PIZZA SALES REPORT

INTRODUCTION

Welcome to the Pizza Sales Report presentation, where we delve into the analysis of pizza sales data using SQL. This comprehensive report provides insights into various aspects of our pizza business, from basic metrics to advanced analytics.

OBJECTIVE

The objective of this project is to analyze pizza sales data using SQL to extract valuable insights that can aid in business decision-making. By querying the data, we aim to answer several key questions related to sales performance, customer preferences, and revenue generation.

ORDERS TABLE

This table contains information about each order placed, including the order ID, order date, and order time. It helps in tracking when orders were placed and how many orders were made over time.

ORDER_DETAILS TABLE

This table provides detailed information about each pizza ordered, including the order ID, pizza ID, quantity, and price. It helps in understanding the specifics of each order, such as the number of pizzas and the total price.

PIZZAS TABLE

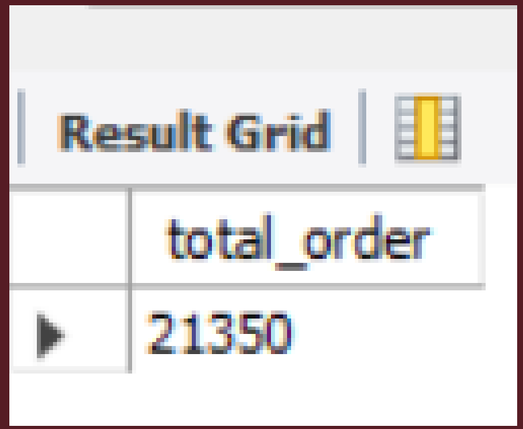
This table lists all available pizzas along with their ID, name, size, and price. It is essential for identifying the types of pizzas available, their sizes, and their pricing.

PIZZA_TYPES TABLE

This table categorizes pizzas into various types and categories, including the pizza type ID, name, and category. It helps in grouping pizzas into broader categories for more generalized analysis.

Retrieve the total number of orders placed.

```sql
SELECT
    COUNT(order_id) AS total_order
FROM
    orders;
```



Calculate the total revenue generated from pizza sales.

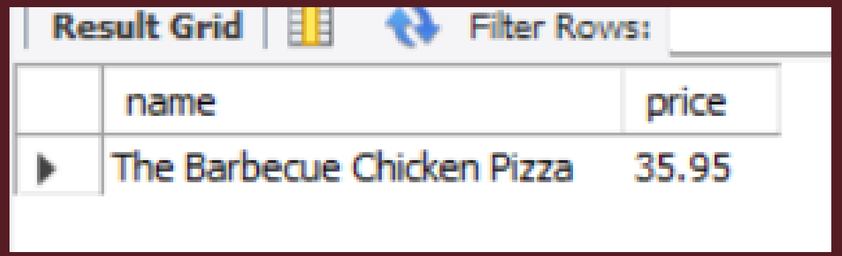
```sql
SELECT
    round(sum(order_details.quantity * pizzas.price),2) AS total_sales
FROM
    order_details
        JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id;
```

Result Grid

	total_sales
▶	817860.05

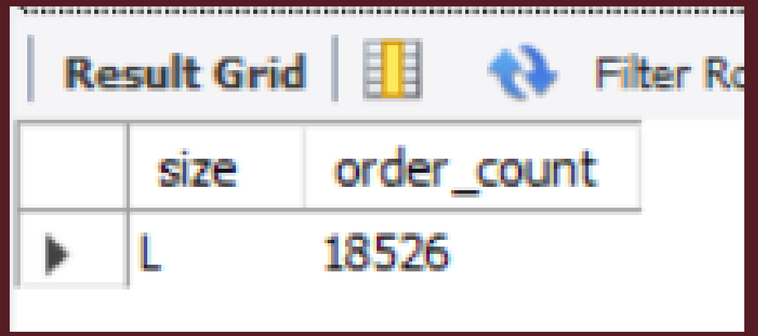
Identify the highest-priced pizza.

```sql
SELECT
    pizza_types.name, pizzas.price
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizza_types.pizza_type_id
ORDER BY pizzas.price DESC
LIMIT 1
```



Identify the most common pizza size ordereders placed.

```sql
SELECT
    pizzas.size,
    COUNT(order_details.order_details_id) AS order_count
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY order_count DESC
LIMIT 1 ;
```



	size	order_count
▶	L	18526

List the top 5 most ordered pizza types along with their quantities.

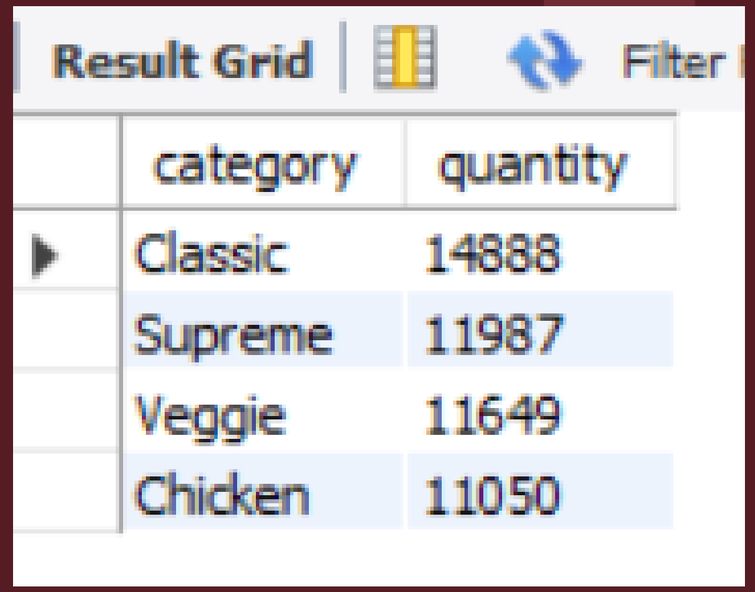
```sql
SELECT pizza_types.name, SUM(order_details.quantity)
AS quantity FROM pizza_types JOIN pizzas ON
pizza_types.pizza_type_id = pizzas.pizza_type_id
JOIN order_details ON
pizzas.pizza_id = order_details.pizza_id
GROUP BY pizza_types.name
ORDER BY quantity DESC
LIMIT 5;
```

Result Grid | Filter Rows:

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.

```sql
SELECT pizza_types.category,
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
pizzas.pizza_id = order_details.pizza_id
GROUP BY pizza_types.category
ORDER BY quantity DESC;
```



category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Determine the distribution of orders by hour of the day.

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

hour(order_time)	count(order_id)
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.

```sql
SELECT category, COUNT(category) AS count
FROM pizza_types
GROUP BY category
ORDER BY count DESC;
```

Result Grid

	category	count
▶	Supreme	9
	Veggie	9
	Classic	8
	Chicken	6

Group the orders by date and calculate the average number of pizzas ordered per day.

```
SELECT ROUND(AVG(per_day), 0) AS avg
FROM (SELECT SUM(order_details.quantity)
AS per_day, orders.order_date FROM
order_details JOIN orders ON
order_details.order_id = orders.order_id
GROUP BY orders.order_date) AS order_quantity;
```

Result Grid

	avg
▶	138

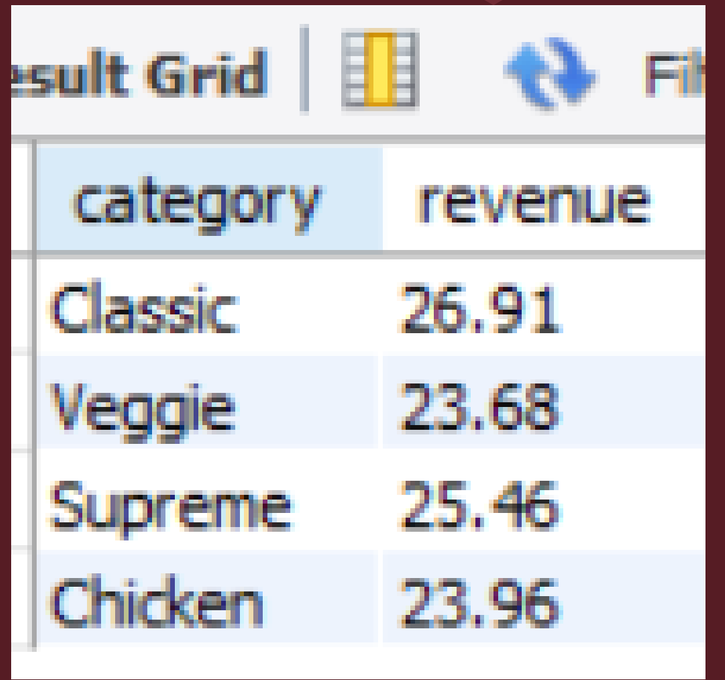
Determine the top 3 most ordered pizza types based on revenue.

```sql
SELECT pizza_types.name,
SUM(order_details.quantity * pizzas.price)
 AS revenu FROM order_details JOIN pizzas
 ON order_details.pizza_id = pizzas.pizza_id
JOIN pizza_types ON
pizzas.pizza_type_id = pizza_types.pizza_type_id
GROUP BY pizza_types.name
ORDER BY revenue DESC 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.

```sql
SELECT pizza_types.category,
ROUND(SUM(pizzas.price * order_details.quantity) / (SELECT
ROUND(SUM(order_details.quantity * pizzas.price),2) AS
total_revenue FROM order_details JOIN pizzas ON
order_details.pizza_id = pizzas.pizza_id) * 100,
2) AS revenue FROM order_details JOIN pizzas
ON order_details.pizza_id = pizzas.pizza_id JOIN
pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id
GROUP BY pizza_types.category;
```



esult Grid | Fil

category	revenue
Classic	26.91
Veggie	23.68
Supreme	25.46
Chicken	23.96

Analyze the cumulative revenue generated over time.

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

order_date	revenue	cum_revenue
2015-01-01 00:00:00	2713.8500000000004	2713.8500000000004
2015-01-02 00:00:00	2731.8999999999996	5445.75
2015-01-03 00:00:00	2662.3999999999996	8108.15
2015-01-04 00:00:00	1755.4500000000003	9863.6
2015-01-05 00:00:00	2065.95	11929.55
2015-01-06 00:00:00	2428.95	14358.5
2015-01-07 00:00:00	2202.2000000000003	16560.7
2015-01-08 00:00:00	2838.3499999999995	19399.05
2015-01-09 00:00:00	2127.3500000000004	21526.4
2015-01-10 00:00:00	2463.95	23990.350000000002

Determine the top 3 most ordered pizza types based on revenue for each pizza category.

```sql
select category,name,revenue ,rn
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(order_details.quantity*pizzas.price) as revenue 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 pizza_types.category,pizza_types.name) as a ) as b
 where rn<=3 ;
```

Filter Rows: Export:

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

CONCLUSION

This project utilizes SQL queries to provide a comprehensive analysis of pizza sales data. By examining various aspects of sales performance and customer preferences, the analysis offers valuable insights that can help drive business decisions and strategies. From basic metrics like total orders and revenue to more advanced analyses like revenue contributions and order distributions, the project delivers a detailed understanding of the pizza sales landscape.

THANK YOU