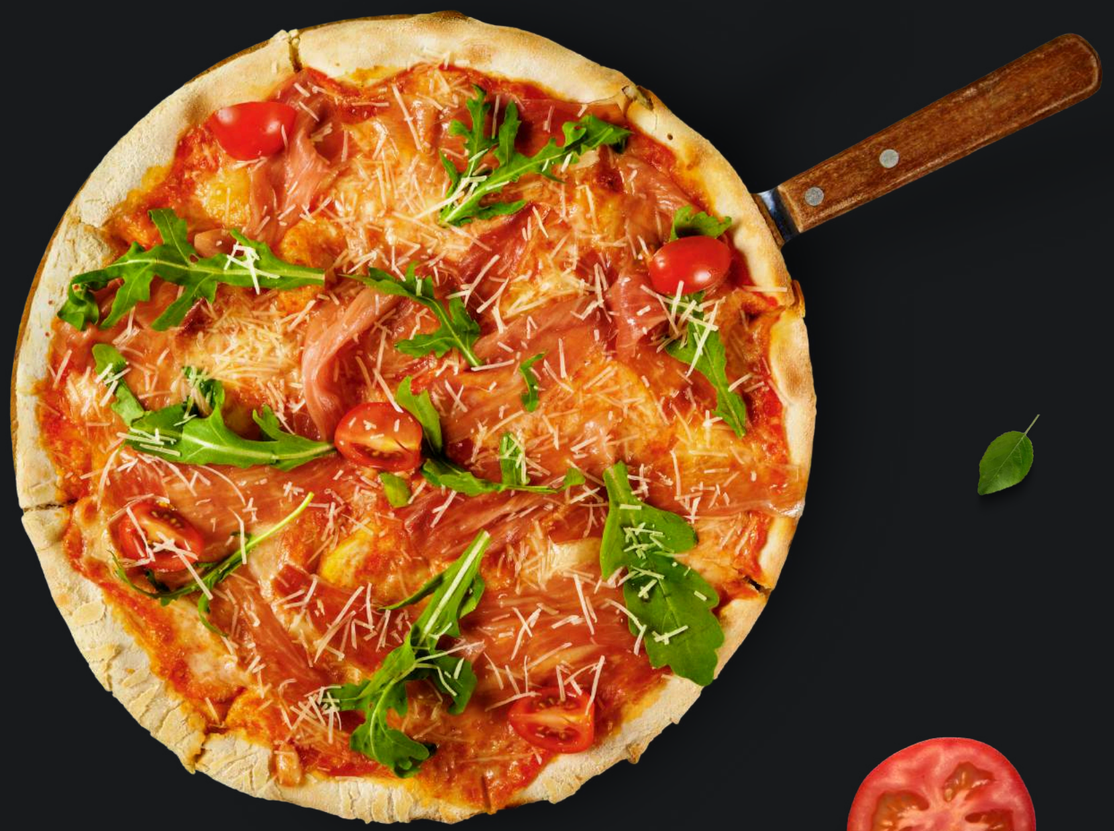
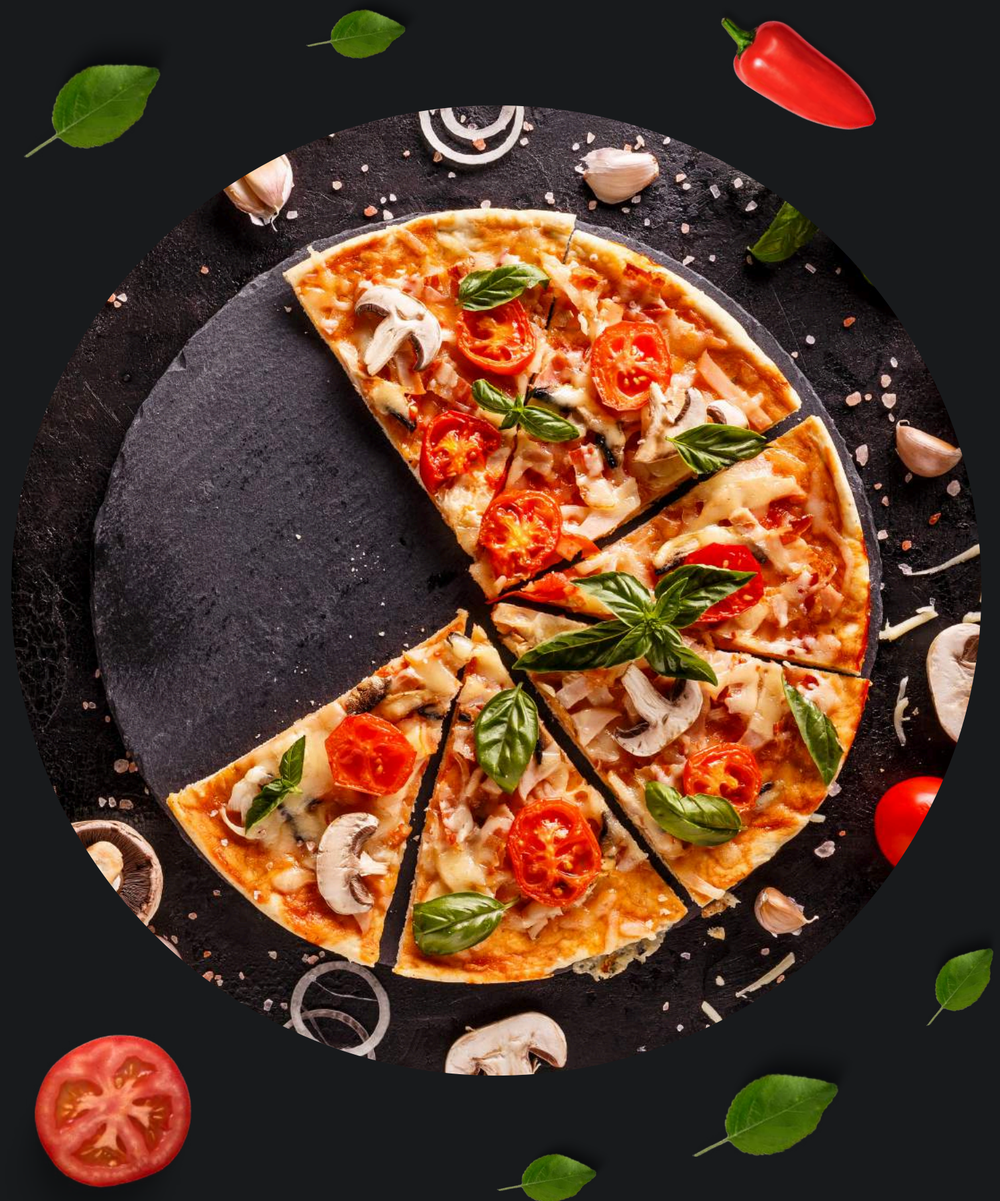
PIZZA SALES ANALYSIS





Welcome To PROJECT:

HELLO! My name is Nimra Iman and in this project i have made data analysis using mysql about pizza sales analysis. There are total of four datasets having some similarity in variables that are used.

ABOUT DATASETS:

PIZZAS:

pizza_id	pizza_type_id	size	price
bbq_ckn_s	bbq_ckn	S	12.75
bbq_ckn_m	bbq_ckn	M	16.75
bbq_ckn_l	bbq_ckn	L	20.75
cali_ckn_s	cali_ckn	S	12.75
cali_ckn_m	cali_ckn	M	16.75

ORDER_DETAILS:

order_details_id	order_id	pizza_id	quantity
1	1	hawaiian_m	1
2	2	classic_dlx_m	1
3	2	five_cheese_l	1
4	2	ital_supr_l	1
5	2	mexicana_m	1

ORDERS:

order_id	order_date	order_time
1	2015-01-01	11:38:36
2	2015-01-01	11:57:40
3	2015-01-01	12:12:28

PIZZA_TYPES:

pizza_type_id	name	category	ingredients
bbq_ckn	The Barbecue Chicken Pizza	Chicken	Barbecued Chicken, Red Peppers, Green Peppe...
cali_ckn	The California Chicken Pizza	Chicken	Chicken, Artichoke, Spinach, Garlic, Jalapeno P...
ckn_alfredo	The Chicken Alfredo Pizza	Chicken	Chicken, Red Onions, Red Peppers, Mushrooms...
ckn_pesto	The Chicken Pesto Pizza	Chicken	Chicken, Tomatoes, Red Peppers, Spinach, Garl...
southw_ckn	The Southwest Chicken Pizza	Chicken	Chicken, Tomatoes, Red Peppers, Red Onions, ...
thai_ckn	The Thai Chicken Pizza	Chicken	Chicken, Pineapple, Tomatoes, Red Peppers, T...

Calculate the total revenue generated from piza sales:

```
select round(sum(order_details.quantity * pizzas.price),2)
as revenue
from order_details join pizzas on
order_details.pizza_id=pizzas.pizza_id order by revenue desc ;
```

Identify the highest-priced pizza.

```sql
select pizza_types.name, pizzas.price from pizzas join pizza_types
on pizzas.pizza_type_id=pizza_types.pizza_type_id order by price desc
limit 1;
```

Calculate the total revenue generated from piza sales:

```
select round(sum(order_details.quantity * pizzas.price),2)
as revenue
from order_details join pizzas on
order_details.pizza_id=pizzas.pizza_id order by revenue desc ;
```

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

```
select round(avg(total_quantity),0) from (select orders.order_date,
sum(order_details.quantity) as total_quantity
 from orders
 join order_details on orders.order_id=order_details.order_id
group by orders.order_date) as  average  ;
```

Identify the most common pizza size ordered.

```sql
select
pizzas.size , count(pizzas.size) as total_order
from order_details join pizzas
on order_details.pizza_id=pizzas.pizza_id
group by pizzas.size
order by  total_order desc;
```

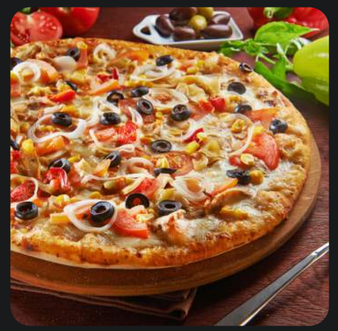
List the top 5 most ordered pizza types along with their quantities.

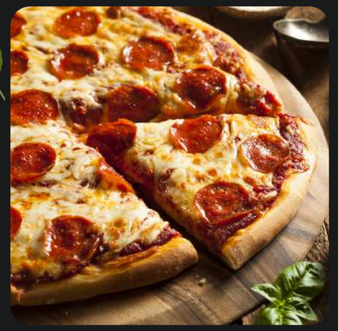
```sql
select pizza_types.name,
SUM(order_details.quantity) as total_quantity
from pizza_types join pizzas on
(pizzas.pizza_type_id=pizza_types.pizza_type_id)
 join  order_details on pizzas.pizza_id=order_details.pizza_id
group by pizza_types.name order by total_quantity desc limit 5;
```

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

```
select pizza_types.category, sum(order_details.quantity) as total_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 category order by total_quantity desc ;
```

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

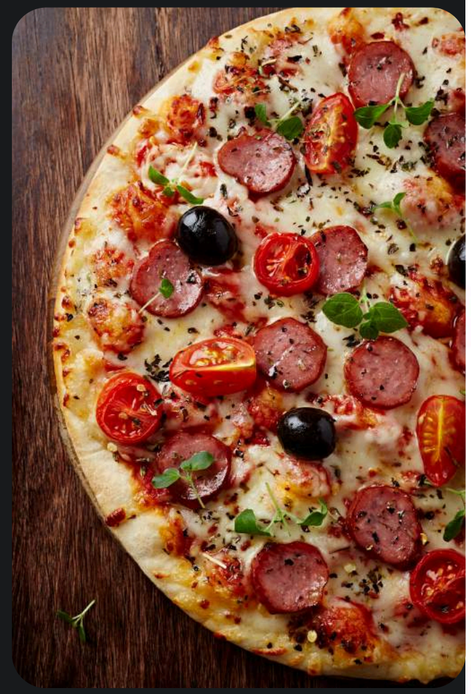




```sql
SELECT
    pizza_types.name,
    sum(order_details.quantity * pizzas.price) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name order by revenue limit 3;
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod

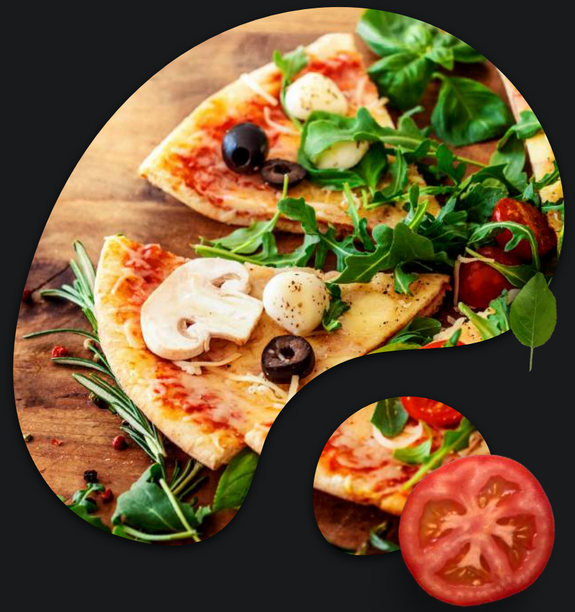
Calculate the percentage contribution of each pizza type to total revenue.



```
SELECT
    pizza_types.category,
    round((SUM(order_details.quantity * pizzas.price) / (SELECT
            ROUND(SUM(order_details.quantity * pizzas.price),
                    2) AS revenue
        FROM
            order_details
                JOIN
            pizzas ON order_details.pizza_id = pizzas.pizza_id
        ORDER BY revenue DESC))*100,2) 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
order by revenue desc;
```

Analyze the cumulative revenue generated over time

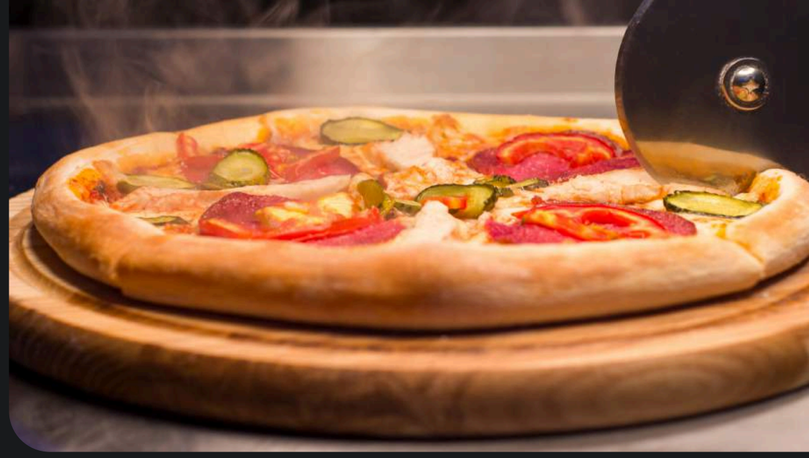
```sql
select order_date, sum(revenue_per_day) over(order by order_date)
 as cum_revenue from
(select order_date, sum(order_details.quantity * pizzas.price)
as revenue_per_day
 from orders join order_details on
 orders.order_id=order_details.order_id join pizzas on
 pizzas.pizza_id = order_details.pizza_id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

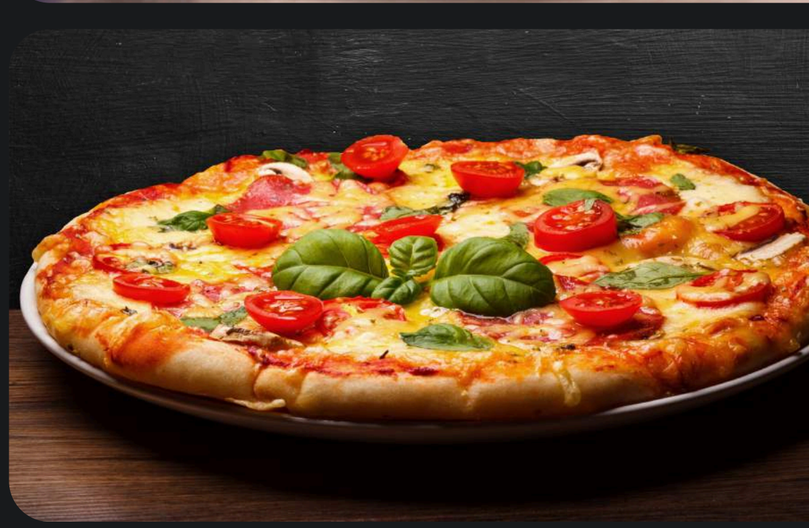


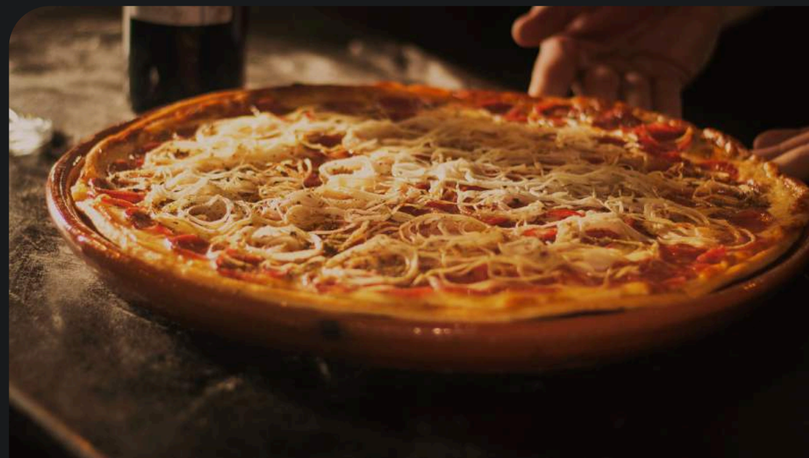
Determine the top 3 most ordered pizza types based on revenue for each pizza category.

```sql
select category, name, revenue 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 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, pizza_types.name
order by pizza_types.category, revenue desc) as a) as b
where rn <=3
;
```

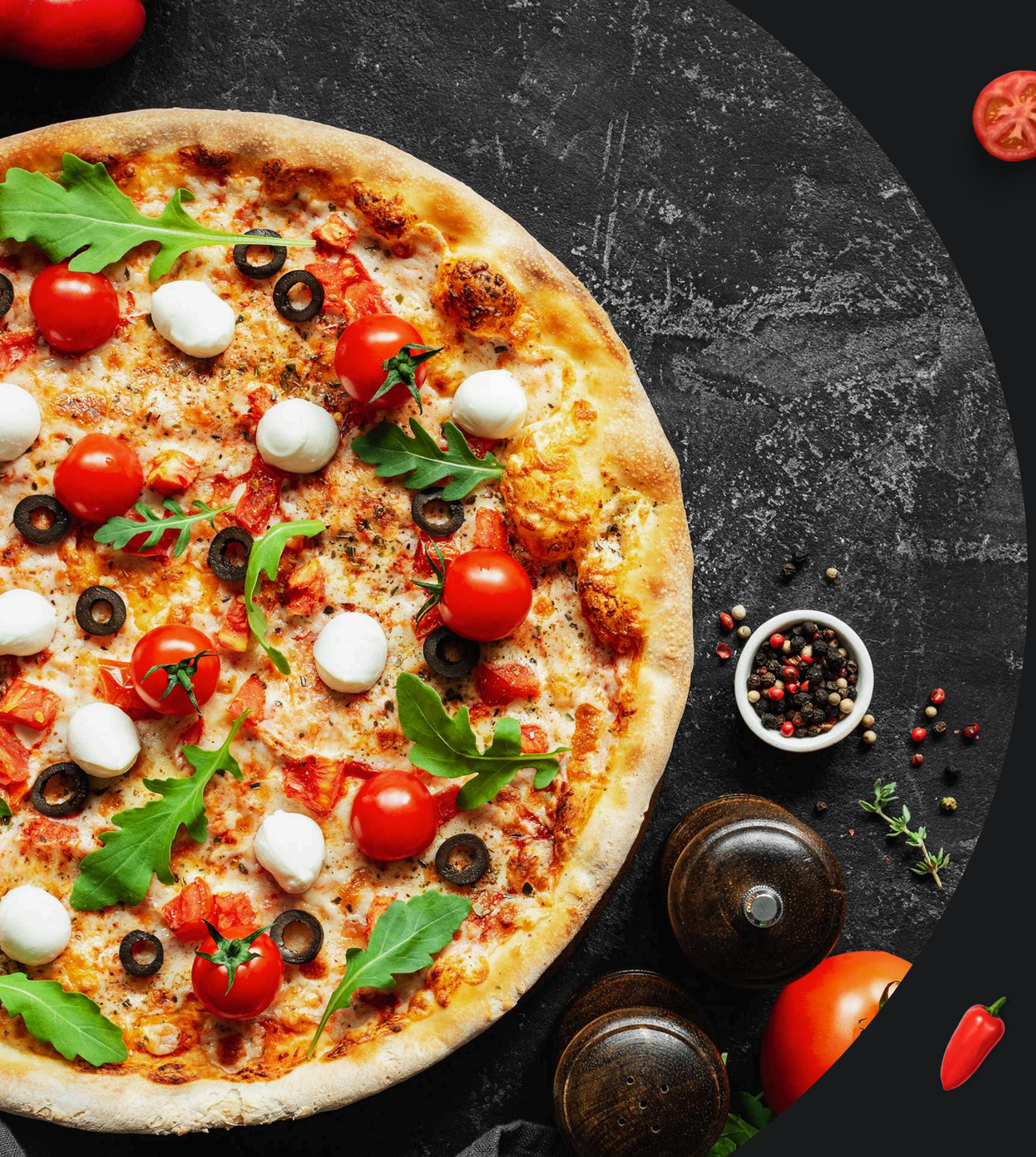
In conclusion, this data analysis project has provided valuable insights and use basic to advance sql queries enabling to make informed decisions and strategize effectively for future growth. Thank you for your attention and engagement throughout this journey.







ConNECT
for more such
Information



Thank You