Pizza Sales

Delicious Pizza Deals!

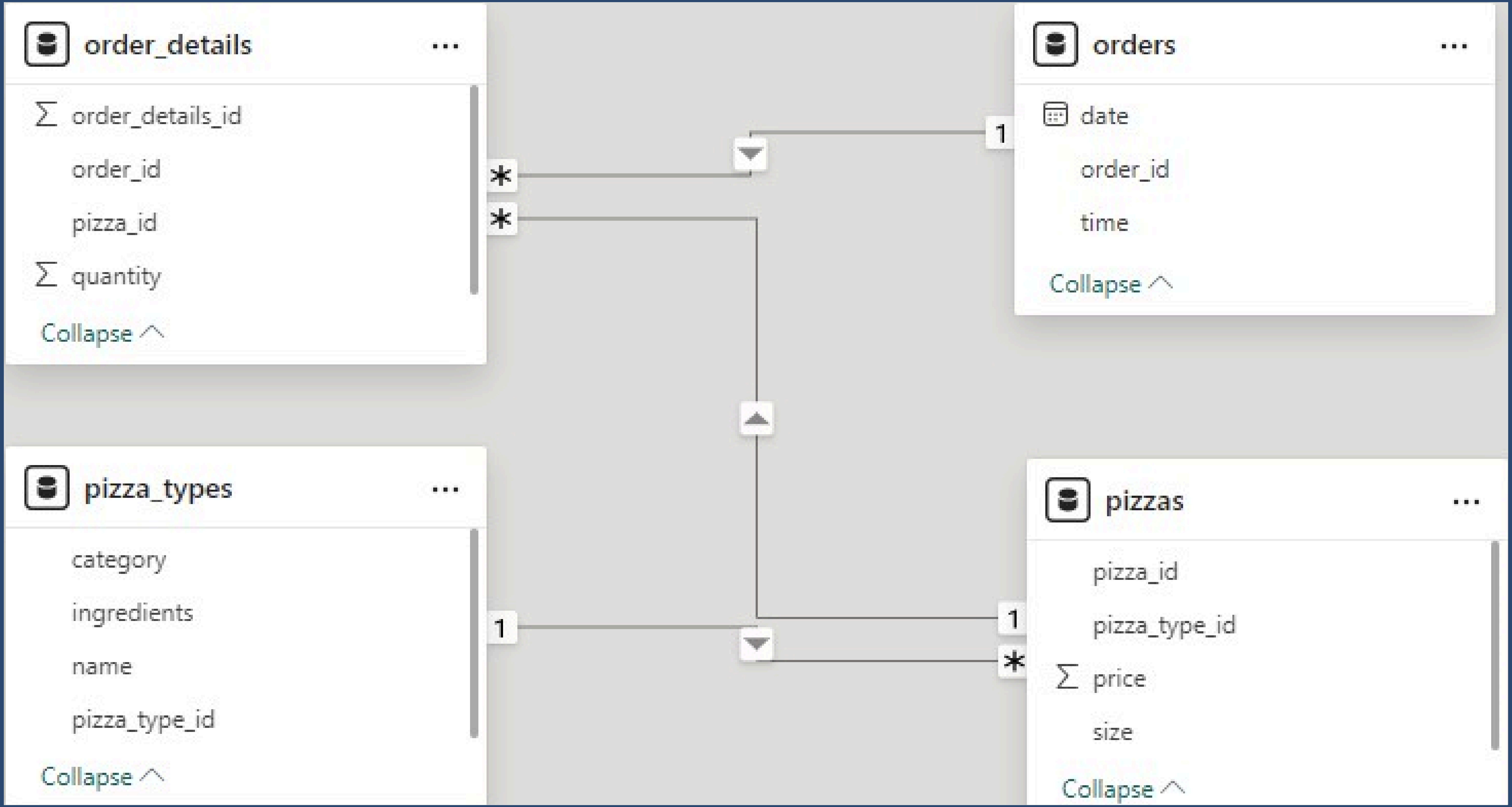
• Introduction

1 In Pizza Sales Analysis project we analyze the dataset and solve the sales related problem by using MySQL .

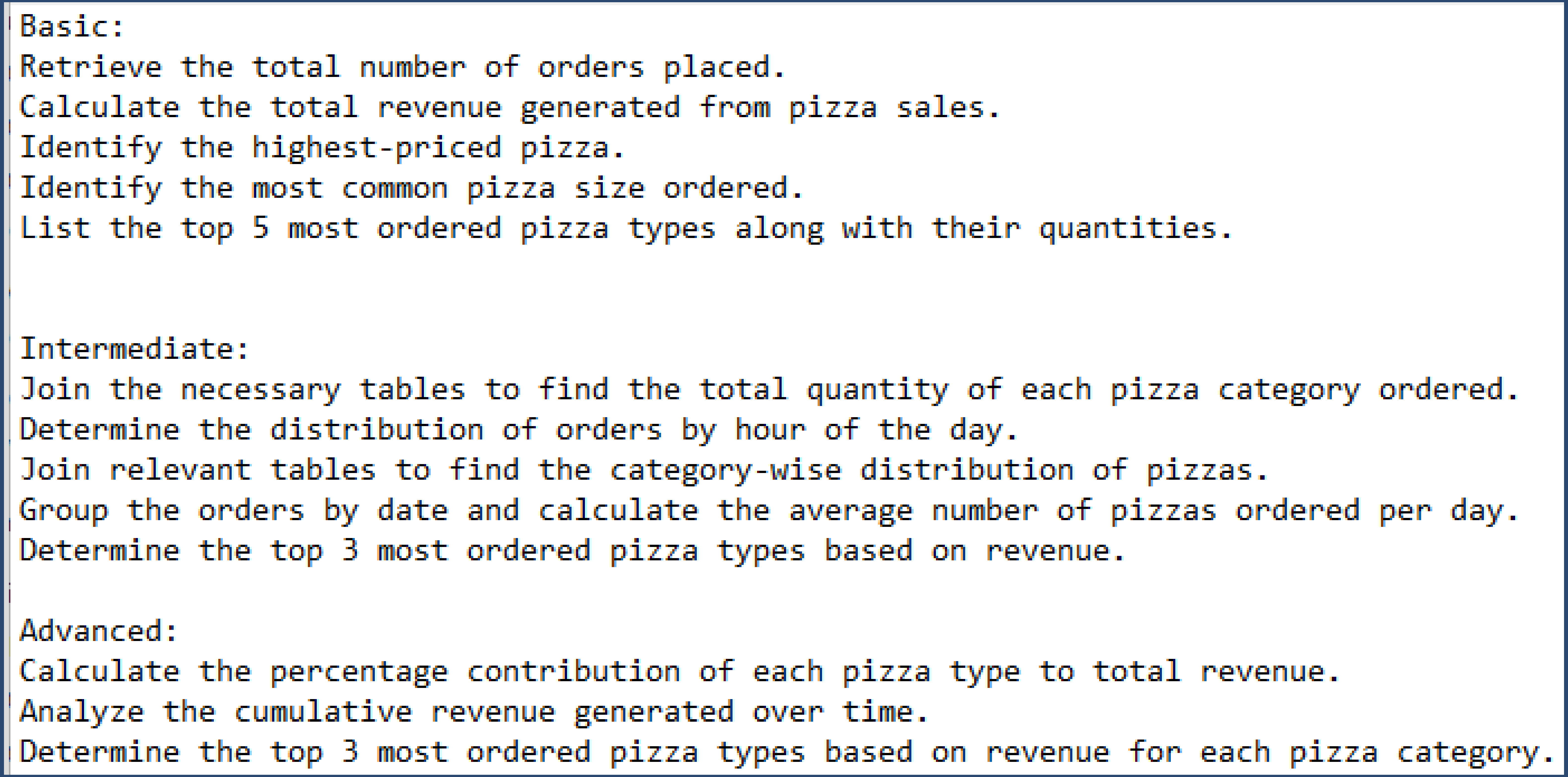
2 In dataset have four tabels order Details, Orders, Pizza Type, and Pizzas .

3 These Four tables are relational to each other.

• Data Modelling



• Questions



Basic:

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.

Intermediate:

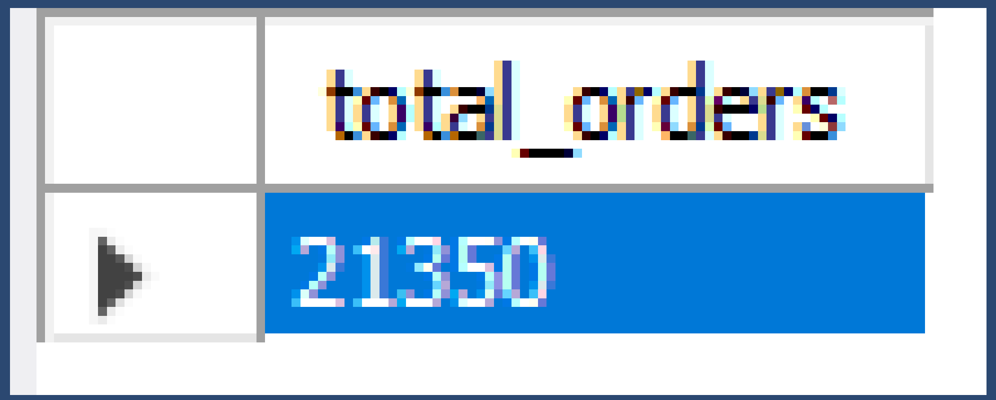
Join the necessary tables to find the total quantity of each pizza category ordered.
Determine the distribution of orders by hour of the day.
Join relevant tables to find the category-wise distribution of pizzas.
Group the orders by date and calculate the average number of pizzas ordered per day.
Determine the top 3 most ordered pizza types based on revenue.

Advanced:

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.

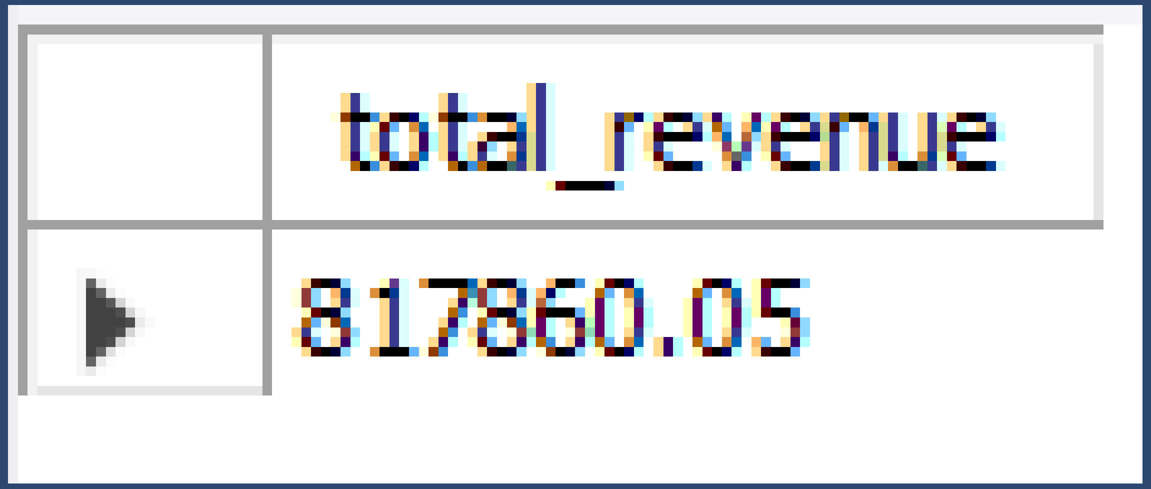
```
SELECT
    COUNT(order_id) AS total_orders
FROM
    orders;
```



	total_orders
▶	21350

Calculate the total revenue generated from pizza sales.

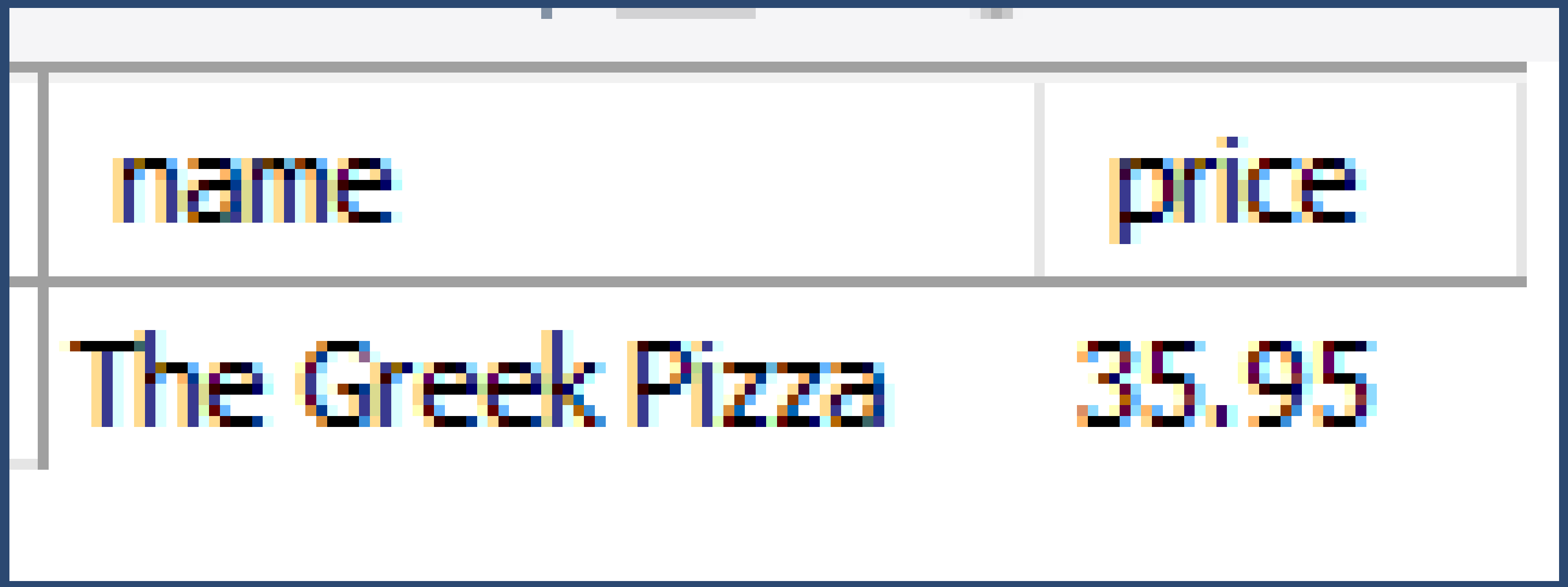
```
SELECT
    ROUND(SUM(order_details.quantity *
pizzas.price),
            2) AS total_revenue
FROM
    order_details
        JOIN
    pizzas ON order_details.pizza_id =
pizzas.pizza_id;
```



	total_revenue
▶	817860.05

Identify the highest-priced pizza.

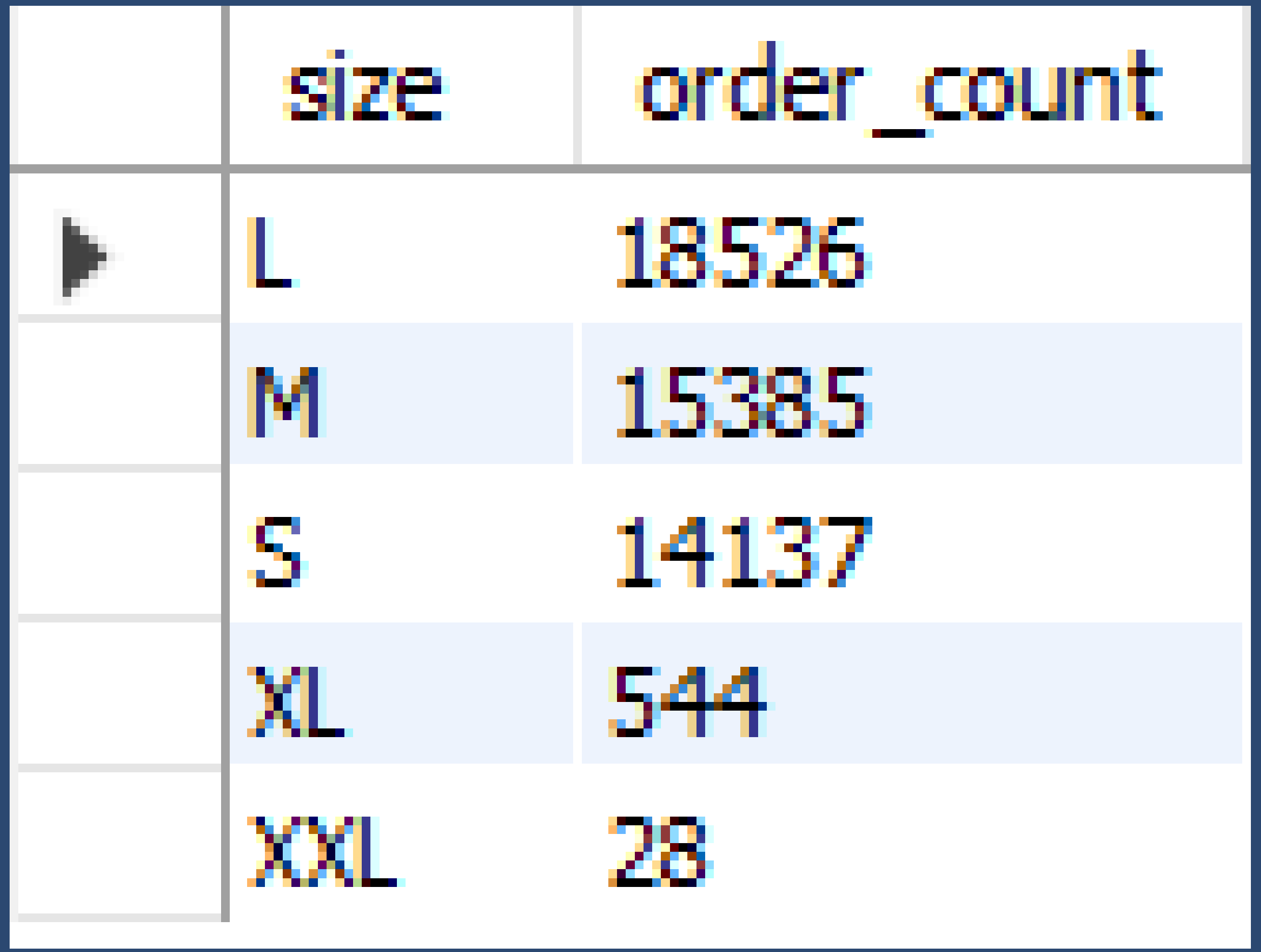
```
SELECT
    pizza_types.name, pizzas.price
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id =
pizzas.pizza_type_id
ORDER BY pizzas.price DESC
LIMIT 1;
```



name	price
The Greek Pizza	35.95

Identify the most common pizza size ordered.

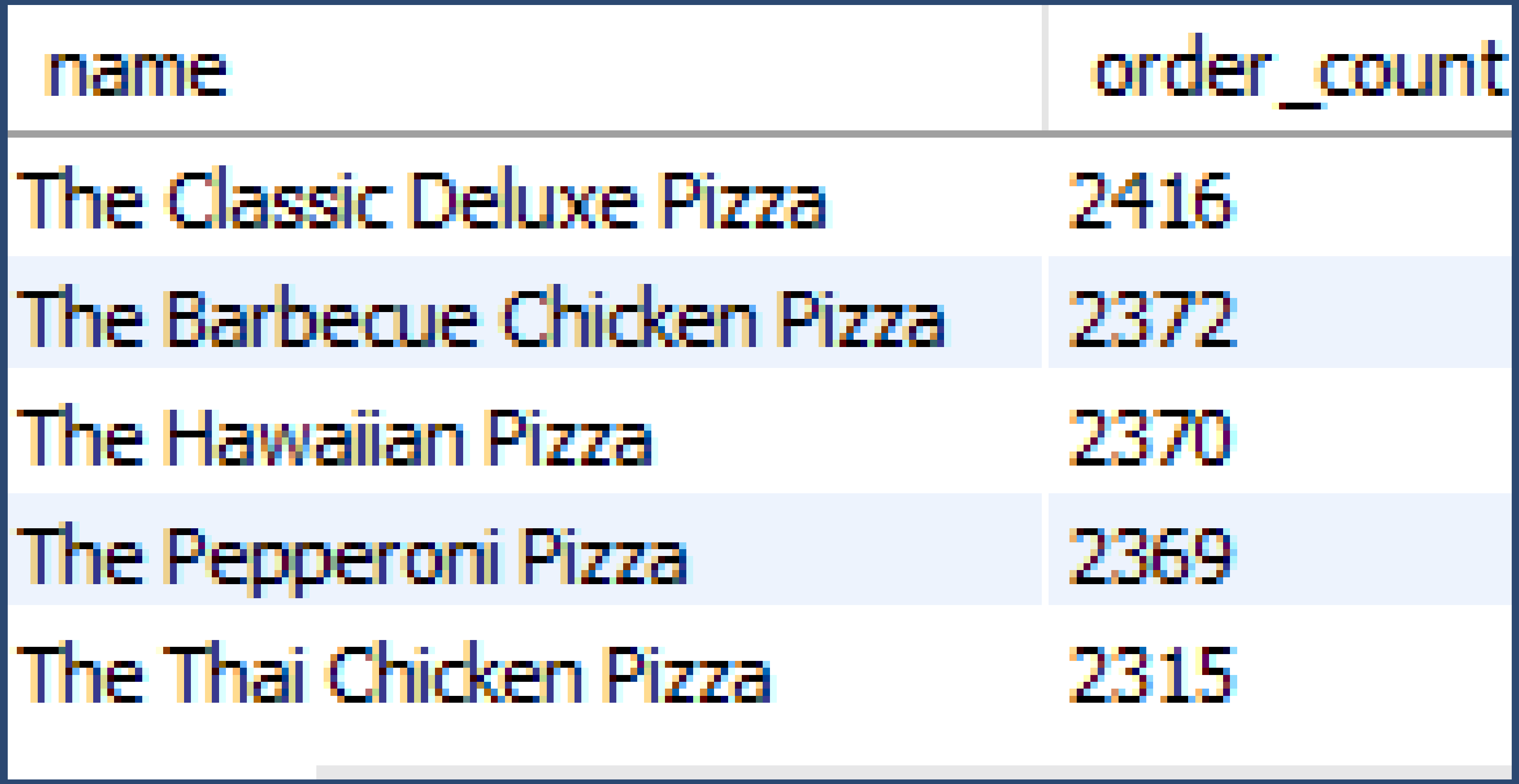
```
SELECT
    pizzas.size, COUNT(order_details.order_id) AS
order_count
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id =
order_details.pizza_id
GROUP BY pizzas.size
ORDER BY order_count DESC;
```



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.

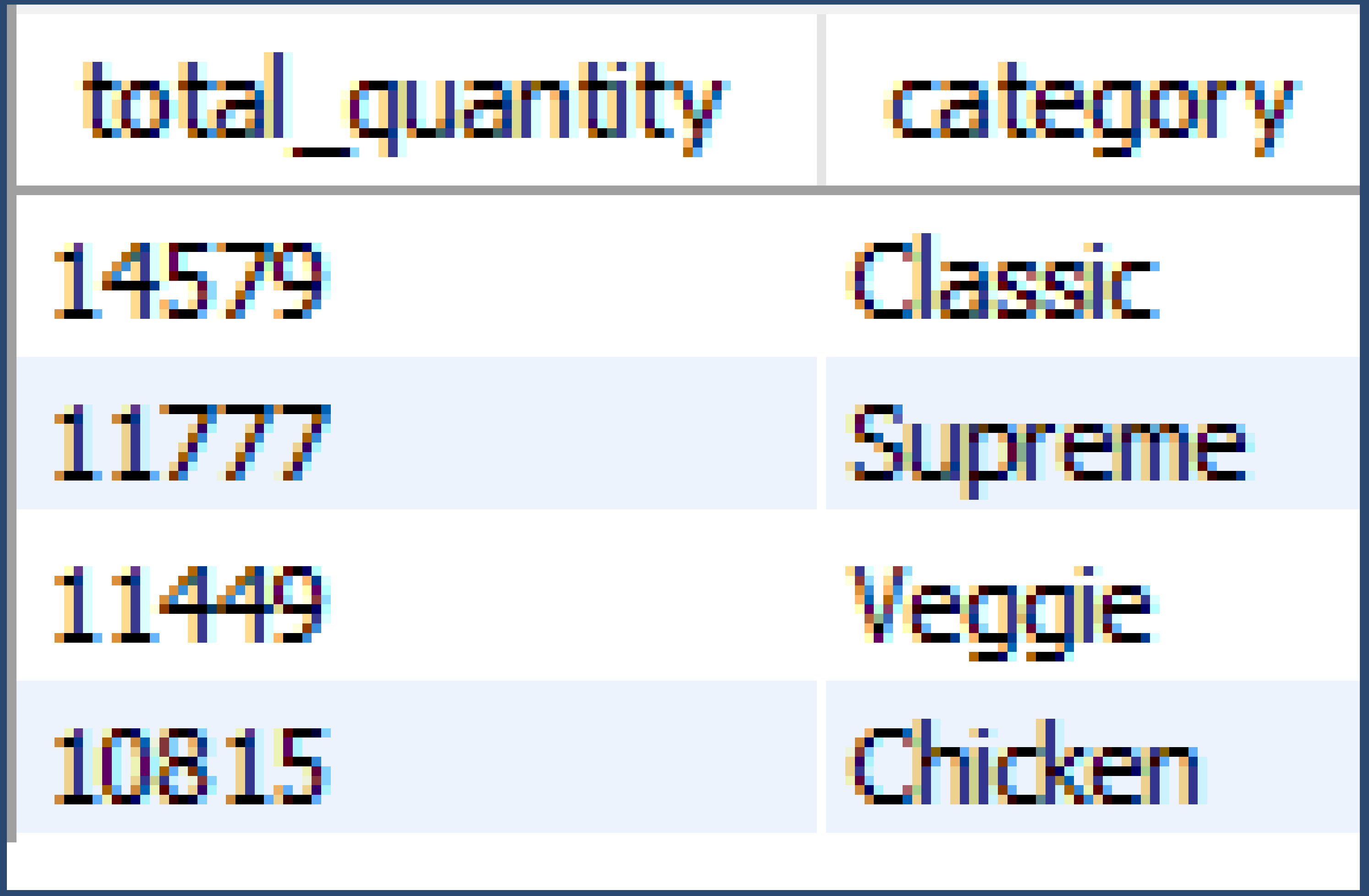
```
select
pizza_types.name,count(order_details.quantity) as
order_count
from pizza_types join pizzas
on pizza_types.pizza_type_id =
pizzas.pizza_type_id
join order_details
on order_details.pizza_id = pizzas.pizza_id
group by pizza_types.name
order by order_count desc limit 5;
```



name	order_count
The Classic Deluxe Pizza	2416
The Barbecue Chicken Pizza	2372
The Hawaiian Pizza	2370
The Pepperoni Pizza	2369
The Thai Chicken Pizza	2315

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

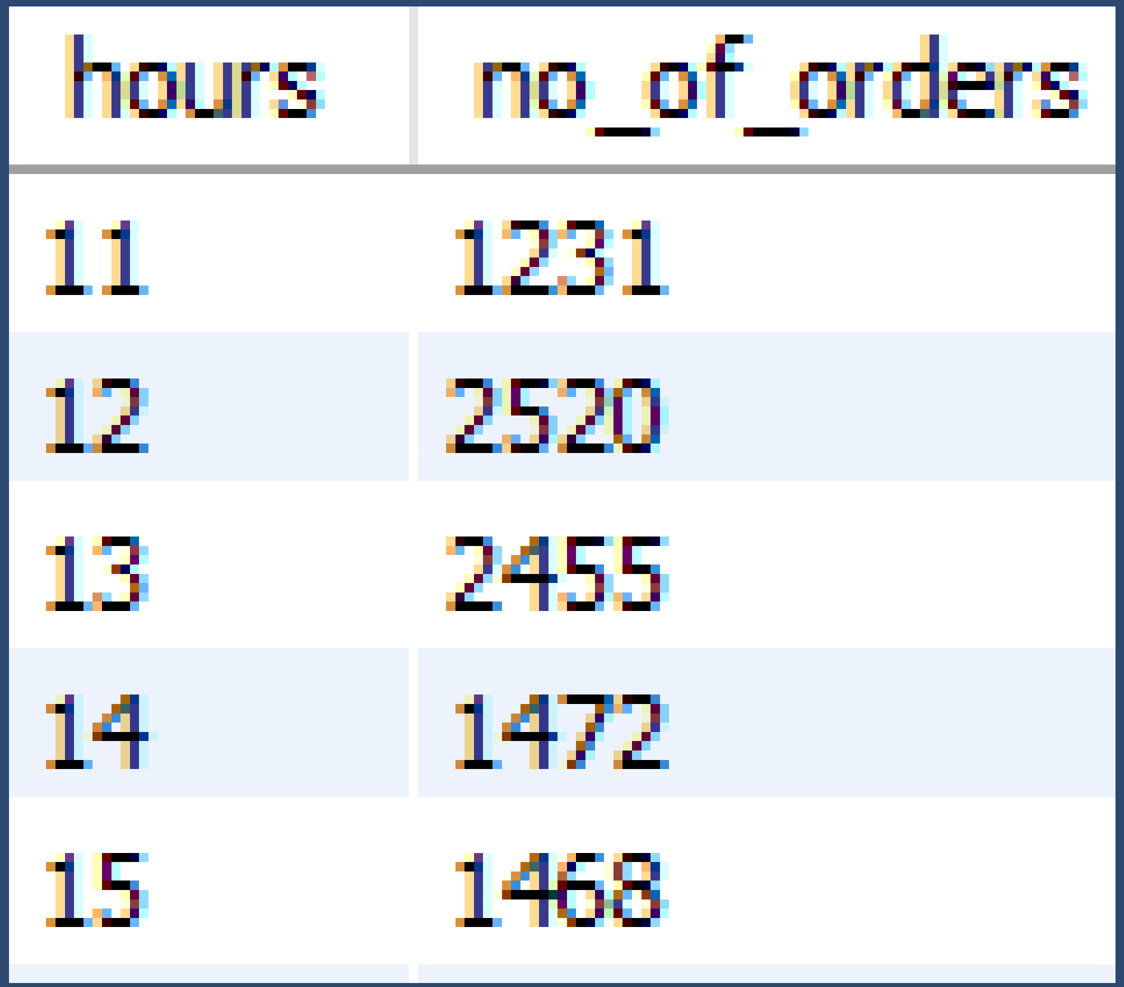
```
SELECT
    COUNT(order_details.quantity * pizzas.price) AS
total_quantity,
    pizza_types.category
FROM
    order_details
        JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id
        JOIN
    pizza_types ON pizza_types.pizza_type_id =
pizzas.pizza_type_id
GROUP BY pizza_types.category
ORDER BY total_quantity DESC;
```



total_quantity	category
14579	Classic
11777	Supreme
11449	Veggie
10815	Chicken

Determine the distribution of orders by hour of the day.

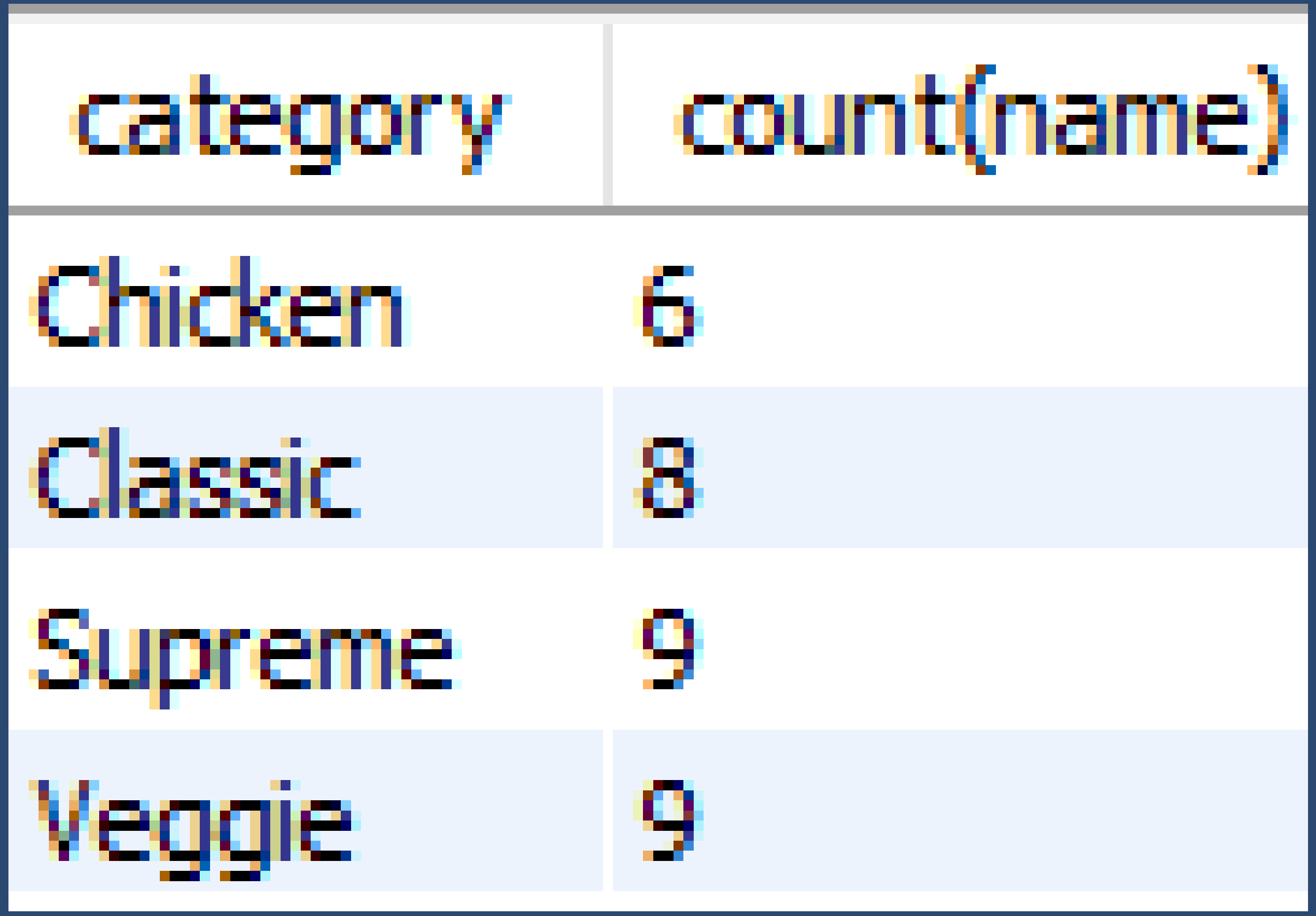
```
SELECT
    HOUR(order_time) AS hours,
COUNT(order_id) AS no_of_orders
FROM
    orders
GROUP BY hours;
```



hours	no_of_orders
11	1231
12	2520
13	2455
14	1472
15	1468

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

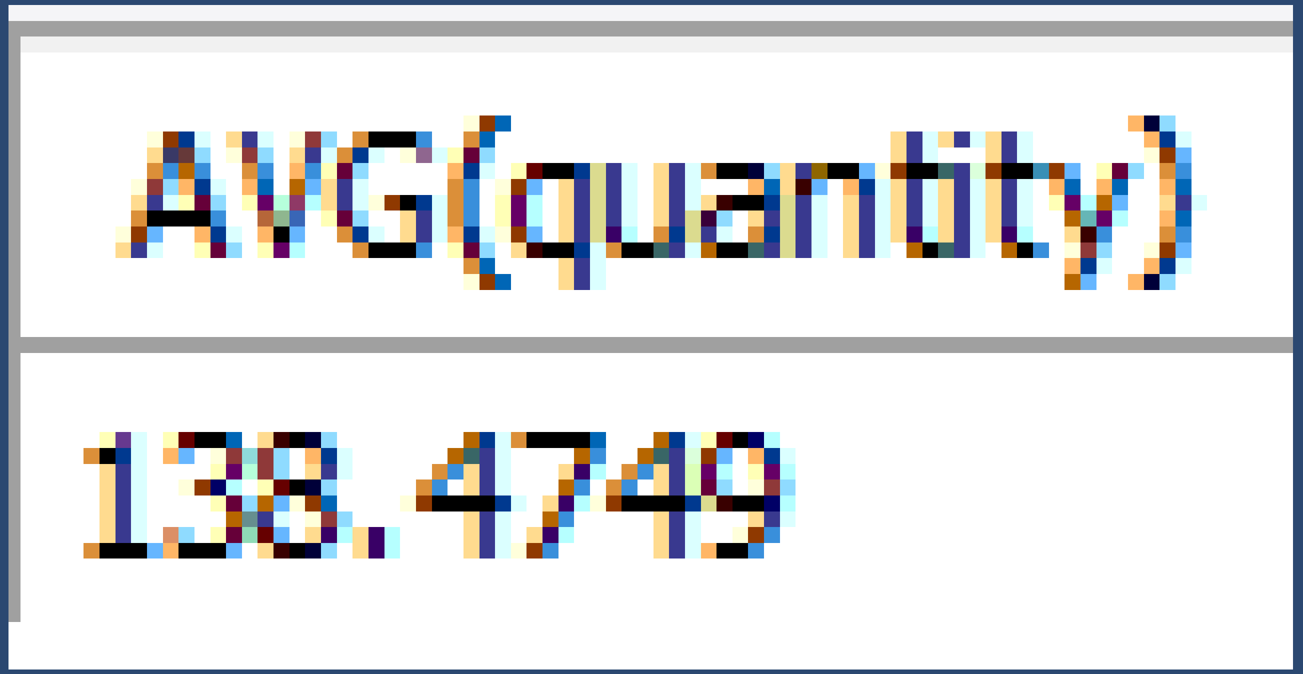
```
select category,
count(name) from
pizza_types group by
category;
```



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.

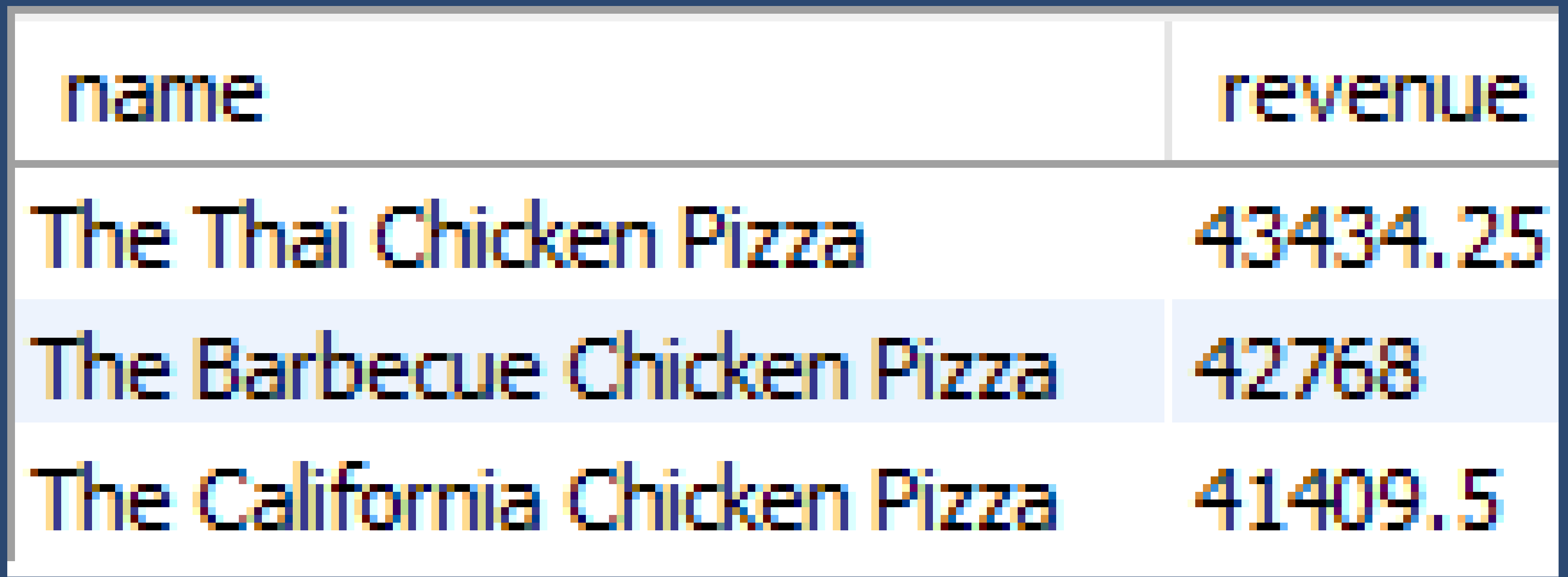
```
SELECT
    AVG(quantity)
FROM
    (SELECT
        orders.order_date,
SUM(order_details.quantity) as quantity
    FROM
        orders
    JOIN order_details ON orders.order_id =
order_details.order_id
    GROUP BY orders.order_date) AS
order_quantity;
```



```
AVG(quantity)
138.4749
```

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

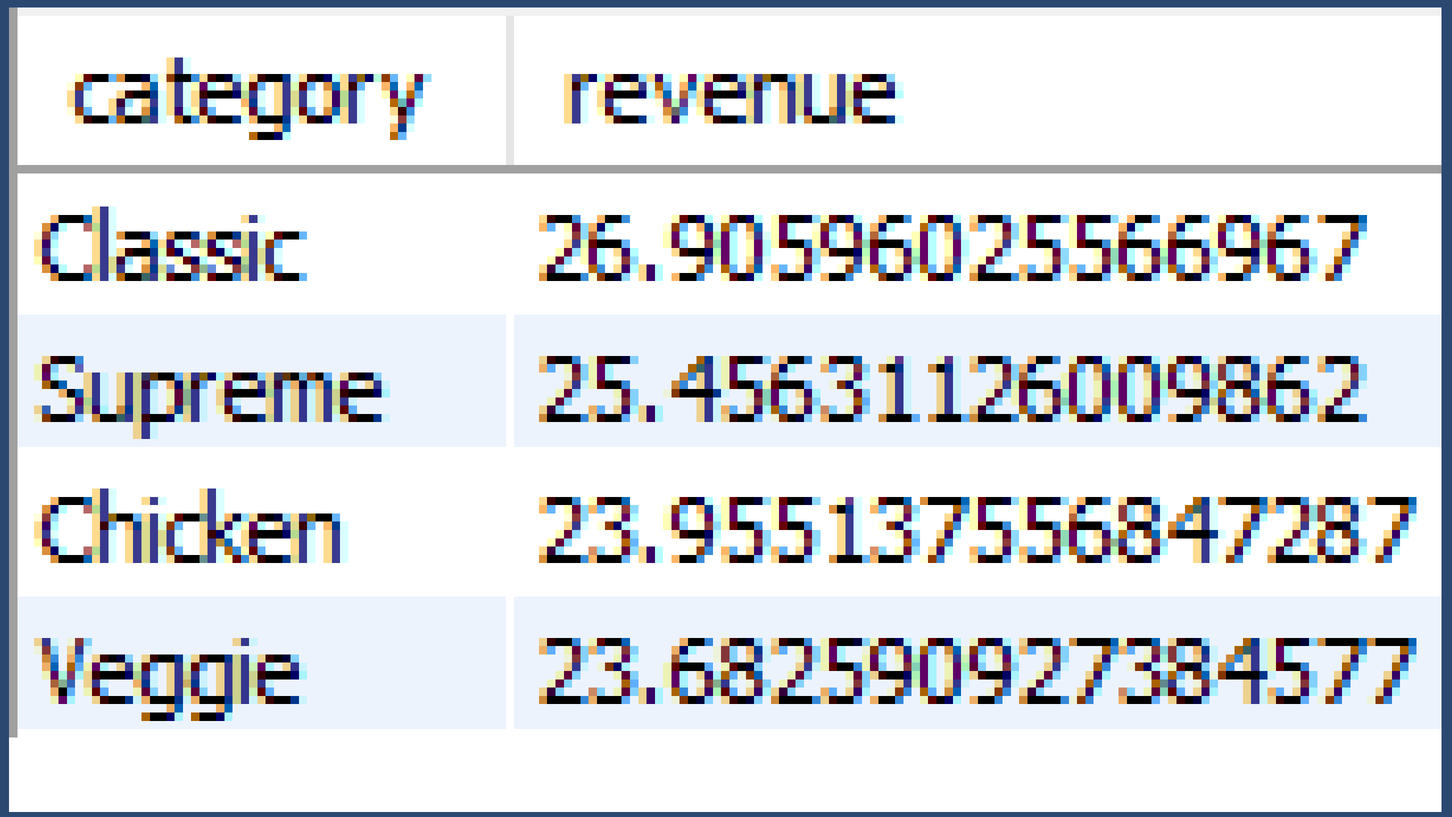
```
select pizza_types.name,
sum(order_details.quantity*pizzas.price) as
revenue
from pizza_types join pizzas
on pizza_types.pizza_type_id =
pizzas.pizza_type_id
join order_details
on order_details.pizza_id = pizzas.pizza_id
group by pizza_types.name
order by revenue desc
limit 3;
```



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.

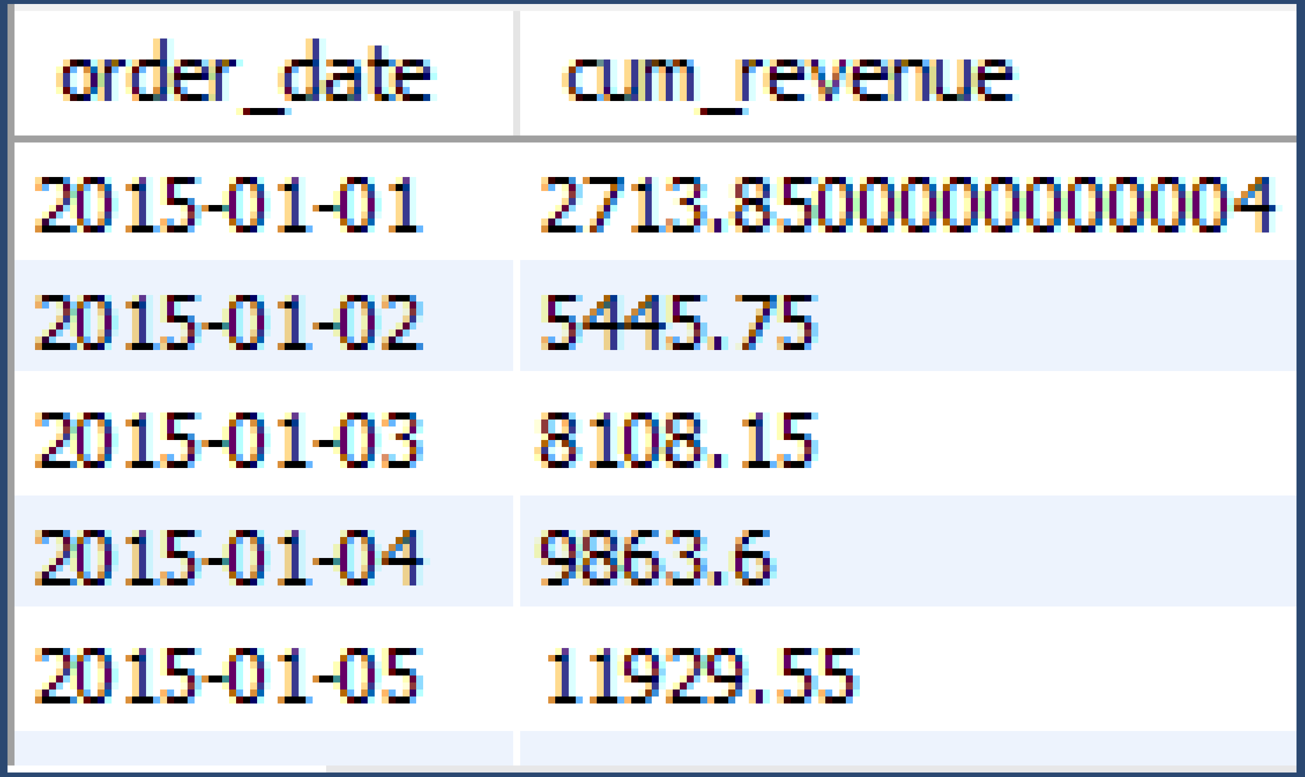
```
select pizza_types.category,
(sum(order_details.quantity*pizzas.price)/
(select round(sum(order_details.quantity*pizzas.price),2)
as total_revenue
from order_details join pizzas
on order_details.pizza_id=pizzas.pizza_id))*100 as
revenue
from pizza_types join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
join order_details
on order_details.pizza_id = pizzas.pizza_id
group by pizza_types.category
order by revenue desc;
```



category	revenue
Classic	26.90596025566967
Supreme	25.45631126009862
Chicken	23.955137556847287
Veggie	23.682590927384577

Analyze the cumulative revenue generated over time.

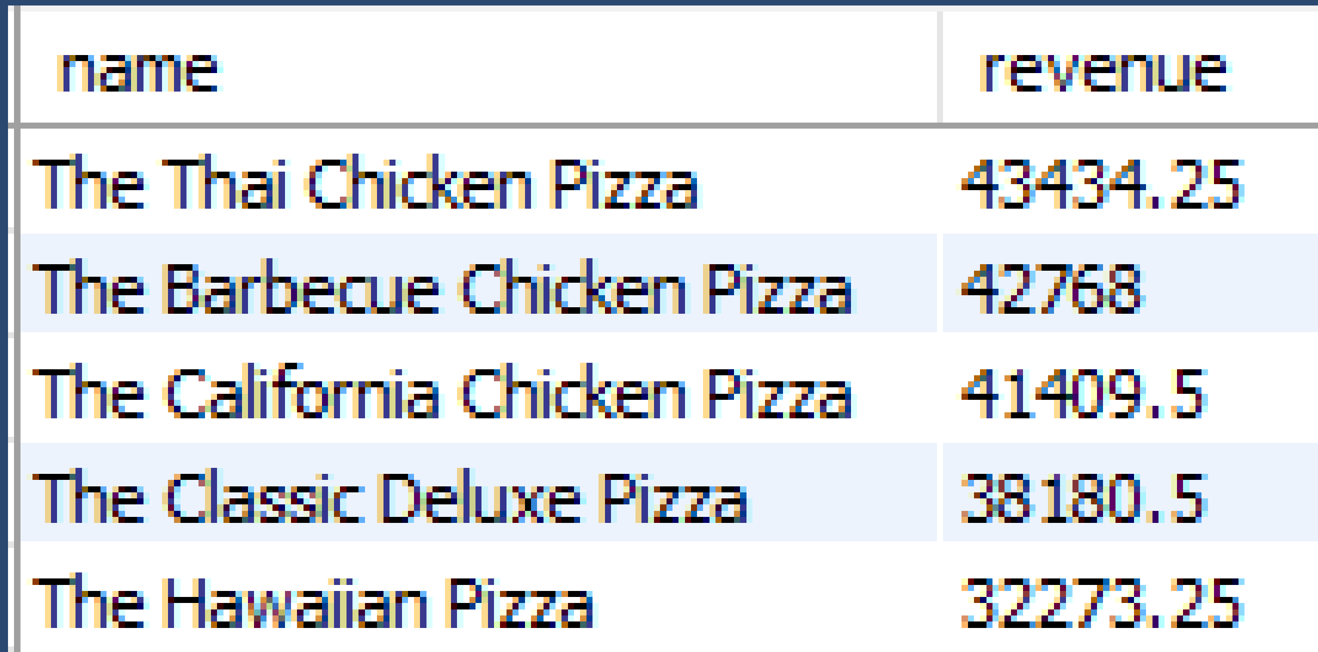
```
select order_date,
sum(revenue) over (order by order_date) as
cum_revenue
from
(select orders.order_date,
sum(order_details.quantity*pizzas.price) as
revenue
from orders join order_details
on orders.order_id = order_details.order_id
join pizzas
on order_details.pizza_id = pizzas.pizza_id
group by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



order_date	cum_revenue
2015-01-01	2713.8500000000004
2015-01-02	5445.75
2015-01-03	8108.15
2015-01-04	9863.6
2015-01-05	11929.55

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

```
select name, revenue from
(select name, category, revenue, rank() over(partition by
category order by revenue desc) as rn
from
(select pizza_types.name , pizza_types.category,
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;
```



name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5
The Classic Deluxe Pizza	38180.5
The Hawaiian Pizza	32273.25

Thank you