**SQL CASE STUDY – BURGER BASH**

**Name: Vikas Reddy Gorantla**

**Date: 09-11-2024**

use [Burger Bash];

**Creating customer\_orders Table:**

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

);

ALTER TABLE customer\_orders

ADD CONSTRAINT fk\_customer\_orders\_burger

FOREIGN KEY (burger\_id) REFERENCES burger\_names(burger\_id);

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');

A screenshot of a table

Description automatically generated

**Creating burger\_runner Table:**

CREATE TABLE burger\_runner (

runner\_id INTEGER NOT NULL PRIMARY KEY,

registration\_date DATE NOT NULL

);

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');

A screenshot of a computer

Description automatically generated

**Creating runner\_orders Table:**

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)

);

ALTER TABLE runner\_orders

ADD CONSTRAINT fk\_runner\_orders\_runner

FOREIGN KEY (runner\_id) REFERENCES burger\_runner(runner\_id);

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);

A screenshot of a computer

Description automatically generated

**Creating burger\_names Table:**

CREATE TABLE burger\_names (

burger\_id INTEGER NOT NULL PRIMARY KEY,

burger\_name VARCHAR(10) NOT NULL

);

INSERT INTO burger\_names (burger\_id, burger\_name) VALUES (1, 'Meatlovers');

INSERT INTO burger\_names (burger\_id, burger\_name) VALUES (2, 'Vegetarian');

A screenshot of a computer

Description automatically generated

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

select count(order\_id) as Total\_Burgers\_Ordered from customer\_orders;

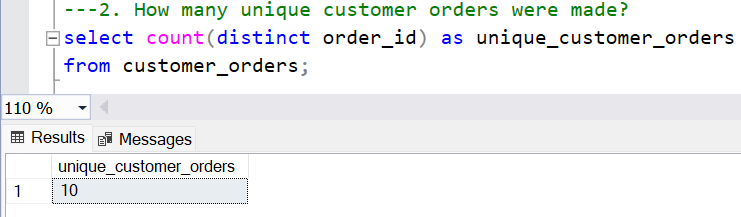
A screenshot of a computer

Description automatically generated

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

select count(distinct order\_id) as unique\_customer\_orders

from customer\_orders;



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

select count(order\_id) as sucessfull\_orders

from runner\_orders

WHERE cancellation IS NULL

group by runner\_id;

A screen shot of a computer code

Description automatically generated

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

SELECT b.burger\_name, COUNT(c.burger\_id) AS burger\_count

FROM customer\_orders c

JOIN burger\_names b ON b.burger\_id = c.burger\_id

GROUP BY b.burger\_name;

A screenshot of a computer program

Description automatically generated

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

SELECT c.customer\_id, b.burger\_name, COUNT(c.burger\_id) AS burger\_count

FROM customer\_orders c

JOIN burger\_names b ON b.burger\_id = c.burger\_id

WHERE b.burger\_name IN ('Meatlovers', 'Vegetarian')

GROUP BY c.customer\_id, b.burger\_name;

A screenshot of a computer

Description automatically generated

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

SELECT MAX(order\_count) AS max\_burgers\_in\_single\_order

FROM (

SELECT order\_id, COUNT(burger\_id) AS order\_count

FROM customer\_orders

GROUP BY order\_id

) AS order\_counts;

A screenshot of a computer code

Description automatically generated