

Where Every Slice is a Taste of Perfection

WELCOME TO PIZZA RESTO

ORDER
NOW

Start Your Slide





HELLO !

My name is SHAMAL

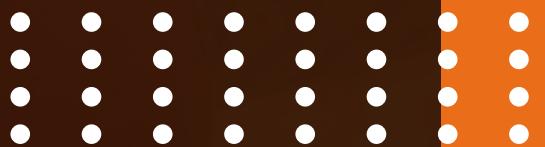
This project aims to analyze sales data from a pizza restaurant, such as Pizza Hut, using SQL. By querying a structured database containing information on orders, customers, menu items, and sales trends, we will extract valuable business insights to enhance decision-making.

Objectives:

- Understand Customer Purchasing Behavior.
- Identify Best-Selling and Least-Popular Items.
- Analyze Sales Trends.
- Optimize Inventory Management
- Enhance Business Strategies with Data-Driven Decisions.



RELATIONS



Order Details

2	2	classic_dlx_m	1
3	2	five_cheese_l	1
4	2	ital_supr_l	1
5	2	mexicana_m	1
6	2	thai_ckn_l	1
7	3	ital_supr_m	1
8	3	prsc_argla_l	1
9	4	ital_supr_m	1
10	5	ital_supr_m	1

Orders

order_id	order_date	order_time
1	2015-01-01	11:38:36
2	2015-01-01	11:57:40
3	2015-01-01	12:12:28
4	2015-01-01	12:16:31
5	2015-01-01	12:21:30

pizzas_types

pizza_type_id	name	category	ingredients
bbq_ckn	The Barbecue Chicken Pizza	Chicken	Barbecued Chicken, Red Peppers, Onions
cali_dkn	The California Chicken Pizza	Chicken	Chicken, Artichoke, Spinach, Garlic, Tomatoes
dkn_alfredo	The Chicken Alfredo Pizza	Chicken	Chicken, Red Onions, Red Peppers, Alfredo Sauce
dkn_pesto	The Chicken Pesto Pizza	Chicken	Chicken, Tomatoes, Red Peppers, Pesto
southw_ckn	The Southwest Chicken Pizza	Chicken	Chicken, Tomatoes, Red Peppers, Pineapple, Lettuce
thai_dkn	The Thai Chicken Pizza	Chicken	Chicken, Pineapple, Tomatoes, Red Peppers, Lettuce
big_meat	The Big Meat Pizza	Classic	Bacon, Pepperoni, Italian Sausage, Mushrooms, Onions
classic_dlx	The Classic Deluxe Pizza	Classic	Pepperoni, Mushrooms, Red Onions, Green Peppers, Lettuce
hawaiian	The Hawaiian Pizza	Classic	Sliced Ham, Pineapple, Mozzarella Cheese
ital_cpclo	The Italian Capocollo Pizza	Classic	Capocollo, Red Peppers, Tomatoes, Lettuce
napolitana	The Napolitana Pizza	Classic	Tomatoes, Anchovies, Green Olives, Lettuce
pep_msh_pep	The Pepperoni, Mushroom, ...	Classic	Pepperoni, Mushrooms, Green Peppers, Lettuce

Pizzas

order_id	order_date	order_time
1	2015-01-01	11:38:36
2	2015-01-01	11:57:40
3	2015-01-01	12:12:28
4	2015-01-01	12:16:31
5	2015-01-01	12:21:30
6	2015-01-01	12:29:36

RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED



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

Result Grid	
	total_orders
▶	21350

CALCULATED THE TOTAL REVENUE GENERATED FROM PIZZA SALES



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

Result Grid	
	total_sales
→	817860.05

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 R

	name	price
▶	The Greek Pizza	35.95

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED

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

Result Grid | Filter Row

	size	order_count
▶	L	18526
	M	15385

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

```
SELECT
    pizza_types.name, SUM(oder_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    oder_details ON oder_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
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

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

```
SELECT
    pizza_types.category, SUM(oder_details.quantity) AS quantity
FROM
    pizza_types
    JOIN
        pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_ids
    JOIN
        oder_details ON oder_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY quantity DESC;
```

Result Grid | Filter Rows:

	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

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

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

Result Grid | Filter Rows:

	hour(order_time)	order_count
20	1642	
21	1198	
22	663	
23	28	
10	8	
9	1	

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

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

Result Grid | Filter Rows:

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

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

- **SELECT**

```
ROUND(AVG(quantity), 0) as avg_pizza_ordered_per_day  
FROM  
(SELECT  
    orders.order_date, SUM(oder_details.quantity) AS quantity  
FROM  
    orders  
JOIN oder_details ON orders.order_id = oder_details.order_id  
GROUP BY orders.order_date) AS order_quantity;
```

Result Grid | Filter Rows:

	avg_pizza_ordered_per_day
▶	138

DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE

```
select pizza_types.name,  
sum(oder_details.quantity * pizzas.price) as revenue  
from pizza_types join pizzas  
on pizzas.pizza_type_id = pizza_types.pizza_type_id  
join oder_details  
on oder_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

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

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

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(oder_details.quantity * pizzas.price) as revenue
 from oder_details join pizzas
 on oder_details.pizza_id = pizzas.pizza_id
 join orders
 on orders.order_id = oder_details.order_id
 group by orders.order_date) as sales;
```

Result Grid | Filter Rows:

|   | 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           |

# 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((oder_details.quantity) * pizzas.price) as revenue|
 from pizza_types join pizzas
 on pizza_types.pizza_type_id = pizzas.pizza_type_id
 join oder_details
 on oder_details.pizza_id = pizzas.pizza_id
 group by pizza_types.category, pizza_types.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 |

Pizza Resto Presentation

THANK YOU  
FOR ATTENTION

See You Next