Where Every Slice is a Taste of Perfection



Start Your Slide

About this

This SQL presentation leverages extensive datasets, including tables like PizzaTypes, Orders, Pizza, and Order_Details, to demonstrate complex queries. It covers essential SQL operations such as data retrieval, filtering, aggregation, and joins to analyze customer orders, pizza types, and order details. The queries explore various scenarios, such as identifying popular pizza choices, tracking order statuses, and calculating total sales, offering a practical application of SQL skills in the context of a pizza business.


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



Calculate the total revenue generated from pizza sales

```
SELECT

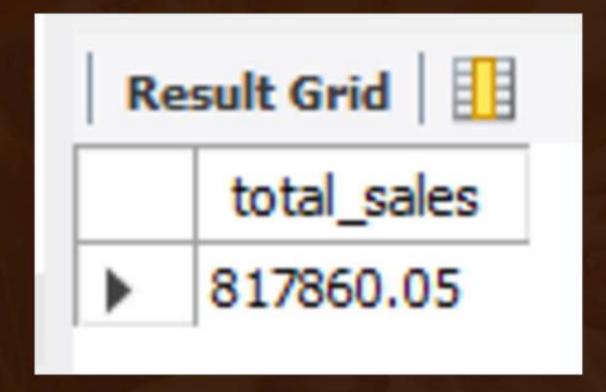
ROUND(SUM(order_details.quantity*pizzas.price),2) AS total_sales

FROM

order_details

JOIN

pizzas ON pizzas.pizza_id =order_details.pizza_id;
```





Identify the highest priced pizza.

```
pizza_types.name,pizzas.price

FROM

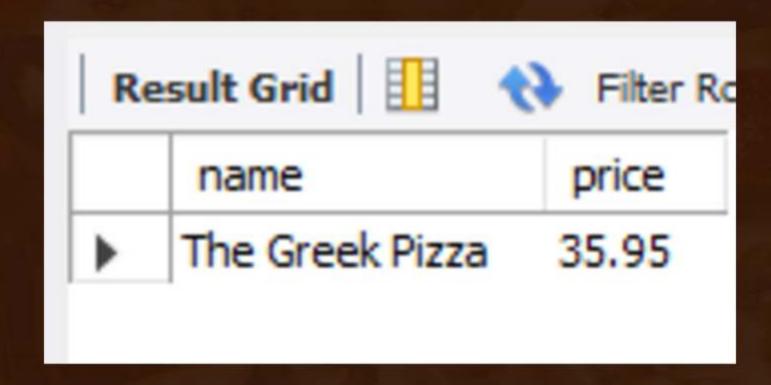
pizza_types

JOIN

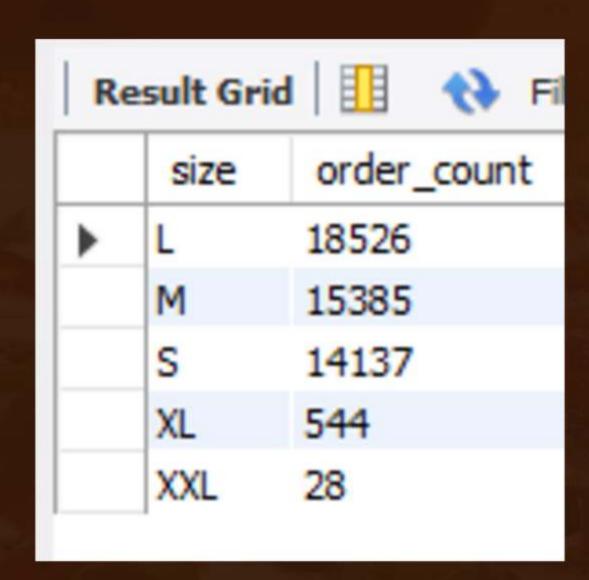
pizzas ON pizza_types.pizza_type_id=pizzas.pizza_type_id

ORDER BY pizzas.price DESC

LIMIT 1;
```



Identify the most common pizza size ordered.



List the top 5 most ordered pizza types along with their quantity

```
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;
```

esult Grid Filter Row	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 necessarry tables to find thetotal quantity of each pizza catagory ordered.

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

Re	Result Grid Fi	
	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);
```

nour	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 relevent tables to find the catagory-wise distribution of pizzas

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

	category	COUNT(name)	
	Chicken	6	
L	Classic	8	
	Supreme	9	
	Veggie	9	
Г			

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

```
SELECT

ROUND(AVG(quantity),0) AS avg_pizza_ordered_per_Day

FROM

(SELECT orders.order_date,SUM(order_details.quantity) AS quantity

FROM

orders

JOIN order_details ON orders.order_id=order_details.order_id

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

```
avg_pizza_ordered_per_Day

138
```

Determine the top 3 most ordered pizza types based on revenue

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

analyze the cumulative revenue generated over time.

Result Grid		Thiter 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

Pizza Resto Presentation

THANK YOU FOR ATTENTION

See You Next

www.reallygreatsite.com