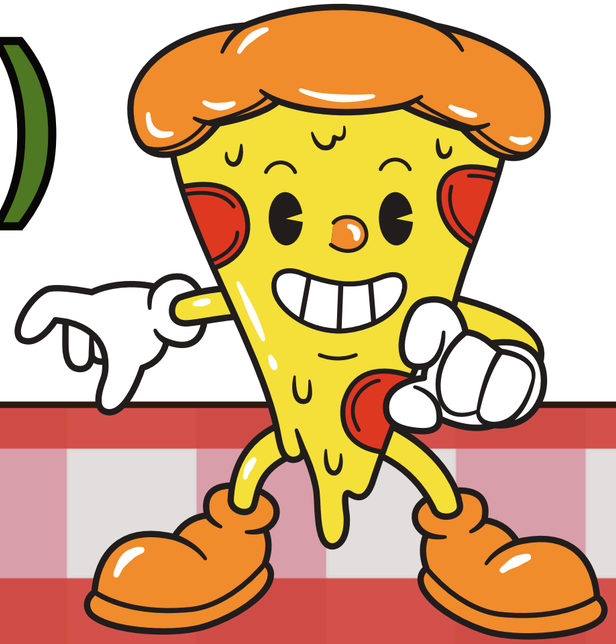
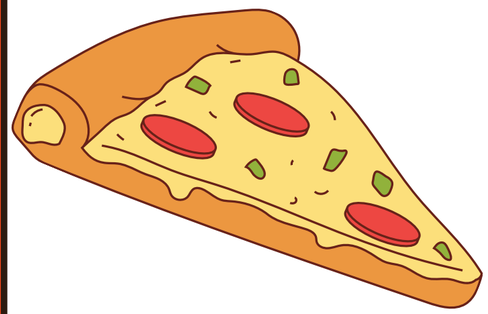
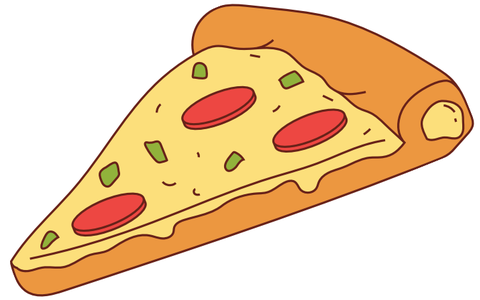
PIZZA SALES ANALYSIS (SQL)



Retrieve the total number of orders placed.

```sql
-- retrieve the total number of order placed
Select count(order_id) from orders;
```



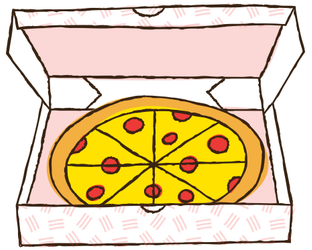


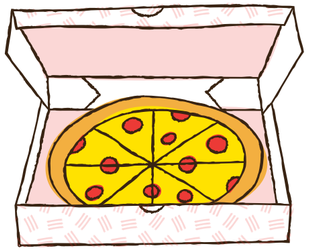
Calculate the total revenue generated from pizza sales.

```
2 •  select round(sum(od.quantity * p.price),2) as revenue from order_details od
3    join pizzas p
4    using (pizza_id)
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

revenue
817860.05



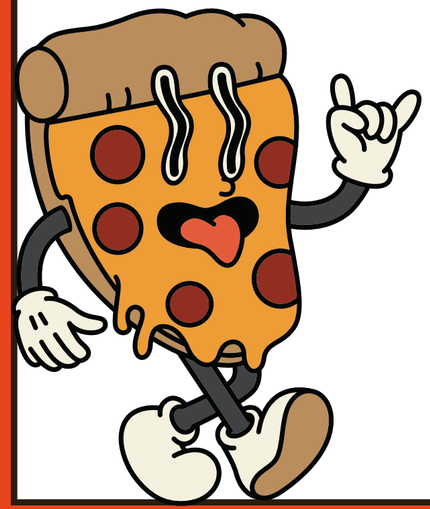


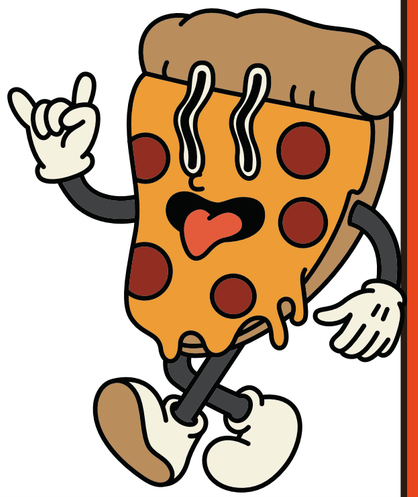
Identify the highest-priced pizza.

```sql
-- Identify the highest-priced pizza.
select pizza_type_id, price from pizzas
order by price desc
limit 1
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

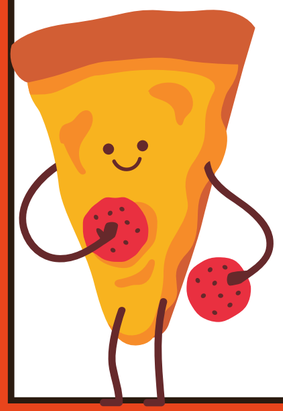
pizza_type_id	price
the_greek	35.95

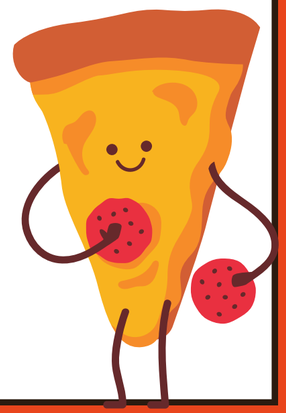




Identify the most common pizza size ordered.

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

select count(od.quantity) as count_pizza_size, p.size
from pizzas p
join order_details od
using (pizza_id)
group by p.size
order by count_pizza_size desc
```





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

```
2 •  select sum(od.quantity) as pizza_order, pt.name from order_details od
3    JOIN pizzas p
4    using (pizza_id)
5    JOIN pizza_types pt
6    using (pizza_type_id)
7    group by od.quantity, pt.name
8    order by pizza_order desc
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

pizza_order	name
2382	The Classic Deluxe Pizza
2321	The Pepperoni Pizza
2319	The Hawaiian Pizza
2316	The Barbecue Chicken Pizza
2260	The Thai Chicken Pizza

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

```
3 •  select pt.category, sum(od.quantity) as pizza_ordered from pizza_types pt
4    JOIN pizzas p
5    using(pizza_type_id)
6    JOIN order_details od
7    using (pizza_id)
8    group by pt.category
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

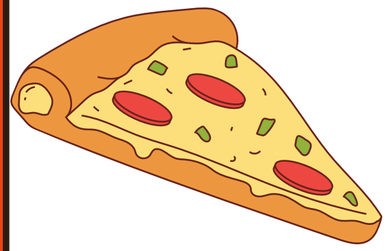
category	pizza_ordered
Classic	14888
Veggie	11649
Supreme	11987
Chicken	11050

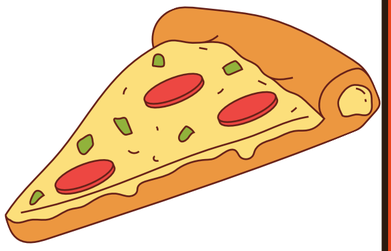
Determine the distribution of orders by hour of the day.

```
2 •  select hour(time) as hour, count(order_id) from orders
3    group by hour
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

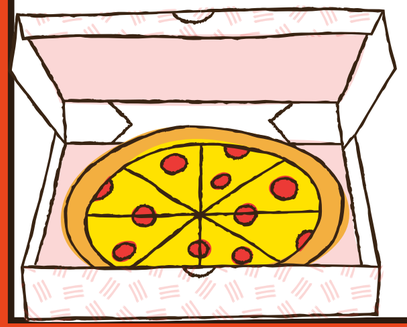
hour	count(order_id)
11	1231
12	2520
13	2455
14	1472

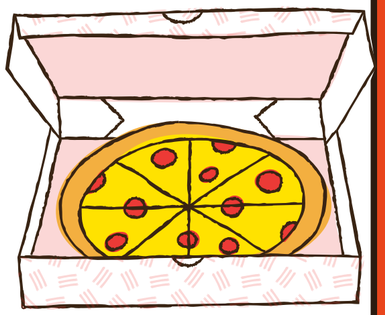




find the category-wise distribution of pizzas.

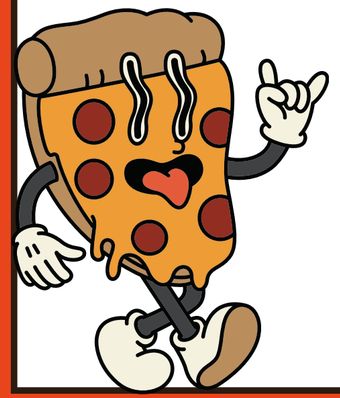
```
-- find the category-wise distribution of pizzas.

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

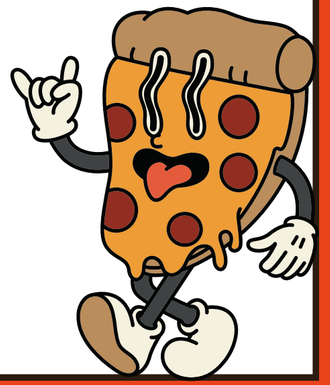




Group the orders by date and calculate the average number of pizzas ordered per day.

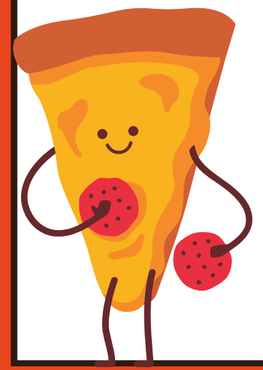
```sql
with table1 as (select o.date , sum(od.quantity) as quantity from orders o
JOIN order_details od
using (order_id)
group by o.date)
Select round(avg(quantity),0) from table1
```

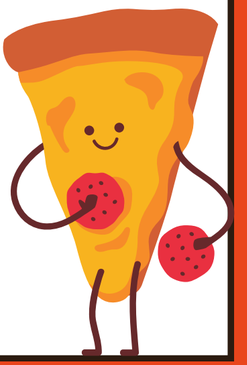




Determine the top 3 most ordered pizza types based on revenue.

```sql
select pt.name, sum(od.quantity*p.price) as revenue from pizza_types pt
JOIN pizzas p
using (pizza_type_id)
join order_details od
using (pizza_id)
group by pt.name
order by revenue desc
limit 3
```





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

```sql
select pt.category, round(sum(od.quantity*p.price)/(
SELECT round(sum(od.quantity*p.price),2) AS total_sales
FROM order_details od
JOIN pizzas p
USING (pizza_id))*100,2) as revenue_pct from pizza_types pt
JOIN pizzas p
using (pizza_type_id)
JOIN order_details od
using (pizza_id)
group by pt.category
```

Analyze the cumulative revenue generated over time.

```sql
with table1 as (select o.date, sum(od.quantity * p.price) as revenue
from order_details od
join pizzas p
using (pizza_id)
join orders o
using (order_id)
group by o.date)
select date, round(sum(revenue) over(order by date),2) as cum_rev
from table1
```

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

```sql
with table1 as(select pt.category, pt.name, round(sum(od.quantity*p.price),2) as revenue
from order_details od
join pizzas p
using (pizza_id)
join pizza_types pt
using (pizza_type_id)
group by pt.category, pt.name)
select *, dense_rank() over (partition by category order by revenue desc) as ranking
from table1
```