

SQL PROJECT

# PIZZA SALES



Atharv nampalliwar



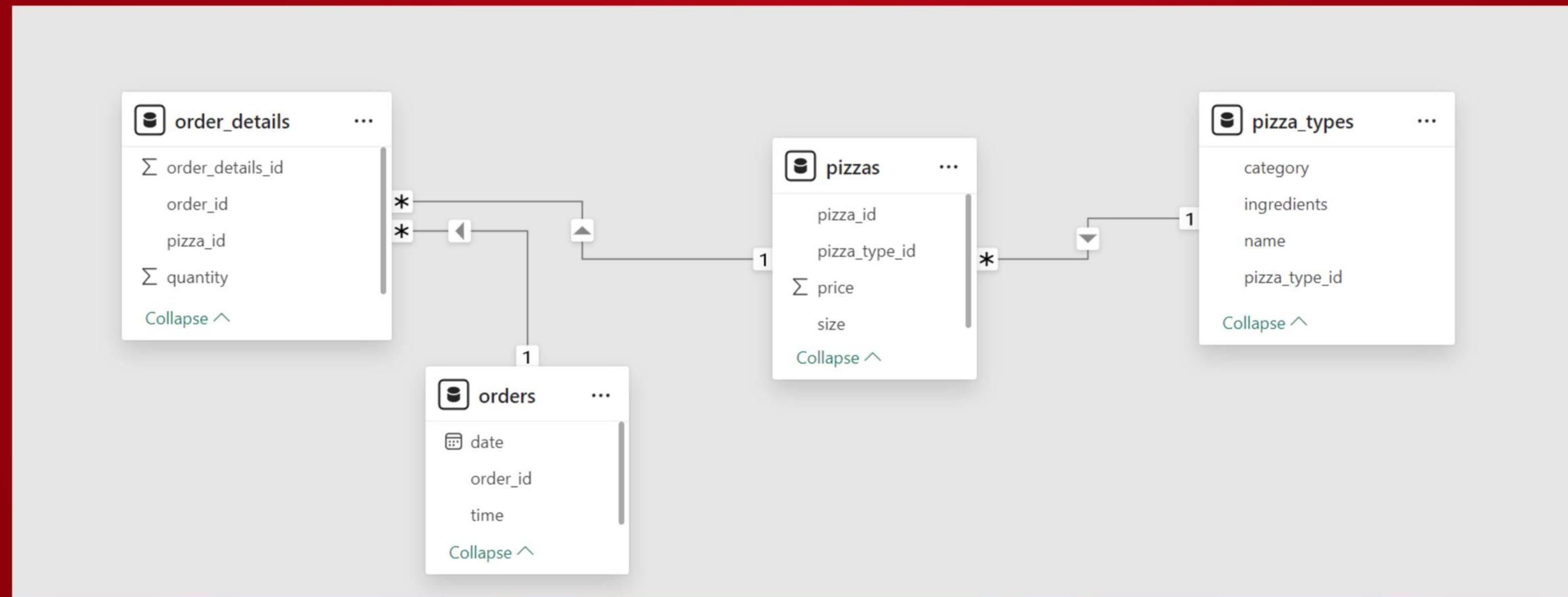
# OBJECTIVE

THE OBJECTIVE OF THIS PROJECT IS TO ANALYZE PIZZA SALES DATA BY RETRIEVING TOTAL ORDERS, REVENUE, AND TOP-PERFORMING PIZZAS, IDENTIFYING TRENDS IN CATEGORY-WISE AND SHOWCASING SQL PROFICIENCY AND REAL-WORLD DATA ANALYSIS SKILLS.





# DATABASE SCHEMA





# PROBLEM STATEMENTS

## BASIC:

- RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.
- CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.
- IDENTIFY THE HIGHEST-PRICED PIZZA.
- IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.
- LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES.



# PROBLEM STATEMENTS

## INTERMEDIATE :

- JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED.
- DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.
- 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.



# PROBLEM STATEMENTS

## ADVANCED :

- 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.



PIZZA SALES

CATEGORY

BASIC





PIZZA SALES



RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

```
select count(order_id) as total_orders from orders;
```



Result Grid	
	total_orders
▶	21350



# CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

```
select  
    round(sum(quantity * price),2) as total_sales  
from order_details as od  
join pizzas as p on od.pizza_id = p.pizza_id
```



Result Grid	
	total_sales
▶	817860.05



# IDENTIFY THE HIGHEST-PRICED PIZZA.

```
select max(price) from pizzas;  
  
-- or --  
  
select name , price from pizza_types join pizzas on pizza_types.pizza_type_id = pizzas.pizza_type_id  
order by price desc limit 1;
```



Result Grid | Filter Rows:

	name	price
▶	The Greek Pizza	35.95





# IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

```
select size , count(order_details_id) from
pizzas join order_details on order_details.pizza_id = pizzas.pizza_id
group by size;
```

Result Grid | Filter Rows:

	size	count(order_details_id)
▶	M	15385
	L	18526
	S	14137
	XL	544
	XXL	28





PIZZA SALES



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

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

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



PIZZA SALES

# CATEGORY INTERMEDIATE





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

```
select category , sum(quantity) as total_qty  
from pizza_types  
join pizzas on pizza_types.pizza_type_id = pizzas.pizza_type_id  
join order_details on pizzas.pizza_id = order_details.pizza_id  
group by category  
order by total_qty desc;
```



Result Grid | Filter

	category	total_qty
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050



PIZZA SALES



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

```
select hour(time) as hours , count(order_id) as total_orders from orders  
group by hours;
```

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



PIZZA SALES



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

```
select category , count(name) from pizza_types  
group by category;
```

Result Grid | Filter Row

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



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

```
SELECT  
    date, round(AVG(quantity),2) AS avg_pizzas_ordered_per_day  
FROM  
    orders  
JOIN order_details ON orders.order_id = order_details.order_id  
GROUP BY date ;
```



Result Grid		Filter Rows:
	date	avg_pizzas_ordered_per_day
▶	2015-01-01	1.01
	2015-01-02	1.03
	2015-01-03	1.03
	2015-01-04	1.00
	2015-01-05	1.03
	2015-01-06	1.02
	2015-01-07	1.04
	2015-01-08	1.01
	2015-01-09	1.03
	2015-01-10	1.01
	2015-01-11	1.02
	2015-01-12	1.01
	2015-01-13	1.03
	2015-01-14	1.04
	2015-01-15	1.00
	2015-01-16	1.02
	2015-01-17	1.02



PIZZA SALES

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

```
select name , sum(quantity * price) as revenue  
from pizza_types  
join pizzas on pizzas.pizza_type_id = pizza_types.pizza_type_id  
join order_details on order_details.pizza_id = pizzas.pizza_id  
group by 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





PIZZA SALES

# CATEGORY ADVANCED →





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

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

Result Grid | Filter Rows:

category	revenue
Classic	26.90596025566967
Veggie	23.682590927384577
Supreme	25.45631126009862
Chicken	23.955137556847287



# ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

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

	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
	2015-01-12	27781.7
	2015-01-13	29831.300000000003



# 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 category , name ,
sum(quantity * 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 category , 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
	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	32265.7000000
	The Mexicana Pizza	26780.75
	The Five Cheese Pizza	26066.5



PIZZA SALES

# THANK YOU!

