SQL Case Study 2: Burger Bash

Name: PRIYESHWAR M D

Mail: priyesh2664@gmail.com

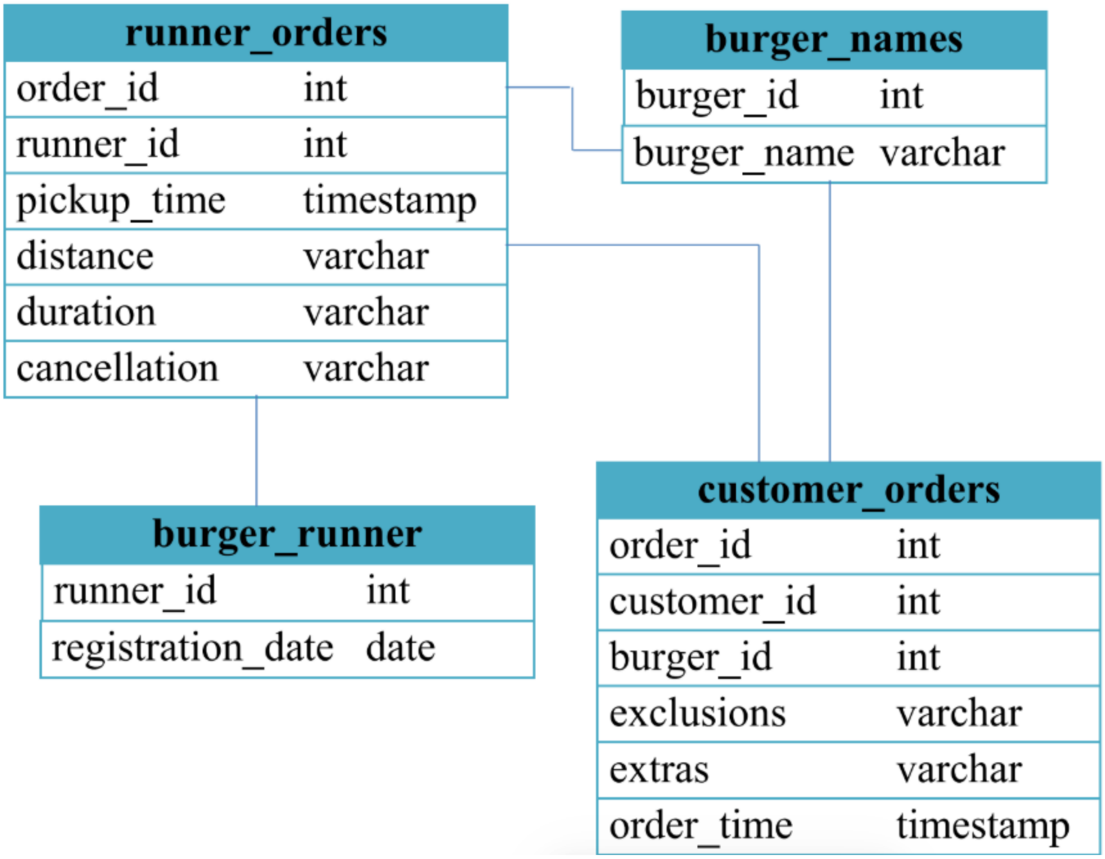
INTRODUCTION:

I have a started a new business of selling burger because I read on my Instagram feed that „Burger Is the Future!

But I knew that burger alone was not going to help me get seed funding to expand my new Burger Empire - so I had one more genius idea to combine with it - I was going to Uberize it - and so Burger Runner was launched!

I started by recruiting "runners" to deliver fresh burger from Burger Runner Headquarters and also maxed out my credit card to pay freelance developers to build a mobile app to accept orders from customers.

SCHEMA USED



runner_orders	
order_id	int
runner_id	int
pickup_time	timestamp
distance	varchar
duration	varchar
cancellation	varchar

burger_names	
burger_id	int
burger_name	varchar

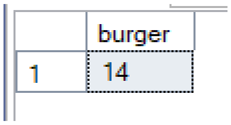
burger_runner	
runner_id	int
registration_date	date

customer_orders	
order_id	int
customer_id	int
burger_id	int
exclusions	varchar
extras	varchar
order_time	timestamp

CASE STUDY QUESTIONS

1. How many burgers were ordered?

select count(burger_id) burger from customer_orders;

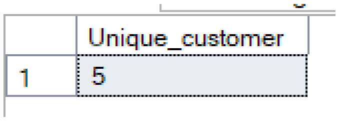


	burger
1	14

---

2. How many unique customer orders were made?

select count(distinct customer_id) Unique_customer from customer_orders;



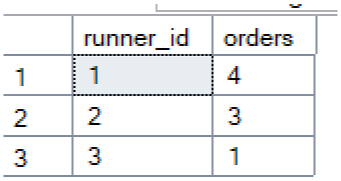
	Unique_customer
1	5

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

select runner_id,count(*) as orders from runner_orders

where cancellation is NULL

group by runner_id;



	runner_id	orders
1	1	4
2	2	3
3	3	1

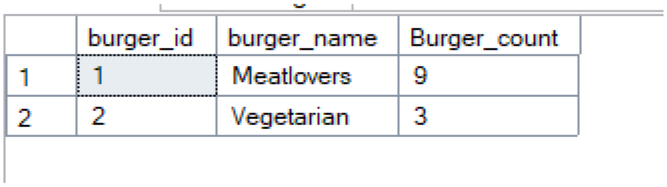
4. How many of each type of burger was delivered?

select c.burger_id,burger_name,count(*) Burger_count

from customer_orders c join burger_names b on c.burger_id =b.burger_id join runner_orders r on c.order_id=r.order_id

where r.cancellation is NULL

group by c.burger_id,b.burger_name;

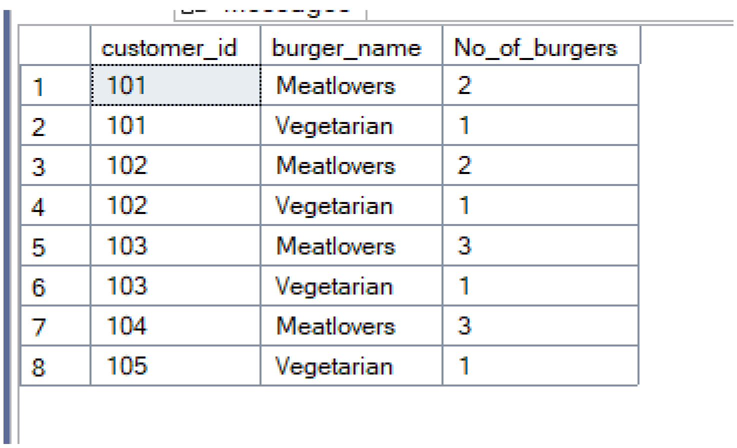


	burger_id	burger_name	Burger_count
1	1	Meatlovers	9
2	2	Vegetarian	3

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

select customer_id,burger_name, count(c.burger_id) as No_of_burgers from customer_orders c join burger_names b on c.burger_id = b.burger_id

group by c.burger_id,customer_id,burger_name;

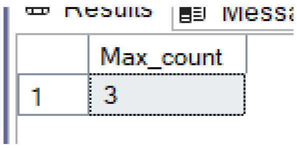


	customer_id	burger_name	No_of_burgers
1	101	Meatlovers	2
2	101	Vegetarian	1
3	102	Meatlovers	2
4	102	Vegetarian	1
5	103	Meatlovers	3
6	103	Vegetarian	1
7	104	Meatlovers	3
8	105	Vegetarian	1

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

```
select max(burger_count) Max_count from (select order_id,count(*) as burger_count from customer_orders

group by order_id)a
```



	Max_count
1	3

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

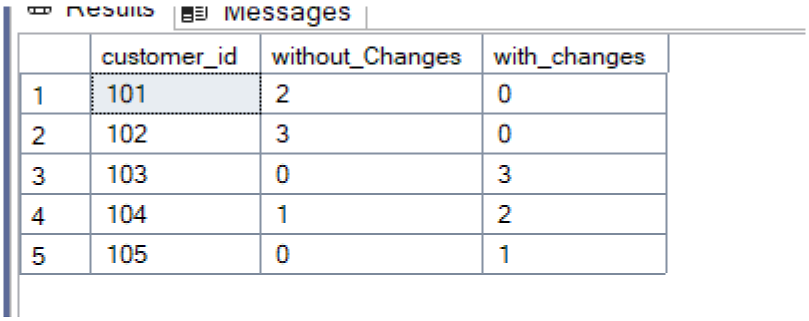
```
select customer_id,sum(case when (exclusions is NULL and extras is null )then 1 else 0 end) as without_Changes,

sum(case when (exclusions is not NULL or extras is not null )then 1 else 0 end) as with_changes

from customer_orders c join runner_orders r on c.order_id=r.order_id

where cancellation is null

group by customer_id;
```



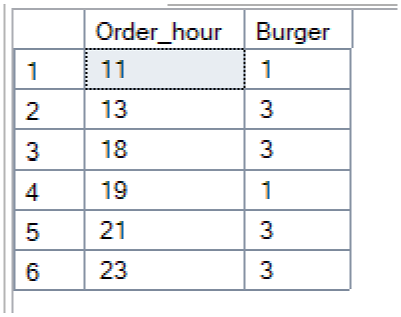
	customer_id	without_Changes	with_changes
1	101	2	0
2	102	3	0
3	103	0	3
4	104	1	2
5	105	0	1

8. What was the total volume of burgers ordered for each hour of the day?

```
select DATEPART(HOUR,order_time) as Order_hour, count(*) as Burger from customer_orders

group by DATEPART(HOUR,order_time)

order by 1;
```



	Order_hour	Burger
1	11	1
2	13	3
3	18	3
4	19	1
5	21	3
6	23	3

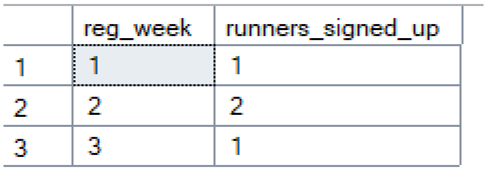
9. How many runners signed up for each 1 week period?

```
select  datepart(week, registration_date) as reg_week,count(*) as runners_signed_up

from burger_runner

group by datepart(year, registration_date), datepart(week, registration_date)

order by  reg_week;
```



	reg_week	runners_signed_up
1	1	1
2	2	2
3	3	1

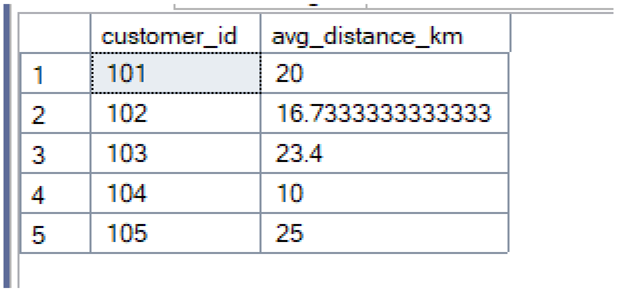
10. What was the average distance travelled for each customer?

```
select c.customer_id,avg(try_cast(replace(replace(r.distance, 'km', ''), ' ', '') as float)) as avg_distance_km

from customer_orders c join runner_orders r on c.order_id = r.order_id

where r.cancellation is null

group by c.customer_id;
```



	customer_id	avg_distance_km
1	101	20
2	102	16.7333333333333
3	103	23.4
4	104	10
5	105	25