

SQL CASE STUDY : BURGER BASH

BY - NITESH KUMAR

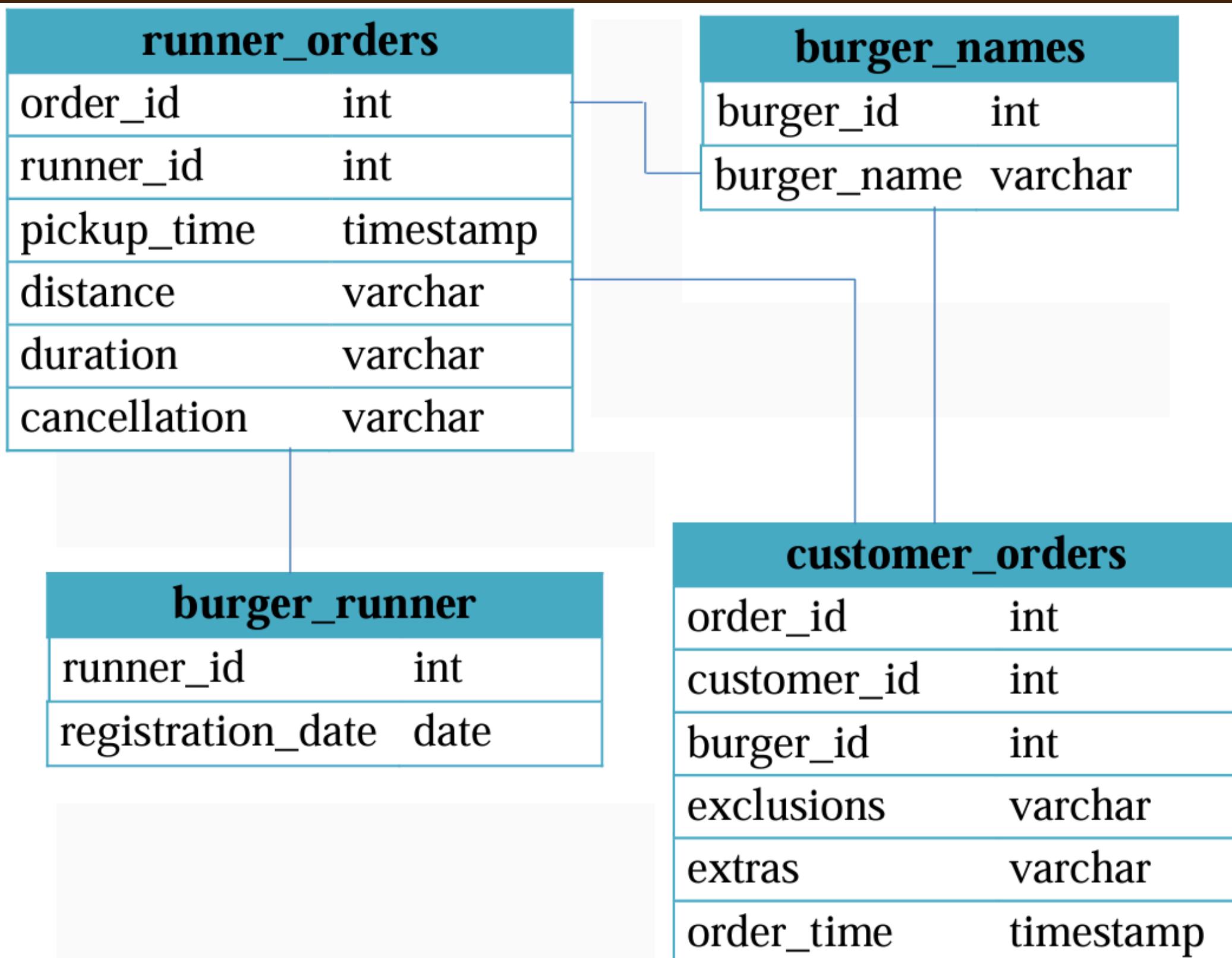


INTRODUCTION

In this case study we will be looking for some insights from a Burger shop dataset and ultimately helping the shop to improve it's operation as well as it's productivity!



SCHEMA USED



CASE STUDY QUESTIONS

1. How many burgers were ordered?
2. How many unique customer orders were made?
3. How many successful orders were delivered by each runner?
4. How many of each type of burger was delivered?
5. How many Vegetarian and Meatlovers were ordered by each customer?
6. What was the maximum number of burgers delivered in a single order?
7. For each customer, how many delivered burgers had at least 1 change and how many had no changes?
8. What was the total volume of burgers ordered for each hour of the day?
9. How many runners signed up for each 1 week period?
- 10.What was the average distance travelled for each customer?

1. HOW MANY BURGERS WERE ORDERED?

```
SELECT  
    COUNT(*) AS no_of_orders  
FROM  
    runner_orders;
```

no_of_orders

10

2. HOW MANY UNIQUE CUSTOMER ORDERS WERE MADE?

```
SELECT  
    COUNT(DISTINCT order_id) AS no_of_unique_orders  
FROM  
    customer_orders;
```

no_of_unique_orders
10

3. HOW MANY SUCCESSFUL ORDERS WERE DELIVERED BY EACH RUNNER?

```
SELECT
    runner_id, COUNT(DISTINCT order_id) successful_orders
FROM
    runner_orders
WHERE
    cancellation IS NULL
GROUP BY 1
ORDER BY 2 DESC;
```

runner_id	successful_orders
1	4
2	3
3	1

4. HOW MANY OF EACH TYPE OF BURGER WAS DELIVERED?

```
SELECT  
    b.burger_name, COUNT(c.burger_id) burger_delivered_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 1;
```

burger_name	burger_delivered_count
Meatlovers	9
Vegetarian	3

5. HOW MANY VEGETARIAN AND MEATLOVERS WERE ORDERED BY EACH CUSTOMER?

SELECT

```
c.customer_id,  
p.burger_name,  
COUNT(p.burger_name) AS order_count  
  
FROM  
customer_orders AS c  
JOIN  
burger_names AS p ON c.burger_id = p.burger_id  
GROUP BY c.customer_id , p.burger_name  
ORDER BY c.customer_id;
```

customer_id	burger_name	order_count
101	Meatlovers	2
101	Vegetarian	1
102	Meatlovers	2
102	Vegetarian	1
103	Meatlovers	3

6. WHAT WAS THE MAXIMUM NUMBER OF BURGERS DELIVERED IN A SINGLE ORDER?

SELECT

```
MAX(burger_per_order) AS burger_count  
FROM  
(SELECT  
    c.order_id, COUNT(c.burger_id) AS burger_per_order  
FROM  
    customer_orders AS c  
JOIN runner_orders AS r ON c.order_id = r.order_id  
WHERE  
    r.distance <> 0  
GROUP BY c.order_id) burger_count
```

burger_count

3

7. FOR EACH CUSTOMER, HOW MANY DELIVERED BURGERS HAD AT LEAST 1 CHANGE AND HOW MANY HAD NO CHANGES?

```
SELECT c.customer_id,
       SUM(CASE
             WHEN c.exclusions <> '' OR c.extras <> '' THEN 1
             ELSE 0
           END) AS at_least_1_change,
       SUM(CASE
             WHEN c.exclusions = '' AND c.extras = '' 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 <> 0
 GROUP BY 1
 ORDER BY 1;
```

customer_id	at_least_1_change	no_change
101	0	0
102	0	0
103	3	0
104	2	0
105	1	0

8. WHAT WAS THE TOTAL VOLUME OF BURGERS ORDERED FOR EACH HOUR OF THE DAY?

SELECT

```
EXTRACT(HOUR FROM order_time) AS hour_of_day,  
COUNT(order_id) AS burger_count
```

FROM

```
customer_orders
```

GROUP BY 1;

hour_of_day	burger_count
18	3
19	1
23	3
13	3
21	3

9. HOW MANY RUNNERS SIGNED UP FOR EACH 1 WEEK PERIOD?

```
SELECT  
    EXTRACT(WEEK FROM registration_date) AS registration_week,  
    COUNT(runner_id) AS runner_signup  
FROM  
    burger_runner  
GROUP BY 1;
```

registration_week	runner_signup
0	1
1	2
2	1

10.WHAT WAS THE AVERAGE DISTANCE TRAVELED FOR EACH CUSTOMER?

SELECT

```
c.customer_id, round(AVG(r.distance),2) AS avg_distance
```

FROM

```
customer_orders AS c
```

JOIN

```
runner_orders AS r ON c.order_id = r.order_id
```

WHERE

```
r.duration <> 0
```

GROUP BY 1;

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

THANK YOU

