Case Study – 1
Burger Bash

Name: Aathirainathan P

Date: 07-11-2024

I. Creating the Database & Tables:

```
select * from burger_names;
select * from runner_orders;
select * from burger_runner;
select * from customer_orders;
```

97 %

Results | Messages

	burger_id	burger_name
1	1	Meatlovers
2	2	Vegetarian

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

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

	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

II.Questions:

1. How many burgers were ordered?

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

97 %

Results | Messages

	total_burgers_ordered
1	14

2. How many unique customer orders were made?

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

97 %

Results | Messages

	unique_customer_orders
1	10

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

```
SELECT runner_id, COUNT(order_id) AS successful_orders
FROM runner_orders
WHERE cancellation IS NULL
GROUP BY runner_id;
```

97 %

Results | Messages

	runner_id	successful_orders
1	1	4
2	2	3
3	3	1

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

```
SELECT co.customer_id, bn.burger_name, COUNT(co.order_id) AS burgers_ordered
FROM customer_orders AS co
JOIN burger_names AS bn ON co.burger_id = bn.burger_id
GROUP BY co.customer_id, bn.burger_name;
```

97 %

Results | Messages

	customer_id	burger_name	burgers_ordered
1	101	Meatlovers	2
2	102	Meatlovers	2
3	103	Meatlovers	3
4	104	Meatlovers	3
5	101	Vegetarian	1
6	102	Vegetarian	1
7	103	Vegetarian	1
8	105	Vegetarian	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 burgers_ordered
FROM customer_orders
GROUP BY DATEPART(HOUR, order_time)
ORDER BY order_hour;
```

97 %

Results | Messages

	order_hour	burgers_ordered
1	11	1
2	13	3
3	18	3
4	19	1
5	21	3
6	23	3

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

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

97 %

Results | Messages

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