

PIZZA SALES ANALYSIS

● PRIYANSHU YADAV



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

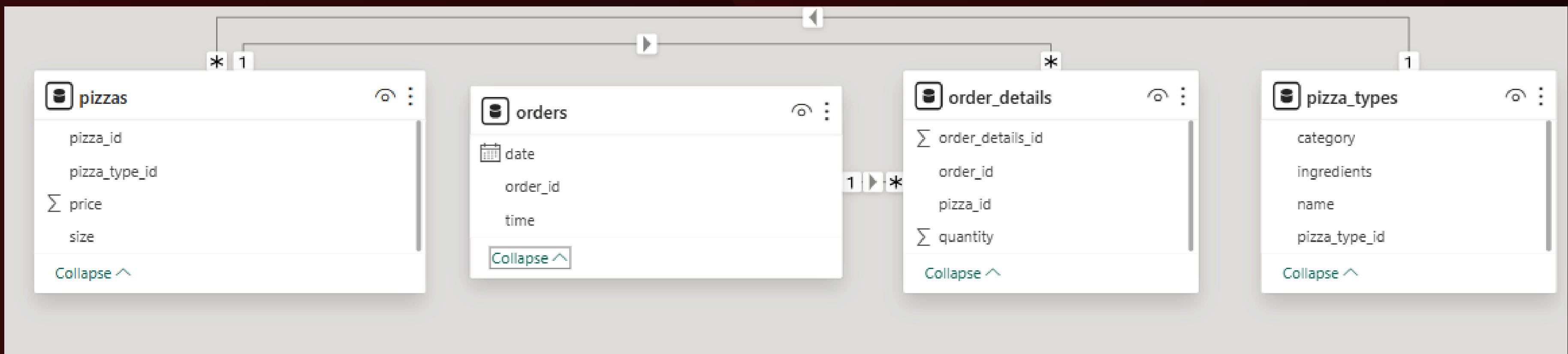
OBJECTIVE

To design a relational database that efficiently stores and manages pizza order data, and to perform in-depth sales analysis using advanced SQL queries. The project showcases skills in database design, data analysis, and query optimization, aligned with real-world business scenarios.



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

PIZZA SALES DATABASE SCHEMA





RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

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

Result Grid	
	total_orders
▶	21350





QUESTION - 2

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

Calculate the total revenue generated from pizza sales.

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

Result Grid	
	total_revenue
▶	572385.7





QUESTION - 3

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

Identify the highest-priced pizza.

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

Result Grid | Filter Rows:

	name	price
▶	The Greek Pizza	35.95





QUESTION - 4

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

DENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

```
SELECT
    pizzas.size,
    COUNT(order_details.order_details_id) AS order_count
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY order_count DESC;
```

	size	order_count
▶	L	12969
	M	10776
	S	9865
	XL	396
	XXL	19



QUESTION - 5

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

SELECT

```
pizza_types.name, COUNT(order_details.quantity) AS quantity
```

FROM

```
pizza_types
```

JOIN

```
pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
```

JOIN

```
order_details ON order_details.pizza_id = pizzas.pizza_id
```

GROUP BY pizza_types.name

ORDER BY quantity **DESC**

LIMIT 5;

[Home](#)

[About](#)

[Contact](#)

	name	quantity
▶	The Barbecue Chicken Pizza	1698
	The Pepperoni Pizza	1673
	The Classic Deluxe Pizza	1664
	The Hawaiian Pizza	1625
	The California Chicken Pizza	1612



QUESTION - 1



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

```
SELECT
    pizza_types.category,
    SUM(order_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY quantity DESC;
```

Result Grid | Filter R

	category	quantity
▶	Classic	10393
	Supreme	8379
	Veggie	8216
	Chicken	7694



QUESTION - 2

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

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

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

Result Grid | Filter

	hours	order_count
▶	11	1231
	12	2520
	13	2455
	14	1472
	15	1468
	16	1920
	17	2336
	18	2399
	19	2009



QUESTION - 3

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

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

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

	category	COUNT(name)
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9



QUESTION - 4

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

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

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

Result Grid | Filter Rows:

	avg_pizza_ordered_perday
▶	138





QUESTION - 5

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

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

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

	name	revenue
▶	The Barbecue Chicken Pizza	30457.25
	The Thai Chicken Pizza	29728.25
	The California Chicken Pizza	28881.25



QUESTION - 1

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

```
SELECT
    pizza_types.category,
    ROUND(SUM(order_details.quantity * pizzas.price) / (SELECT
        ROUND(SUM(order_details.quantity * pizzas.price),
        2) AS total_sales
    )
FROM
    order_details
        JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id) *100,2) as revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY revenue DESC;
```

	category	revenue
▶	Classic	26.87
	Supreme	25.47
	Veggie	23.87
	Chicken	23.79



QUESTION - 2

[About](#)

Analyze the cumulative revenue generated over time.

```
...  
...  
...  
...  
...  
SELECT order_date,  
       sum(revenue) over(order by order_date) as cum_revenue  
from  
(SELECT  
    orders.order_date,  
    SUM(order_details.quantity * pizzas.price) AS revenue  
FROM  
    order_details  
    JOIN  
    pizzas ON order_details.pizza_id = pizzas.pizza_id  
    JOIN  
    orders ON orders.order_id = order_details.order_id  
GROUP BY orders.order_date) as sales;
```

	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
	2015-01-06	14358.5



QUESTION - 3

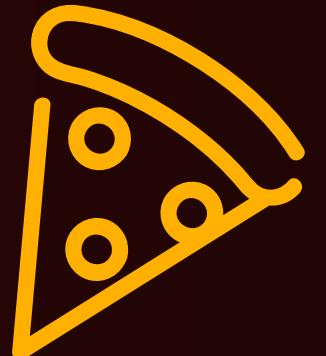
Determine the top 3 most ordered pizza types based on revenue for each pizza category.

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

	name	revenue
▶	The Barbecue Chicken Pizza	30457.25
	The Thai Chicken Pizza	29728.25
	The California Chicken Pizza	28881.25
	The Classic Deluxe Pizza	26241.5
	The Hawaiian Pizza	22157.75
	The Pepperoni Pizza	21226.5



THANK YOU



Home

About

Contact

