SALES
Pizza Resto

SQL PROJECT

- ON PIZZA SALES

SALES
Pizza Resto

ABOUT ME

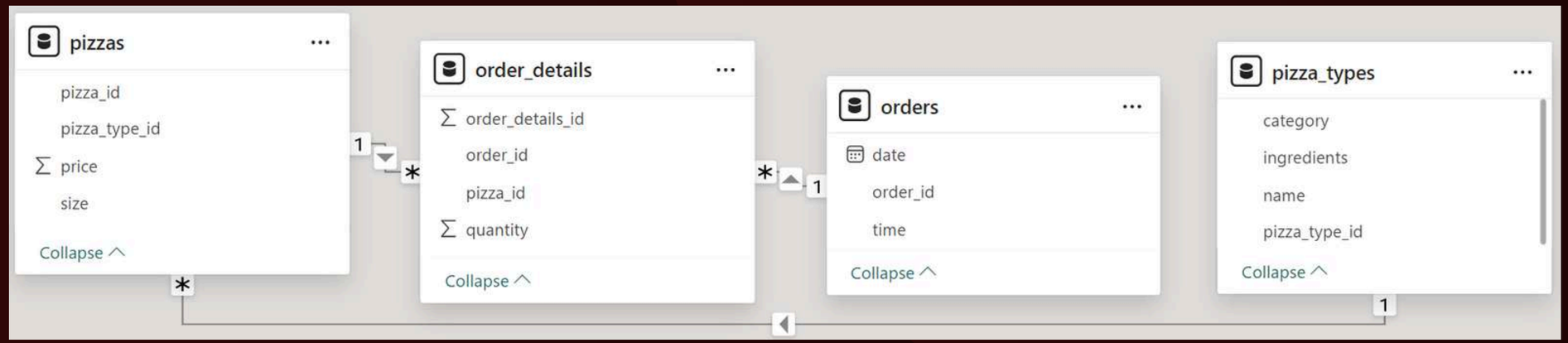
I'm Purvi Kapoor, and I'm passionate about using SQL to solve business problems. In my recent project, I analyzed pizza sales data to answer questions like:

- What are the top 5 best-selling pizzas?
- What is the total revenue for each pizza size?

I used SQL queries to extract the data and analyze the results. Through this project, I learned a lot about SQL and data analysis, and I'm excited to continue learning and growing in this field.

SALES
Pizza Resto

SCHEMA



PIZZAS

Pizza Resto

ORDER_DETAILS

Pizza Resto

ORDERS

Pizza Resto

PIZZA_TYPES

Pizza Resto

SALES
Pizza Resto

RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED

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

Result Grid

	total_orders
▶	21350

SALES
Pizza Resto

CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

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

Result Grid

	total_sales
▶	817860.05

SALES
Pizza Resto

IDENTIFY THE HIGHEST-PRICED PIZZA

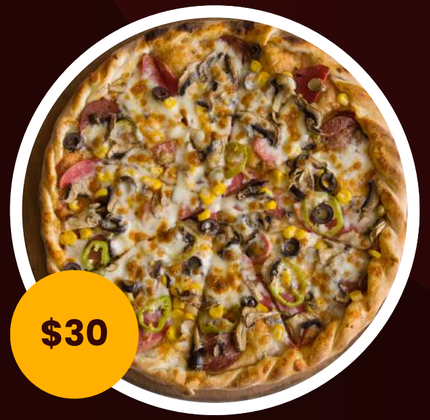
```
SELECT
    pizza_types.name, pizzas.price
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
ORDER BY pizzas.price DESC
LIMIT 1;
```

Result Grid | Filter Rows

name	price
The Greek Pizza	35.95

SALES
Pizza Resto

DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE




```sql
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;
```

Result Grid | Filter Rows:

NAME	REVENUE
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

SALES
Pizza Resto

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED

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

Result Grid | Filter

	size	order_count
▶	L	18526

SALES
Pizza Resto

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

```sql
SELECT
    pizza_types.name, 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.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

SALES
Pizza Resto

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


```sql
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
                        INNER JOIN
                    pizzas ON pizzas.pizza_id = order_details.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 SUM(order_details.quantity * pizzas.price) DESC;
```

Result Grid | Filter

	category	revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

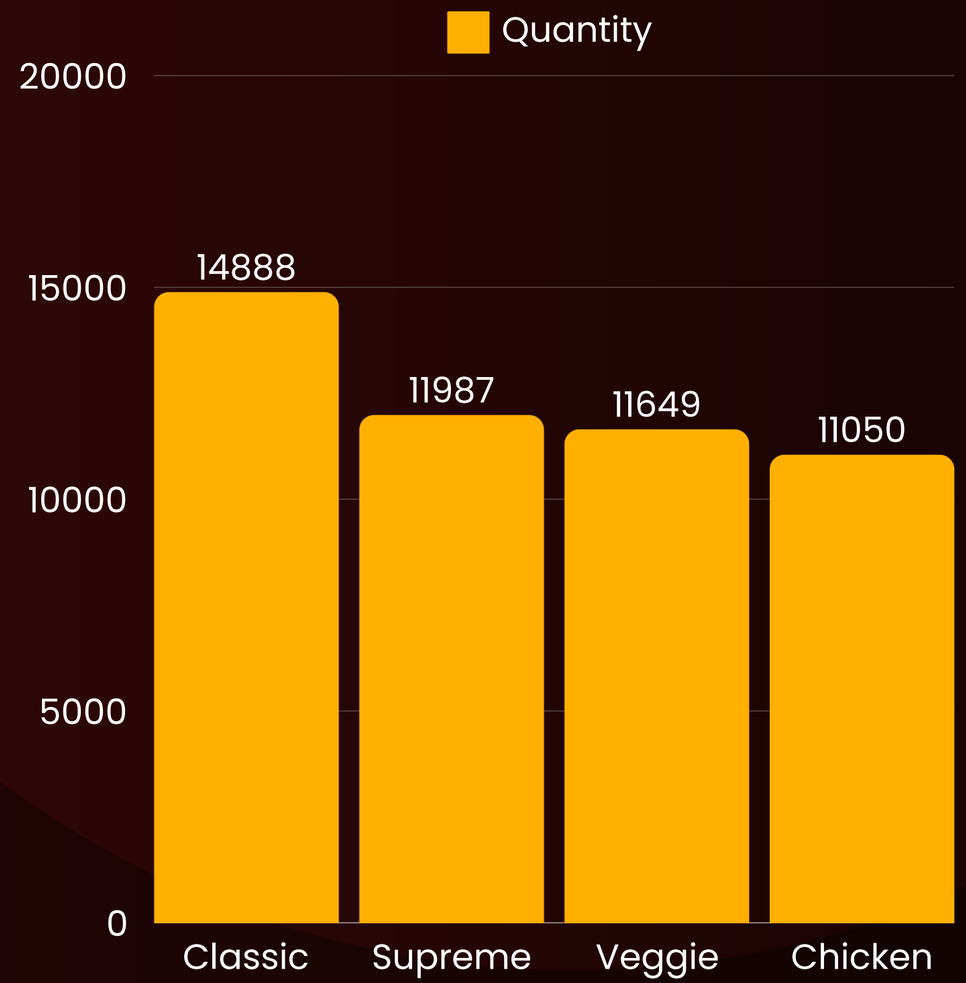
SALES
Pizza Resto

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

```sql
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

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050



SALES
Pizza Resto

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY

ORDERS PER HOUR

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

Result Grid | Filter

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

SALES
Pizza Resto

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

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

Result Grid | Filter Rows:

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

SALES
Pizza Resto

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

```sql
SELECT
    ROUND(AVG(QUANTITY), 0) AS AVG_PIZZAS_ORDERED_PER_DAY
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_PIZZAS_ORDERED_PER_DAY
138

SALES
Pizza Resto

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME


```sql
select order_date , sum(revenue) over (order by order_date) as cum_revenue
from
(select round(sum(order_details.quantity * pizzas.price),2) as revenue , orders.order_date
from order_details
join orders
on order_details.order_id = orders.order_id
join pizzas
on order_details.pizza_id = pizzas.pizza_id
group by orders.order_date) as sales;
```

Result Grid | Filter Rows:

order_date	cum_revenue
2015-01-01	2713.85
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
2015-01-07	16560.7
2015-01-08	19399.05
2015-01-09	21526.399999999998
2015-01-10	23990.35
2015-01-11	25862.649999999998
2015-01-12	27781.699999999997
2015-01-13	29831.299999999996
2015-01-14	32358.699999999997

Result Grid | Filter Rows:

order_date	cum_revenue
2015-12-17	791892.55
2015-12-18	794778.8500000001
2015-12-19	797083.05
2015-12-20	799187.9500000001
2015-12-21	801288.65
2015-12-22	803171.6
2015-12-23	805415.9
2015-12-24	807553.75
2015-12-26	809196.8
2015-12-27	810615.8
2015-12-28	812253
2015-12-29	813606.25
2015-12-30	814944.05
2015-12-31	817860.05

SALES
Pizza Resto

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 desc ) as rn
from
(SELECT
    pizza_types.NAME,
    SUM(order_details.quantity * PIZZAS.PRICE) AS REVENUE , pizza_types.category
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 , pizza_types.category) 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
The Pepperoni Pizza	30161.75
The Spicy Italian Pizza	34831.25
The Italian Supreme Pizza	33476.75
The Sicilian Pizza	30940.5
The Four Cheese Pizza	32265.70000000065
The Mexicana Pizza	26780.75
The Five Cheese Pizza	26066.5

SALES
Pizza Resto

THANK YOU

FOR ATTENTION

SQL PROJECT PRESENTATION