SQL CASE STUDY

Pooja Agrawal

Table 1: runners

The runners table shows the registration_date for each new runner

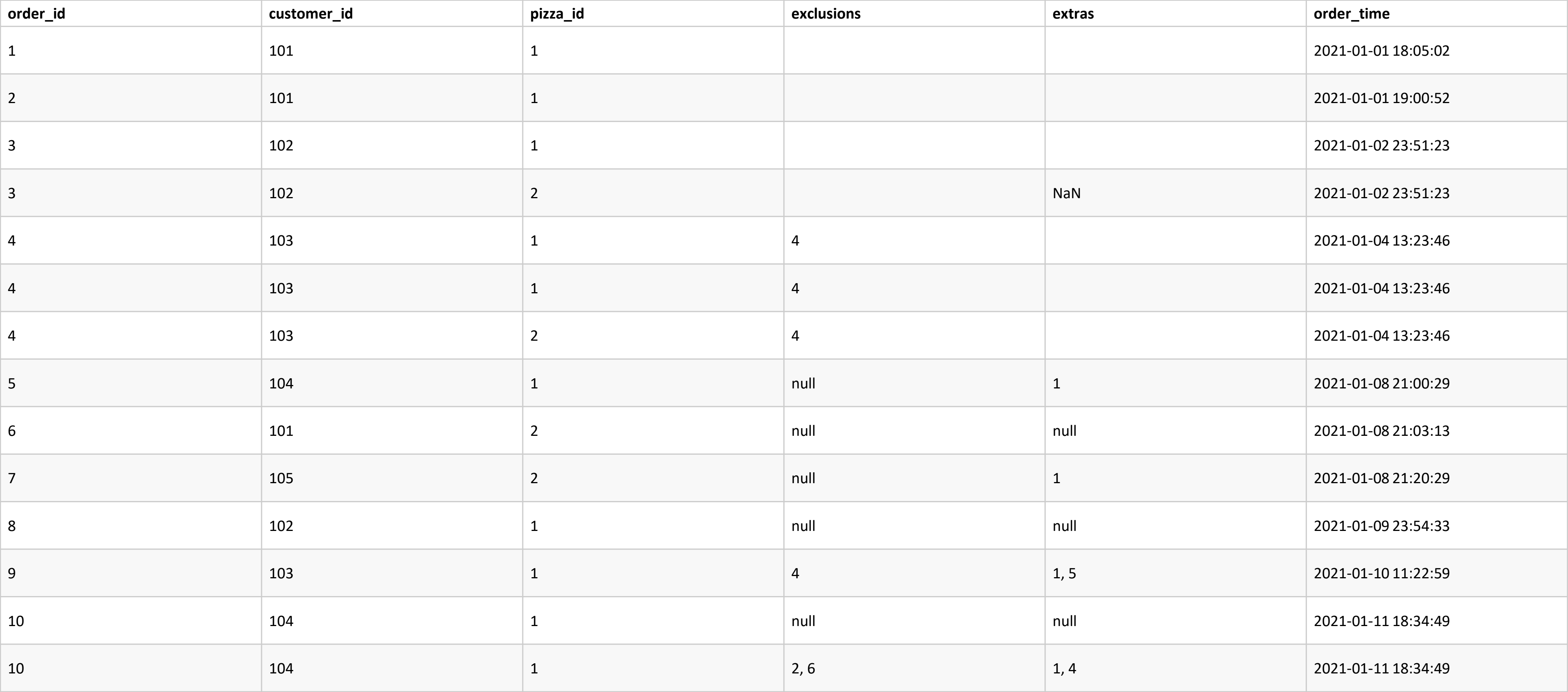
runner_id	registration_date
1	2021-01-01
2	2021-01-03
3	2021-01-08
4	2021-01-15

Table 2: customer_orders

Customer pizza orders are captured in the customer_orders table with 1 row for each individual pizza that is part of the order.

The pizza_id relates to the type of pizza which was ordered whilst the exclusions are the ingredient_id values which should be removed from the pizza and the extras are the ingredient_id values which need to be added to the pizza.

Note that customers can order multiple pizzas in a single order with varying exclusions and extras values even if the pizza is the same type!



order_id	customer_id	pizza_id	exclusions	extras	order_time
1	101	1			2021-01-01 18:05:02
2	101	1			2021-01-01 19:00:52
3	102	1			2021-01-02 23:51:23
3	102	2		NaN	2021-01-02 23:51:23
4	103	1	4		2021-01-04 13:23:46
4	103	1	4		2021-01-04 13:23:46
4	103	2	4		2021-01-04 13:23:46
5	104	1	null	1	2021-01-08 21:00:29
6	101	2	null	null	2021-01-08 21:03:13
7	105	2	null	1	2021-01-08 21:20:29
8	102	1	null	null	2021-01-09 23:54:33
9	103	1	4	1, 5	2021-01-10 11:22:59
10	104	1	null	null	2021-01-11 18:34:49
10	104	1	2, 6	1, 4	2021-01-11 18:34:49

Table 3: runner_orders

After each orders are received through the system - they are assigned to a runner - however not all orders are fully completed and can be cancelled by the restaurant or the customer.

The pickup_time is the timestamp at which the runner arrives at the Pizza Runner headquarters to pick up the freshly cooked pizzas. The distance and duration fields are related to how far and long the runner had to travel to deliver the order to the respective customer.

order_id	runner_id	pickup_time	distance	duration	cancellation
1	1	2021-01-01 18:15:34	20km	32 minutes	
2	1	2021-01-01 19:10:54	20km	27 minutes	
3	1	2021-01-03 00:12:37	13.4km	20 mins	NaN
4	2	2021-01-04 13:53:03	23.4	40	NaN
5	3	2021-01-08 21:10:57	10	15	NaN
6	3	null	null	null	Restaurant 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	Customer Cancellation
10	1	2020-01-11 18:50:20	10km	10minutes	null

Table 4: pizza_names

At the moment - Pizza Runner only has 2 pizzas available the Meat Lovers or Vegetarian!

pizza_id	pizza_name
1	Meat Lovers
2	Vegetarian

Table 5: pizza_recipes

Each pizza_id has a standard set of toppings which are used as part of the pizza recipe.

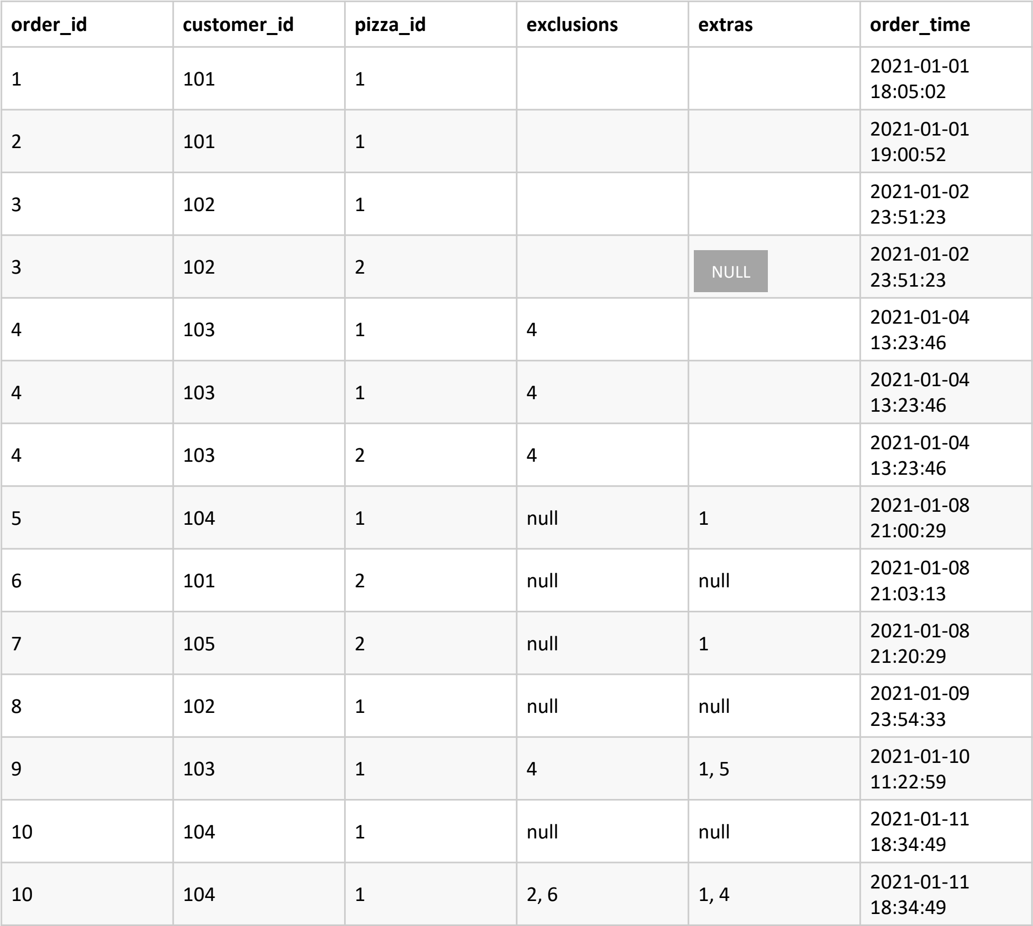
pizza_id	toppings
1	1, 2, 3, 4, 5, 6, 8, 10
2	4, 6, 7, 9, 11, 12

Table 6: pizza_toppings

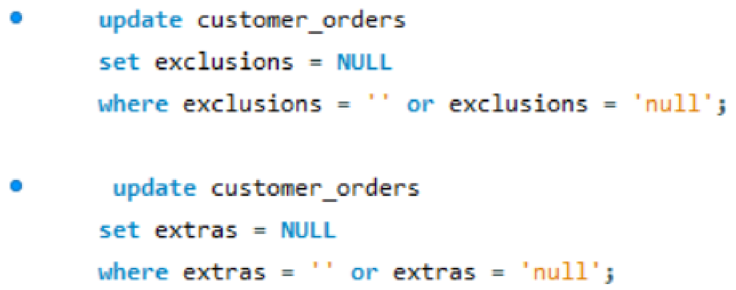
This table contains all of the topping_name values with their corresponding topping_id value

topping_id	topping_name
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

Data Cleansing

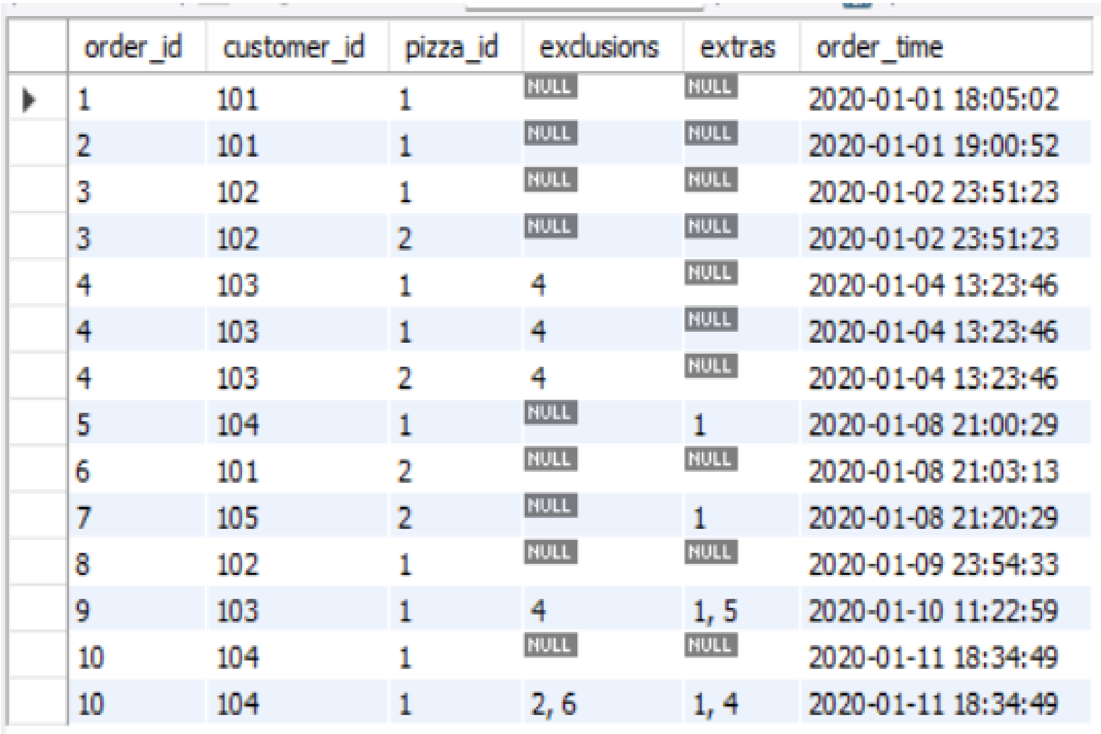


order_id	customer_id	pizza_id	exclusions	extras	order_time
1	101	1			2021-01-01 18:05:02
2	101	1			2021-01-01 19:00:52
3	102	1			2021-01-02 23:51:23
3	102	2		NULL	2021-01-02 23:51:23
4	103	1	4		2021-01-04 13:23:46
4	103	1	4		2021-01-04 13:23:46
4	103	2	4		2021-01-04 13:23:46
5	104	1	null	1	2021-01-08 21:00:29
6	101	2	null	null	2021-01-08 21:03:13
7	105	2	null	1	2021-01-08 21:20:29
8	102	1	null	null	2021-01-09 23:54:33
9	103	1	4	1, 5	2021-01-10 11:22:59
10	104	1	null	null	2021-01-11 18:34:49
10	104	1	2, 6	1, 4	2021-01-11 18:34:49



-
  ```
  update customer_orders
  set exclusions = NULL
  where exclusions = '' or exclusions = 'null';
  ```

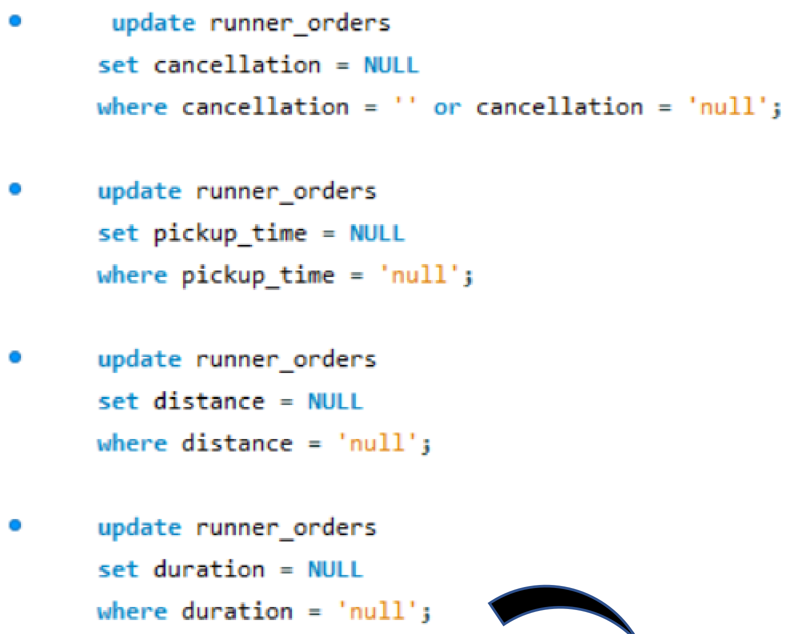
-
  ```
   update customer_orders
  set extras = NULL
  where extras = '' or extras = 'null';
  ```



order_id	customer_id	pizza_id	exclusions	extras	order_time
1	101	1	NULL	NULL	2020-01-01 18:05:02
2	101	1	NULL	NULL	2020-01-01 19:00:52
3	102	1	NULL	NULL	2020-01-02 23:51:23
3	102	2	NULL	NULL	2020-01-02 23:51:23
4	103	1	4	NULL	2020-01-04 13:23:46
4	103	1	4	NULL	2020-01-04 13:23:46
4	103	2	4	NULL	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

Data Cleansing

order_id	runner_id	pickup_time	distance	duration	cancellation
1	1	2021-01-01 18:15:34	20km	32 minutes	
2	1	2021-01-01 19:10:54	20km	27 minutes	
3	1	2021-01-03 00:12:37	13.4km	20 mins	NULL
4	2	2021-01-04 13:53:03	23.4	40	NULL
5	3	2021-01-08 21:10:57	10	15	NULL
6	3	null	null	null	Restaurant 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	Customer Cancellation
10	1	2020-01-11 18:50:20	10km	10minutes	null

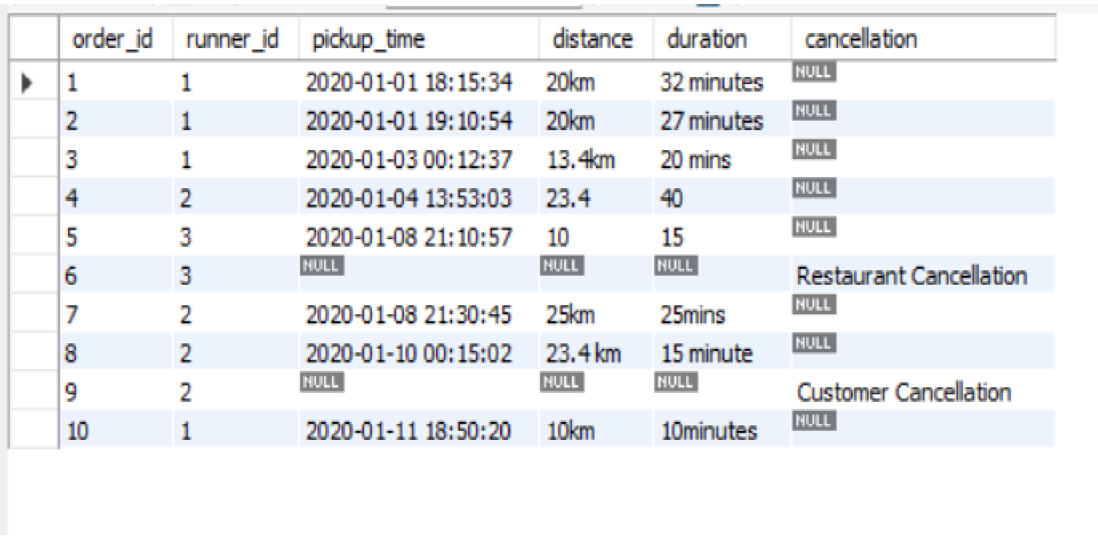


-
  ```
  update runner_orders
  set cancellation = NULL
  where cancellation = '' or cancellation = 'null';
  ```

-
  ```
  update runner_orders
  set pickup_time = NULL
  where pickup_time = 'null';
  ```

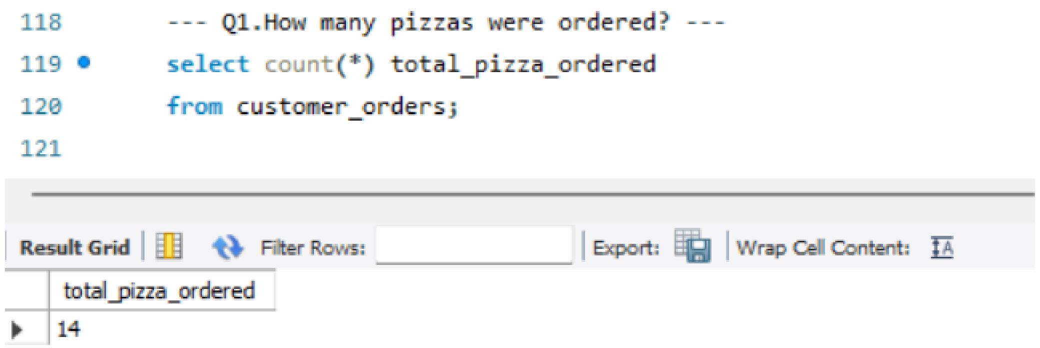
-
  ```
  update runner_orders
  set distance = NULL
  where distance = 'null';
  ```

-
  ```
  update runner_orders
  set duration = NULL
  where duration = 'null';
  ```



order_id	runner_id	pickup_time	distance	duration	cancellation
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	Restaurant 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	Customer Cancellation
10	1	2020-01-11 18:50:20	10km	10minutes	NULL

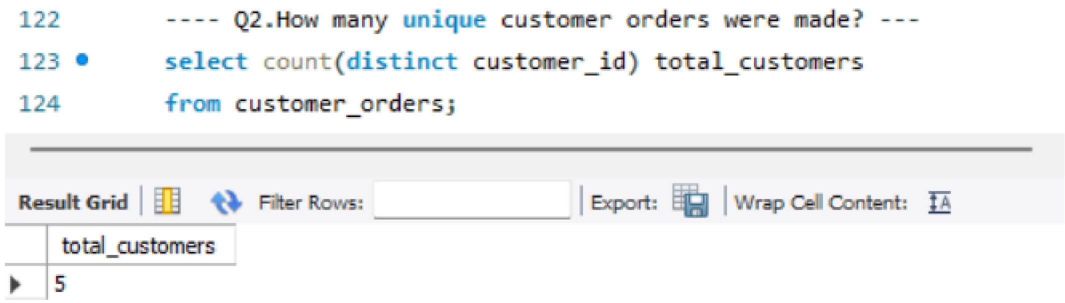
Q1.How many pizzas were ordered?



```sql
--- Q1.How many pizzas were ordered? ---
select count(*) total_pizza_ordered
from customer_orders;
```

```
total_pizza_ordered
14
```

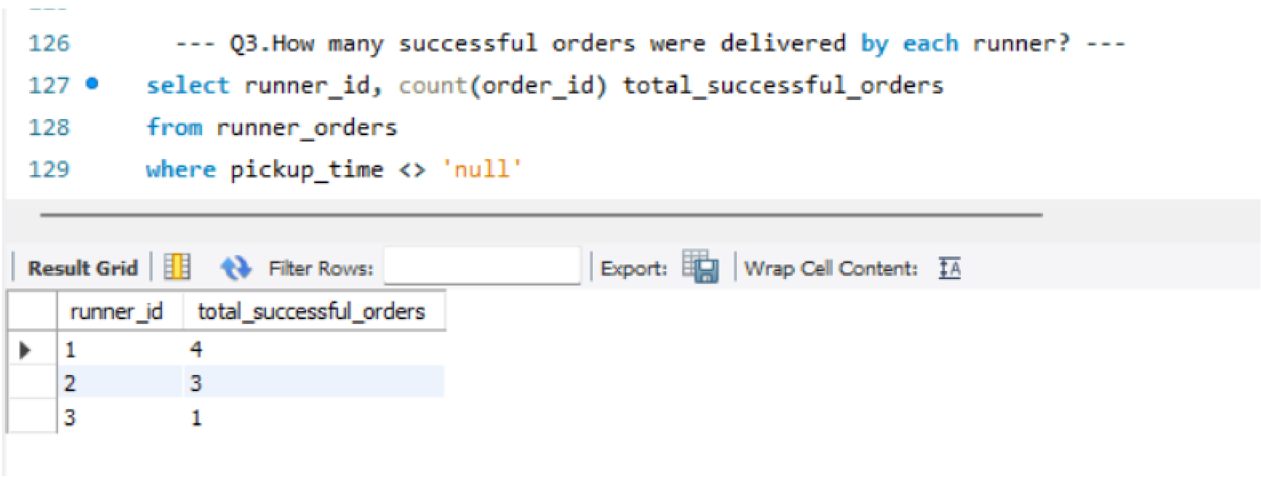
Q2.How many unique customer orders were made?



```sql
---- Q2.How many unique customer orders were made? ---
select count(distinct customer_id) total_customers
from customer_orders;
```

```
total_customers
5
```

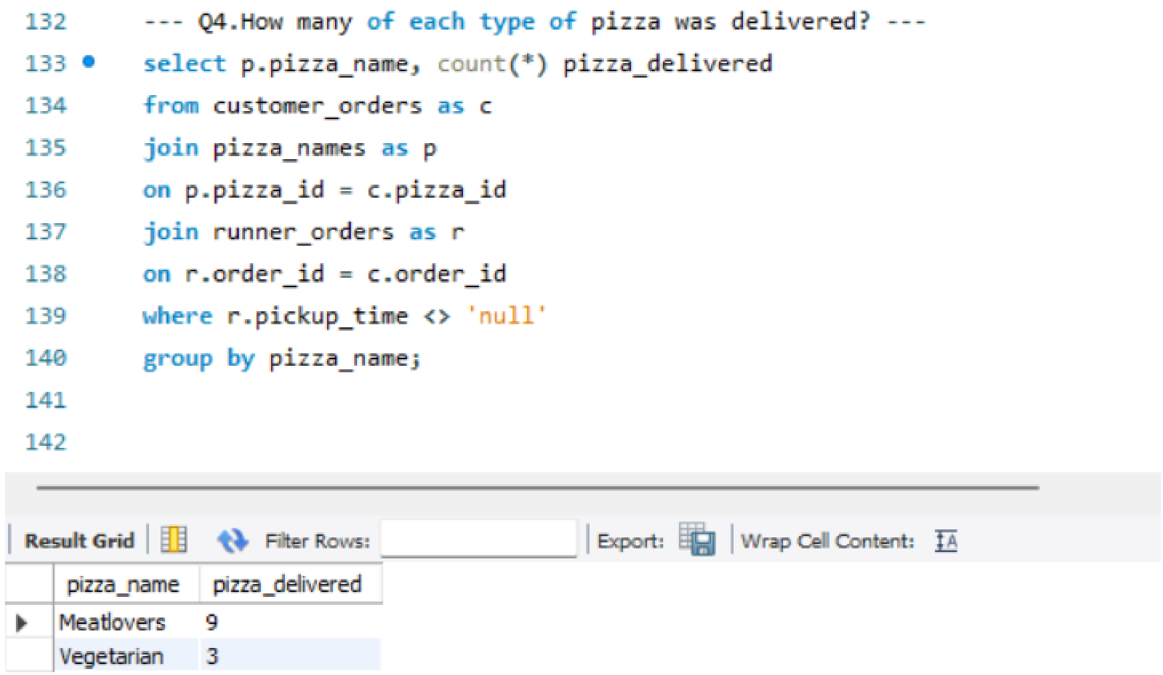
Q3.How many successful orders were delivered by each runner?



```
    --- Q3.How many successful orders were delivered by each runner? ---
select runner_id, count(order_id) total_successful_orders
from runner_orders
where pickup_time <> 'null'
```

runner_id	total_successful_orders
1	4
2	3
3	1

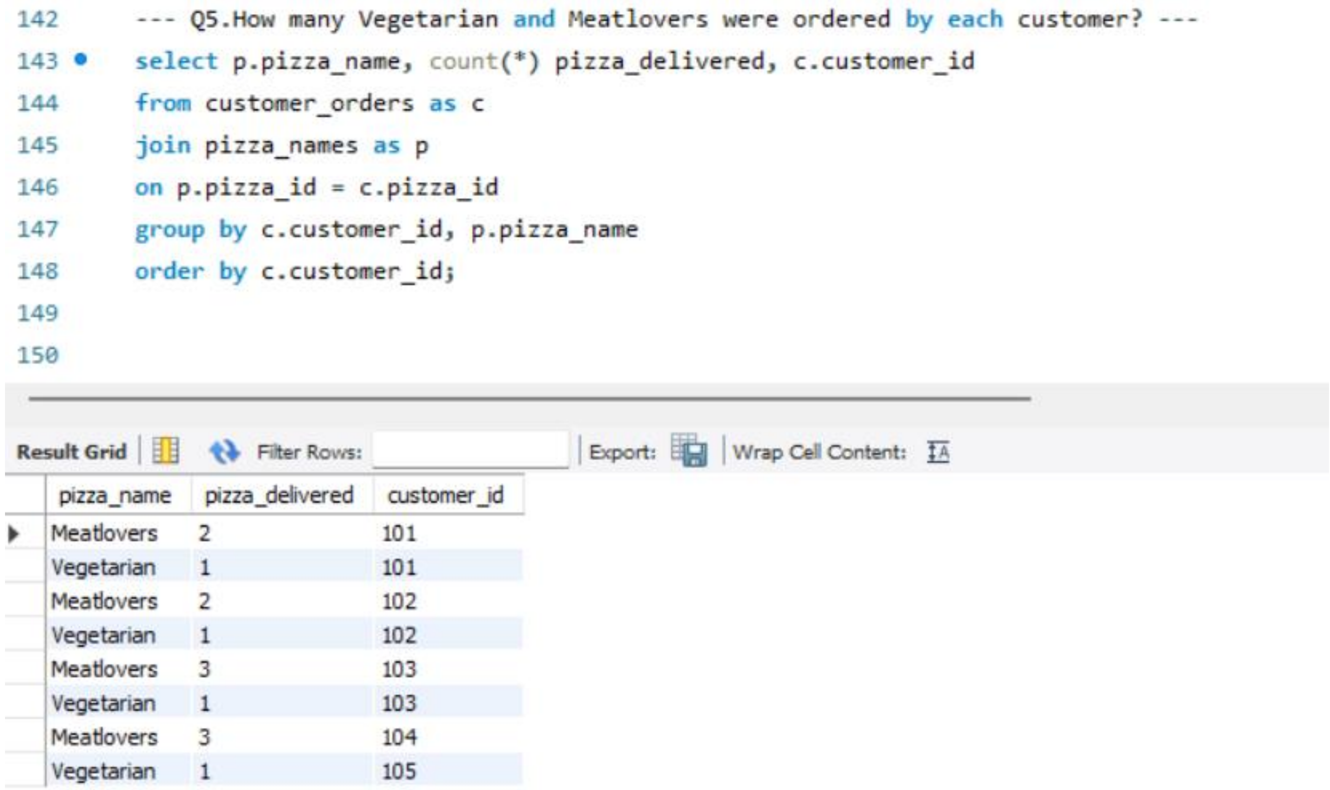
Q4.How many of each type of pizza was delivered?



```
--- Q4.How many of each type of pizza was delivered? ---
select p.pizza_name, count(*) pizza_delivered
from customer_orders as c
join pizza_names as p
on p.pizza_id = c.pizza_id
join runner_orders as r
on r.order_id = c.order_id
where r.pickup_time <> 'null'
group by pizza_name;
```

pizza_name	pizza_delivered
Meatlovers	9
Vegetarian	3

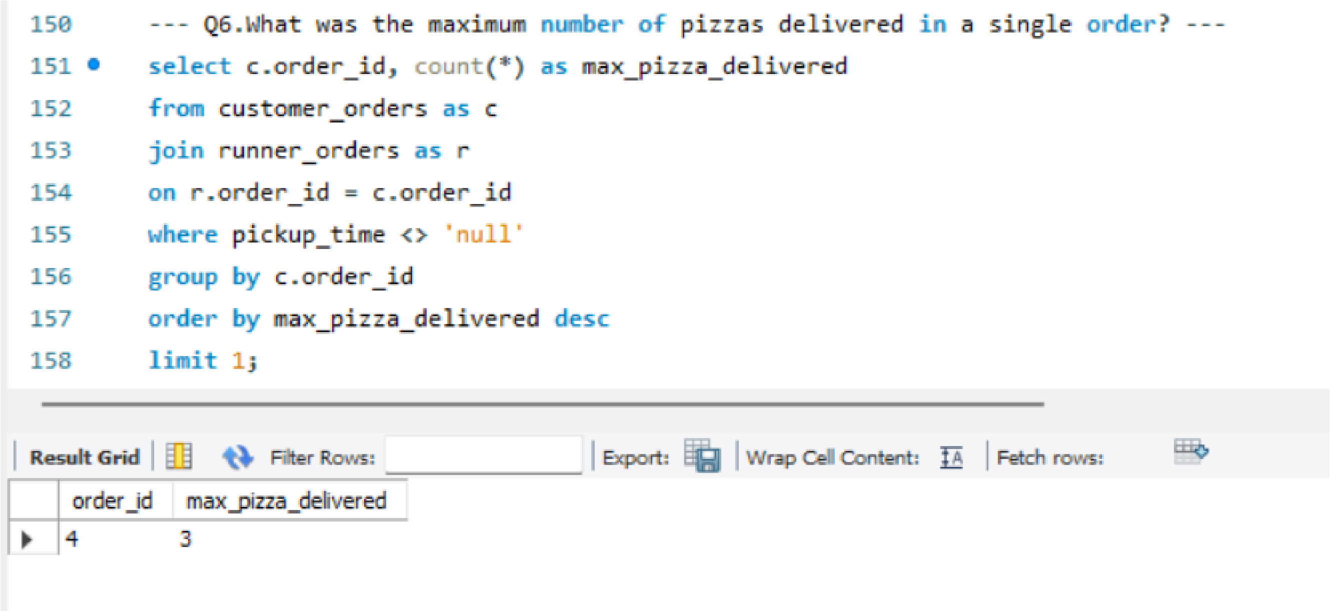
Q5.How many Vegetarian and Meatlovers were ordered by each customer?



```sql
--- Q5.How many Vegetarian and Meatlovers were ordered by each customer? ---
select p.pizza_name, count(*) pizza_delivered, c.customer_id
from customer_orders as c
join pizza_names as p
on p.pizza_id = c.pizza_id
group by c.customer_id, p.pizza_name
order by c.customer_id;
```

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

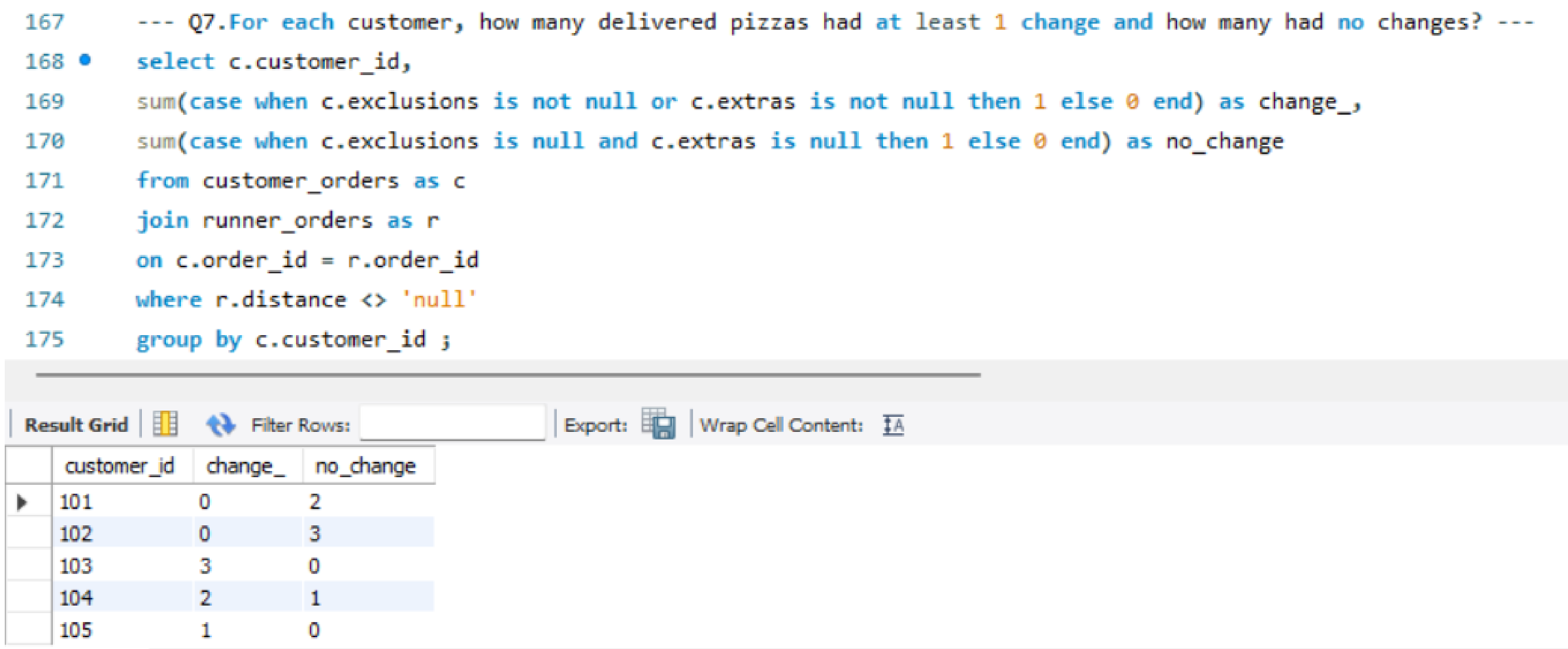
Q6What was the maximum number of pizzas delivered in a single order?



```sql
--- Q6.What was the maximum number of pizzas delivered in a single order? ---
select c.order_id, count(*) as max_pizza_delivered
from customer_orders as c
join runner_orders as r
on r.order_id = c.order_id
where pickup_time <> 'null'
group by c.order_id
order by max_pizza_delivered desc
limit 1;
```

```
order_id  max_pizza_delivered
4         3
```

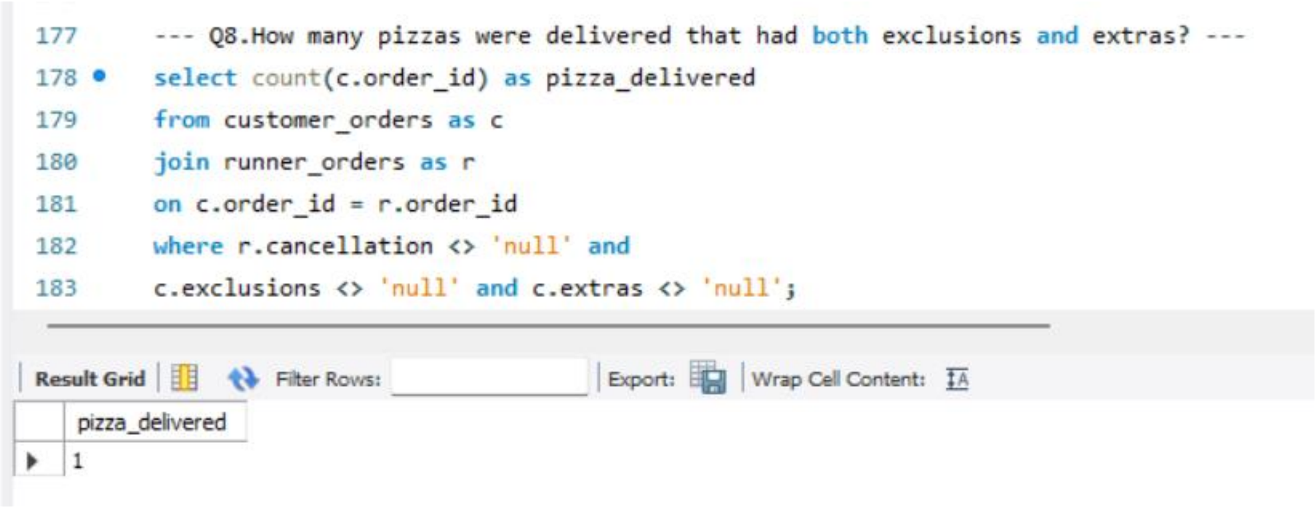
Q7.For each customer, how many delivered pizzas had at least 1 change and how many had no changes?



```
--- Q7.For each customer, how many delivered pizzas had at least 1 change and how many had no changes? ---
select c.customer_id,
sum(case when c.exclusions is not null or c.extras is not null then 1 else 0 end) as change_,
sum(case when c.exclusions is null and c.extras is null then 1 else 0 end) as no_change
from customer_orders as c
join runner_orders as r
on c.order_id = r.order_id
where r.distance <> 'null'
group by c.customer_id ;
```

customer_id	change_	no_change
101	0	2
102	0	3
103	3	0
104	2	1
105	1	0

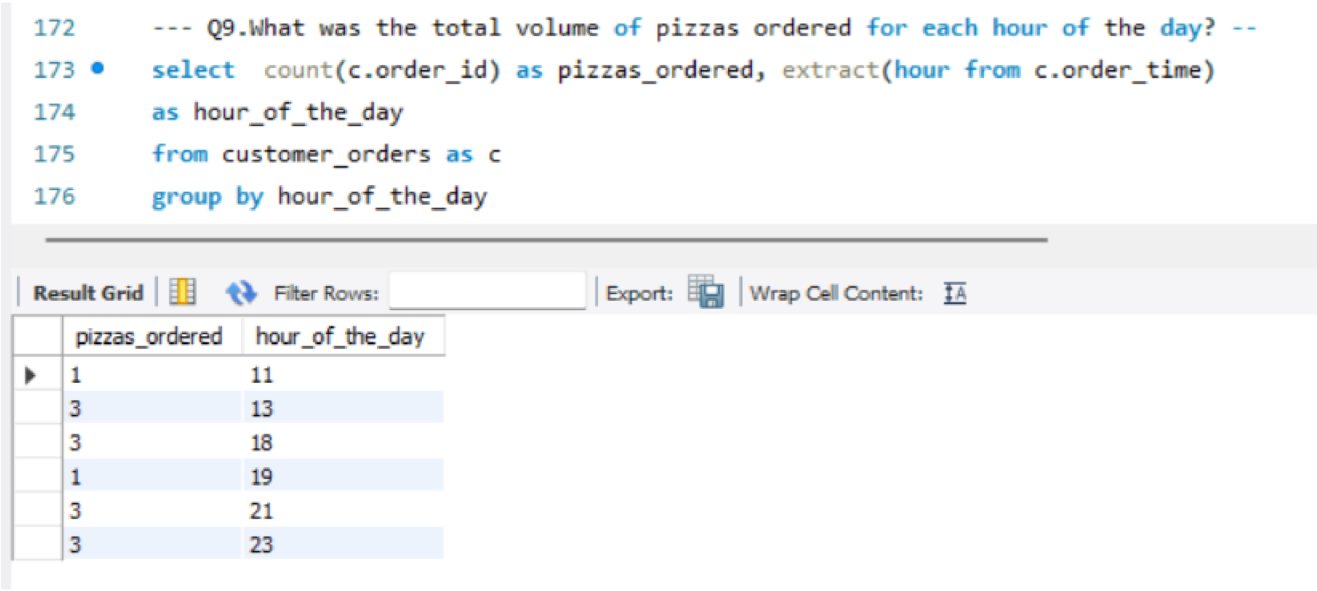
Q8.How many pizzas were delivered that had both exclusions and extras?



```
--- Q8.How many pizzas were delivered that had both exclusions and extras? ---
select count(c.order_id) as pizza_delivered
from customer_orders as c
join runner_orders as r
on c.order_id = r.order_id
where r.cancellation <> 'null' and
c.exclusions <> 'null' and c.extras <> 'null';
```

```
pizza_delivered
1
```

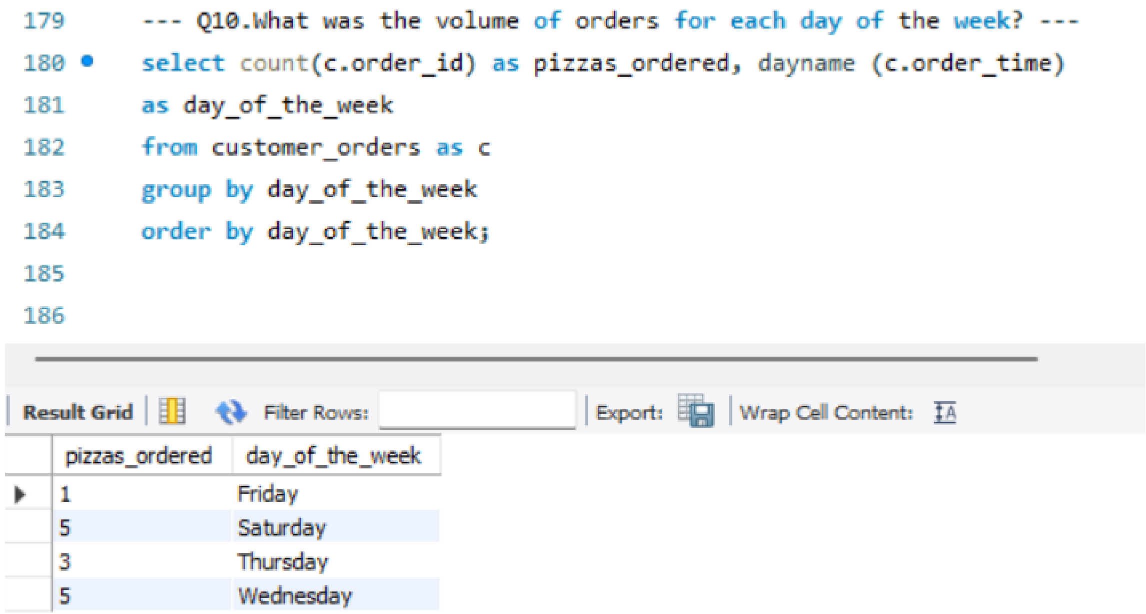
Q9.What was the total volume of pizzas ordered for each hour of the day?



```
--- Q9.What was the total volume of pizzas ordered for each hour of the day? --
select  count(c.order_id) as pizzas_ordered, extract(hour from c.order_time)
as hour_of_the_day
from customer_orders as c
group by hour_of_the_day
```

pizzas_ordered	hour_of_the_day
1	11
3	13
3	18
1	19
3	21
3	23

Q10.What was the volume of orders for each day of the week?



```
--- Q10.What was the volume of orders for each day of the week? ---
select count(c.order_id) as pizzas_ordered, dayname (c.order_time)
as day_of_the_week
from customer_orders as c
group by day_of_the_week
order by day_of_the_week;
```

pizzas_ordered	day_of_the_week
1	Friday
5	Saturday
3	Thursday
5	Wednesday