

Tasty Spicy 

SQL PROJECT ON

**PIZZA SALES** 



**ORDER NOW**





LOGO HERE

# HELLO MY NAME IS NIKITA MESHRAM.

IN THIS PROJECT I HAVE  
UTILIZED SQL QUERIES TO  
SOLVE QUESTIONS THAT WERE  
RELATED TO PIZZA SALES



**ORDER NOW**

- -- 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.
- -- 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

-- RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

```
4 ● SELECT
5     COUNT(order_id) total_orders
6 FROM
7     orders;
```

	total_orders
▶	21350

-- CALCULATE THE TOTAL REVENUE GENERATED  
FROM PIZZA SALES.

```
4 ● SELECT
5   ROUND(SUM(order_details.quantity * pizzas.price),
6         2) AS total_sales
7 FROM
8   order_details
9   JOIN
10  pizzas ON pizzas.pizza_id = order_details.pizza_id
```

	total_sales
▶	817860.05

-- IDENTIFY THE HIGHEST-PRICED PIZZA.

```
3 ● SELECT
4     pizza_types.name, pizzas.price
5 FROM
6     pizza_types
7     JOIN
8     pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
9 ORDER BY pizzas.price DESC
10 LIMIT 1;
```

	name	price
▶	The Greek Pizza	35.95

-- IDENTIFY THE MOST COMMON PIZZA SIZE  
ORDERED.

```
3 ● SELECT
4     pizzas.size,
5     COUNT(order_details.order_detail_id) AS order_count
6 FROM
7     pizzas
8     JOIN
9     order_details ON pizzas.pizza_id = order_details.pizza_id
10 GROUP BY pizzas.size
11 ORDER BY order_count DESC
12 LIMIT 1;
```

	size	order_count
▶	L	18526

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

```
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;
```

	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.

```
3 ● SELECT
4     pizza_types.category,
5     SUM(order_details.quantity) AS total_quantity
6 FROM
7     pizza_types
8     JOIN
9     pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
10    JOIN
11    order_details ON pizzas.pizza_id = order_details.pizza_id
12 GROUP BY pizza_types.category;
```

	category	total_quantity
▶	Classic	14888
	Veggie	11649
	Supreme	11987
	Chicken	11050

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

```
3  ●  SELECT
4      HOUR(order_time), COUNT(order_id)
5  FROM
6      orders
7  GROUP BY HOUR(order_time);
```

	HOUR(order_time)	COUNT(order_id)
▶	11	1231
	12	2520
	13	2455
	14	1472
	15	1468
	16	1920
	17	2336
	18	2399
	19	2009

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

```
3 ● SELECT
4     category, COUNT(name)
5 FROM
6     pizza_types
7 GROUP BY category;
```

	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.

```
3  •  SELECT
4      AVG(quantity)
5  FROM
6  (SELECT
7      orders.order_date, SUM(order_details.quantity) AS quantity
8  FROM
9      orders
10     JOIN order_details ON orders.order_id = order_details.order_id
11     GROUP BY orders.order_date) AS oredr_quantity;
```

	AVG(quantity)
▶	138.4749

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

```
3 • SELECT
4     pizza_types.name,
5     SUM(order_details.quantity * pizzas.price) AS revenue
6 FROM
7     pizza_types
8     JOIN
9     pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
10    JOIN
11    order_details ON order_details.pizza_id = pizzas.pizza_id
12 GROUP BY pizza_types.name
13 ORDER BY revenue DESC
14 LIMIT 3;
```

	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(order_details.quantity * pizzas.price) / (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;
```

	category	revenue
	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68



-- ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

```
3 • select order_date,  
4    sum(revenue) over (order by order_date) as cum_revenue  
5    from  
6    (select orders.order_date,  
7     sum(order_details.quantity * pizzas.price) as revenue  
8     from order_details join pizzas  
9     on order_details.pizza_id = pizzas.pizza_id  
10    join orders  
11    on orders.order_id = order_details.order_id  
12    group by orders.order_date) as sales;  
13
```

	order_date	cum_revenue
	2015-01-01	2713.85000000000004
	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	23990.3500000000002

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

```
3 • select name, revenue from
4   (select category, name, revenue,
5    rank() over (partition by category order by revenue desc) as rn
6   from
7    (select pizza_types.category, pizza_types.name,
8     sum((order_details.quantity) * pizzas.price) as revenue
9    from pizza_types join pizzas
10     on pizza_types.pizza_type_id = pizzas.pizza_type_id
11    join order_details
12     on order_details.pizza_id = pizzas.pizza_id
13   group by pizza_types.category, pizza_types.name) as a) as b
14  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 Supreme Pizza	32473.25