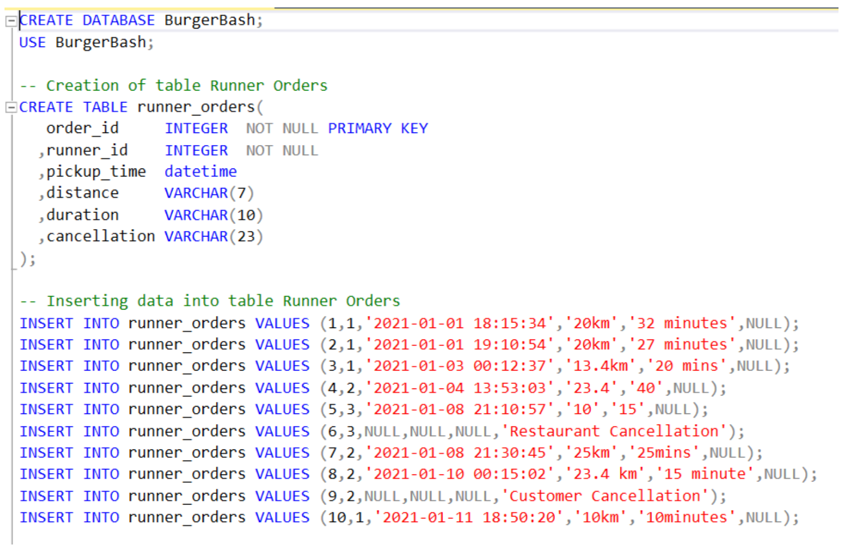
CASE STUDY – 2 (SQL)

TRAINEE NAME: Swathi Baskaran

BURGER BASH

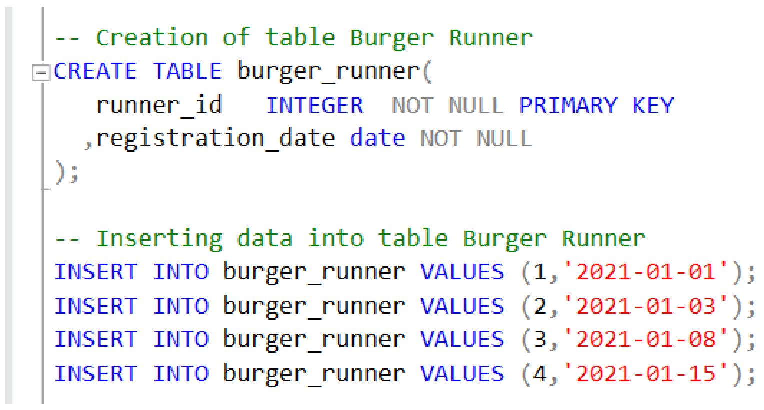
Creation of Tables and Insertion of Data

1. Runner Orders Table



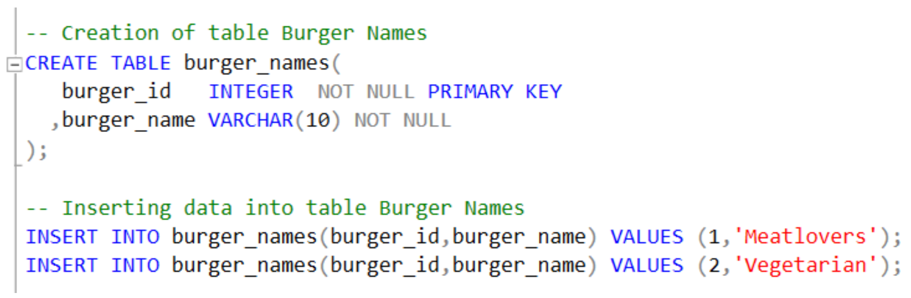
```sql
CREATE DATABASE BurgerBash;
USE BurgerBash;

-- Creation of table Runner Orders
CREATE TABLE runner_orders(
   order_id     INTEGER  NOT NULL PRIMARY KEY
  ,runner_id    INTEGER  NOT NULL
  ,pickup_time  datetime
  ,distance     VARCHAR(7)
  ,duration     VARCHAR(10)
  ,cancellation VARCHAR(23)
);

-- Inserting data into table Runner Orders
INSERT INTO runner_orders VALUES (1,1,'2021-01-01 18:15:34','20km','32 minutes',NULL);
INSERT INTO runner_orders VALUES (2,1,'2021-01-01 19:10:54','20km','27 minutes',NULL);
INSERT INTO runner_orders VALUES (3,1,'2021-01-03 00:12:37','13.4km','20 mins',NULL);
INSERT INTO runner_orders VALUES (4,2,'2021-01-04 13:53:03','23.4','40',NULL);
INSERT INTO runner_orders VALUES (5,3,'2021-01-08 21:10:57','10','15',NULL);
INSERT INTO runner_orders VALUES (6,3,NULL,NULL,NULL,'Restaurant Cancellation');
INSERT INTO runner_orders VALUES (7,2,'2021-01-08 21:30:45','25km','25mins',NULL);
INSERT INTO runner_orders VALUES (8,2,'2021-01-10 00:15:02','23.4 km','15 minute',NULL);
INSERT INTO runner_orders VALUES (9,2,NULL,NULL,NULL,'Customer Cancellation');
INSERT INTO runner_orders VALUES (10,1,'2021-01-11 18:50:20','10km','10minutes',NULL);
```

2. Burger Runner Table



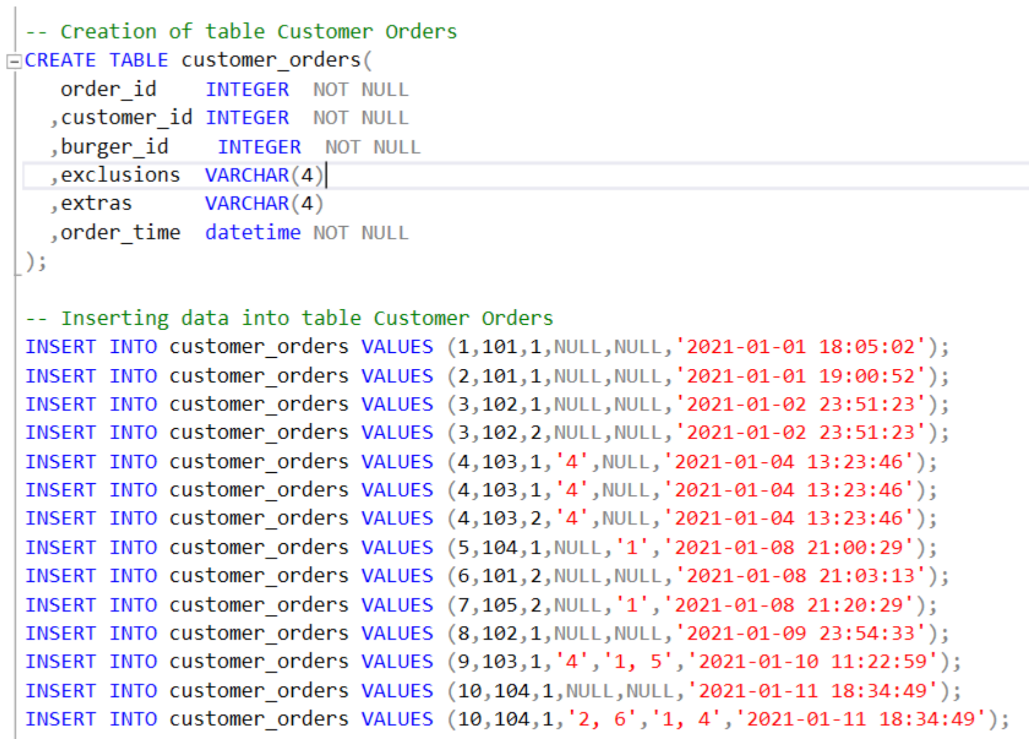
```sql
-- Creation of table Burger Runner
CREATE TABLE burger_runner(
   runner_id   INTEGER  NOT NULL PRIMARY KEY
  ,registration_date date NOT NULL
);

-- Inserting data into table Burger Runner
INSERT INTO burger_runner VALUES (1,'2021-01-01');
INSERT INTO burger_runner VALUES (2,'2021-01-03');
INSERT INTO burger_runner VALUES (3,'2021-01-08');
INSERT INTO burger_runner VALUES (4,'2021-01-15');
```

3. Burger Names Table



```sql
-- Creation of table Burger Names
CREATE TABLE burger_names(
   burger_id   INTEGER  NOT NULL PRIMARY KEY
  ,burger_name VARCHAR(10) NOT NULL
);

-- Inserting data into table Burger Names
INSERT INTO burger_names(burger_id,burger_name) VALUES (1,'Meatlovers');
INSERT INTO burger_names(burger_id,burger_name) VALUES (2,'Vegetarian');
```

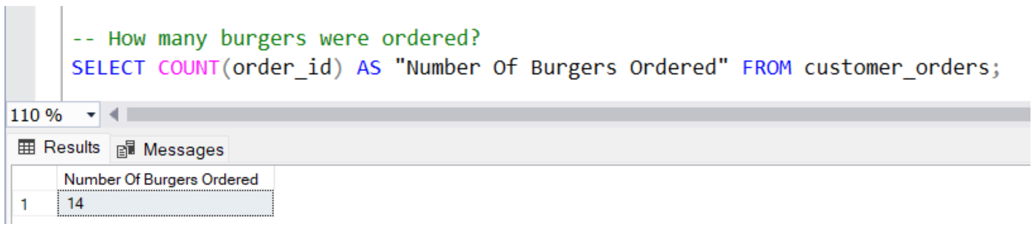
4. Customer Orders



```sql
-- Creation of table Customer Orders
CREATE TABLE customer_orders(
   order_id    INTEGER  NOT NULL
  ,customer_id INTEGER  NOT NULL
  ,burger_id    INTEGER  NOT NULL
  ,exclusions  VARCHAR(4)
  ,extras      VARCHAR(4)
  ,order_time  datetime NOT NULL
);

-- Inserting data into table Customer Orders
INSERT INTO customer_orders VALUES (1,101,1,NULL,NULL,'2021-01-01 18:05:02');
INSERT INTO customer_orders VALUES (2,101,1,NULL,NULL,'2021-01-01 19:00:52');
INSERT INTO customer_orders VALUES (3,102,1,NULL,NULL,'2021-01-02 23:51:23');
INSERT INTO customer_orders VALUES (3,102,2,NULL,NULL,'2021-01-02 23:51:23');
INSERT INTO customer_orders VALUES (4,103,1,'4',NULL,'2021-01-04 13:23:46');
INSERT INTO customer_orders VALUES (4,103,1,'4',NULL,'2021-01-04 13:23:46');
INSERT INTO customer_orders VALUES (4,103,2,'4',NULL,'2021-01-04 13:23:46');
INSERT INTO customer_orders VALUES (5,104,1,NULL,'1','2021-01-08 21:00:29');
INSERT INTO customer_orders VALUES (6,101,2,NULL,NULL,'2021-01-08 21:03:13');
INSERT INTO customer_orders VALUES (7,105,2,NULL,'1','2021-01-08 21:20:29');
INSERT INTO customer_orders VALUES (8,102,1,NULL,NULL,'2021-01-09 23:54:33');
INSERT INTO customer_orders VALUES (9,103,1,'4','1, 5','2021-01-10 11:22:59');
INSERT INTO customer_orders VALUES (10,104,1,NULL,NULL,'2021-01-11 18:34:49');
INSERT INTO customer_orders VALUES (10,104,1,'2, 6','1, 4','2021-01-11 18:34:49');
```

Queries

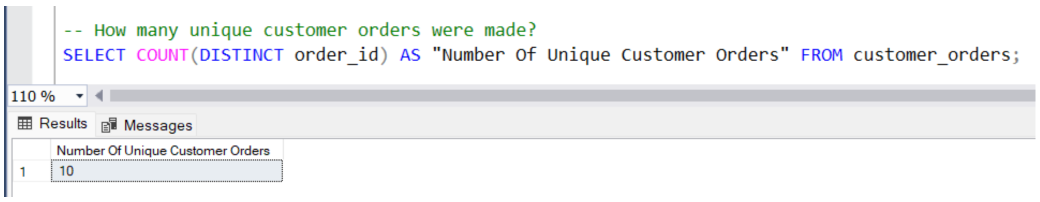
1. How many burgers were ordered?



```sql
-- How many burgers were ordered?
SELECT COUNT(order_id) AS "Number Of Burgers Ordered" FROM customer_orders;
```

	Number Of Burgers Ordered
1	14

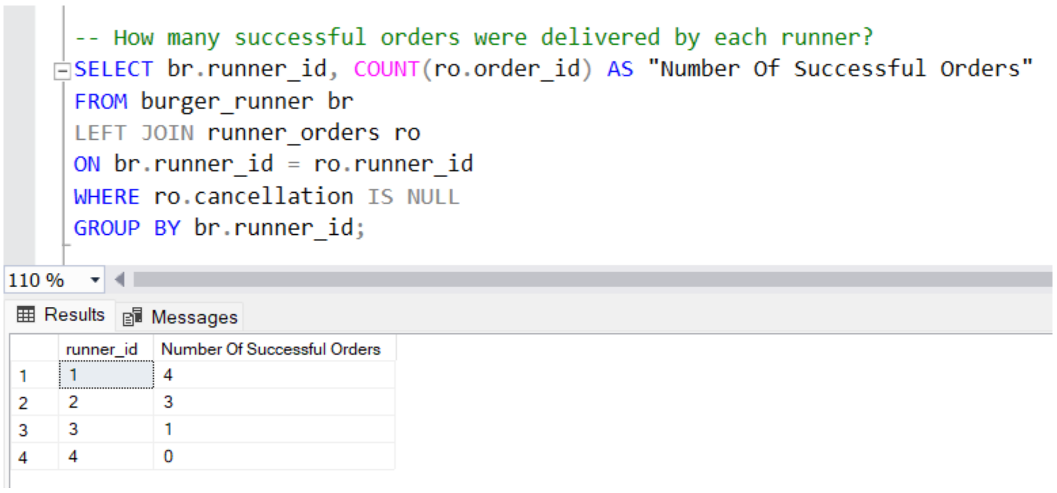
2. How many unique customer orders were made?



```sql
-- How many unique customer orders were made?
SELECT COUNT(DISTINCT order_id) AS "Number Of Unique Customer Orders" FROM customer_orders;
```

	Number Of Unique Customer Orders
1	10

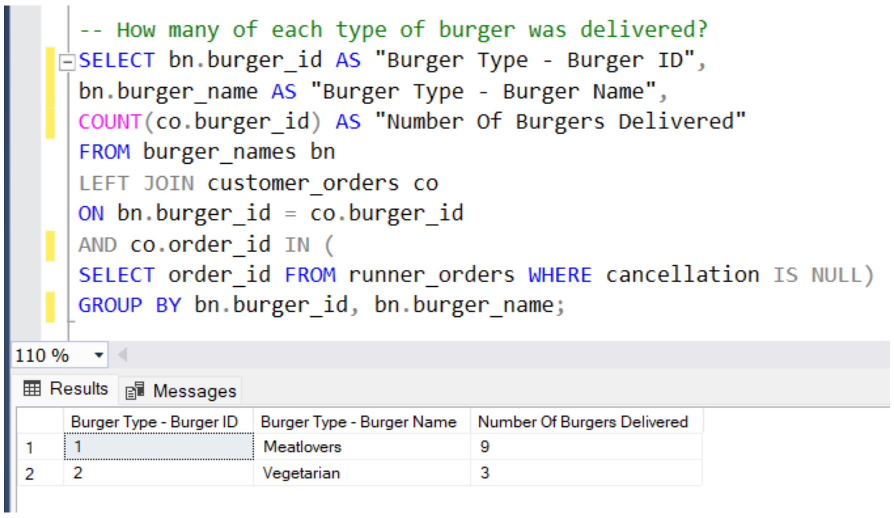
3. How many successful orders were delivered by each runner?



```sql
-- How many successful orders were delivered by each runner?
SELECT br.runner_id, COUNT(ro.order_id) AS "Number Of Successful Orders"
FROM burger_runner br
LEFT JOIN runner_orders ro
ON br.runner_id = ro.runner_id
WHERE ro.cancellation IS NULL
GROUP BY br.runner_id;
```

	runner_id	Number Of Successful Orders
1	1	4
2	2	3
3	3	1
4	4	0

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



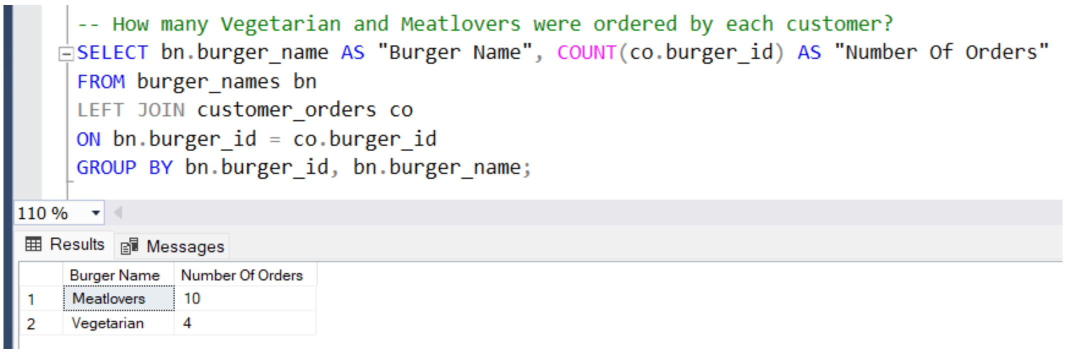
```
-- How many of each type of burger was delivered?
SELECT bn.burger_id AS "Burger Type - Burger ID",
bn.burger_name AS "Burger Type - Burger Name",
COUNT(co.burger_id) AS "Number Of Burgers Delivered"
FROM burger_names bn
LEFT JOIN customer_orders co
ON bn.burger_id = co.burger_id
AND co.order_id IN (
SELECT order_id FROM runner_orders WHERE cancellation IS NULL)
GROUP BY bn.burger_id, bn.burger_name;
```

110 %

Results | Messages

	Burger Type - Burger ID	Burger Type - Burger Name	Number Of Burgers Delivered
1	1	Meatlovers	9
2	2	Vegetarian	3

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



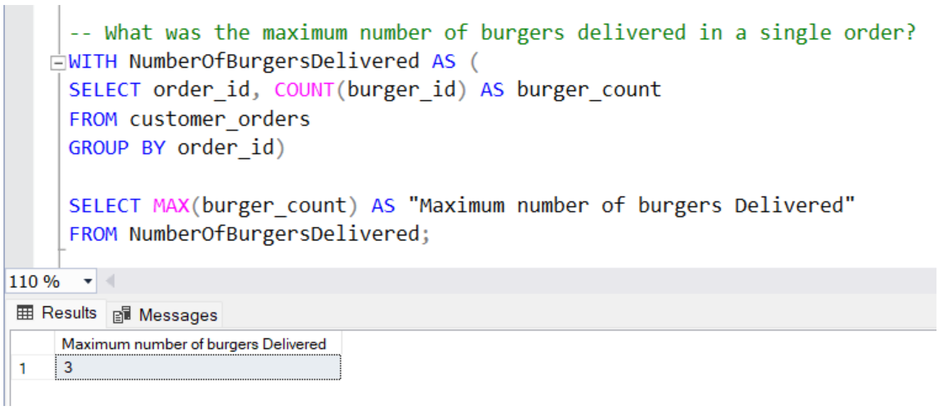
```
-- How many Vegetarian and Meatlovers were ordered by each customer?
SELECT bn.burger_name AS "Burger Name", COUNT(co.burger_id) AS "Number Of Orders"
FROM burger_names bn
LEFT JOIN customer_orders co
ON bn.burger_id = co.burger_id
GROUP BY bn.burger_id, bn.burger_name;
```

110 %

Results | Messages

	Burger Name	Number Of Orders
1	Meatlovers	10
2	Vegetarian	4

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



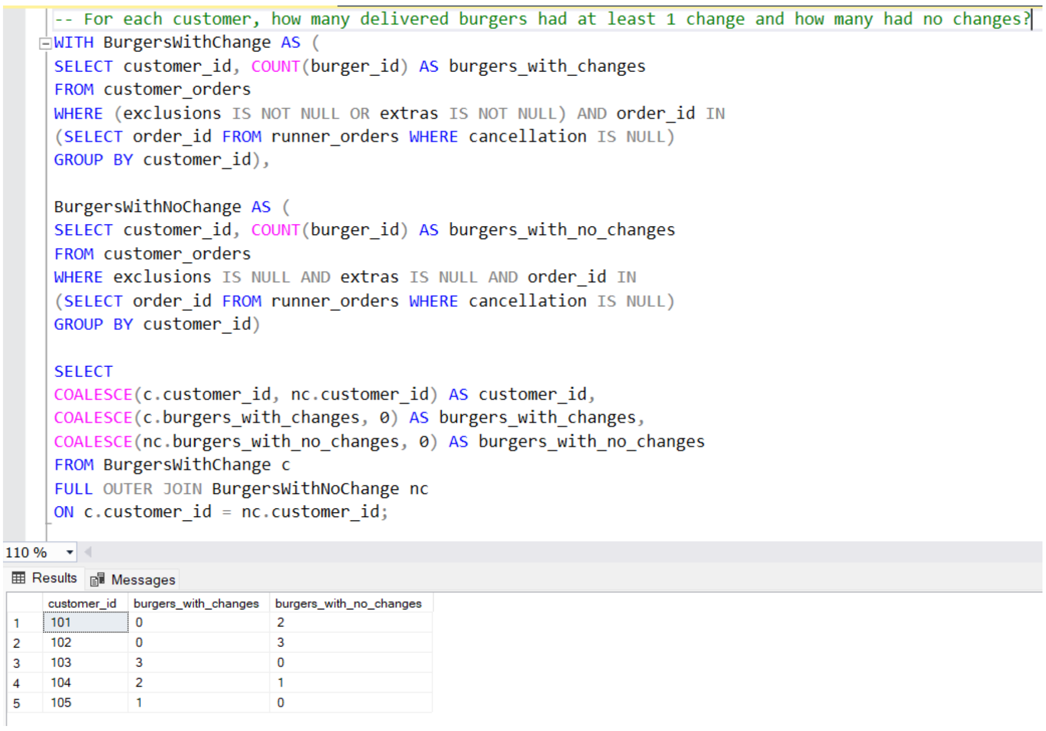
```
-- What was the maximum number of burgers delivered in a single order?
WITH NumberOfBurgersDelivered AS (
SELECT order_id, COUNT(burger_id) AS burger_count
FROM customer_orders
GROUP BY order_id)

SELECT MAX(burger_count) AS "Maximum number of burgers Delivered"
FROM NumberOfBurgersDelivered;
```

110 %

Results | Messages

	Maximum number of burgers Delivered
1	3

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



```sql
-- For each customer, how many delivered burgers had at least 1 change and how many had no changes?
WITH BurgersWithChange AS (
SELECT customer_id, COUNT(burger_id) AS burgers_with_changes
FROM customer_orders
WHERE (exclusions IS NOT NULL OR extras IS NOT NULL) AND order_id IN
(SELECT order_id FROM runner_orders WHERE cancellation IS NULL)
GROUP BY customer_id),

BurgersWithNoChange AS (
SELECT customer_id, COUNT(burger_id) AS burgers_with_no_changes
FROM customer_orders
WHERE exclusions IS NULL AND extras IS NULL AND order_id IN
(SELECT order_id FROM runner_orders WHERE cancellation IS NULL)
GROUP BY customer_id)

SELECT
COALESCE(c.customer_id, nc.customer_id) AS customer_id,
COALESCE(c.burgers_with_changes, 0) AS burgers_with_changes,
COALESCE(nc.burgers_with_no_changes, 0) AS burgers_with_no_changes
FROM BurgersWithChange c
FULL OUTER JOIN BurgersWithNoChange nc
ON c.customer_id = nc.customer_id;
```

110 %

Results Messages

	customer_id	burgers_with_changes	burgers_with_no_changes
1	101	0	2
2	102	0	3
3	103	3	0
4	104	2	1
5	105	1	0