

# PIZZA

Pizza Sales analysis - SQL



DHAVAL KOLADIYA



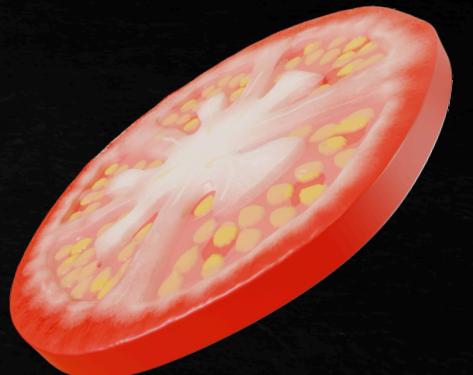
# Pizza Sales analysis - SQL

Retrieve the total number of orders placed

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



Result Grid	
	Total_order
▶	21350



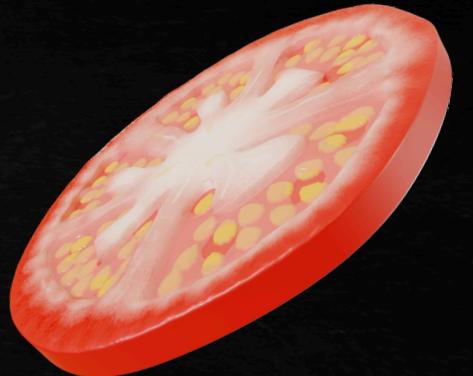
# Pizza Sales analysis - SQL

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;
```



Result Grid	
	total_revenue
▶	817860.05



# Pizza Sales analysis - SQL

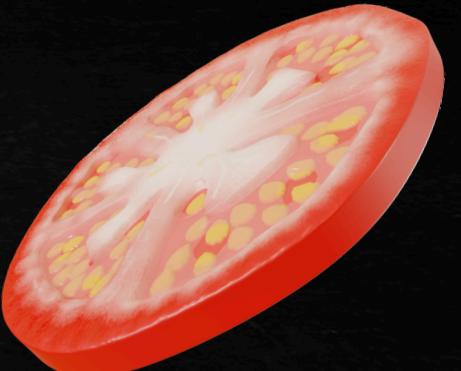
Identify the highest-priced pizza

```
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;
```



Result Grid | Filter Results

	name	price
▶	The Greek Pizza	35.95



# Pizza Sales analysis - SQL

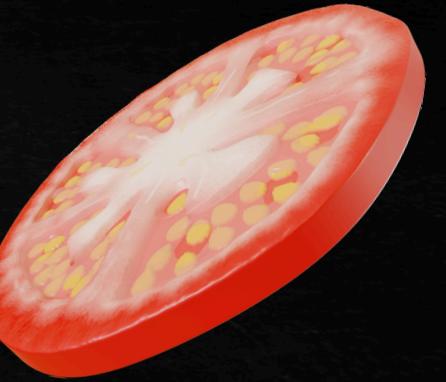
Identify the most common pizza size ordered

```
SELECT
    pizzas.size,
    COUNT(order_details.order_details_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
LIMIT 1;
```



A screenshot of a database query results interface. At the top, it says "Result Grid". Below is a table with two columns: "size" and "order\_count". There is one row showing the value "L" in the "size" column and "18526" in the "order\_count" column.

	size	order_count
▶	L	18526

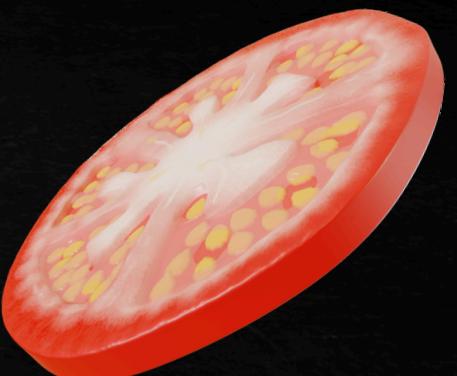


# Pizza Sales analysis - SQL

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

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

name	Total
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371



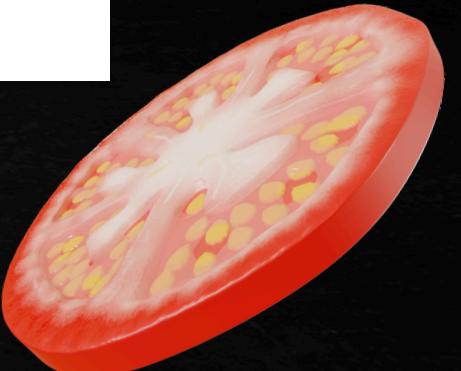
# Pizza Sales analysis - SQL

Determine the distribution of orders by hour of the day

```
SELECT  
    HOUR(time) AS Hour, COUNT(order_id) AS Count  
FROM  
    orders  
GROUP BY HOUR(time);
```



Hour	Count
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642
21	1198
22	663
23	28
10	8
9	1



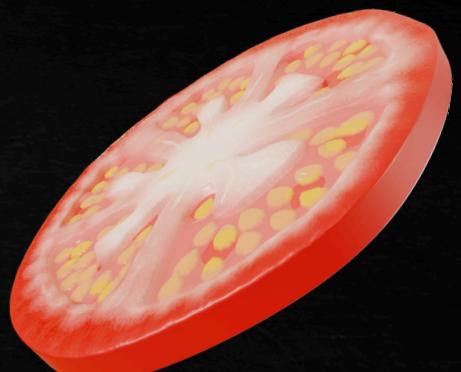
# Pizza Sales analysis - SQL

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

```
SELECT  
    category, COUNT(name) AS Count  
FROM  
    pizza_types  
GROUP BY category;
```



category	Count
Chicken	6
Classic	8
Supreme	9
Veggie	9



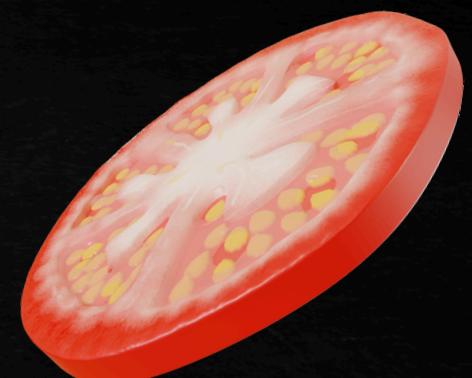
# Pizza Sales analysis - SQL

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

```
SELECT  
    ROUND(AVG(quantity), 0)  
FROM  
    (SELECT  
        orders.date, SUM(order_details.quantity) AS quantity  
    FROM  
        orders  
    JOIN order_details ON orders.order_id = order_details.order_id  
    GROUP BY orders.date) AS Avg_order;
```



```
ROUND(AVG(quantity), 0)  
138
```

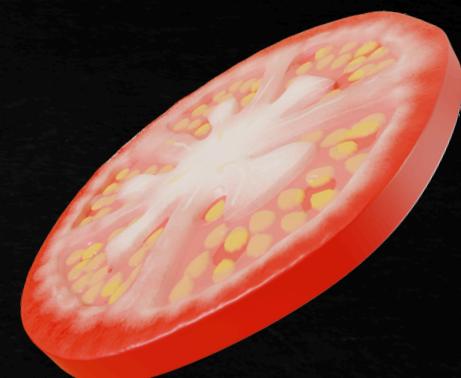


# Pizza Sales analysis - SQL

Determine the top 3 most ordered pizza types based on revenue

```
SELECT
    pizza_types.name,
    ROUND(SUM(order_details.quantity * pizzas.price),
        2) AS total_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 total_revenue DESC
LIMIT 3;
```

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



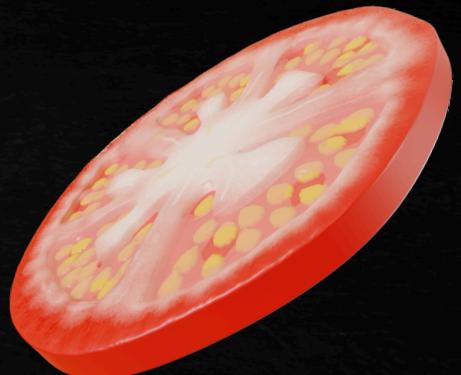
# Pizza Sales analysis - SQL

## Analyze the cumulative revenue generated over time

```
select date ,Revenue_of_the_day,sum(Revenue_of_the_day) over (order by date) as cum_revenue  
from  
(select orders.date ,  
round(sum(order_details.quantity*pizzas.price),2) As Revenue_of_the_day  
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.date)as sales;
```



date	Revenue_of_the_day	cum_revenue
2015-01-01	2713.85	2713.85
2015-01-02	2731.9	5445.75
2015-01-03	2662.4	8108.15
2015-01-04	1755.45	9863.6
2015-01-05	2065.95	11929.55
2015-01-06	2428.95	14358.5



# Pizza Sales analysis - SQL

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

```
select name, total from  
  (select category, name ,Total,  
    rank() over(partition by category order by total desc) as rn  
  from  
    (select pizza_types.category,pizza_types.name,round(sum(order_details.quantity*pizzas.price),2) As Total  
     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	total
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
The Pepperoni Pizza	30161.75

