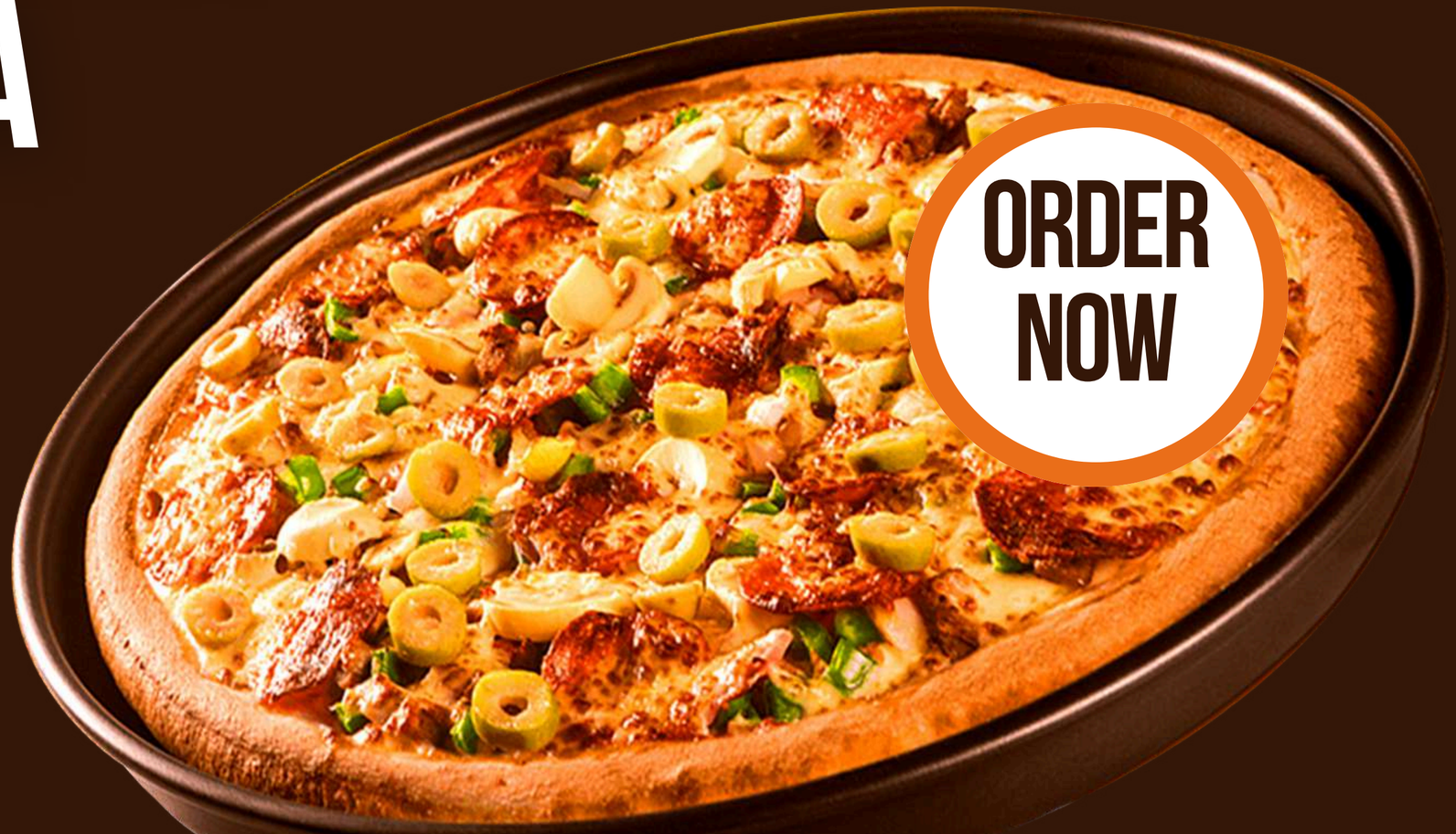
Where Every Slice is a Taste of Perfection

SQL PROJECT ON PIZZA SALES



👋 HELLO!
I'M NIDHI BOPCHE

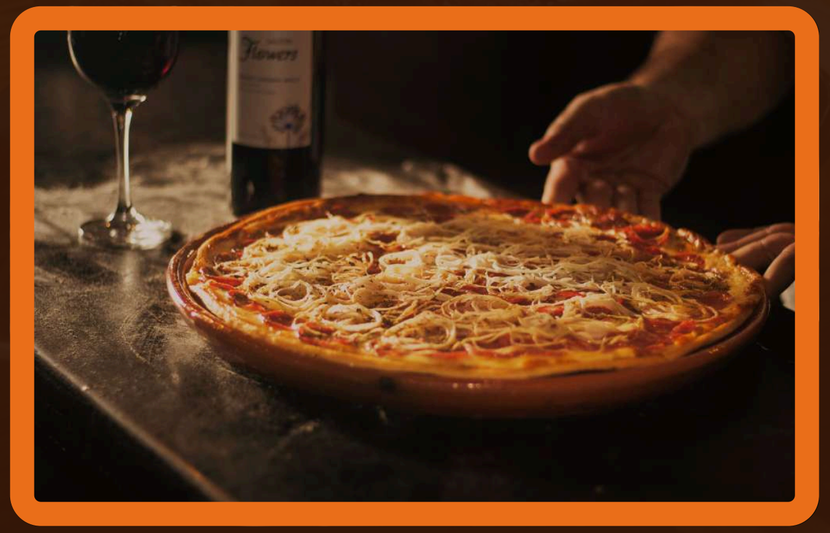
INTRODUCTION

Welcome to my Pizza Sales Analysis Project
In this project, I explored a pizza sales dataset and used SQL queries to answer key business questions related to sales performance, order patterns, and customer preferences. The goal was to extract meaningful insights from raw data to help drive smarter business decisions — all through the power of structured queries.
Let’s dive in!

Q1 RETRIVE THE TOTAL NUMBER OF ORDERS PLACED.

Easy Questions

```
SELECT
    COUNT(order_id) AS total_orders
FROM
    orders;
```



Result Grid

	total_orders
▶	21350

Q2 CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

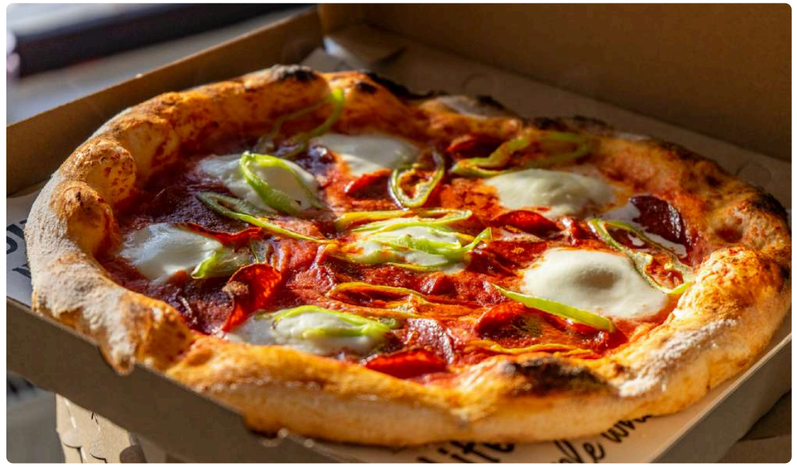
```
SELECT
    ROUND(SUM(od.quantity * p.price), 2) AS total_sales
FROM
    order_details od
        JOIN
    pizzas p ON od.pizza_id = p.pizza_id;
```

Result Grid

	total_sales
▶	817860.05

Q3 IDENTIFY THE HIGHEST-PRICED PIZZA.

```
SELECT
    pt.name, p.price
FROM
    pizzas p
        JOIN
    pizza_types pt ON p.pizza_type_id = pt.pizza_type_id
ORDER BY p.price DESC
LIMIT 1;
```



Result Grid | Filte

name	price
The Greek Pizza	35.95

Q4 IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

```
SELECT
    quantity, COUNT(order_details_id)
FROM
    order_details
GROUP BY quantity;
```

```
select p.size, count(od.order_details_id) as order_count
from pizzas p
join order_details od
on p.pizza_id = od.pizza_id
group by p.size
order by order_count desc ;
```

Result Grid | Filte

quantity	COUNT(orde
1	47693
2	903
3	21
4	3

Result Grid | Filter

size	order_count
L	18526
M	15385
S	14137
XL	544
XXL	28

Q5 LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES.

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

Result Grid | Filter Rows:

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

Q1 JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED.

Intermediate Questions

```
SELECT
    pt.category, SUM(od.quantity) AS 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 od.pizza_id = p.pizza_id
GROUP BY pt.category
ORDER BY quantity DESC;
```

Result Grid

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Q2 DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

Intermediate Questions

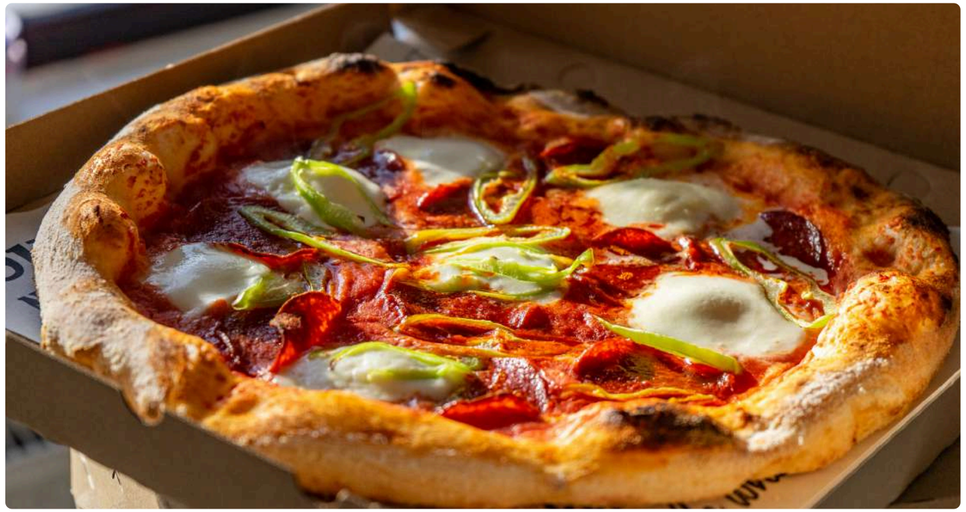
```
SELECT
    HOUR(order_time) AS hour, COUNT(order_id) AS
order_count
FROM
    orders
GROUP BY HOUR(order_time);
```

Result Grid

	hour	order_count
▶	11	1231
	12	2520
	13	2455
	14	1472
	15	1468

Q3 JOIN RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS.

```
SELECT
    category, COUNT(name)
FROM
    pizza_types
GROUP BY category
```

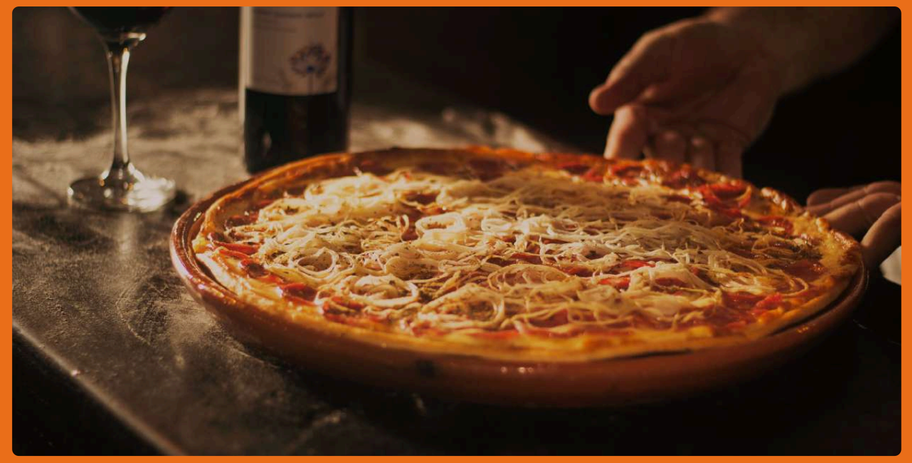


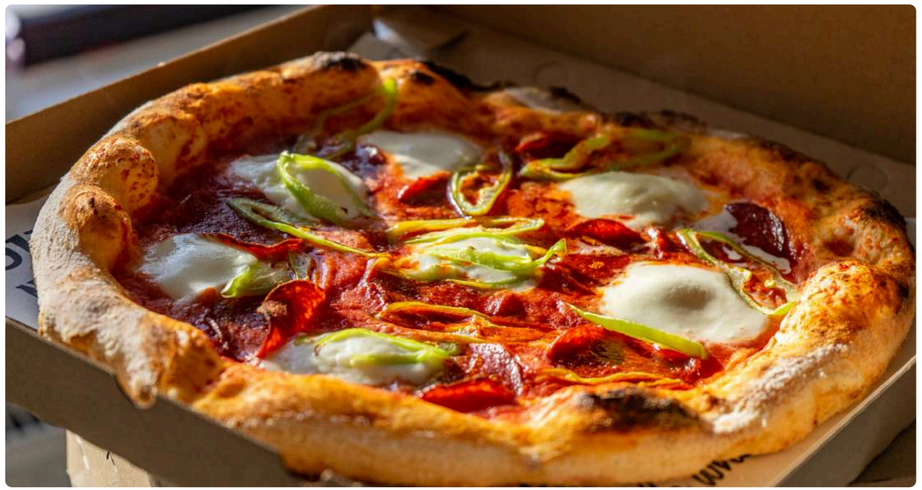
Result Grid

category	COUNT(na
Chicken	6
Classic	8
Supreme	9
Veggie	9

Q4 GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY.

```
SELECT
    ROUND(AVG(quantity), 0)
FROM
    (SELECT
        o.order_date, SUM(od.quantity) AS quantity
    FROM
        orders o
    JOIN order_details od ON o.order_id = od.order_id
    GROUP BY o.order_date) AS order_quantity;
```



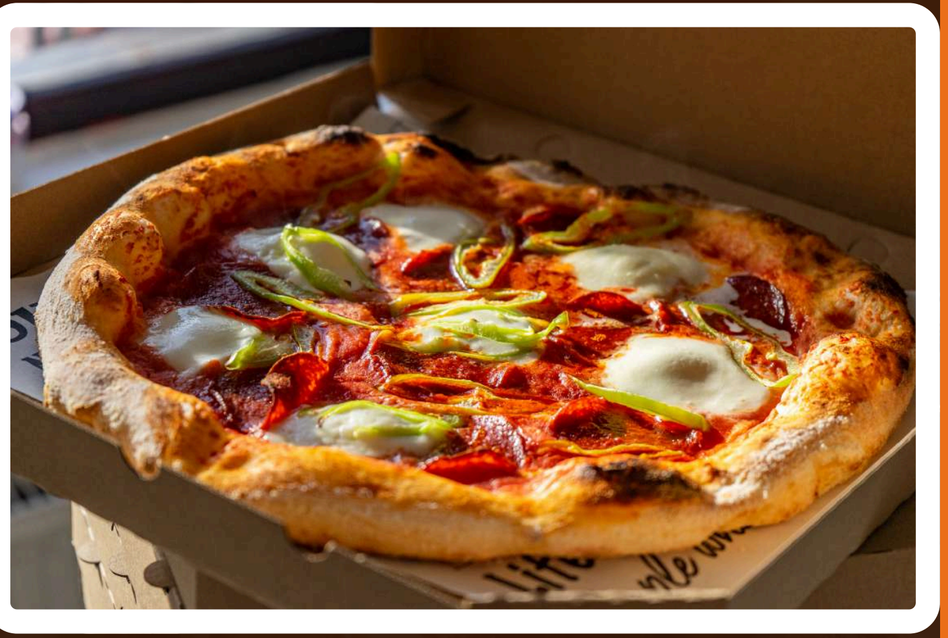


Result Grid | Filter Ro

	ROUND(AVG(quantity), 0)
▶	138

Q5 DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.

```
SELECT
    pt.name, SUM(p.price * od.quantity) AS revenue
FROM
    pizza_types pt
        JOIN
    pizzas p ON pt.pizza_type_id = p.pizza_type_id
        JOIN
    order_details od ON od.pizza_id = p.pizza_id
GROUP BY pt.name
ORDER BY revenue DESC
LIMIT 3;
```



Result Grid | Filter Rows:

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

Difficult Questions

Q1. CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE.

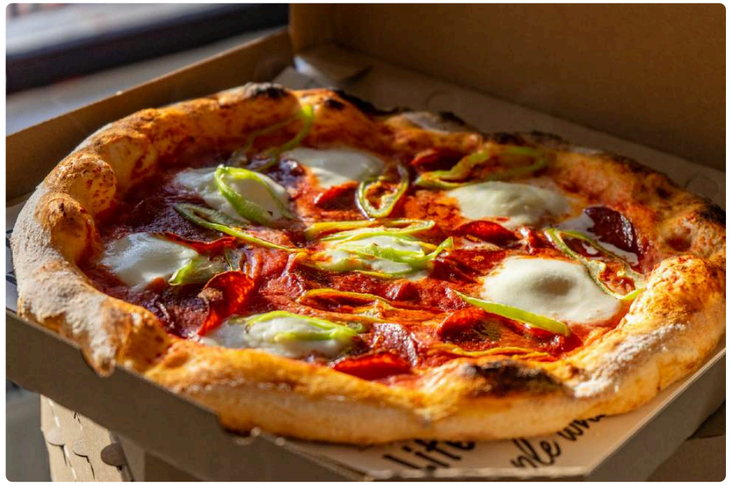
```
select pt.category, round(sum(od.quantity*p.price)/ ( select
round(sum(od.quantity*p.price),2) as total_sales
from order_details od join pizzas p
on od.pizza_id = p.pizza_id)*100,2) as revenue
from pizza_types pt join pizzas p
on pt.pizza_type_id = p.pizza_type_id
join order_details od
on od.pizza_id = p.pizza_id
group by pt.category order by revenue desc;
```

Result Grid | Filter

category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

Q2. ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

```
select order_date,
sum(revenue) over(order by order_date) as cum_revenue
from
(select o.order_date, sum(od.quantity*p.price) as revenue
from order_details od join orders o
on od.order_id = o.order_id
join pizzas p
on p.pizza_id = od.pizza_id
group by o.order_date) as sales;
```

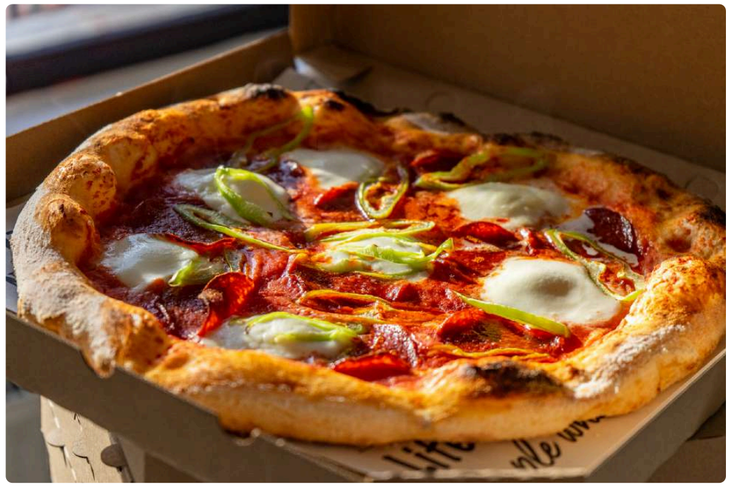


Result Grid | Filter Rows:

order_date	cum_revenue
2015-01-01	2713.8500000000004
2015-01-02	5445.75
2015-01-03	8108.15
2015-01-04	9863.6
2015-01-05	11929.55

Q3. DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY.

```
select name, revenue from
(select category, name, revenue,
rank() over(partition by category order by revenue desc) as rn
from
(select pt.category, pt.name,
sum(od.quantity*p.price) as revenue
from order_details od join pizzas p
ON od.pizza_id = p.pizza_id
join pizza_types pt
on pt.pizza_type_id = p.pizza_type_id
group by pt.category, pt.name) as a) as b
where rn <= 3;
```



Result Grid | Filter Rows:

name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5
The Classic Deluxe Pizza	38180.5
The Hawaiian Pizza	32273.25

Result 5

Pizza Sales Presentation

THANK YOU FOR ATTENTION

https://github.com/nidhibopche?tab=repositories