**Assignment – Case Study-3 Burger Bash**

By Afreen Sultana A

create database BurgerBash;

use BurgerBash;

create table burger\_runner (

runner\_id INT PRIMARY KEY,

registration\_date DATE

);

create table burger\_names (

burger\_id INT PRIMARY KEY,

burger\_name VARCHAR(50)

);

create table customer\_orders (

order\_id INT,

customer\_id INT,

burger\_id INT,

exclusions VARCHAR(100),

extras VARCHAR(100),

order\_time DATETIME

);

create table runner\_orders (

order\_id INT,

runner\_id INT,

pickup\_time DATETIME,

distance VARCHAR(20),

duration VARCHAR(20),

cancellation VARCHAR(20)

);

insert into burger\_runner (runner\_id, registration\_date) values

(1, '2025-07-01'),

(2, '2025-07-02'),

(3, '2025-07-09'),

(4, '2025-07-12'),

(5, '2025-07-15');

insert into burger\_names (burger\_id, burger\_name) values

(1, 'Vegetarian'),

(2, 'Meatlovers'),

(3, 'Cheeseburger'),

(4, 'BBQ Burger'),

(5, 'Paneer Tikka Burger');

insert into customer\_orders (order\_id, customer\_id, burger\_id, exclusions, extras, order\_time) values

(101, 1001, 1, '', 'cheese', '2025-07-19 11:30:00'),

(102, 1002, 2, 'onion', '', '2025-07-19 12:45:00'),

(103, 1003, 1, '', '', '2025-07-19 14:15:00'),

(104, 1001, 3, 'lettuce', 'bacon', '2025-07-19 15:30:00'),

(105, 1004, 2, '', '', '2025-07-19 16:00:00');

insert into runner\_orders (order\_id, runner\_id, pickup\_time, distance, duration, cancellation) values

(101, 1, '2025-07-19 11:45:00', '5 km', '20 minutes', NULL),

(102, 2, '2025-07-19 13:00:00', '3.5 km', '15 minutes', NULL),

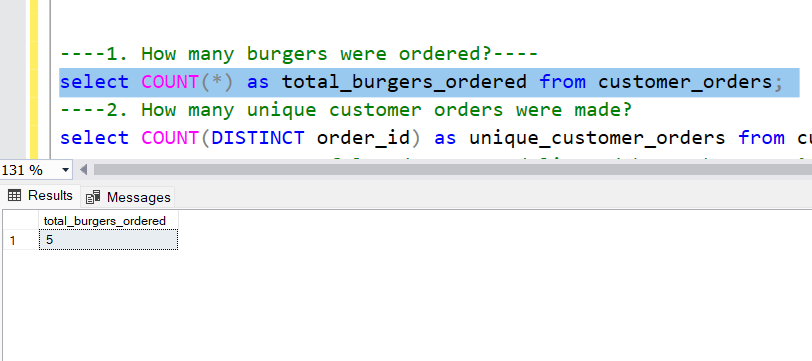
(103, 3, '2025-07-19 14:30:00', '7.2 km', '25 minutes', NULL),

(104, 4, '2025-07-19 15:50:00', '4 km', '18 minutes', NULL),

(105, 5, NULL, NULL, NULL, 'Cancelled');

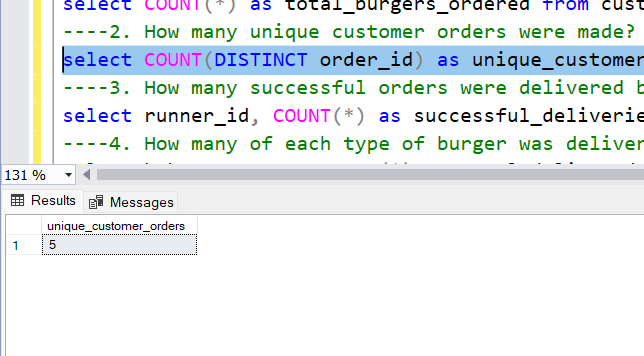
**----1. How many burgers were ordered?----**

select COUNT(\*) as total\_burgers\_ordered from customer\_orders;



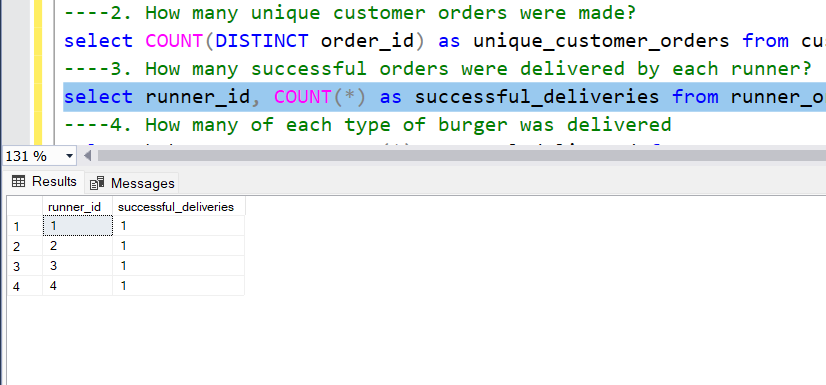
**----2. How many unique customer orders were made?**

select COUNT(DISTINCT order\_id) as unique\_customer\_orders from customer\_orders;



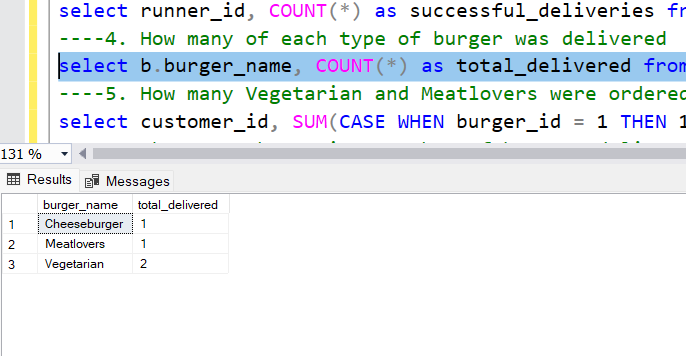
**----3. How many successful orders were delivered by each runner?**

select runner\_id, COUNT(\*) as successful\_deliveries from runner\_orders where cancellation IS NULL OR cancellation = 'null' group by runner\_id;



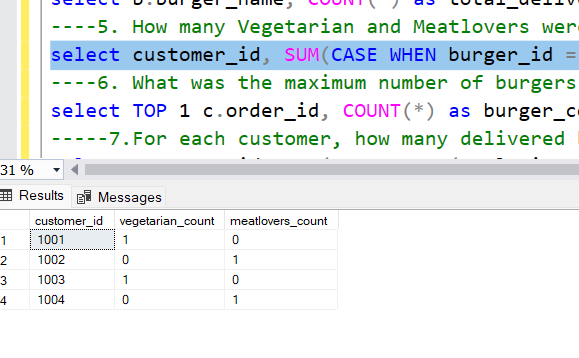
**----4. How many of each type of burger was delivered**

select b.burger\_name, COUNT(\*) as total\_delivered 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 OR r.cancellation = 'null' group by b.burger\_name;



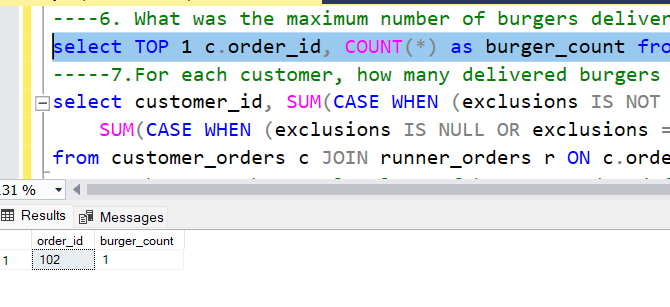
**----5. How many Vegetarian and Meatlovers were ordered by each customer?**

select customer\_id, SUM(CASE WHEN burger\_id = 1 THEN 1 ELSE 0 END) as vegetarian\_count, SUM(CASE WHEN burger\_id = 2 THEN 1 ELSE 0 END) as meatlovers\_count from customer\_orders group by customer\_id;



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

select TOP 1 c.order\_id, COUNT(\*) as burger\_count from customer\_orders c JOIN runner\_orders r ON c.order\_id = r.order\_id where r.cancellation IS NULL OR r.cancellation = 'null' group by c.order\_id order by burger\_count DESC;

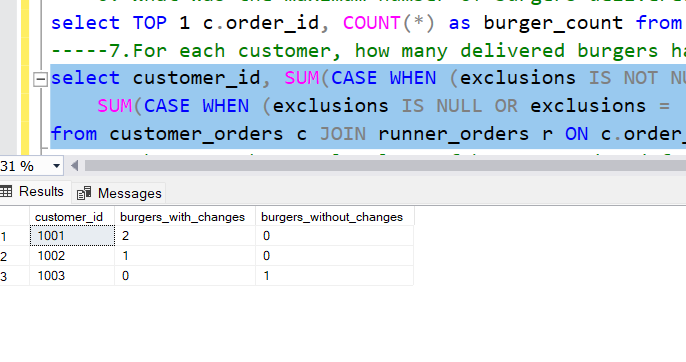


**-----7.For each customer, how many delivered burgers had at least 1 change and how many had no changes?**

select customer\_id, SUM(CASE WHEN (exclusions IS NOT NULL AND exclusions <> '') OR (extras IS NOT NULL AND extras <> '') THEN 1 ELSE 0 END) AS burgers\_with\_changes,

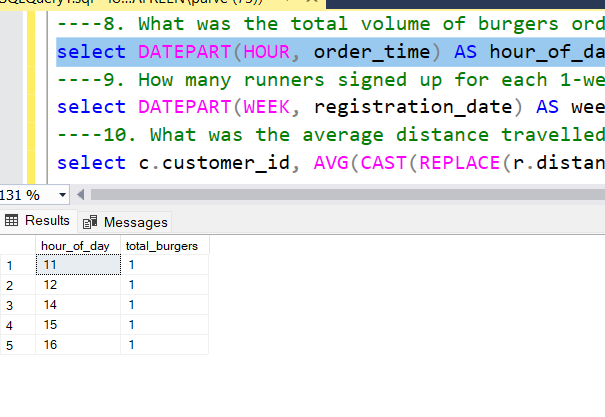
SUM(CASE WHEN (exclusions IS NULL OR exclusions = '') AND (extras IS NULL OR extras = '') THEN 1 ELSE 0 END) AS burgers\_without\_changes

from customer\_orders c JOIN runner\_orders r ON c.order\_id = r.order\_id where r.cancellation IS NULL OR r.cancellation = 'null' group by customer\_id;



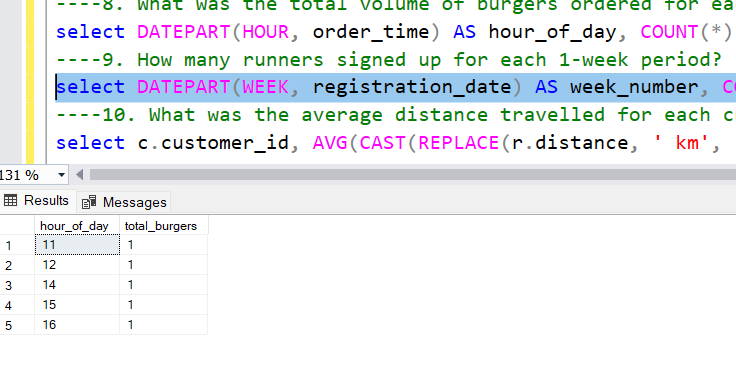
**----8. What was the total volume of burgers ordered for each hour of the day?**

select DATEPART(HOUR, order\_time) AS hour\_of\_day, COUNT(\*) AS total\_burgers from customer\_orders group by DATEPART(HOUR, order\_time) order by hour\_of\_day;



**----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;



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

select c.customer\_id, AVG(CAST(REPLACE(r.distance, ' km', '') AS FLOAT)) AS avg\_distance\_km from customer\_orders c JOIN runner\_orders r ON c.order\_id = r.order\_id where r.cancellation IS NULL OR r.cancellation = 'null' group by c.customer\_id;

