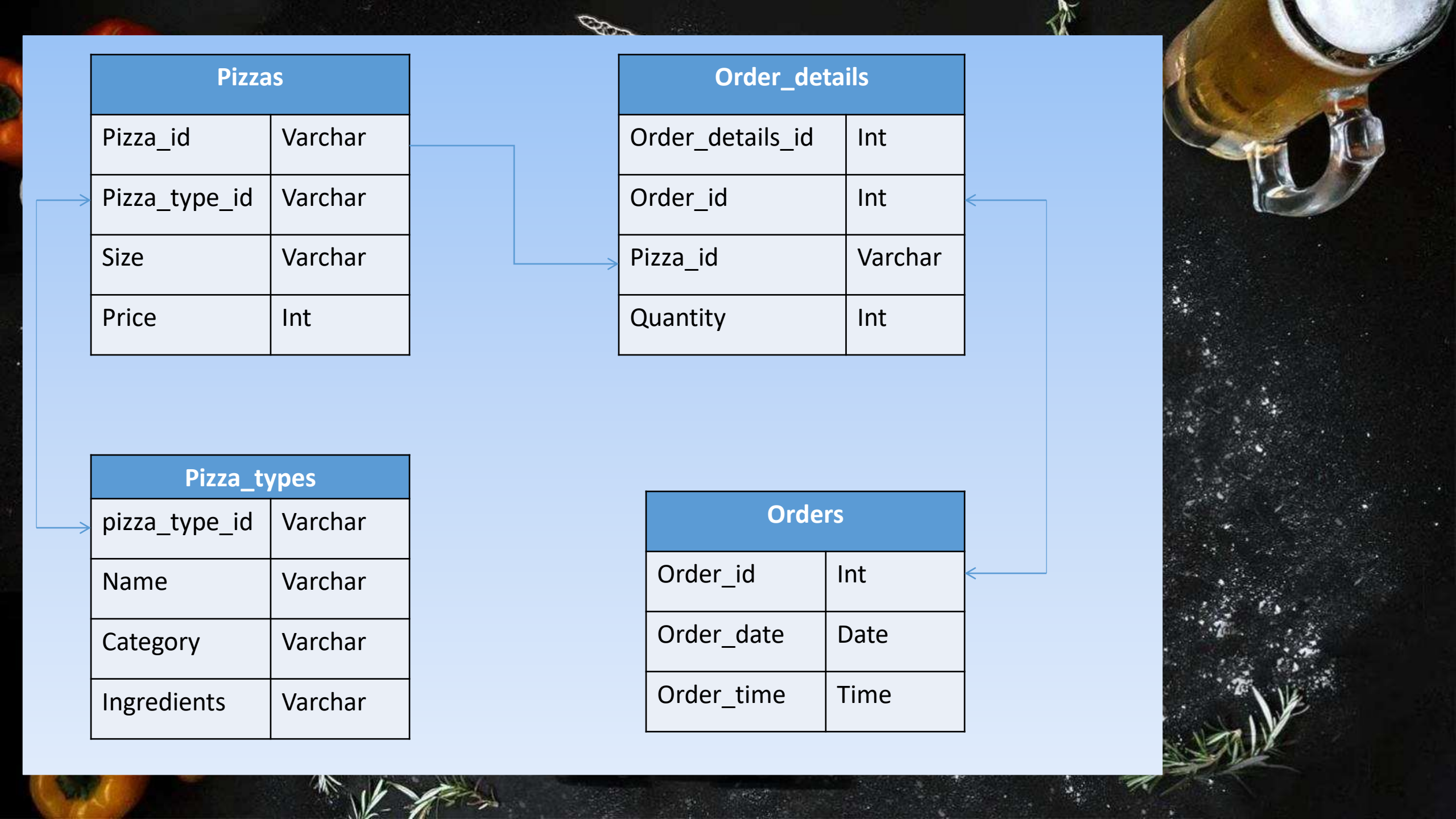
# Data Analytics Project

In today's dynamic food industry, understanding customer preferences and operational efficiencies is crucial for success. This data analytics project focuses on analyzing pizza sales data to derive meaningful insights using
SQL (Structured Query Language).
By leveraging the power of relational databases, we aim to uncover patterns, trends, and correlations that can inform strategic decisions for a pizza chain.

Skills Used - MS Excel, SQL, MS PowerPoint

Presented By:- Krishan Kumar Sharma
B.Tech (Computer Science & Engg.)
Software Engineer – CDAC, Noida
E-mail – krishanm90@gmail.com
LinkedIn -

## LET'S HAVE SOME FUN WITH NUMBERS USING SQL...



| Pizzas | |
|---|---|
| Pizza_id | Varchar |
| Pizza_type_id | Varchar |
| Size | Varchar |
| Price | Int |

| Order_details | |
|---|---|
| Order_details_id | Int |
| Order_id | Int |
| Pizza_id | Varchar |
| Quantity | Int |

| Pizza_types | |
|---|---|
| pizza_type_id | Varchar |
| Name | Varchar |
| Category | Varchar |
| Ingredients | Varchar |

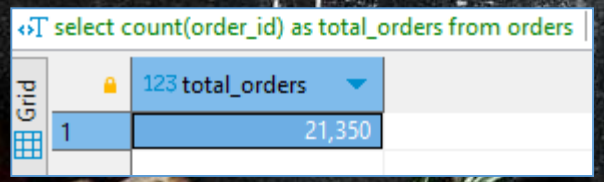
| Orders | |
|---|---|
| Order_id | Int |
| Order_date | Date |
| Order_time | Time |

# Problems to be addressed

- Retrieve the total number of orders placed.
- Calculate the total revenue generated from pizza sales.
- Identify the highest-priced pizza.
- Identify the most common pizza size ordered.
- List the top 5 most ordered pizza types along with their quantities.
- Join the necessary tables to find the total quantity of each pizza category ordered.
- Determine the distribution of orders by hour of the day.
- Join relevant tables to find the category-wise distribution of pizzas.
- Group the orders by date and calculate the average number of pizzas ordered per day.
- Determine the top 3 most ordered pizza types based on revenue
- Calculate the percentage contribution of each pizza type to total revenue.
- Analyze the cumulative revenue generated over time.
- Determine the top 3 most ordered pizza types based on revenue for each pizza category.

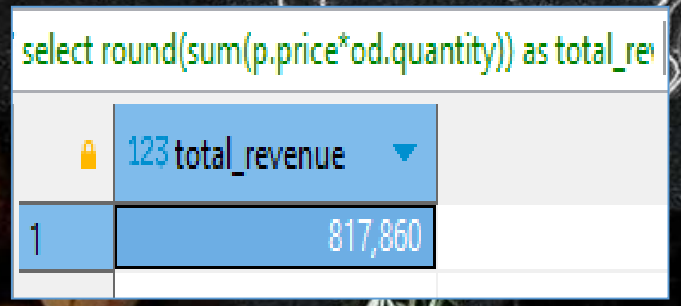
# Question-1 - Retrieve the total number of orders placed.

```sql
select
    count(order_id) as total_orders
from
    orders;
```



# Question-2 - Calculate the total revenue generated from pizza sales..

```sql
select
    round(sum(p.price * od.quantity)) as total_revenue
from
    pizzas p
join order_details od
on
    p.pizza_id = od.pizza_id ;
```



# Question-3 - Identify the highest-priced pizza.

```sql
select
    pt.name,
    p.price
from
    pizzas p
join pizza_types pt
on
    p.pizza_type_id = pt.pizza_type_id
where
    p.price = (
                select
                    max(price)
                from
                    pizzas p2);
```

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

select pt.name, p.price from pizzas p join pizza_typ

| | name | price |
|---|---|---|
| 1 | The Greek Pizza | 35.95 |

# Question-4 - Identify the most common pizza size ordered.

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

```sql
select
    p.size,
    count(od.quantity) as order_count
from
    pizzas p
join order_details od
on
    p.pizza_id = od.pizza_id
group by
    p.size
order by
    order_count desc
limit 1;
```

select p.size, count(od.quantity) as order_count frc

| | size | order_count |
|---|---|---|
| 1 | L | 18,526 |
| 2 | M | 15,385 |
| 3 | S | 14,137 |
| 4 | XL | 544 |
| 5 | XXL | 28 |

select p.size, count(od.quantity) as order_count frc

| | size | order_count |
|---|---|---|
| 1 | L | 18,526 |

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

```sql
select
    pt.name,
    sum(od.quantity) as order_count
from
    pizzas p
join order_details od
on
    p.pizza_id = od.pizza_id
join pizza_types pt on
    p.pizza_type_id = pt.pizza_type_id
group by
    pt.name
order by
    order_count desc
limit 5;
```

```
select pt.name, sum(od.quantity) as order_count f | Enter a SQL
```

| | name | order_count |
|---|---|---|
| 1 | The Classic Deluxe Pizza | 2,453 |
| 2 | The Barbecue Chicken Pizza | 2,432 |
| 3 | The Hawaiian Pizza | 2,422 |
| 4 | The Pepperoni Pizza | 2,418 |
| 5 | The Thai Chicken Pizza | 2,371 |

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

```sql
select
    pt.category,
    sum(od.quantity) as order_count
from
    pizza_types pt
join pizzas p
on
    pt.pizza_type_id = p.pizza_type_id
join order_details od on
    p.pizza_id = od.pizza_id
group by
    pt.category
order by
    order_count desc;
```

select pt.category, sum(od.qua | Enter a SQL expression

| | category | order_count |
|---|---|---|
| 1 | Classic | 14,888 |
| 2 | Supreme | 11,987 |
| 3 | Veggie | 11,649 |
| 4 | Chicken | 11,050 |

# Question-7- Determine the distribution of orders by hour of the day.

```sql
select
    extract(hour from order_time) as hour_slot,
    count(o.order_id) as order_count
from
    orders o
group by
    hour_slot
order by
    hour_slot;
```

select extract(hour from order_time) as hour_slot,

| | 123 hour_slot | 123 order_count |
|---|---|---|
| 1 | 9 | 1 |
| 2 | 10 | 8 |
| 3 | 11 | 1,231 |
| 4 | 12 | 2,520 |
| 5 | 13 | 2,455 |
| 6 | 14 | 1,472 |
| 7 | 15 | 1,468 |
| 8 | 16 | 1,920 |
| 9 | 17 | 2,336 |
| 10 | 18 | 2,399 |
| 11 | 19 | 2,009 |
| 12 | 20 | 1,642 |
| 13 | 21 | 1,198 |
| 14 | 22 | 663 |
| 15 | 23 | 28 |

## Question-8- Join relevant tables to find the category-wise distribution of pizzas.

```sql
select
    category,
    count(name)
from
    pizza_types pt
group by
    category ;
```

| | category | count |
|---|---|---|
| 1 | Supreme | 9 |
| 2 | Chicken | 6 |
| 3 | Classic | 8 |
| 4 | Veggie | 9 |

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

```sql
select
    round(avg(total_sale_per_day)) as average_pizzas_ordered_per_day
from
    (
    select
        o.order_date as date_slot,
        sum(quantity) as total_sale_per_day
    from
        orders o
    join order_details od
    on
            o.order_id = od.order_id
        group by
            o.order_date
        order by
        o.order_date) a;
```

Final Query Result

| | average_pizzas_ordered_per_day |
|---|---|
| 1 | 138 |

Inner/Sub-Query Query Result

| | date_slot | total_sale_per_day |
|---|---|---|
| 1 | 2015-01-01 | 162 |
| 2 | 2015-01-02 | 165 |
| 3 | 2015-01-03 | 158 |
| 4 | 2015-01-04 | 106 |
| 5 | 2015-01-05 | 125 |
| 6 | 2015-01-06 | 147 |
| 7 | 2015-01-07 | 138 |
| 8 | 2015-01-08 | 173 |
| 9 | 2015-01-09 | 127 |
| 10 | 2015-01-10 | 146 |
| 11 | 2015-01-11 | 116 |
| 12 | 2015-01-12 | 119 |
| 13 | 2015-01-13 | 120 |
| 14 | 2015-01-14 | 150 |
| 15 | 2015-01-15 | 123 |
| 16 | 2015-01-16 | 158 |
| 17 | 2015-01-17 | 125 |
| 18 | 2015-01-18 | 122 |
| 19 | 2015-01-19 | 142 |
| 20 | 2015-01-20 | 143 |
| 21 | 2015-01-21 | 129 |
| 22 | 2015-01-22 | 158 |
| 23 | 2015-01-23 | 152 |
| 24 | 2015-01-24 | 142 |
| 25 | 2015-01-25 | 102 |
| 26 | 2015-01-26 | 113 |
| 27 | 2015-01-27 | 151 |
| 28 | 2015-01-28 | 118 |
| 29 | 2015-01-29 | 119 |
| 30 | 2015-01-30 | 138 |
| 31 | 2015-01-31 | 145 |

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

```sql
select
    pt.name,
    round(sum(p.price * od.quantity)) as total_revenue
from
    pizzas p
join order_details od
on
    p.pizza_id = od.pizza_id
join pizza_types pt
on
    p.pizza_type_id = pt.pizza_type_id
group by
    pt.name
order by
    total_revenue desc
limit 3;
```

| | name | total_revenue |
|---|---|---|
| 1 | The Thai Chicken Pizza | 43,434 |
| 2 | The Barbecue Chicken Pizza | 42,768 |
| 3 | The California Chicken Pizza | 41,410 |

# Question-11- Calculate the percentage contribution of each pizza type to total revenue.

```sql
select
    pt.category as category,
    (round(((round(sum(p.price * od.quantity)) / (
            select
                round(sum(p2.price * od2.quantity))
            from
                pizzas p2
            join order_details od2
            on
                p2.pizza_id = od2.pizza_id)))* 100)|| '%')
            as revenue_percentage
from
    pizzas p
join order_details od
on
    p.pizza_id = od.pizza_id
join pizza_types pt
on
    p.pizza_type_id = pt.pizza_type_id
group by
    category
order by
    revenue_percentage;
```

| | category | revenue_percentage |
|---|---|---|
| 1 | Chicken | 24% |
| 2 | Veggie | 24% |
| 3 | Supreme | 25% |
| 4 | Classic | 27% |

# Question-12- Analyze the cumulative revenue generated over time.

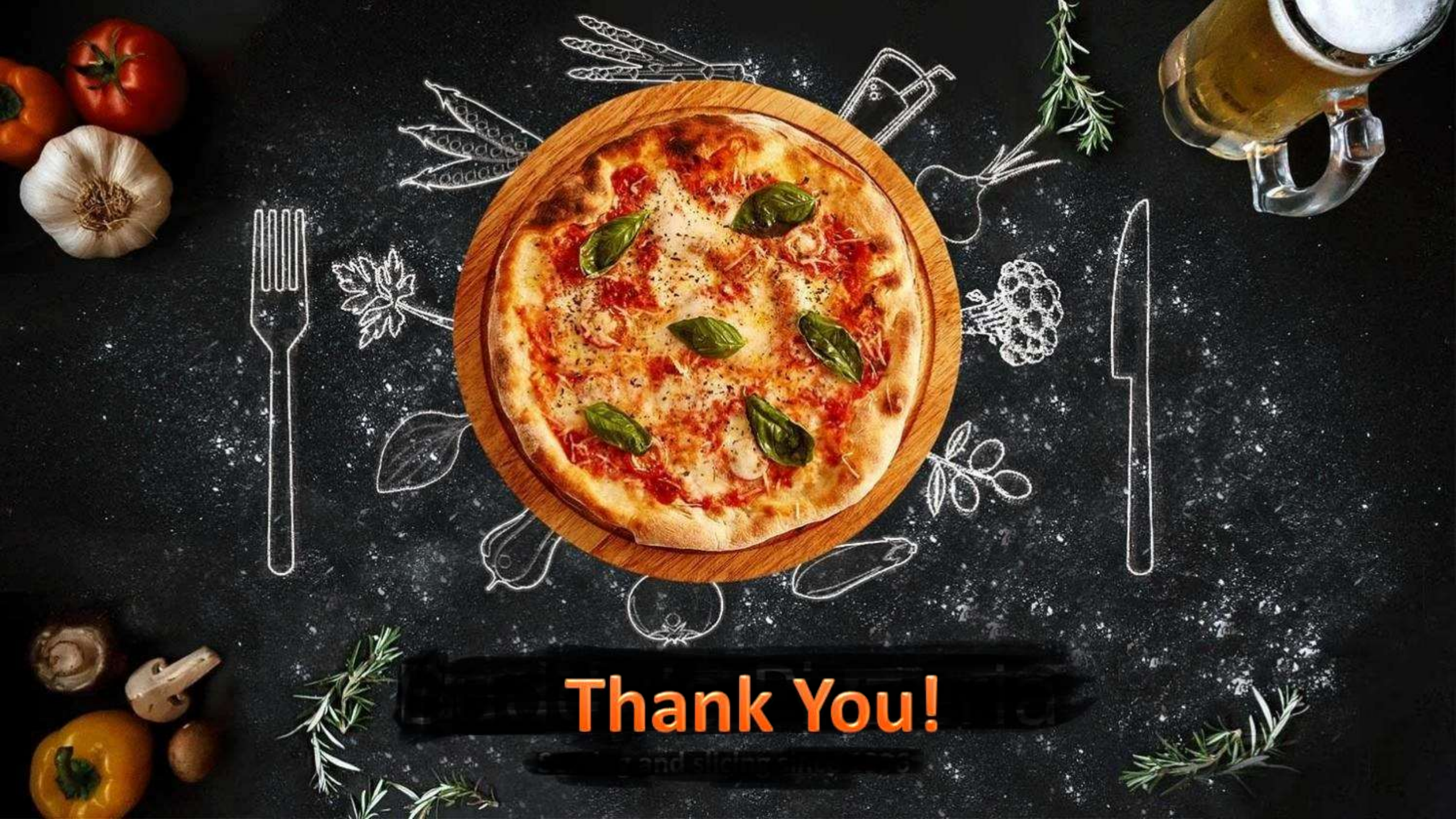
```sql
select
    order_date,
    sum(revenue) over(
    order by order_date) as cumulative_revenue
from
    (
    select
        o.order_date,
        round(sum(p.price * od.quantity)) as revenue
    from
        pizzas p
    join order_details od
on
        p.pizza_id = od.pizza_id
    join orders o
on
        od.order_id = o.order_id
    group by
        o.order_date
    order by
        o.order_date) as revenue_data;
```

| order_date | cumulative_revenue |
|---|---|
| 2015-01-01 | 2,714 |
| 2015-01-02 | 5,446 |
| 2015-01-03 | 8,108 |
| 2015-01-04 | 9,863 |
| 2015-01-05 | 11,929 |
| 2015-01-06 | 14,358 |
| 2015-01-07 | 16,560 |
| 2015-01-08 | 19,398 |
| 2015-01-09 | 21,525 |
| 2015-01-10 | 23,989 |
| 2015-01-11 | 25,861 |
| 2015-01-12 | 27,780 |
| 2015-01-13 | 29,830 |
| 2015-01-14 | 32,357 |
| 2015-01-15 | 34,342 |
| 2015-01-16 | 36,936 |
| 2015-01-17 | 39,000 |
| 2015-01-18 | 40,977 |
| 2015-01-19 | 43,364 |
| 2015-01-20 | 45,762 |
| 2015-01-21 | 47,803 |
| 2015-01-22 | 50,300 |
| 2015-01-23 | 52,724 |
| 2015-01-24 | 55,013 |
| 2015-01-25 | 56,631 |
| 2015-01-26 | 58,515 |
| 2015-01-27 | 61,043 |
| 2015-01-28 | 63,059 |
| 2015-01-29 | 65,104 |
| 2015-01-30 | 67,374 |
| 2015-01-31 | 69,792 |
| 2015-02-01 | 72,981 |

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

```sql
select *
from
    (
    select
        pt.category,
        pt.name ,
        round(sum(p.price * od.quantity)) as revenue,
        rank() over(partition by category
    order by
        round(sum(p.price * od.quantity)) desc) as rn
    from
        pizzas p
    join order_details od
on
        p.pizza_id = od.pizza_id
    join pizza_types pt
on
        p.pizza_type_id = pt.pizza_type_id
    group by
        pt.category,
        pt.name) as a
where
    rn<4 ;
```

| category | name | revenue | rn |
|---|---|---|---|
| Chicken | The Thai Chicken Pizza | 43,434 | 1 |
| Chicken | The Barbecue Chicken Pizza | 42,768 | 2 |
| Chicken | The California Chicken Pizza | 41,410 | 3 |
| Classic | The Classic Deluxe Pizza | 38,180 | 1 |
| Classic | The Hawaiian Pizza | 32,273 | 2 |
| Classic | The Pepperoni Pizza | 30,162 | 3 |
| Supreme | The Spicy Italian Pizza | 34,831 | 1 |
| Supreme | The Italian Supreme Pizza | 33,477 | 2 |
| Supreme | The Sicilian Pizza | 30,940 | 3 |
| Veggie | The Four Cheese Pizza | 32,266 | 1 |
| Veggie | The Mexicana Pizza | 26,781 | 2 |
| Veggie | The Five Cheese Pizza | 26,066 | 3 |


Thank You!