sql query and inshight from dataset

SQL PROJECT ON PIZZA SALES

Start Your Slide

ABOUT US

WELCOME TO LAPINOZ PIZZA, WHERE PASSION MEETS PERFECTION IN EVERY SLICE! WE BELIEVE THAT GREAT PIZZA STARTS WITH FRESH INGREDIENTS, AUTHENTIC RECIPES, AND A LOVE FOR FOOD.

OUR STORY

FOUNDED IN 2023, LAPI PIZZA BEGAN AS A SMALL PIZZERIA WITH A BIG DREAM – TO SERVE MOUTHWATERING, HANDCRAFTED PIZZAS THAT BRING PEOPLE TOGETHER. OVER THE YEARS, OUR COMMITMENT TO QUALITY, TASTE, AND CUSTOMER SATISFACTION HAS MADE US A FAVORITE AMONG PIZZA LOVERS.

PROJECT GOAL

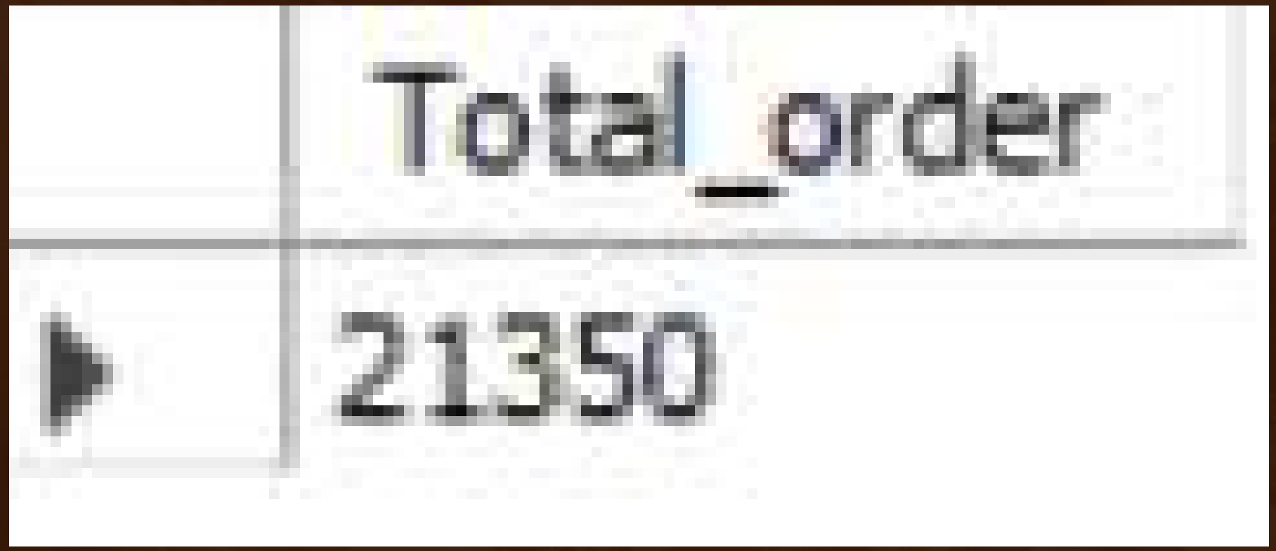
ANALYZE DATA

DERIVE INSHIGHT

DRAW CONCLUSIOBN

RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED

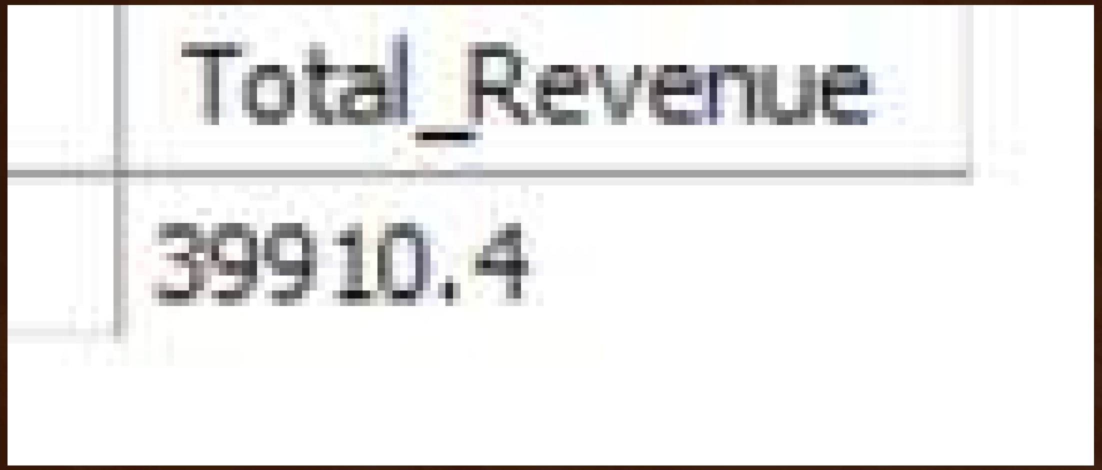
```
SELECT COUNT(ORDER_ID) AS
TOTAL_ORDER FROM ORDERS;
```



```
Total_order
21350
```

CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES

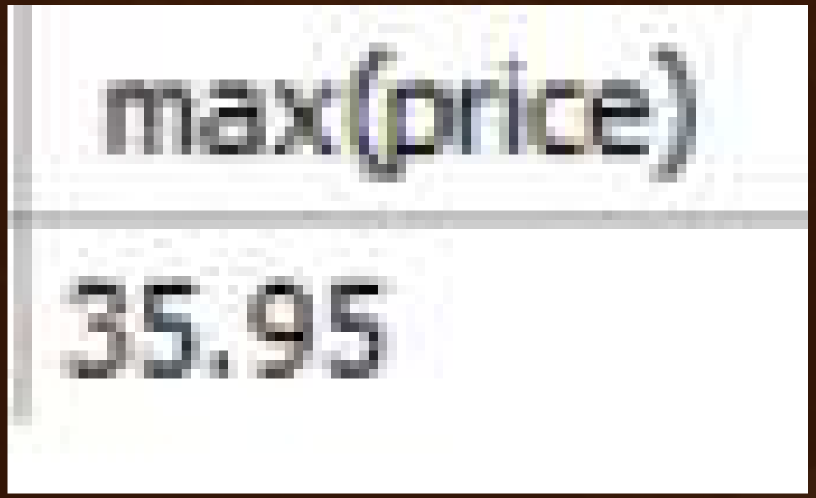
```
SELECT ROUND(SUM(PRICE* QUANTITY),2)
AS TOTAL_REVENUE
FROM PIZZAS AS P
INNER JOIN ORDER_DETAILS AS O
ON P.PIZZA_ID =O.PIZZA_ID;
```



Total_Revenue
39910.4

Identify the highest-priced pizza

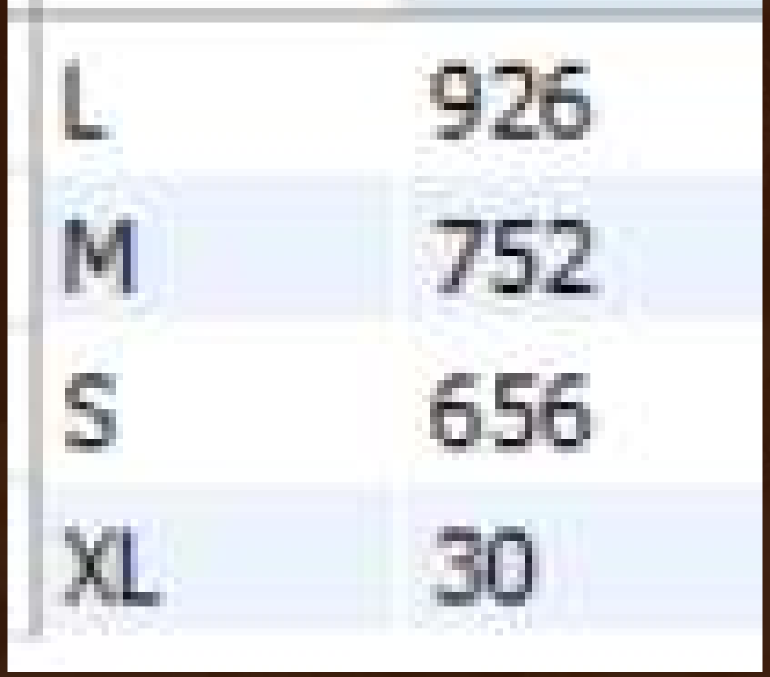
```
SELECT MAX(PRICE)FROM PIZZAS;
SELECT NAME , PRICE
FROM PIZZA_TYPES AS P1
INNER JOIN PIZZAS AS P2
ON P1.PIZZA_TYPE_ID = P2.PIZZA_TYPE_ID
ORDER BY PRICE DESC
LIMIT 1;
```



max(price)
35.95

Identify the most common pizza size ordered.

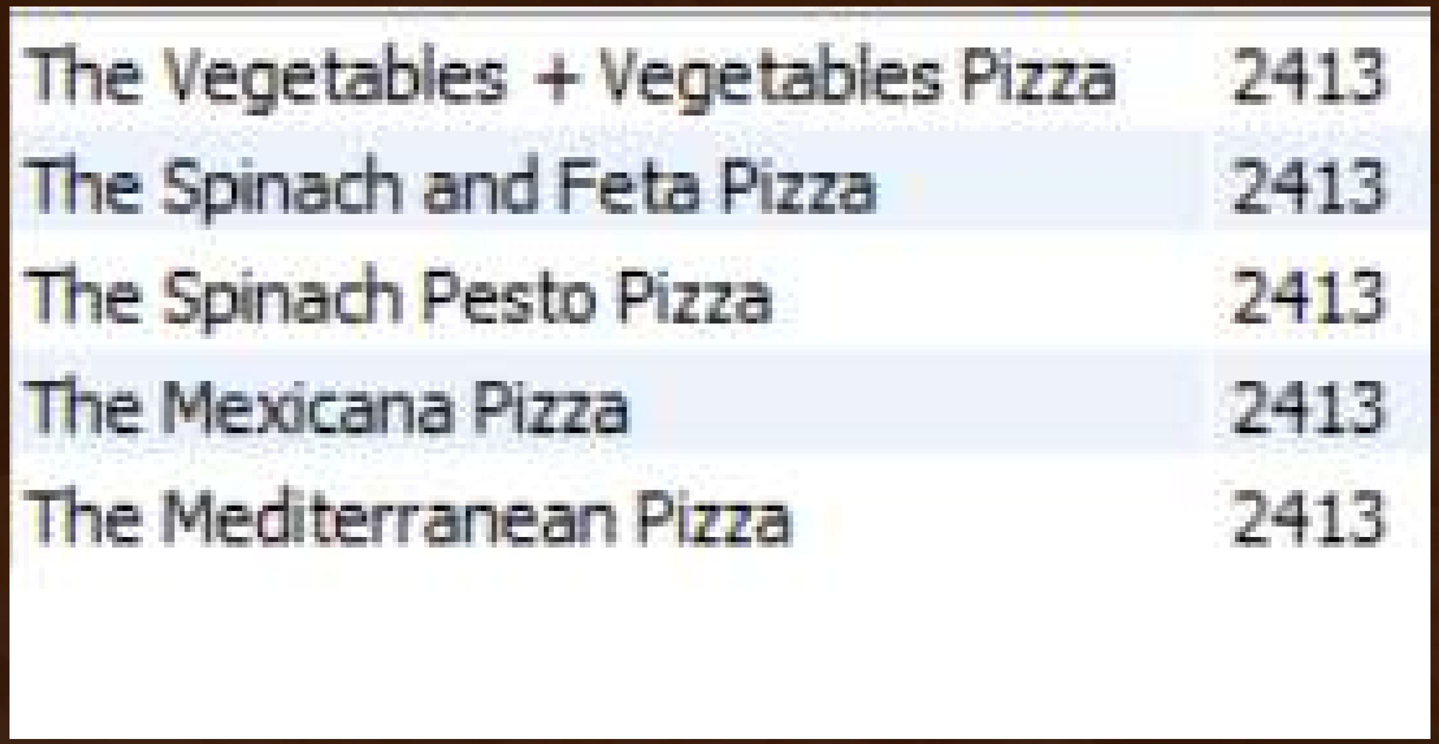
```
SELECT SIZE, COUNT(ORDER_ID)
FROM PIZZAS AS P
INNER JOIN ORDER_DETAILS AS O
ON P.PIZZA_ID = O.PIZZA_ID
GROUP BY SIZE
ORDER BY COUNT(ORDER_ID) DESC;
```



```
L     926
M     752
S     656
XL    30
```

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

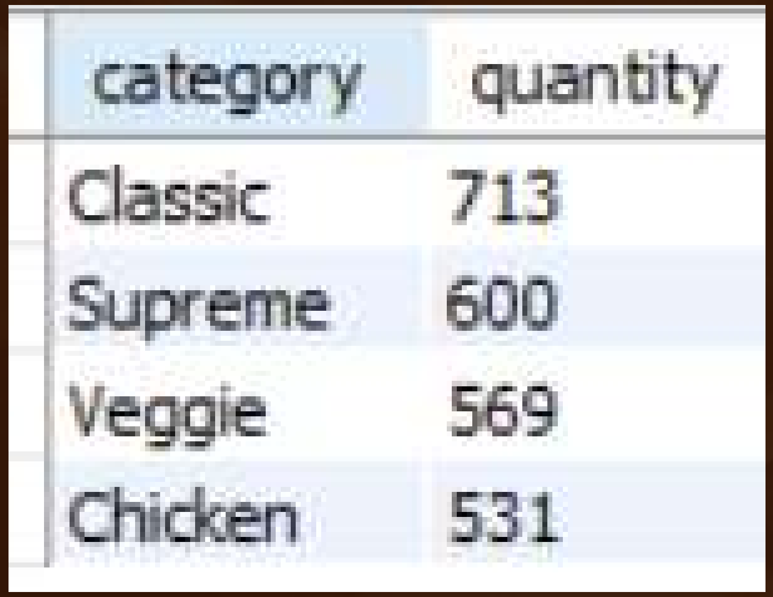
```
SELECT NAME, SUM(QUANTITY) AS
QUANTITY
FROM PIZZAS AS P1
INNER JOIN PIZZA_TYPES AS P2
ON P1.PIZZA_TYPE_ID = P1.PIZZA_TYPE_ID
INNER JOIN ORDER_DETAILS AS O
ON P1.PIZZA_ID=O.PIZZA_ID
GROUP BY NAME
ORDER BY SUM(QUANTITY) DESC
LIMIT  5;
```



The Vegetables + Vegetables Pizza	2413
The Spinach and Feta Pizza	2413
The Spinach Pesto Pizza	2413
The Mexicana Pizza	2413
The Mediterranean Pizza	2413

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

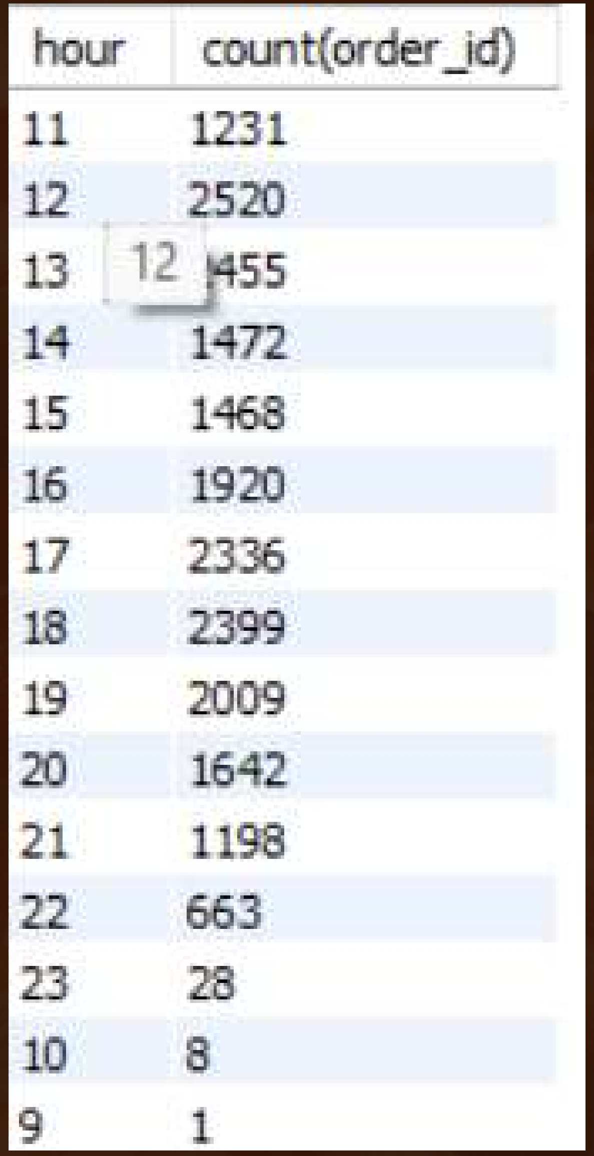
```
SELECT CATEGORY, SUM(QUANTITY) AS
QUANTITY
FROM PIZZA_TYPES AS P1
INNER JOIN PIZZAS AS P2
ON P1.PIZZA_TYPE_ID = P2.PIZZA_TYPE_ID
INNER JOIN ORDER_DETAILS AS O
ON P2.PIZZA_ID = O.PIZZA_ID
GROUP BY CATEGORY
ORDER BY SUM(QUANTITY) DESC;
```



category	quantity
Classic	713
Supreme	600
Veggie	569
Chicken	531

Determine the distribution of orders by hour of the day

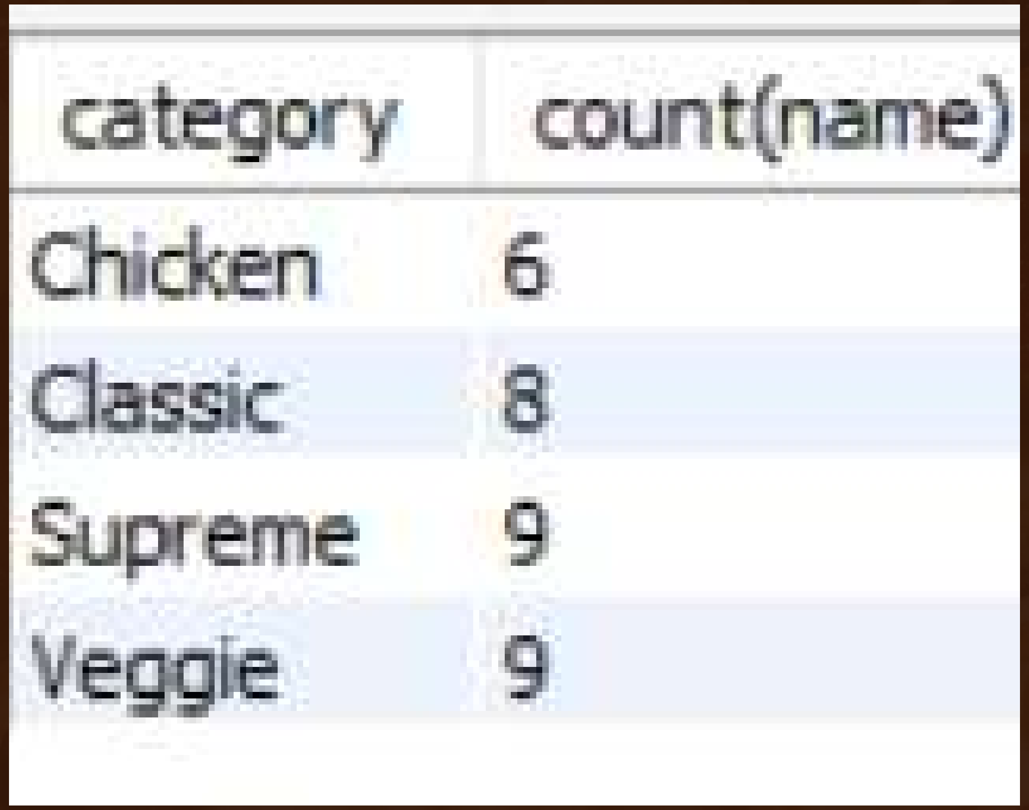
```
SELECT EXTRACT(HOUR FROM ORDER_TIME)
AS HOUR , COUNT(ORDER_ID)
FROM ORDERS
GROUP BY EXTRACT(HOUR FROM
ORDER_TIME);
```



hour	count(order_id)
11	1231
12	2520
13	455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642
21	1198
22	663
23	28
10	8
9	1

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

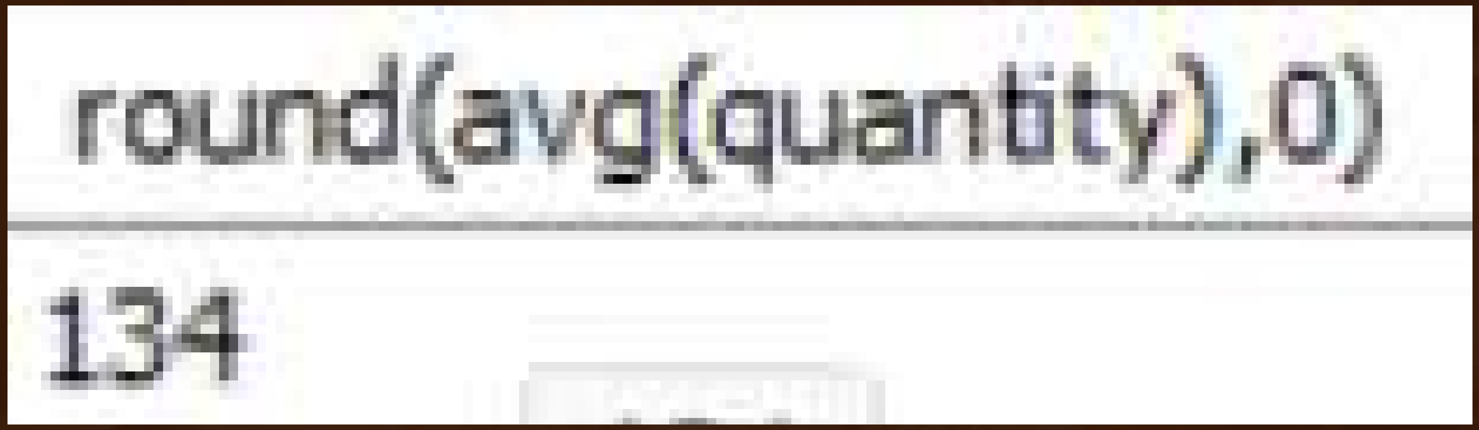
```
SELECT CATEGORY, COUNT(NAME)
FROM PIZZA_TYPES
GROUP BY CATEGORY;
```



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

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

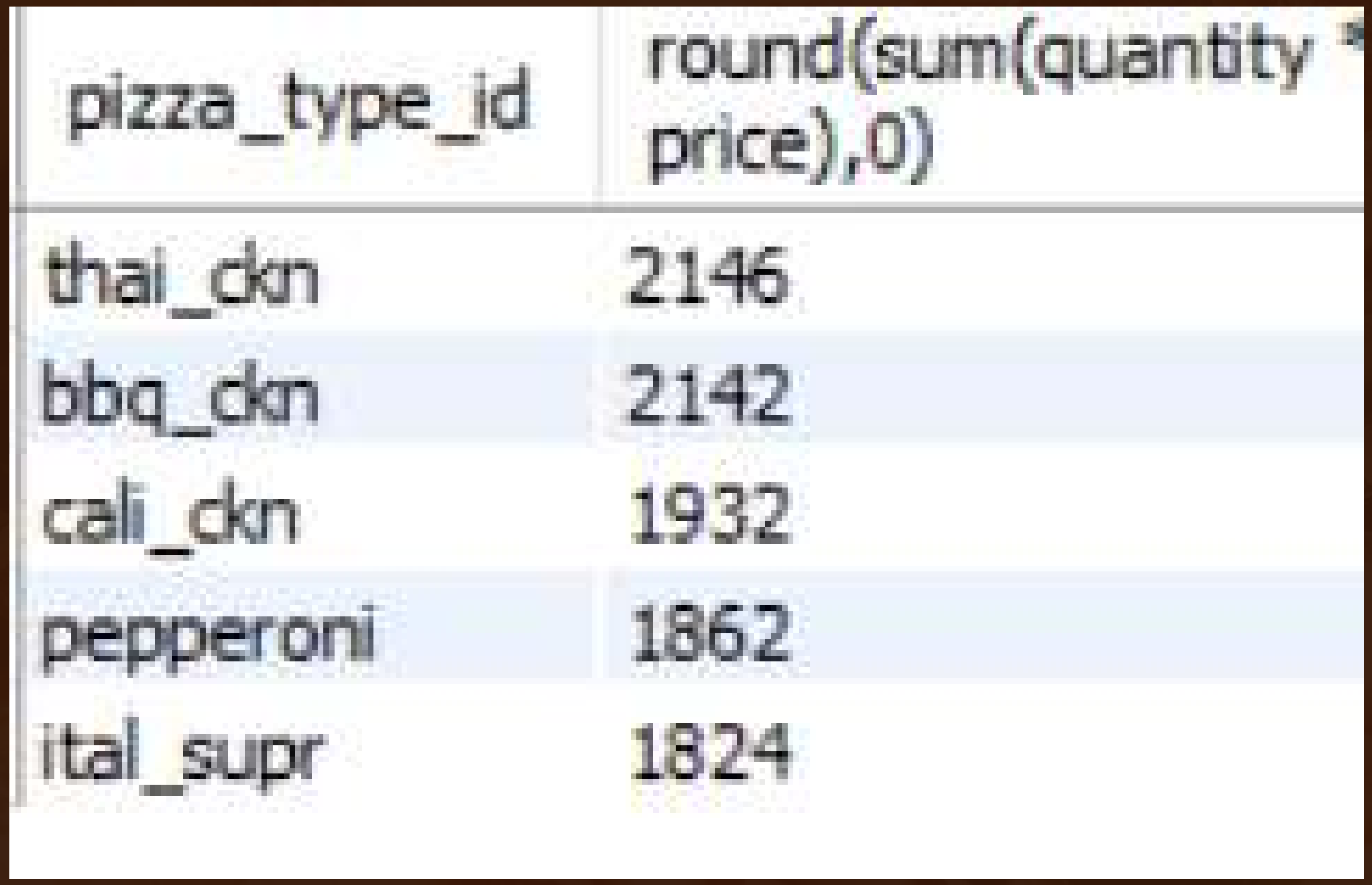
```
SELECT ROUND(AVG(QUANTITY),0) FROM
(SELECT ORDER_DATE, DAY(ORDER_DATE)
AS DAY , SUM(QUANTITY) AS QUANTITY
FROM ORDER_DETAILS AS O1
INNER JOIN ORDERS AS O2
ON O1.ORDER_ID = O2.ORDER_ID
GROUP BY ORDER_DATE) AS
ORDER_QUANTITY;
```



```
round(avg(quantity),0)
134
```

Determine the top 3 most ordered pizza types based on revenue

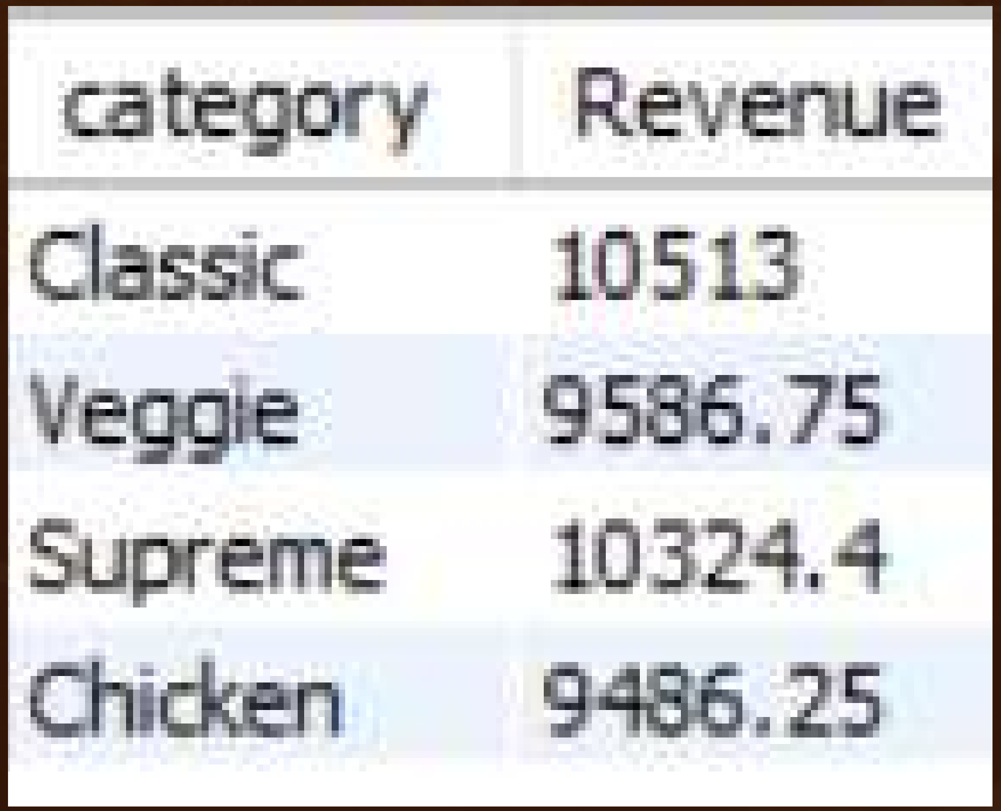
```
SELECT PIZZA_TYPE_ID ,
ROUND(SUM(QUANTITY * PRICE),0)
FROM PIZZAS AS P
INNER JOIN ORDER_DETAILS  AS O
ON P.PIZZA_ID = O.PIZZA_ID
GROUP BY PIZZA_TYPE_ID
ORDER BY SUM(QUANTITY * PRICE) DESC
LIMIT 5;
```



pizza_type_id	round(sum(quantity * price),0)
thai_ckn	2146
bbq_ckn	2142
cali_ckn	1932
pepperoni	1862
ital_supr	1824

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

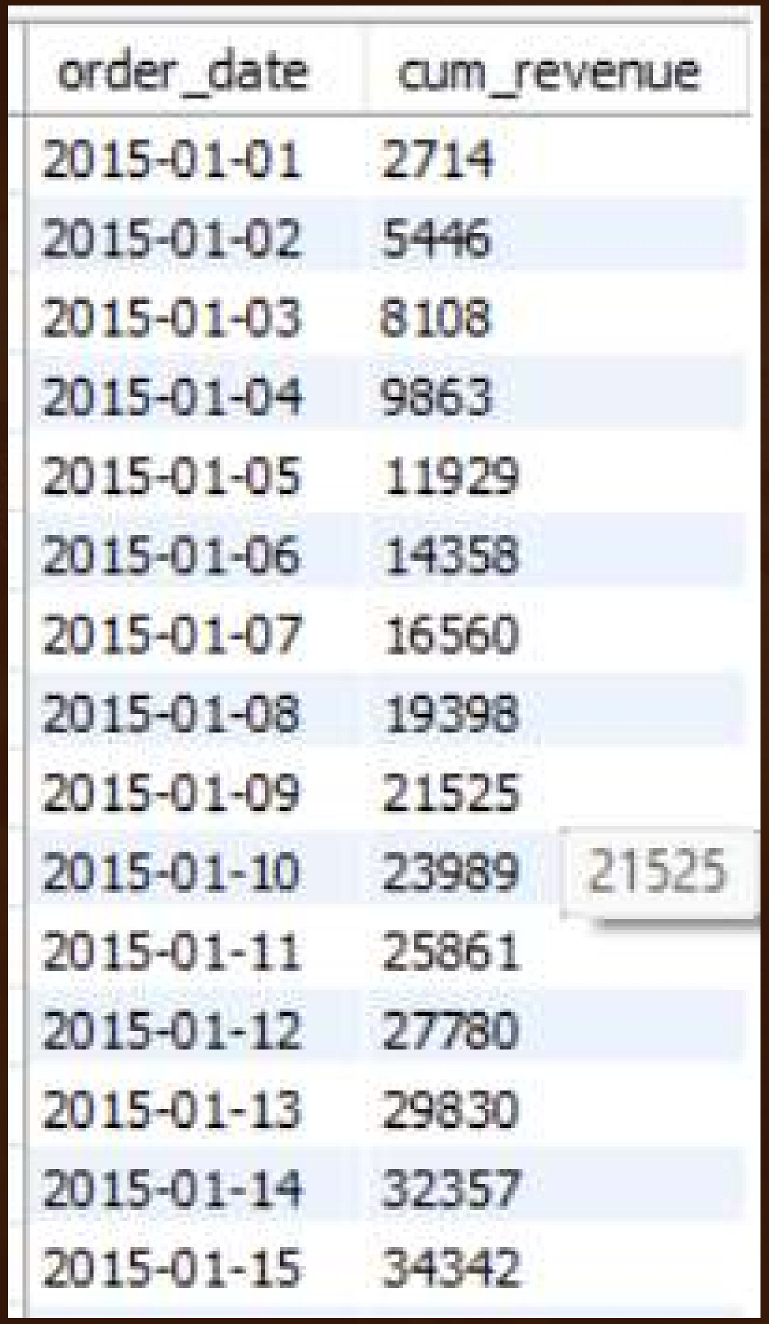
```
SELECT CATEGORY, ROUND(SUM(QUANTITY
* PRICE),2) AS REVENUE
FROM PIZZA_TYPES AS P1
INNER JOIN PIZZAS AS P2
ON P1.PIZZA_TYPE_ID = P2.PIZZA_TYPE_ID
INNER JOIN  ORDER_DETAILS AS O
ON P2.PIZZA_ID = O.PIZZA_ID
GROUP BY CATEGORY;
```



category	Revenue
Classic	10513
Veggie	9586.75
Supreme	10324.4
Chicken	9486.25

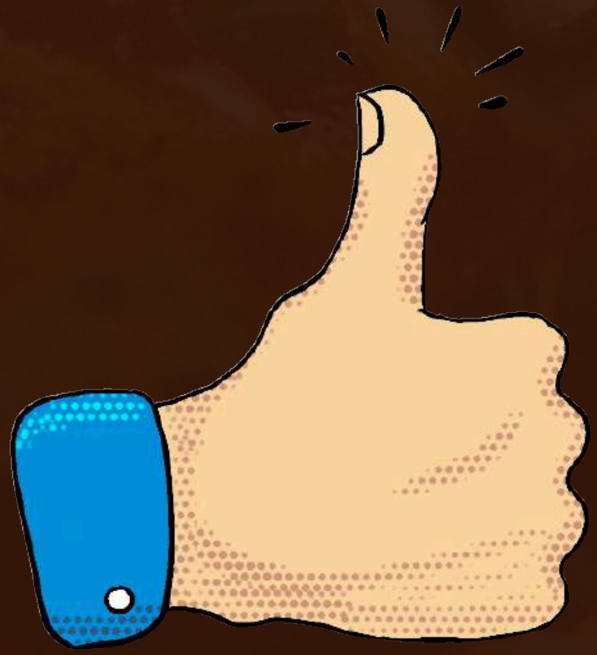
Analyze the cumulative revenue generated over time

```
SELECT  ORDER_DATE, SUM(REVENUE)
OVER(ORDER BY ORDER_DATE) AS
CUM_REVENUE
FROM
(SELECT ORDER_DATE, ROUND(
SUM(QUANTITY * PRICE),0) AS REVENUE
FROM ORDERS AS  O1
INNER JOIN ORDER_DETAILS AS  O2
ON O1.ORDER_ID = O2.ORDER_ID
INNER JOIN PIZZAS AS O3
ON O2.PIZZA_ID = O3.PIZZA_ID
GROUP BY ORDER_DATE) AS SALES;
```



order_date	cum_revenue
2015-01-01	2714
2015-01-02	5446
2015-01-03	8108
2015-01-04	9863
2015-01-05	11929
2015-01-06	14358
2015-01-07	16560
2015-01-08	19398
2015-01-09	21525
2015-01-10	23989
2015-01-11	25861
2015-01-12	27780
2015-01-13	29830
2015-01-14	32357
2015-01-15	34342

THANK YOU FOR ATTENTION



See You Next