DATAWITHDANNY.COM
START YOUR SQL ENGINES
PIZZA
RUNNER
CASE STUDY #2
8 WEEK SQL
CHALLENGE
8WEEKSQLCHALLENGE.COM

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.

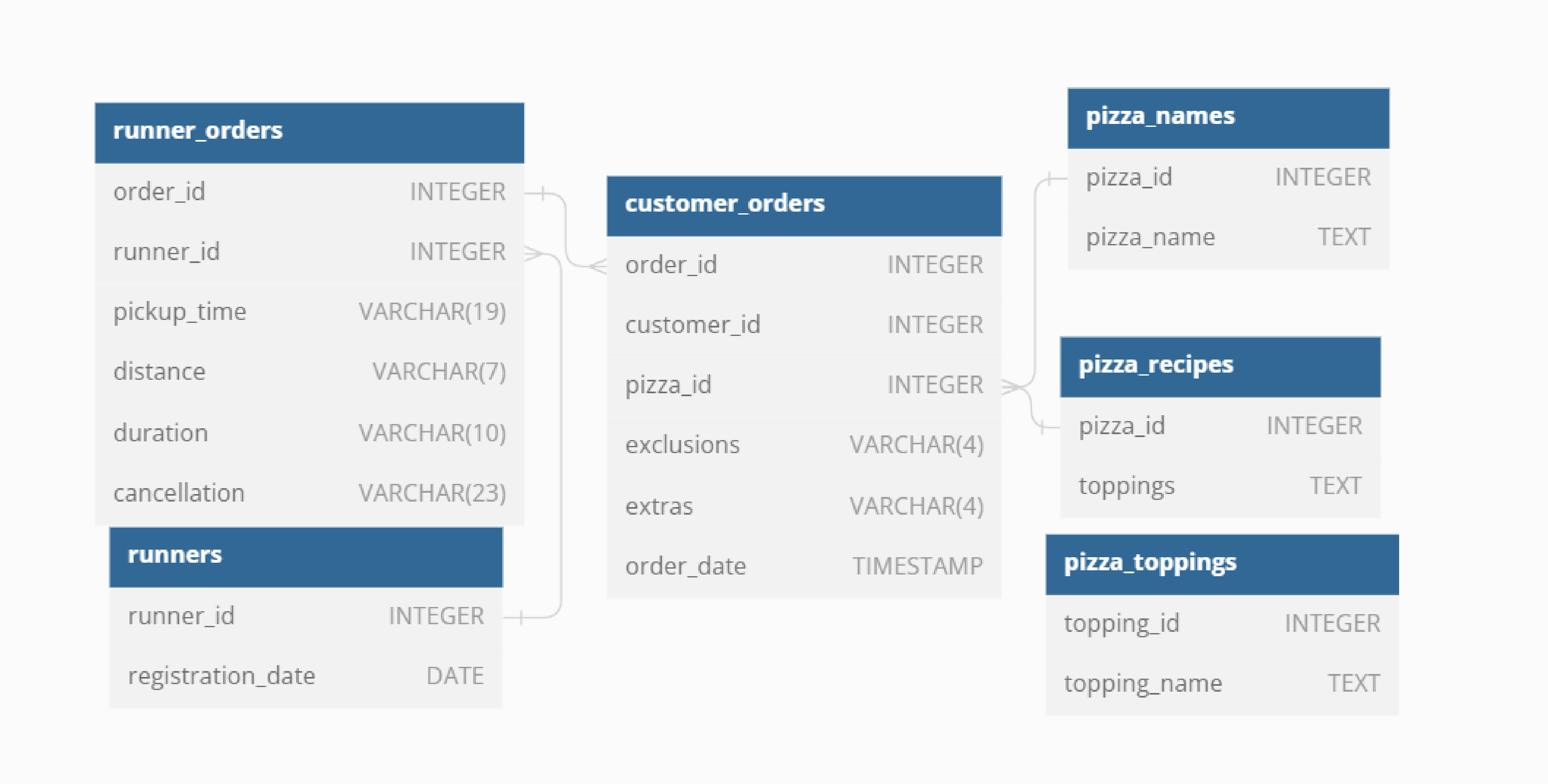
Available Data

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

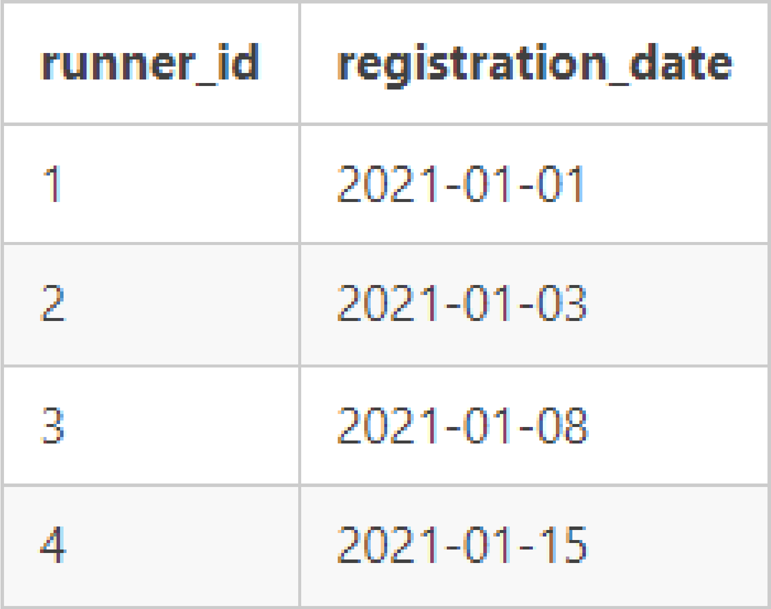
pizza_names	
pizza_id	INTEGER
pizza_name	TEXT

pizza_recipes	
pizza_id	INTEGER
toppings	TEXT

pizza_toppings	
topping_id	INTEGER
topping_name	TEXT

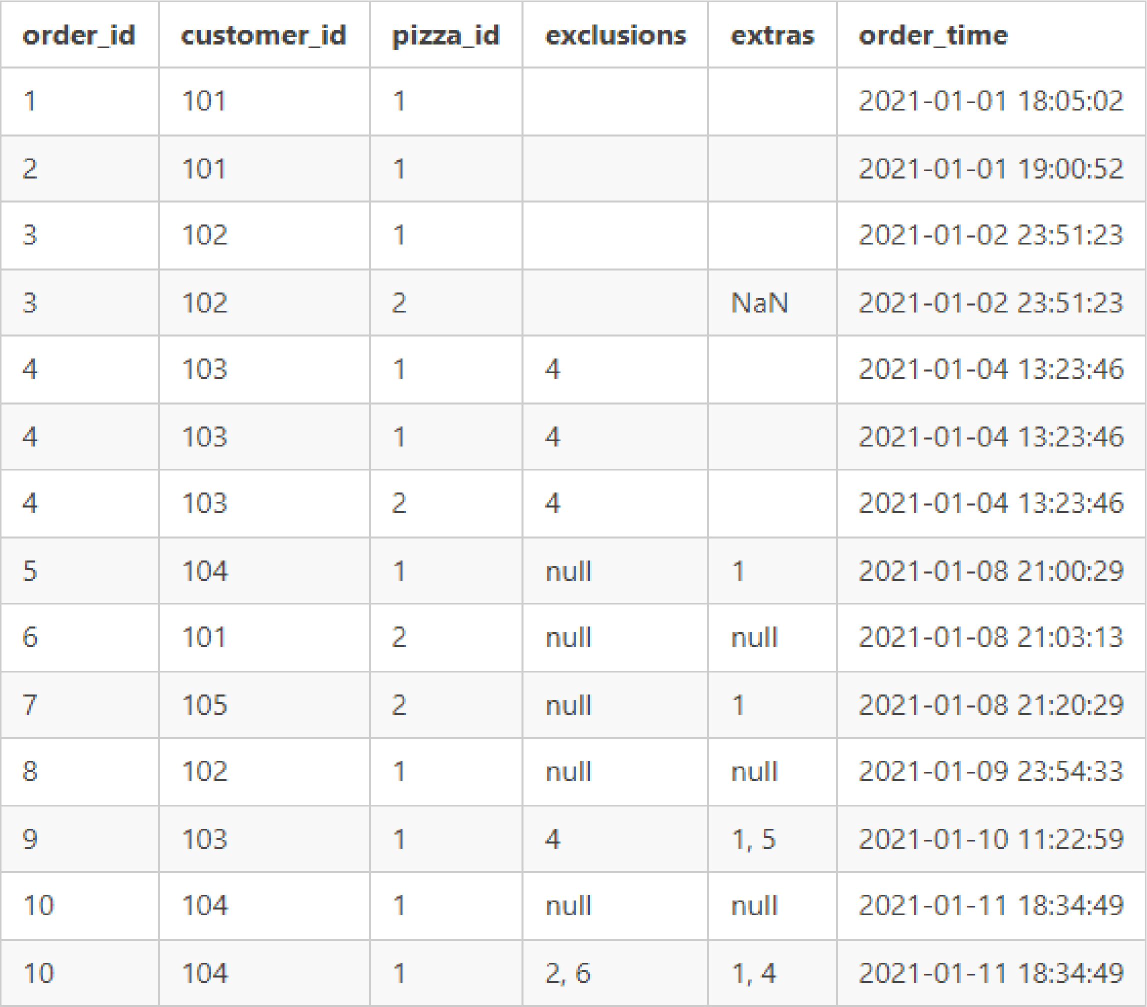
Table 1 : Runners

The runners table shows the registration_date for each new runner



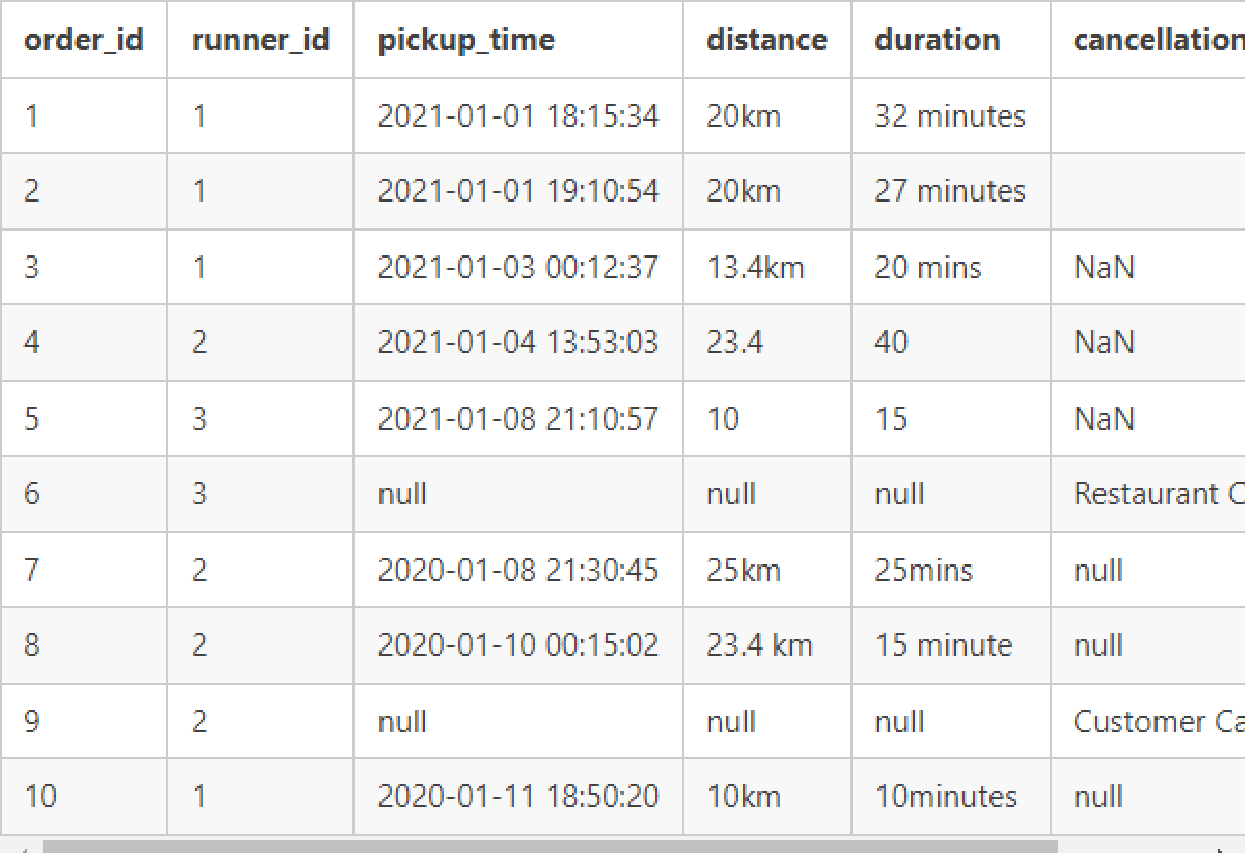
runner_id	registration_date
1	2021-01-01
2	2021-01-03
3	2021-01-08
4	2021-01-15

Table 2: customer_orders



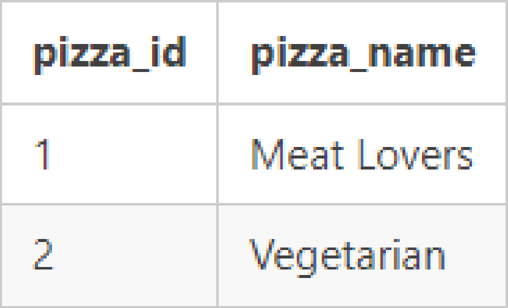
order_id	customer_id	pizza_id	exclusions	extras	order_time
1	101	1			2021-01-01 18:05:02
2	101	1			2021-01-01 19:00:52
3	102	1			2021-01-02 23:51:23
3	102	2		NaN	2021-01-02 23:51:23
4	103	1	4		2021-01-04 13:23:46
4	103	1	4		2021-01-04 13:23:46
4	103	2	4		2021-01-04 13:23:46
5	104	1	null	1	2021-01-08 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



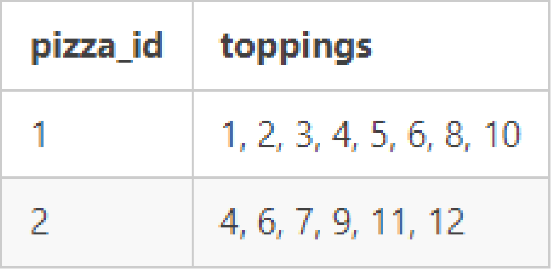
order_id	runner_id	pickup_time	distance	duration	cancellation
1	1	2021-01-01 18:15:34	20km	32 minutes	
2	1	2021-01-01 19:10:54	20km	27 minutes	
3	1	2021-01-03 00:12:37	13.4km	20 mins	NaN
4	2	2021-01-04 13:53:03	23.4	40	NaN
5	3	2021-01-08 21:10:57	10	15	NaN
6	3	null	null	null	Restaurant C
7	2	2020-01-08 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 Ca
10	1	2020-01-11 18:50:20	10km	10minutes	null

Table 4: pizza_names



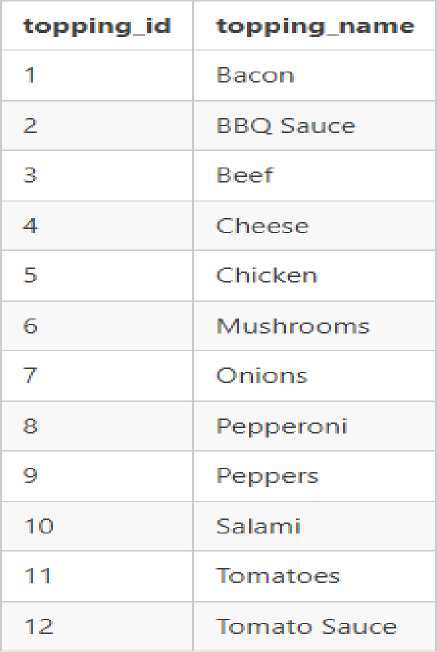
pizza_id	pizza_name
1	Meat Lovers
2	Vegetarian

Table 5:pizza_recipes



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

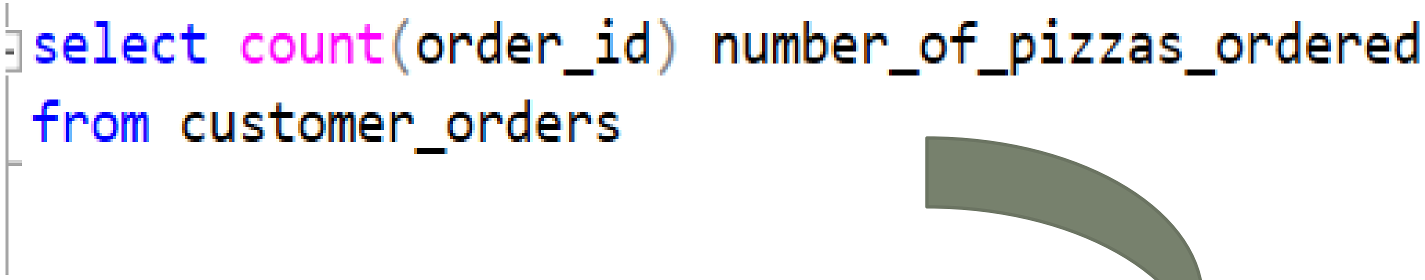
Table 6:pizza_toppings



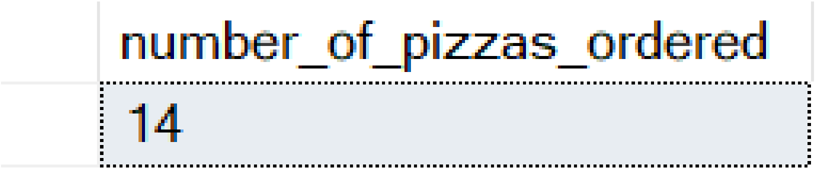
topping_id	topping_name
1	Bacon
2	BBQ Sauce
3	Beef
4	Cheese
5	Chicken
6	Mushrooms
7	Onions
8	Pepperoni
9	Peppers
10	Salami
11	Tomatoes
12	Tomato Sauce

A. PIZZA METRICS

QUES1.HOW MANY PIZZAS WERE ORDERED?

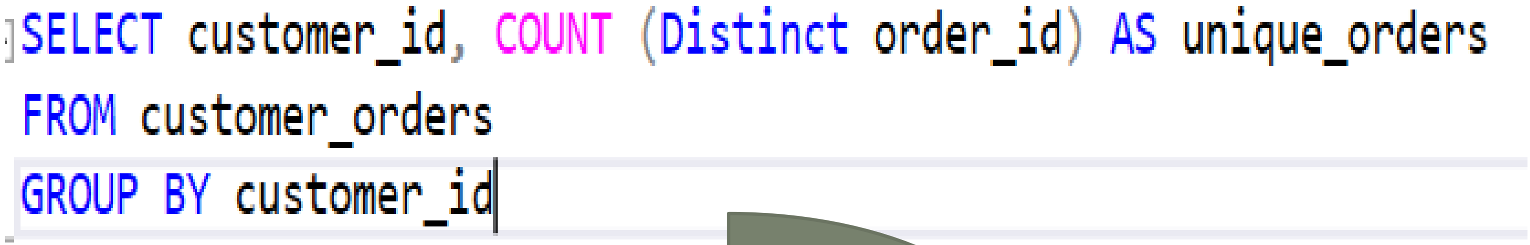


```sql
select count(order_id) number_of_pizzas_ordered
from customer_orders
```

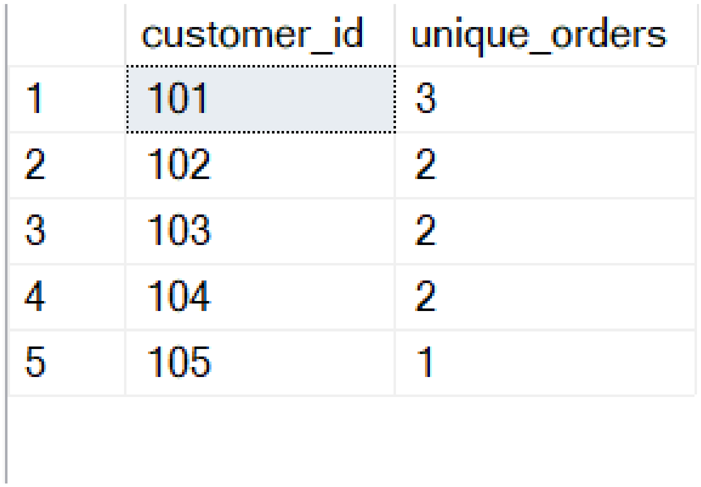


number_of_pizzas_ordered
14

Ques2.How many unique customer orders were made?

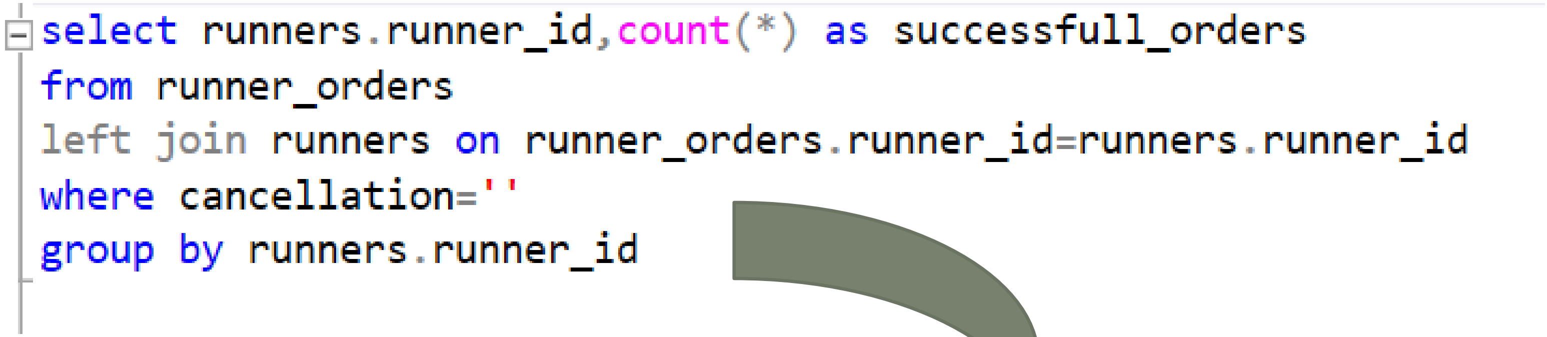


```
SELECT customer_id, COUNT (Distinct order_id) AS unique_orders
FROM customer_orders
GROUP BY customer_id
```

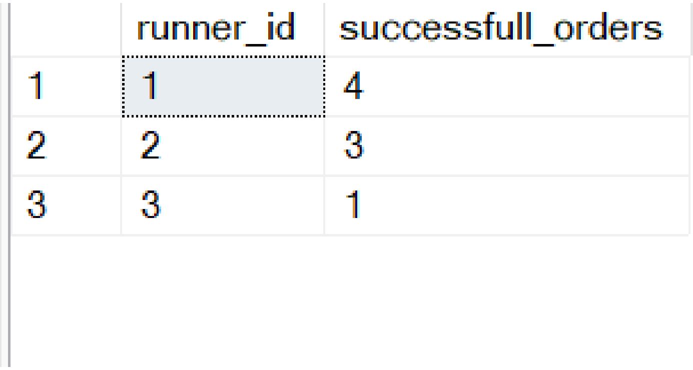


	customer_id	unique_orders
1	101	3
2	102	2
3	103	2
4	104	2
5	105	1

Ques3. How many successful orders were delivered by each runner?

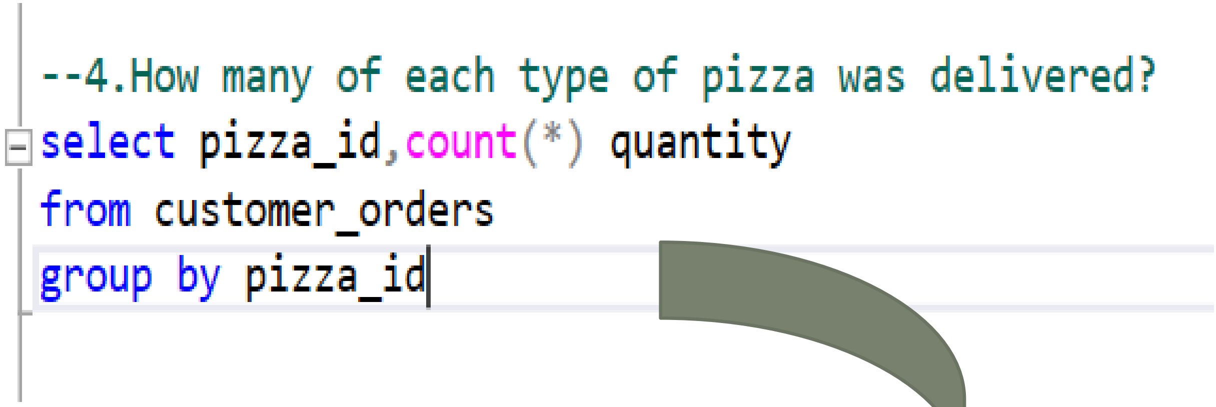


```
select runners.runner_id,count(*) as successfull_orders
from runner_orders
left join runners on runner_orders.runner_id=runners.runner_id
where cancellation=''
group by runners.runner_id
```

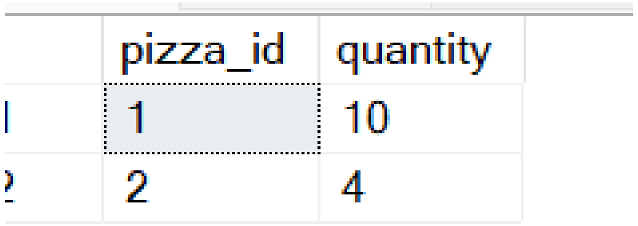


	runner_id	successfull_orders
1	1	4
2	2	3
3	3	1

Ques4. How many of each type of pizza was delivered?

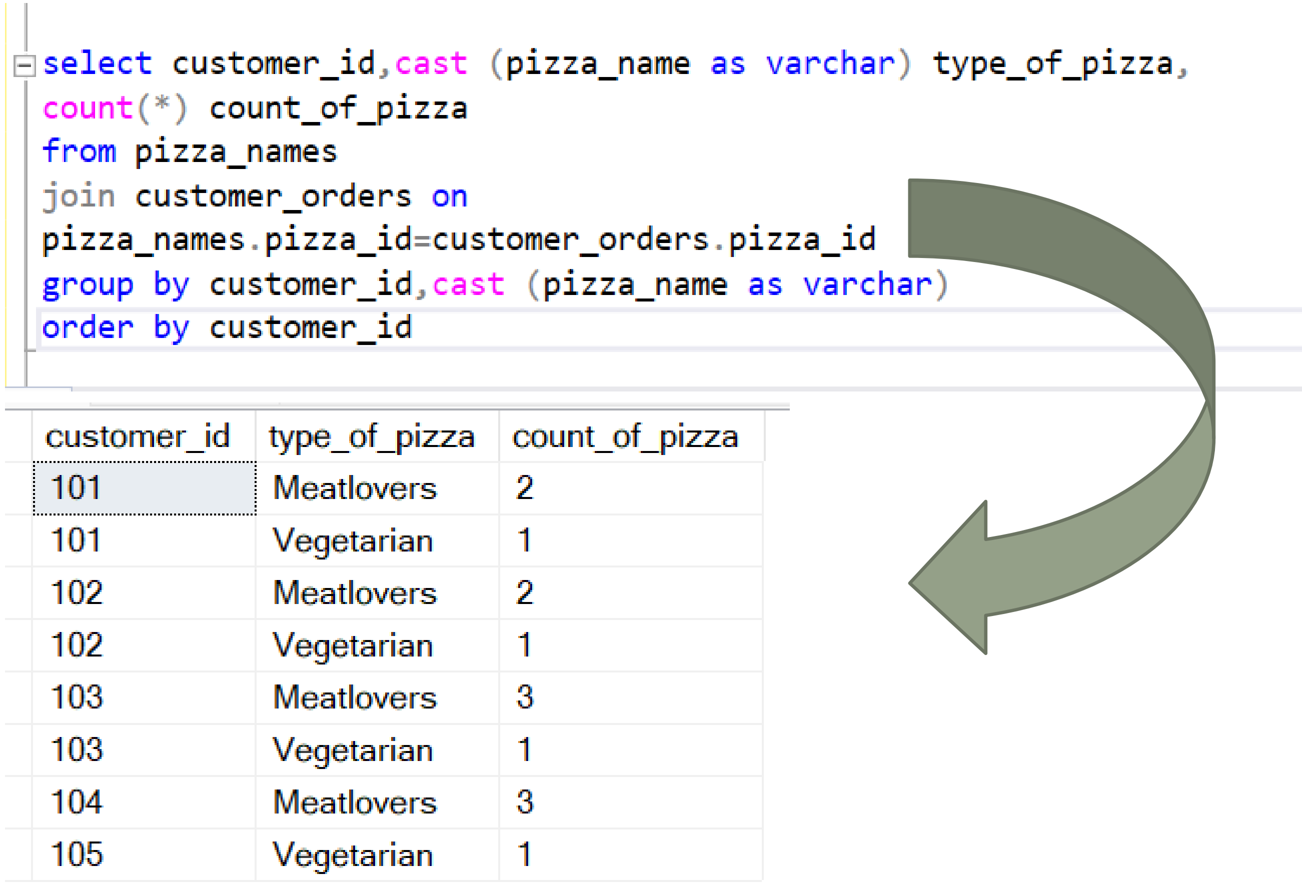


```
--4.How many of each type of pizza was delivered?
select pizza_id,count(*) quantity
from customer_orders
group by pizza_id
```



pizza_id	quantity
1	10
2	4

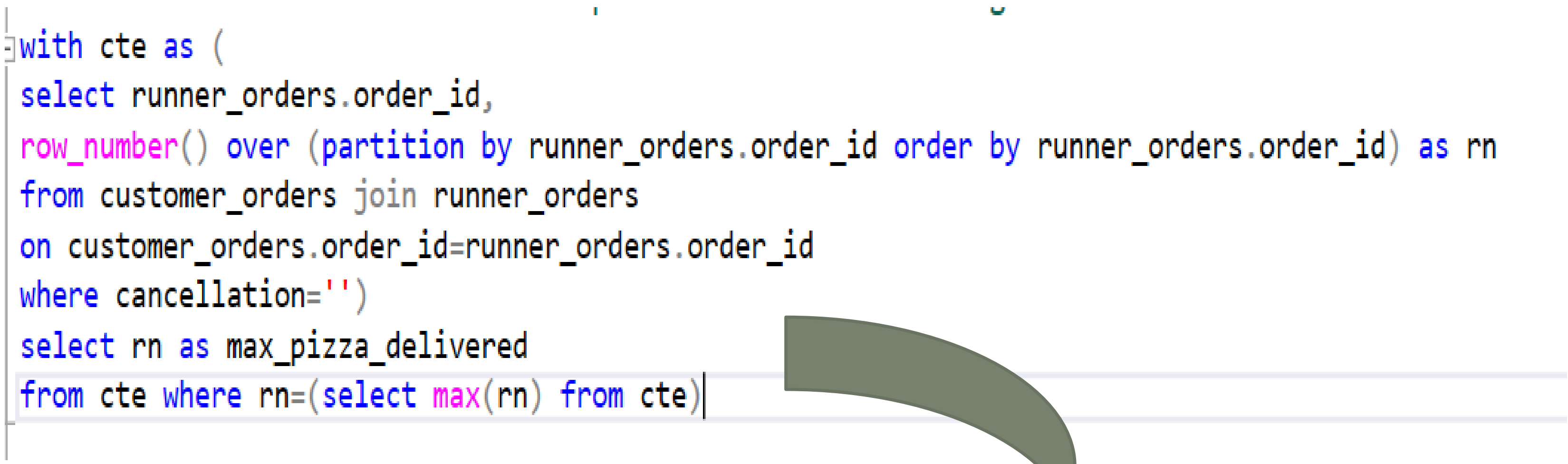
Ques5. How many Vegetarian and Meatlovers were ordered by each customer?



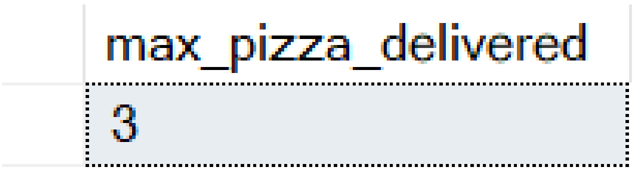
```sql
select customer_id,cast (pizza_name as varchar) type_of_pizza,
count(*) count_of_pizza
from pizza_names
join customer_orders on
pizza_names.pizza_id=customer_orders.pizza_id
group by customer_id,cast (pizza_name as varchar)
order by customer_id
```

customer_id	type_of_pizza	count_of_pizza
101	Meatlovers	2
101	Vegetarian	1
102	Meatlovers	2
102	Vegetarian	1
103	Meatlovers	3
103	Vegetarian	1
104	Meatlovers	3
105	Vegetarian	1

Ques6. What was the maximum number of pizzas delivered in a single order?

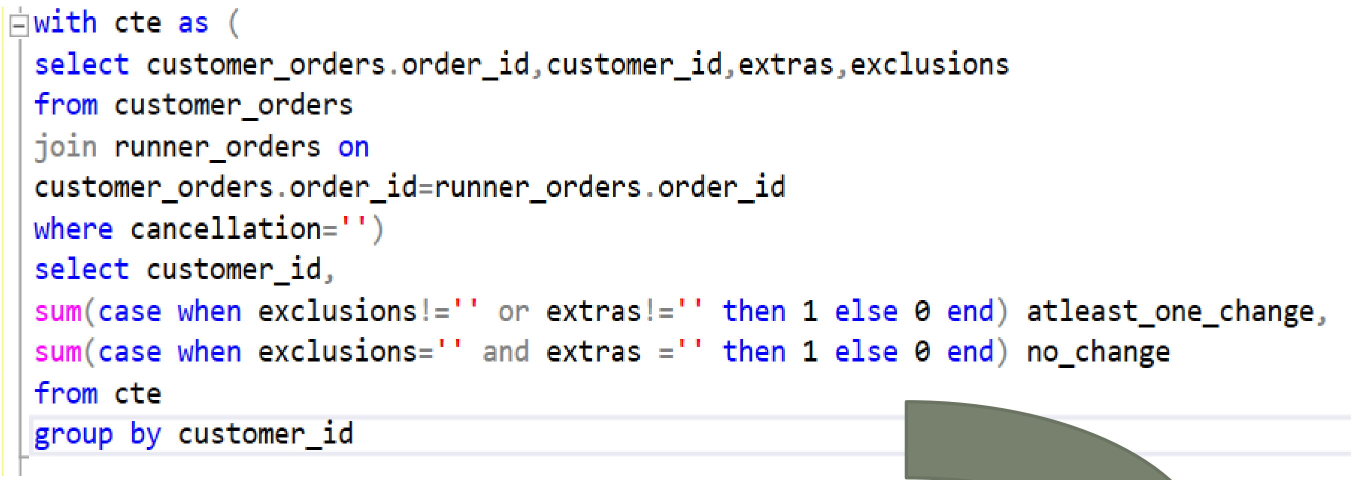


```
with cte as (
select runner_orders.order_id,
row_number() over (partition by runner_orders.order_id order by runner_orders.order_id) as rn
from customer_orders join runner_orders
on customer_orders.order_id=runner_orders.order_id
where cancellation='')
select rn as max_pizza_delivered
from cte where rn=(select max(rn) from cte)
```

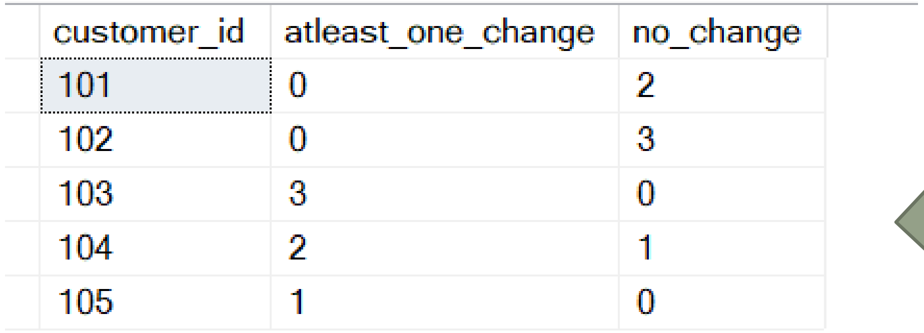


```
max_pizza_delivered
3
```

Ques7. For each customer, how many delivered pizzas had at least 1 change and how many had no changes?

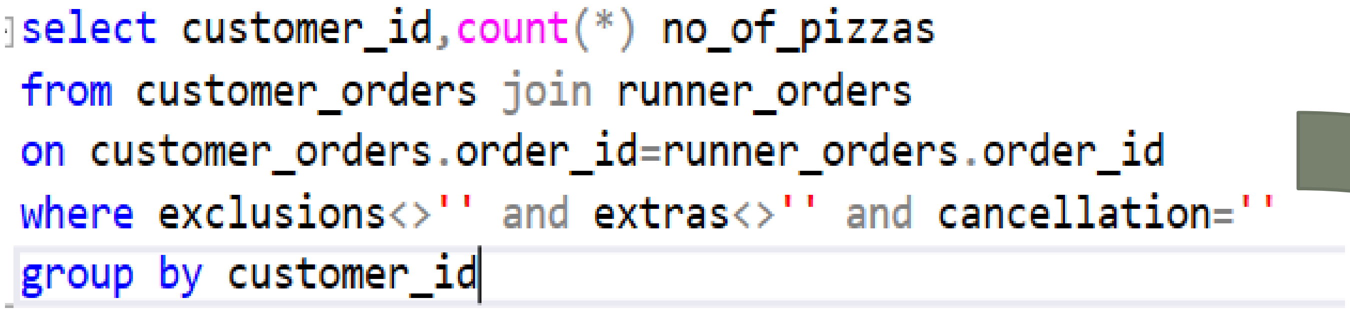


```
with cte as (
select customer_orders.order_id,customer_id,extras,exclusions
from customer_orders
join runner_orders on
customer_orders.order_id=runner_orders.order_id
where cancellation='')
select customer_id,
sum(case when exclusions!='' or extras!='' then 1 else 0 end) atleast_one_change,
sum(case when exclusions='' and extras ='' then 1 else 0 end) no_change
from cte
group by customer_id
```

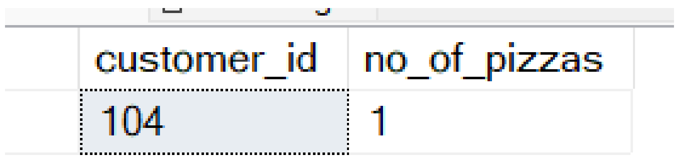


customer_id	atleast_one_change	no_change
101	0	2
102	0	3
103	3	0
104	2	1
105	1	0

Ques8.How many pizzas were delivered that had both exclusions and extras?

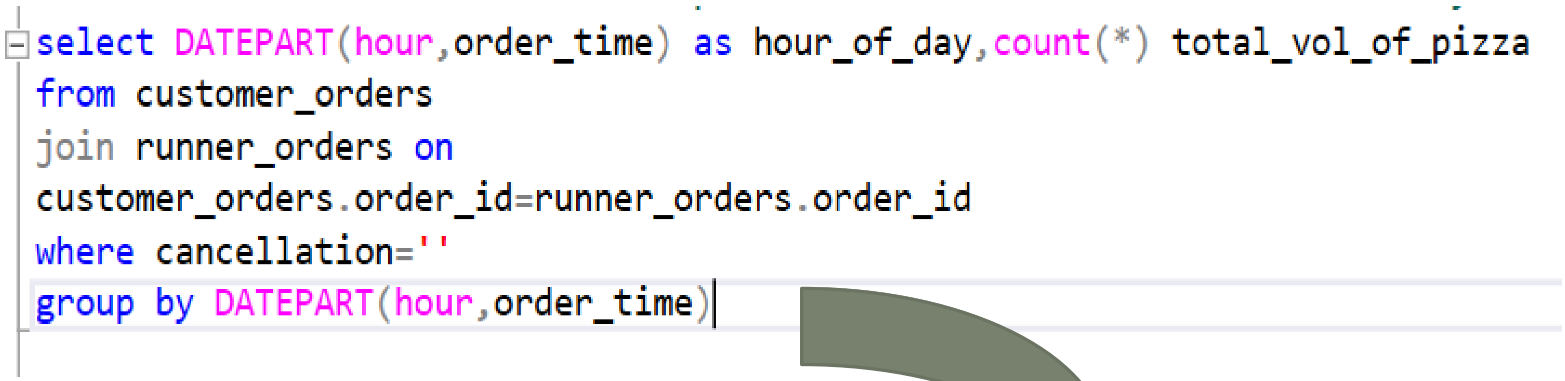


```
select customer_id,count(*) no_of_pizzas
from customer_orders join runner_orders
on customer_orders.order_id=runner_orders.order_id
where exclusions<>'' and extras<>'' and cancellation=''
group by customer_id
```

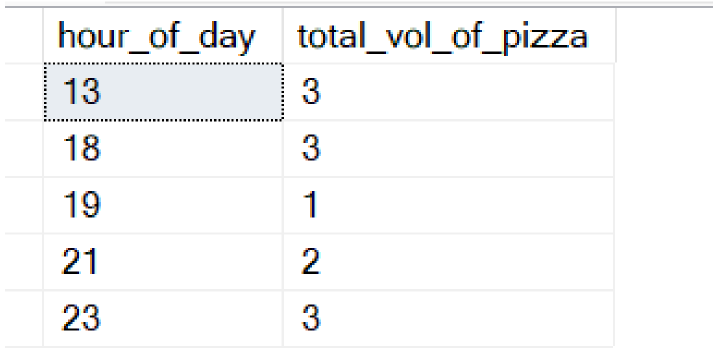


customer_id	no_of_pizzas
104	1

Ques9.What was the total volume of pizzas ordered for each hour of the day?

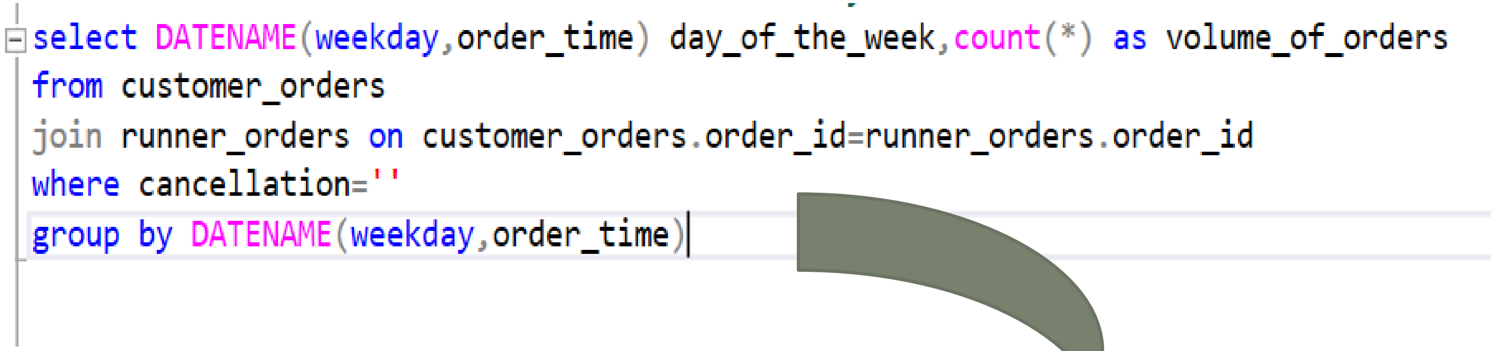


```
select DATEPART(hour,order_time) as hour_of_day,count(*) total_vol_of_pizza
from customer_orders
join runner_orders on
customer_orders.order_id=runner_orders.order_id
where cancellation=''
group by DATEPART(hour,order_time)
```

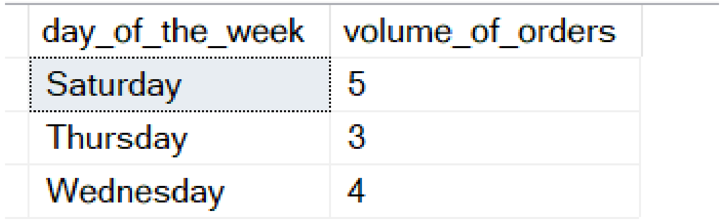


hour_of_day	total_vol_of_pizza
13	3
18	3
19	1
21	2
23	3

Ques10.What was the volume of orders for each day of the week?



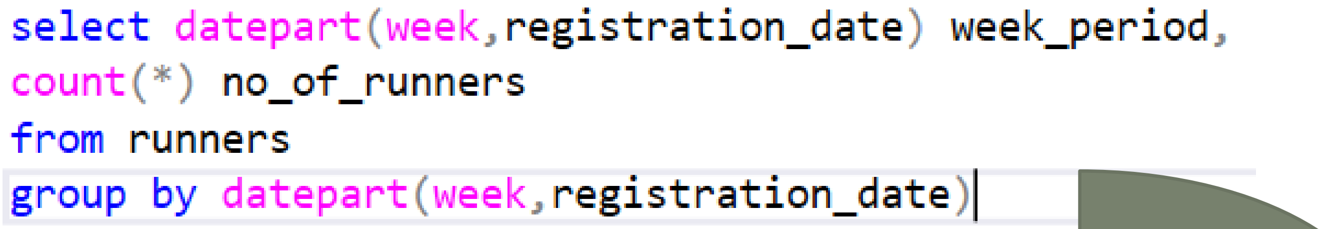
```
select DATENAME(weekday,order_time) day_of_the_week,count(*) as volume_of_orders
from customer_orders
join runner_orders on customer_orders.order_id=runner_orders.order_id
where cancellation=''
group by DATENAME(weekday,order_time)
```



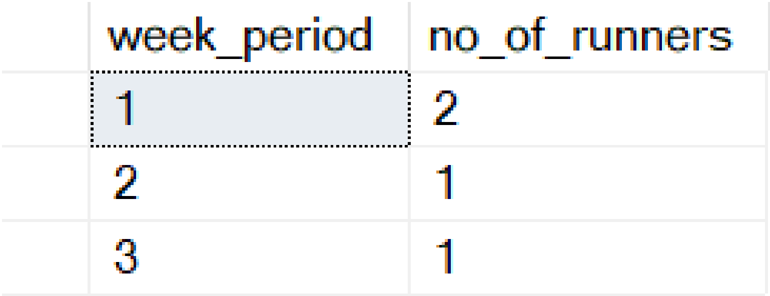
day_of_the_week	volume_of_orders
Saturday	5
Thursday	3
Wednesday	4

B. Runner and Customer Experience

Ques1. How many runners signed up for each 1 week period?

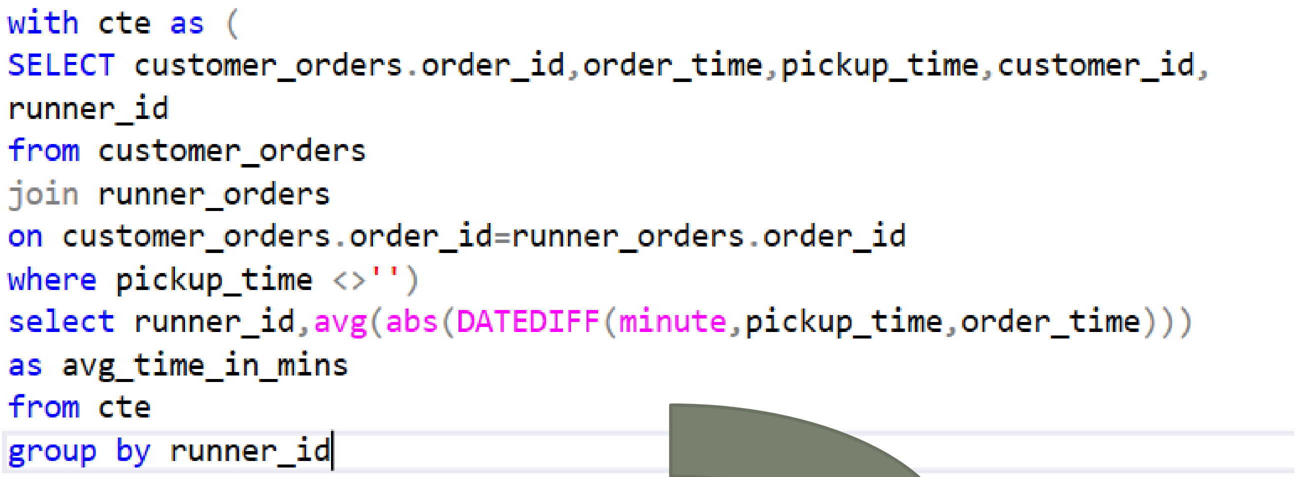


```
select datepart(week,registration_date) week_period,
count(*) no_of_runners
from runners
group by datepart(week,registration_date)
```

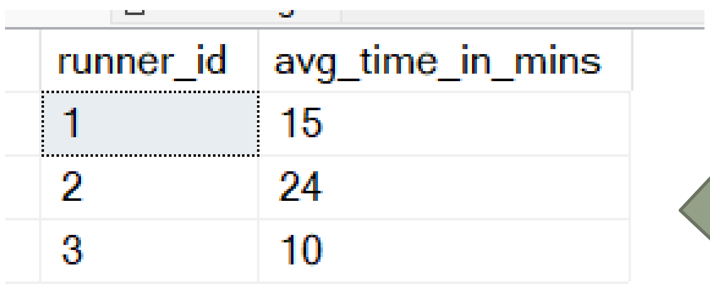


week_period	no_of_runners
1	2
2	1
3	1

Ques2. What was the average time in minutes it took for each runner to arrive at the Pizza Runner HQ to pickup the order?

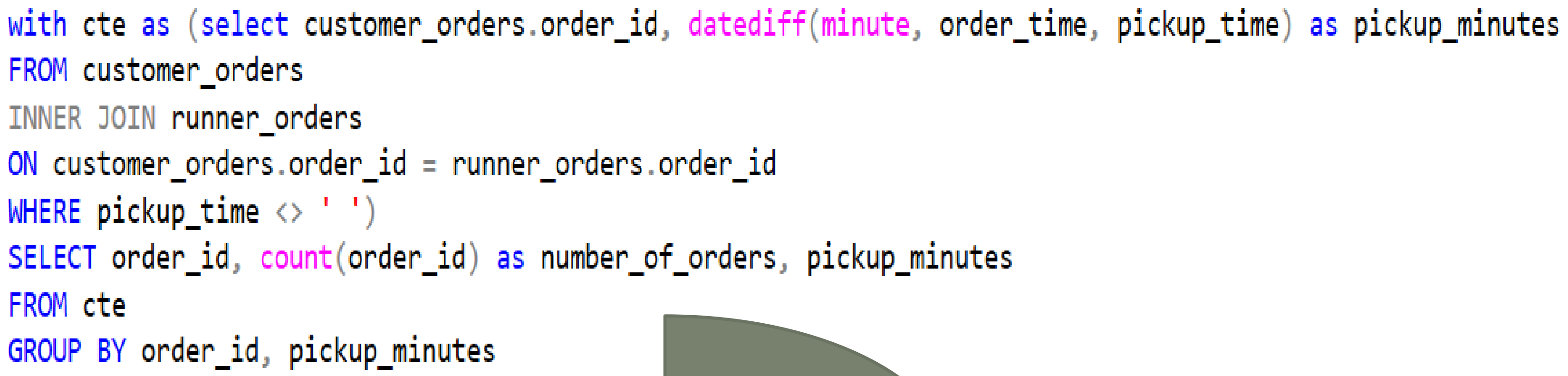


```
with cte as (
SELECT customer_orders.order_id,order_time,pickup_time,customer_id,
runner_id
from customer_orders
join runner_orders
on customer_orders.order_id=runner_orders.order_id
where pickup_time <>'')
select runner_id,avg(abs(DATEDIFF(minute,pickup_time,order_time)))
as avg_time_in_mins
from cte
group by runner_id
```

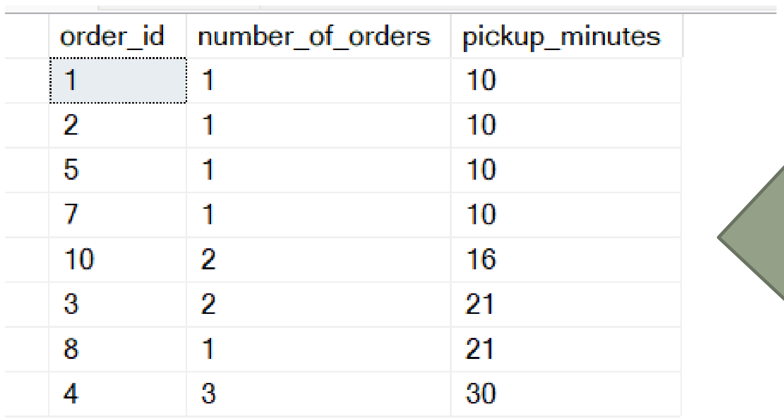


runner_id	avg_time_in_mins
1	15
2	24
3	10

Ques3. Is there any relationship between the number of pizzas and how long the order takes to prepare?

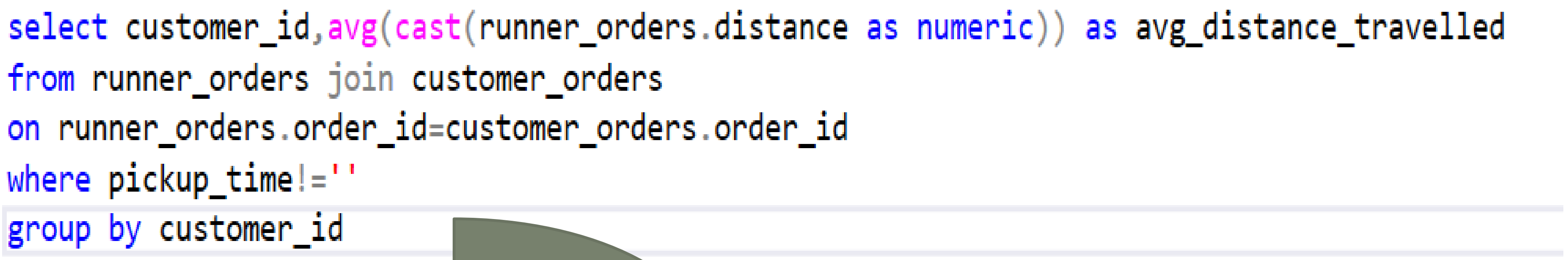


```sql
with cte as (select customer_orders.order_id, datediff(minute, order_time, pickup_time) as pickup_minutes
FROM customer_orders
INNER JOIN runner_orders
ON customer_orders.order_id = runner_orders.order_id
WHERE pickup_time <> ' ')
SELECT order_id, count(order_id) as number_of_orders, pickup_minutes
FROM cte
GROUP BY order_id, pickup_minutes
```

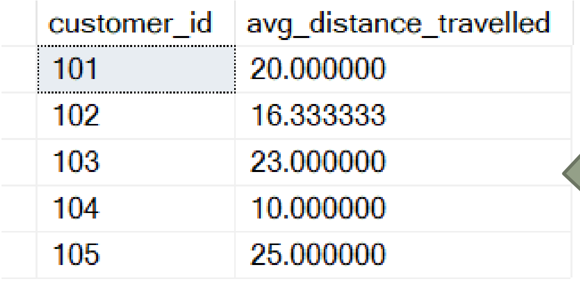


order_id	number_of_orders	pickup_minutes
1	1	10
2	1	10
5	1	10
7	1	10
10	2	16
3	2	21
8	1	21
4	3	30

Ques4. What was the average distance travelled for each customer?

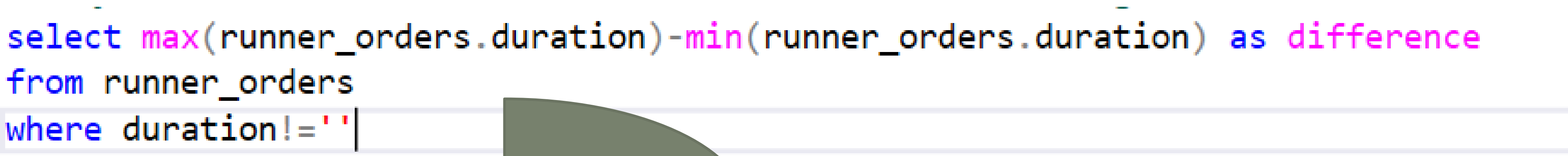


```
select customer_id,avg(cast(runner_orders.distance as numeric)) as avg_distance_travelled
from runner_orders join customer_orders
on runner_orders.order_id=customer_orders.order_id
where pickup_time!=''
group by customer_id
```

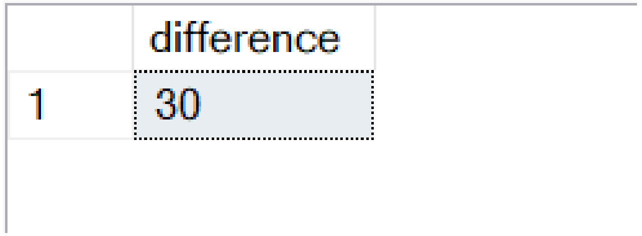


customer_id	avg_distance_travelled
101	20.000000
102	16.333333
103	23.000000
104	10.000000
105	25.000000

Ques5. What was the difference between the longest and shortest delivery times for all orders?

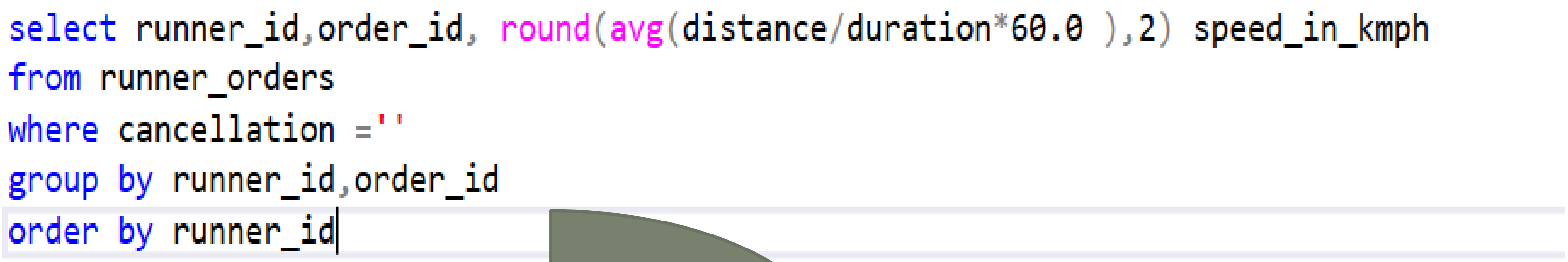


```
select max(runner_orders.duration)-min(runner_orders.duration) as difference
from runner_orders
where duration!=''
```

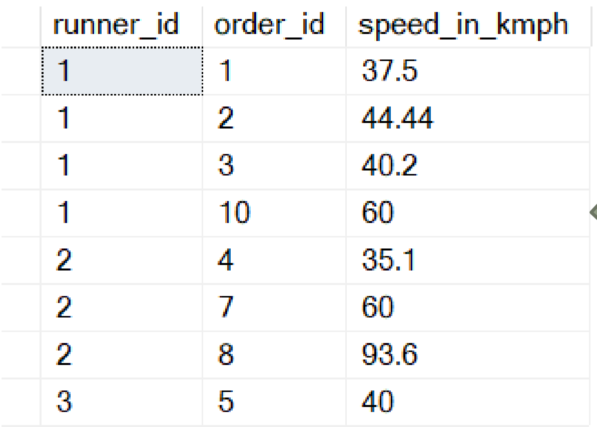


	difference
1	30

Ques6. What was the average speed for each runner for each delivery and do you notice any trend for these values?

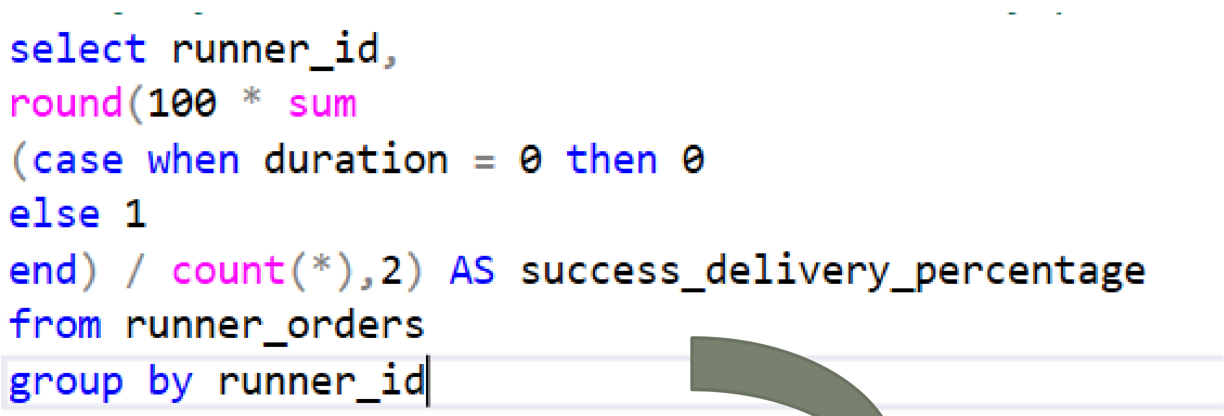


```sql
select runner_id,order_id, round(avg(distance/duration*60.0 ),2) speed_in_kmph
from runner_orders
where cancellation =''
group by runner_id,order_id
order by runner_id
```

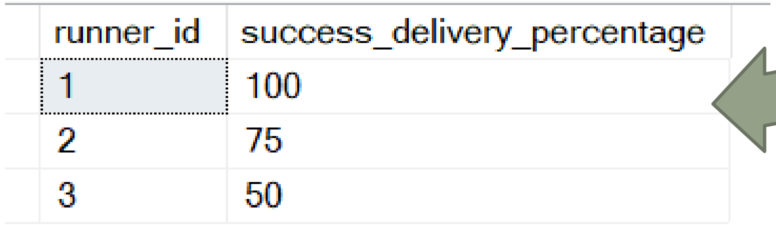


runner_id	order_id	speed_in_kmph
1	1	37.5
1	2	44.44
1	3	40.2
1	10	60
2	4	35.1
2	7	60
2	8	93.6
3	5	40

Ques7.What is the successful delivery percentage for each runner?



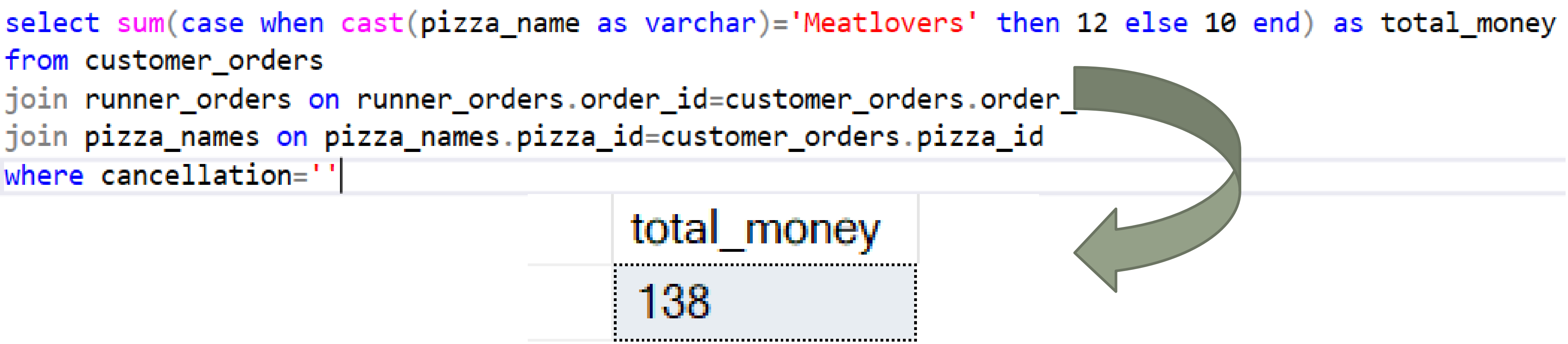
```
select runner_id,
round(100 * sum
(case when duration = 0 then 0
else 1
end) / count(*),2) AS success_delivery_percentage
from runner_orders
group by runner_id
```



runner_id	success_delivery_percentage
1	100
2	75
3	50

D. Pricing and Ratings

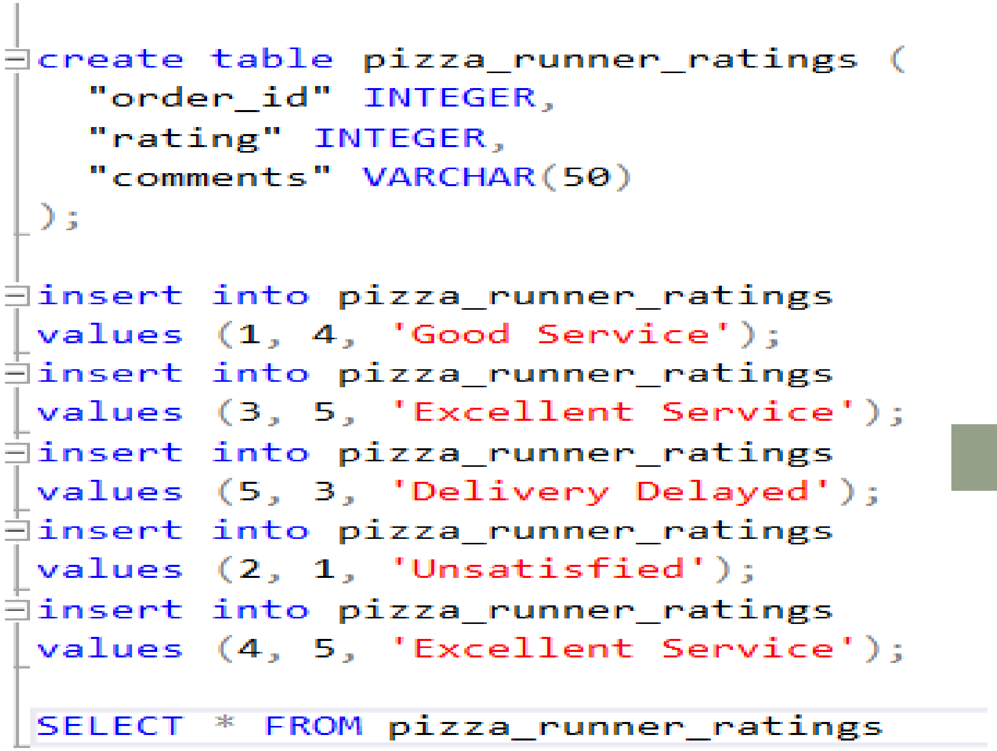
Ques1. If a Meat Lovers pizza costs $12 and Vegetarian costs $10 and there were no charges for changes - how much money has Pizza Runner made so far if there are no delivery fees?



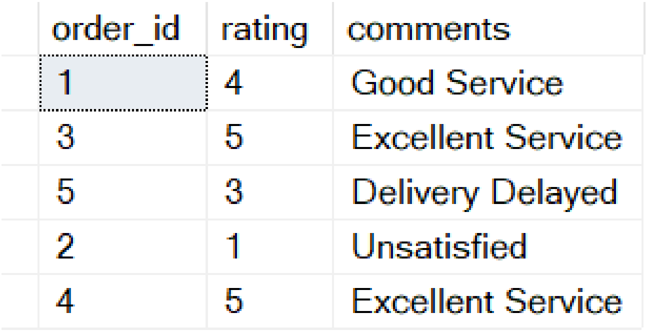
```sql
select sum(case when cast(pizza_name as varchar)='Meatlovers' then 12 else 10 end) as total_money
from customer_orders
join runner_orders on runner_orders.order_id=customer_orders.order_
join pizza_names on pizza_names.pizza_id=customer_orders.pizza_id
where cancellation=''
```

```
total_money
138
```

Ques3.The Pizza Runner team now wants to add an additional ratings system that allows customers to rate their runner, how would you design an additional table for this new dataset - generate a schema for this new table and insert your own data for ratings for each successful customer order between 1 to 5.

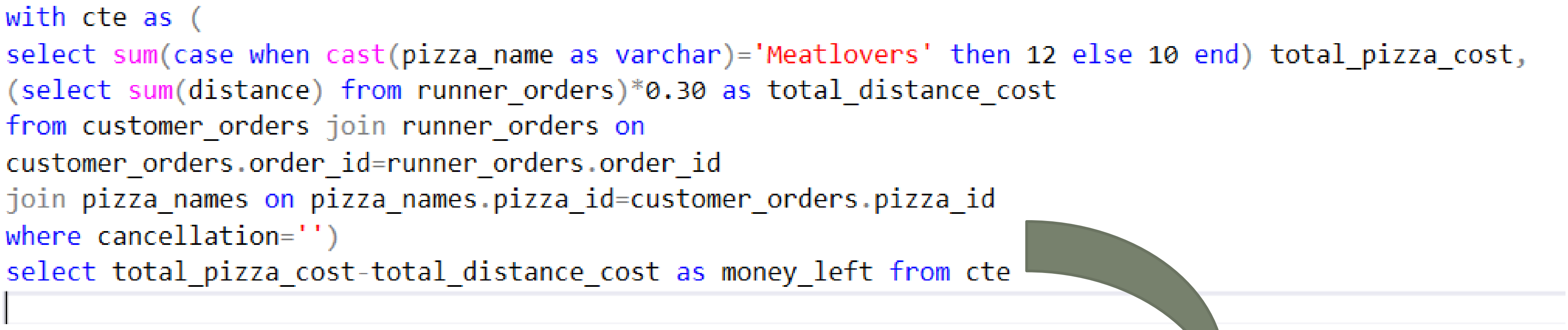


```
create table pizza_runner_ratings (
   "order_id" INTEGER,
   "rating" INTEGER,
   "comments" VARCHAR(50)
);

insert into pizza_runner_ratings
values (1, 4, 'Good Service');
insert into pizza_runner_ratings
values (3, 5, 'Excellent Service');
insert into pizza_runner_ratings
values (5, 3, 'Delivery Delayed');
insert into pizza_runner_ratings
values (2, 1, 'Unsatisfied');
insert into pizza_runner_ratings
values (4, 5, 'Excellent Service');

SELECT * FROM pizza_runner_ratings
```

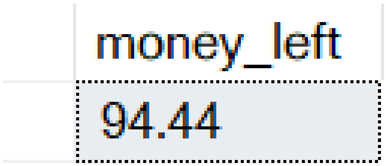


order_id	rating	comments
1	4	Good Service
3	5	Excellent Service
5	3	Delivery Delayed
2	1	Unsatisfied
4	5	Excellent Service

Ques4.If a Meat Lovers pizza was $12 and Vegetarian $10 fixed prices with no cost for extras and each runner is paid $0.30 per kilometre traveled - how much money does Pizza Runner have left over after these deliveries?



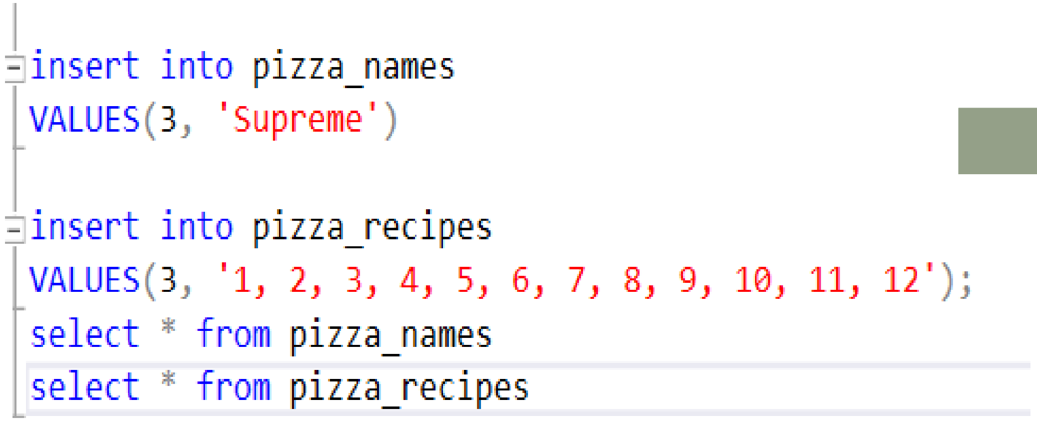
```
with cte as (
select sum(case when cast(pizza_name as varchar)='Meatlovers' then 12 else 10 end) total_pizza_cost,
(select sum(distance) from runner_orders)*0.30 as total_distance_cost
from customer_orders join runner_orders on
customer_orders.order_id=runner_orders.order_id
join pizza_names on pizza_names.pizza_id=customer_orders.pizza_id
where cancellation='')
select total_pizza_cost-total_distance_cost as money_left from cte
```



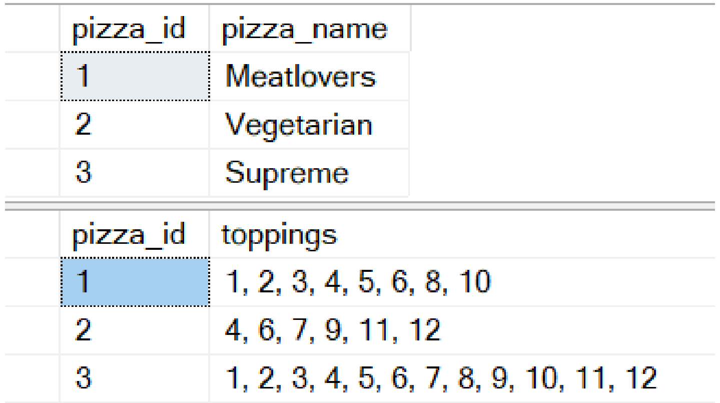
money_left
94.44

E. Bonus Questions

If Danny wants to expand his range of pizzas - how would this impact the existing data design? Write an INSERT statement to demonstrate what would happen if a new Supreme pizza with all the toppings was added to the Pizza Runner menu?



```
insert into pizza_names
VALUES(3, 'Supreme')

insert into pizza_recipes
VALUES(3, '1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12');
select * from pizza_names
select * from pizza_recipes
```



pizza_id	pizza_name
1	Meatlovers
2	Vegetarian
3	Supreme

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