

Project Title: "Pizza  
Sales Analysis Using  
SQL"

Vikram paliwal



**Brief about the project:**  
This project aims to analyze pizza sales data using SQL.  
We used SQL queries to extract useful business insights  
**Tools Used:** SQL (MySQL, CSV FILE)

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.

Retrieve the total number of orders placed.

```
select count(order_id) from orders;
```

Result Grid	
	count(order_id)
	21350



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** order\_details.pizza\_id = pizzas.pizza\_id;

	<b>total_sales</b>
▶	817860.05

# Identify the most common pizza size ordered.

```
1 • SELECT
2     pizzas.size,
3     COUNT(order_details.order_details_id) AS order_count
4
5 FROM
6     pizzas
7     JOIN
8         order_details ON pizzas.pizza_id = order_details.pizza_id
9
10 GROUP BY pizzas.size
11
12 ORDER BY order_count DESC;
```

size	order_count
L	18526
M	15385
S	14137
XL	544
XXL	28



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

```
SELECT  
    pizza_types.name, SUM(order_details.quantity) AS sum_qut  
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 sum_qut DESC LIMIT 5;
```

name	sum_qut
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418



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

- **SELECT**

```
    pizza_types.category,  
    SUM(order_details.quantity) AS sum_qunt  
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 sum_qunt DESC;
```

	<b>category</b>	<b>sum_qunt</b>
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050



# 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 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 DESC  
LIMIT 3;
```

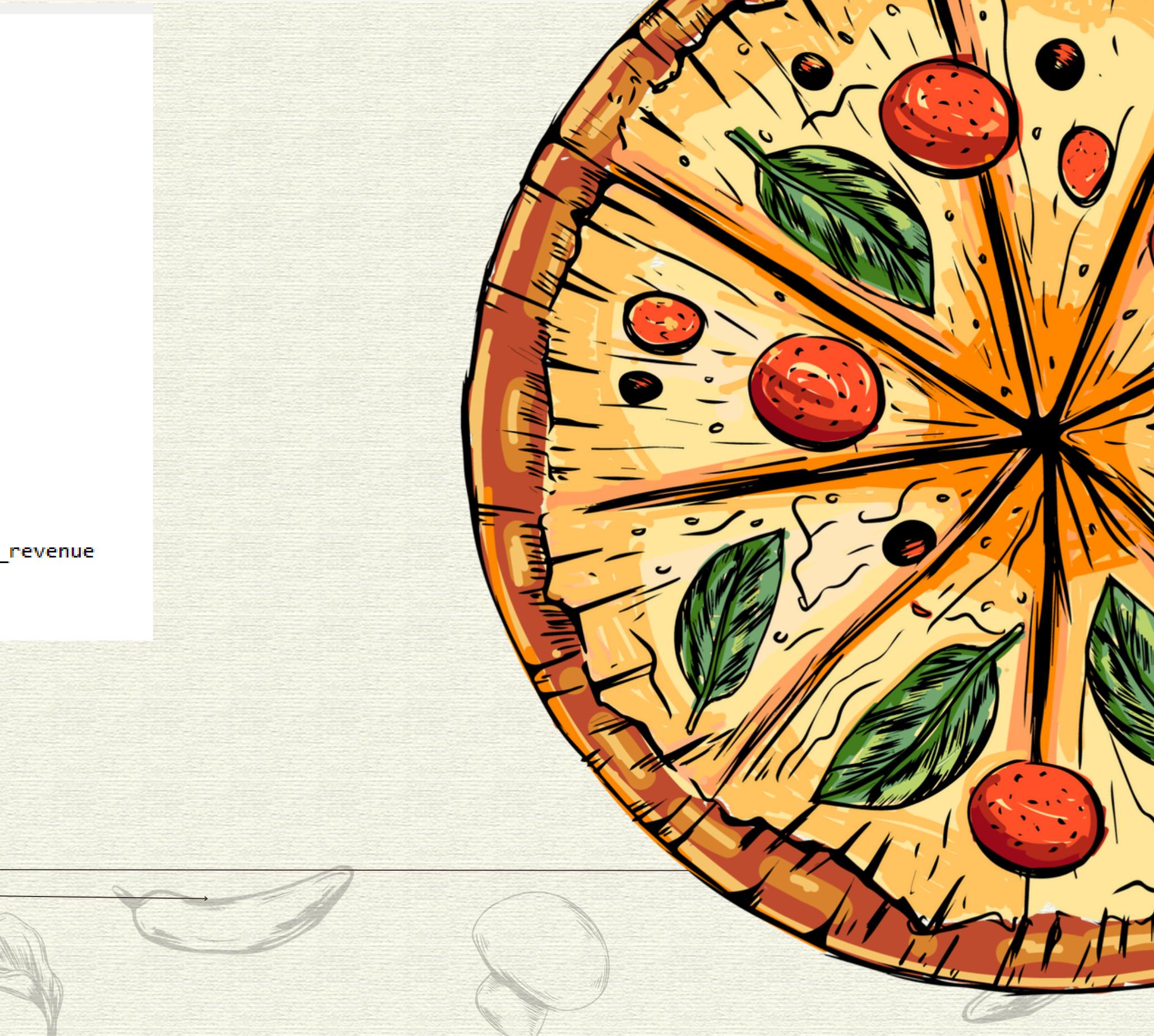
name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5



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

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

	category	revenue_percentage
1	Classic	27.00
2	Supreme	25.00
3	Veggie	24.00
4	Chicken	24.00

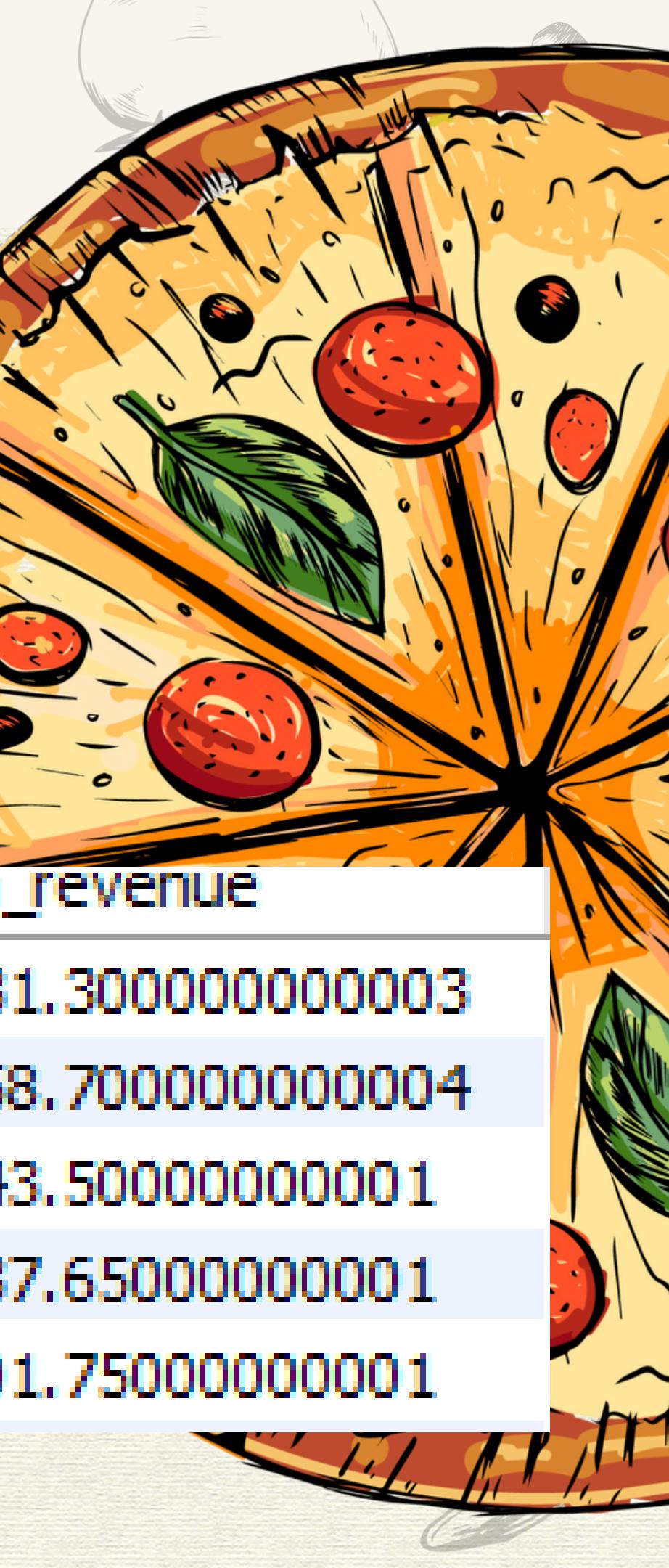


# Analyze the cumulative revenue generated over time.

```
select order_date,  
       sum(revenue) over(order by order_date) as cum_revenue 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 daily_revenue;
```

order_date	cum_revenue
2015-01-01 00:00:00	2713.850000000004
2015-01-02 00:00:00	5445.75
2015-01-03 00:00:00	8108.15
2015-01-04 00:00:00	9863.6
2015-01-05 00:00:00	11929.55

order_date	cum_revenue
2015-01-13 00:00:00	29831.30000000003
2015-01-14 00:00:00	32358.70000000004
2015-01-15 00:00:00	34343.5000000001
2015-01-16 00:00:00	36937.6500000001
2015-01-17 00:00:00	39001.7500000001



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

`SELECT`

```
    ROUND(AVG(quantity), 0) as avg_pizza_par_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_par_day
-------------------

138
-----



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

```
SELECT  
    category, COUNT(pizza_type_id)  
FROM  
    pizza_types  
GROUP BY category;
```

category	count(pizza_type_id)
Chicken	6
Classic	8
Supreme	9
Veggie	9



# Determine the distribution of orders by hour of the day.

SELECT

```
HOUR(order_time), COUNT(order_id) AS order_count
```

FROM

```
orders
```

```
GROUP BY HOUR(order_time)
```

```
ORDER BY order_num DESC;
```

SELECT

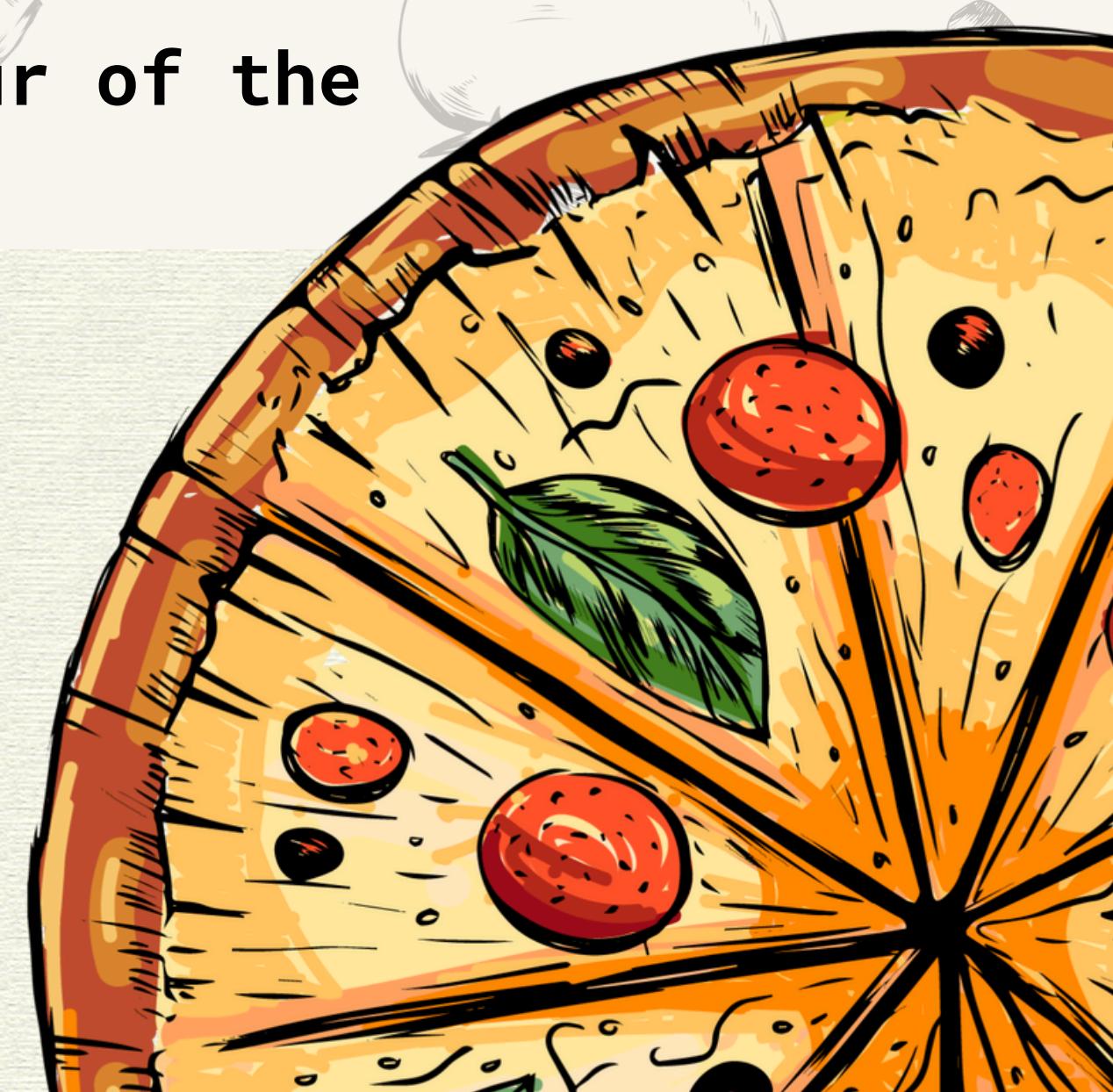
```
HOUR(order_time) AS ord_time, COUNT(order_id) AS order_count
```

FROM

```
orders
```

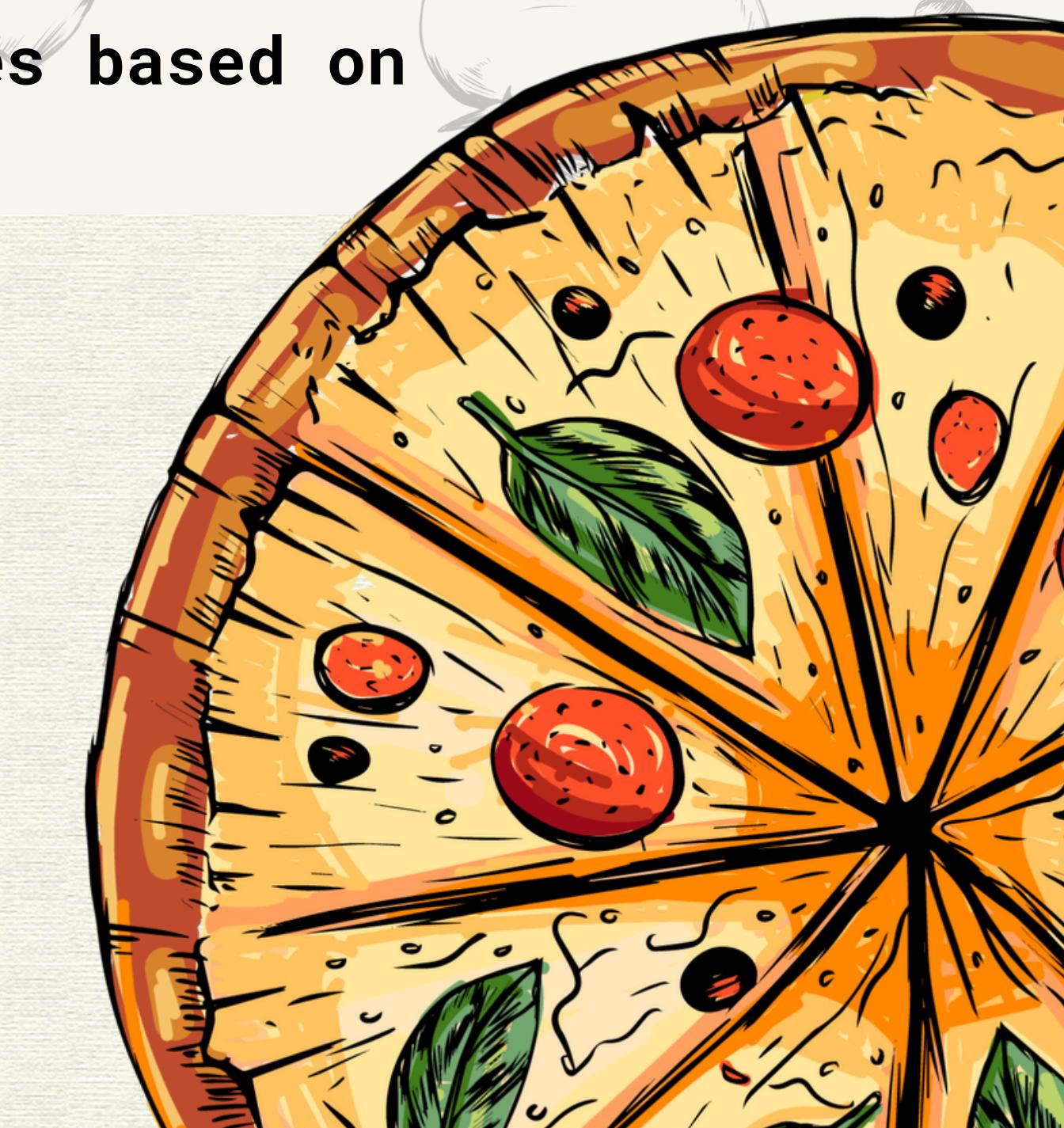
```
GROUP BY ord_time ;
```

ord_time	order_count	ord_time	order_count	ord_time	order_count
9	1	14	1472	19	2009
10	8	15	1468	20	1642
11	1231	16	1920	21	1198
12	2520	17	2336	22	663
13	2455	18	2399		



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

```
SELECT
    category,
    name AS pizza_type,
    revenue
FROM (
    SELECT
        pt.category,
        pt.name,
        SUM(od.quantity * p.price) AS revenue,
        RANK() OVER (PARTITION BY pt.category ORDER BY SUM(od.quantity * p.price) DESC) AS rnk
    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, pt.name
)
```



category	pizza_type	revenue	category	pizza_type	revenue
Chicken	The Thai Chicken Pizza	43434.25	Supreme	The Spicy Italian Pizza	34831.25
Chicken	The Barbecue Chicken Pizza	42768	Supreme	The Italian Supreme Pizza	33476.75
Chicken	The California Chicken Pizza	41409.5	Supreme	The Sicilian Pizza	30940.5
Classic	The Classic Deluxe Pizza	38180.5	Veggie	The Four Cheese Pizza	32265.700000000004
Classic	The Hawaiian Pizza	32273.25	Veggie	The Mexicana Pizza	26780.75