

PROJECT LA-PINOZ PIZZA SALES



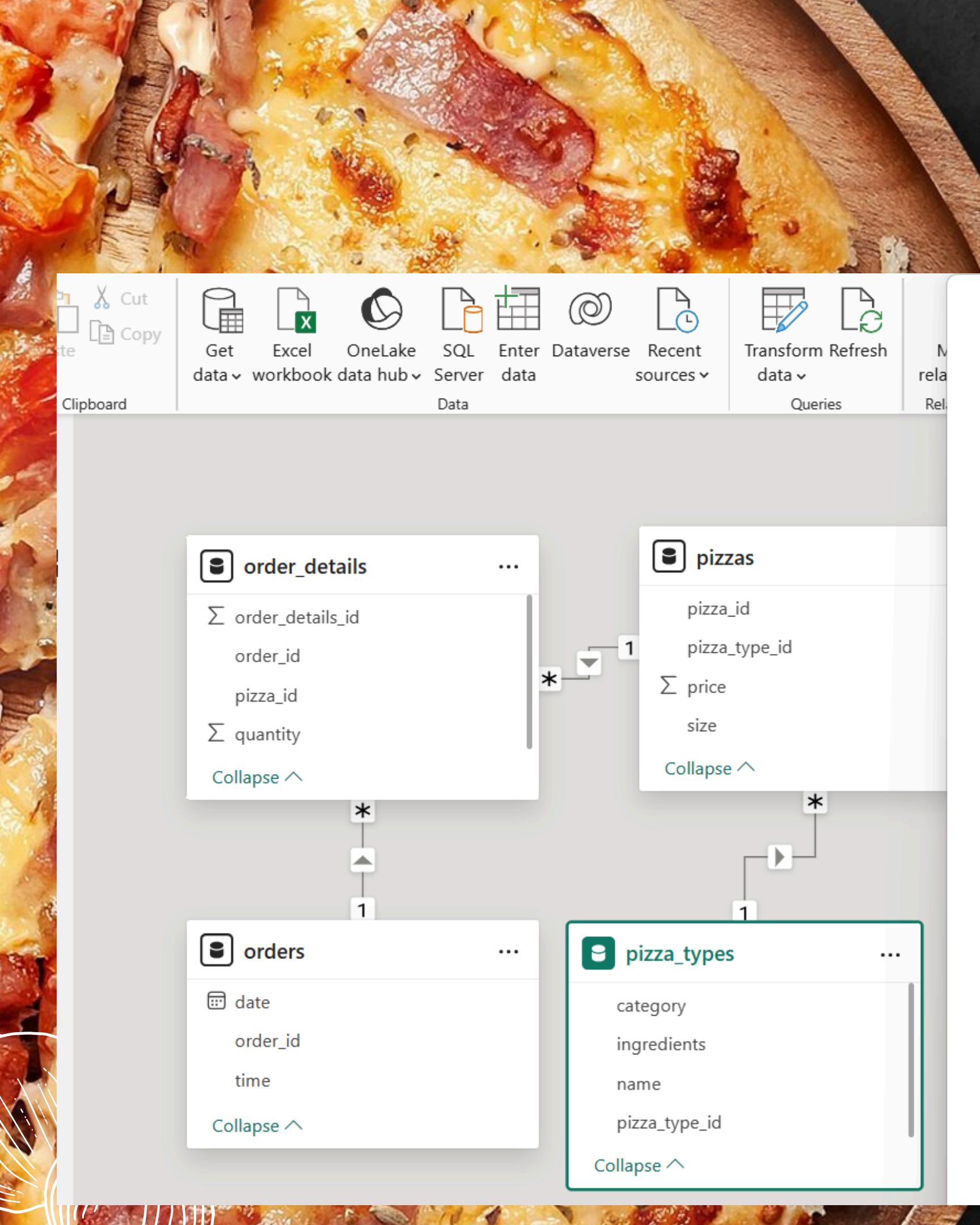
WELCOME TO

LA-PINOZ PIZZA SQL PROJECT

My Name is Shubham Malviy, In this project, I have utilized SQL Queries to solve questions that were related to La-Pinoz Pizza Sales.



DATA MODEL



The image shows a detailed view of a pizza slice, highlighting its texture and toppings.

This image is overlaid with a Microsoft Power BI interface, illustrating how data models can be used to analyze such food items.

Power BI ribbon:

- Clipboard: Cut, Copy
- Data: Get data, Excel, OneLake, SQL Server, Enter data, Dataverse, Recent sources, Transform data, Refresh data
- Queries: Transform, Refresh, Edit, Delete

Manage relationships:

New relationship | **Autodetect**

From: table (column)	Relationship	To: table (column)	Status
order_details (order_id)	* —> 1	orders (order_id)	Active
order_details (pizza_id)	* —> 1	pizzas (pizza_id)	Active
pizzas (pizza_type_id)	1 —> *	pizza_types (pizza_type_id)	Active

Data Model Diagram:

```
graph TD; order_details[order_details] -- "*" --> 1 --> pizzas[pizzas]; pizzas -- 1 --> * --> pizza_types[pizza_types]; orders[orders] -- 1 --> * --> order_details; pizza_types -- 1 --> * --> pizzas;
```

The diagram illustrates the relationships between four tables:

- order_details:** Contains columns like order_details_id, order_id, pizza_id, and quantity.
- pizzas:** Contains columns like pizza_id, pizza_type_id, price, and size.
- orders:** Contains columns like date, order_id, and time.
- pizza_types:** Contains columns like category, ingredients, name, and pizza_type_id.

Relationships are defined as follows:

- A many-to-one relationship exists from order_details (pizza_id) to pizzas (pizza_id).
- A many-to-one relationship exists from order_details (order_id) to orders (order_id).
- A one-to-many relationship exists from pizza_types (pizza_type_id) to pizzas (pizza_type_id).
- A one-to-many relationship exists from orders (order_id) to order_details (order_id).



QUESTIONS:

- Identify the most common pizza size ordered.
- List the top 5 most ordered pizza types along with their quantities.
- Join the necessary tables to find the total quantity of each pizza category ordered.
- Determine the distribution of orders by hour of the day.
- Join relevant tables to find the category-wise distribution of pizzas.
- Group the orders by date and calculate the average number of pizzas ordered per day.
- Determine the top 3 most ordered pizza types based on revenue.
- Calculate the percentage contribution of each pizza type to total revenue.
- Analyze the cumulative revenue generated over time.
- Determine the top 3 most ordered pizza types based on revenue for each pizza category.



-- Identify the most common pizza size ordered.

SELECT

pizzas.size,
COUNT(order_details.order_details_id) **AS** common_pizza_size

FROM

pizzas

JOIN

order_details **ON** pizzas.pizza_id = order_details.pizza_id

GROUP BY pizzas.size

ORDER BY common_pizza_size **DESC**

| Result Grid | Filter Rows:

	size	common_pizza_size
▶	L	18526
	M	15385
	S	14137
	XL	544
	XXL	28



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

SELECT

pizza_types.name, SUM(order_details.quantity) AS quantities

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 quantities DESC

LIMIT 5

Result Grid | Filter Rows:

	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.

SELECT

pizza_types.category, SUM(order_details.quantity) AS quantities

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 quantities DESC

Result Grid | Filter Rows:

	category	quantities
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050



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

SELECT
    ROUND(AVG(quantity), 0) as avg_pizza_order_perday
FROM
    (SELECT
        orders.order_date, SUM(order_details.quantity) AS quantity
    FROM
        order_details
    JOIN orders ON order_details.order_id = orders.order_id
    GROUP BY orders.order_date) AS ordered_quantity
```

Result Grid | Filter Rows:

	avg_pizza_order_perday
▶	138



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

SELECT

```
    pizza_types.name,  
    SUM(pizzas.price * order_details.quantity) 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

Result Grid | Filter Rows:

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



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

```
SELECT
    pizza_types.category,
    round(sum(pizzas.price * order_details.quantity) / (SELECT
        ROUND(SUM(order_details.quantity * pizzas.price),
        2) AS total_sales
    )
FROM
    order_details
        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 revenue desc
```

Result Grid | Filter

	category	revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68



-- 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
	2015-01-07	16560.7
	2015-01-08	19399.05
	2015-01-09	21526.4
	2015-01-10	23990.350000000002
	2015-01-11	25862.65

BEST SELLER



-- 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(pizzas.price * order_details.quantity) 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) b
where rn <= 3
```

	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



LA-PINOZ PIZZA

THANK YOU!

