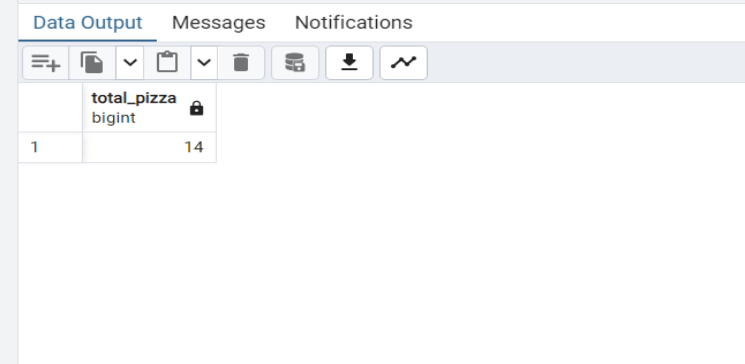
Trịnh Hoàng Phú 21110370

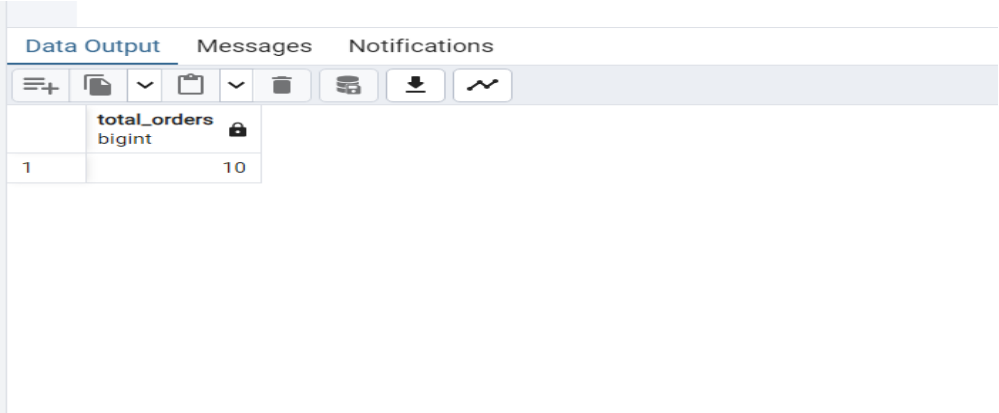
Câu 1

```
SELECT COUNT( order_id) AS total_pizza
FROM customer_orders;
```



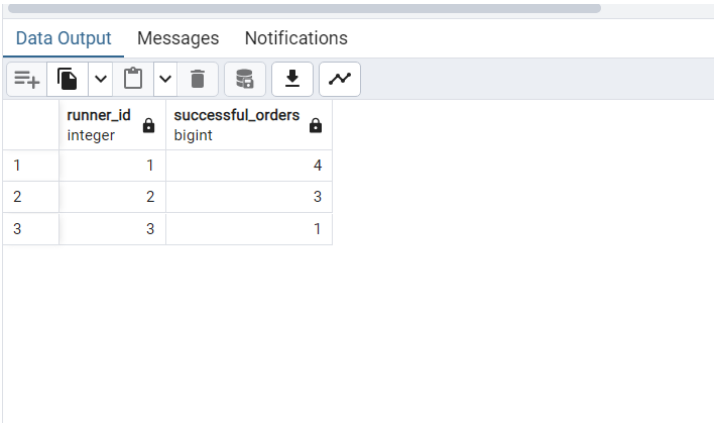
Câu 2

```
SELECT COUNT(DISTINCT order_id) AS total_orders
FROM customer_orders;
```



Câu 3

```
SELECT ro.runner_id, COUNT(ro.order_id) AS successful_orders
FROM runner_orders ro
WHERE ro.cancellation IS NULL OR ro.cancellation = '' or ro.cancellation like 'null'
GROUP BY ro.runner_id;
```

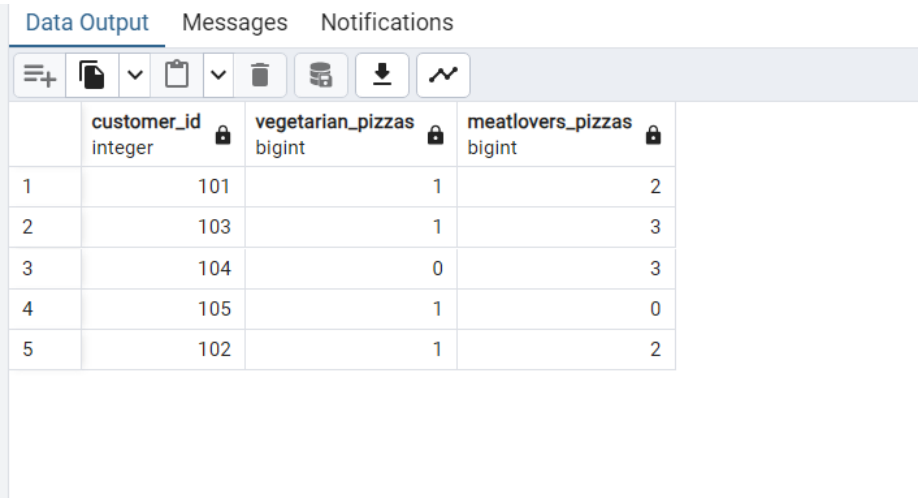


Data Output | Messages | Notifications

	runner_id integer	successful_orders bigint
1	1	4
2	2	3
3	3	1

Câu 4

```
SELECT co.customer_id,
    SUM(CASE WHEN pn.pizza_name = 'Vegetarian' THEN 1 ELSE 0 END) AS vegetarian_pizzas,
    SUM(CASE WHEN pn.pizza_name = 'Meatlovers' THEN 1 ELSE 0 END) AS meatlovers_pizzas
FROM customer_orders co
JOIN pizza_names pn ON co.pizza_id = pn.pizza_id
GROUP BY co.customer_id;
```



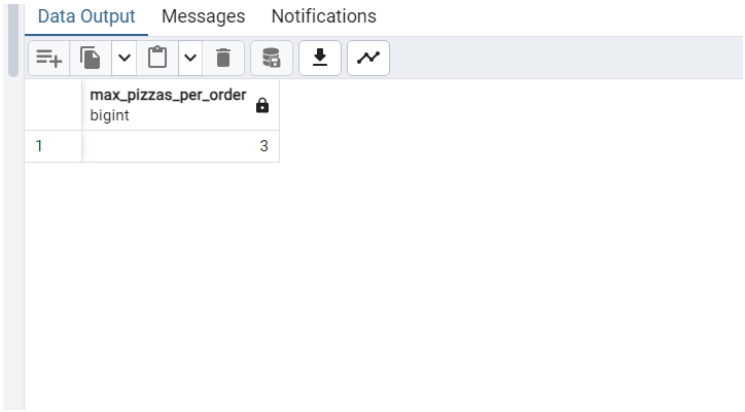
Data Output | Messages | Notifications

	customer_id integer	vegetarian_pizzas bigint	meatlovers_pizzas bigint
1	101	1	2
2	103	1	3
3	104	0	3
4	105	1	0
5	102	1	2

Câu 5

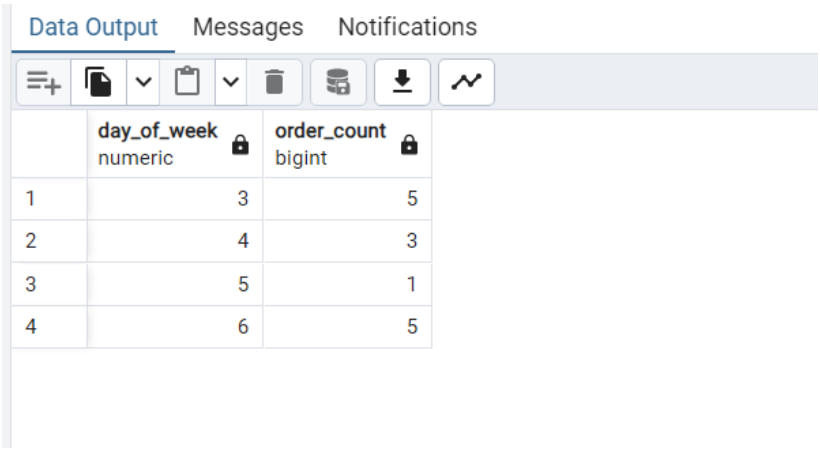
```
SELECT MAX(pizza_count) AS max_pizzas_per_order
FROM (
    SELECT order_id, COUNT(pizza_id) AS pizza_count
```

```
    FROM customer_orders
    GROUP BY order_id
) AS pizza_counts;
```



Câu 6

```
SELECT EXTRACT(DOW FROM order_time) AS day_of_week,
    COUNT(order_id) AS order_count
FROM customer_orders
GROUP BY day_of_week
ORDER BY day_of_week;
```



	day_of_week numeric	order_count bigint
1	3	5
2	4	3
3	5	1
4	6	5

Câu 7

```
SELECT EXTRACT(WEEK FROM registration_date) AS week_number,
    COUNT(runner_id) AS registered_runners
```

```
FROM runners
WHERE registration_date >= '2021-01-01'
GROUP BY week_number
ORDER BY week_number;
```

Data Output | Messages | Notifications

	week_number numeric	registered_runners bigint
1	1	1
2	2	1
3	53	2

Câu 8

```
SELECT runner_id, AVG(EXTRACT(EPOCH FROM CAST(pickup_time AS
TIMESTAMP)) / 60) AS average_pickup_time_minutes
FROM runner_orders
WHERE cancellation IS NULL OR cancellation = '' or cancellation like 'null'
AND pickup_time IS NOT NULL
GROUP BY runner_id;
```

Data Output | Messages | Notifications

	runner_id integer	average_pickup_time_minutes numeric
1	1	26302447.354166666667
2	2	26307112.944444444444
3	3	26308630.950000000000

Câu 9

```
SELECT co.customer_id,
    AVG(ro.distance) AS avg_distance
FROM customer_orders co
JOIN runner_orders ro ON co.order_id = ro.order_id
WHERE ro.distance IS NOT NULL
GROUP BY co.customer_id;
```

Data Output | Messages | Notifications

	customer_id integer	avg_distance numeric
1	101	20.0000000000000000
2	103	23.4000000000000000
3	104	10.0000000000000000
4	105	25.0000000000000000
5	102	16.7333333333333333

Câu 10

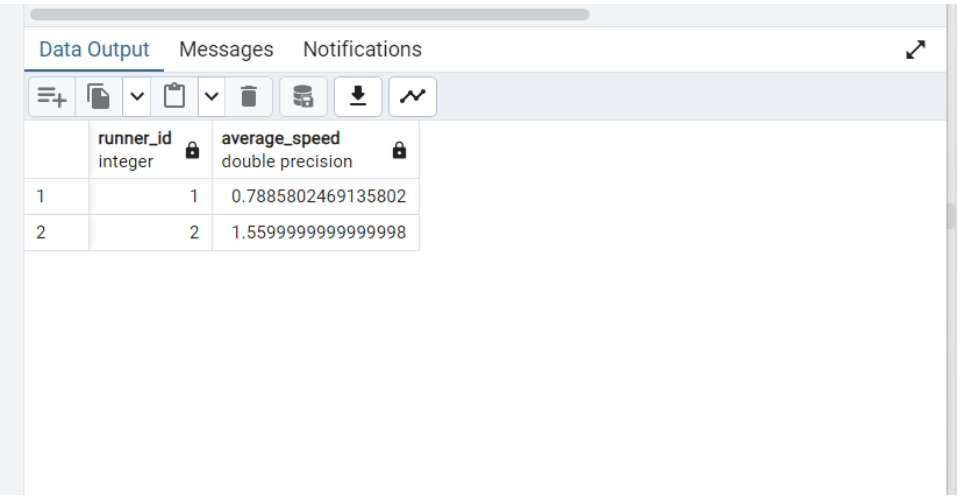
```
SELECT MAX(pickup_time::timestamp) - MIN(pickup_time::timestamp) AS delivery_time_difference
FROM runner_orders WHERE cancellation IS NULL OR cancellation = '' or cancellation like 'null'
AND pickup_time IS NOT NULL
```

Data Output | Messages | Notifications

	delivery_time_difference interval
1	10 days 00:34:46

câu 11

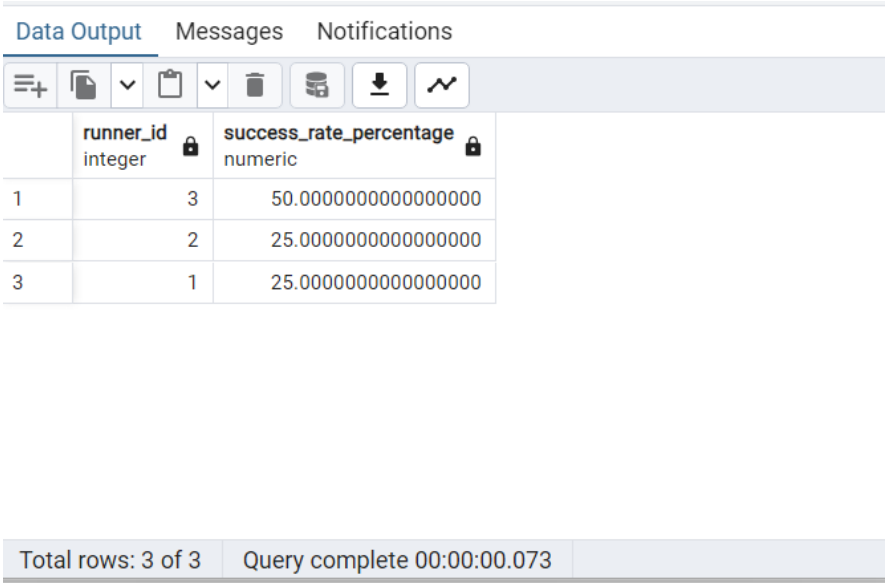
```
SELECT runner_id, AVG(CAST(NULLIF(SUBSTRING(distance, 1, GREATEST(LENGTH(distance) - 2, 0)), '') AS FLOAT) /

CAST(NULLIF(SUBSTRING(duration, 1, GREATEST(LENGTH(duration) - 7, 0)), '') AS FLOAT)) as

average_speed

FROM runner_orders

WHERE pickup_time IS NOT NULL AND LENGTH(duration) > 7 AND LENGTH(distance) > 2

GROUP BY runner_id
```



	runner_id integer	average_speed double precision
1	1	0.7885802469135802
2	2	1.5599999999999998

Câu 12

```
SELECT runner_id, COUNT(*) FILTER (WHERE cancellation IS NULL) * 100.0 / COUNT(*) AS success_rate_percentage

FROM runner_orders

 GROUP BY runner_id
```



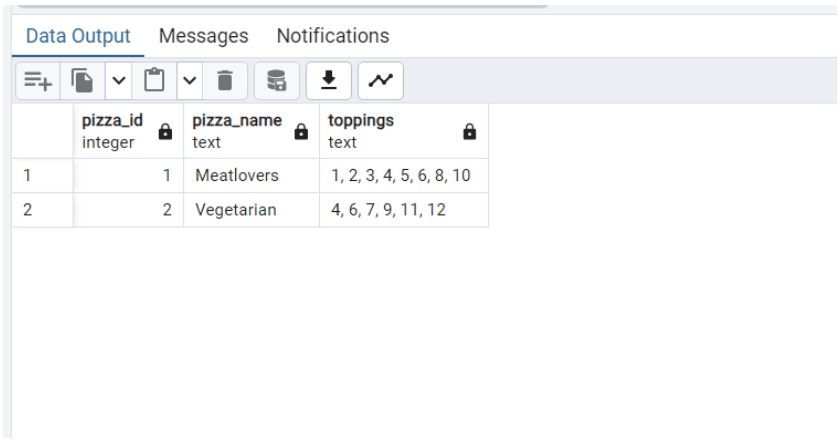
Data Output | Messages | Notifications

	runner_id integer	success_rate_percentage numeric
1	3	50.0000000000000000
2	2	25.0000000000000000
3	1	25.0000000000000000

Total rows: 3 of 3 | Query complete 00:00:00.073

Câu 13

```
SELECT pn.pizza_id, pn.pizza_name, pr.toppings
FROM pizza_names pn
JOIN pizza_recipes pr ON pn.pizza_id = pr.pizza_id;
```

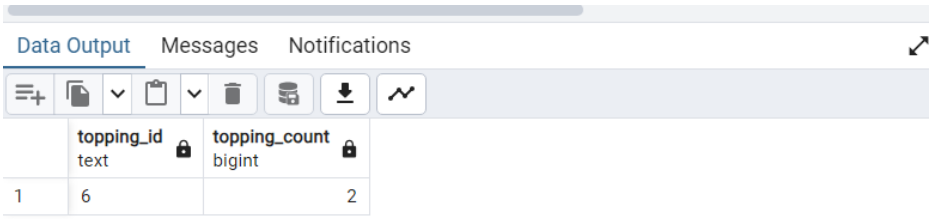


Data Output | Messages | Notifications

	pizza_id integer	pizza_name text	toppings text
1	1	Meatlovers	1, 2, 3, 4, 5, 6, 8, 10
2	2	Vegetarian	4, 6, 7, 9, 11, 12

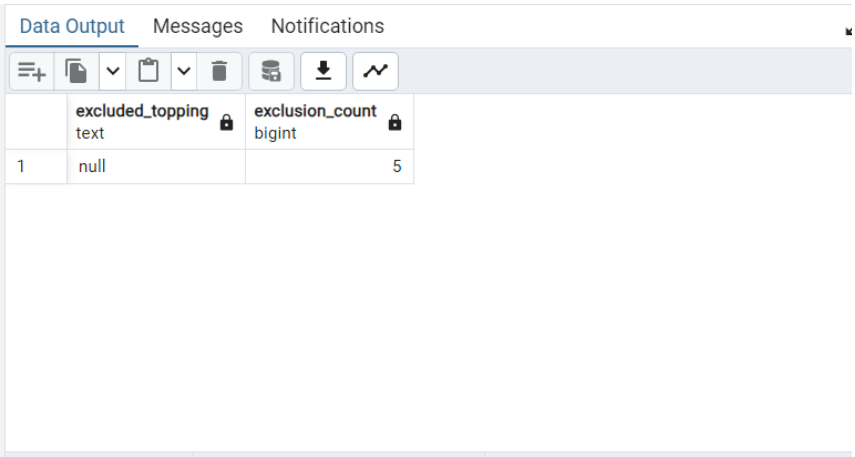
Câu 14

```
SELECT unnest(string_to_array(toppings, ', ')) AS topping_id, COUNT(*) AS topping_count
FROM pizza_recipes
GROUP BY topping_id
ORDER BY topping_count DESC
LIMIT 1;
```



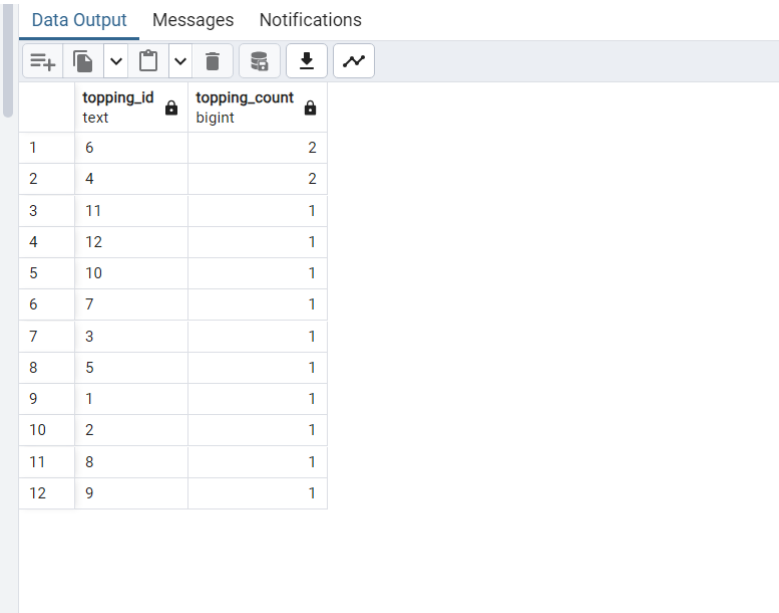
Câu 15

```
SELECT
TRIM(unnested_toppings) AS excluded_topping, COUNT(*) AS exclusion_count
FROM (
SELECT
UNNEST(string_to_array(exclusions, ',')) AS
unnested_toppings
FROM
customer_orders
WHERE
exclusions IS NOT NULL
) AS unnested_toppings
GROUP BY
excluded_topping
ORDER BY
exclusion_count DESC
LIMIT 1;
```



Câu 16

```
WITH all_toppings AS (
SELECT unnest(string_to_array(toppings, ', ')) AS
topping_id
FROM pizza_recipes
)
SELECT topping_id, COUNT(*) AS topping_count
FROM all_toppings
GROUP BY topping_id
ORDER BY topping_count DESC;
```

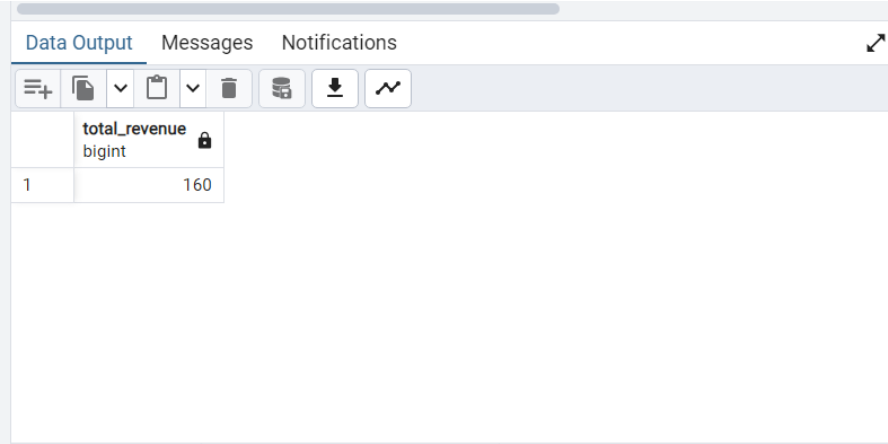


Data Output | Messages | Notifications

	topping_id text	topping_count bigint
1	6	2
2	4	2
3	11	1
4	12	1
5	10	1
6	7	1
7	3	1
8	5	1
9	1	1
10	2	1
11	8	1
12	9	1

Câu 17

```
SELECT
SUM(CASE WHEN pizza_name = 'Meatlovers' THEN 12
ELSE 10 END) AS total_revenue
FROM
customer_orders co
JOIN
pizza_names pn ON co.pizza_id = pn.pizza_id;
```

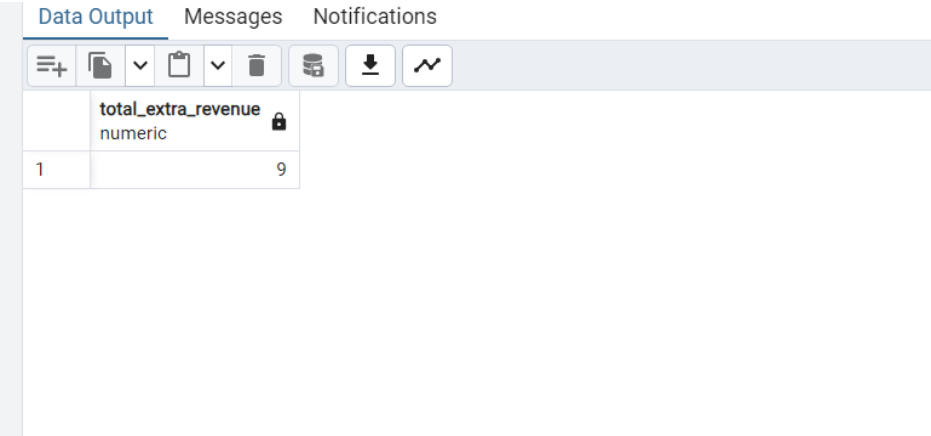


Data Output | Messages | Notifications

	total_revenue bigint
1	160

Câu 18

```
SELECT
SUM(
CASE
WHEN co.extras IS NOT NULL THEN
(SELECT COUNT(*) FROM
unnest(string_to_array(co.extras, ', '))) * 1
ELSE
0
END
) AS total_extra_revenue
FROM
customer_orders co;
```



Câu 19

```
WITH delivery_earnings AS (
SELECT
ro.runner_id, SUM(CAST(REPLACE(ro.distance, 'km', '') AS
DECIMAL) * 0.30) AS earnings
FROM
runner_orders ro
```

```
WHERE
ro.cancellation IS NULL
GROUP BY
ro.runner_id
),total_delivery_earnings AS (
SELECT
SUM(earnings) AS total_earnings
FROM
delivery_earnings
),total_order_cost AS (
SELECT
SUM(
CASE
WHEN co.pizza_id = 1 THEN 12
WHEN co.pizza_id = 2 THEN 10
ELSE 0
END
) AS total_cost
FROM
customer_orders co
)
SELECT
total_delivery_earnings.total_earnings - (
SELECT
COUNT(*) * 5
FROM
```

runner_orders

WHERE

cancellation IS NULL

) - total_order_cost.total_cost AS net_earnings

FROM

total_delivery_earnings, total_order_cost;