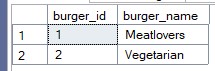
**CODING ASSESSMENT – 21-07-2025**

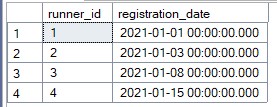
USE burger\_bash;

**TABLES**

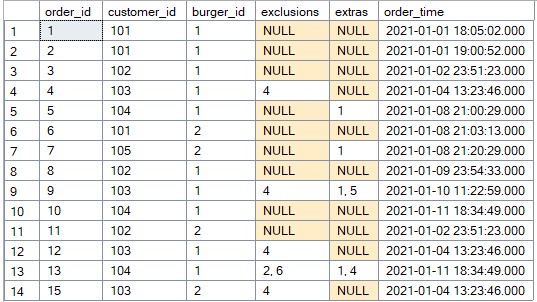
**1.burger\_names**



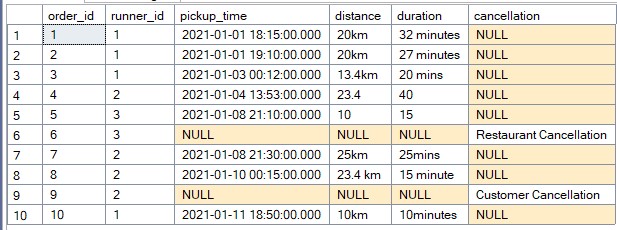
**2.burger\_runner**



**3.customer\_orders**



**4.burger\_runner\_orders**



**-- 1. Total Revenue Generated per Runner**

SELECT

r.runner\_id,

SUM(CASE WHEN bro.distance LIKE '%km%'

THEN CAST(REPLACE(bro.distance, 'km', '') AS FLOAT) \* 1.5 -- Assume €1.5/km

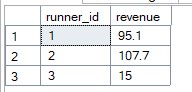
ELSE CAST(bro.distance AS FLOAT) \* 1.5 END) AS revenue

FROM burger\_runner\_orders bro

JOIN burger\_runner r ON bro.runner\_id = r.runner\_id

WHERE bro.cancellation IS NULL

GROUP BY r.runner\_id;



**-- 2. Average Delivery Time per Runner (Minutes)**

SELECT

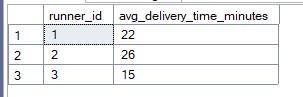
runner\_id,

AVG(CAST(SUBSTRING(duration, 1, PATINDEX('%[^0-9]%', duration + 'x') - 1) AS INT)) AS avg\_delivery\_time\_minutes

FROM burger\_runner\_orders

WHERE cancellation IS NULL AND duration IS NOT NULL

GROUP BY runner\_id;



**-- 3. Most Popular Burger with Exclusions/Extras**

SELECT TOP 1

b.burger\_name,

COUNT(\*) AS customized\_orders\_count

FROM customer\_orders co

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

WHERE co.exclusions IS NOT NULL OR co.extras IS NOT NULL

GROUP BY b.burger\_name

ORDER BY customized\_orders\_count DESC;



**--4. Percentage of Cancelled Orders by Cause**

SELECT

cancellation,

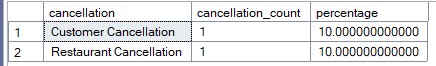
COUNT(\*) AS cancellation\_count,

ROUND(COUNT(\*) \* 100.0 / (SELECT COUNT(\*) FROM burger\_runner\_orders), 2) AS percentage

FROM burger\_runner\_orders

WHERE cancellation IS NOT NULL

GROUP BY cancellation;



**--5. Customer Order Frequency Analysis**

SELECT

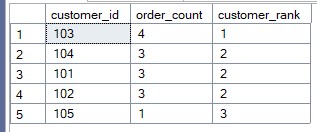
customer\_id,

COUNT(\*) AS order\_count,

DENSE\_RANK() OVER (ORDER BY COUNT(\*) DESC) AS customer\_rank

FROM customer\_orders

GROUP BY customer\_id;



**--6. Runner Efficiency by Order Volume & Distance**

SELECT

bro.runner\_id,

COUNT(bro.order\_id) AS completed\_orders,

SUM(CAST(REPLACE(REPLACE(bro.distance, 'km', ''), ' ', '') AS FLOAT)) AS total\_distance\_km,

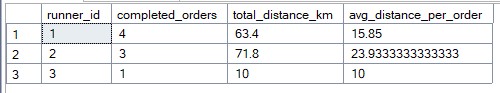
SUM(CAST(REPLACE(REPLACE(bro.distance, 'km', ''), ' ', '') AS FLOAT)) / COUNT(bro.order\_id) AS avg\_distance\_per\_order

FROM burger\_runner\_orders bro

WHERE bro.cancellation IS NULL

GROUP BY bro.runner\_id

ORDER BY completed\_orders DESC;



**--7. Minimum Delivery Time (Fastest Order)**

SELECT

MIN(CAST(SUBSTRING(duration, 1, PATINDEX('%[^0-9]%', duration + 'x') - 1) AS INT)) AS min\_delivery\_time\_minutes

FROM burger\_runner\_orders

WHERE cancellation IS NULL AND duration IS NOT NULL;



**--8. Maximum Distance Covered for Successful Deliveries**

SELECT

MAX(CAST(REPLACE(REPLACE(distance, 'km', ''), ' ', '') AS FLOAT)) AS max\_distance\_km

FROM burger\_runner\_orders

WHERE cancellation IS NULL AND distance IS NOT NULL;



**--9. Runner with Minimum Deliveries (WHERE + Aggregate)**

SELECT TOP 1

runner\_id,

COUNT(order\_id) AS deliveries

FROM burger\_runner\_orders

WHERE cancellation IS NULL

GROUP BY runner\_id

ORDER BY deliveries ASC;

SELECT \* FROM customer\_orders;

