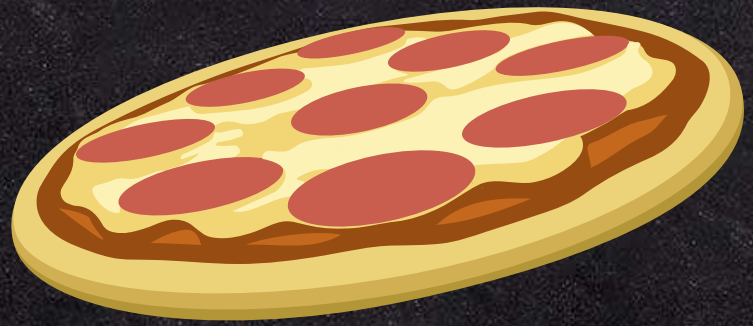
PIZZA SALES ANALYSIS USING SQL

B A S I C T O A D V A N C E

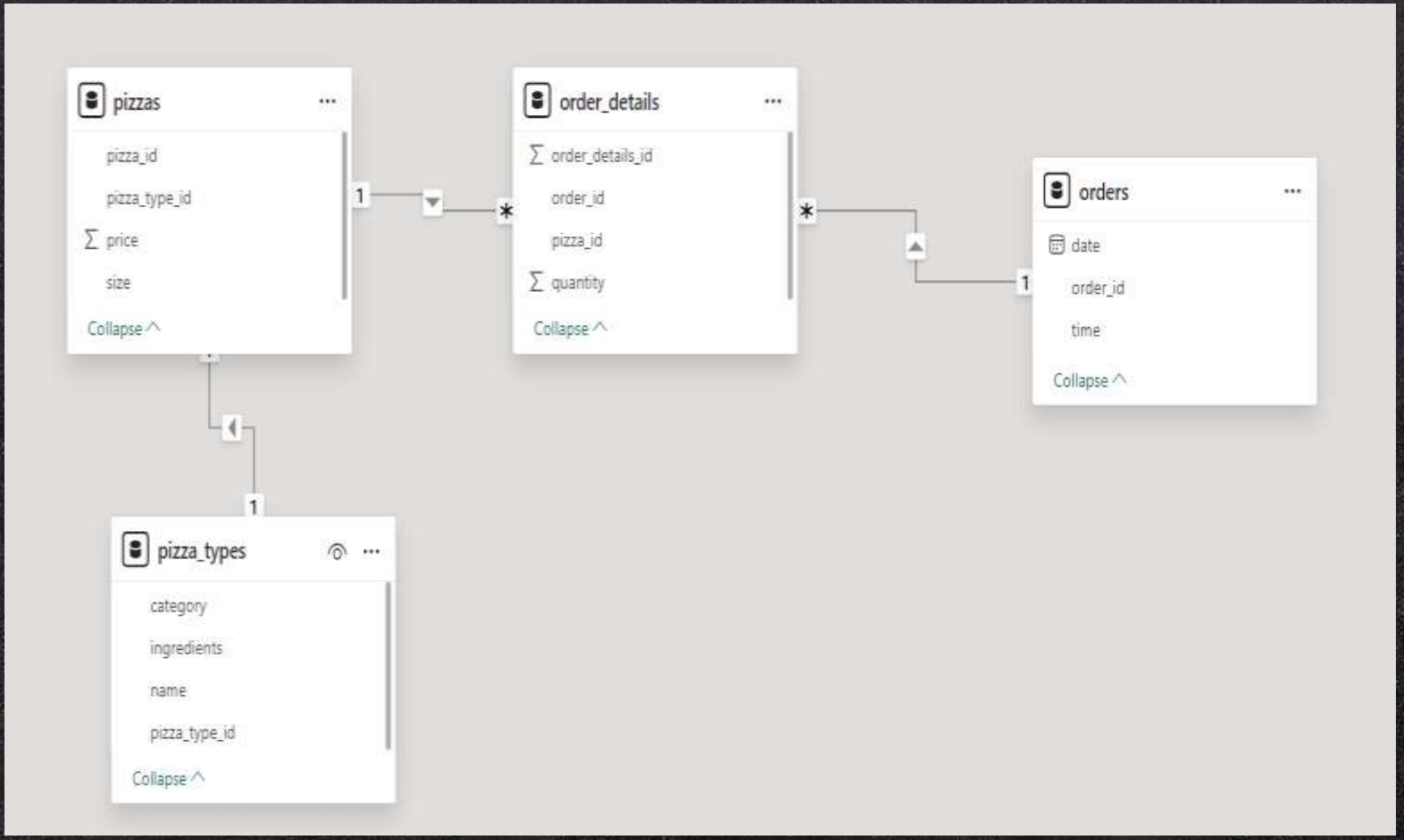
Who doesn’t like pizza?



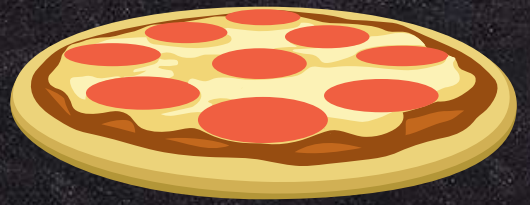
A B O U T
P R O J E C T

- I recently completed an exciting project using SQL to analyze pizza sales data across four tables: Orders, Order Details, Pizzas, and Pizza Type.
- By writing SQL queries, I uncovered patterns in total revenue, popular pizza types, peak sales times, and customer preferences.
- This project not only sharpened my SQL skills but also demonstrated how data can drive better business decisions.

RELATIONSHIP BETWEEN TABLES



A many-to-one relationship exists when multiple rows in a child table relate to a single row in a parent table.



Retrieve the total number of orders placed

```sql
SELECT
    COUNT(id) AS Total_Orders
FROM
    Orders;
```

Result Grid

	Total_Orders
▶	21350

Calculate the total revenue generated from pizza sales.

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

Result Grid

	total_revenue
▶	817860.05

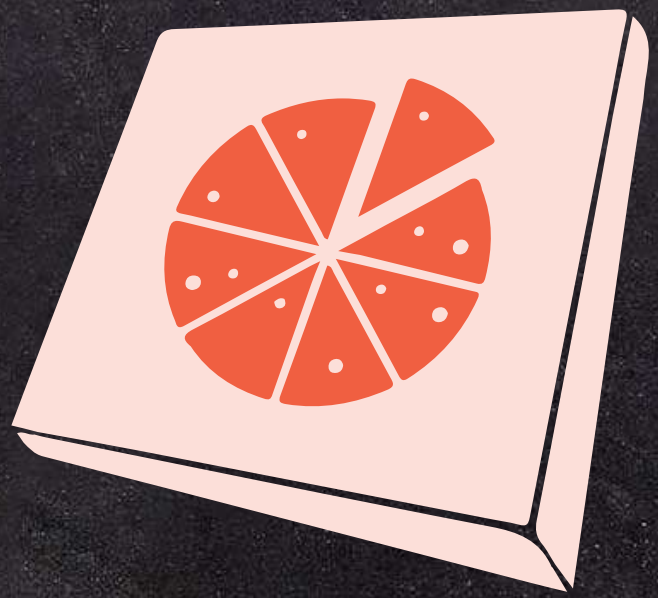
Identify the highest-priced pizza

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

Result Grid | Filter Rows:

name	highest_price
The Greek Pizza	35.95

Identify the most common pizza size ordered.



```sql
SELECT
    pizzas.size,
    COUNT(orders_details.order_details_id) AS order_count
FROM
    pizzas
        JOIN
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizzas.size
ORDER BY order_count DESC
LIMIT 1;
```

Result Grid | Filter Rows:

	size	order_count
▶	L	18526

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

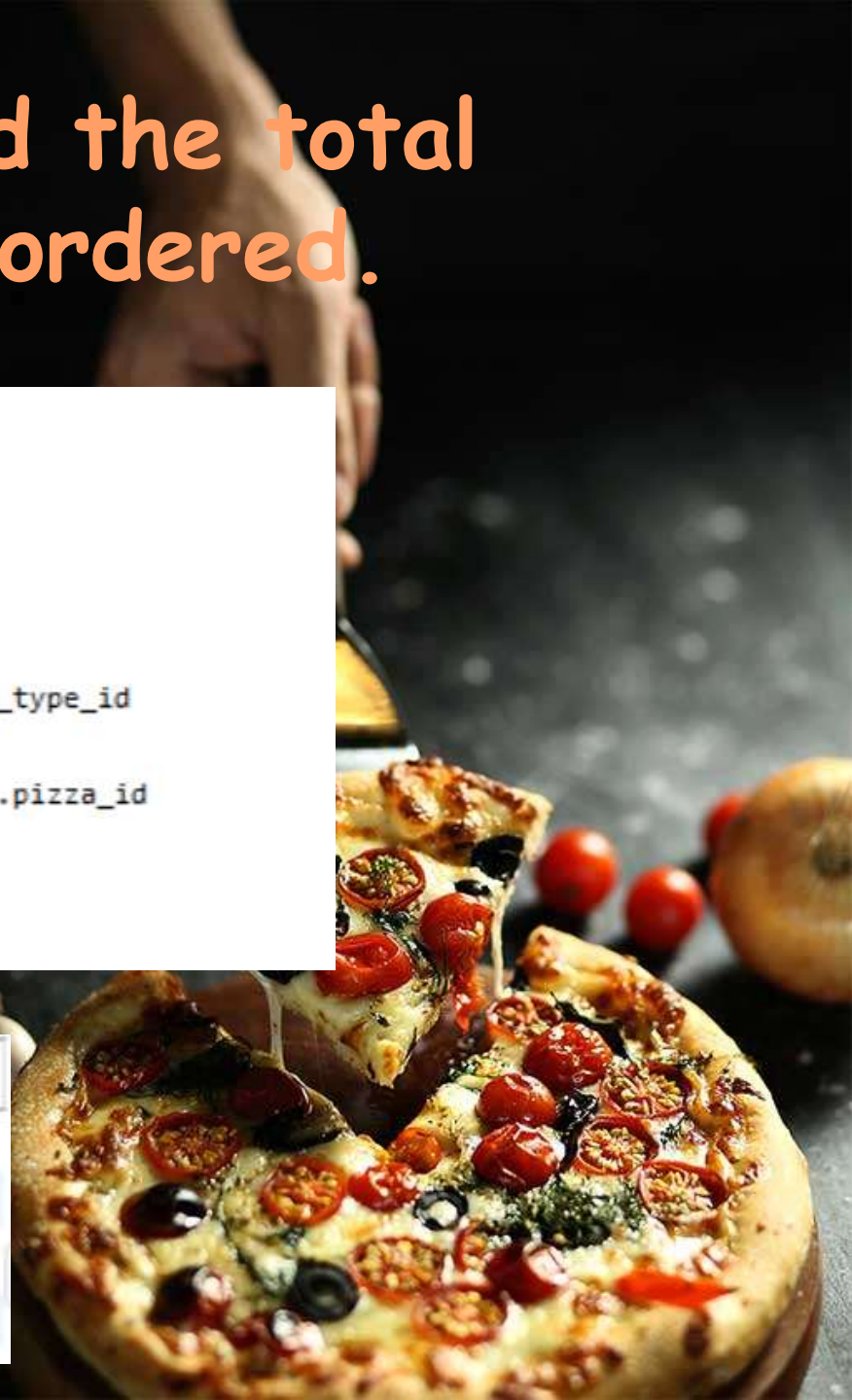
```sql
SELECT
    pizza_types.name, SUM(orders_details.quantity) AS quantities
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY quantities DESC
LIMIT 5;
```

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

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

```sql
SELECT
    pizza_types.category,
    SUM(orders_details.quantity) AS quantities
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY quantities DESC
LIMIT 5;
```

	category	quantities
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050



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

```
SELECT
    pizza_types.category,
    COUNT(pizza_types.pizza_type_id) AS count
FROM
    pizza_types
GROUP BY pizza_types.category
```

	category	count
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

Determine the distribution of orders by hour of the day

```
select hour(order_time) as hour, count(id) from orders
group by hour order by count(id) desc ;
```

	hour	count(id)
▶	12	2520
	13	2455
	18	2399
	17	2336
	19	2009

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

```sql
SELECT
    ROUND(AVG(quantity), 2) AS avg_quantity
FROM
    (SELECT
        orders.order_date AS order_date,
            SUM(orders_details.quantity) AS quantity
    FROM
        orders
    JOIN orders_details ON orders_details.order_id = orders.id
    GROUP BY orders.order_date) AS order_quantity;
```

	avg_quantity
▶	138.47

Determine the top 3 most ordered pizza types based on revenue

```sql
SELECT
    pizza_types.name,
    round(SUM(orders_details.quantity * pizzas.price),2) AS Revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    orders_details ON pizzas.pizza_id = orders_details.pizza_id
GROUP BY pizza_types.name
ORDER BY Revenue
LIMIT 3;
```

	name	Revenue
▶	The Brie Carre Pizza	11588.5
	The Green Garden Pizza	13955.75
	The Spinach Supreme Pizza	15277.75

Calculate the percentage contribution of each pizza type to total revenue

category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

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

Analyze the cumulative revenue generated over time.

```sql
select order_date,
sum(revenue) over(order by order_date)  as cumulative_revenue
from
(select orders.order_date, sum(pizzas.price*orders_details.quantity) as revenue
from orders_details
join pizzas on pizzas.pizza_id = orders_details.pizza_id
join orders on orders_details.order_id = orders.id
group by orders.order_date) as rev_sales;
```

order_date	cumulative_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

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

```sql
SELECT name, revenue from
        (select category,name, revenue ,
        rank() over (partition by category order by revenue) as rn from
                (select pizza_types.category,pizza_types.name,
                sum(orders_details.quantity*pizzas.price) as revenue from pizza_types
            join pizzas
            on pizzas.pizza_type_id = pizza_types.pizza_type_id
            join orders_details
            on orders_details.pizza_id = pizzas.pizza_id
            group by pizza_types.category,pizza_types.name) as cte) as cte2
            where rn<= 3;
```

name	revenue
The Chicken Pesto Pizza	16701.75
The Chicken Alfredo Pizza	16900.25
The Southwest Chicken Pizza	34705.75
The Pepperoni, Mushroom, and Peppers Pizza	18834.5
The Big Meat Pizza	22968


THANK YOU