Case Study - 2

Friday, January 28, 2022 9:53 AM

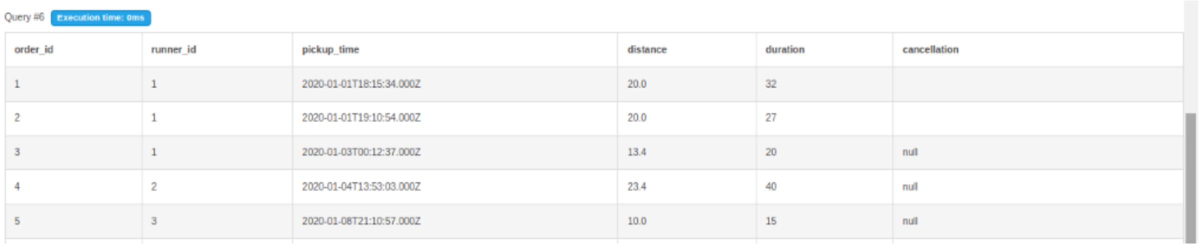
NOTE :- To clean the customer_orders table: I created a new table called customer_orders1 and cleaned it to avoid any data loss of the original table.

First Cleaned The Data

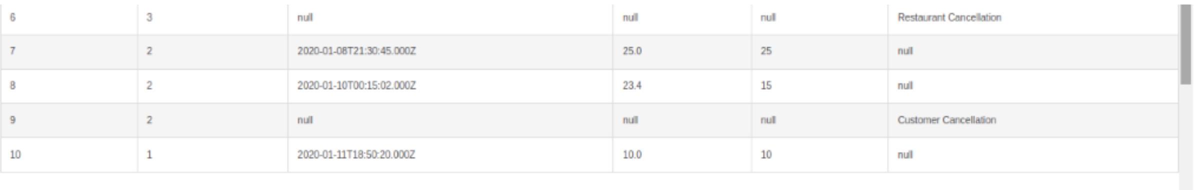
```
create table customer_orders1 as (select order_id, customer_id, pizza_id, exclusions, extras,
order_time
from pizza_runner.customer_orders);

update customer_orders1
set
exclusions = case exclusions when 'null' then null else exclusions end,
extras = case extras when 'null' then null else extras end;

create table runner_orders1 as
(select order_id, runner_id, pickup_time,
case
when distance like '%km' then trim('km' from distance)
else distance
end as distance,
case
when duration like '%minutes' then trim('minutes' from duration)
when duration like '%mins' then trim('mins' from duration)
when duration like '%minute' then trim('minute' from duration)
else duration
end as duration, cancellation
from pizza_runner.runner_orders);

update runner_orders1
set
pickup_time = case pickup_time when 'null' then null else pickup_time end,
distance = case distance when 'null' then null else distance end,
duration = case duration when 'null' then null else duration end,
cancellation = case cancellation when 'null' then null else cancellation end;

alter table runner_orders1
alter column pickup_time TYPE TIMESTAMP WITH TIME ZONE
    USING to_timestamp(pickup_time, 'YYYY-MM-DD HH24:MI:SS'),
alter column distance TYPE NUMERIC(5,1) USING(distance::NUMERIC),
alter column duration TYPE INT USING(duration::INT);
```

After The cleaning The data

```
select * from runner_orders1;
```

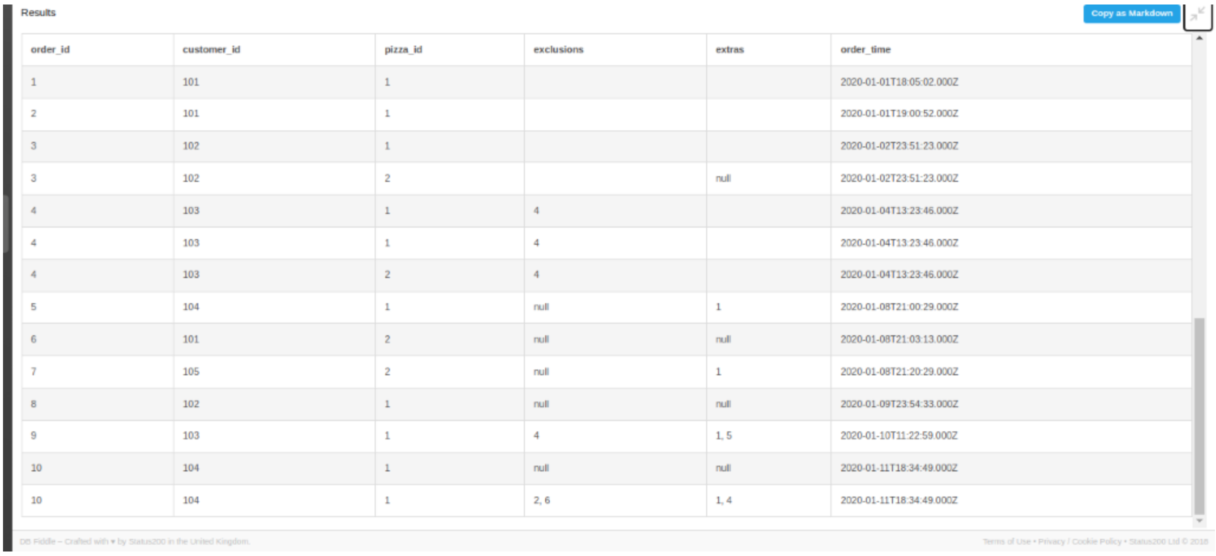


order_id	runner_id	pickup_time	distance	duration	cancellation
1	1	2020-01-01T18:15:34.000Z	20.0	32	
2	1	2020-01-01T19:10:54.000Z	20.0	27	
3	1	2020-01-03T00:12:37.000Z	13.4	20	null
4	2	2020-01-04T13:53:03.000Z	23.4	40	null
5	3	2020-01-08T21:10:57.000Z	10.0	15	null



6	3	null	null	null	Restaurant Cancellation
7	2	2020-01-08T21:30:45.000Z	25.0	25	null
8	2	2020-01-10T00:15:02.000Z	23.4	15	null
9	2	null	null	null	Customer Cancellation
10	1	2020-01-11T18:50:20.000Z	10.0	10	null

```
select * from pizza_runner.customer_orders;
```

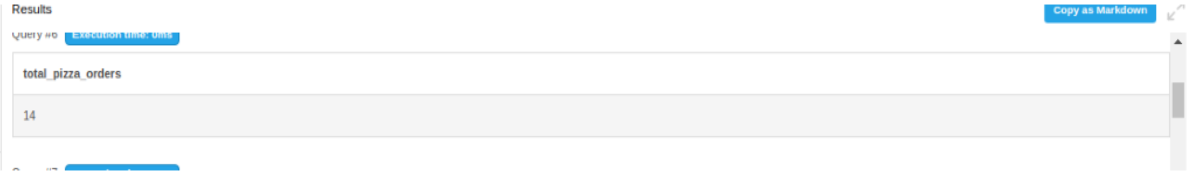

Results
Copy as Markdown
DB Fiddle – Crafted with ♥ by Status200 in the United Kingdom.
Terms of Use • Privacy / Cookie Policy • Status200 Ltd © 2018

order_id	customer_id	pizza_id	exclusions	extras	order_time
1	101	1			2020-01-01T18:05:02.000Z
2	101	1			2020-01-01T19:00:52.000Z
3	102	1			2020-01-02T23:51:23.000Z
3	102	2		null	2020-01-02T23:51:23.000Z
4	103	1	4		2020-01-04T13:23:46.000Z
4	103	1	4		2020-01-04T13:23:46.000Z
4	103	2	4		2020-01-04T13:23:46.000Z
5	104	1	null	1	2020-01-08T21:00:29.000Z
6	101	2	null	null	2020-01-08T21:03:13.000Z
7	105	2	null	1	2020-01-08T21:20:29.000Z
8	102	1	null	null	2020-01-09T23:54:33.000Z
9	103	1	4	1, 5	2020-01-10T11:22:59.000Z
10	104	1	null	null	2020-01-11T18:34:49.000Z
10	104	1	2, 6	1, 4	2020-01-11T18:34:49.000Z

Section - 1

1. How many pizzas were ordered?

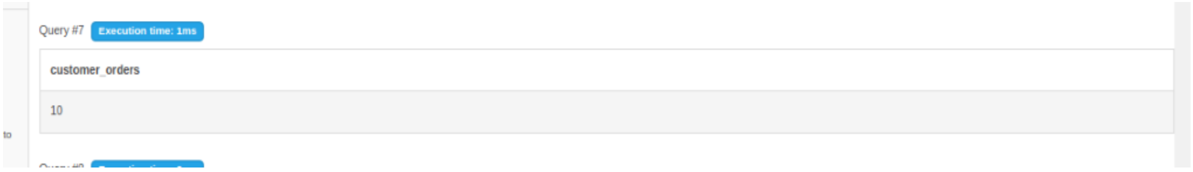
```
select count(pizza_id) As Total_Pizza_Orders
        from pizza_runner.customer_orders;
```



total_pizza_orders
14

2. How many unique customer orders were made?

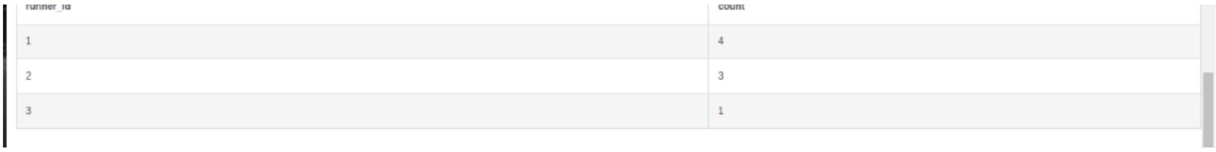
```
select count(distinct(order_id)) As Customer_Orders
        from pizza_runner.customer_orders;
```



customer_orders
10

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

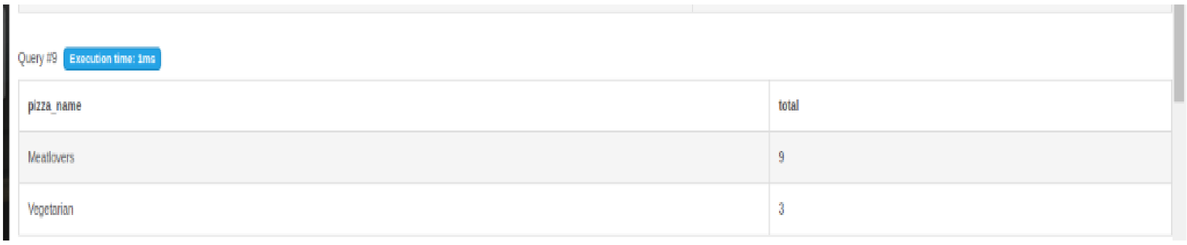
```
select runner_id,
        count(order_id) from runner_orders1
        where distance is not null
        group by runner_id
        ORDER BY runner_id;
```



runner_id	count
1	4
2	3
3	1

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

```
select p.pizza_name,
        count(c.pizza_id) As Total
        from customer_orders1 as c inner join runner_orders1 r
        on c.order_id=r.order_id inner join pizza_runner.pizza_names p
        on c.pizza_id=p.pizza_id
        where r.distance is not null
        group by p.pizza_id,p.pizza_name;
```

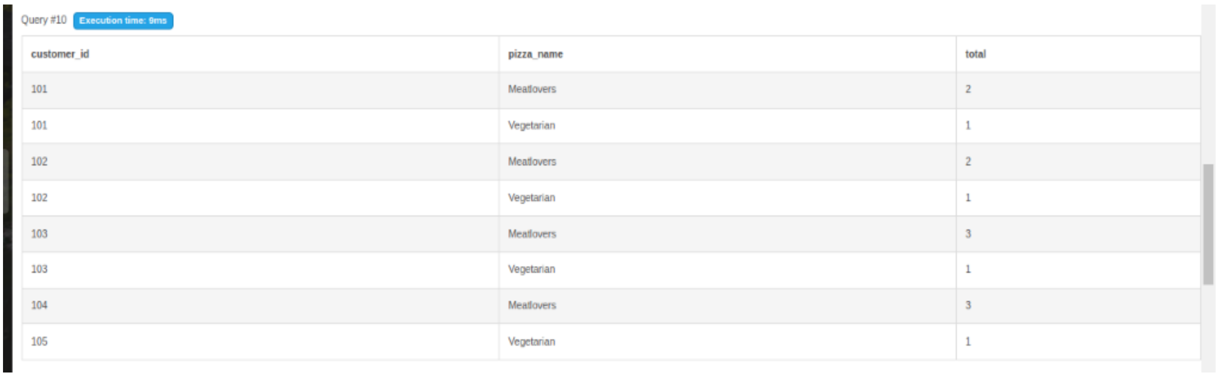


Query #9 Execution time: 1ms

pizza_name	total
Meatlovers	9
Vegetarian	3

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

```
select c.customer_id,
        p.pizza_name,
        count(c.pizza_id) As Total
        from customer_orders1 as c inner join pizza_runner.pizza_names p
        on c.pizza_id=p.pizza_id
        group by c.customer_id,p.pizza_id,p.pizza_name
        order by c.customer_id;
```



Query #10 Execution time: 9ms

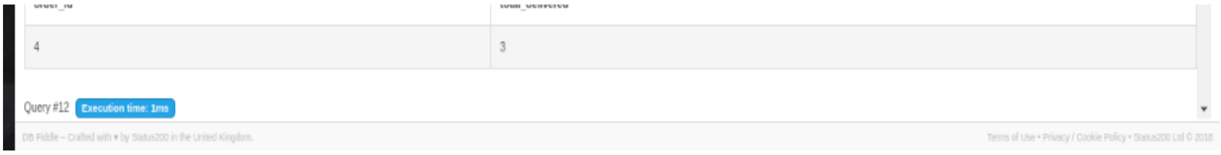
customer_id	pizza_name	total
101	Meatlovers	2
101	Vegetarian	1
102	Meatlovers	2
102	Vegetarian	1
103	Meatlovers	3
103	Vegetarian	1
104	Meatlovers	3
105	Vegetarian	1

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

```
select r.order_id,
        count(c.pizza_id) As Total_Delivered
        from runner_orders1 r inner join customer_orders1 c
        on c.order_id=r.order_id
        where r.distance is not null
        group by r.order_id
        order by Total_Delivered Desc LIMIT 1;
```

Query #11 Execution time: 0ms

order_id	total_delivered



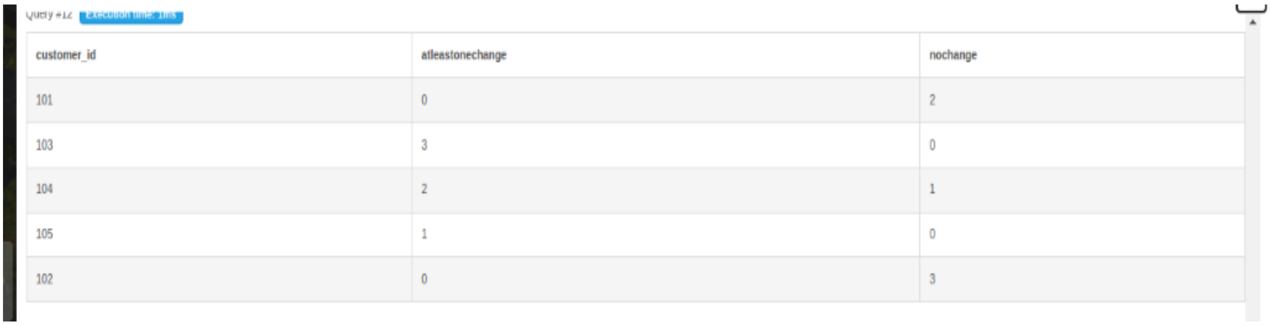
order_id	total_delivered
4	3

Query #12 Execution time: 1ms

DB Fiddle – Crafted with ♥ by Status200 in the United Kingdom.
Terms of Use • Privacy / Cookie Policy • Status200 Ltd © 2018

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

```
select customer_orders1.customer_id,
        sum(case
                when (exclusions is not null and exclusions != '') or (extras is not null and
extras                            != '') then 1
              else 0
                end )as AtleastOneChange,
          sum(case
                when (exclusions is null  or exclusions = '') and (extras is null or extras = '') then 1
              else 0
                 end ) as NoChange
                 from customer_orders1
                 inner join runner_orders1
                 on runner_orders1.order_id = customer_orders1.order_id
                 where runner_orders1.distance != 0
                 group by customer_orders1.customer_id;
```

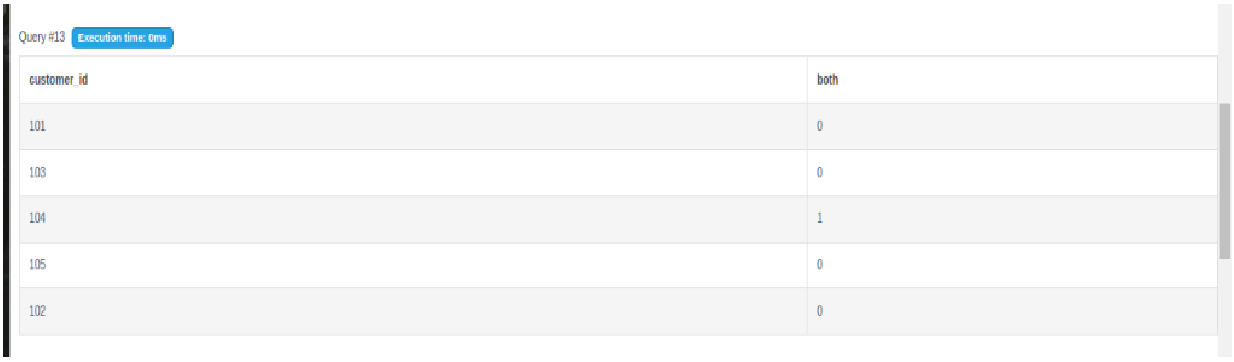


Query #12 Execution time: 1ms

customer_id	atleastonechange	nochange
101	0	2
103	3	0
104	2	1
105	1	0
102	0	3

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

```
select customer_orders1.customer_id,
        sum(case  when (exclusions is not null and exclusions != '') and (extras is not null and extras
!= '') then 1 else 0 end )as Both
        from customer_orders1 inner join runner_orders1
        on runner_orders1.order_id = customer_orders1.order_id
        where runner_orders1.distance != 0  group by customer_orders1.customer_id;
```

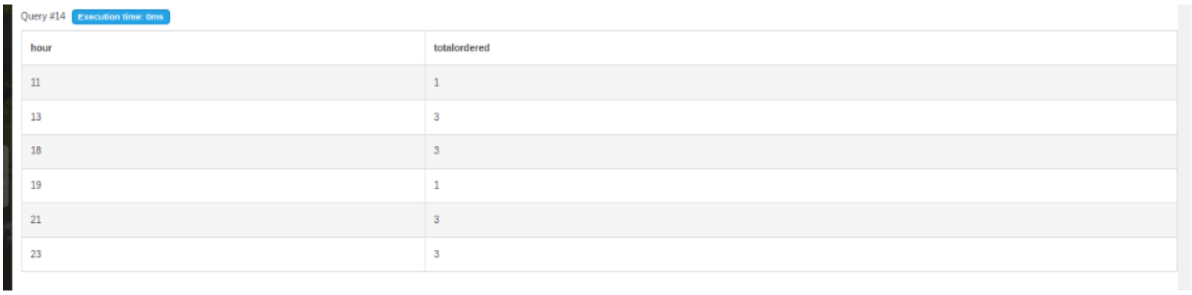


Query #13 Execution time: 0ms

customer_id	both
101	0
103	0
104	1
105	0
102	0

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

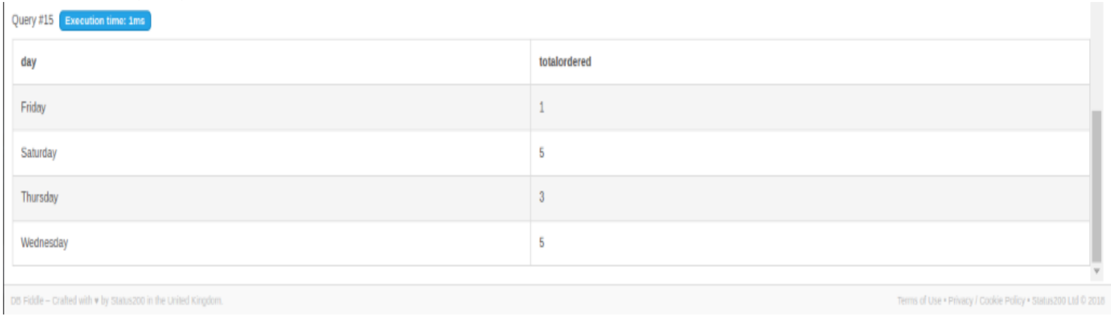
```
select extract(hour from order_time) as Hour,
        count(order_id) as TotalOrdered
        from customer_orders1
        group by Hour
        order by Hour;
```


Query #14 Execution time: 0ms

hour	totalordered
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 to_char(order_time, 'Day') as Day,
        count(order_id) as TotalOrdered
        from customer_orders1
        group by Day
        order by Day;
```


Query #15 Execution time: 1ms

day	totalordered
Friday	1
Saturday	5
Thursday	3
Wednesday	5

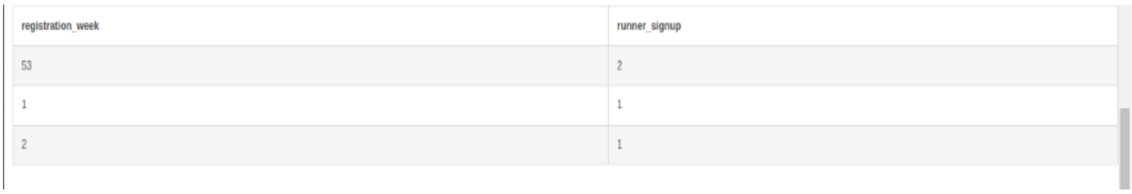
DB Fiddle – Crafted with ♥ by Status200 in the United Kingdom.

Terms of Use • Privacy / Cookie Policy • Status200 Ltd © 2018

Section – 2

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

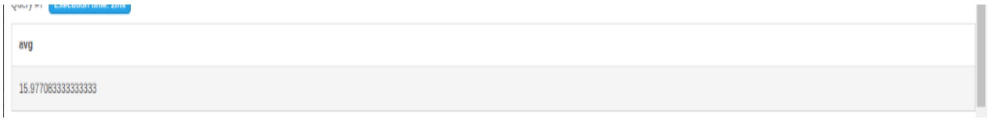
```
select EXTRACT(week FROM registration_date) AS registration_week,
        COUNT(runner_id) AS runner_signup
        from pizza_runner.runners
        GROUP BY registration_week;
```



registration_week	runner_signup
53	2
1	1
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?

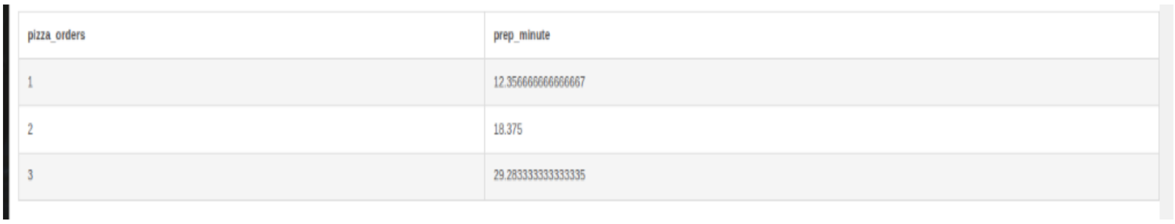
```
select avg(MInutes) from
        (select c.order_id,
                c.order_time,
                r.pickup_time,
                extract(epoch from r.pickup_time-c.order_time)/60 AS MInutes from
                customer_orders1 as c inner join runner_orders1 as r
                on c.order_id=r.order_id where r.distance is not null
                GROUP BY c.order_id,c.order_time,r.pickup_time
                order by order_id) As Minute;
```



avg
15.9770833333333

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

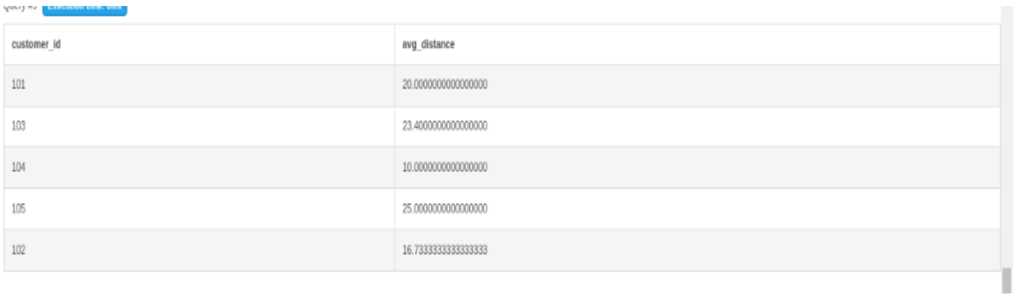
```
with CTE As(
        select count(c.order_id) As Pizza_orders,
                c.order_time,
                r.pickup_time,
                extract(epoch from r.pickup_time-c.order_time)/60 AS Prep_MInutes from
                customer_orders1 as c inner join runner_orders1 as r
                on c.order_id=r.order_id where r.distance is not null
                GROUP BY c.order_id,c.order_time,r.pickup_time)
        select Pizza_orders,avg(Prep_MInutes) As Prep_minute from CTE group by Pizza_orders order by
Pizza_orders;
```



pizza_orders	prep_minute
1	12.356666666666667
2	18.375
3	29.283333333333335

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

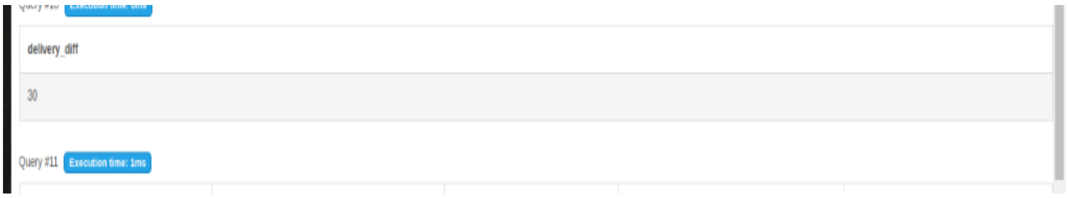
```
select c.customer_id,
        avg(r.distance) As Avg_distance from
        customer_orders1 c inner join runner_orders1 r
        on c.order_id=r.order_id
        where r.distance is not null group by c.customer_id;
```



customer_id	avg_distance
101	20.0000000000000000
103	23.4000000000000000
104	10.0000000000000000
105	25.0000000000000000
102	16.7333333333333333

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

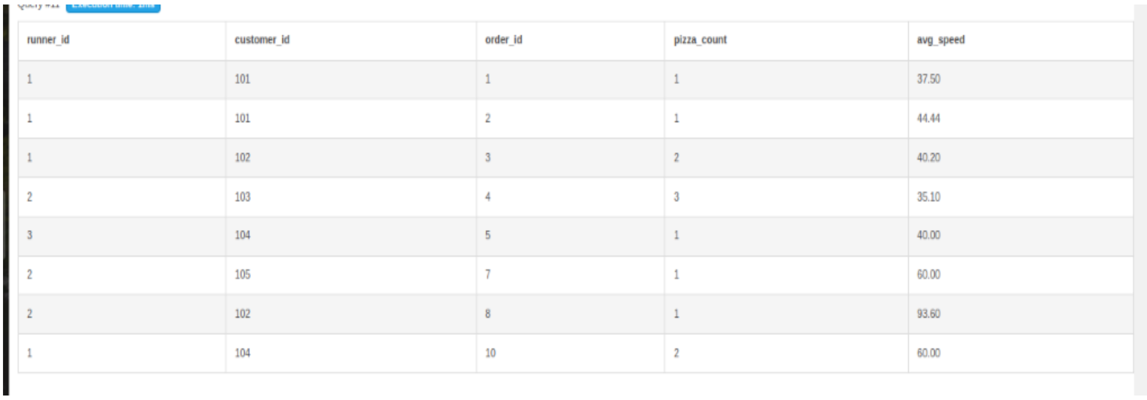
```
select
        max(duration::NUMERIC)-min(duration::NUMERIC) As Delivery_diff
        from runner_orders1;
```


Query #11
Execution time: 1ms

delivery_diff
30

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

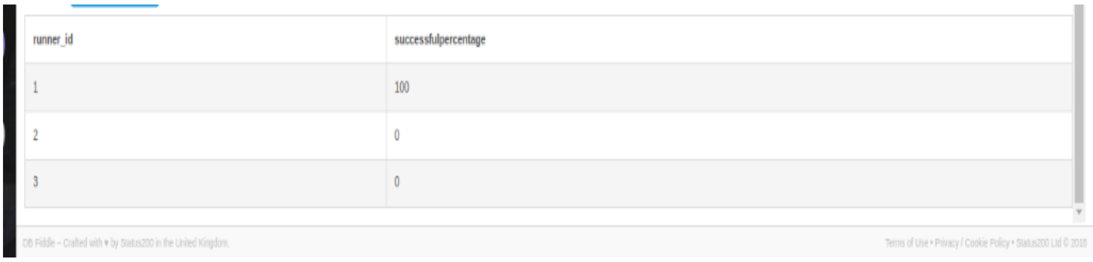
```
SELECT r.runner_id,
        c.customer_id,
        c.order_id,
        COUNT(c.order_id) AS pizza_count,
        ROUND((r.distance/r.duration * 60), 2) AS avg_speed
        FROM runner_orders1 AS r
        JOIN customer_orders1 AS c
        ON r.order_id = c.order_id
        WHERE distance != 0
        GROUP BY r.runner_id, c.customer_id, c.order_id, r.distance, r.duration
        ORDER BY c.order_id;
```



runner_id	customer_id	order_id	pizza_count	avg_speed
1	101	1	1	37.50
1	101	2	1	44.44
1	102	3	2	40.20
2	103	4	3	35.10
3	104	5	1	40.00
2	105	7	1	60.00
2	102	8	1	93.60
1	104	10	2	60.00

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

```
with cte as(
            select runner_id,
            sum(case
                when distance != 0 then 1
                else 0
                end) as percsucc, count(order_id) as TotalOrders
            from runner_orders1
            group by runner_id)
select runner_id,
        round((percsucc/TotalOrders)*100) as Successfulpercentage
        from cte
        order by runner_id;
```



runner_id	successfulpercentage
1	100
2	0
3	0

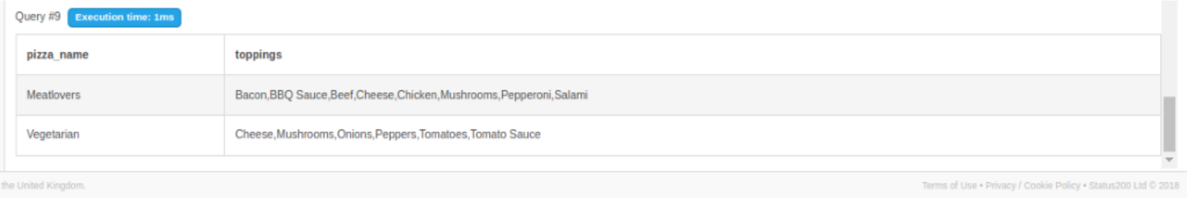
DB Fiddle – Crafted with ♥ by Status200 in the United Kingdom.

Terms of Use • Privacy / Cookie Policy • Status200 Ltd © 2018

Section – 3

1. What are the standard ingredients for each pizza?

```
create table Pizza_toppings(
         pizza_id int,toppings int);
        insert into Pizza_toppings values
        (1,1),
        (1,2),
        (1,3),
        (1,4),
        (1,5),
        (1,6),
        (1,8),
        (1,10),
        (2,4),
        (2,6),
        (2,7),
        (2,9),
        (2,11),
        (2,12);

        select * from pizza_toppings;

        select P.pizza_name,
                string_agg(pt.topping_name, ',')As Toppings
                from pizza_runner.pizza_names p inner join Pizza_toppings t
                on p.pizza_id=t.pizza_id inner join pizza_runner.pizza_toppings pt
                on pt.topping_id=t.toppings group by pizza_name;
```



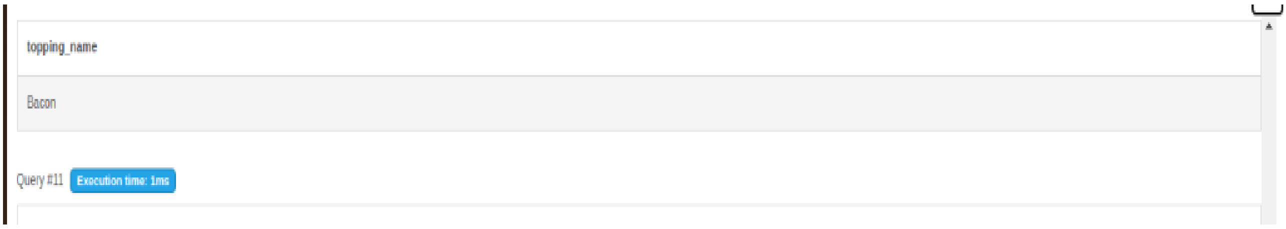
Query #9 Execution time: 1ms

pizza_name	toppings
Meatlovers	Bacon,BBQ Sauce,Beef,Cheese,Chicken,Mushrooms,Pepperoni,Salami
Vegetarian	Cheese,Mushrooms,Onions,Peppers,Tomatoes,Tomato Sauce

the United Kingdom. Terms of Use • Privacy / Cookie Policy • Status200 Ltd © 2018

2. What was the most commonly added extra?

```
select p.topping_name
        from pizza_runner.pizza_toppings p
        where topping_id=
        (select regexp_split_to_table(extras,',')::int as string
        from customer_orders1
        where  extras != '' and extras is not null
        group by  regexp_split_to_table(extras,',')::int
         order by count(*) DESC LIMIT 1);
```

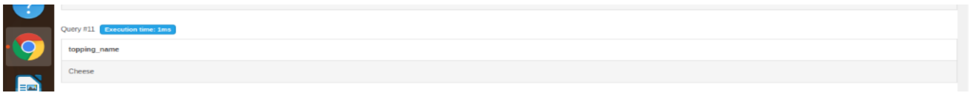


topping_name
Bacon

Query #11 Execution time: 1ms

3. What was the most common exclusion?

```
select p.topping_name
        from pizza_runner.pizza_toppings p
        where topping_id=
                (select regexp_split_to_table(exclusions,',')::int as string
                        from customer_orders1
                        where  exclusions != '' and exclusions is not null
                        group by  regexp_split_to_table(exclusions,',')::int
                        order by count(*) DESC LIMIT 1);
```



Query #11 Execution time: 1ms

topping_name
Cheese

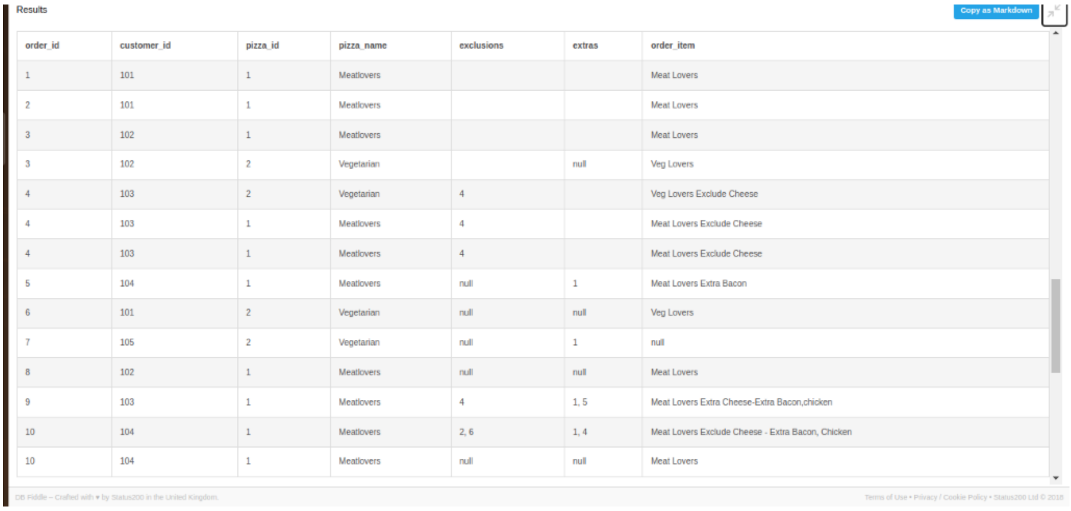
IMPORTANT TO BE NOTED........Special for string how to Aggregate a string then split by comma..

(e.g

```
select unnest(string_to_array(string,',')) from (select string_agg(extras,',') as string from
customer_orders1 where  extras != '' and extras is not null) as t; )
```

4. Generate an order item for each record in the customers_orders table in the format of one of the following:
   - Meat Lovers
   - Meat Lovers - Exclude Beef
   - Meat Lovers - Extra Bacon
   - Meat Lovers - Exclude Cheese, Bacon - Extra Mushroom, Peppers

```
select c.order_id,
        c.customer_id,
        c.pizza_id,
        p.pizza_name,
        c.exclusions,
        c.extras,
        case
                when c.pizza_id=1 and (exclusions = '' or exclusions is null) and (extras = '' or extras is
                null)  then 'Meat Lovers'
                when c.pizza_id=2 and (exclusions = '' or exclusions is null) and (extras = '' or extras is
                null)  then 'Veg Lovers'
                when c.pizza_id=1 and (exclusions Like '%3%' or exclusions='3') and (extras = '' or
                extras is null)  then 'Meat Lovers Exclude Beeef'
                when c.pizza_id=2 and (exclusions = '4' ) and (extras = '' or extras is null)  then 'Veg
                Lovers Exclude Cheese'
                when c.pizza_id=1 and (exclusions = '4' ) and (extras = '' or extras is null)  then 'Meat
                Lovers Exclude Cheese'
                when c.pizza_id=1 and (exclusions is null or exclusions='') and (extras = '1')  then 'Meat
                Lovers Extra Bacon'
                when c.pizza_id=1 and (exclusions='4') and (extras like '1, 5')  then 'Meat Lovers Extra
                Cheese-Extra Bacon,chicken'
                when c.pizza_id=1 and (exclusions like '1, 4' ) and (extras like '6, 9') then 'Meat Lovers
                - Exclude Cheese, Bacon - Extra Mushroom, Peppers'
                when c.pizza_id=1 and (exclusions like '2, 6') and (extras like '1, 4')  then 'Meat Lovers
                Exclude Cheese - Extra Bacon, Chicken'
        end as Order_Item
        from customer_orders1 c inner join pizza_runner.pizza_names p
        on c.pizza_id=p.pizza_id
        order by order_id;
```



Results

order_id	customer_id	pizza_id	pizza_name	exclusions	extras	order_item
1	101	1	Meatlovers			Meat Lovers
2	101	1	Meatlovers			Meat Lovers
3	102	1	Meatlovers			Meat Lovers
3	102	2	Vegetarian		null	Veg Lovers
4	103	2	Vegetarian	4		Veg Lovers Exclude Cheese
4	103	1	Meatlovers	4		Meat Lovers Exclude Cheese
4	103	1	Meatlovers	4		Meat Lovers Exclude Cheese
5	104	1	Meatlovers	null	1	Meat Lovers Extra Bacon
6	101	2	Vegetarian	null	null	Veg Lovers
7	105	2	Vegetarian	null	1	null
8	102	1	Meatlovers	null	null	Meat Lovers
9	103	1	Meatlovers	4	1, 5	Meat Lovers Extra Cheese-Extra Bacon,chicken
10	104	1	Meatlovers	2, 6	1, 4	Meat Lovers Exclude Cheese - Extra Bacon, Chicken
10	104	1	Meatlovers	null	null	Meat Lovers

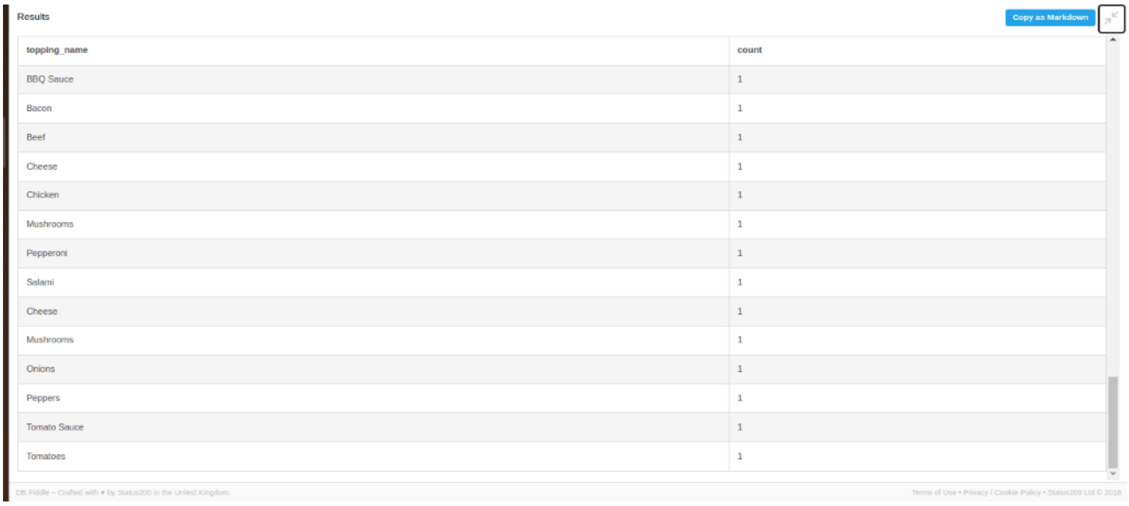
DB Fiddle – Crafted with ♥ by Status200 in the United Kingdom.

Terms of Use • Privacy / Cookie Policy • Status200 Ltd © 2018

5. What is the total quantity of each ingredient used in all delivered pizzas sorted by most frequent first?

```
select pz.topping_
        name,count(topping_id)
```

```
from Pizza_toppings p inner join pizza_runner.pizza_toppings pz
on p.toppings=pz.topping_id group by pizza_id,topping_name order by pizza_id;
```



Results

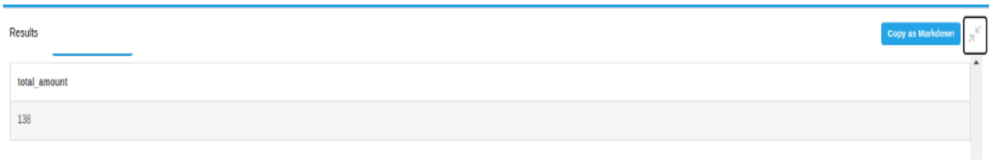
topping_name	count
BBQ Sauce	1
Bacon	1
Beef	1
Cheese	1
Chicken	1
Mushrooms	1
Pepperoni	1
Salami	1
Cheese	1
Mushrooms	1
Onions	1
Peppers	1
Tomato Sauce	1
Tomatoes	1

DB Fiddle – Crafted with ♥ by Status200 in the United Kingdom. Terms of Use • Privacy / Cookie Policy • Status200 Ltd © 2018

Section – 4

1. 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?

```
select sum(case when c.pizza_id=1 then 12 else 10 end) as Total_Amount
     from runner_orders1 as r inner join customer_orders1 c
     on c.order_id=r.order_id
     where r.distance is not null;
```

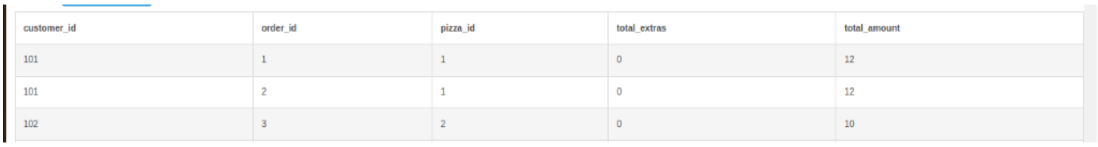


Results

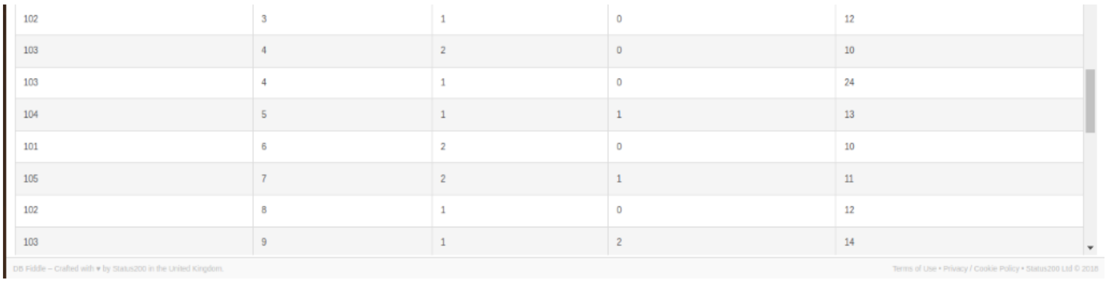
total_amount
138

2. What if there was an additional $1 charge for any pizza extras?
   a. Add cheese is $1 extra

```
with CTE AS(select *,
  CASE
    WHEN EXTRAS IS NOT NULL AND EXTRAS != '' AND lower(EXTRAS) != 'null'
      THEN (LENGTH(EXTRAS) - LENGTH(REPLACE(EXTRAS,',',''))) + 1)
    ELSE 0
  END total_extras from customer_orders1)
 SELECT customer_id,order_id,pizza_id,total_extras,
 SUM(CASE
    WHEN pizza_id=1 AND total_extras > 0 then 12+total_extras
       WHEN pizza_id=2 AND total_extras > 0  then 10+total_extras
       WHEN pizza_id=1 then 12 ELSE 10
  END)Total_Amount

FROM CTE group by customer_id,order_id,pizza_id,total_extras order by order_id;
```



customer_id	order_id	pizza_id	total_extras	total_amount
101	1	1	0	12
101	2	1	0	12
102	3	2	0	10

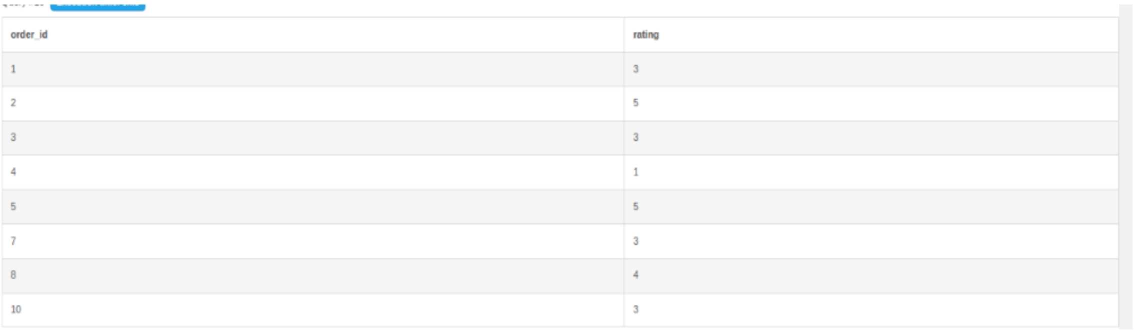


102	3	1	0	12
103	4	2	0	10
103	4	1	0	24
104	5	1	1	13
101	6	2	0	10
105	7	2	1	11
102	8	1	0	12
103	9	1	2	14

DB Fiddle – Crafted with ♥ by Status200 in the United Kingdom. Terms of Use • Privacy / Cookie Policy • Status200 Ltd © 2018

3. 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.

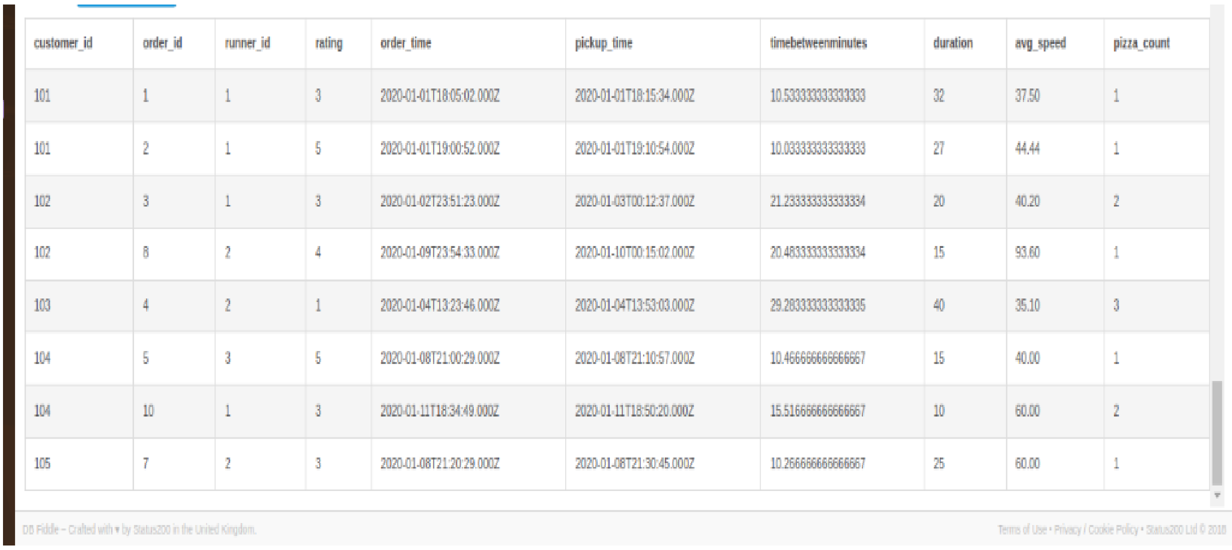
```
create table ratings (
        order_id integer,
        rating integer);
        insert into ratings
        (order_id, rating)
                values
                (1,3),
                (2,5),
                (3,3),
                (4,1),
                (5,5),
                (7,3),
                (8,4),
                (10,3);
        select * from ratings;
```



order_id	rating
1	3
2	5
3	3
4	1
5	5
7	3
8	4
10	3

4. Using your newly generated table - can you join all of the information together to form a table which has the following information for successful deliveries?
   - customer_id
   - order_id
   - runner_id
   - rating
   - order_time
   - pickup_time
   - Time between order and pickup
   - Delivery duration
   - Average speed
   - Total number of pizzas

```
select c.customer_id,
        c.order_id,
        r.runner_id,
        ra.rating,
        c.order_time,
        r.pickup_time,
        extract(epoch from r.pickup_time-c.order_time)/60 AS TimeBetweenMInutes, r.duration,
        ROUND((r.distance/r.duration * 60), 2) AS avg_speed,
        COUNT(c.order_id) AS pizza_count
        from customer_orders1 c inner join runner_orders1 r
        on c.order_id=r.order_id
        inner join ratings ra
        on ra.order_id=c.order_id
        group by
        c.customer_id,c.order_id,r.runner_id,ra.rating,c.order_time,r.pickup_time,TimeBetweenMI
        nutes,r.duration,r.distance;
```



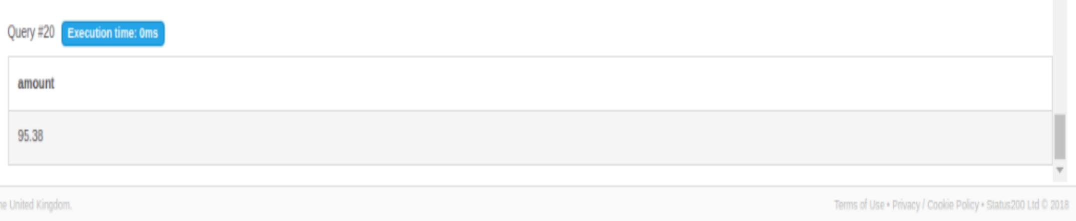
customer_id	order_id	runner_id	rating	order_time	pickup_time	timebetweenminutes	duration	avg_speed	pizza_count
101	1	1	3	2020-01-01T18:05:02.000Z	2020-01-01T18:15:34.000Z	10.533333333333333	32	37.50	1
101	2	1	5	2020-01-01T19:00:52.000Z	2020-01-01T19:10:54.000Z	10.033333333333333	27	44.44	1
102	3	1	3	2020-01-02T23:51:23.000Z	2020-01-03T00:12:37.000Z	21.233333333333334	20	40.20	2
102	8	2	4	2020-01-09T23:54:33.000Z	2020-01-10T00:15:02.000Z	20.483333333333334	15	93.60	1
103	4	2	1	2020-01-04T13:23:46.000Z	2020-01-04T13:53:03.000Z	29.283333333333335	40	35.10	3
104	5	3	5	2020-01-08T21:00:29.000Z	2020-01-08T21:10:57.000Z	10.466666666666667	15	40.00	1
104	10	1	3	2020-01-11T18:34:49.000Z	2020-01-11T18:50:20.000Z	15.516666666666667	10	60.00	2
105	7	2	3	2020-01-08T21:20:29.000Z	2020-01-08T21:30:45.000Z	10.266666666666667	25	60.00	1

DB Fiddle – Crafted with ♥ by Status200 in the United Kingdom.

Terms of Use • Privacy / Cookie Policy • Status200 Ltd © 2018

5. 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?

```
select (Total_Amount-distance) As Amount
        from
                (select sum(case when c.pizza_id=1 then 12 else 10 end) as Total_Amount,
                (sum(r.distance))*0.3 as distance from runner_orders1 r inner join customer_orders1
                c on c.order_id=r.order_id) As Delivery;
```



Query #20 Execution time: 0ms

amount
95.38

e United Kingdom.

Terms of Use • Privacy / Cookie Policy • Status200 Ltd © 2018