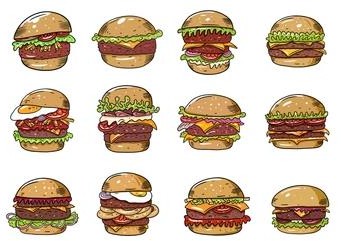
**SQL Case Study 2: Burger Bash**

****

**INTRODUCTION:**

I have a started a new business of selling burger because I read on my Instagram feed that „Burger Is the Future!

But I knew that burger alone was not going to help me get seed funding to expand my new Burger Empire - so I had one more genius idea to combine with it - I was going to Uberize it - and so Burger Runner was launched!

I started by recruiting “runners” to deliver fresh burger from Burger Runner Headquarters and also maxed out my credit card to pay freelance developers to build a mobile app to accept orders from customers.

**SCHEMA USED**

**burger\_runner** runner\_id int registration\_date date

**customer\_orders**

order\_id int

customer\_id int

burger\_id int

exclusions varchar

extras varchar

order\_time timestamp

|  |  |
| --- | --- |
| **runner\_orders** | |
| order\_id | int |
| runner\_id | int |
| pickup\_time | timestamp |
| distance | varchar |
| duration | varchar |
| cancellation | varchar |

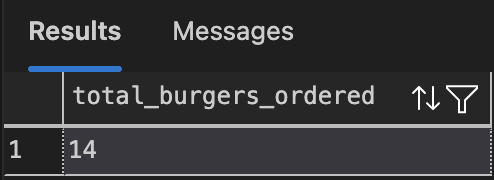
|  |  |
| --- | --- |
| **burger\_names** | |
| burger\_id | int |
| burger\_name | varchar |

**CASE STUDY QUESTIONS**

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

SELECT COUNT(\*) AS total\_burgers\_ordered

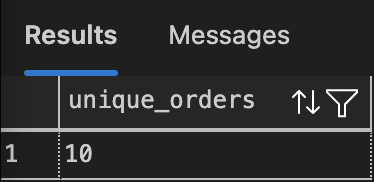
FROM customer\_orders;



1. **How many unique customer orders were made?**

SELECT COUNT(DISTINCT order\_id) AS unique\_orders

FROM customer\_orders;



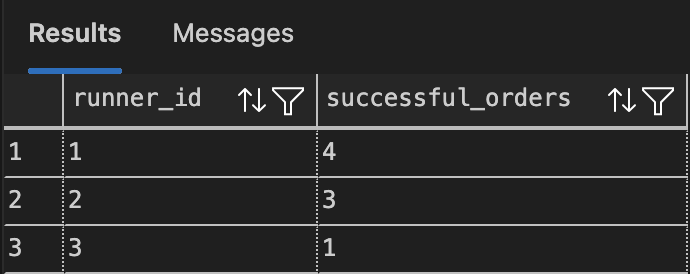
1. **How many successful orders were delivered by each runner?**

SELECT runner\_id, COUNT(\*) AS successful\_orders

FROM runner\_orders

WHERE cancellation IS NULL

GROUP BY runner\_id;



1. **How many of each type of burger was delivered?**

SELECT bo.burger\_name, COUNT(\*) AS total\_delivered

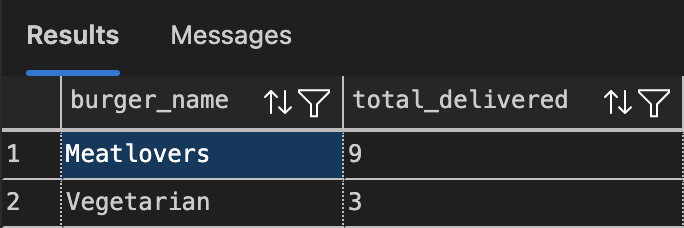
FROM customer\_orders co

JOIN burger\_names bo ON co.burger\_id = bo.burger\_id

JOIN runner\_orders ro ON co.order\_id = ro.order\_id

WHERE ro.cancellation IS NULL

GROUP BY bo.burger\_name;



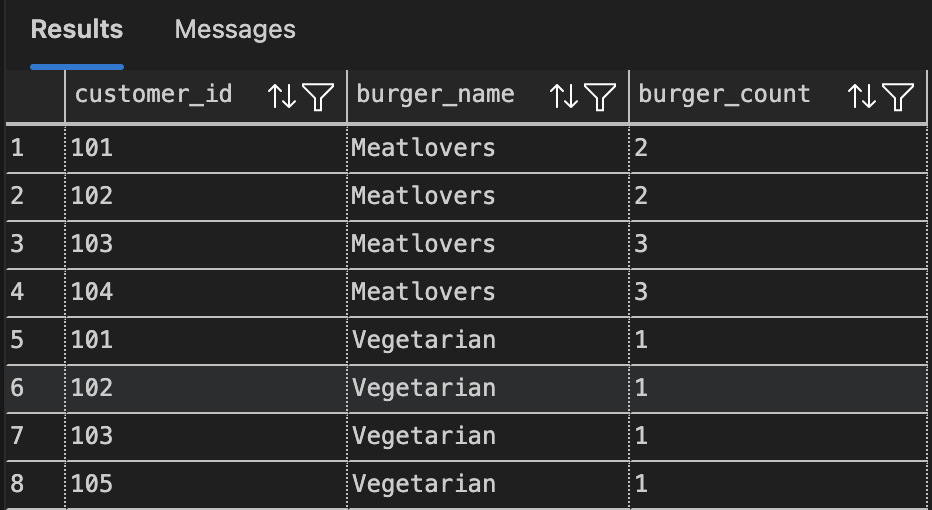
1. **How many Vegetarian and Meatlovers were ordered by each customer?**

SELECT co.customer\_id, bn.burger\_name, COUNT(\*) AS burger\_count

FROM customer\_orders co

JOIN burger\_names bn ON co.burger\_id = bn.burger\_id

GROUP BY co.customer\_id, bn.burger\_name;



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

SELECT TOP 1 order\_id, COUNT(\*) AS burger\_count

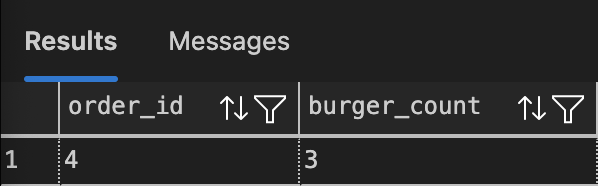
FROM customer\_orders co

JOIN runner\_orders ro ON co.order\_id = ro.order\_id

WHERE ro.cancellation IS NULL

GROUP BY co.order\_id

ORDER BY burger\_count DESC;



1. **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 OR extras IS NOT NULL THEN 1 ELSE 0 END) AS changed\_burgers,

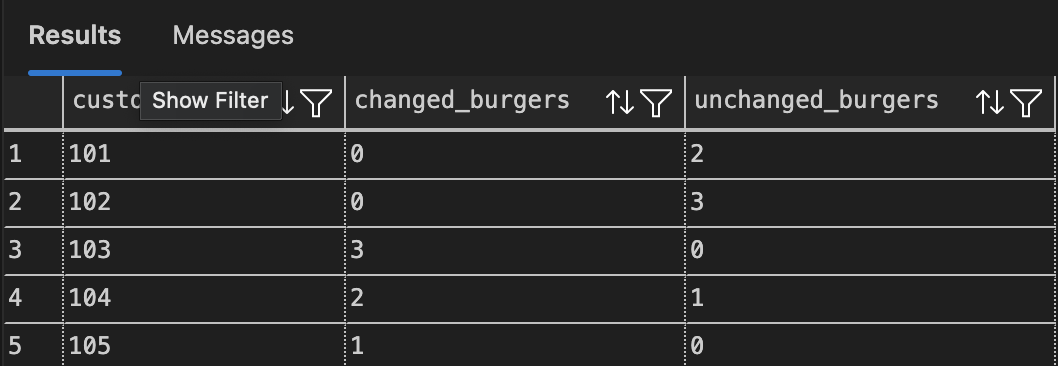
SUM(CASE WHEN exclusions IS NULL AND extras IS NULL THEN 1 ELSE 0 END) AS unchanged\_burgers

FROM customer\_orders co

JOIN runner\_orders ro ON co.order\_id = ro.order\_id

WHERE ro.cancellation IS NULL

GROUP BY customer\_id;



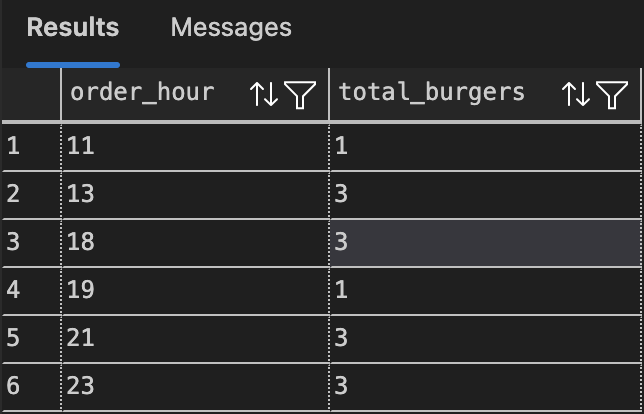
1. **What was the total volume of burgers ordered for each hour of the day?**

SELECT DATEPART(HOUR, order\_time) AS order\_hour, COUNT(\*) AS total\_burgers

FROM customer\_orders

GROUP BY DATEPART(HOUR, order\_time)

ORDER BY order\_hour;



1. **How many runners signed up for each 1 week period?**

SELECT

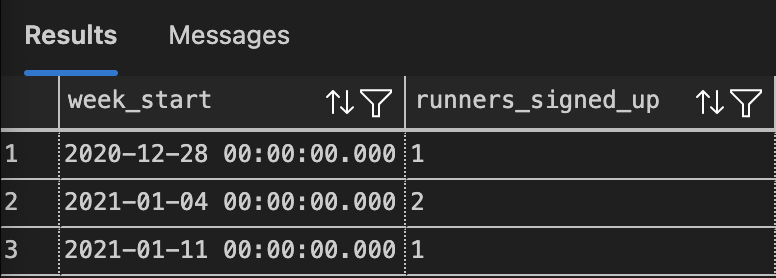
DATEADD(WEEK, DATEDIFF(WEEK, 0, registration\_date), 0) AS week\_start,

COUNT(\*) AS runners\_signed\_up

FROM burger\_runner

GROUP BY DATEADD(WEEK, DATEDIFF(WEEK, 0, registration\_date), 0)

ORDER BY week\_start;



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

SELECT co.customer\_id,

ROUND(AVG(CAST(REPLACE(ro.distance, 'km', '') AS FLOAT)), 2) AS avg\_distance\_km

FROM customer\_orders co

JOIN runner\_orders ro ON co.order\_id = ro.order\_id

WHERE ro.cancellation IS NULL

GROUP BY co.customer\_id;

