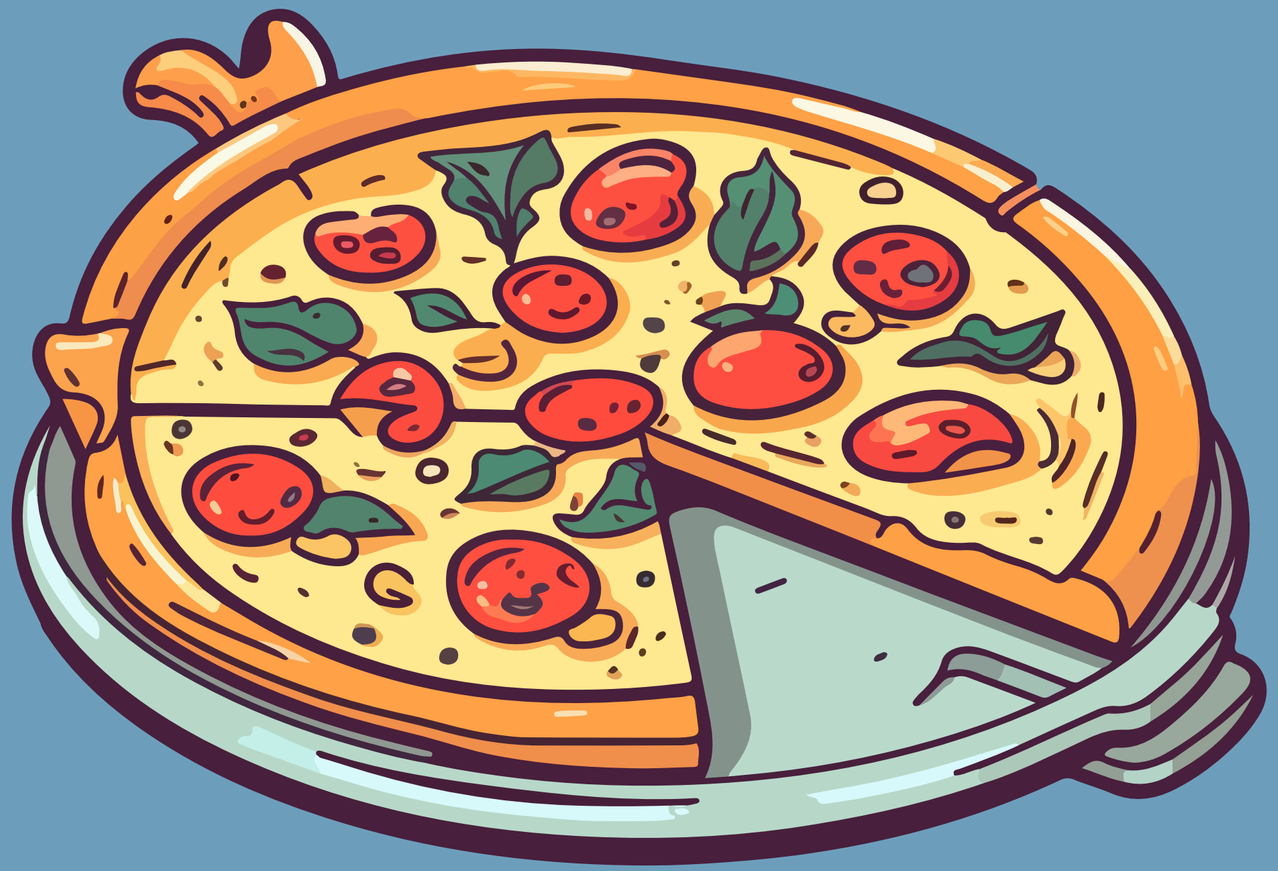
# PIZZA SALE



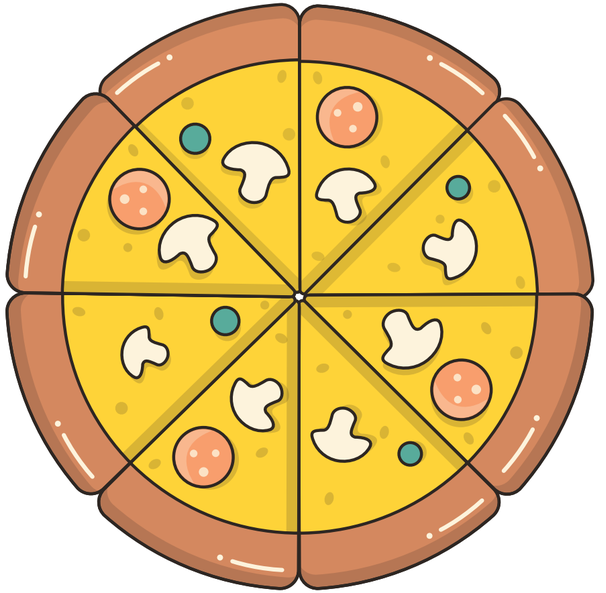
# Schema

| Pizzas |
| --- |
| pizza_id |
| pizza_type_id |
| size |
| price |

| Pizza_types |
| --- |
| pizza_type_id |
| name |
| category |
| ingredients |

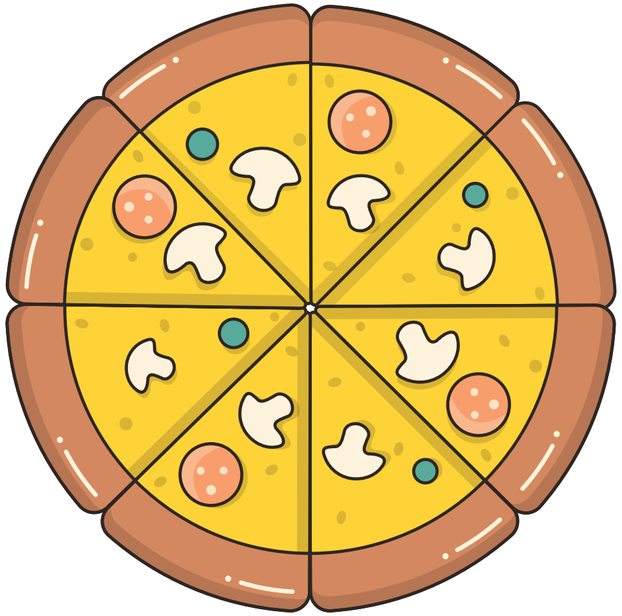
| Orders |
| --- |
| order_id |
| orders_date |
| orders_time |

| Order_details |
| --- |
| order_details_id |
| order_id |
| pizza_id |
| quantity |

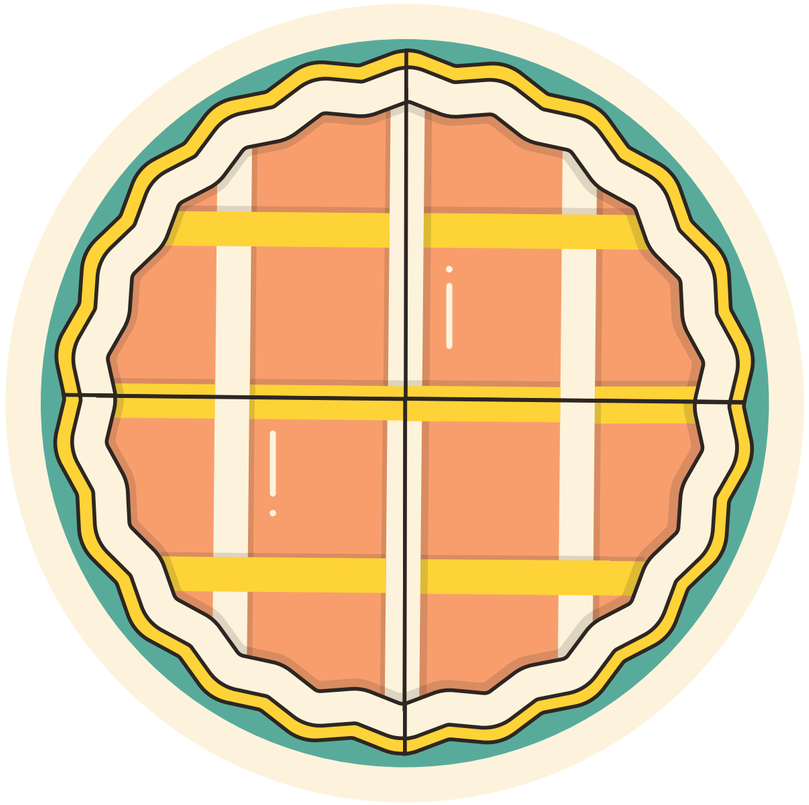


# QUESTION

- 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.



# RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.



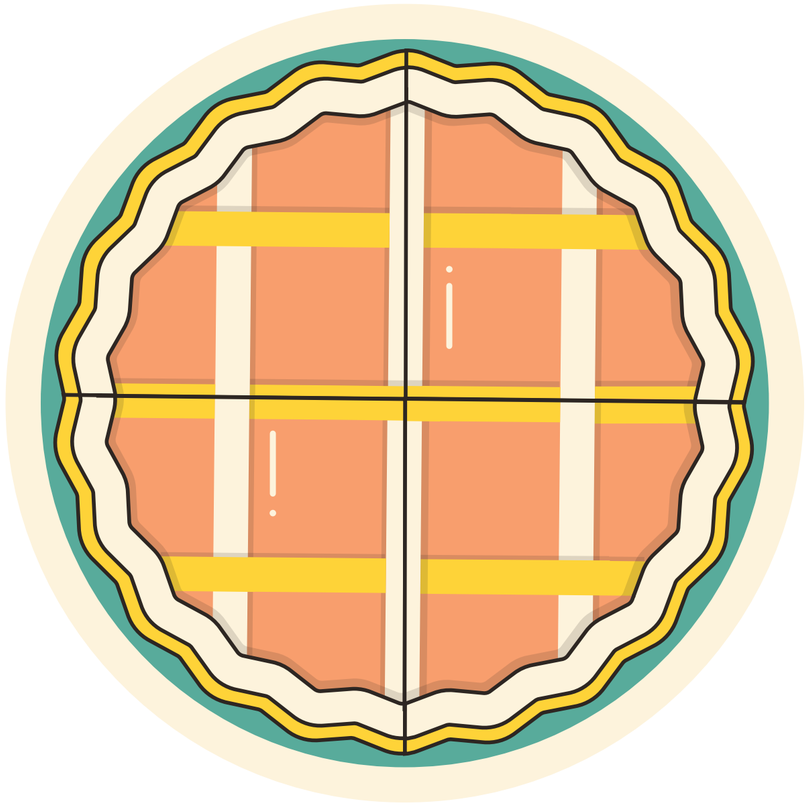
```
--Retrieve the total number of orders placed.
select count(order_id) as total_order_placed
from orders
```

Data Output | Messages | Notifications

SQL

| | total_order_placed bigint |
|---|---|
| 1 | 21350 |

# CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.



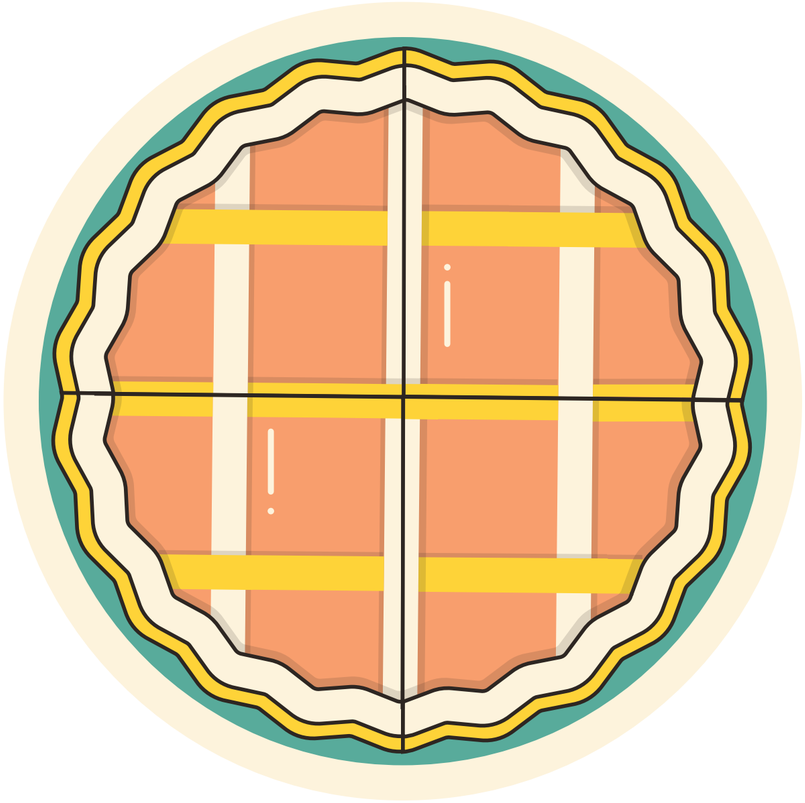
```
73
74  --Calculate the total revenue generated from pizza sales.
75
76  select *from order_details
77  select *from pizzas
78
79  select sum(pizzas.price*order_details.quantity) as total_revenue
80  from pizzas
81  join order_details on pizzas.pizza_id = order_details.pizza_id;
82
83
84
85
86
87
88
```

Data Output | Messages | Notifications

SQL

| | total_revenue<br>double precision |
|---|---|
| 1 | 817860.0508384705 |

# IDENTIFY THE HIGHEST-PRICED PIZZA.



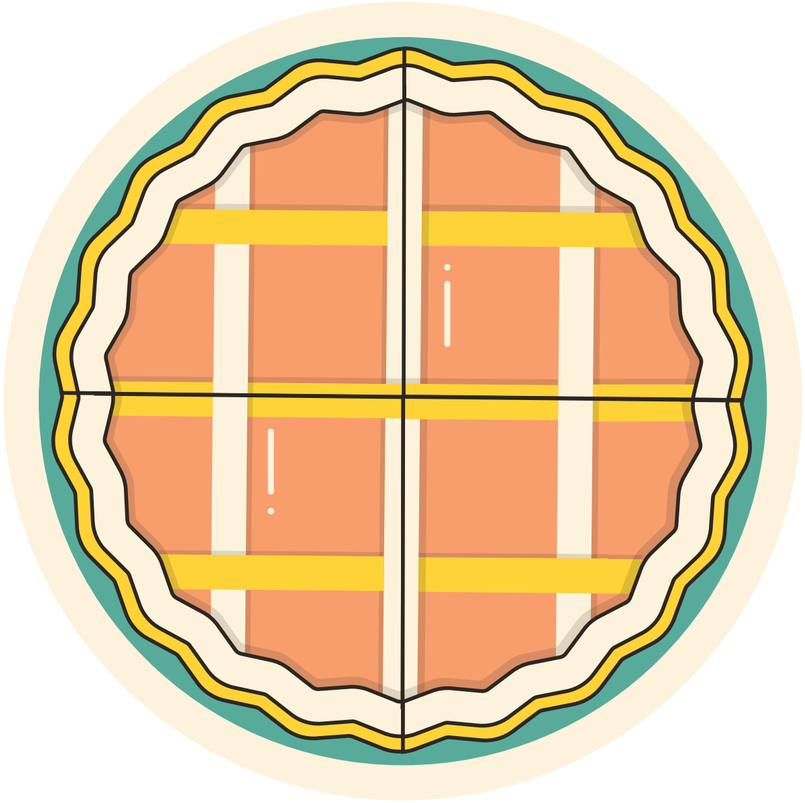
```
82
83  --Identify the highest-priced pizza.
84
85  select *from pizzas
86
87  select pizzas.price as highest_price_pizza, pizza_types.name, pizza_types.category
88  from pizzas
89  join pizza_types on pizzas.pizza_type_id = pizza_types.pizza_type_id
90  group by pizza_types.name, pizza_types.category, highest_price_pizza
91  Order by highest_price_pizza desc
92  limit 1
93
94
95
96
97
98
99
100
101
102
```

Data Output | Messages | Notifications

SQL

| | highest_price_pizza real | name character varying (50) | category character varying (50) |
|---|---|---|---|
| 1 | 35.95 | The Greek Pizza | Classic |

# IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.



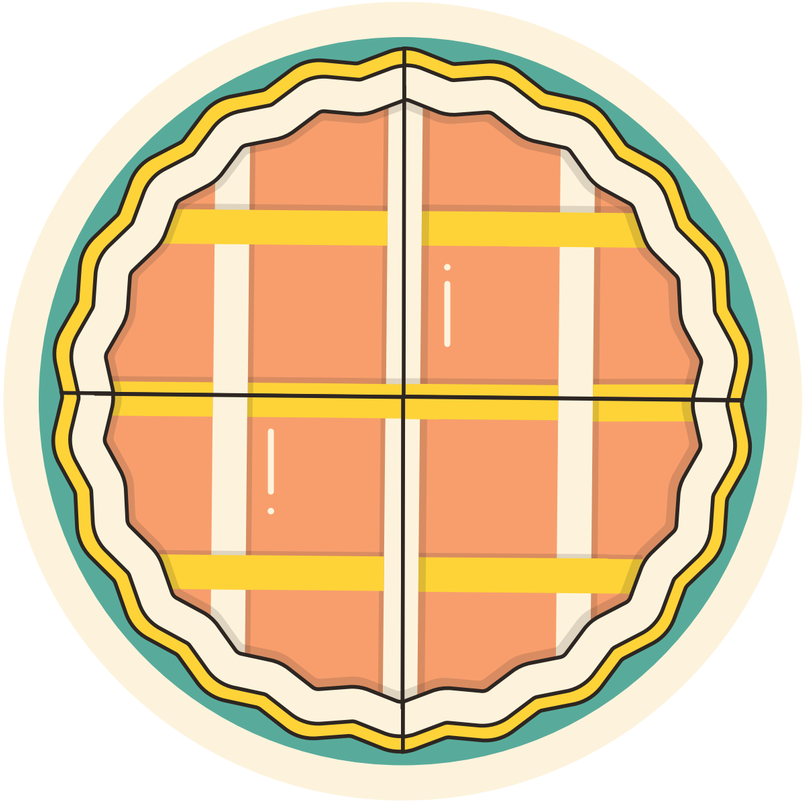
```sql
--Identify the most common pizza size ordered.

select *from order_details

select pizzas.size, count(order_details.quantity) as total_count_pizza
from order_details
join pizzas on order_details.pizza_id = pizzas.pizza_id
group by pizzas.size
order by total_count_pizza desc
LIMIT 1


select *from pizzas
```

Data Output | Messages | Notifications

SQL

| | size character (8) | total_count_pizza bigint |
|---|---|---|
| 1 | L | 18526 |

# LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES.

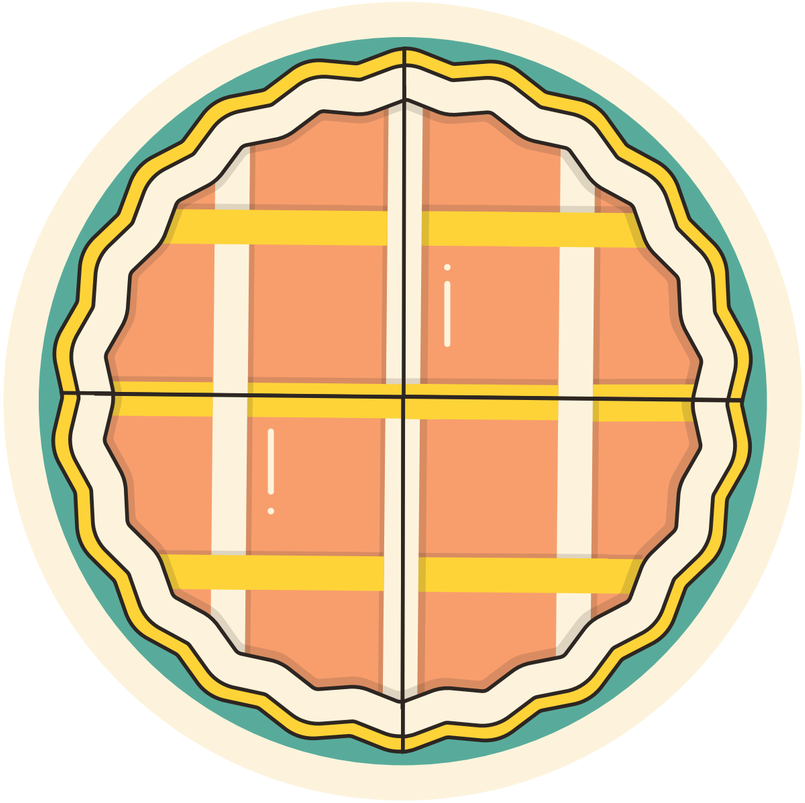


```sql
--List the top 5 most ordered pizza types along with their quantities.

select pizza_types.name, sum(order_details.quantity) as total_count_pizzas
from pizzas
join pizza_types 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_count_pizzas desc
limit 5
```

Data Output | Messages | Notifications

| | name<br>character varying (50) | total_count_pizzas<br>numeric |
|---|---|---|
| 1 | The Classic Deluxe Pizza | 2453 |
| 2 | The Barbecue Chicken Pizza | 2432 |
| 3 | The Hawaiian Pizza | 2422 |
| 4 | The Pepperoni Pizza | 2418 |
| 5 | The Thai Chicken Pizza | 2371 |

# JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED.



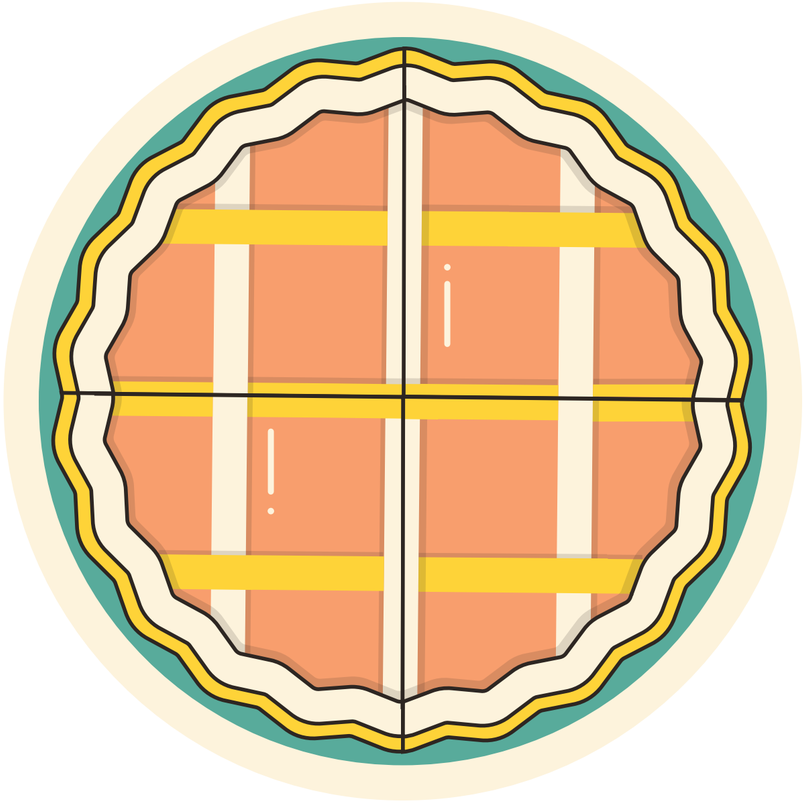
```
116
117  --Join the necessary tables to find the total quantity of each pizza category ordered.
118
119  select pizza_types.category, sum(order_details.quantity) as total_count_pizzas
120  from pizzas
121  join pizza_types on pizzas.pizza_type_id = pizza_types.pizza_type_id
122  join order_details on pizzas.pizza_id = order_details.pizza_id
123  group by  pizza_types.category
124  order by total_count_pizzas
125
126
127
```

Data Output | Messages | Notifications

SQL

| | category character varying (50) | total_count_pizzas numeric |
|---|---|---|
| 1 | Chicken | 11050 |
| 2 | Veggie | 11649 |
| 3 | Supreme | 11987 |
| 4 | Classic | 14888 |

# DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY

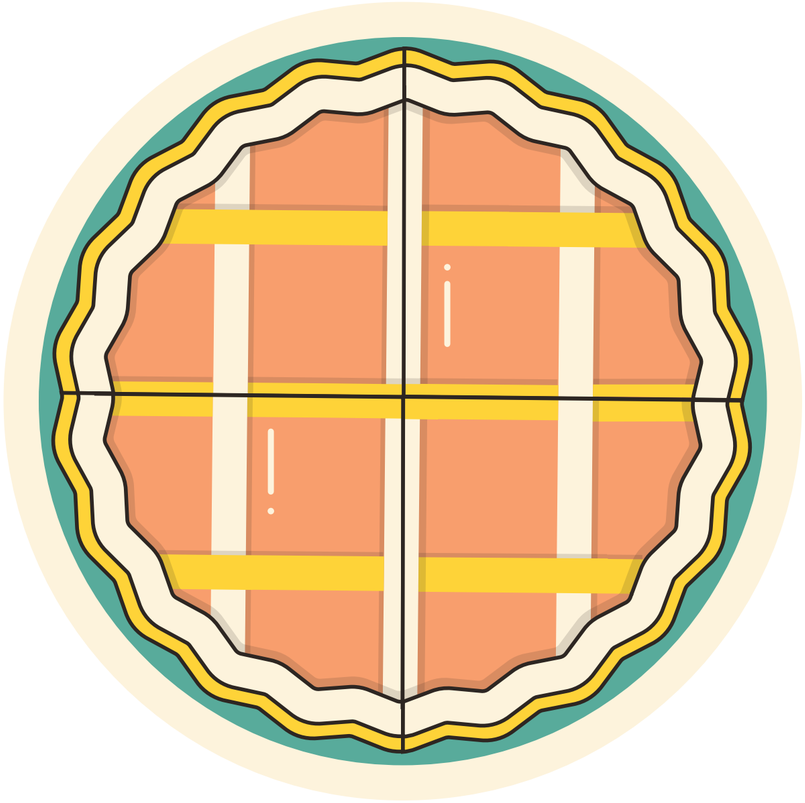


```sql
--Determine the top 3 most ordered pizza types based on revenue.

select * from pizza_types

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

Data Output | Messages | Notifications

| | name<br>character varying (50) | total_revenue<br>double precision |
|---|---|---|
| 1 | The Thai Chicken Pizza | 43434.25 |
| 2 | The Barbecue Chicken Pizza | 42768 |
| 3 | The California Chicken Pizza | 41409.5 |

# JOIN RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS.



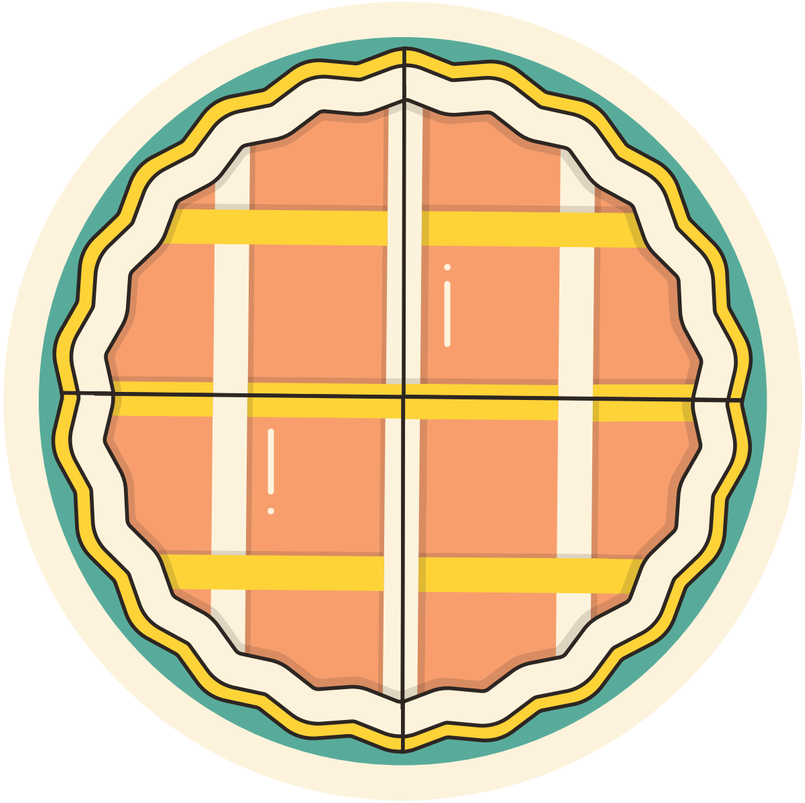
```
140
141  --Group the orders by date and calculate the average number of pizzas ordered per day.
142
143  select round(avg(pizzas_per_day),0) from(
144  select orders.orders_date, sum(order_details.quantity) as pizzas_per_day
145  from orders
146  join order_details on orders.order_id = order_details.order_id
147  group by orders_date
148  order by orders_date
149  );
150
151
```

Data Output | Messages | Notifications

SQL

| | round numeric |
|---|---|
| 1 | 138 |

# JOIN RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS.



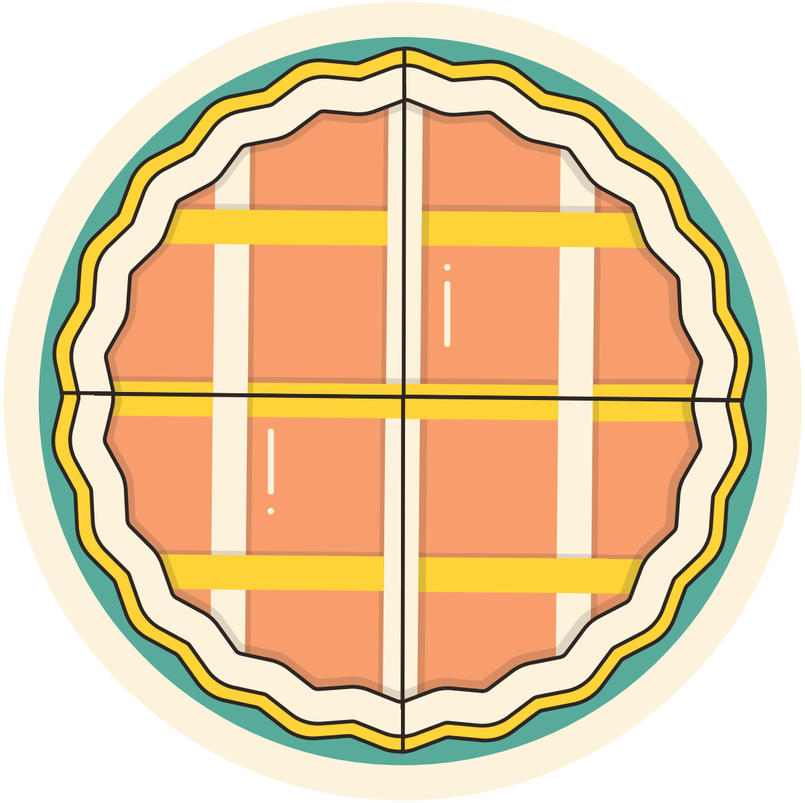
```
140
141  --Group the orders by date and calculate the average number of pizzas ordered per day.
142
143  select round(avg(pizzas_per_day),0) from(
144  select orders.orders_date, sum(order_details.quantity) as pizzas_per_day
145  from orders
146  join order_details on orders.order_id = order_details.order_id
147  group by orders_date
148  order by orders_date
149  );
150
151
```

Data Output | Messages | Notifications

SQL

| | round numeric |
|---|---|
| 1 | 138 |

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



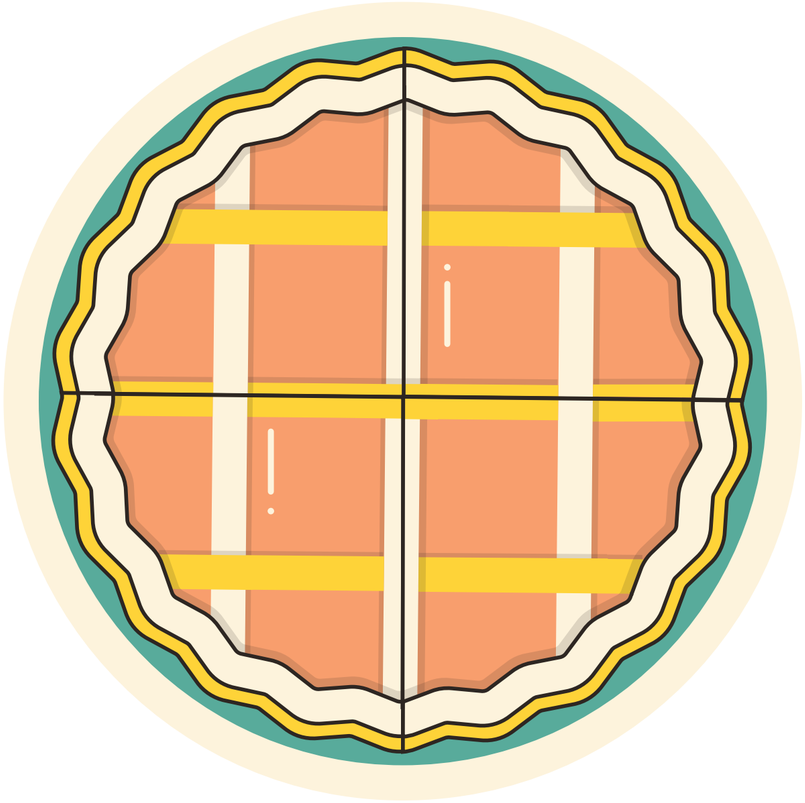
```
143
144  --Group the orders by date and calculate the average number of pizzas ordered per day.
145
146  select round(avg(pizzas_per_day),0) from(
147  select orders.orders_date, sum(order_details.quantity) as pizzas_per_day
148  from orders
149  join order_details on orders.order_id = order_details.order_id
150  group by orders_date
151  order by orders_date
152  );
153
```

Data Output | Messages | Notifications

SQL

| | round numeric |
|---|---|
| 1 | 138 |

# DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE



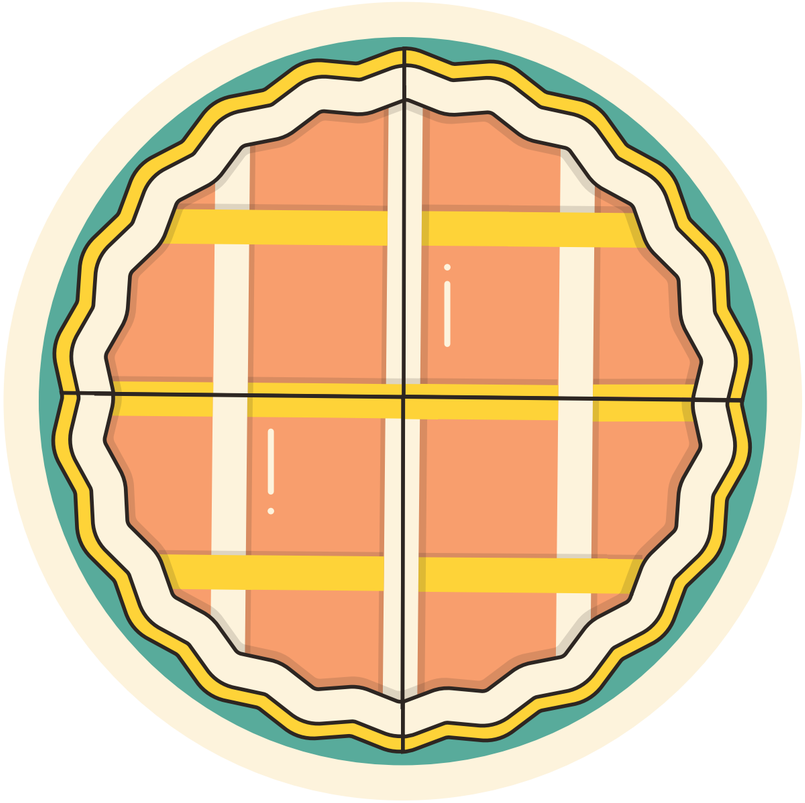
```
156
157  --Determine the top 3 most ordered pizza types based on revenue.
158
159  select * from pizza_types
160
161  select pizza_types.name, sum(pizzas.price*order_details.quantity) as total_revenue
162  from pizzas
163  join order_details on pizzas.pizza_id = order_details.pizza_id
164  join pizza_types on pizzas.pizza_type_id = pizza_types.pizza_type_id
165  group by pizza_types.name
166  order by total_revenue desc
167  limit 3;
168
```

Data Output | Messages | Notifications

SQL

| | name<br>character varying (50) | total_revenue<br>double precision |
|---|---|---|
| 1 | The Thai Chicken Pizza | 43434.25 |
| 2 | The Barbecue Chicken Pizza | 42768 |
| 3 | The California Chicken Pizza | 41409.5 |

# CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE.



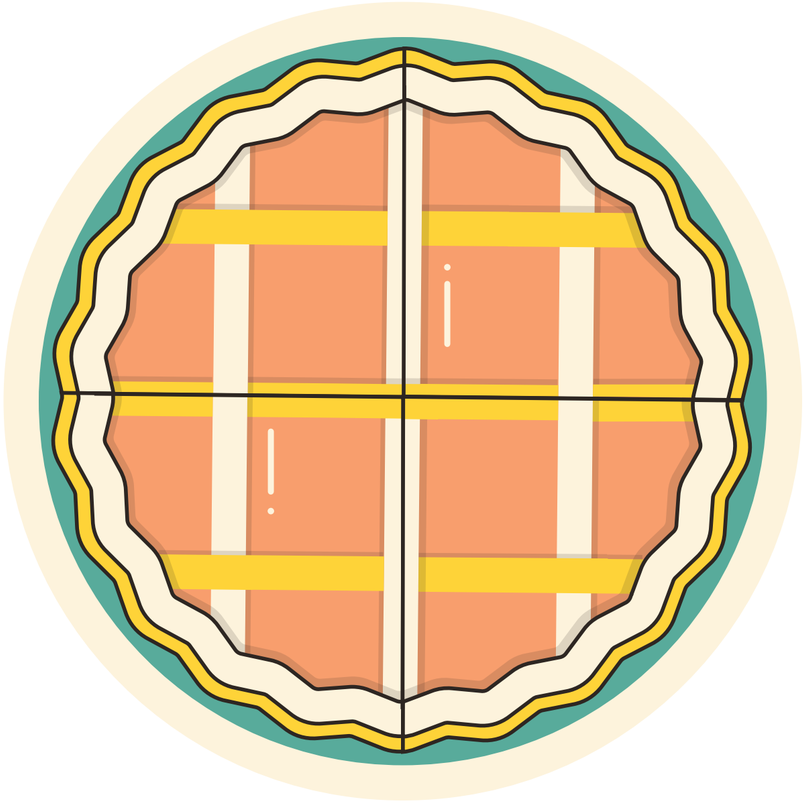
```sql
--Determine the percentage revenue category wise
--Calculate the percentage contribution of each pizza type to total revenue.

WITH total_revenues as (
            select sum(pizzas.price*order_details.quantity) as total_revenue
            from pizzas
           join order_details on pizzas.pizza_id = order_details.pizza_id
)

select pizza_types.category, (sum(pizzas.price*order_details.quantity)
    /(select total_revenue from total_revenues) *100) as category_wise_revenue
from pizzas
join pizza_types 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.category
order by category_wise_revenue desc
```

Data Output | Messages | Notifications

SQL

| | category<br>character varying (50) | category_wise_revenue<br>double precision |
|---|---|---|
| 1 | Classic | 26.905960230697634 |
| 2 | Supreme | 25.456311211146232 |
| 3 | Chicken | 23.95513753228847 |
| 4 | Veggie | 23.682591025867662 |

# ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.



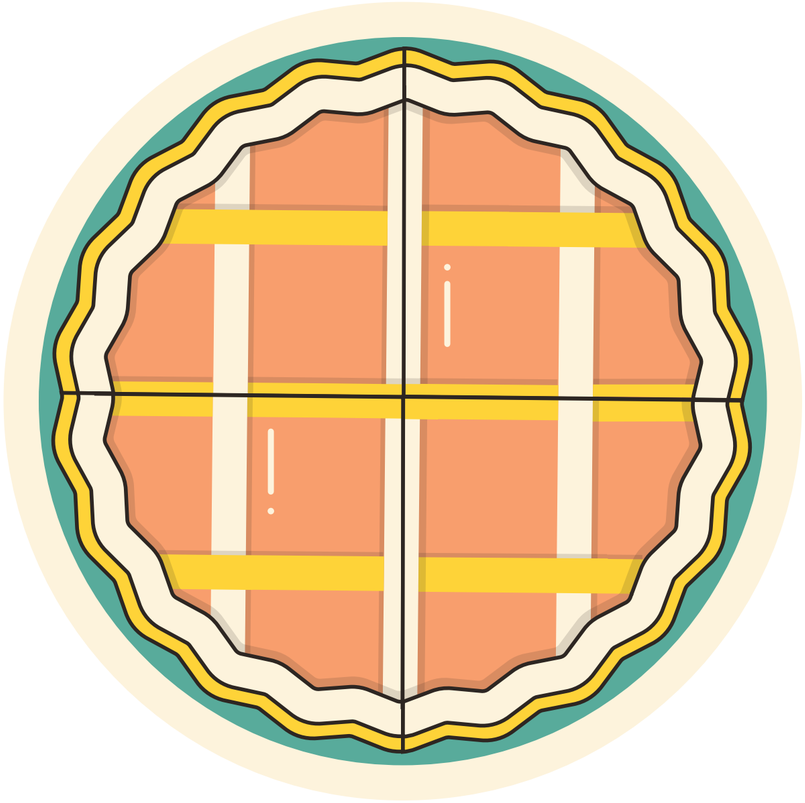
```sql
--Analyze the cumulative revenue generated over time.

With total_revenue as(
    select orders.orders_date as Sales_date, sum(order_details.quantity*pizzas.price) as revenue
    from pizzas
    join order_details on pizzas.pizza_id = order_details.pizza_id
    join orders on order_details.order_id = orders.order_id
    group by Sales_date
    order by Sales_date
)

select sales_date, sum(revenue) over(order by sales_date)
from total_revenue
```

Data Output | Messages | Notifications

SQL

| | sales_date date | sum double precision |
|---|---|---|
| 1 | 2015-01-01 | 2713.8500022888184 |
| 2 | 2015-01-02 | 5445.750003814697 |
| 3 | 2015-01-03 | 8108.150007247925 |
| 4 | 2015-01-04 | 9863.600008010864 |
| 5 | 2015-01-05 | 11929.550008773804 |
| 6 | 2015-01-06 | 14358.500011444092 |
| 7 | 2015-01-07 | 16560.70001220703 |
| 8 | 2015-01-08 | 19399.050018310547 |

# ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.



```sql
--Analyze the cumulative revenue generated over time.(2nd Menthod)

select orders_date,
sum(revenue) over(order by orders_date) as cum_revenue
from
(select orders.orders_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.orders_date) as sales;
```

Data Output | Messages | Notifications

SQL

| | sales_date date | sum double precision |
|---|---|---|
| 1 | 2015-01-01 | 2713.8500022888184 |
| 2 | 2015-01-02 | 5445.750003814697 |
| 3 | 2015-01-03 | 8108.150007247925 |
| 4 | 2015-01-04 | 9863.600008010864 |
| 5 | 2015-01-05 | 11929.550008773804 |
| 6 | 2015-01-06 | 14358.500011444092 |
| 7 | 2015-01-07 | 16560.70001220703 |
| 8 | 2015-01-08 | 19399.050018310547 |