

SQL PROJECT ON PIZZA SALES

- WHERE EVERY SLICE TELLS A STORY

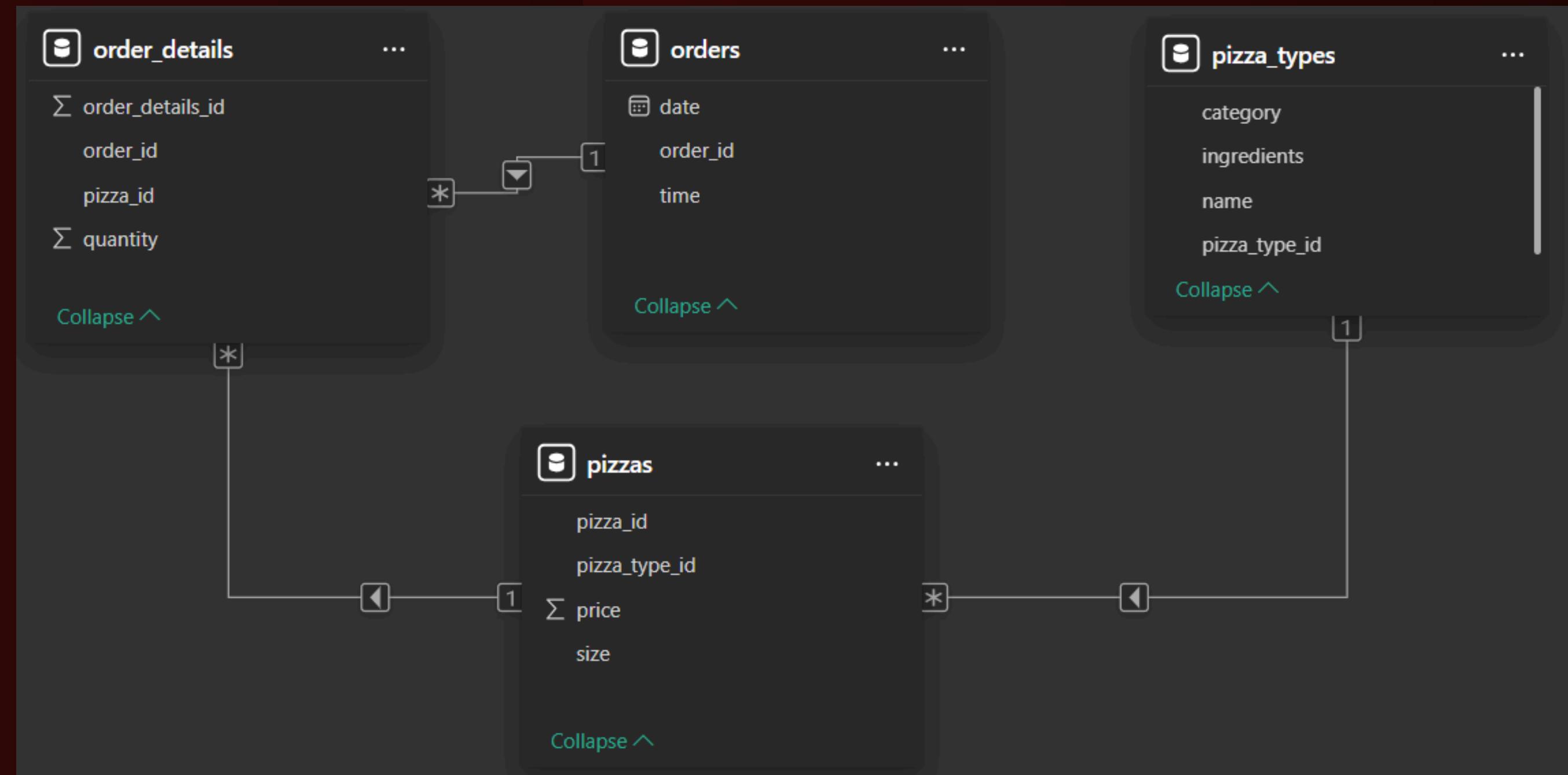
[Home](#)

[About](#)

[Contact](#)



SALES DATASETS





ABOUT PROJECT

This project is developed on MySQL. In this data analytics project , I applied a wide range of queries for data retrieval varying from simple, medium to complex ones.

The aim of this study was to practice and find out some general but necessary insights needed by a pizza selling association to maximize its profits.

RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED

```
3 ● select count(order_id) from orders;
```

Result Grid	
	count(order_id)
▶	21350



CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES

```
7 •   select round(sum( order_details.quantity * pizzas.price), 2)
8     as total_revenue
9     from order_details join pizzas
10    on pizzas.pizza_id = order_details.pizza_id;
```

Result Grid	
	total_revenue
▶	817860.05



IDENTIFY THE HIGHEST-PRICED PIZZA

```
14 • select pizza_types.name, pizzas.price
15   from pizza_types join pizzas
16     on pizza_types.pizza_type_id = pizzas.pizza_type_id
17   order by pizzas.price desc limit 1;
```

Result Grid		Filter Row
	name	price
▶	The Greek Pizza	35.95



IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED

```
21 •   select pizzas.size,  
22       count(order_details.order_details_id) as order_count  
23   from pizzas join order_details  
24     on pizzas.pizza_id = order_details.pizza_id  
25   group by pizzas.size  
26   order by order_count desc;
```

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



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

```
31 • select pizza_types.name,  
32     sum(order_details.quantity) as Quantity  
33     from pizza_types join pizzas  
34         on pizza_types.pizza_type_id = pizzas.pizza_type_id  
35     join order_details  
36         on order_details.pizza_id = pizzas.pizza_id  
37     group by pizza_types.name  
38     order by quantity desc limit 5;
```

	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



FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED

```
3 •   select pizza_types.category,  
4       sum(order_details.quantity) as quantity  
5       from pizza_types join pizzas  
6           on pizza_types.pizza_type_id = pizzas.pizza_type_id  
7       join order_details  
8           on order_details.pizza_id = pizzas.pizza_id  
9       group by pizza_types.category  
10      order by quantity desc;
```

Result Grid		
	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050



DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY

```
14 • select hour(order_time) as hour,  
15   count(order_id) as order_count  
16   from orders  
17   group by hour;
```

	hour	order_count
▶	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



FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS

```
21 •     select category, count(name) as 'number of pizzas'  
22      from pizza_types  
23      group by category;
```

Result Grid | Filter Rows:

	category	number of pizzas
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9



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

```
28 •     select round(avg(quantity),0) as 'avg pizza ordered per day'  
29      from  
30      (select orders.order_date,  
31          sum(order_details.quantity) as quantity  
32      from orders join order_details  
33      on orders.order_id = order_details.order_id  
34      group by orders.order_date) as order_quantity;
```

	Result Grid	Filter Row
	avg pizza ordered per day	
▶	138	



DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE

```
39 •   select pizza_types.name,  
40       sum(order_details.quantity * pizzas.price) as revenue  
41   from pizza_types join pizzas  
42     on pizza_types.pizza_type_id = pizzas.pizza_type_id  
43   join order_details  
44     on order_details.pizza_id = pizzas.pizza_id  
45   group by pizza_types.name  
46   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



CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE

```
4 •   select pizza_types.category,  
5     round(sum(order_details.quantity * pizzas.price) /  
6     (select round(sum(order_details.quantity * pizzas.price), 2)  
7     as total_revenue  
8     from order_details join pizzas  
9     on pizzas.pizza_id = order_details.pizza_id) * 100, 2)  
10    as revenue  
11    from pizza_types join pizzas  
12    on pizza_types.pizza_type_id = pizzas.pizza_type_id  
13    join order_details  
14    on order_details.pizza_id = pizzas.pizza_id  
15    group by pizza_types.category  
16    order by revenue desc;
```

	category	revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68



ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME

```
20 ● select order_date,  
21     sum(revenue) over(order by order_date) as cum_revenue  
22     from  
23     (select orders.order_date,  
24         sum(order_details.quantity * pizzas.price) as revenue  
25         from order_details join pizzas  
26             on order_details.pizza_id = pizzas.pizza_id  
27         join orders  
28             on orders.order_id = order_details.order_id  
29         group by orders.order_date) as sales;
```

	order_date	cum_revenue
▶	2015-01-01	2713.850000000004
	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	23990.35000000002
	2015-01-11	25862.65
	2015-01-12	27781.7
	2015-01-13	29831.30000000003



DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY

```
34 ●    select category, name, revenue
35      from
36      (select category, name, revenue,
37          rank()
38          over(partition by category order by revenue desc) as rn
39      from
40      (select pizza_types.name,
41      pizza_types.category,
42      sum(order_details.quantity * pizzas.price) as revenue
43      from order_details join pizzas
44      on order_details.pizza_id = pizzas.pizza_id
45      join pizza_types
46      on pizza_types.pizza_type_id = pizzas.pizza_type_id
47      group by pizza_types.name, pizza_types.category)
48      as a) as b
49      where rn <= 3;
```

category	name	revenue
Chicken	The Thai Chicken Pizza	43434.25
Chicken	The Barbecue Chicken Pizza	42768
Chicken	The California Chicken Pizza	41409.5
Classic	The Classic Deluxe Pizza	38180.5
Classic	The Hawaiian Pizza	32273.25
Classic	The Pepperoni Pizza	30161.75
Supreme	The Spicy Italian Pizza	34831.25
Supreme	The Italian Supreme Pizza	33476.75
Supreme	The Sicilian Pizza	30940.5
Veggie	The Four Cheese Pizza	32265.7000000065
Veggie	The Mexicana Pizza	26780.75
Veggie	The Five Cheese Pizza	26066.5



2025 PIZZA RESTO PRESENTATION

THANK
YOU FOR
ATTENTION