DPDzero

INDIAN DATA CLUB

IDC PIZZA

THE GREAT PIZZA ANALYTICS CHALLENGE

TRANSFORMING RAW PIZZA SALES DATA INTO ACTIONABLE INSIGHTS USING SQL

Organized by: Indian Data Club

Sponsored by: DPDZero

ABOUT THE ANALYST

MOHD RAHIL

Skills: SQL • Python • Data Analytics • Data Visualization • Power BI • Advanced Excel • MySQL • ETL • Time Series Analysis • LSTM Models • Machine Learning (Basics) • Data Cleaning • Reporting & Dashboards

Passionate about turning messy datasets into meaningful business insights.

PROJECT OVERVIEW

A HANDS-ON SQL ANALYTICS MINI-PROJECT WHERE, AS THE DATA ANALYST FORIDC 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

ABOUT THE IDC PIZZA DATASET

THE PROJECT USES 4 RELATIONALTABLES 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.

QUESTIONS COVERED (TOTAL 18)

PHASE I: FOUNDATION

1. INSTALL IDC _ PIZZA.DUMP AS IDC _ PIZZA SERVER

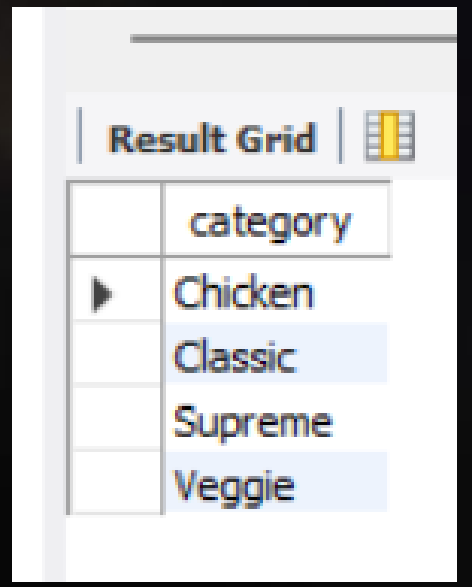
IDC _ Pizza.dump is a PostgreSQL backup file. It cannot be imported into MySQL;
it must be restored in PostgreSQL/pgAdmin.

2. List all unique pizza categories (DISTINCT).

QUERY

```
select distinct
category
From
pizza_types;
```

OUTPUT



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

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

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	'Nduja Salami, Pancetta, Tomatoes, Red Onions...
cali_ckn	The California Chicken Pizza	Chicken, Artichoke, Spinach, Garlic, Jalapeno P...

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

QUERY

```
select pizza_id,
price
from pizzas
where price is
null;
```

OUTPUT

PHASE 2: FILTERING & EXPLORATION

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

QUERY

```
select order_id,
 date
 from orders
where
date = '2015-01-01';
```

OUTPUT

Result Grid | Filter Rows

order_id	date
1	2015-01-01
2	2015-01-01
3	2015-01-01
4	2015-01-01
5	2015-01-01
6	2015-01-01
7	2015-01-01
8	2015-01-01
9	2015-01-01
10	2015-01-01
11	2015-01-01
12	2015-01-01

orders 7 | orders 8 ×

2. List pizzas with `price` descending.

QUERY

```
select pizza_id,
 price
from pizzas
 order by
 price desc;
```

OUTPUT

Result Grid | Filter Rows:

pizza_id	price
the_greek_xxl	35.95
the_greek_xl	25.50
brie_carre_s	23.65
ital_veggie_l	21.00
bbq_ckn_l	20.75
soppressata_l	20.75
southw_ckn_l	20.75
spicy_ital_l	20.75
peppr_salami_l	20.75
spin_pesto_l	20.75
thai_ckn_l	20.75
ckn_pesto_l	20.75

orders 7 | pizzas 9

3. Pizzas sold in sizes ``L'' or ``XL''.

QUERY

```
select pizza_id,
size
 from
 pizzas
Where
 size = 'L’
or size = 'XL';
```

OUTPUT

Result Grid | Filter Rows

pizza_id	size
big_meat_l	L
calabrese_l	L
cali_ckn_l	L
ckn_alfredo_l	L
ckn_pesto_l	L
classic_dlx_l	L
five_cheese_l	L
four_cheese_l	L
green_garden_l	L
hawaiian_l	L
ital_cpclo_l	L
ital_supr_l	L

orders 7 | pizzas 11 ×

4. Pizzas priced between $15.00 and $17.00.

QUERY

```
select pizza_id,
price
From
 pizzas
Where
Price
between 15.00
and 17.00;
```

OUTPUT

Result Grid | Filter Rows:

pizza_id	price
bbq_ckn_m	16.75
big_meat_m	16.00
calabrese_m	16.25
cali_ckn_m	16.75
ckn_alfredo_m	16.75
ckn_pesto_m	16.75
classic_dlx_m	16.00
five_cheese_m	15.50
green_garden_m	16.00
hawaiian_l	16.50
ital_cpclo_m	16.00
ital_supr_m	16.50

orders 7 | pizzas 16

5. Pizzas with `"Chicken"` in the name.

QUERY

```
select
pizza_type_id,
name
From
pizza_types
where
name like
'%Chicken%';
```

OUTPUT

Result Grid | Filter Rows:

pizza_type_id	name
bbq_ckn	The Barbecue Chicken Pizza
cali_ckn	The California Chicken Pizza
ckn_alfredo	The Chicken Alfredo Pizza
ckn_pesto	The Chicken Pesto Pizza
southw_ckn	The Southwest Chicken Pizza
thai_ckn	The Thai Chicken Pizza

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

QUERY

```
select order_id,
date,
 time
 from orders
where date =
'2015-02-15’
 or time >
'20:00:00';
```

OUTPUT

Result Grid | Filter Rows:

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	20:51:42
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
68	2015-01-01	22:07:32
69	2015-01-01	22:12:13
123	2015-01-02	20:12:09
124	2015-01-02	20:12:34

orders 7 | orders 18

PHASE 3: SALES PERFORMANCE

1. Total quantity of pizzas sold (`SUM`).

QUERY

```
select
sum(quantity) as
total_pizza_sold
from
order_details;
```

OUTPUT

Result Grid

	total_pizza_sold
▶	49574

2. Average pizza price (`AVG`).

QUERY

```
select
round(avg(price),2)
as avg_pizza_price
from pizzas;
```

OUTPUT

Result Grid | Filter Row

avg_pizza_price
16.44

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

QUERY

```
select od.order_id,
sum(p.price * od.quantity)
as total_order_value
from order_details od
 join pizzas p on
od.pizza_id = p.pizza_id
group by
od.order_id;
```

OUTPUT

Result Grid | Filter Rows:

order_id	total_order_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

orders 7 | Result 33

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

QUERY

```
select pt.category,
sum(od.quantity) as
pizza_sold_by_category
from pizza_types pt
join pizzas p on
pt.pizza_type_id = p.pizza_type_id
join order_details od
on p.pizza_id = od.pizza_id
group by pt.category;
```

OUTPUT

Result Grid | Filter Rows:

category	pizza_sold_by_category
Classic	14888
Veggie	11649
Supreme	11987
Chicken	11050

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

QUERY

```
select pt.category,
sum(od.quantity) as
pizza_sold_by_category
from pizza_types pt
join pizzas p
on pt.pizza_type_id = p.pizza_type_id
join order_details od
on p.pizza_id = od.pizza_id
group by pt.category
having
sum(od.quantity) >5000;
```

OUTPUT

Result Grid | Filter Rows:

category	pizza_sold_by_category
Classic	14888
Veggie	11649
Supreme	11987
Chicken	11050

6. Pizzas never ordered (`LEFT/RIGHT JOIN`).

QUERY

```
SELECT
p.*
from order_details od
Right join pizzas p
    on
 p.pizza_id = od.pizza_id
Where
 od.pizza_id is null;
```

OUTPUT

Result Grid | Filter Rows:

pizza_id	pizza_type_id	size	price
big_meat_l	big_meat	L	20.50
big_meat_m	big_meat	M	16.00
five_cheese_m	five_cheese	M	15.50
five_cheese_s	five_cheese	S	12.50
four_cheese_s	four_cheese	S	11.75

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

QUERY

```
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 p1.size < p2.
Order by p1.pizza_type_id,
p1.size;
```

OUTPUT

Result Grid | Filter Rows: | Export: | Wrap Cell

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	M	16.75	4.00
bbq_ckn	S	12.75	L	20.75	8.00
big_meat	S	12.00	M	16.00	4.00
big_meat	M	16.00	L	20.50	4.50
big_meat	S	12.00	L	20.50	8.50
calabrese	M	16.25	L	20.25	4.00
calabrese	S	12.25	M	16.25	4.00
calabrese	S	12.25	L	20.25	8.00
cali_ckn	M	16.75	L	20.75	4.00
cali_ckn	S	12.75	M	16.75	4.00
cali_ckn	S	12.75	L	20.75	8.00

orders 7 | Result 37

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

THANK YOU

PREPARED BY: MOHD RAHIL