Mini Project-The great pizza analytics challenge

Participants will apply the SQL concepts learned up to Day 15, including database creation, filtering, data cleaning, and joins, to analyze sales performance and customer trends. The project emphasises practical application of SQL in real-world analytics scenarios and helps strengthen query-building and data interpretation skills.

In this mini project, participants will assume the role of a Data Analyst at The Great Pizza, responsible for analyzing pizza sales data to uncover meaningful business insights using SQL. The challenge involves solving a structured set of questions across multiple phases; from exploring and filtering data to performing aggregations and joins.

-- 2.List all unique pizza categories (DISTINCT).

SELECT DISTINCT category FROM pizza_types;

	category
▶	Chicken
	Classic
	Supreme
	Veggie

-- 3. Display pizza_type_id, name, and ingredients, replacing NULL ingredients with "Missing Data". Show first 5 rows.

SELECT pizza_type_id, name,

COALESCE(ingredients, 'Missing Data') AS ingredients

FROM pizza_types

LIMIT 5;

	pizza_type_id	name	ingredients
▶	bbq_ckn	The Barbecue Chicken Pizza	Barbecued Chicken, Red Peppers, Green Peppe...
	big_meat	The Big Meat Pizza	Bacon, Pepperoni, Italian Sausage, Chorizo Sau...
	brie_carre	The Brie Carre Pizza	Brie Carre Cheese, Prosciutto, Caramelized Oni...
	calabrese	The Calabrese Pizza	ëNduja Salami, Pancetta, Tomatoes, Red Onion...
	cali_ckn	The California Chicken Pizza	Chicken, Artichoke, Spinach, Garlic, Jalapeno P...

-- 4. Check for pizzas missing a price (IS NULL).

SELECT * FROM pizzas

WHERE price IS NULL ;

	pizza_id	pizza_type_id	size	price
*	NULL	NULL	NULL	NULL

Phase 2: Filtering & Exploration

-- 5.Orders placed on '2015-01-01' (SELECT + WHERE).

SELECT * FROM orders
WHERE date = '2015-01-01';

order_id	date	time
1	2015-01-01	11:38:36
2	2015-01-01	11:57:40
3	2015-01-01	12:12:28
4	2015-01-01	12:16:31
5	2015-01-01	12:21:30
6	2015-01-01	12:29:36
7	2015-01-01	12:50:37
8	2015-01-01	12:51:37
9	2015-01-01	12:52:01
10	2015-01-01	13:00:15
11	2015-01-01	13:02:59
12	2015-01-01	13:04:41
13	2015-01-01	13:11:55
14	2015-01-01	13:14:19
15	2015-01-01	13:33:00
16	2015-01-01	13:34:07

6. List pizzas with price descending.

SELECT * FROM pizzas

ORDER BY price DESC;

pizza_id	pizza_type_id	size	price
the_greek_xxl	the_greek	XXL	35.95
the_greek_xl	the_greek	XL	25.50
brie_carre_s	brie_carre	S	23.65
ital_veggie_l	ital_veggie	L	21.00
bbq_ckn_l	bbq_ckn	L	20.75
soppressata_l	soppressata	L	20.75
southw_ckn_l	southw_ckn	L	20.75
spicy_ital_l	spicy_ital	L	20.75
peppr_salami_l	peppr_salami	L	20.75
spin_pesto_l	spin_pesto	L	20.75
thai_ckn_l	thai_ckn	L	20.75
ckn_pesto_l	ckn_pesto	L	20.75
spinach_supr_l	spinach_supr	L	20.75
cali_ckn_l	cali_ckn	L	20.75
prsc_argla_l	prsc_argla	L	20.75

7. Pizzas sold in sizes 'L' or 'XL'.

SELECT * FROM pizzas

WHERE size = 'L' OR size='XL';

pizza_id	pizza_type_id	size	price
spicy_ital_l	spicy_ital	L	20.75
spin_pesto_l	spin_pesto	L	20.75
spinach_fet_l	spinach_fet	L	20.25
spinach_supr_l	spinach_supr	L	20.75
thai_ckn_l	thai_ckn	L	20.75
the_greek_l	the_greek	L	20.50
the_greek_xl	the_greek	XL	25.50
veggie_veg_l	veggie_veg	L	20.25

8. Pizzas priced between $15.00 and $17.00.

SELECT * FROM pizzas

WHERE price BETWEEN 15 AND 17;

pizza_id	pizza_type_id	size	price
▶ bbq_ckn_m	bbq_ckn	M	16.75
big_meat_m	big_meat	M	16.00
calabrese_m	calabrese	M	16.25
cali_ckn_m	cali_ckn	M	16.75
ckn_alfredo_m	ckn_alfredo	M	16.75
ckn_pesto_m	ckn_pesto	M	16.75
classic_dlx_m	classic_dlx	M	16.00
five_cheese_m	five_cheese	M	15.50

9. Pizzas with "Chicken" in the name.

SELECT * FROM pizza_types

WHERE name LIKE '%chicken%';

pizza_type_id	name	category	ingredients
bbq_ckn	The Barbecue Chicken Pizza	Chicken	Barbecued Chicken, Red Peppers, Green Peppe…
cali_ckn	The California Chicken Pizza	Chicken	Chicken, Artichoke, Spinach, Garlic, Jalapeno P…
ckn_alfredo	The Chicken Alfredo Pizza	Chicken	Chicken, Red Onions, Red Peppers, Mushrooms…
ckn_pesto	The Chicken Pesto Pizza	Chicken	Chicken, Tomatoes, Red Peppers, Spinach, Garl…
southw_ckn	The Southwest Chicken Pizza	Chicken	Chicken, Tomatoes, Red Peppers, Red Onions, …
thai_ckn	The Thai Chicken Pizza	Chicken	Chicken, Pineapple, Tomatoes, Red Peppers, T…
NULL	NULL	NULL	NULL

10. Orders on '2015-02-15' or placed after 8 PM.

SELECT * FROM orders

WHERE date = '2015-02-15' OR time> '20:00:00' ;

order_id	date	time
▶ 60	2015-01-01	20:05:16
61	2015-01-01	20:08:43
62	2015-01-01	20:50:16
63	2015-01-01	2015-01-01
64	2015-01-01	20:52:08
65	2015-01-01	21:16:00
66	2015-01-01	21:47:55
67	2015-01-01	22:03:40

Phase 3: Sales Performance

11. Total quantity of pizzas sold (SUM).

 SELECT SUM(quantity) AS total_pizzas_sold

 FROM order_details;

total_pizzas_sold
49574

12. Average pizza price (AVG).

SELECT ROUND(AVG(price),2) AS avg_pizza_price

 FROM pizzas;

avg_pizza_price
16.44

13. Total order value per order (JOIN, SUM, GROUP BY).

 SELECT order_id, SUM(price*quantity) AS total_order_value

 FROM order_details od JOIN pizzas p

 ON od.pizza_id= p.pizza_id

 GROUP BY order_id

 ORDER BY total_order_value DESC;

order_id	total_order_value
18845	444.20
10760	417.15
1096	285.15
6169	284.00
740	280.95
12257	276.75
1685	272.75
4482	267.20

14. Total quantity sold per pizza category (JOIN, GROUP BY).

        SELECT category, SUM(quantity) AS total_qty_sold

        FROM pizzas p JOIN order_details od

        ON p.pizza_id= od.pizza_id

        JOIN pizza_types pt

        ON p.pizza_type_id= pt.pizza_type_id

        GROUP BY category;

category	total_qty_sold
Classic	14888
Veggie	11649
Supreme	11987
Chicken	11050

15. Categories with more than 5,000 pizzas sold (HAVING).

        SELECT category, SUM(quantity) AS total_qty_sold

        FROM pizzas p JOIN order_details od

        ON p.pizza_id= od.pizza_id

JOIN pizza_types pt

ON p.pizza_type_id= pt.pizza_type_id

GROUP BY category

HAVING total_qty_sold> 5000;

category	total_qty_sold
Classic	14888
Veggie	11649
Supreme	11987
Chicken	11050

16. Pizzas never ordered (LEFT/RIGHT JOIN).

SELECT p.pizza_id AS pizzas_never_ordered

FROM pizzas p

LEFT JOIN order_details od

ON p.pizza_id = od.pizza_id

WHERE od.pizza_id IS NULL;

pizzas_never_ordered
big_meat_l
four_cheese_s
five_cheese_m
big_meat_m
five_cheese_s

17. Price differences between different sizes of the same pizza (SELF JOIN).

SELECT

   p1.pizza_type_id,

   p1.size AS size_1,

   p1.price AS price_1,

   p2.size AS size_2,

   p2.price AS price_2,

   (p2.price - p1.price) AS price_difference

FROM pizzas p1

JOIN pizzas p2

   ON p1.pizza_type_id = p2.pizza_type_id

   AND CASE p1.size

        WHEN 'S' THEN 1

```
        WHEN 'M' THEN 2

        WHEN 'L' THEN 3

        WHEN 'XL' THEN 4

        WHEN 'XXL' THEN 5

    END

   <

    CASE p2.size

        WHEN 'S' THEN 1

        WHEN 'M' THEN 2

        WHEN 'L' THEN 3

        WHEN 'XL' THEN 4

        WHEN 'XXL' THEN 5

    END

ORDER BY p1.pizza_type_id, size_1;
```

pizza_type_id	size_1	price_1	size_2	price_2	price_difference
bbq_ckn	M	16.75	L	20.75	4.00
bbq_ckn	S	12.75	L	20.75	8.00
bbq_ckn	S	12.75	M	16.75	4.00
big_meat	M	16.00	L	20.50	4.50
big_meat	S	12.00	L	20.50	8.50
big_meat	S	12.00	M	16.00	4.00
calabrese	M	16.25	L	20.25	4.00
calabrese	S	12.25	L	20.25	8.00