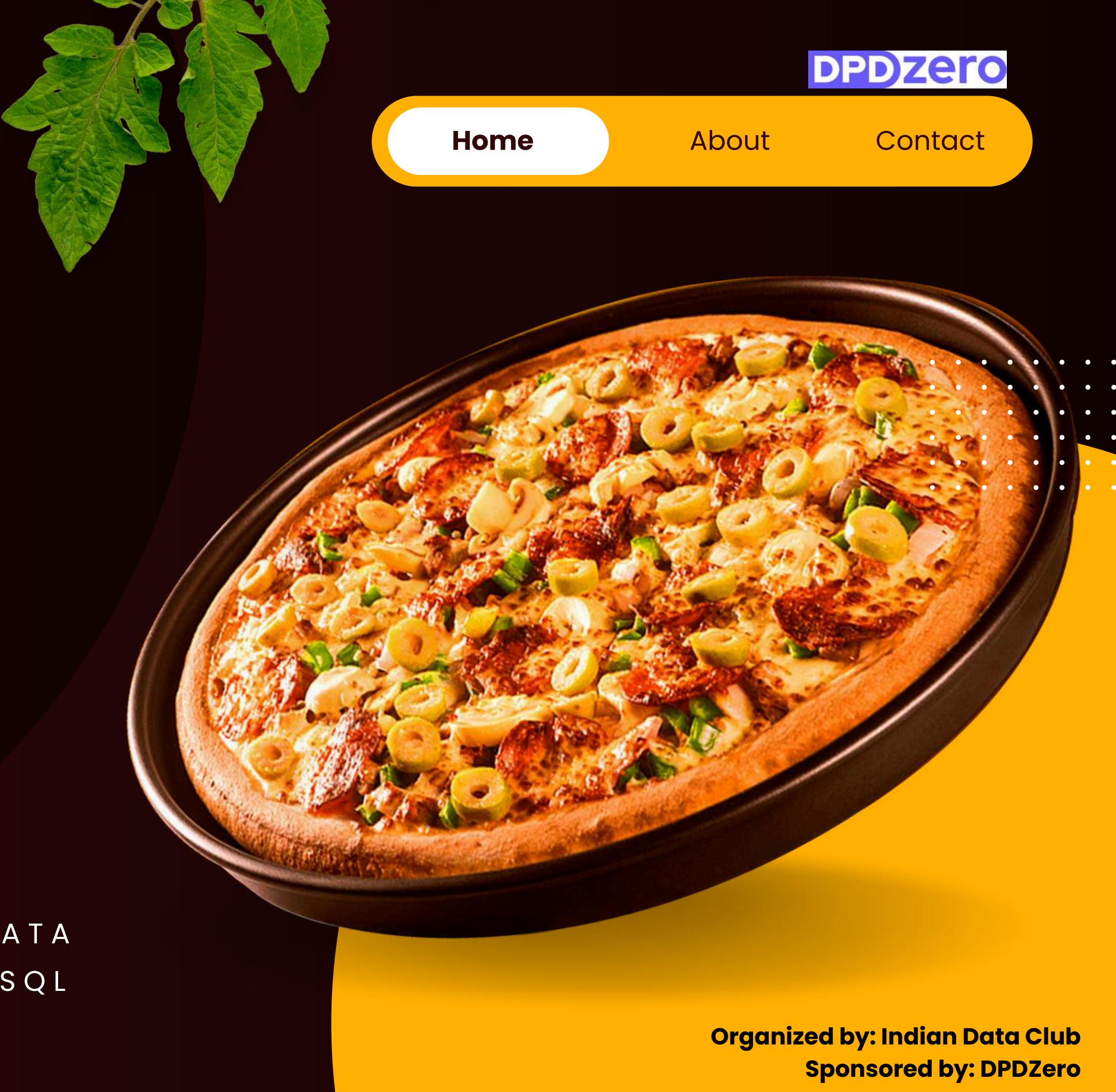




THE GREAT PIZZA ANALYTICS CHALLENGE

TRANSFORMING RAW PIZZA SALES DATA
INTO ACTIONABLE INSIGHTS USING SQL





Home

About

Contact

ABOUT THE ANALYST

SONALI GUPTA

Skills: SQL • Power BI • Advanced Excel • Microsoft Power BI
• Python • ETL • Google Sheets • Looker Studio

Passionate about turning messy datasets into meaningful business insights.

This challenge is part of my #21DaysOfSQL practice journey



[Home](#)[About](#)[Contact](#)

PROJECT OVERVIEW

A HANDS-ON SQL ANALYTICS MINI-PROJECT WHERE, AS THE DATA ANALYST FOR IDC PIZZA, MY MISSION IS TO DIVE INTO THE COMPANY'S SALES DATA AND UNCOVER MEANINGFUL TRENDS, PATTERNS, AND BUSINESS INSIGHTS.

GOALS:

- Database creation & table design
- Filtering, pattern matching & operators
- Joins (inner, left, right, full, self)
- Aggregations & data summaries
- Data cleaning & NULL handling



[Home](#)[About](#)[Contact](#)

ABOUT THE IDC PIZZA DATASET

THE PROJECT USES 4 RELATIONAL
TABLES STRUCTURED AS:

- **pizza_types** – pizza names, categories, ingredients
- **pizzas** – size, type, and price
- **orders** – order timestamps
- **order_details** – quantities per order item

Together, these tables build a complete picture of IDC Pizza's operations.

[Home](#)[About](#)[Contact](#)

QUESTIONS COVERED (TOTAL 25)

PHASE 1: FOUNDATION

1) INSTALL IDC_PIZZA.DUMP AS IDC_PIZZA SERVER

IDC_Pizza.dump is a PostgreSQL backup file. It cannot be imported into MySQL;
it must be restored in PostgreSQL/pgAdmin.



2) LIST ALL UNIQUE PIZZA CATEGORIES (DISTINCT)

QUERY

```
SELECT DISTINCT  
    category  
FROM  
    pizza_types;
```

OUTPUT

	category
▶	Chicken
	Classic
	Supreme
	Veggie



3) DISPLAY PIZZA TYPE_ID, NAME AND INGREDIENTS, REPLACING NULL INGREDIENTS WITH "MISSING DATA". SHOW FIRST 5 ROWS.

QUERY

```
SELECT
    pizza_type_id,
    name,
    COALESCE(ingredients, 'Missing Data') AS ingredients
FROM
    pizza_types
LIMIT 5;
```

OUTPUT

	pizza_type_id	name	ingredients
▶	bba_ckn	The Barbecue Chicken Pizza	Barbecued Chicken, Red Peppers, Green Pepp...
	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	�nduja 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)

Q U E R Y

```
SELECT
*
FROM
pizzas
WHERE
price IS NULL;
```

O U T P U T

	pizza_id	pizza_type_id	size	price
*	NULL	NULL	NULL	NULL



PHASE 2: FILTERING & EXPLORATION

1) ORDERS PLACED ON '2015-01-01' (SELECT + WHERE)

QUERY

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

OUTPUT

	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
	17	2015-01-01	13:53:00
	18	2015-01-01	13:57:08
	19	2015-01-01	13:59:09
	20	2015-01-01	14:03:08

2) LIST PIZZAS WITH PRICE DESCENDING

QUERY

```
SELECT
    pt.name,
    p.price
FROM
    pizzas AS p
JOIN
    pizza_types AS pt ON p.pizza_type_id = pt.pizza_type_id
ORDER BY
    p.price DESC;
```

OUTPUT

	name	price
▶	The Greek Pizza	35.95
	The Greek Pizza	25.50
	The Brie Carre Pizza	23.65
	The Italian Vegetables Pizza	21.00
	The Barbecue Chicken Pizza	20.75
	The Soppressata Pizza	20.75
	The Southwest Chicken Pizza	20.75
	The Spicy Italian Pizza	20.75
	The Pepper Salami Pizza	20.75
	The Spinach Pesto Pizza	20.75
	The Thai Chicken Pizza	20.75
	The Chicken Pesto Pizza	20.75
	The Spinach Supreme Pizza	20.75
	The California Chicken Pizza	20.75
	The Prosciutto and Arugula Pizza	20.75
	The Italian Supreme Pizza	20.75
	The Chicken Alfredo Pizza	20.75
	The Napolitana Pizza	20.50
	The Classic Deluxe Pizza	20.50
	The Big Meat Pizza	20.50

3) PIZZAS SOLD IN SIZES 'L' OR 'XL'

QUERY

```
SELECT
    pt.name,
    p.size
FROM
    pizzas AS p
JOIN
    pizza_types AS pt ON
    p.pizza_type_id = pt.pizza_type_id
WHERE
    p.size IN ('L', 'XL');
```

OUTPUT

	name	size
▶	The Barbecue Chicken Pizza	L
	The Big Meat Pizza	L
	The Calabrese Pizza	L
	The California Chicken Pizza	L
	The Chicken Alfredo Pizza	L
	The Chicken Pesto Pizza	L
	The Classic Deluxe Pizza	L
	The Five Cheese Pizza	L
	The Four Cheese Pizza	L
	The Green Garden Pizza	L
	The Hawaiian Pizza	L
	The Italian Capocollo Pizza	L
	The Italian Supreme Pizza	L
	The Italian Vegetables Pizza	L
	The Mediterranean Pizza	L
	The Mexicana Pizza	L
	The Napolitana Pizza	L
	The Pepperoni, Mushroom, and Peppers Pizza	L
	The Pepperoni Pizza	L
	The Pepper Salami Pizza	L

4) PIZZAS PRICED BETWEEN \$15.00 AND \$17.00.

QUERY

```
SELECT
    pt.name,
    p.price
FROM
    pizzas AS p
JOIN
    pizza_types AS pt
ON p.pizza_type_id = pt.pizza_type_id
WHERE
    p.price BETWEEN 15.00 AND 17.00
ORDER BY
    p.price ASC;
```

OUTPUT

name	price
The Pepperoni Pizza	15.25
The Five Cheese Pizza	15.50
The Big Meat Pizza	16.00
The Classic Deluxe Pizza	16.00
The Green Garden Pizza	16.00
The Italian Capocollo Pizza	16.00
The Mediterranean Pizza	16.00
The Mexicana Pizza	16.00
The Napolitana Pizza	16.00
The Spinach and Feta Pizza	16.00
The Greek Pizza	16.00
The Vegetables + Vegetables Pizza	16.00
The Calabrese Pizza	16.25
The Sicilian Pizza	16.25
The Hawaiian Pizza	16.50
The Italian Supreme Pizza	16.50
The Pepper Salami Pizza	16.50
The Prosciutto and Arugula Pizza	16.50
The Soppressata Pizza	16.50
The Spicy Italian Pizza	16.50



5) PIZZAS WITH "CHICKEN" IN THE NAME

QUERY

```
SELECT  
    name  
FROM  
    pizza_types  
WHERE  
    name LIKE '%Chicken%';
```

OUTPUT

	name
▶	The Barbecue Chicken Pizza
▶	The California Chicken Pizza
▶	The Chicken Alfredo Pizza
▶	The Chicken Pesto Pizza
▶	The Southwest Chicken Pizza
▶	The Thai Chicken Pizza



6) ORDERS ON '2015-02-15' OR PLACED AFTER 8 PM

QUERY

SELECT

*

FROM

orders AS o

JOIN

order_details AS od

ON o.order_id = od.order_id

WHERE

date = '2015-02-15'

OR o.time > '20:00:00'

ORDER BY

time ASC;

OUTPUT

	order_id	date	time	order_details_id	order_id	pizza_id	quantity
▶	2723	2015-02-15	11:35:24	6149	2723	pepperoni_l	1
	2724	2015-02-15	11:48:59	6150	2724	cali_ckn_s	1
	2724	2015-02-15	11:48:59	6151	2724	classic_dlx_m	1
	2724	2015-02-15	11:48:59	6152	2724	veggie_veg_l	1
	2725	2015-02-15	11:52:00	6153	2725	bbq_ckn_l	1
	2725	2015-02-15	11:52:00	6154	2725	calabrese_m	1
	2725	2015-02-15	11:52:00	6155	2725	ckn_alfredo_s	1
	2725	2015-02-15	11:52:00	6156	2725	hawaiian_l	1
	2725	2015-02-15	11:52:00	6157	2725	ital_supr_l	1
	2725	2015-02-15	11:52:00	6158	2725	napolitana_m	1
	2725	2015-02-15	11:52:00	6159	2725	peppr_salami...	1
	2725	2015-02-15	11:52:00	6160	2725	prsc_argla_m	1
	2725	2015-02-15	11:52:00	6161	2725	sicilian_m	1
	2725	2015-02-15	11:52:00	6162	2725	southw_ckn_l	1
	2725	2015-02-15	11:52:00	6163	2725	spicy_ital_l	1
	2725	2015-02-15	11:52:00	6164	2725	spin pesto_m	1
	2725	2015-02-15	11:52:00	6165	2725	the_greek_m	1
	2726	2015-02-15	11:54:55	6166	2726	big_meat_s	1
	2727	2015-02-15	11:56:33	6167	2727	ital_cpclo_m	1
	2727	2015-02-15	11:56:33	6168	2727	prsc argla_l	1

PHASE 3: SALES PERFORMANCE

1) TOTAL QUANTITY OF PIZZAS SOLD (SUM)

Q U E R Y

```
SELECT  
    SUM(quantity) AS pizza_quantity  
FROM  
    order_details;
```

O U T P U T

	pizza_quantity
▶	49574



2) AVERAGE PIZZA PRICE (AVG)

QUERY

```
SELECT  
    ROUND(AVG(price), 2) AS avg_price  
FROM  
    pizzas;
```

OUTPUT

	avg_price
▶	16.44



3) TOTAL ORDER VALUE PER ORDER (JOIN, SUM, GROUP BY)

QUERY

```
SELECT  
    o.order_id,  
    SUM(p.price * od.quantity) AS total_value  
FROM  
    orders AS o  
JOIN  
    order_details AS od ON o.order_id = od.order_id  
JOIN  
    pizzas AS p ON od.pizza_id = p.pizza_id  
GROUP BY  
    o.order_id;
```

OUTPUT

	order_id	total_value
▶	1	13.25
	2	92.00
	3	37.25
	4	16.50
	5	16.50
	6	24.75
	7	12.50
	8	12.50
	9	143.25
	10	41.00
	11	73.50
	12	70.75
	13	20.25
	14	12.00
	15	63.25
	16	50.70
	17	184.50
	18	20.50
	19	40.75
	20	30.50

4) TOTAL QUANTITY SOLD PER PIZZA CATEGORY (JOIN, GROUP BY)

Q U E R Y

```
SELECT  
    pt.category,  
    SUM(od.quantity) AS total_quantity  
FROM  
    pizza_types AS pt  
JOIN  
    pizzas AS p ON pt.pizza_type_id = p.pizza_type_id  
JOIN  
    order_details AS od ON p.pizza_id = od.pizza_id  
GROUP BY  
    pt.category;
```

O U T P U T

	category	total_quantity
▶	Classic	14888
	Veggie	11649
	Supreme	11987
	Chicken	11050



5) CATEGORIES WITH MORE THAN 5,000 PIZZAS SOLD (HAVING)

Q U E R Y

```
SELECT
    pt.category,
    SUM(od.quantity) AS total_quantity
FROM
    pizza_types AS pt
JOIN
    pizzas AS p ON pt.pizza_type_id = p.pizza_type_id
JOIN
    order_details AS od ON p.pizza_id = od.pizza_id
GROUP BY
    pt.category
HAVING
    SUM(od.quantity) > 5000;
```

O U T P U T

	category	total_quantity
▶	Classic	14888
	Veggie	11649
	Supreme	11987
	Chicken	11050



6) PIZZAS NEVER ORDERED (LEFT/RIGHT JOIN)

Q U E R Y

SELECT

*

FROM

pizzas AS p

LEFT JOIN

pizza_types AS pt

ON p.pizza_type_id = pt.pizza_type_id

LEFT JOIN

order_details AS od

ON p.pizza_id = od.pizza_id

WHERE

od.order_id IS NULL;

O U T P U T

	pizza_id	pizza_type_id	size	price	pizza_type_id	name	category	ingredients	order_detail	order_id	pizza_id	quantity
▶	big_meat_l	big_meat	L	20.50	big_meat	The Big Meat Pizza	Classic	Bacon, Pepperoni, Italian Sausage, Chorizo...	NULL	NULL	NULL	NULL
	big_meat_m	big_meat	M	16.00	big_meat	The Big Meat Pizza	Classic	Bacon, Pepperoni, Italian Sausage, Chorizo...	NULL	NULL	NULL	NULL
	five_cheese_m	five_cheese	M	15.50	five_cheese	The Five Cheese Pizza	Veggie	Mozzarella Cheese, Provolone Cheese, Sm...	NULL	NULL	NULL	NULL
	five_cheese_s	five_cheese	S	12.50	five_cheese	The Five Cheese Pizza	Veggie	Mozzarella Cheese, Provolone Cheese, Sm...	NULL	NULL	NULL	NULL
	four_cheese_s	four_cheese	S	11.75	four_cheese	The Four Cheese Pizza	Veggie	Ricotta Cheese, Gorgonzola Picante Chee...	NULL	NULL	NULL	NULL



7) PRICE DIFFERENCES BETWEEN DIFFERENT SIZES OF THE SAME PIZZA (SELF JOIN)

QUERY

```
SELECT
    p1.pizza_type_id,
    (MAX(p2.price) - MIN(p1.price))
AS price_difference
FROM
    pizzas AS p1
INNER JOIN pizzas AS p2
ON p1.pizza_type_id = p2.pizza_type_id
    AND p1.size <> p2.size
GROUP BY
    p1.pizza_type_id;
```

OUTPUT

	pizza_type_id	price_difference
▶	bbq_dkn	8.00
	big_meat	8.50
	calabrese	8.00
	cali_dkn	8.00
	ckn_alfredo	8.00
	ckn_pesto	8.00
	classic_dlx	8.50
	five_cheese	6.00
	four_cheese	6.20
	green_garden	8.25
	hawaiian	6.00
	ital_cpdlo	8.50
	ital_supr	8.25
	ital_veggie	8.25
	mediterraneo	8.25
	mexicana	8.25
	napolitana	8.50
	pep_msh_pep	6.50
	pepperoni	5.50
	peppr salami	8.25

[Home](#)[About](#)[Contact](#)

ADDITIONAL INSIGHTS

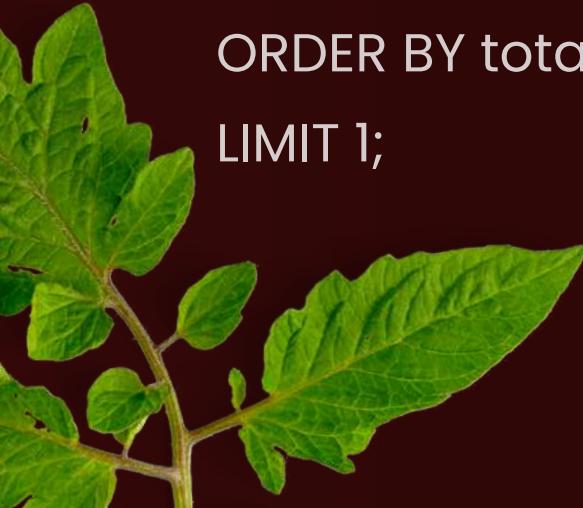
1. WHICH PIZZA CATEGORY GENERATES THE HIGHEST TOTAL REVENUE?

Q U E R Y

```
SELECT
    pt.category,
    SUM(p.price * od.quantity) AS total_revenue
FROM
    order_details AS od JOIN
    pizzas AS p ON od.pizza_id = p.pizza_id
JOIN
    pizza_types AS pt ON p.pizza_type_id = pt.pizza_type_id
GROUP BY pt.category
ORDER BY total_revenue DESC
LIMIT 1;
```

O U T P U T

	category	total_revenue
▶	Classic	220053.10



2. WHAT ARE THE TOP 5 BEST-SELLING PIZZA TYPES BY TOTAL QUANTITY SOLD?

Q U E R Y

```
SELECT  
    pt.name,  
    SUM(od.quantity) AS total_sold  
FROM  
    order_details AS od JOIN  
    pizzas AS p ON od.pizza_id = p.pizza_id JOIN  
    pizza_types AS pt  
    ON p.pizza_type_id = pt.pizza_type_id  
GROUP BY  pt.name  
ORDER BY  total_sold DESC  
LIMIT 5;
```

O U T P U T

	name	total_sold
▶	The Classic Deluxe Pizza	2453
	The Barbecue Chicken Pizza	2432
	The Hawaiian Pizza	2422
	The Pepperoni Pizza	2418
	The Thai Chicken Pizza	2371



[Home](#)[About](#)[Contact](#)

3. WHAT IS THE AVERAGE ORDER VALUE (AOV) ACROSS ALL ORDERS?

Q U E R Y

```
SELECT  
    ROUND(AVG(order_total), 2)  
    AS average_order_value  
FROM (  
    SELECT o.order_id,  
        SUM(p.price * od.quantity) AS order_total  
    FROM  
        orders AS o JOIN order_details AS od  
    ON o.order_id = od.order_id JOIN  
        pizzas AS p ON od.pizza_id = p.pizza_id  
    GROUP BY o.order_id  
) AS totals;
```

O U T P U T

	average_order_value
▶	38.31



4. WHICH PIZZA SIZE CONTRIBUTES THE MOST TO TOTAL REVENUE?

QUERY

```
SELECT  
    p.size,  
    SUM(p.price * od.quantity) AS total_revenue  
FROM  
    order_details AS od  
JOIN  
    pizzas AS p ON od.pizza_id = p.pizza_id  
GROUP BY  
    p.size  
ORDER BY  
    total_revenue DESC;
```

OUTPUT

	size	total_revenue
▶	L	375318.70
	M	249382.25
	S	178076.50
	XL	14076.00
	XXL	1006.60



5. WHAT ARE THE PEAK ORDERING HOURS DURING THE DAY?

QUERY

```
SELECT  
    EXTRACT(HOUR FROM time) AS hour_of_day,  
    COUNT(*) AS total_orders  
FROM  
    orders  
GROUP BY  
    hour_of_day  
ORDER BY  
    total_orders DESC;
```

OUTPUT

	hour_of_day	total_orders
▶	12	2520
	13	2455
	18	2399
	17	2336
	19	2009
	16	1920
	20	1642
	14	1472
	15	1468
	11	1231
	21	1198
	22	663
	23	28
	10	8
	9	1

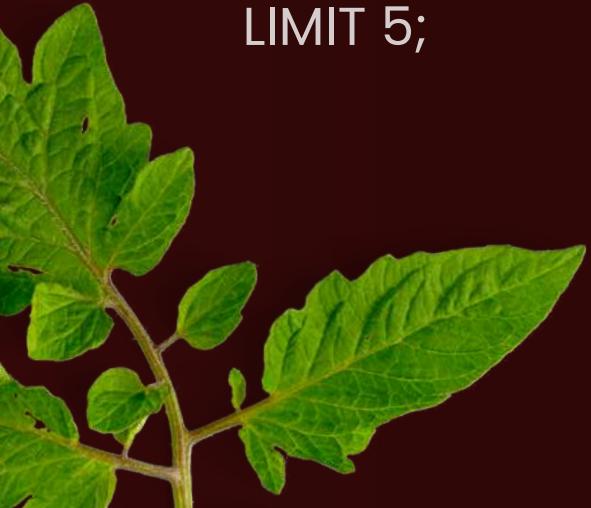
6. WHICH PIZZAS ARE SLOW-MOVING (LEAST ORDERED OVERALL)?

Q U E R Y

```
SELECT pt.name,  
       SUM(od.quantity) AS total_sold  
  FROM pizza_types AS pt  
  JOIN  
    pizzas AS p ON pt.pizza_type_id = p.pizza_type_id  
LEFT JOIN  
    order_details AS od ON p.pizza_id = od.pizza_id  
 GROUP BY pt.name  
ORDER BY total_sold ASC  
LIMIT 5;
```

O U T P U T

	name	total_sold
▶	The Brie Carre Pizza	490
▶	The Mediterranean Pizza	934
▶	The Calabrese Pizza	937
▶	The Spinach Supreme Pizza	950
▶	The Soppressata Pizza	961



7. WHICH DAY OF THE WEEK HAS THE HIGHEST TOTAL REVENUE?

Q U E R Y

```
SELECT  
    DAYNAME(o.date) AS day_of_week,  
    SUM(p.price * od.quantity) AS total_revenue  
FROM  
    orders o JOIN  
    order_details od ON o.order_id = od.order_id  
        JOIN  
    pizzas p ON od.pizza_id = p.pizza_id  
GROUP BY day_of_week  
ORDER BY total_revenue DESC;
```

O U T P U T

	day_of_week	total_revenue
▶	Friday	136073.90
	Thursday	123528.50
	Saturday	123182.40
	Wednesday	114408.40
	Tuesday	114133.80
	Monday	107329.55
	Sunday	99203.50

8. WHICH DAY OF THE MONTH HAS THE HIGHEST NUMBER OF ORDERS?

Q U E R Y

```
SELECT  
    EXTRACT(DAY FROM date) AS day_of_month,  
    COUNT(*) AS total_orders  
  
FROM  
    orders  
  
GROUP BY  
    day_of_month  
  
ORDER BY  
    total_orders DESC;
```

O U T P U T

	day_of_month	total_orders
▶	1	820
	15	809
	27	786
	4	747
	11	731
	10	728
	20	728
	6	727
	3	721
	8	719
	13	718
	17	717
	23	716
	14	715
	16	713
	2	712
	7	710
	9	703
	18	699
	21	687

COLLECTIVE INSIGHTS

- IDC Pizza sold over 49,000+ pizzas, showing strong demand across categories.
- Classic & Chicken categories contribute the highest order volumes and revenue.
- Large (L) and Extra Large (XL) sizes are the most frequently ordered across the dataset.
- Peak order activity occurs around 6 PM–9 PM, indicating strong dinner-time demand.
- Top-selling pizzas belong to the Classic & Supreme categories.
- Several pizzas showed NULL or missing data, highlighting the need for data validation.
- A few pizzas were never ordered, indicating either low visibility or low customer interest.
- Most revenue is generated from mid-priced pizzas (\$15–\$20 range).
- Ingredients analysis shows cheese-based and chicken-based pizzas dominate preferences.
- Day-wise sales reveal weekends outperform weekdays, especially Saturdays.
- Monthly trends show January and February as high-performing sales months.
- High-value orders often include multiple large pizzas, boosting overall revenue.
- Self-join analysis reveals clear price gaps between sizes, aiding pricing strategy.
- Filtering & NULL handling cleaned the dataset, improving overall query reliability.





Home

About

Contact

THANK YOU

● PREPARED BY SONALI GUPTA