

Sql Case Study 3 - Burger Bash

```
SQLQuery1.s... \masih (74))* ×
1  create database Burger;
2  use Burger;
3
4  CREATE TABLE burger_names(
5      burger_id INTEGER NOT NULL PRIMARY KEY
6      ,burger_name VARCHAR(10) NOT NULL
7  );
8  INSERT INTO burger_names(burger_id,burger_name) VALUES (1,'Meatlovers');
9  INSERT INTO burger_names(burger_id,burger_name) VALUES (2,'Vegetarian');
10 select * from burger_names;
11
```

OUTPUT:

Results			Messages	
	burger_id	burger_name		
1	1	Meatlovers		
2	2	Vegetarian		

```
SQLQuery1.s... \masih (74))* ×
12
13 CREATE TABLE runner_orders (
14     order_id INT PRIMARY KEY,
15     runner_id INT NOT NULL,
16     pickup_time DATETIME,
17     distance VARCHAR(20),
18     duration VARCHAR(20),
19     cancellation VARCHAR(50)
20 );
21 INSERT INTO runner_orders VALUES (1,1,'2021-01-01 18:15:34','20km','32 minutes',NULL);
22 INSERT INTO runner_orders VALUES (2,1,'2021-01-01 19:10:54','20km','27 minutes',NULL);
23 INSERT INTO runner_orders VALUES (3,1,'2021-01-03 00:12:37','13.4km','20 mins',NULL);
24 INSERT INTO runner_orders VALUES (4,2,'2021-01-04 13:53:03','23.4','40',NULL);
25 INSERT INTO runner_orders VALUES (5,3,'2021-01-08 21:10:57','10','15',NULL);
26 INSERT INTO runner_orders VALUES (6,3,NULL,NULL,NULL,'Restaurant Cancellation');
27 INSERT INTO runner_orders VALUES (7,2,'2021-01-08 21:30:45','25km','25mins',NULL);
28 INSERT INTO runner_orders VALUES (8,2,'2021-01-10 00:15:02','23.4 km','15 minute',NULL);
29 INSERT INTO runner_orders VALUES (9,2,NULL,NULL,NULL,'Customer Cancellation');
30 INSERT INTO runner_orders VALUES (10,1,'2021-01-11 18:50:20','10km','10minutes',NULL);
31 select * from runner_orders;
```

OUTPUT :

Results		Messages				
	order_id	runner_id	pickup_time	distance	duration	cancellation
1	1	1	2021-01-01 18:15:34.000	20km	32 minutes	NULL
2	2	1	2021-01-01 19:10:54.000	20km	27 minutes	NULL
3	3	1	2021-01-03 00:12:37.000	13.4km	20 mins	NULL
4	4	2	2021-01-04 13:53:03.000	23.4	40	NULL
5	5	3	2021-01-08 21:10:57.000	10	15	NULL
6	6	3	NULL	NULL	NULL	Restaurant Cancellation
7	7	2	2021-01-08 21:30:45.000	25km	25mins	NULL
8	8	2	2021-01-10 00:15:02.000	23.4 km	15 minute	NULL
9	9	2	NULL	NULL	NULL	Customer Cancellation
10	10	1	2021-01-11 18:50:20.000	10km	10minutes	NULL

```

SQLQuery1.s...\masih (74))* ✕
32
33 ✓ CREATE TABLE burger_runner(
34     runner_id  INTEGER NOT NULL PRIMARY KEY
35     ,registration_date date NOT NULL
36 );
37 INSERT INTO burger_runner VALUES (1, '2021-01-01');
38 INSERT INTO burger_runner VALUES (2, '2021-01-03');
39 INSERT INTO burger_runner VALUES (3, '2021-01-08');
40 INSERT INTO burger_runner VALUES (4, '2021-01-15');
41 select * from burger_runner;

```

OUTPUT:

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

```

SQLQuery1.s...\masih (74))* ✕
44 ✓ CREATE TABLE customer_orders (
45     order_id  INT NOT NULL,
46     customer_id INT NOT NULL,
47     burger_id  INT NOT NULL,
48     exclusions VARCHAR(50),
49     extras     VARCHAR(50),
50     order_time DATETIME NOT NULL
51 );
52 INSERT INTO customer_orders VALUES (1,101,1,NULL,NULL,'2021-01-01 18:05:02');
53 INSERT INTO customer_orders VALUES (2,101,1,NULL,NULL,'2021-01-01 19:00:52');
54 INSERT INTO customer_orders VALUES (3,102,1,NULL,NULL,'2021-01-02 23:51:23');
55 INSERT INTO customer_orders VALUES (3,102,2,NULL,NULL,'2021-01-02 23:51:23');
56 INSERT INTO customer_orders VALUES (4,103,1,'4',NULL,'2021-01-04 13:23:46');
57 INSERT INTO customer_orders VALUES (4,103,1,'4',NULL,'2021-01-04 13:23:46');
58 INSERT INTO customer_orders VALUES (4,103,2,'4',NULL,'2021-01-04 13:23:46');
59 INSERT INTO customer_orders VALUES (5,104,1,NULL,'1','2021-01-08 21:00:29');
60 INSERT INTO customer_orders VALUES (6,101,2,NULL,NULL,'2021-01-08 21:03:13');
61 INSERT INTO customer_orders VALUES (7,105,2,NULL,'1','2021-01-08 21:20:29');
62 INSERT INTO customer_orders VALUES (8,102,1,NULL,NULL,'2021-01-09 23:54:33');
63 INSERT INTO customer_orders VALUES (9,103,1,'4','1, 5','2021-01-10 11:22:59');
64 INSERT INTO customer_orders VALUES (10,104,1,NULL,NULL,'2021-01-11 18:34:49');
65 INSERT INTO customer_orders VALUES (10,104,1,'2, 6','1, 4','2021-01-11 18:34:49');
66 select * from customer_orders ;

```

OUTPUT:

Results

Messages

	order_id	customer_id	burger_id	exclusions	extras	order_time
1	1	101	1	NULL	NULL	2021-01-01 18:05:02.000
2	2	101	1	NULL	NULL	2021-01-01 19:00:52.000
3	3	102	1	NULL	NULL	2021-01-02 23:51:23.000
4	3	102	2	NULL	NULL	2021-01-02 23:51:23.000
5	4	103	1	4	NULL	2021-01-04 13:23:46.000
6	4	103	1	4	NULL	2021-01-04 13:23:46.000
7	4	103	2	4	NULL	2021-01-04 13:23:46.000
8	5	104	1	NULL	1	2021-01-08 21:00:29.000
9	6	101	2	NULL	NULL	2021-01-08 21:03:13.000
10	7	105	2	NULL	1	2021-01-08 21:20:29.000
11	8	102	1	NULL	NULL	2021-01-09 23:54:33.000
12	9	103	1	4	1, 5	2021-01-10 11:22:59.000
13	10	104	1	NULL	NULL	2021-01-11 18:34:49.000
14	10	104	1	2, 6	1, 4	2021-01-11 18:34:49.000

1. How many burgers were ordered?

```
SELECT COUNT(*) AS total_burgers_ordered  
FROM customer_orders;
```

OUTPUT :

Results		Messages	
	total_burgers_ordered		
1	14		

2. How many unique customer orders were made?

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

OUTPUT :

Results		Messages	
	unique_orders		
1	10		

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

```
SELECT runner_id, COUNT(*) AS successful_deliveries  
FROM runner_orders  
WHERE cancellation IS NULL AND pickup_time IS NOT NULL  
GROUP BY runner_id;
```

OUTPUT :

	runner_id	successful_deliveries
1	1	4
2	2	3
3	3	1

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

```
SELECT bn.burger_name, COUNT(*) AS total_delivered  
FROM customer_orders co  
JOIN runner_orders ro ON co.order_id = ro.order_id  
JOIN burger_names bn ON co.burger_id = bn.burger_id  
WHERE ro.cancellation IS NULL AND ro.pickup_time IS NOT NULL  
GROUP BY bn.burger_name;
```

OUTPUT :

	burger_name	total_delivered
1	Meatlovers	9
2	Vegetarian	3

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

```
SELECT customer_id,
SUM(CASE WHEN burger_id = 1 THEN 1 ELSE 0 END) AS meatlovers,
SUM(CASE WHEN burger_id = 2 THEN 1 ELSE 0 END) AS vegetarian
FROM customer_orders
GROUP BY customer_id;
```

OUTPUT :

	customer_id	meatlovers	vegetarian
1	101	2	1
2	102	2	1
3	103	3	1
4	104	3	0
5	105	0	1

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

```
SELECT TOP 1 co.order_id, COUNT(*) AS burger_count
FROM customer_orders co
JOIN runner_orders ro ON co.order_id = ro.order_id
WHERE ro.cancellation IS NULL
AND ro.pickup_time IS NOT NULL
GROUP BY co.order_id
ORDER BY burger_count DESC;
```

OUTPUT :

	order_id	burger_count
1	4	3

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

```
SELECT co.customer_id,
SUM(CASE WHEN exclusions IS NOT NULL OR extras IS NOT NULL THEN 1 ELSE 0 END) AS with_changes,
SUM(CASE WHEN exclusions IS NULL AND extras IS NULL THEN 1 ELSE 0 END) AS without_changes
FROM customer_orders co
JOIN runner_orders ro ON co.order_id = ro.order_id
WHERE ro.cancellation IS NULL AND ro.pickup_time IS NOT NULL
GROUP BY co.customer_id;
```

OUTPUT :

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

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 burgers_ordered
FROM customer_orders
GROUP BY DATEPART(HOUR, order_time)
ORDER BY order_hour;
```

OUTPUT :

	order_hour	burgers_ordered
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 week_number,
COUNT(*) AS runners_signed_up
FROM burger_runner
GROUP BY DATEPART(WEEK, registration_date)
ORDER BY week_number;
```

OUTPUT :

	week_number	runners_signed_up
1	1	1
2	2	2
3	3	1

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

```
SELECT co.customer_id,
AVG(CAST(REPLACE(REPLACE(ro.distance, 'km', ''), ' ', '' ) AS FLOAT)) AS avg_distance_km
FROM customer_orders co
JOIN runner_orders ro ON co.order_id = ro.order_id
WHERE ro.cancellation IS NULL AND ro.distance IS NOT NULL
GROUP BY co.customer_id;
```

OUTPUT :

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