Pizza Sales Analysis

By Purva Pharat

INTRODUCTION

My name is Purva Pharat, and I undertook this project to enhance my SQL skills, explore its capabilities, and gain hands-on experience with database management.

This project focuses on analyzing pizza sales data to uncover trends and insights, such as popular pizza types, revenue patterns, and customer preferences.

Through this, I aim to practice SQL effectively and develop a deeper understanding of data-driven decision-making.

OBJECTIVE

- To enhance SQL skills through hands-on practice.
- To analyze pizza sales data for trends and patterns.
- To uncover insights like popular pizzas and revenue contributions.
- To deepen understanding of data-driven decision-making.
- To improve proficiency in managing relational databases.

DATASET DESCRIPTION

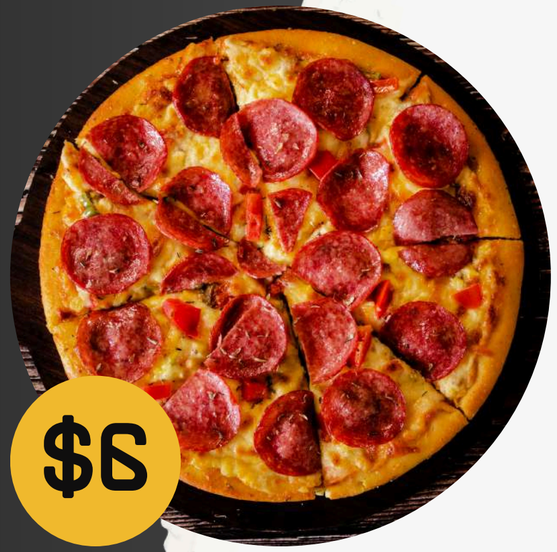
The Pizzadb dataset consists of four tables designed to manage and analyze the operations of a pizza business:

- **Orders: order_id, order_date, order_time –** Tracks order trends.
- **Order Details: order_detailed_id, order_id, pizza_id, quantity –** For revenue and order composition.
- **Pizzas: pizza_type_id, size, price, pizza_id –** Analyzes sales by size and pricing.
- **Pizza Types: pizza_id, pizza_type_id, pizza_category_name –** Analyzes sales by pizza category.

TO DOWNLOAD THE DATA

CLICK HERE

BASIC QUERY



Retrieve the total number of orders placed.

```
select count(order_id) as order_count
from orders;
```

	order_count
▶	21350

Calculate the total revenue generated from pizza sales.

```
select round(sum(o_d.quantity*p.price),2) as total_revenue
from order_details o_d
join pizzas p
using(pizza_id);
```

	total_revenue
▶	817860.05

Identify the highest-priced pizza.

```
select p_t.name,p.pizza_id,p.price
from pizzas p
        join pizza_types p_t using(pizza_type_id)
where price = (
        select max(price)
        from pizzas);
```

	name	pizza_id	price
▶	The Greek Pizza	the_greek_xxl	35.95

Identify the most common pizza size ordered.

```
select p.size  , count(p.size) as pizza_count
from order_details o
        join pizzas p using(pizza_id)
group by p.size
order by count(p.size) desc
limit 1;
```

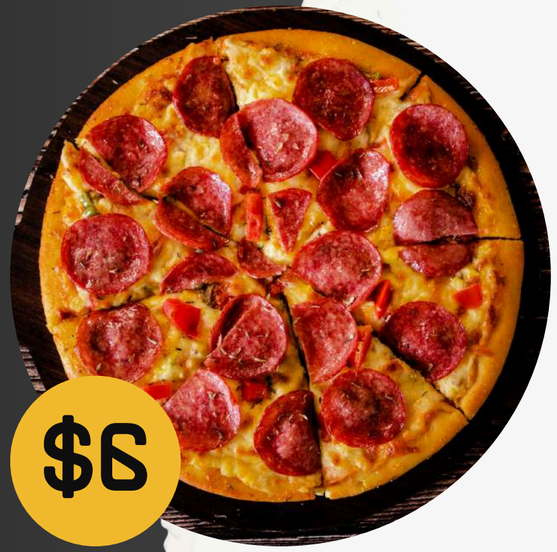
	size	pizza_count
▶	L	18526

List the top 5 most ordered pizza types along with their quantities.

```
select p_t.name,sum(o.quantity) as quantity
from order_details o
        join pizzas p using(pizza_id)
        join pizza_types p_t  using(pizza_type_id)
group by p_t.name
order by sum(o.quantity) desc
limit 5;
```

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

INTERMEDIATE QUERY



Join the necessary tables to find the total quantity of each pizza category ordered.

```
select p_t.category,sum(o.quantity) as quantity
from order_details o
        join pizzas p using(pizza_id)
        join pizza_types p_t using(pizza_type_id)
group by p_t.category
order by sum(o.quantity) desc;
```

	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

Determine the distribution of orders by hour of the day.

```
with cte as(
select hour(time) as hour,count(order_id) as order_count
from orders
group by hour(time)),
cte1 as(
select *,hour+1 as hour2
from cte)

select concat(hour,"-",hour2) as time,order_count
from cte1;
```

time	order_count
11-12	1231
12-13	2520
13-14	2455
14-15	1472
15-16	1468
16-17	1920
17-18	2336
18-19	2399
19-20	2009
20-21	1642
21-22	1198
22-23	663
23-24	28
10-11	8
9-10	1

Join relevant tables to find the category-wise distribution of pizzas.

```
select category,count(category) as category_available
from pizza_types
group by category;
```

category	category_available
Chicken	6
Classic	8
Supreme	9
Veggie	9

Group the orders by date and calculate the average number of pizzas ordered per day.

```
select round(avg(order_count)) as avg_no_of_pizzas_ordered_per_day
from(
        select o.date,sum(od.quantity) as order_count
        from orders o
                join order_details od using(order_id)
        group by o.date) as data;
```

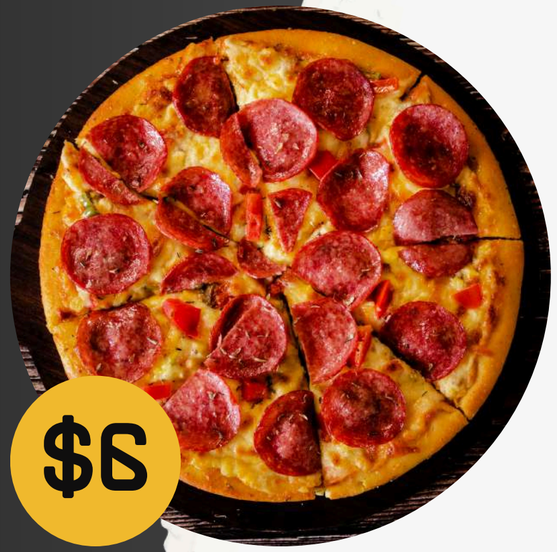
	avg_no_of_pizzas_ordered_per_day
▶	138

Determine the top 3 most ordered pizza types based on revenue.

```
select p_t.name , round(sum(p.price*o.quantity),2) as revenue
from order_details o
        join pizzas p using(pizza_id)
        join pizza_types p_t using(pizza_type_id)
group by p_t.name
order by sum(p.price*o.quantity) desc
limit 3;
```

	name	revenue
▶	The Thai Chicken Pizza	43434.25
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41409.5

ADVANCED QUERY



Calculate the percentage contribution of each pizza type to total revenue.

```
select p_t.category,round(round(sum(p.price* o1.quantity)) /
(select sum(p.price* o1.quantity)
from order_details o1
        join pizzas p using(pizza_id)
        join pizza_types p_t using(pizza_type_id)) * 100 , 2 ) as revenue
from order_details o1
        join pizzas p using(pizza_id)
        join pizza_types p_t using(pizza_type_id)
group by p_t.category;
```

category	revenue
Classic	26.91
Veggie	23.68
Supreme	25.46
Chicken	23.96

Analyze the cumulative revenue generated over time.

```
with cte as(
select round(sum(p.price* o1.quantity),2) as revenue,o.date
from orders o
        join order_details o1 using(order_id)
        join pizzas p using(pizza_id)
        join pizza_types p_t using(pizza_type_id)
group by o.date
order by o.date)

select round(sum(revenue) over(partition by month(date) order by date),2) as cumulative_sum,date
from cte;
```

cumulative_sum	date
2713.85	2015-01-01
5445.75	2015-01-02
8108.15	2015-01-03
9863.6	2015-01-04
11929.55	2015-01-05
14358.5	2015-01-06
16560.7	2015-01-07
19399.05	2015-01-08
21526.4	2015-01-09
23990.35	2015-01-10
25862.65	2015-01-11
27781.7	2015-01-12
29831.3	2015-01-13
32358.7	2015-01-14
34343.5	2015-01-15
36937.65	2015-01-16
39001.75	2015-01-17
40978.6	2015-01-18

Determine the top 3 most ordered pizza types based on revenue for each pizza category.

```
with cte as(
select round(sum(p.price* o1.quantity)) as revenue , p_t.name , p_t.category
from orders o
        join order_details o1 using(order_id)
        join pizzas p using(pizza_id)
        join pizza_types p_t using(pizza_type_id)
group by p_t.name,p_t.category
),cte1 as(
select *,dense_rank()over(partition by category order by revenue desc) as rnk
from cte)

select name,category,revenue
from cte1
where rnk <= 3;
```

Determine the top 3 most ordered pizza types based on revenue for each pizza category.

name	category	revenue
The Thai Chicken Pizza	Chicken	43434
The Barbecue Chicken Pizza	Chicken	42768
The California Chicken Pizza	Chicken	41410
The Classic Deluxe Pizza	Classic	38180
The Hawaiian Pizza	Classic	32273
The Pepperoni Pizza	Classic	30162
The Spicy Italian Pizza	Supreme	34831
The Italian Supreme Pizza	Supreme	33477
The Sicilian Pizza	Supreme	30940
The Four Cheese Pizza	Veggie	32266
The Mexicana Pizza	Veggie	26781
The Five Cheese Pizza	Veggie	26066

CONTACT

7391922894

purvapharat17@gmail.com

https://github.com/pbf1712/

THANK YOU!