Pizza Runner

Case Study #2

Sankar Doss K

Introduction

Did you know that over **115 million kilograms** of pizza is consumed daily worldwide??? (Well according to Wikipedia anyway…)

Danny was scrolling through his Instagram feed when something really caught his eye - “80s Retro Styling and Pizza Is The Future!”

Danny was sold on the idea, but he knew that pizza alone was not going to help him get seed funding to expand his new Pizza Empire - so he had one more genius idea to combine with it - he was going to *Uberize* it - and so Pizza Runner was launched!

Danny started by recruiting “runners” to deliver fresh pizza from Pizza Runner Headquarters (otherwise known as Danny’s house) and also maxed out his credit card to pay freelance developers to build a mobile app to accept orders from customers.

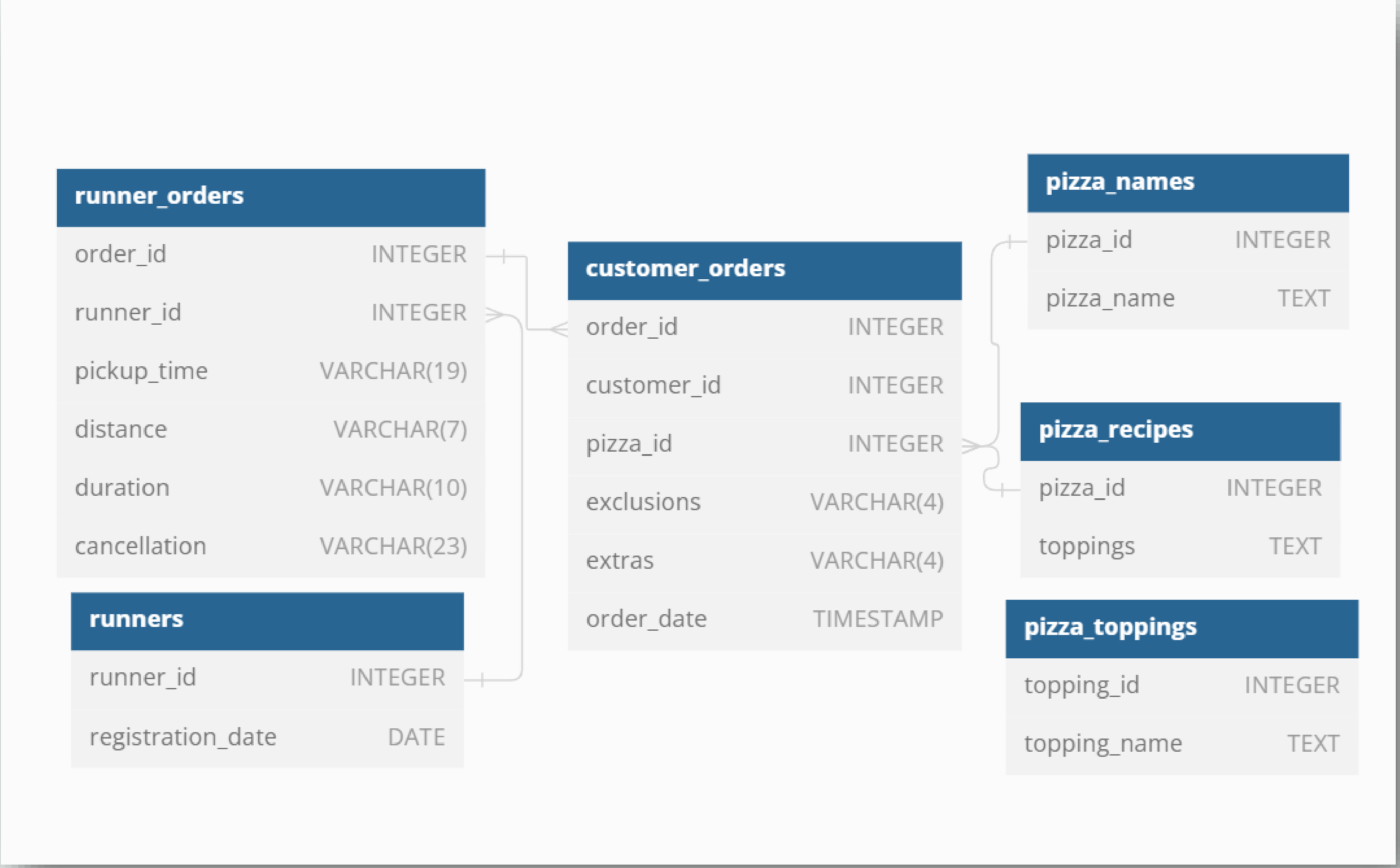
Available Data

Because Danny had a few years of experience as a data scientist - he was very aware that data collection was going to be critical for his business' growth.

He has prepared for us an entity relationship diagram of his database design but requires further assistance to clean his data and apply some basic calculations so he can better direct his runners and optimise Pizza Runner's operations.

All datasets exist within the pizza_runner database schema - be sure to include this reference within your SQL scripts as you start exploring the data and answering the case study questions.

Entity Relationship Diagram



runner_orders	
order_id	INTEGER
runner_id	INTEGER
pickup_time	VARCHAR(19)
distance	VARCHAR(7)
duration	VARCHAR(10)
cancellation	VARCHAR(23)

runners	
runner_id	INTEGER
registration_date	DATE

customer_orders	
order_id	INTEGER
customer_id	INTEGER
pizza_id	INTEGER
exclusions	VARCHAR(4)
extras	VARCHAR(4)
order_date	TIMESTAMP

pizza_names	
pizza_id	INTEGER
pizza_name	TEXT

pizza_recipes	
pizza_id	INTEGER
toppings	TEXT

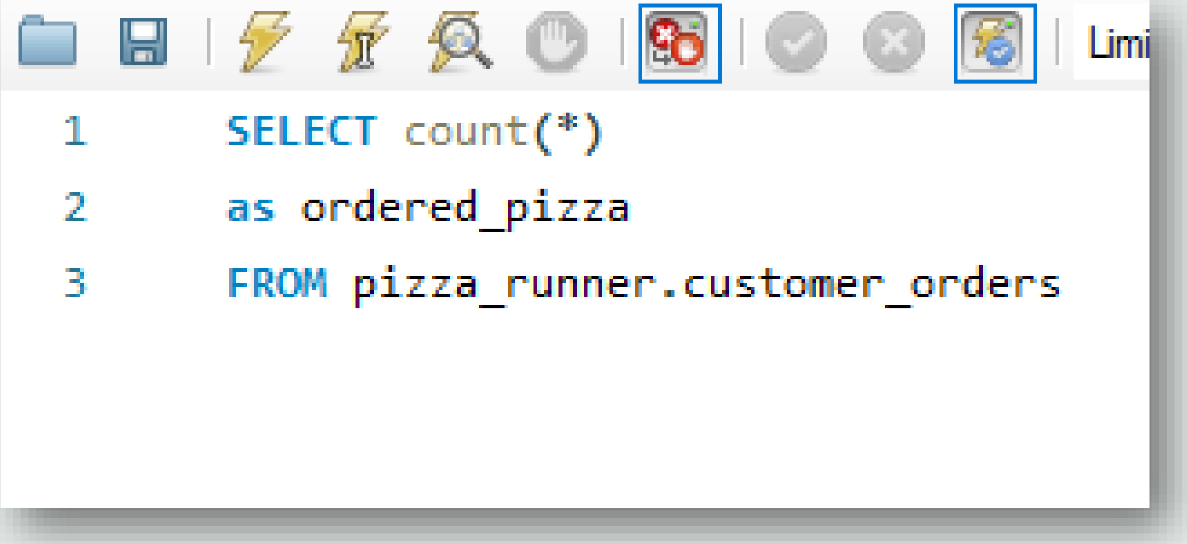
pizza_toppings	
topping_id	INTEGER
topping_name	TEXT

Case Study Questions

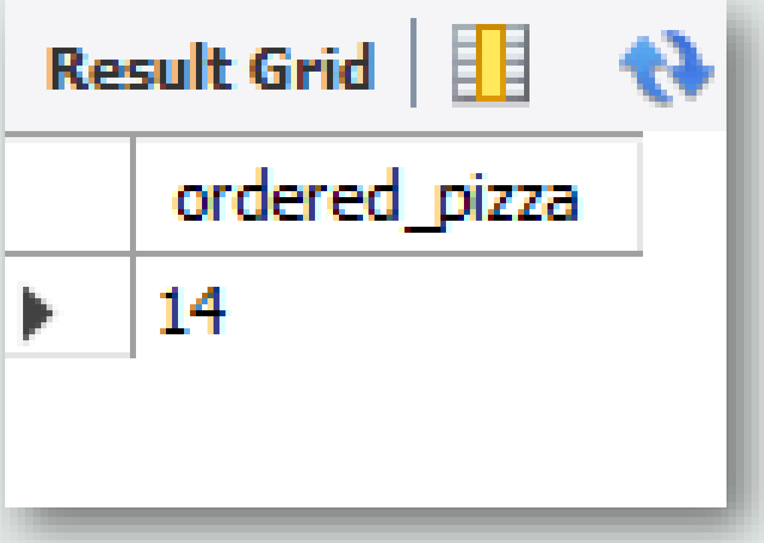
- Pizza Metrics
- Runner and Customer Experience

A. Pizza Metrics

1. *How many pizzas were ordered?*

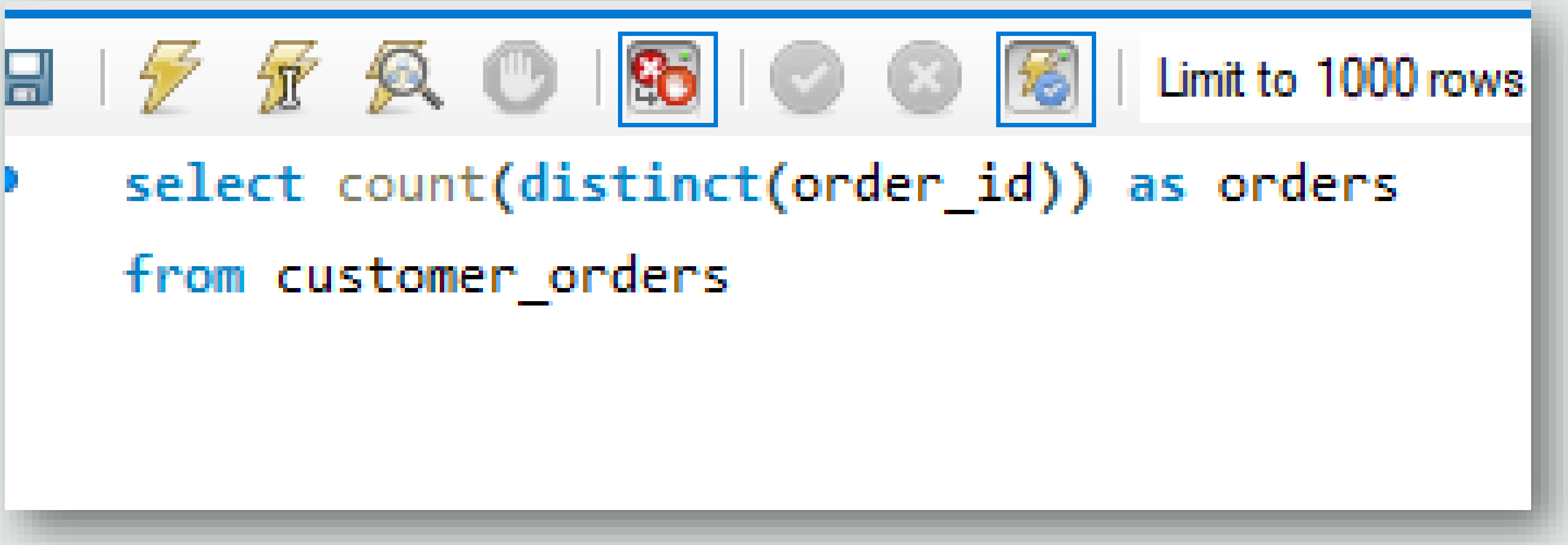


```
SELECT count(*)
as ordered_pizza
FROM pizza_runner.customer_orders
```

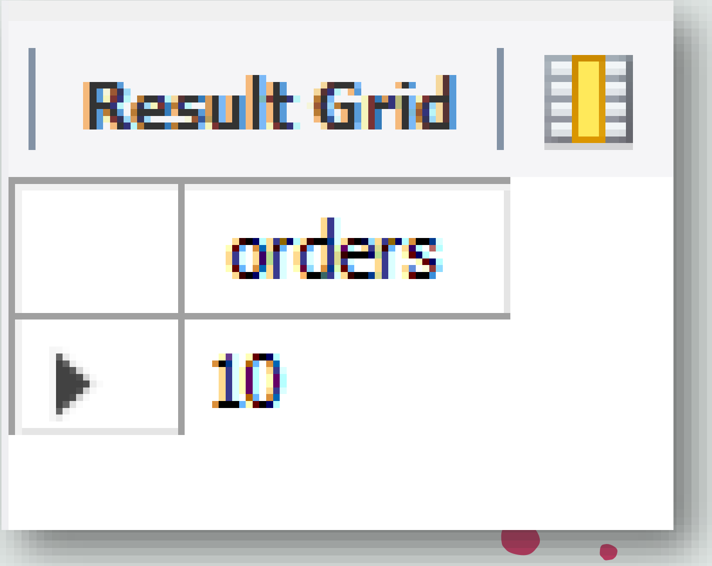


ordered_pizza
14

2.How many unique customer orders were made?

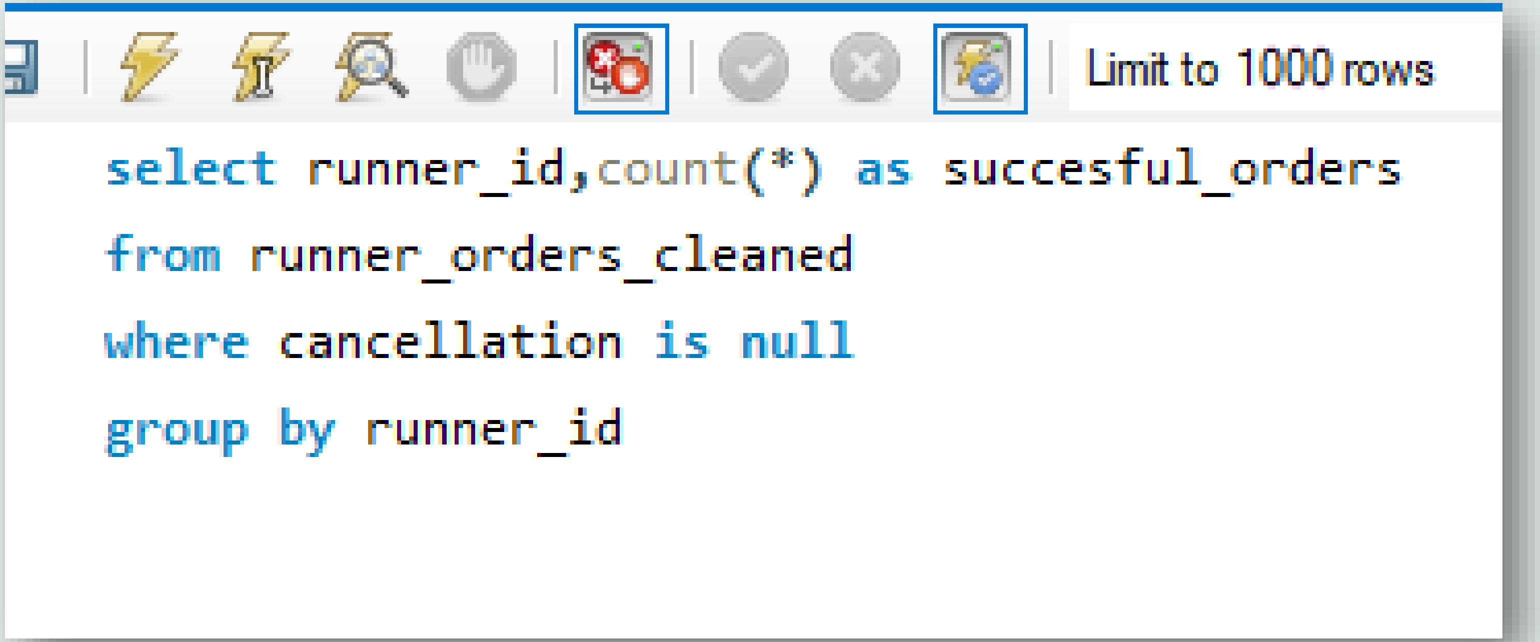


```
select count(distinct(order_id)) as orders
from customer_orders
```

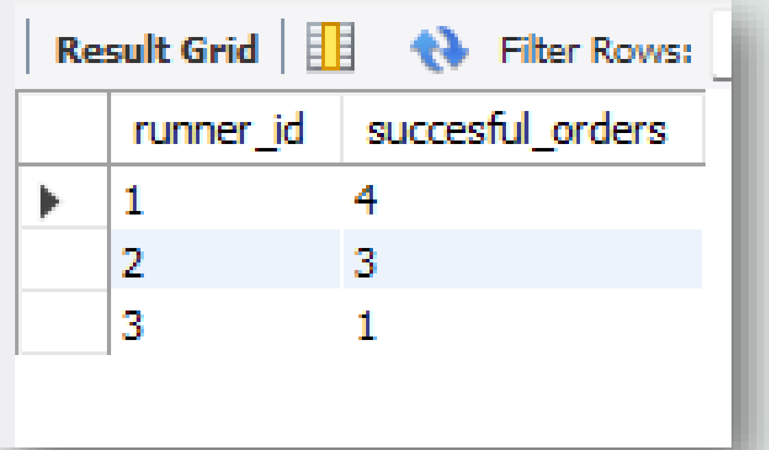


orders
10

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

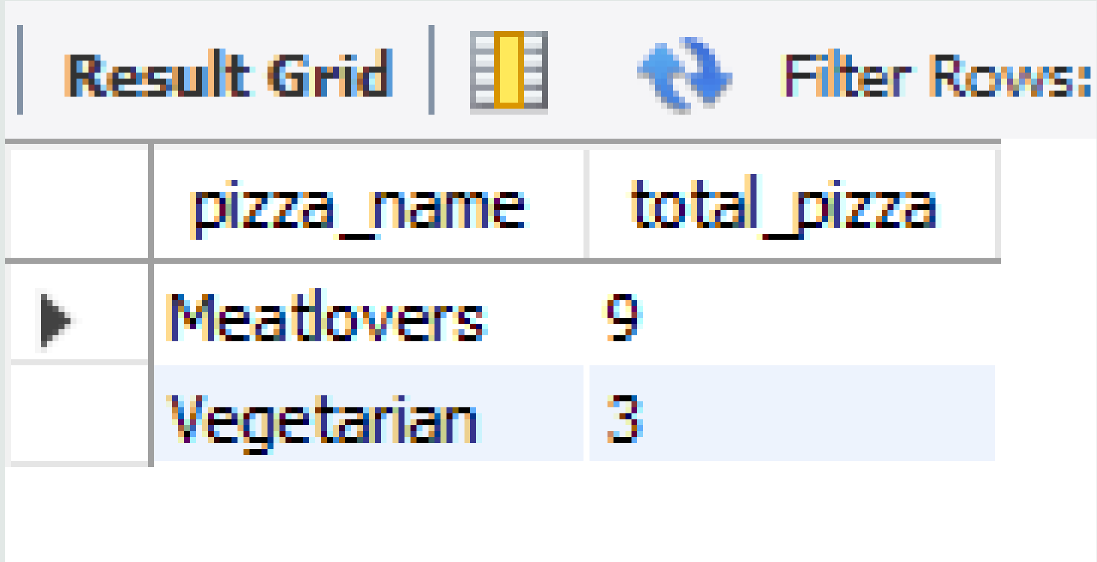


```sql
select runner_id,count(*) as succesful_orders
from runner_orders_cleaned
where cancellation is null
group by runner_id
```

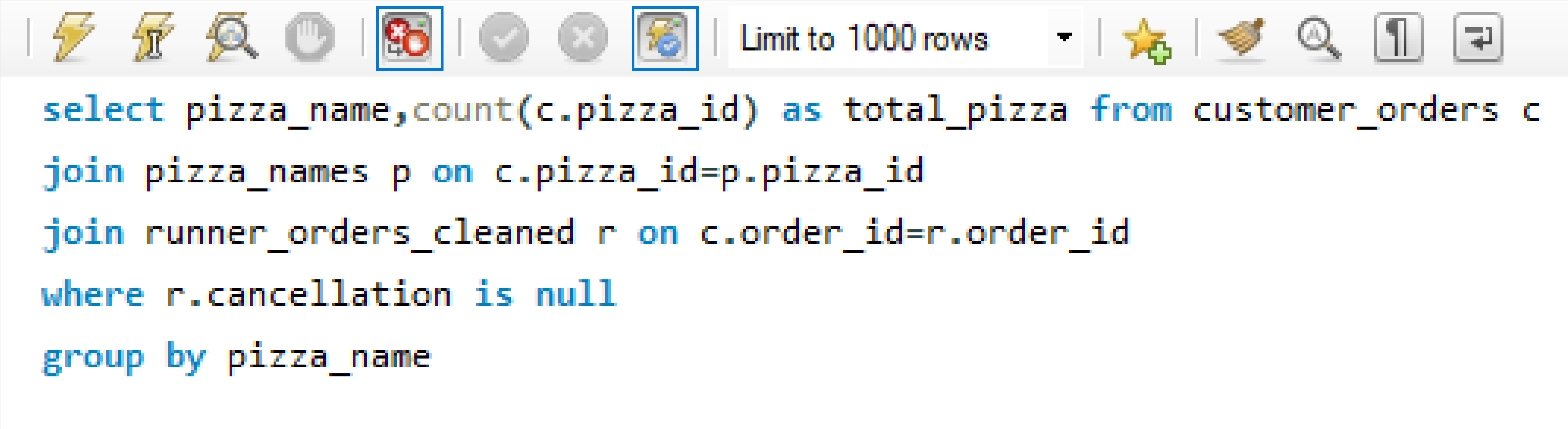


runner_id	succesful_orders
1	4
2	3
3	1

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

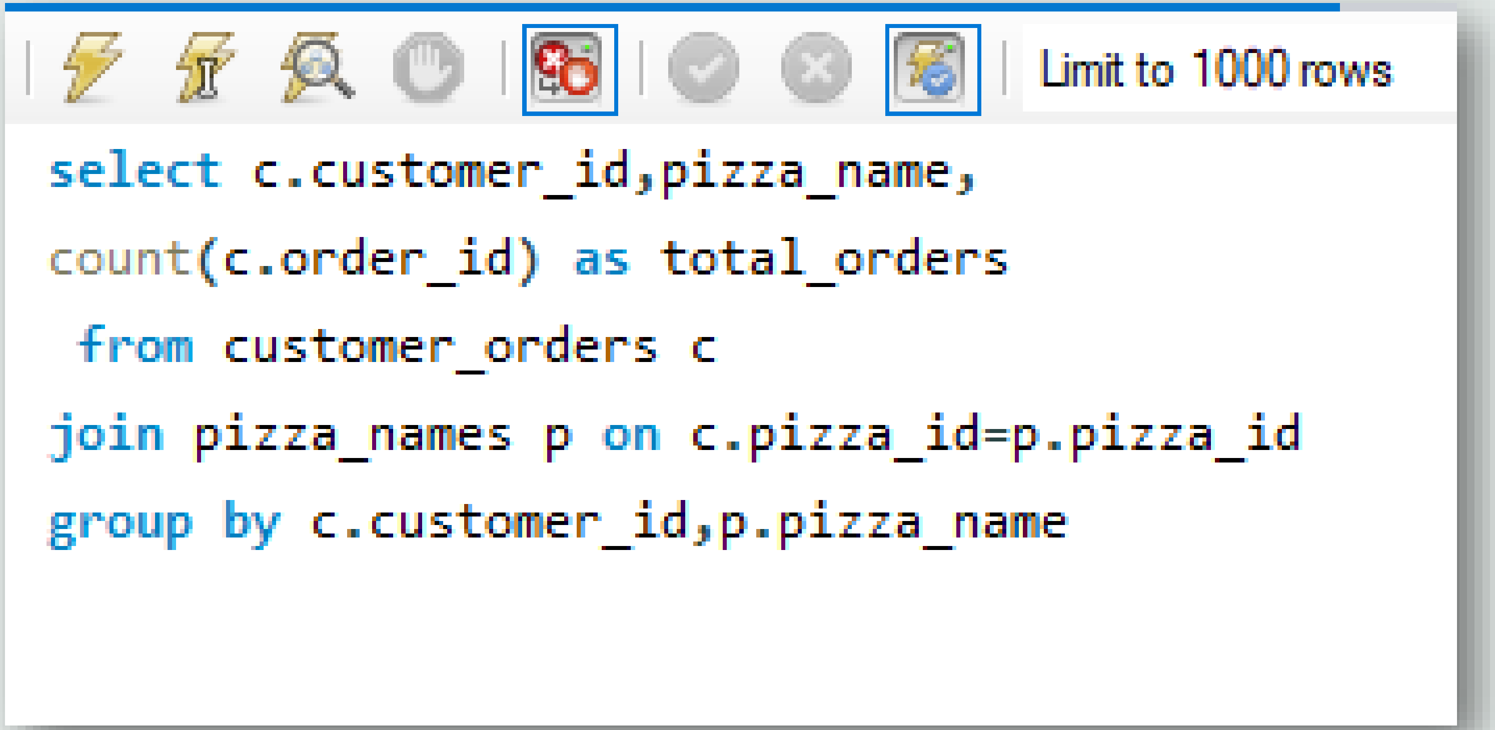


pizza_name	total_pizza
Meatlovers	9
Vegetarian	3

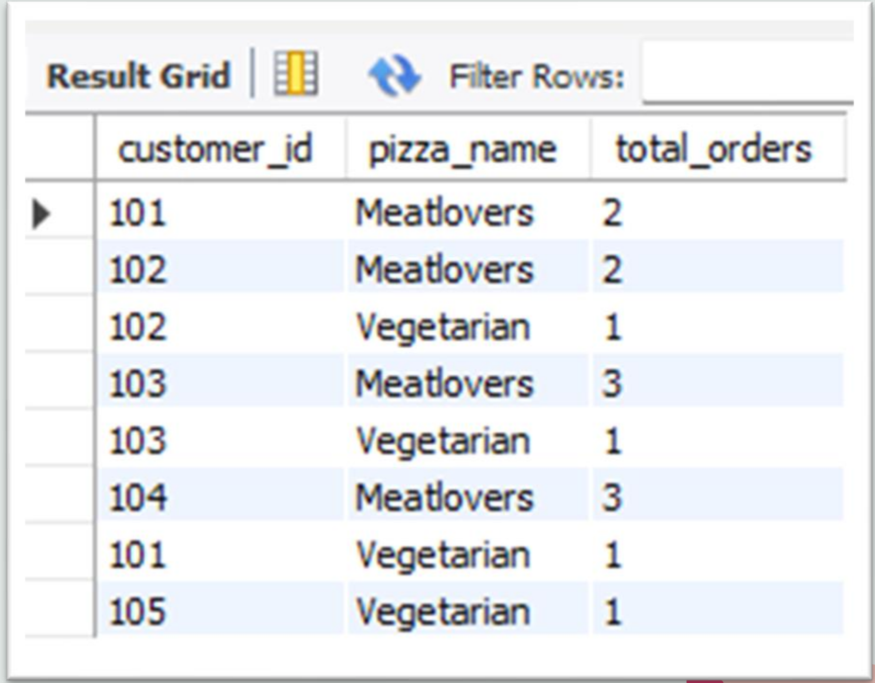


```
select pizza_name,count(c.pizza_id) as total_pizza from customer_orders c
join pizza_names p on c.pizza_id=p.pizza_id
join runner_orders_cleaned r on c.order_id=r.order_id
where r.cancellation is null
group by pizza_name
```

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

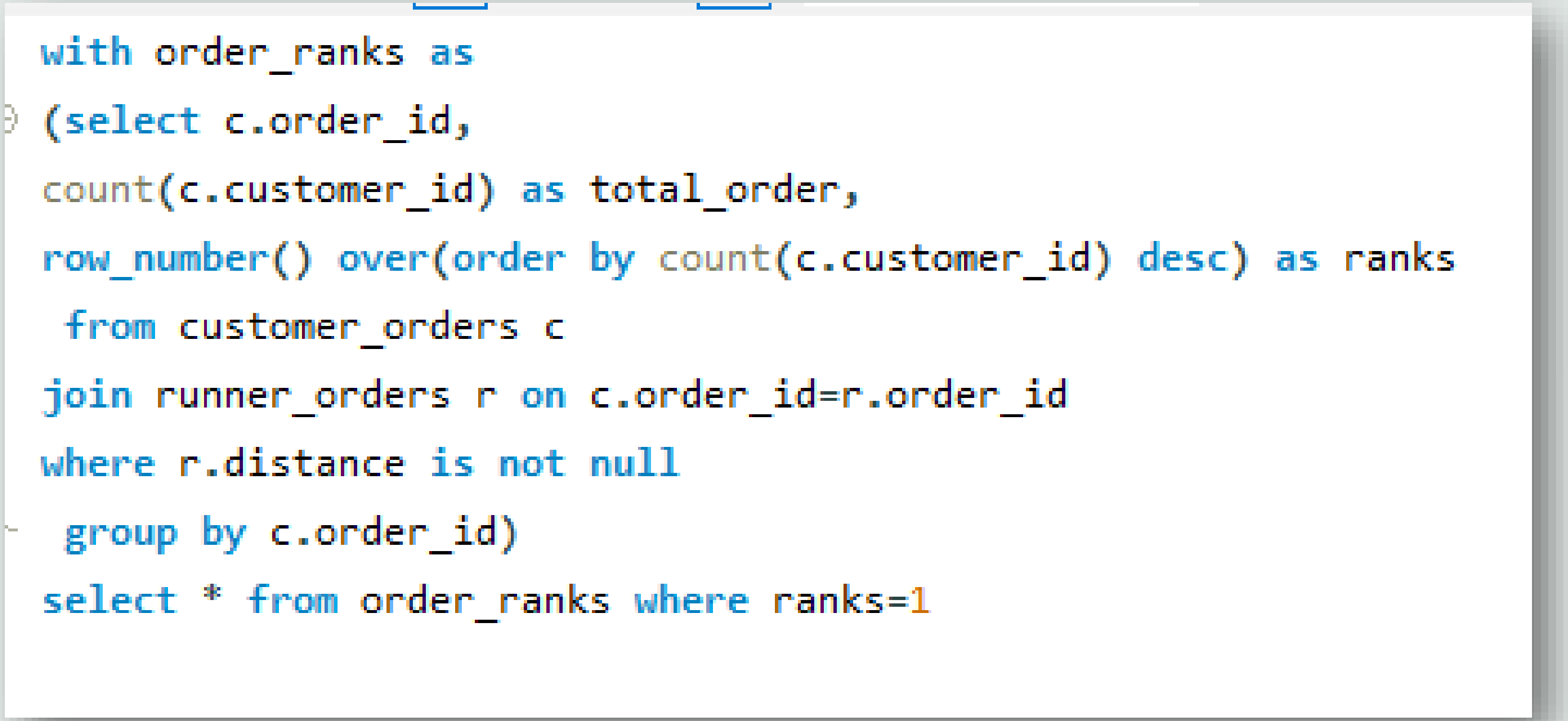


```
select c.customer_id,pizza_name,
count(c.order_id) as total_orders
 from customer_orders c
join pizza_names p on c.pizza_id=p.pizza_id
group by c.customer_id,p.pizza_name
```

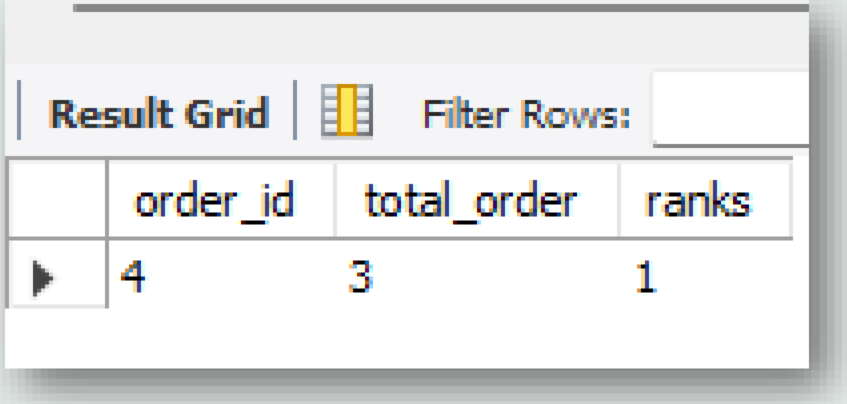


customer_id	pizza_name	total_orders
101	Meatlovers	2
102	Meatlovers	2
102	Vegetarian	1
103	Meatlovers	3
103	Vegetarian	1
104	Meatlovers	3
101	Vegetarian	1
105	Vegetarian	1

6.What was the maximum number of pizzas delivered in a single order?

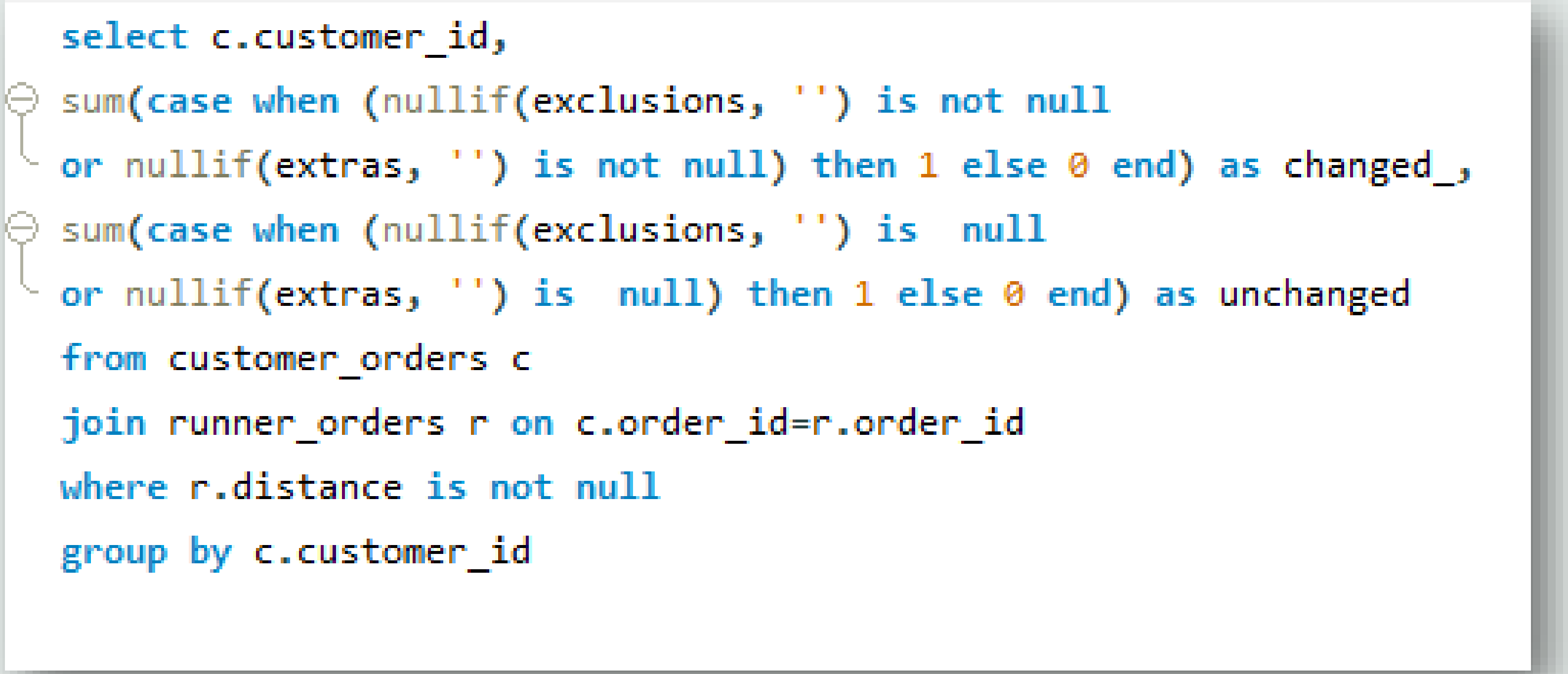


```
with order_ranks as
(select c.order_id,
count(c.customer_id) as total_order,
row_number() over(order by count(c.customer_id) desc) as ranks
 from customer_orders c
join runner_orders r on c.order_id=r.order_id
where r.distance is not null
 group by c.order_id)
select * from order_ranks where ranks=1
```

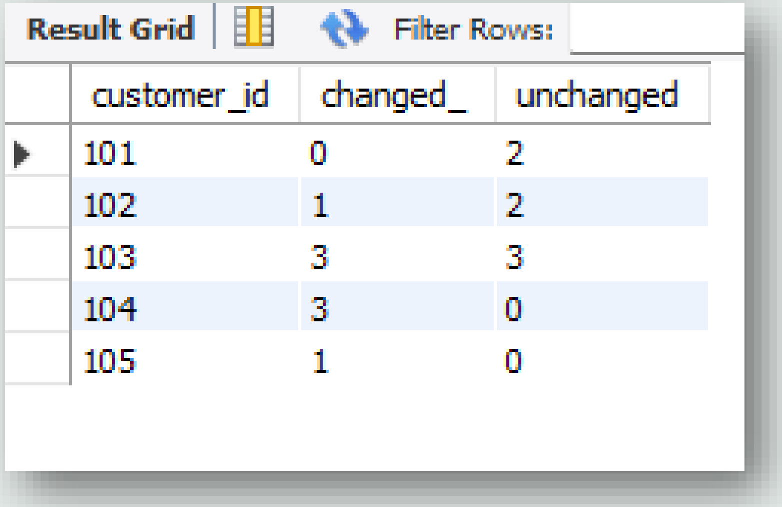


order_id	total_order	ranks
4	3	1

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

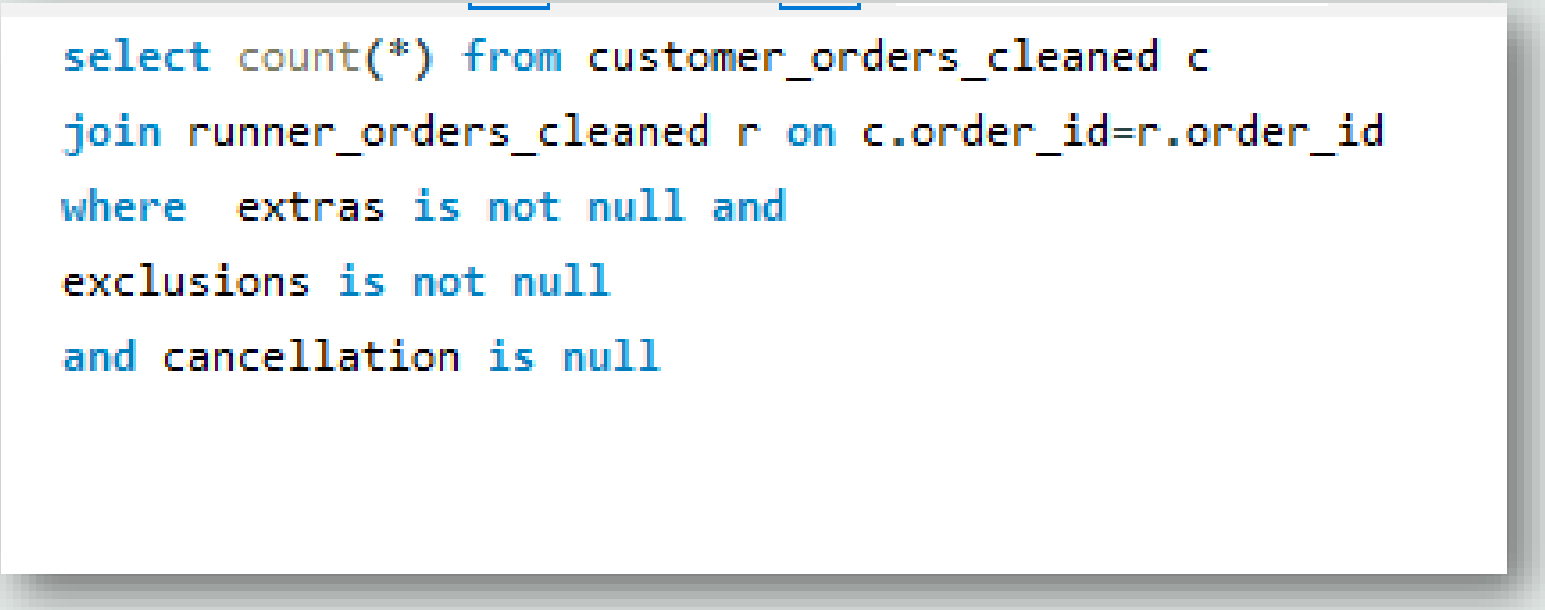


```
select c.customer_id,
sum(case when (nullif(exclusions, '') is not null
or nullif(extras, '') is not null) then 1 else 0 end) as changed_,
sum(case when (nullif(exclusions, '') is  null
or nullif(extras, '') is  null) then 1 else 0 end) as unchanged
from customer_orders c
join runner_orders r on c.order_id=r.order_id
where r.distance is not null
group by c.customer_id
```

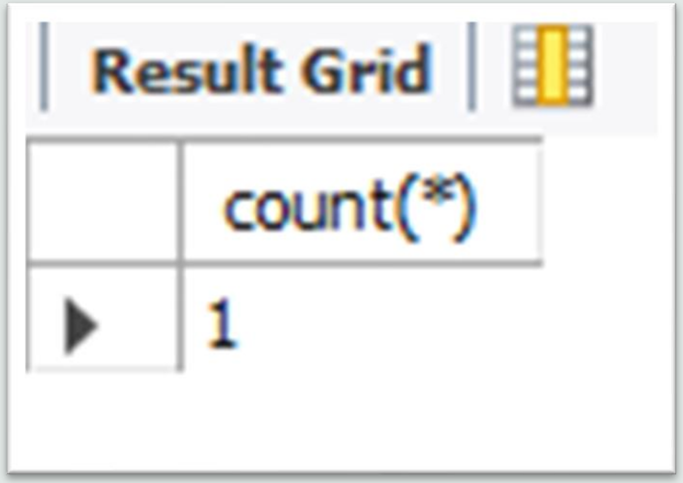


customer_id	changed_	unchanged
101	0	2
102	1	2
103	3	3
104	3	0
105	1	0

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

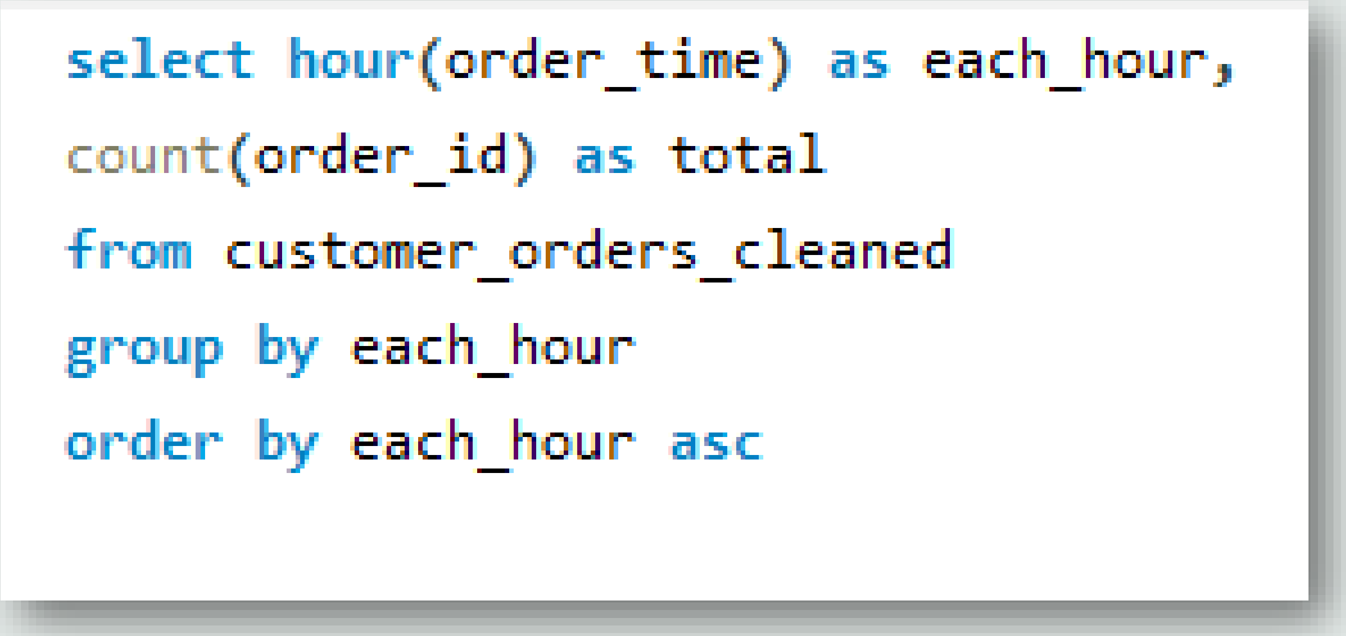


```
select count(*) from customer_orders_cleaned c
join runner_orders_cleaned r on c.order_id=r.order_id
where  extras is not null and
exclusions is not null
and cancellation is null
```

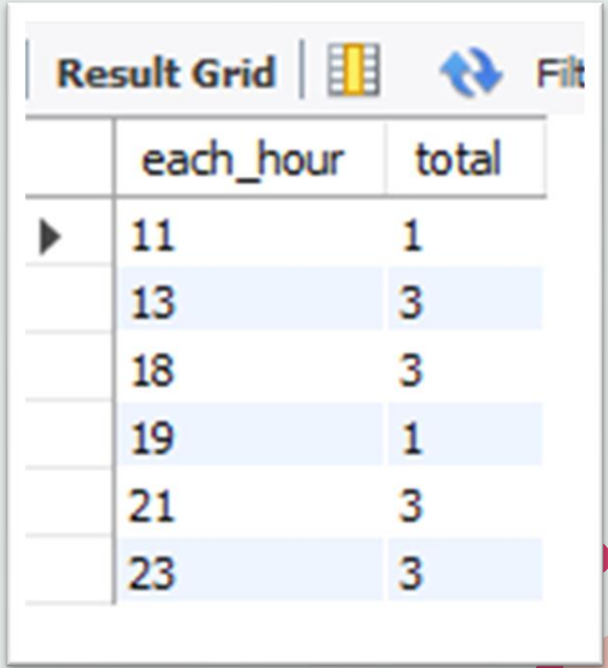


count(*)
1

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

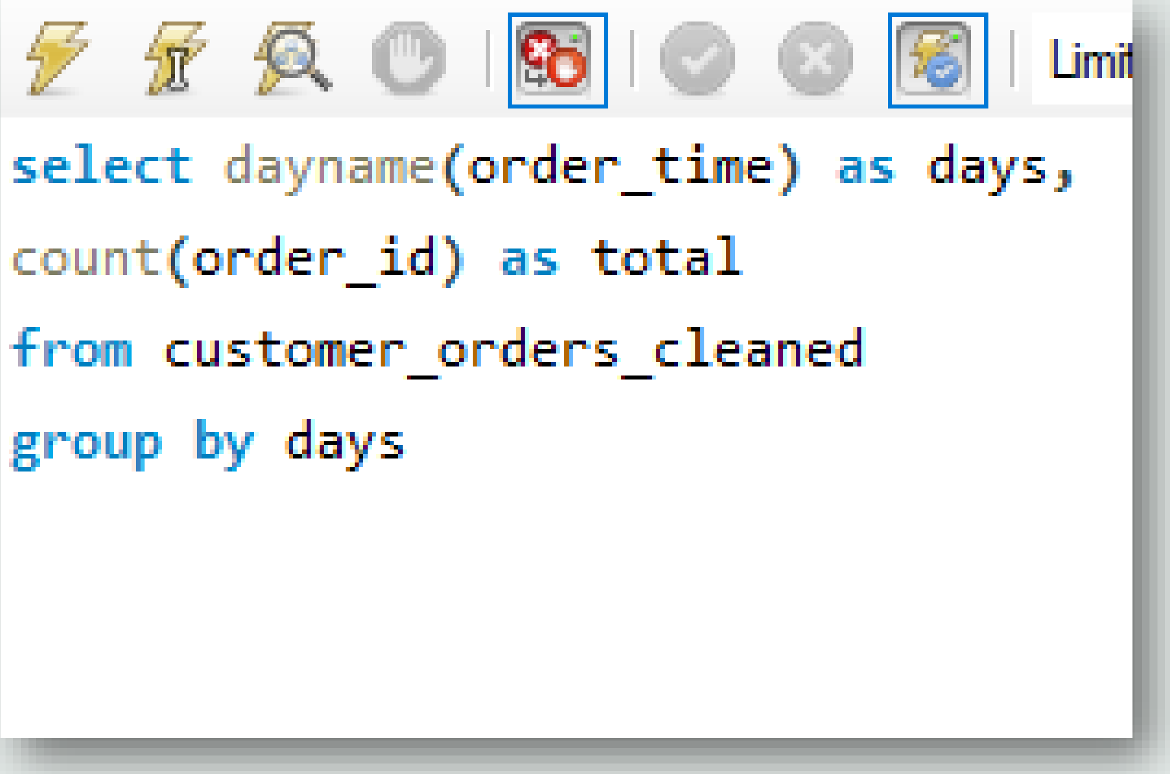


```
select hour(order_time) as each_hour,
count(order_id) as total
from customer_orders_cleaned
group by each_hour
order by each_hour asc
```

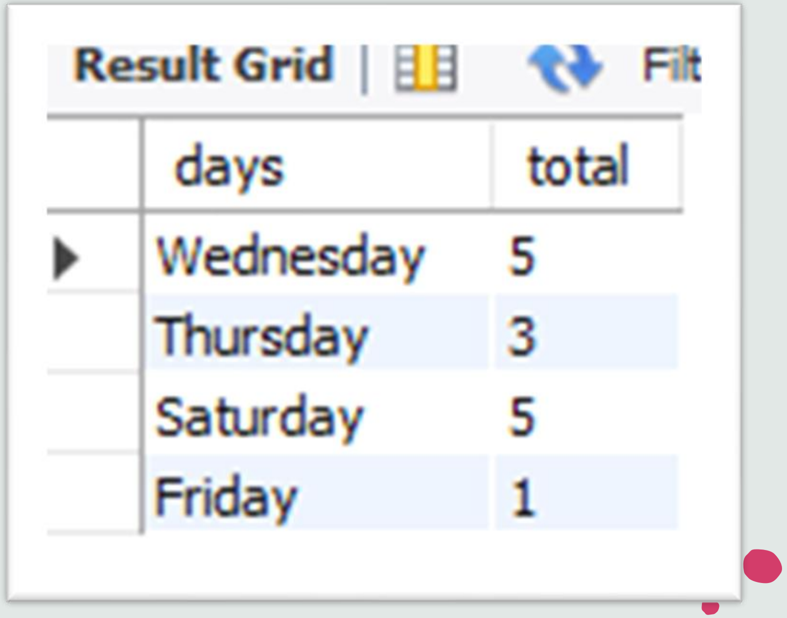


each_hour	total
11	1
13	3
18	3
19	1
21	3
23	3

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



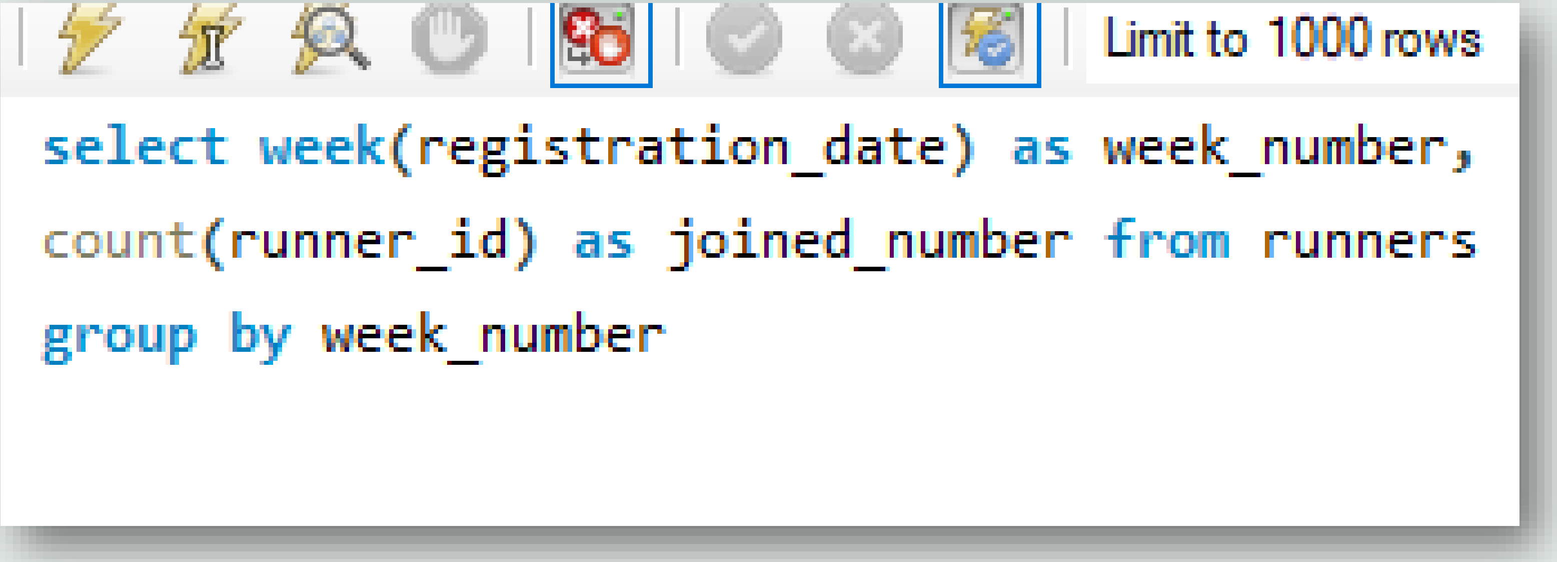
```
select dayname(order_time) as days,
count(order_id) as total
from customer_orders_cleaned
group by days
```



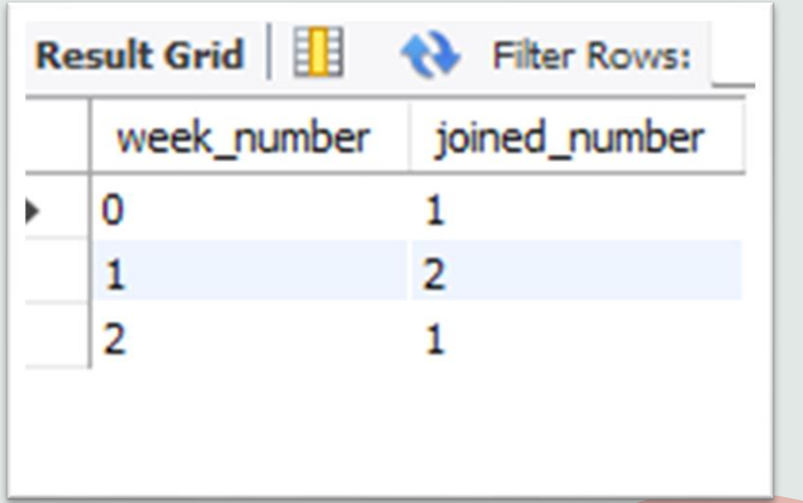
days	total
Wednesday	5
Thursday	3
Saturday	5
Friday	1

B.Runner and Customer Experience

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

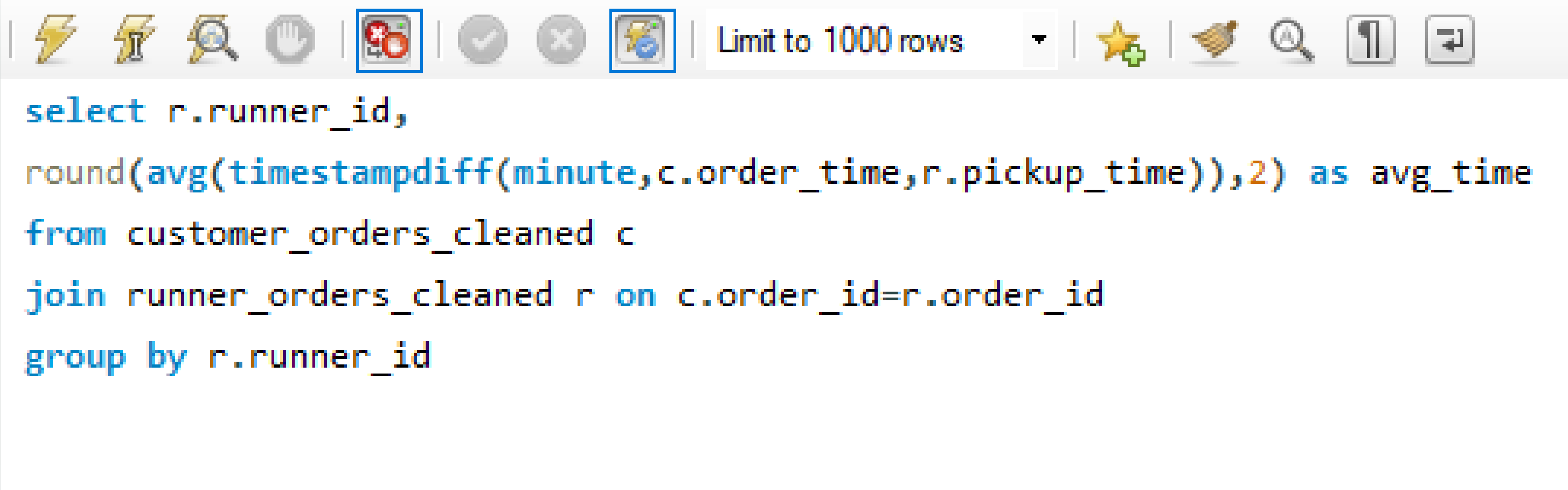


```
select week(registration_date) as week_number,
count(runner_id) as joined_number from runners
group by week_number
```

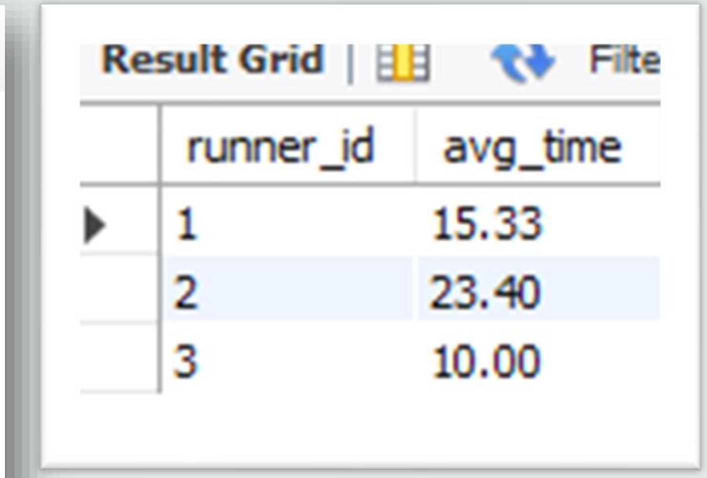


week_number	joined_number
0	1
1	2
2	1

2.What was the average time in minutes it took for each runner to arrive at the Pizza Runner HQ to pickup the order?

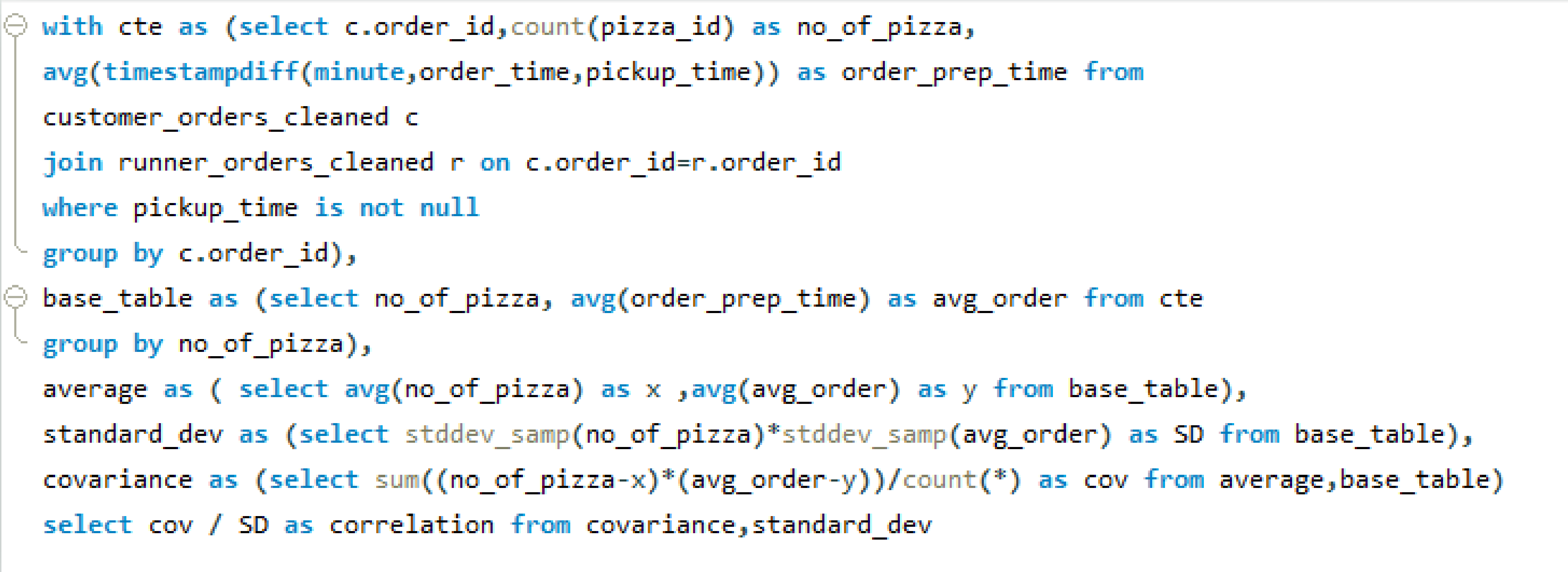


```sql
select r.runner_id,
round(avg(timestampdiff(minute,c.order_time,r.pickup_time)),2) as avg_time
from customer_orders_cleaned c
join runner_orders_cleaned r on c.order_id=r.order_id
group by r.runner_id
```



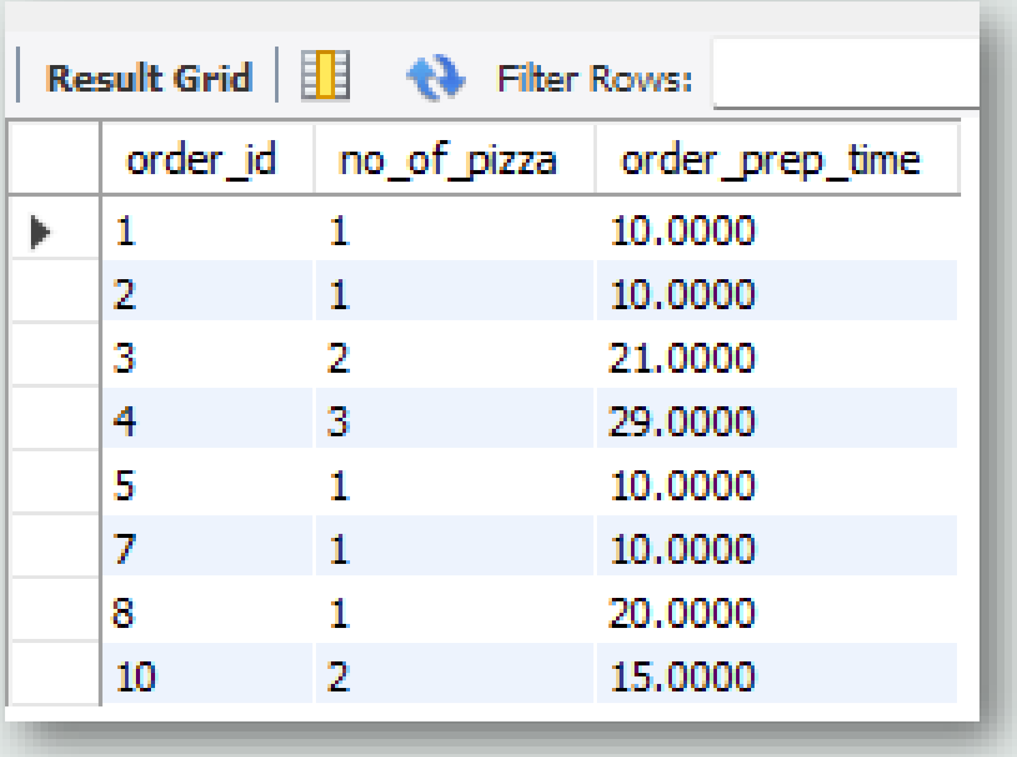
runner_id	avg_time
1	15.33
2	23.40
3	10.00

3.Is there any relationship between the number of pizzas and how long the order takes to prepare?

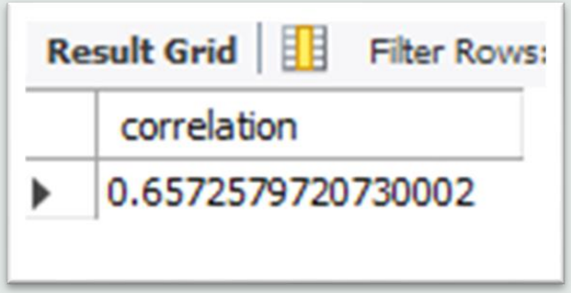


```
with cte as (select c.order_id,count(pizza_id) as no_of_pizza,
avg(timestampdiff(minute,order_time,pickup_time)) as order_prep_time from
customer_orders_cleaned c
join runner_orders_cleaned r on c.order_id=r.order_id
where pickup_time is not null
group by c.order_id),
base_table as (select no_of_pizza, avg(order_prep_time) as avg_order from cte
group by no_of_pizza),
average as ( select avg(no_of_pizza) as x ,avg(avg_order) as y from base_table),
standard_dev as (select stddev_samp(no_of_pizza)*stddev_samp(avg_order) as SD from base_table),
covariance as (select sum((no_of_pizza-x)*(avg_order-y))/count(*) as cov from average,base_table)
select cov / SD as correlation from covariance,standard_dev
```

3.Is there any relationship between the number of pizzas and how long the order takes to prepare?



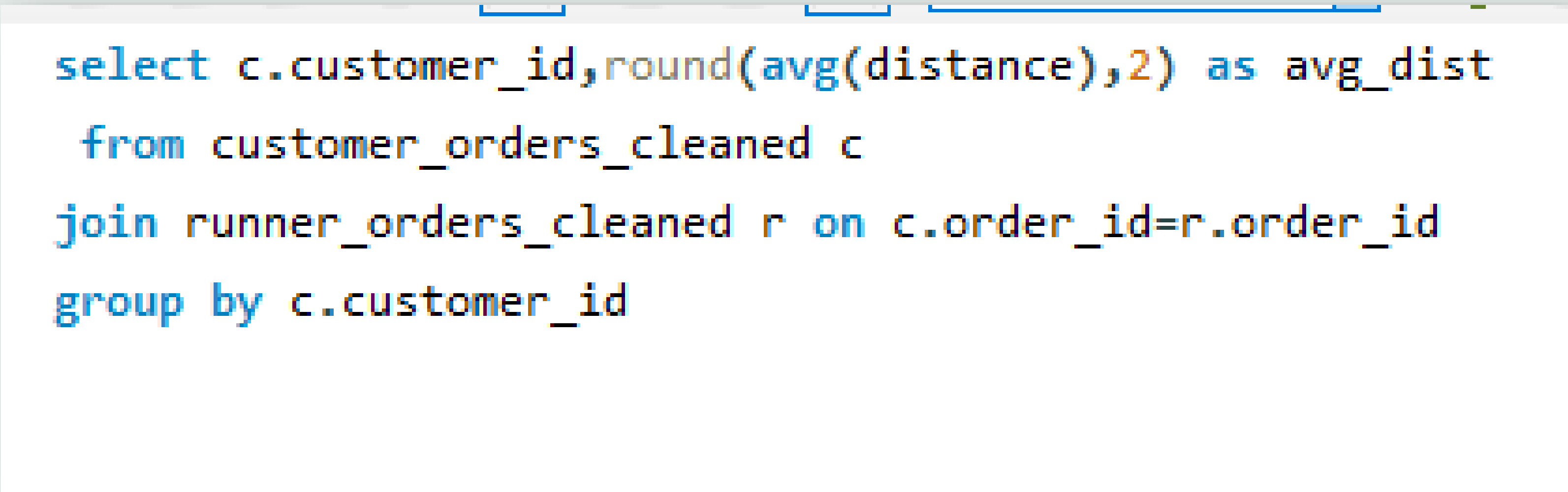
order_id	no_of_pizza	order_prep_time
1	1	10.0000
2	1	10.0000
3	2	21.0000
4	3	29.0000
5	1	10.0000
7	1	10.0000
8	1	20.0000
10	2	15.0000



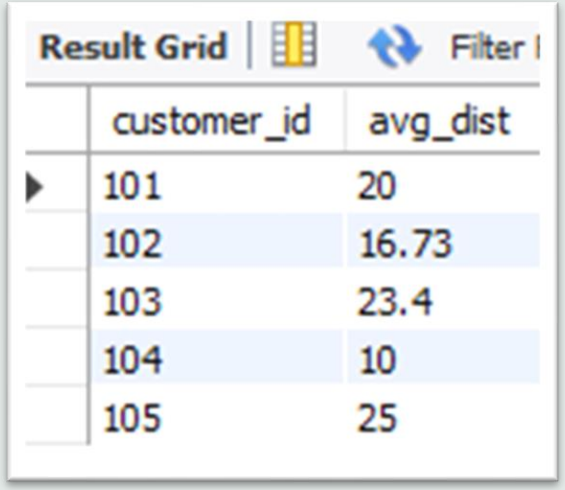
correlation
0.6572579720730002

A correlation coefficient of 0.65 indicates a moderate positive relationship between two variables.

4.What was the average distance travelled for each customer?

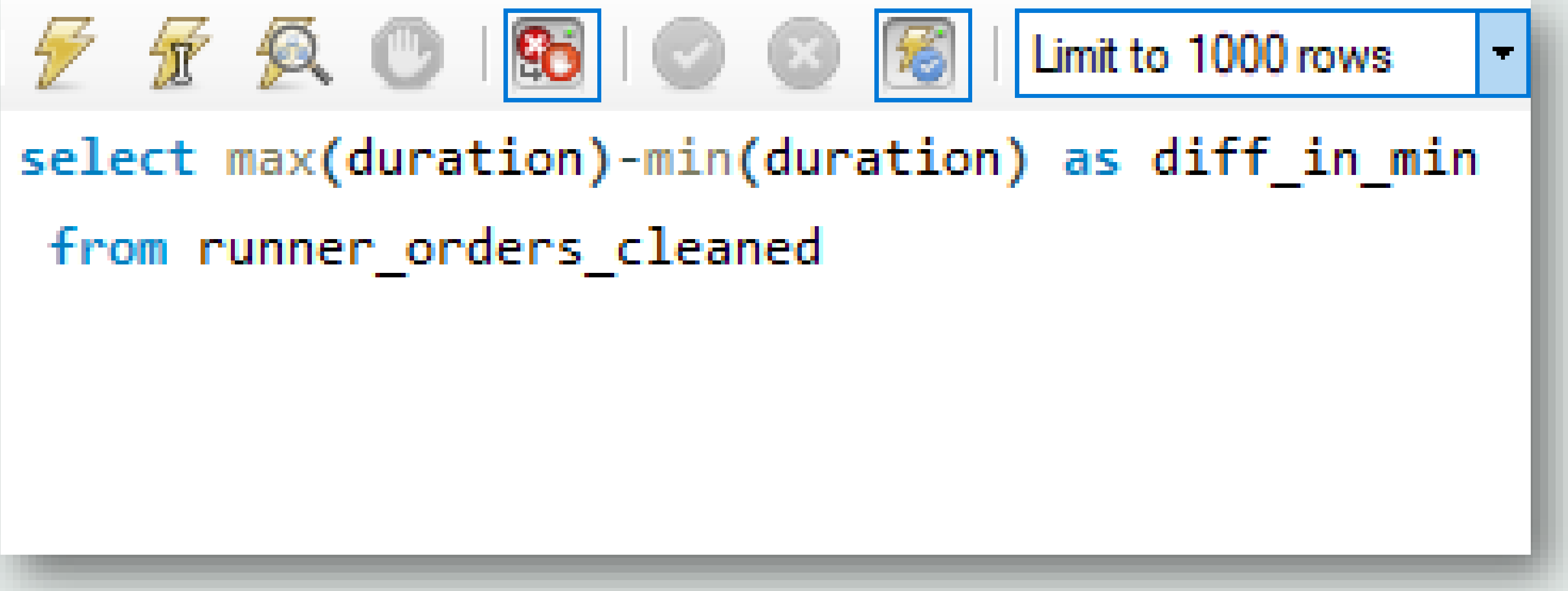


```
select c.customer_id,round(avg(distance),2) as avg_dist
 from customer_orders_cleaned c
join runner_orders_cleaned r on c.order_id=r.order_id
group by c.customer_id
```



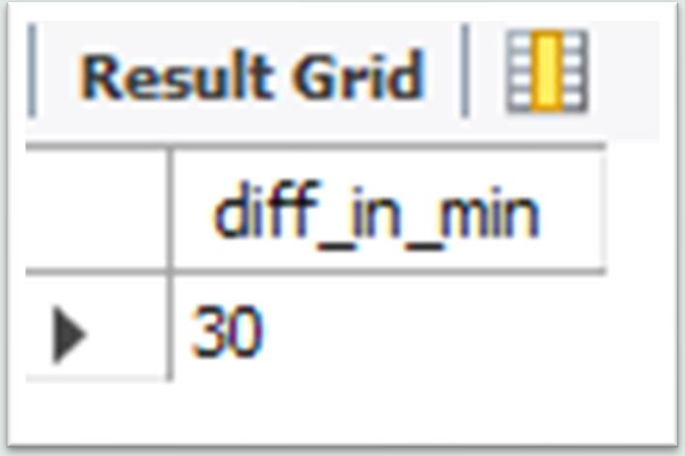
customer_id	avg_dist
101	20
102	16.73
103	23.4
104	10
105	25

5.What was the difference between the longest and shortest delivery times for all orders?



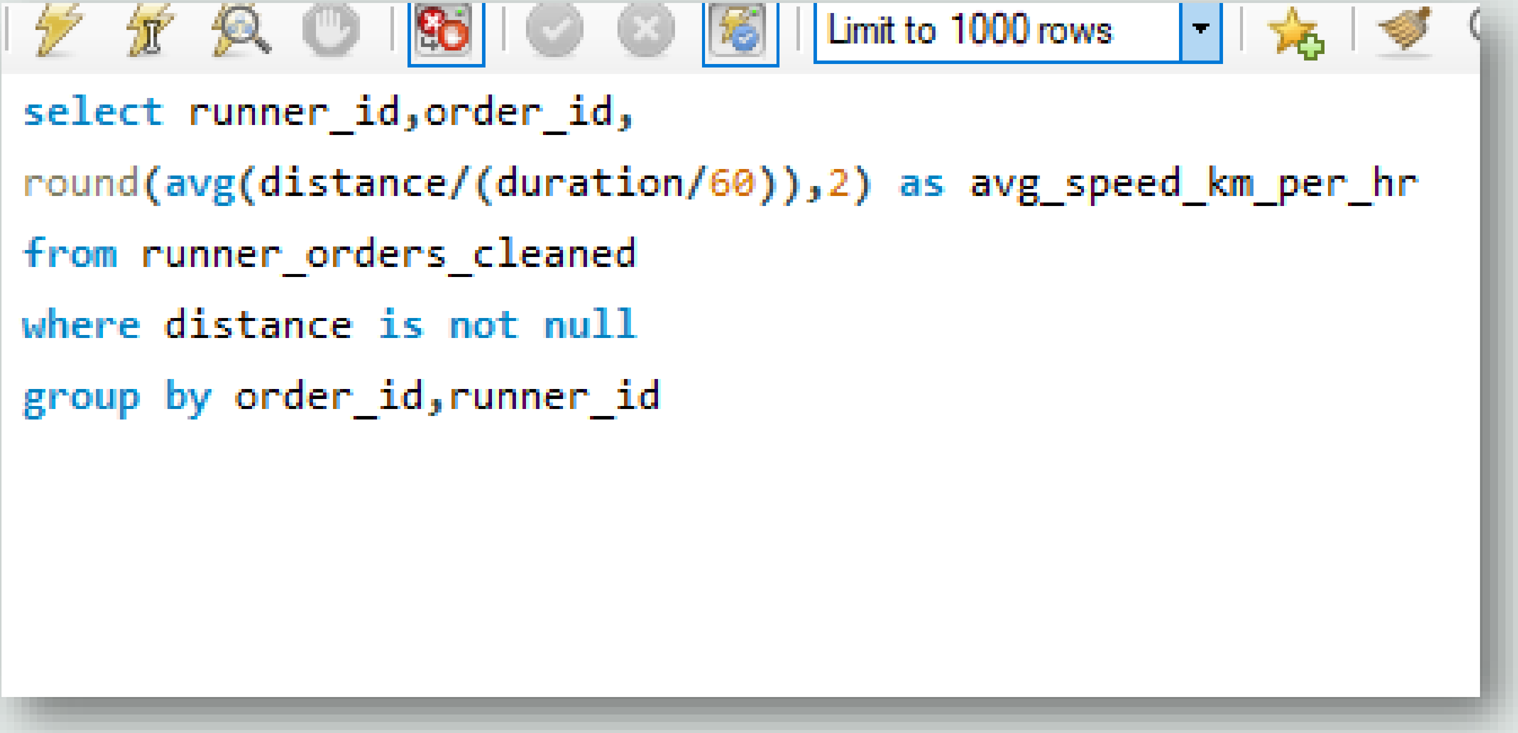
Limit to 1000 rows

```
select max(duration)-min(duration) as diff_in_min
 from runner_orders_cleaned
```

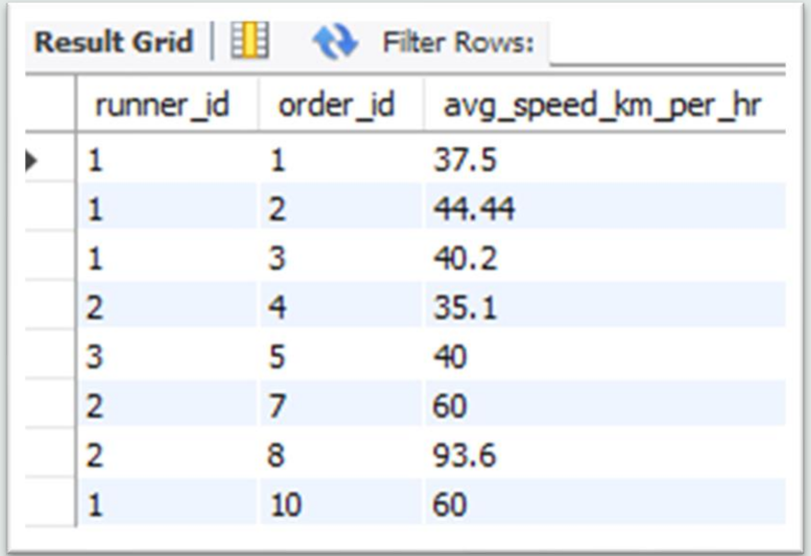


diff_in_min
30

6.What was the average speed for each runner for each delivery and do you notice any trend for these values?

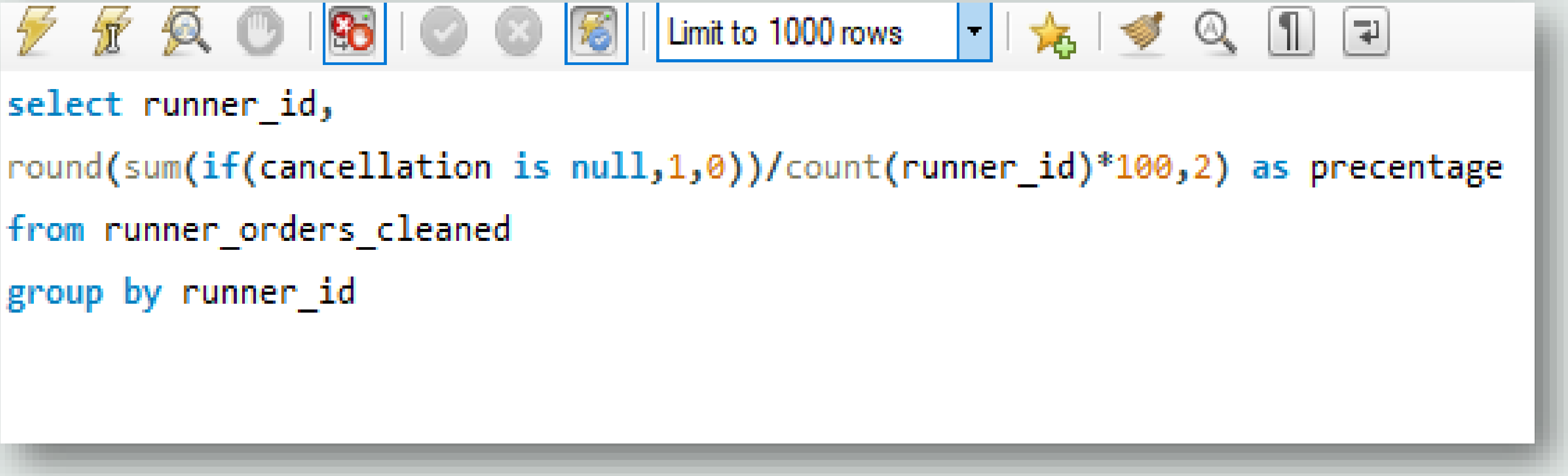


```
select runner_id,order_id,
round(avg(distance/(duration/60)),2) as avg_speed_km_per_hr
from runner_orders_cleaned
where distance is not null
group by order_id,runner_id
```

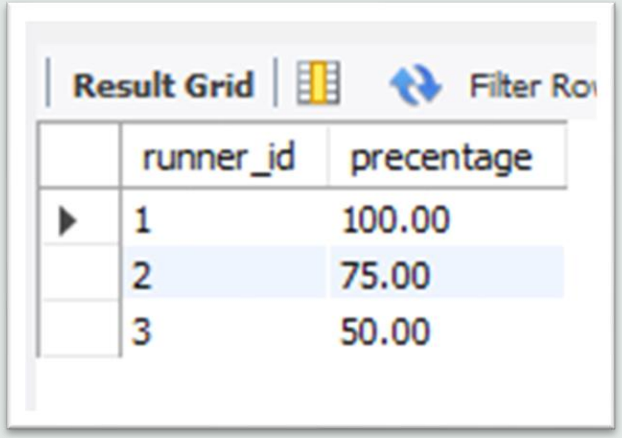


runner_id	order_id	avg_speed_km_per_hr
1	1	37.5
1	2	44.44
1	3	40.2
2	4	35.1
3	5	40
2	7	60
2	8	93.6
1	10	60

7.What is the successful delivery percentage for each runner?



```
select runner_id,
round(sum(if(cancellation is null,1,0))/count(runner_id)*100,2) as precentage
from runner_orders_cleaned
group by runner_id
```



runner_id	precentage
1	100.00
2	75.00
3	50.00

Thank You