Burger Bash SQL Case Study – Documentation

1. Project Objective

The purpose of this case study is to analyze burger order and delivery data from a fictional business called Burger Bash.

2. Tools Used

- ✔ **SQL Platform:** Microsoft SQL Server (SSMS)
- ✔ **Language:** T-SQL (Transact-SQL)
- ✔ **Data Source:** Manually inserted via SQL INSERT statements
- ✔ **Environment:** Windows PC, SSMS interface

3. Schema Creation

3.1 Create Database

4. Table Creation

Created 4 tables in total using SQL Server-compatible data types.

4.1 burger_names

4.2 runner_orders

```sql
-- Create table runner_orders
CREATE TABLE runner_orders(
    order_id INT NOT NULL PRIMARY KEY,
    runner_id INT NOT NULL,
    pickup_time datetime2,
    distance VARCHAR(20),
    duration VARCHAR(20),
    cancellation VARCHAR(50)
);

SELECT * FROM runner_orders;
```

No issues found

order_id	runner_id	pickup_time	distance	duration	cancellation

4.3 burger_runner

```sql
-- Create table burger_runner
CREATE TABLE burger_runner(
    runner_id INT NOT NULL PRIMARY KEY,
    registration_date date NOT NULL
);

SELECT * FROM burger_runner;
```

No issues found

runner_id	registration_date

4.4 customer_orders

```sql
-- Create table customer_orders
CREATE TABLE customer_orders(
    order_id INT NOT NULL,
    customer_id INT NOT NULL,
    burger_id INT NOT NULL,
    exclusions VARCHAR(20),
    extras VARCHAR(20),
    order_time datetime2 NOT NULL
);

SELECT * FROM customer_orders;
```

No issues found

order_id	customer_id	burger_id	exclusions	extras	order_time

5. Data Insertion

Each table was populated manually using INSERT INTO statements.

5.1 burger_names

burger_names...Ragavi (74))* | create_db.sql...\Ragavi (71))

```sql
-- Create table burger_names
CREATE TABLE burger_names(
    burger_id INT NOT NULL PRIMARY KEY,
    burger_name VARCHAR(10) NOT NULL
);

-- Insert values into burger_names
INSERT INTO burger_names VALUES
(1,'Meatlovers'),
(2,'Vegetarian');

SELECT * FROM burger_names;
```

100 % No issues found

Results | Messages

	burger_id	burger_name
1	1	Meatlovers
2	2	Vegetarian

5.2 runner_orders

runner_order...\Ragavi (52))

```sql
    runner_id INT NOT NULL,
    pickup_time datetime2,
    distance VARCHAR(20),
    duration VARCHAR(20),
    cancellation VARCHAR(50)
);


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

SELECT * FROM runner_orders;
```

100 % No issues found

Results | Messages

	order_id	runner_id	pickup_time	distance	duration	cancellation
1	1	1	2021-01-01 18:15:34.0000000	20km	32 minutes	NULL
2	2	1	2021-01-01 19:10:54.0000000	20km	27 minutes	NULL
3	3	1	2021-01-03 00:12:37.0000000	13.4km	20 mins	NULL
4	4	2	2021-01-04 13:53:03.0000000	23.4	40	NULL
5	5	3	2021-01-08 21:10:57.0000000	10	15	NULL
6	6	3	NULL	NULL	NULL	Restaurant Cancellation
7	7	2	2021-01-08 21:30:45.0000000	25km	25mins	NULL
8	8	2	2021-01-10 00:15:02.0000000	23.4 km	15 minute	NULL
9	9	2	NULL	NULL	NULL	Customer Cancellation
10	10	1	2021-01-11 18:50:20.0000000	10km	10minutes	NULL

5.3 burger_runner

burger_runne...\Ragavi (71))

```sql
-- Create table burger_runner
CREATE TABLE burger_runner(
   runner_id INT NOT NULL PRIMARY KEY,
   registration_date date NOT NULL
);


-- Insert values into burger_runner
INSERT INTO burger_runner VALUES
(1, '2021-01-01'),
(2, '2021-01-03'),
(3, '2021-01-08'),
(4, '2021-01-15');

SELECT * FROM burger_runner;
```

100 % No issues found

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

5.4 customer_orders

customer_ord...\Ragavi (52))

```sql

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

SELECT * FROM customer_orders;
```

100 % No issues found

Results Messages

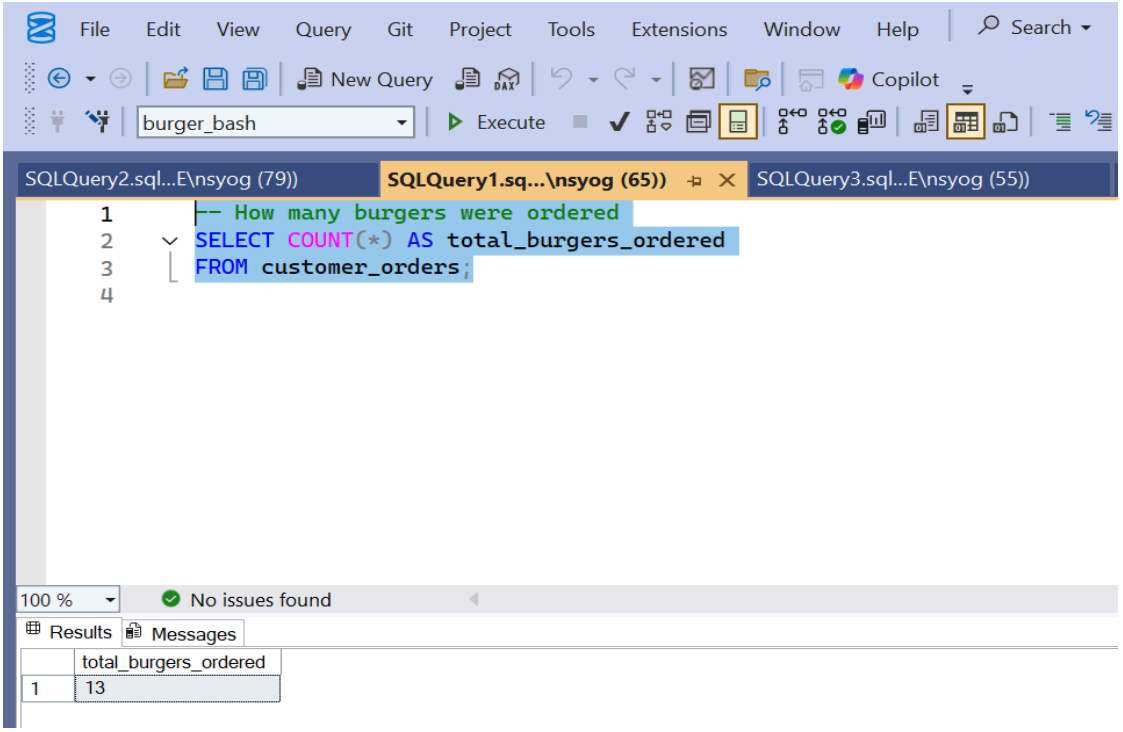
	order_id	customer_id	burger_id	exclusions	extras	order_time
1	1	101	1	NULL	NULL	2021-01-01 18:05:02.0000000
2	2	101	1	NULL	NULL	2021-01-01 19:00:52.0000000
3	3	102	1	NULL	NULL	2021-01-02 23:51:23.0000000
4	3	102	2	NULL	NULL	2021-01-02 23:51:23.0000000
5	4	103	1	4	NULL	2021-01-04 13:23:46.0000000
6	4	103	1	4	NULL	2021-01-04 13:23:46.0000000
7	4	103	2	4	NULL	2021-01-04 13:23:46.0000000
8	5	104	1	NULL	1	2021-01-08 21:00:29.0000000
9	6	101	2	NULL	NULL	2021-01-08 21:03:13.0000000
10	7	105	2	NULL	1	2021-01-08 21:20:29.0000000
11	8	102	1	NULL	NULL	2021-01-09 23:54:33.0000000
12	9	103	1	4	1, 5	2021-01-10 11:22:59.0000000
13	10	104	1	NULL	NULL	2021-01-11 18:34:49.0000000
14	10	104	1	2, 6	1, 4	2021-01-11 18:34:49.0000000

Query executed successfully. LAPTOP-O11N

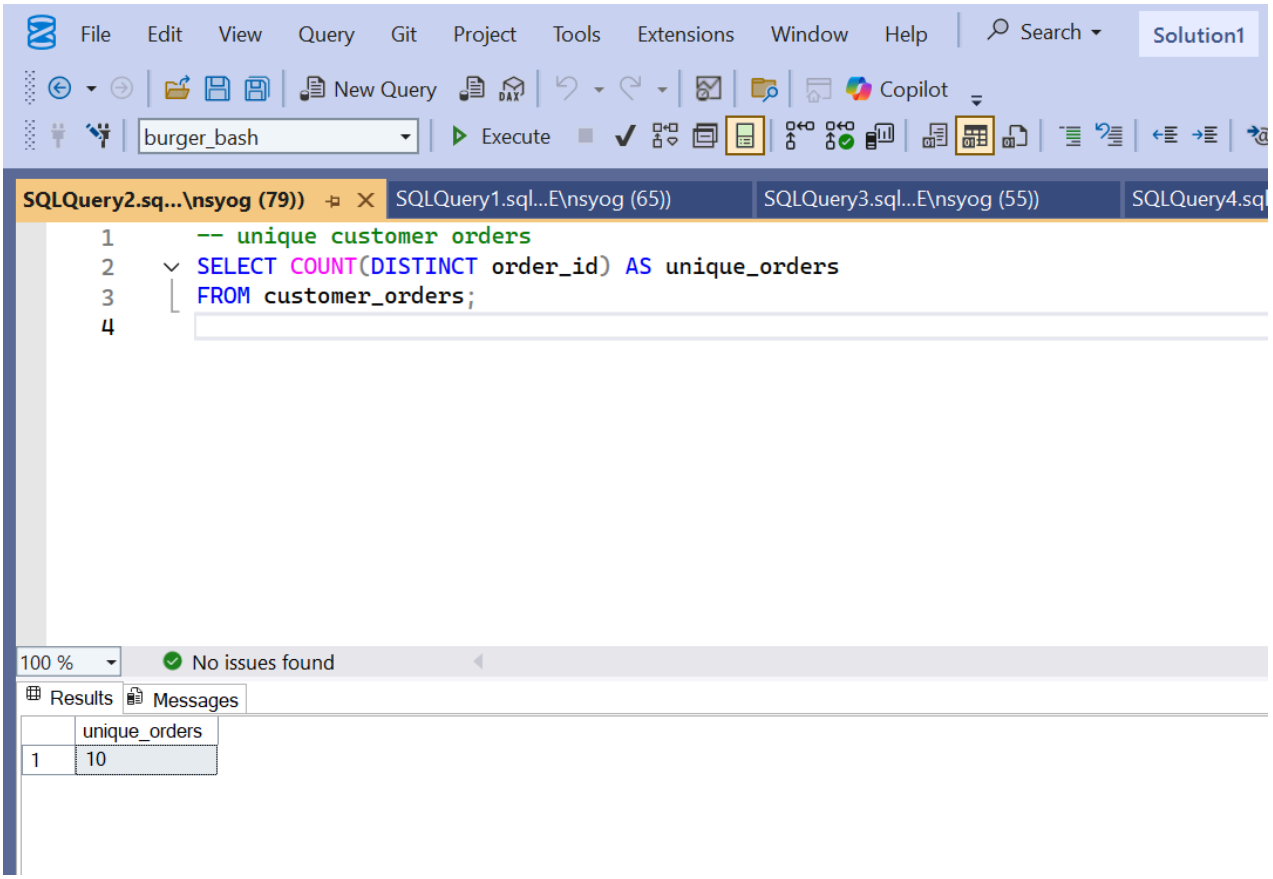
6. Analysis

The following queries were written and executed to analyze Burger Bash operations.

Q1: How many burgers were ordered?



Q2: How many unique customer orders were made?



Q3: How many successful orders were delivered by each runner?

```sql
-- How many successful orders were delivered by each runner
SELECT runner_id, COUNT(*) AS successful_orders
FROM runner_orders
WHERE cancellation IS NULL
GROUP BY runner_id;
```

No issues found

	runner_id	successful_orders
1	1	4
2	2	3
3	3	1

Q4: How many of each type of burger was delivered?

```sql
-- How many of each type of burger was delivered
SELECT b.burger_name, COUNT(*) AS delivered_count
FROM customer_orders c
JOIN runner_orders r ON c.order_id = r.order_id
JOIN burger_names b ON c.burger_id = b.burger_id
WHERE r.cancellation IS NULL
GROUP BY b.burger_name;
```

No issues found

	burger_name	delivered_count
1	Meatlovers	8
2	Vegetarian	3

Q5: How many Vegetarian and Meatlovers were ordered by each customer?

SQLQuery5.sq...\nsyog (52)) | SQLQuery4.sql...E\nsyog (54)) | SQLQuery3.sql...E\nsyog (55))

```sql
-- How many Vegetarian and Meatlovers were ordered by each customer
SELECT
  c.customer_id,
  b.burger_name,
  COUNT(*) AS order_count
FROM customer_orders c
JOIN burger_names b ON c.burger_id = b.burger_id
GROUP BY c.customer_id, b.burger_name
HAVING b.burger_name IN ('Vegetarian', 'Meatlovers');
```

100 % No issues found

Results | Messages

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