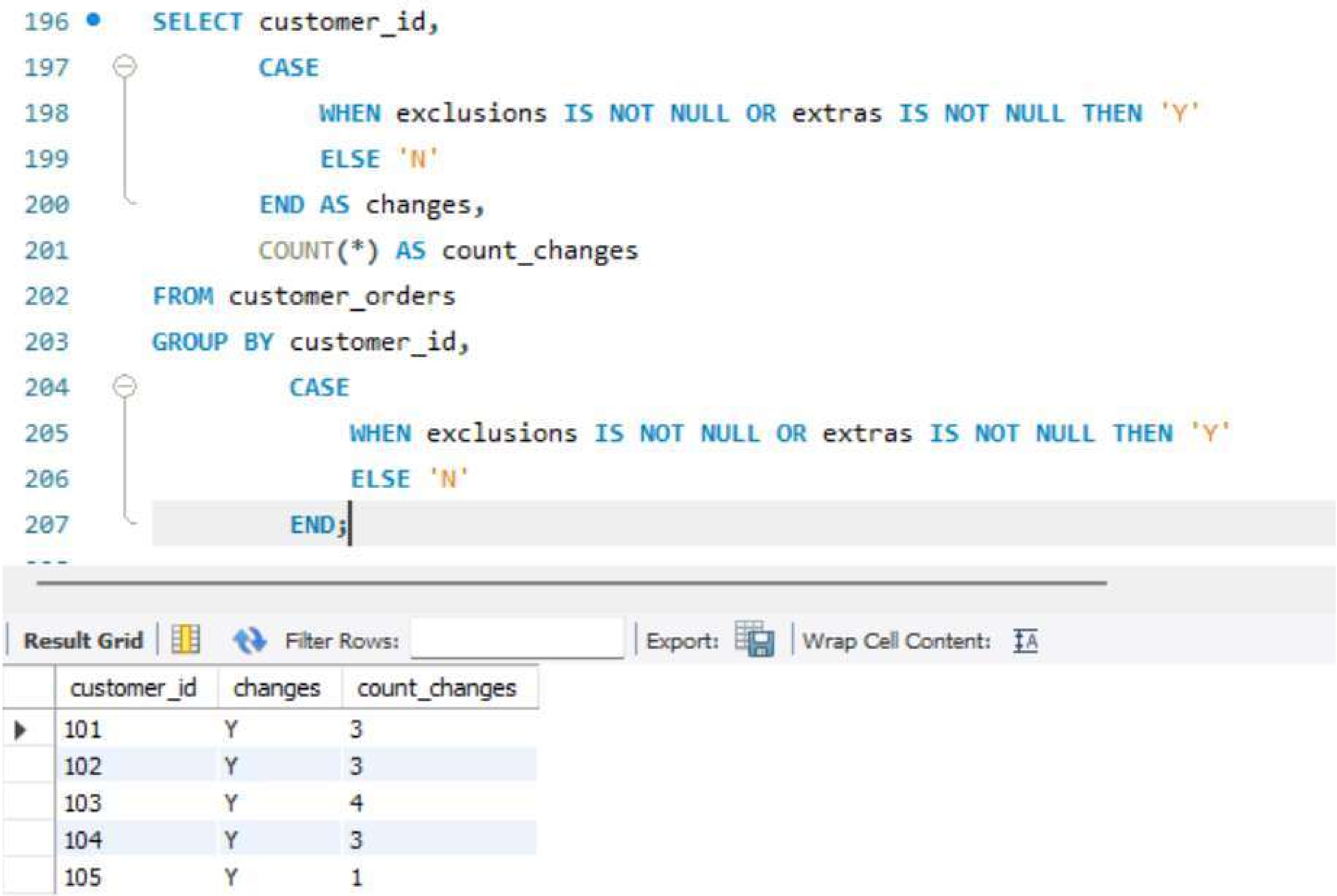
```
-- What was the maximum number of pizzas delivered in a single order?
select count(pizza_id), order_id from customer_orders
group by order_id
having count(pizza_id) >= all
(select count(pizza_id) as order_count from customer_orders
group by order_id);
```

count(pizza_id)	order_id
3	4



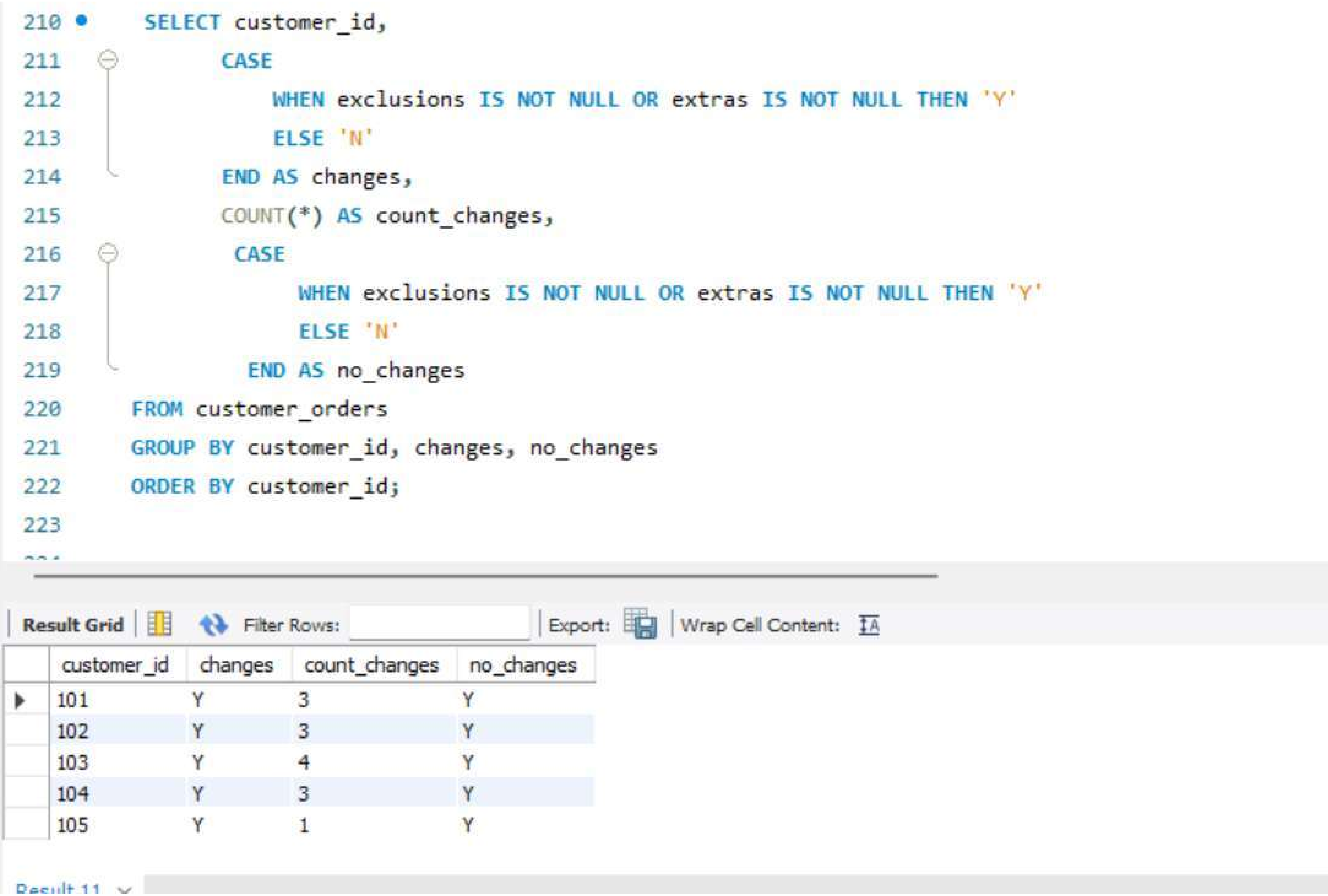
```sql
-- For each customer, how many delivered pizzas had at least 1 change and how many had no changes?

select * from customer_orders;
 select customer_id, changes, count(changes) as changes_count from
 (select *,
 case when exclusions is not null and extras is not null then 'Y'
 when exclusions is not null or extras is null then 'N'
end as changes
from customer_orders) as co
group by changes, customer_id;
```

customer_id	changes	changes_count
101	Y	3
102	Y	2
102	N	1
103	Y	4
104	Y	3
105	Y	1



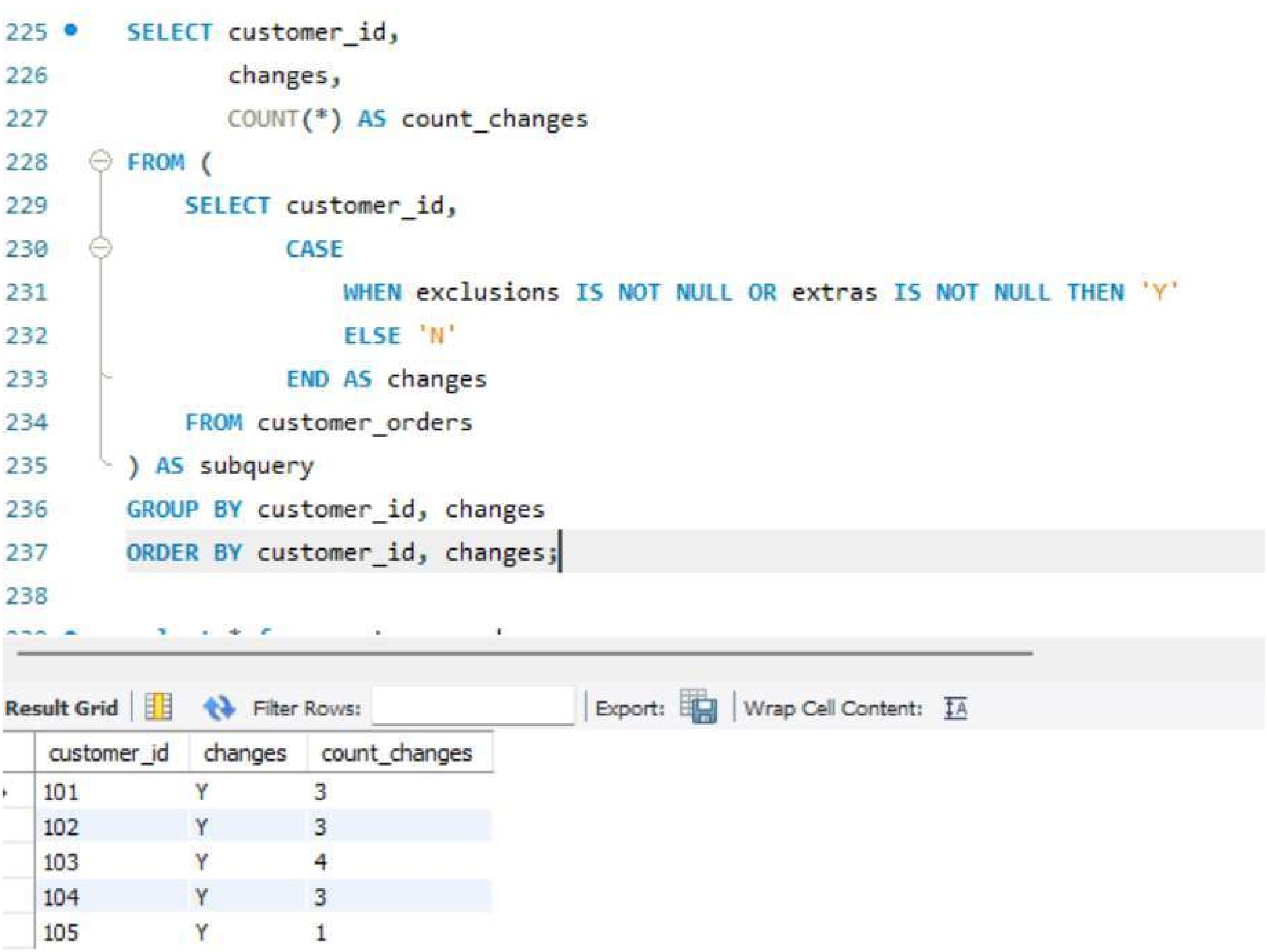
```sql
SELECT customer_id,
       CASE
           WHEN exclusions IS NOT NULL OR extras IS NOT NULL THEN 'Y'
           ELSE 'N'
       END AS changes,
       COUNT(*) AS count_changes
FROM customer_orders
GROUP BY customer_id,
         CASE
             WHEN exclusions IS NOT NULL OR extras IS NOT NULL THEN 'Y'
             ELSE 'N'
         END;
```

customer_id	changes	count_changes
101	Y	3
102	Y	3
103	Y	4
104	Y	3
105	Y	1



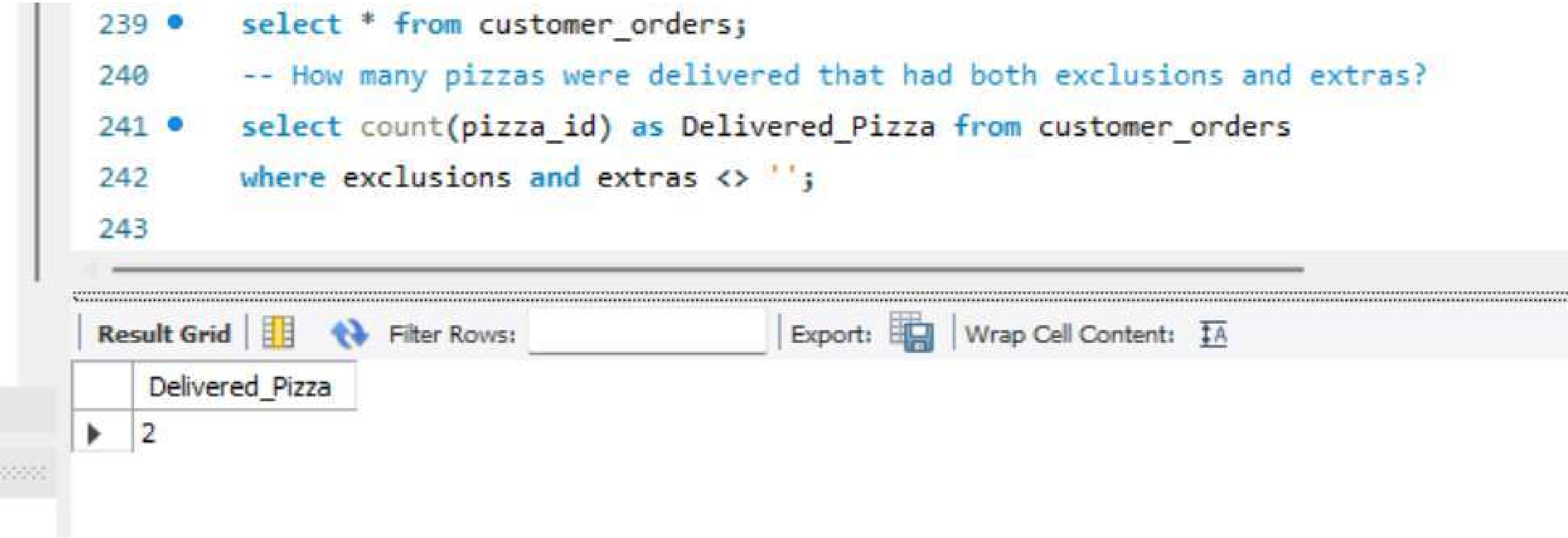
```sql
 SELECT customer_id,
       CASE
           WHEN exclusions IS NOT NULL OR extras IS NOT NULL THEN 'Y'
           ELSE 'N'
       END AS changes,
       COUNT(*) AS count_changes,
        CASE
             WHEN exclusions IS NOT NULL OR extras IS NOT NULL THEN 'Y'
             ELSE 'N'
         END AS no_changes
FROM customer_orders
GROUP BY customer_id, changes, no_changes
ORDER BY customer_id;
```

customer_id	changes	count_changes	no_changes
101	Y	3	Y
102	Y	3	Y
103	Y	4	Y
104	Y	3	Y
105	Y	1	Y



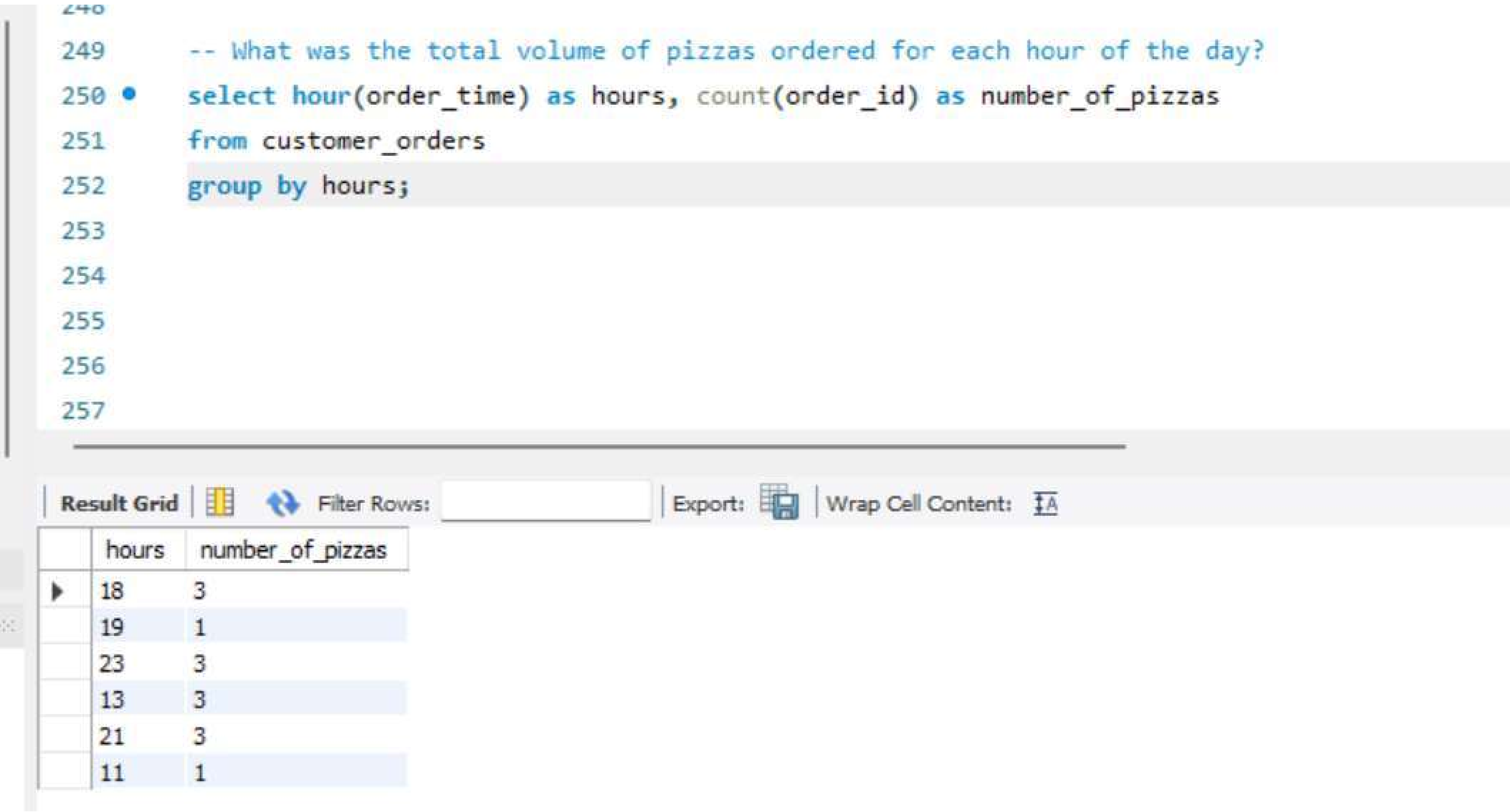
```sql
SELECT customer_id,
       changes,
       COUNT(*) AS count_changes
FROM (
    SELECT customer_id,
           CASE
               WHEN exclusions IS NOT NULL OR extras IS NOT NULL THEN 'Y'
               ELSE 'N'
           END AS changes
    FROM customer_orders
) AS subquery
GROUP BY customer_id, changes
ORDER BY customer_id, changes;
```

customer_id	changes	count_changes
101	Y	3
102	Y	3
103	Y	4
104	Y	3
105	Y	1



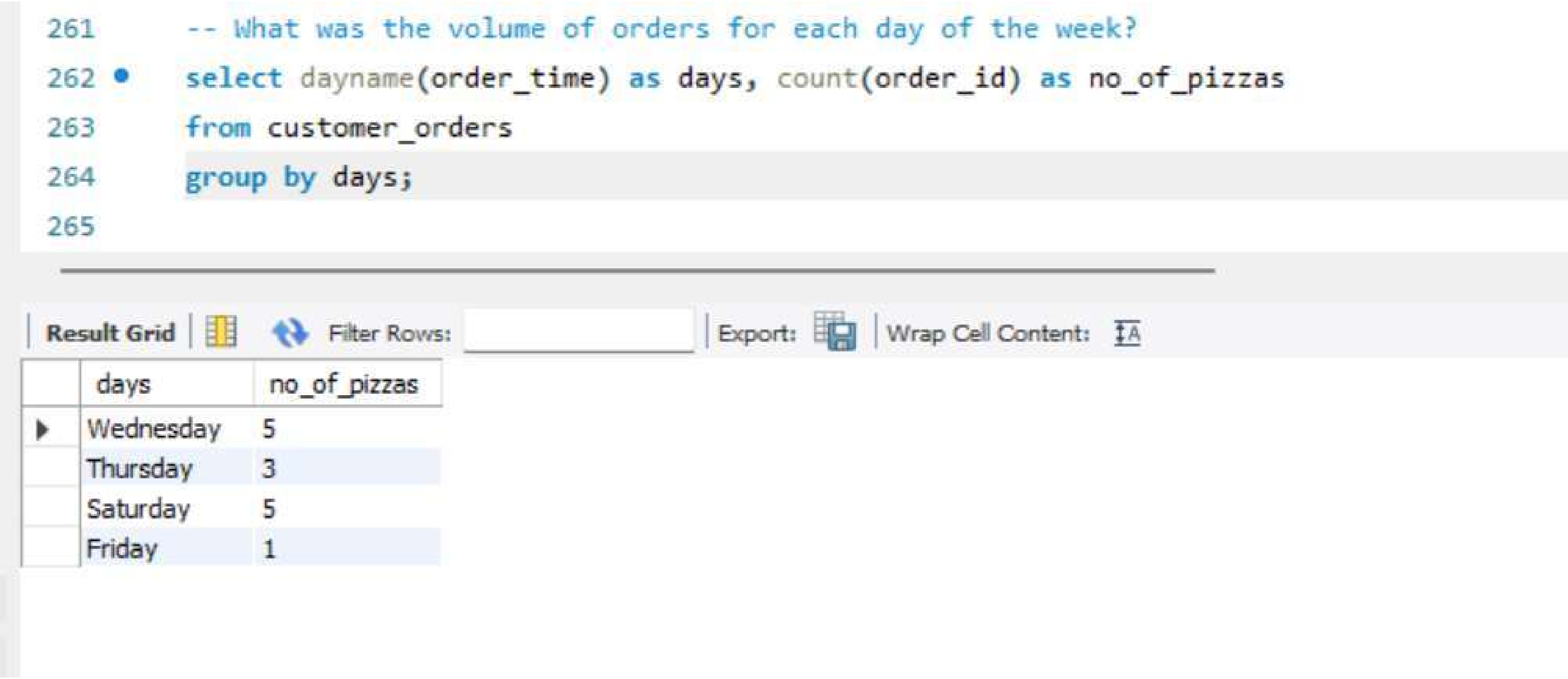
```sql
select * from customer_orders;
-- How many pizzas were delivered that had both exclusions and extras?
select count(pizza_id) as Delivered_Pizza from customer_orders
where exclusions and extras <> '';
```

```
Delivered_Pizza
2
```



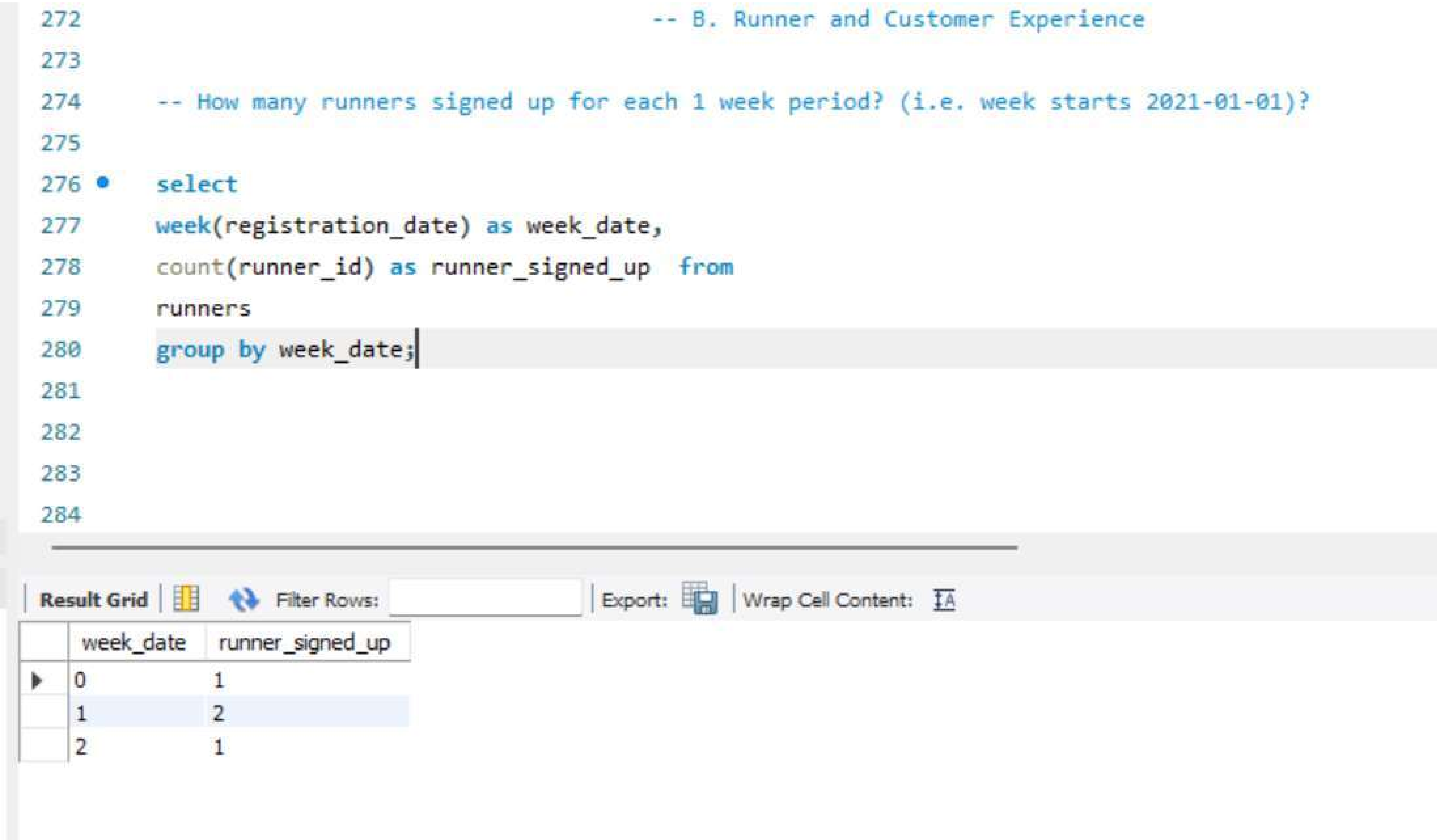
```sql
-- What was the total volume of pizzas ordered for each hour of the day?
select hour(order_time) as hours, count(order_id) as number_of_pizzas
from customer_orders
group by hours;
```

hours	number_of_pizzas
18	3
19	1
23	3
13	3
21	3
11	1



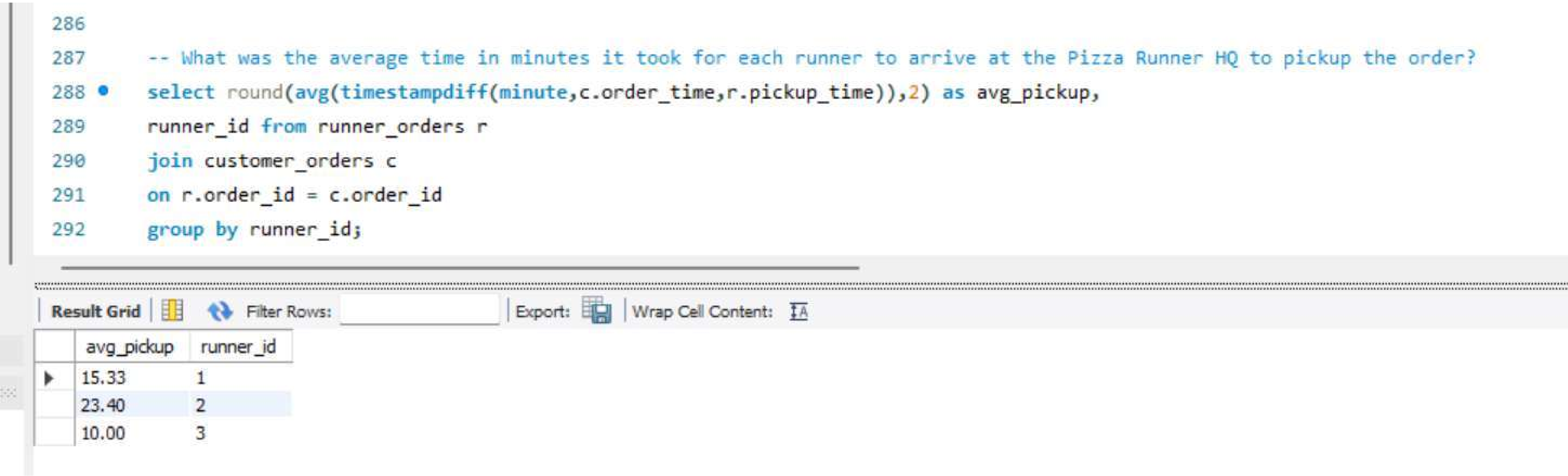
```sql
-- What was the volume of orders for each day of the week?
select dayname(order_time) as days, count(order_id) as no_of_pizzas
from customer_orders
group by days;

```

days	no_of_pizzas
Wednesday	5
Thursday	3
Saturday	5
Friday	1



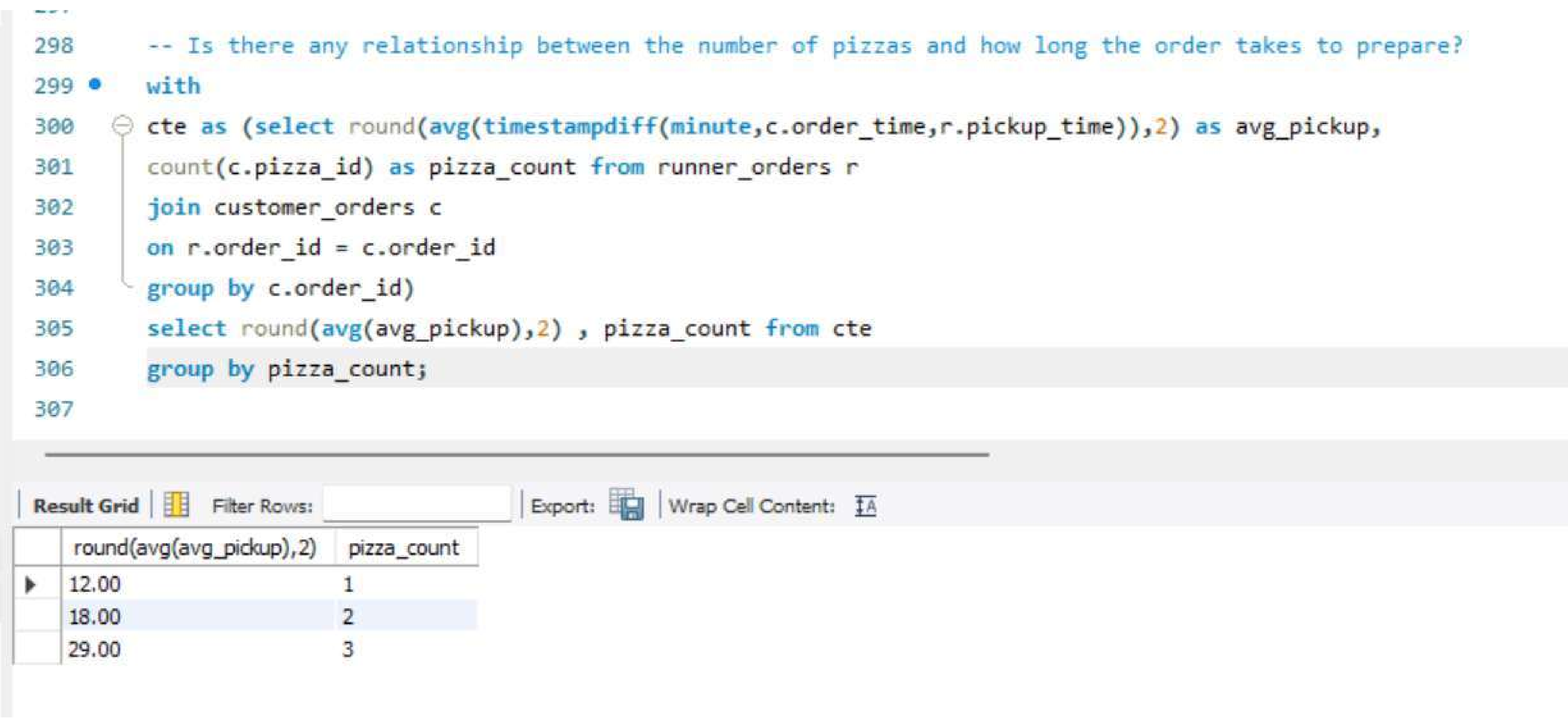
```sql
                                    -- B. Runner and Customer Experience

-- How many runners signed up for each 1 week period? (i.e. week starts 2021-01-01)?

select
week(registration_date) as week_date,
count(runner_id) as runner_signed_up  from
runners
group by week_date;
```

week_date	runner_signed_up
0	1
1	2
2	1



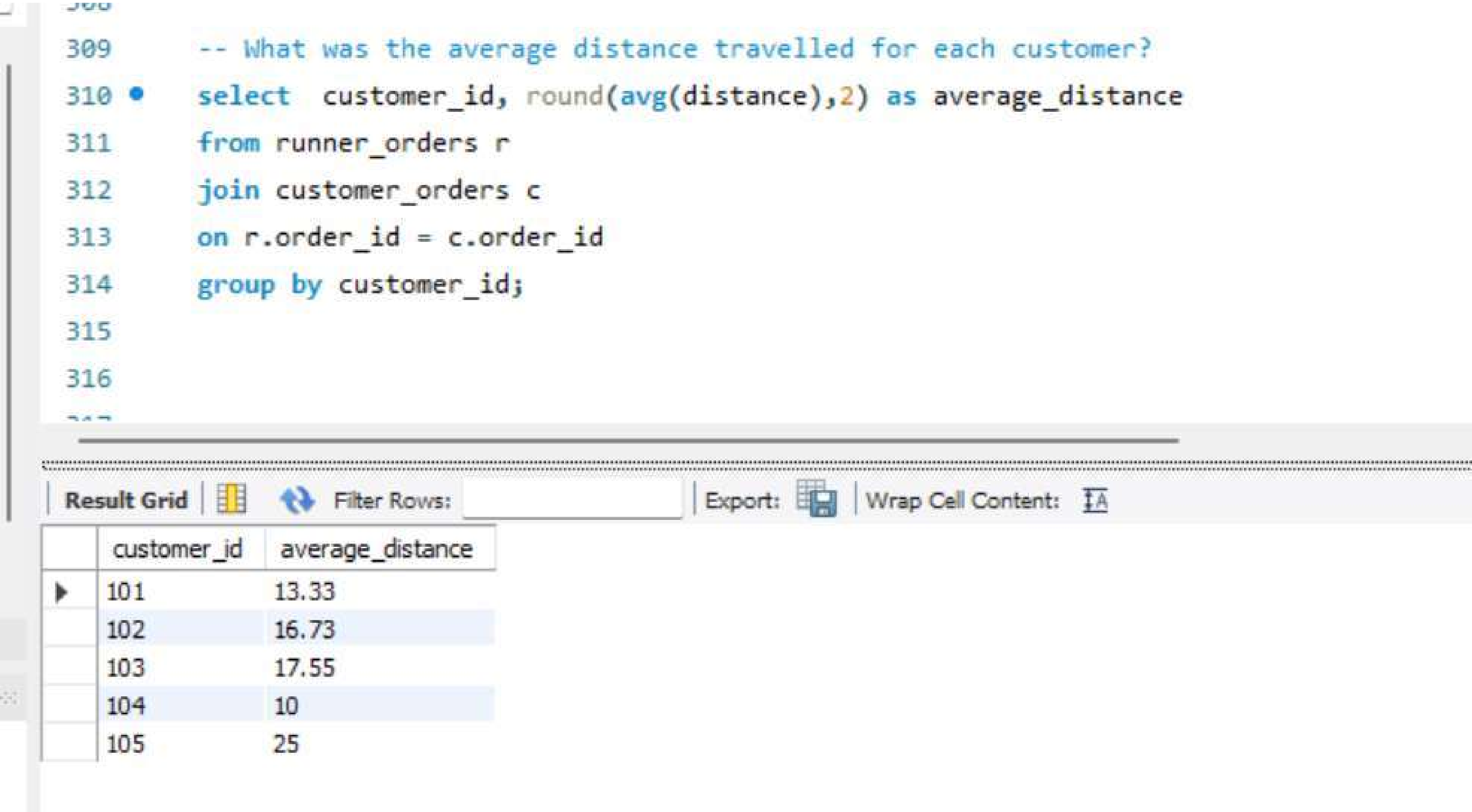
```sql

-- What was the average time in minutes it took for each runner to arrive at the Pizza Runner HQ to pickup the order?
select round(avg(timestampdiff(minute,c.order_time,r.pickup_time)),2) as avg_pickup,
runner_id from runner_orders r
join customer_orders c
on r.order_id = c.order_id
group by runner_id;
```

avg_pickup	runner_id
15.33	1
23.40	2
10.00	3



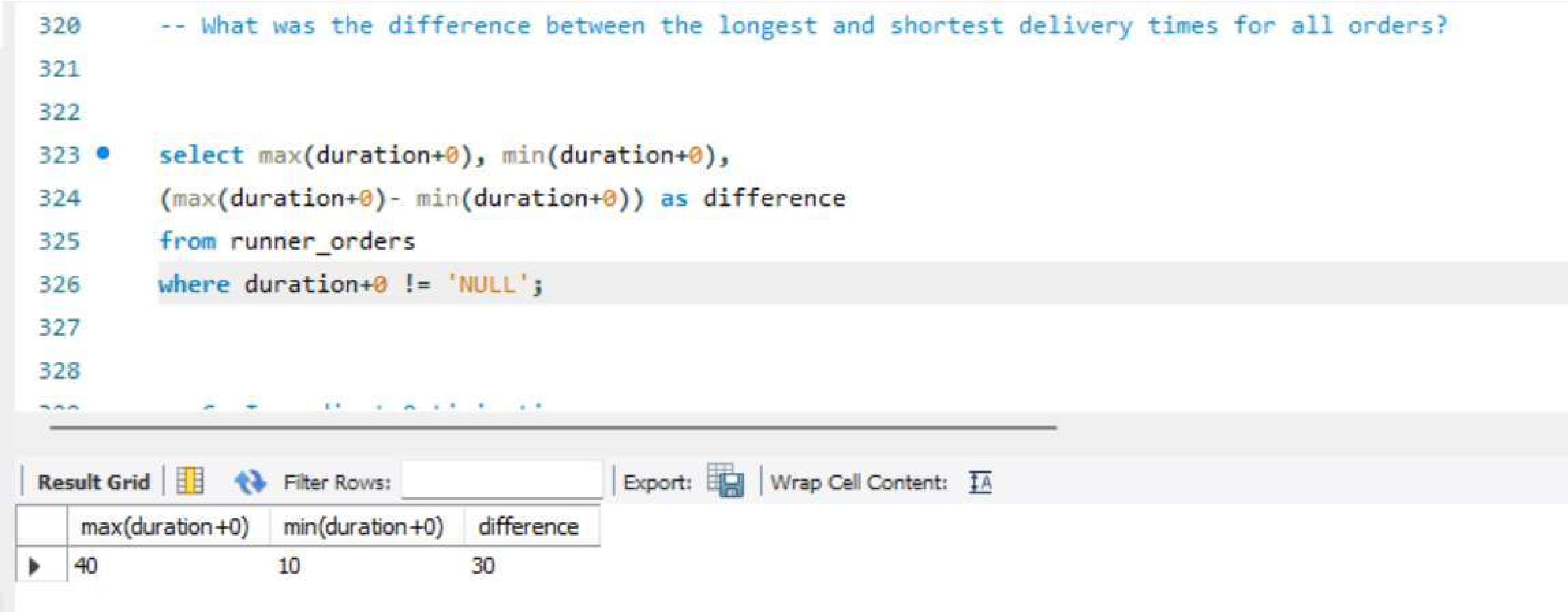
```sql
-- Is there any relationship between the number of pizzas and how long the order takes to prepare?
with
cte as (select round(avg(timestampdiff(minute,c.order_time,r.pickup_time)),2) as avg_pickup,
count(c.pizza_id) as pizza_count from runner_orders r
join customer_orders c
on r.order_id = c.order_id
group by c.order_id)
select round(avg(avg_pickup),2) , pizza_count from cte
group by pizza_count;
```

round(avg(avg_pickup),2)	pizza_count
12.00	1
18.00	2
29.00	3



```
-- What was the average distance travelled for each customer?
select  customer_id, round(avg(distance),2) as average_distance
from runner_orders r
join customer_orders c
on r.order_id = c.order_id
group by customer_id;
```

customer_id	average_distance
101	13.33
102	16.73
103	17.55
104	10
105	25



```
-- What was the difference between the longest and shortest delivery times for all orders?


select max(duration+0), min(duration+0),
(max(duration+0)- min(duration+0)) as difference
from runner_orders
where duration+0 != 'NULL';
```

max(duration+0)	min(duration+0)	difference
40	10	30