

CASE STUDY : PIZZA RUNNER

<https://8weeksqlchallenge.com/case-study-2/>

A. PIZZA METRICS

How many pizzas were ordered?

```
SELECT COUNT(*) AS nb_pizzas_ordered  
FROM customer_orders
```

nb_pizzas_ordered
14

How many unique customer orders were made?

```
SELECT COUNT(DISTINCT order_id) AS nb_orders  
FROM customer_orders  
=> nombre de commandes
```

nb_orders
10

How many successful orders were delivered by each runner?

```
SELECT runner_id, COUNT(*) AS successful_orders  
FROM runner_orders  
WHERE cancellation = ''  
GROUP BY runner_id
```

runner_id	successful_orders
1	4
2	3
3	1

How many of each type of pizza was delivered?

```
SELECT customer_orders.pizza_id, pizza_names.pizza_name, COUNT(*) AS nb_delivered_pizzas  
FROM customer_orders  
LEFT JOIN pizza_names ON customer_orders.pizza_id = pizza_names.pizza_id  
LEFT JOIN runner_orders ON customer_orders.order_id = runner_orders.order_id  
WHERE runner_orders.cancellation = ''  
GROUP BY customer_orders.pizza_id
```

pizza_id	pizza_name	nb_delivered_pizzas
1	Meatlovers	9
2	Vegetarian	3

How many Vegetarian and Meatlovers were ordered by each customer?

```
SELECT customer_orders.customer_id, pizza_names.pizza_name, COUNT(*) AS  
nb_ordered_pizzas  
FROM customer_orders LEFT JOIN pizza_names ON customer_orders.pizza_id =  
pizza_names.pizza_id  
GROUP BY customer_orders.customer_id, pizza_names.pizza_name
```

customer_id	pizza_name	nb_ordered_pizzas
101	Meatlovers	2
101	Vegetarian	1
102	Meatlovers	2
102	Vegetarian	1
103	Meatlovers	3

```
SELECT customer_id, SUM(CASE WHEN pizza_id = 1 THEN 1 ELSE 0 END) AS meat_lovers,  
SUM(CASE WHEN pizza_id = 2 THEN 1 ELSE 0 END) AS vegetarian  
FROM customer_orders  
GROUP BY customer_id
```

customer_id	meat_lovers	vegetarian
101	2	1
102	2	1
103	3	1
104	3	0
105	0	1

What was the maximum number of pizzas delivered in a single order?

```
SELECT customer_orders.order_id, COUNT(*) AS nb_delivered_pizzas  
FROM customer_orders  
LEFT JOIN runner_orders ON customer_orders.order_id = runner_orders.order_id  
WHERE runner_orders.cancellation = ''  
GROUP BY customer_orders.order_id  
ORDER BY nb_delivered_pizzas DESC LIMIT 1
```

order_id	nb_delivered_pizzas
4	3

For each customer, how many delivered pizzas had at least 1 change and how many had no changes?

```
SELECT customer_orders.customer_id, SUM(CASE WHEN customer_orders.exclusions <> ''  
THEN 1 ELSE 0 END) AS ingredients_changes, SUM(CASE WHEN customer_orders.exclusions =  
'' THEN 1 ELSE 0 END) AS ingredients_no_changes  
FROM customer_orders  
LEFT JOIN runner_orders ON customer_orders.order_id = runner_orders.order_id  
WHERE runner_orders.cancellation = ''  
GROUP BY customer_id
```

customer_id	ingredients_changes	ingredients_no_changes
101	0	2
102	0	3
103	3	0
104	1	2
105	0	1

How many pizzas were delivered that had both exclusions and extras?

```
SELECT COUNT(*) AS nb_delivered_pizzas_with_exclusions_and_extras
FROM customer_orders
LEFT JOIN runner_orders ON customer_orders.order_id = runner_orders.order_id
WHERE customer_orders.exclusions <> '' AND customer_orders.extras <> '' AND
runner_orders.cancellation = ''
```

nb_delivered_pizzas_with_exclusions_and_extras
1

What was the total volume of pizzas ordered for each hour of the day?

```
SELECT STRFTIME('%H', order_time) AS hour, COUNT(*) AS nb_ordered_pizzas
FROM customer_orders
GROUP BY hour
```

hour	nb_ordered_pizzas
11	1
13	3
18	3
19	1

What was the volume of orders for each day of the week?

```
SELECT
    (CASE CAST(STRFTIME('%w', order_time) AS integer)
     WHEN 1 THEN 'Monday'
     WHEN 2 THEN 'Tuesday'
     WHEN 3 THEN 'Wednesday'
     WHEN 4 THEN 'Thursday'
     WHEN 5 THEN 'Friday'
     WHEN 6 THEN 'Saturday'
     WHEN 0 THEN 'Sunday'
    END) AS day,
    COUNT(*) AS nb_ordered_pizzas
FROM customer_orders
GROUP BY day
```

day	nb_ordered_pizzas
Friday	1
Saturday	5
Thursday	3
Wednesday	5

B. RUNNER AND CUSTOMER EXPERIENCE

How many runners signed up for each 1 week period? (i.e. week starts 2021-01-01)

```
SELECT STRFTIME('%W', registration_date) AS week, COUNT(*) AS new_registrations
FROM runners
GROUP BY week
```

week	new_registrations
00	2
01	1
02	1

What was the average time in minutes it took for each runner to arrive at the Pizza Runner HQ to pickup the order?

```
SELECT DISTINCT runner_orders.order_id, runner_orders.runner_id,
time(customer_orders.order_time), time(runner_orders.pickup_time),
ROUND((JULIANDAY(runner_orders.pickup_time) - JULIANDAY(customer_orders.order_time)) * 24
* 60) AS pickup_duration
FROM runner_orders
JOIN customer_orders ON runner_orders.order_id = customer_orders.order_id
WHERE runner_orders.pickup_time <> ''
ORDER BY runner_id
```

order_id	runner_id	time(customer_orders...	time(runner_orders.pl...	pickup_duration
1	1	18:05:02	18:15:34	11
2	1	19:00:52	19:10:54	10
3	1	23:51:23	00:12:37	21
10	1	18:34:49	18:50:20	16
4	2	13:23:46	13:53:03	29
7	2	21:20:29	21:30:45	10
8	2	23:54:33	00:15:02	20
5	3	21:00:29	21:10:57	10

```
SELECT runner_orders.runner_id, ROUND(AVG((JULIANDAY(runner_orders.pickup_time) -
JULIANDAY(customer_orders.order_time)) * 24 * 60)) AS avg_pickup_duration
FROM runner_orders
JOIN customer_orders ON runner_orders.order_id = customer_orders.order_id
WHERE runner_orders.pickup_time <> ''
GROUP BY runner_orders.runner_id
```

runner_id	avg_pickup_duration
1	16
2	24
3	10

=> FAUX, fait la moyenne sur les doublons

```
SELECT runner_id, ROUND(AVG((JULIANDAY(pickup_time) - JULIANDAY(order_time)) * 24 * 60))
AS avg_pickup_duration
FROM (
    SELECT MAX(runner_orders.order_id) AS order_id, runner_orders.runner_id AS runner_id,
    customer_orders.order_time AS order_time, runner_orders.pickup_time AS pickup_time
```

```

FROM runner_orders
JOIN customer_orders ON runner_orders.order_id = customer_orders.order_id
WHERE runner_orders.pickup_time <> ''
GROUP BY runner_orders.order_id
GROUP BY runner_id

```

runner_id	avg_pickup_duration
1	14
2	20
3	10

Is there any relationship between the number of pizzas and how long the order takes to prepare?

```

SELECT nb_pizzas, ROUND(AVG(pickup_duration)) AS avg_preparation_time
FROM(
    SELECT customer_orders.order_id,
           COUNT(*) AS nb_pizzas,
           ROUND((JULIANDAY(runner_orders.pickup_time) -
JULIANDAY(customer_orders.order_time)) * 24 * 60) AS pickup_duration
    FROM customer_orders
    JOIN runner_orders ON customer_orders.order_id = runner_orders.order_id
    WHERE runner_orders.pickup_time <> ''
    GROUP BY customer_orders.order_id)
GROUP BY nb_pizzas

```

nb_pizzas	avg_preparation_time
1	12
2	19
3	29

What was the average distance travelled for each customer?

```

SELECT customer_orders.customer_id, ROUND(AVG(runner_orders.distance_km)) AS
avg_distance
FROM runner_orders
JOIN customer_orders ON runner_orders.order_id = customer_orders.order_id
WHERE runner_orders.pickup_time <> ''
GROUP BY customer_orders.customer_id

```

customer_id	avg_distance
101	20
102	17
103	23
104	10
105	25

What was the difference between the longest and shortest delivery times for all orders?

```

SELECT
(SELECT MAX(duration_min)
FROM runner_orders
WHERE pickup_time <> '')

```

```

-
(SELECT MIN(duration_min)
FROM runner_orders
WHERE pickup_time <> '')
AS diff_duration

```

diff_duration
30

What was the average speed for each runner for each delivery and do you notice any trend for these values?

```

SELECT runner_id, distance_km, duration_min, ROUND((distance_km / duration_min) * 60, 1) AS
speed
FROM runner_orders
WHERE cancellation = ''
ORDER BY runner_id

```

runner_id	distance_km	duration_min	speed
1	20	32	37.5
1	20	27	44.4
1	13.4	20	40.2
1	10	10	60
2	23.4	40	35.1
2	25	25	60

What is the successful delivery percentage for each runner?

```

SELECT runner_id, SUM(CASE WHEN cancellation = '' THEN 1 ELSE 0 END) AS
successful_delivery, COUNT(order_id) AS nb_total_orders, SUM(CASE WHEN cancellation = ''
THEN 1 ELSE 0 END) * 100 / COUNT(order_id) AS perc_delivery
FROM runner_orders
GROUP BY runner_id

```

runner_id	successful_delivery	nb_total_orders	perc_delivery
1	4	4	100
2	3	4	75
3	1	2	50

C. INGREDIENT OPTIMISATION

What are the standard ingredients for each pizza?

```
WITH RECURSIVE split(pizza_id, toppings, str) AS (  
  SELECT pizza_id, '', toppings||',' FROM pizza_recipes  
  UNION ALL SELECT  
    pizza_id,  
    substr(str, 0, instr(str, ',')),  
    substr(str, instr(str, ',')+1)  
  FROM split WHERE str!='  
)  
SELECT pizza_id, toppings  
FROM split  
WHERE toppings != ''  
ORDER BY pizza_id
```

=> pour splitter la liste des ingrédients

! pizza_id	toppings
1	1
1	2
1	3
1	4
1	5
1	6
1	8
1	10
2	4

```
WITH RECURSIVE split(pizza_id, toppings, str) AS (  
  SELECT pizza_id, '', toppings||',' FROM pizza_recipes  
  UNION ALL SELECT  
    pizza_id,  
    substr(str, 0, instr(str, ',')),  
    substr(str, instr(str, ',')+1)  
  FROM split WHERE str!='  
)  
SELECT pizza_names.pizza_name, pizza_toppings.topping_name  
FROM split  
JOIN pizza_names ON split.pizza_id = pizza_names.pizza_id  
JOIN pizza_toppings ON split.toppings = pizza_toppings.topping_id  
WHERE toppings != ''  
ORDER BY pizza_names.pizza_name
```

! pizza_name	topping_name
Meatlovers	Bacon
Meatlovers	BBQ Sauce
Meatlovers	Beef
Meatlovers	Cheese
Meatlovers	Chicken
Meatlovers	Mushrooms

What was the most commonly added extra?

```
WITH RECURSIVE split(order_id, extras, str) AS (  
  SELECT order_id, '', extras||',' FROM customer_orders  
  UNION ALL SELECT  
    order_id,  
    substr(str, 0, instr(str, ',')),  
    substr(str, instr(str, ',')+1)  
  FROM split WHERE str!='  
)  
SELECT pizza_toppings.topping_name, COUNT(*) AS nb_extras  
FROM split  
JOIN pizza_toppings ON split.extras = pizza_toppings.topping_id  
WHERE split.extras <> ''  
GROUP BY split.extras  
ORDER BY nb_extras DESC LIMIT 1
```

topping_name	nb_extras
Bacon	4

What was the most common exclusion?

```
WITH RECURSIVE split(order_id, exclusions, str) AS (  
  SELECT order_id, '', exclusions||',' FROM customer_orders  
  UNION ALL SELECT  
    order_id,  
    substr(str, 0, instr(str, ',')),  
    substr(str, instr(str, ',')+1)  
  FROM split WHERE str!='  
)  
SELECT pizza_toppings.topping_name, COUNT(*) AS nb_exclusions  
FROM split  
JOIN pizza_toppings ON split.exclusions = pizza_toppings.topping_id  
WHERE split.exclusions <> ''  
GROUP BY split.exclusions  
ORDER BY nb_exclusions DESC LIMIT 1
```

topping_name	nb_exclusions
Cheese	4

Generate an order item for each record in the customers_orders table in the format of one of the following:

- **Meat Lovers**
- **Meat Lovers - Exclude Beef**
- **Meat Lovers - Extra Bacon**
- **Meat Lovers - Exclude Cheese, Bacon - Extra Mushroom, Peppers**

```
SELECT customer_orders.order_id, customer_orders.pizza_id, pizza_names.pizza_name,  
customer_orders.exclusions, customer_orders.extras,  
CASE  
  WHEN customer_orders.pizza_id = 1 AND customer_orders.exclusions = '' AND  
customer_orders.extras = '' THEN 'Meat Lovers'  
  WHEN customer_orders.pizza_id = 2 AND customer_orders.exclusions = '' AND  
customer_orders.extras = '' THEN 'Vegetarian'  
  WHEN customer_orders.pizza_id = 1 AND customer_orders.exclusions = '4' AND  
customer_orders.extras = '' THEN 'Meat Lovers - Exclude Cheese'
```



```

    WHEN customer_orders.pizza_id = 2 AND customer_orders.exclusions = '4' AND
customer_orders.extras = '' THEN 'Vegetarian - Exclude Cheese'
    WHEN customer_orders.pizza_id = 1 AND customer_orders.exclusions = '' AND
customer_orders.extras = '1' THEN 'Meat Lovers - Extra Bacon'
    WHEN customer_orders.pizza_id = 2 AND customer_orders.exclusions = '' AND
customer_orders.extras = '1' THEN 'Vegetarian - Extra Bacon'
    WHEN customer_orders.pizza_id = 1 AND customer_orders.exclusions = '4' AND
customer_orders.extras = '1, 5' THEN 'Meat Lovers - Exclude Cheese - Extra Bacon and Chicken'
    WHEN customer_orders.pizza_id = 1 AND customer_orders.exclusions = '2, 6' AND
customer_orders.extras = '1, 4' THEN 'Meat Lovers - Exclude BBQ Sauce and Mushroom - Extra
Bacon and Cheese'
    END AS order_item
FROM customer_orders
JOIN pizza_names ON customer_orders.pizza_id = pizza_names.pizza_id

```

order_id	pizza_id	pizza_name	exclusions	extras	order_item
1	1	Meatlovers			Meat Lovers
2	1	Meatlovers			Meat Lovers
3	1	Meatlovers			Meat Lovers
3	2	Vegetarian			Vegetarian
4	1	Meatlovers	4		Meat Lovers - Exclude Cheese
4	1	Meatlovers	4		Meat Lovers - Exclude Cheese
4	2	Vegetarian	4		Vegetarian - Exclude Cheese

**Generate an alphabetically ordered comma separated ingredient list for each pizza order from the customer_orders table and add a 2x in front of any relevant ingredients
For example: "Meat Lovers: 2xBacon, Beef, ... , Salami"**

```

SELECT customer_orders.order_id, customer_orders.pizza_id, pizza_names.pizza_name,
customer_orders.exclusions, customer_orders.extras,
CASE
    WHEN customer_orders.pizza_id = 1 AND customer_orders.exclusions = '' AND
customer_orders.extras = '' THEN 'Bacon, BBQ Sauce, Beef, Cheese, Chicken, Mushrooms,
Pepperoni, Salami'
    WHEN customer_orders.pizza_id = 2 AND customer_orders.exclusions = '' AND
customer_orders.extras = '' THEN 'Cheese, Mushrooms, Onions, Peppers, Tomatoes, Tomato
Sauce'
    WHEN customer_orders.pizza_id = 1 AND customer_orders.exclusions = '4' AND
customer_orders.extras = '' THEN 'Bacon, BBQ Sauce, Beef, Chicken, Mushrooms, Pepperoni,
Salami'
    WHEN customer_orders.pizza_id = 2 AND customer_orders.exclusions = '4' AND
customer_orders.extras = '' THEN 'Mushrooms, Onions, Peppers, Tomatoes, Tomato Sauce'
    WHEN customer_orders.pizza_id = 1 AND customer_orders.exclusions = '' AND
customer_orders.extras = '1' THEN 'BaconX2, BBQ Sauce, Beef, Cheese, Chicken, Mushrooms,
Pepperoni, Salami'
    WHEN customer_orders.pizza_id = 2 AND customer_orders.exclusions = '' AND
customer_orders.extras = '1' THEN 'Bacon, Mushrooms, Onions, Peppers, Tomatoes, Tomato
Sauce'
    WHEN customer_orders.pizza_id = 1 AND customer_orders.exclusions = '4' AND
customer_orders.extras = '1, 5' THEN 'BaconX2, BBQ Sauce, Beef, ChickenX2, Mushrooms,
Pepperoni, Salami'
    WHEN customer_orders.pizza_id = 1 AND customer_orders.exclusions = '2, 6' AND
customer_orders.extras = '1, 4' THEN 'BaconX2, Beef, CheeseX2, Chicken, Pepperoni, Salami'
    END AS ingredients_list
FROM customer_orders
JOIN pizza_names ON customer_orders.pizza_id = pizza_names.pizza_id

```

order_id	pizza_id	piz...	exclusions	ext...	ingredients_list
1	1	Mea...			Bacon, BBQ Sauce, Beef, ...
2	1	Mea...			Bacon, BBQ Sauce, Beef, ...
3	1	Mea...			Bacon, BBQ Sauce, Beef, ...
3	2	Veg...			Cheese, Mushrooms, Onio...
4	1	Mea...	4		Bacon, BBQ Sauce, Beef, ...
4	1	Mea...	4		Bacon, BBQ Sauce, Beef, ...
4	2	Veg...	4		Mushrooms, Onions, Peppe..
5	1	Mea...		1	BaconX2, BBQ Sauce, Bee...

What is the total quantity of each ingredient used in all delivered pizzas sorted by most frequent first?

```

WITH RECURSIVE split(order_id, ingredients_list, str) AS (
  SELECT order_id, '', ingredients_list||',' FROM
  (
    SELECT customer_orders.order_id,
    CASE
      WHEN customer_orders.pizza_id = 1 THEN 'Bacon, BBQ Sauce, Beef, Cheese, Chicken,
Mushrooms, Pepperoni, Salami'
      WHEN customer_orders.pizza_id = 2 THEN 'Cheese, Mushrooms, Onions, Peppers, Tomatoes,
Tomato Sauce'
      WHEN customer_orders.pizza_id = 1 AND customer_orders.exclusions = '4' AND
customer_orders.extras = '' THEN 'Bacon, BBQ Sauce, Beef, Chicken, Mushrooms, Pepperoni,
Salami'
      WHEN customer_orders.pizza_id = 2 AND customer_orders.exclusions = '4' AND
customer_orders.extras = '' THEN 'Mushrooms, Onions, Peppers, Tomatoes, Tomato Sauce'
      WHEN customer_orders.pizza_id = 1 AND customer_orders.exclusions = '' AND
customer_orders.extras = '1' THEN 'BaconX2, BBQ Sauce, Beef, Cheese, Chicken, Mushrooms,
Pepperoni, Salami'
      WHEN customer_orders.pizza_id = 2 AND customer_orders.exclusions = '' AND
customer_orders.extras = '1' THEN 'Bacon, Mushrooms, Onions, Peppers, Tomatoes, Tomato
Sauce'
      WHEN customer_orders.pizza_id = 1 AND customer_orders.exclusions = '4' AND
customer_orders.extras LIKE '1, 5' THEN 'BaconX2, BBQ Sauce, Beef, ChickenX2, Mushrooms,
Pepperoni, Salami'
      WHEN customer_orders.pizza_id = 1 AND customer_orders.exclusions LIKE '2, 6' AND
customer_orders.extras LIKE '1, 4' THEN 'BaconX2, Beef, CheeseX2, Chicken, Pepperoni,
Salami'
    END AS ingredients_list
    FROM customer_orders
    JOIN pizza_names ON customer_orders.pizza_id = pizza_names.pizza_id
  )
  UNION ALL SELECT
    order_id,
    substr(str, 0, instr(str, ',')),
    substr(str, instr(str, ',')+1)
  FROM split WHERE str!='
)

SELECT order_id, ingredients_list
FROM split
ORDER BY order_id

```

D. PRICING

If a Meat Lovers pizza costs \$12 and Vegetarian costs \$10 and there were no charges for changes, how much money has Pizza Runner made so far if there are no delivery fees?

```
SELECT
  SUM(CASE WHEN pizza_names.pizza_name = 'Meatlovers' THEN 1 ELSE 0 END) * 12 AS
  meatlovers,
  SUM(CASE WHEN pizza_names.pizza_name = 'Vegetarian' THEN 1 ELSE 0 END) * 10 AS
  vegetarian,
  (SUM(CASE WHEN pizza_names.pizza_name = 'Meatlovers' THEN 1 ELSE 0 END) * 12) +
  (SUM(CASE WHEN pizza_names.pizza_name = 'Vegetarian' THEN 1 ELSE 0 END) * 10) AS ca
FROM runner_orders
JOIN customer_orders ON runner_orders.order_id = customer_orders.order_id
JOIN pizza_names ON customer_orders.pizza_id = pizza_names.pizza_id
WHERE runner_orders.cancellation = ''
```

meatlovers	vegetarian	ca
108	30	138

What if there was an additional \$1 charge for any pizza extras?

- Add cheese is \$1 extra

If a Meat Lovers pizza was \$12 and Vegetarian \$10 fixed prices with no cost for extras and each runner is paid \$0.30 per kilometre traveled - how much money does Pizza Runner have left over after these deliveries?