# SQLPROJECT ON PIZZAHUT SALES



50%

**OFF** 

ORDER NOW!

### **HELLO**

# MY NAME IS PRATHAM SRIVASTAVA IN THIS PROJECT I HAVE UTILIZED SQL QUERIESTO SOLVE A QUESTIONS RELATED TO PIZZAHUT SALES

#### **QUESTIONS**

#### 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.

#### 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.

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.

#### 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.

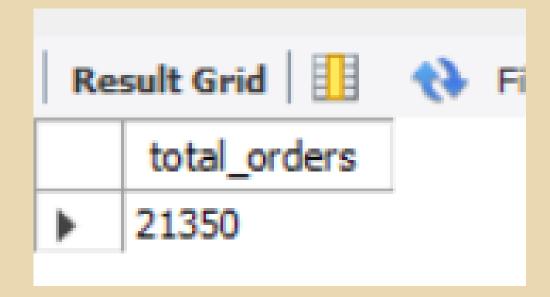
## RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED

```
SELECT

COUNT(order_id) AS total_orders

FROM

orders;
```



## CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES

```
SELECT

ROUND(SUM(orders_details.quantity * pizzas.price),

2) AS total_sales

FROM

orders_details

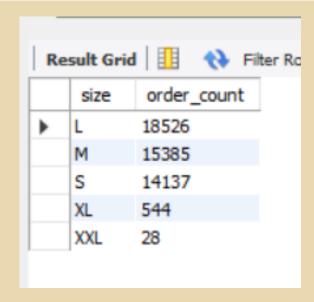
JOIN

pizzas ON pizzas.pizza_id = orders_details.pizza_id;
```

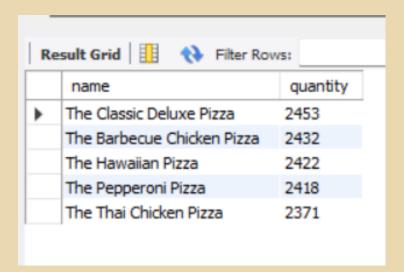
## IDENTIFY THE HIGHEST-PRICED PIZZA



## 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

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

Re	sult Grid	Filte	er Rows:
	category	quantity	
•	Classic	14888	
	Supreme	11987	
	Veggie	11649	
	Chicken	11050	

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

```
SELECT

HOUR(order_time) AS hour, COUNT(order_id) AS order_count

FROM

orders

GROUP BY HOUR(order_time);
```

Re	sult Grid	l 🔢 🙌 Filter R
	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
	9	1

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

```
category, COUNT(name)

FROM

pizza_types

GROUP BY category;
```

Re	sult 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 PIZZAS ORDERED PER DAY

```
SELECT

ROUND(AVG(quantity), 0) as avg_pizza_order_perday

FROM

(SELECT

orders.order_date, SUM(orders_details.quantity) AS quantity

FROM

orders

JOIN orders_details ON orders.order_id = orders_details.order_id

GROUP BY orders.order_date) AS order_quantity;
```

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

```
SELECT
    pizza_types.name,
    SUM(orders_details.quantity * pizzas.price) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY revenue DESC
LIMIT 3;
```

Result Grid		
	name	revenue
•	The Thai Chicken Pizza	43434.25
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41409.5
	The California Chicken Pizza	41409.5

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

```
SELECT

pizza_types.category,

ROUND(SUM(orders_details.quantity * pizzas.price) / (SELECT

ROUND(SUM(orders_details.quantity * pizzas.price),

2) AS total_sales

FROM

orders_details

JOIN

pizzas ON pizzas.pizza_id = orders_details.pizza_id) * 100,

2) AS revenue

FROM

pizza_types

JOIN

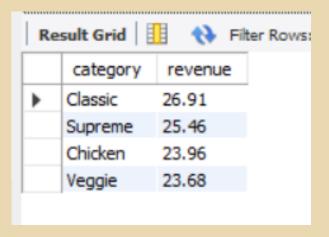
pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id

JOIN

orders_details ON orders_details.pizza_id = pizzas.pizza_tyde_id

GROUP BY pizza_types.category

ORDER BY revenue DESC;
```



## ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME

Re	esult Grid	National Company of the Printer 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
	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
-	li a	

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

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	32265.70000000065
The Mexicana Pizza	26780.75
The Five Cheese Pizza	26066.5