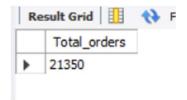
# Pizza Sales Analysis:

1. Retrieve the total number of orders placed.

```
-- Retrieve the total number of orders placed.
select count(order_id) as Total_orders from orders;
```

#### Result



2 Calculate the total revenue generated from pizza sales.

```
-- Calculate the total revenue generated from pizza sales.
```

```
SELECT

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

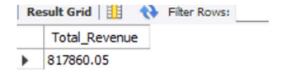
2) AS Total_Revenue

FROM

order_details

JOIN pizzas

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



3. Identify the highest-priced pizza.

```
-- Identify the highest-priced pizza.
 2
 3 •
       SELECT
 4
           pizza_types.name, pizzas.price
       FROM
 5
 6
           pizza_types
 7
                JOIN
 8
           pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
       ORDER BY pizzas.price DESC
 9
10
       LIMIT 1;
11
```

## result



4. Identify the most common pizza size ordered.

```
-- Identify the most common pizza size ordered.

SELECT

pizzas.size,

COUNT(order_details.order_details_id) AS Pizza_count

FROM

pizzas

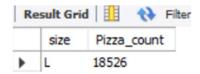
JOIN

order_details ON pizzas.pizza_id = order_details.pizza_id

GROUP BY pizzas.size

ORDER BY pizza_count DESC

LIMIT 1;
```



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

```
-- 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;
```

## result

	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

6. Join the necessary tables to find the total quantity of each pizza category ordered.

```
-- Join the necessary tables to find the total quantity of each pizza category ordered

SELECT

pizza_types.category, SUM(order_details.quantity) AS quanity

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 quanity DESC;
```

```
7. Determine the distribution of orders by hour of the day.
```

```
-- determine the distribution of orders by hour of the day.

SELECT

HOUR(order_time) AS hours, COUNT(order_id) AS order_cout

FROM

orders

GROUP BY hours;
```

# Result

	hours	order_cout
	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. Join relevant tables to find the category-wise distribution of pizzas.

```
-- Join relevant tables to find the category-wise distribution of pizzas.

SELECT
category, COUNT(name)

FROM
pizza_types

GROUP BY category;
```

## Result

	category	count(name)
•	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

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

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

SELECT

AVG(quantity) avg_per_day

FROM

(SELECT

(order_date) AS order_date,

SUM(order_details.quantity) AS quantity

FROM

orders

JOIN order_details ON order_details.order_id = orders.order_id

GROUP BY order_date) AS order_quantity;
```

# Result

```
avg_per_day

▶ 138.4749
```

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

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

SELECT

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 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

11. Calculate the percentage contribution of each pizza type to total 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;
```

#### result



12 . Analyze the cumulative revenue generated over time.

```
-- Analyze the cumulative revenue generated over time.

select order_date, sum(revenue) over (order by order_date) as cumlative_rev

from

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

order_	date	cumlative_rev
2015-0	1-01 00:00:00	2713.8500000000004
2015-0	1-02 00:00:00	5445.75
2015-0	1-03 00:00:00	8108.15
2015-0	1-04 00:00:00	9863.6
2015-0	1-05 00:00:00	11929.55
2015-0	1-06 00:00:00	14358.5
2015-0	1-07 00:00:00	16560.7

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

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

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 order_details.pizza_id = pizzas.pizza_id

group by pizza_types.category, pizza_types.name) as a) as b

where rn <=3;
```

category	name	revenue
Supreme	The Spicy Italian Pizza	34831.25
Supreme	The Italian Supreme Pizza	33476.75
Supreme	The Sicilian Pizza	30940.5
Veggie	The Four Cheese Pizza	32265.70000000065
Veggie	The Mexicana Pizza	26780.75
Veggie	The Five Cheese Pizza	26066.5