



PIZZA SALES





PROJECT

Hi, I am Vishwa. In this project, i have used SQL query to analyze the data related to pizza sales.

The required SQL query and their result are in the next slides.

1. RETRIEVE NUMBER OF ORDERS PLACED

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

Result Grid	
	total_orders
▶	21350



2. CALCULATE TOTAL REVENUE GENERATED BY PIZZA SALES

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

Result Grid	
	Total_revenue
▶	817860.05

A close-up photograph of a person's hand holding a wooden spoon and stirring a pan of pasta. The pasta appears to be fettuccine, mixed with cherry tomatoes and some herbs or meat. The pan is dark-colored.

3. IDENTIFY THE HIGHEST PRICE PIZZA

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

Result Grid |  Filter Rows:

	name	size	price
▶	The Greek Pizza	XXL	35.95

4. IDENTIFY THE MOST COMMON ORDERED PIZZA SIZE

```
SELECT  
    size, COUNT(order_details_id)  
FROM  
    order_details  
    JOIN  
    pizzas ON order_details.pizza_id = pizzas.pizza_id  
GROUP BY size;
```

	size	COUNT(order_details_id)
▶	M	15385
	L	18526
	S	14137
	XL	544

```
SELECT
    name, SUM(quantity) AS quant
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 quant DESC
LIMIT 5;
```

	name	quant
▶	The Classic Deluxe Pizza	2453
	The Barbecue Chicken Pizza	2432
	The Hawaiian Pizza	2422
	The Pepperoni Pizza	2418
	The Thai Chicken Pizza	2371

5. LIST THE TOP 5 MOST ORDER PIZZA TYPES WITH THEIR QUANTITY

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

```
SELECT
    pizza_types.category AS Category,
    SUM(order_details.quantity) AS Total_quantity
FROM pizzas
    JOIN order_details ON
        pizzas.pizza_id = order_details.pizza_id
    JOIN pizza_types ON
        pizzas.pizza_type_id = pizza_types.pizza_type_id
GROUP BY Category;
```

	Category	Total_quantity
▶	Classic	14888
	Veggie	11649
	Supreme	11987
	Chicken	11050





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

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

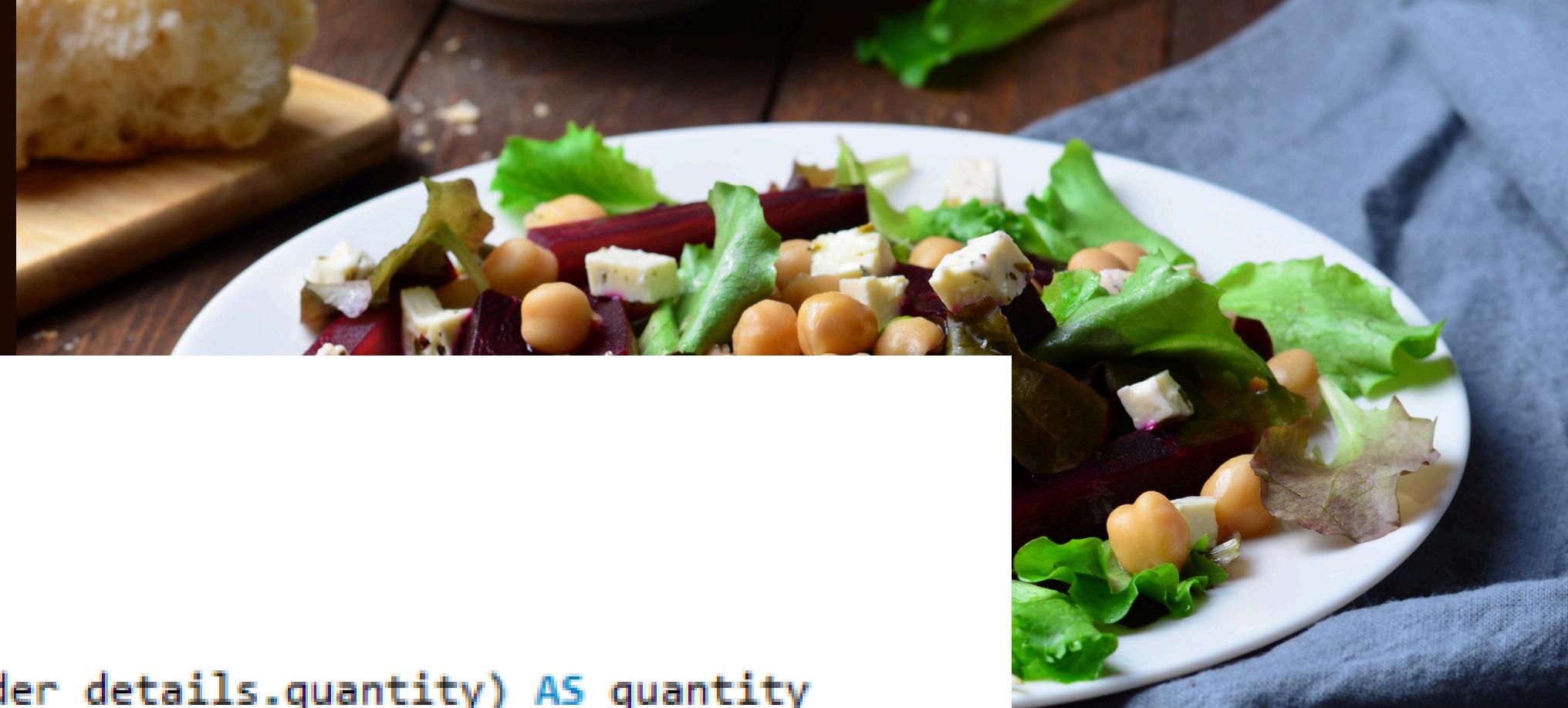


	HOUR(orders.time)	COUNT(order_id)
▶	11	1231
	12	2520
	13	2455
	14	1472
	15	1468
	16	1920
	17	2336
	18	2399
	19	2009
	20	1642
	21	1198
	22	663
	23	28
	10	8
	9	1

8. GROUP THE ORDER BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY

```
SELECT  
    AVG(quantity)  
FROM  
(SELECT  
    orders.date, SUM(order_details.quantity) AS quantity  
FROM  
    orders  
JOIN order_details ON orders.order_id = order_details.order_id  
GROUP BY orders.date) AS subquery;
```

AVG(quantity)
138.4749



9. TOP 3 MOST ORDERED PIZZA BASED ON REVENUE

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

10. CALCULATE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPES TO TOTAL REVENUE

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

	category	percent_rev
	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

ANALYZE THE CUMMULAT IVE REVENUE GENERATED OVER TIME

```
select date ,sum(revenue) over(order by date) as Cum_revenue from  
(select orders.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 orders.order_id=order_details.order_id  
group by orders.date) as b;
```

	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