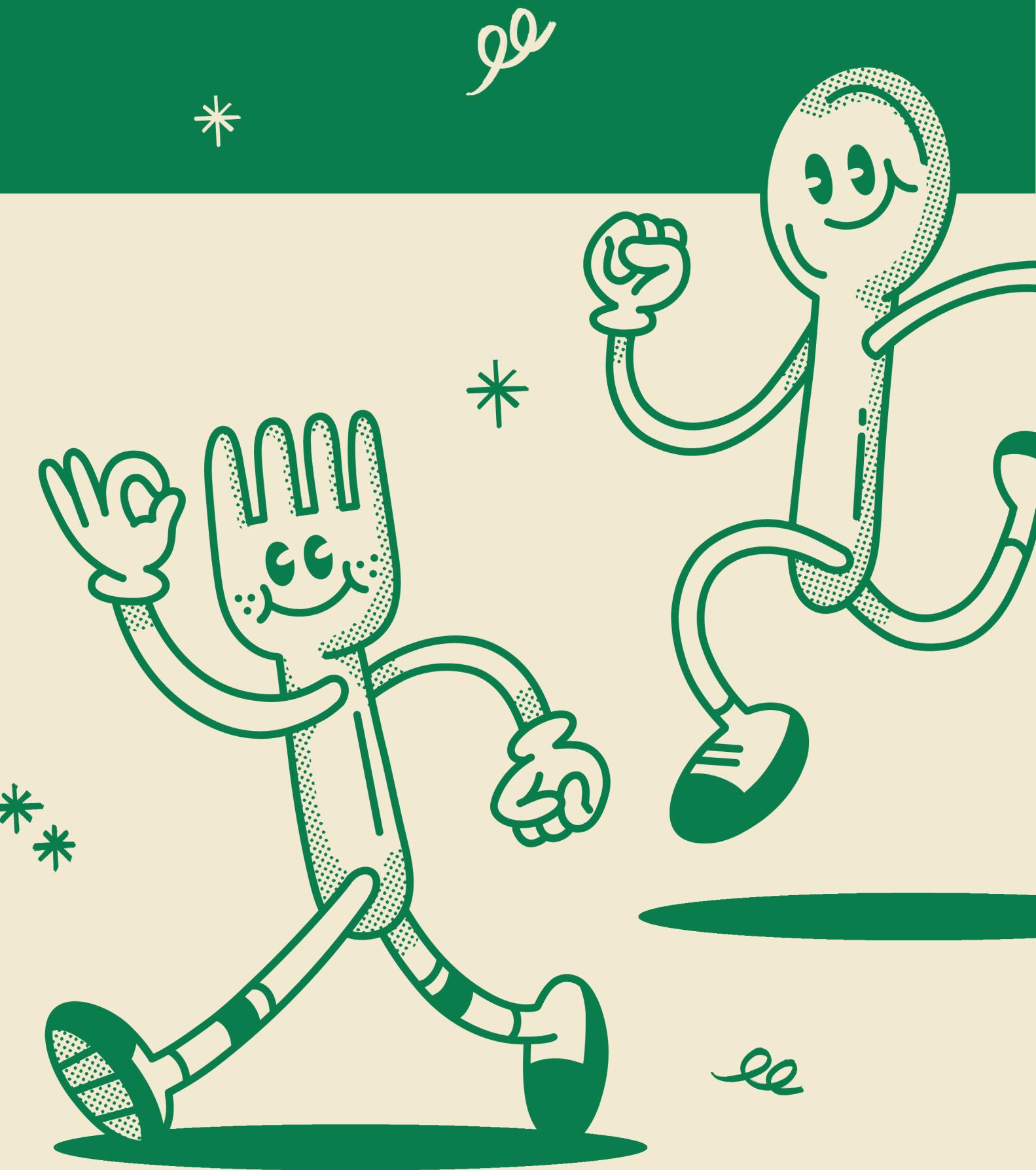
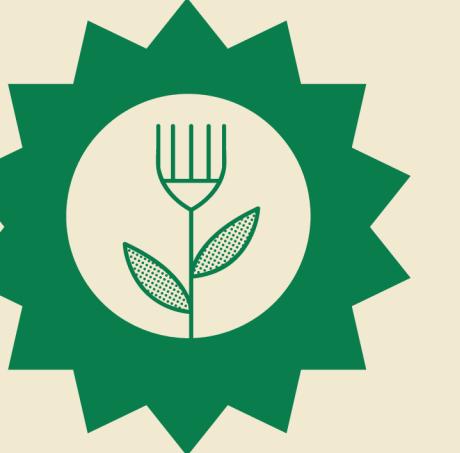


SQL project on Pizza sales





"Hi everyone! I'm Anas Fazil, and I'm excited to take you through my project on Pizza Sales Analysis using SQL."



Pizza Sales Analytics

In this project, I stepped into the shoes of a data analyst for a pizza company — digging deep into their sales data to uncover what's hot, what's not, and where the dough is really rolling in.

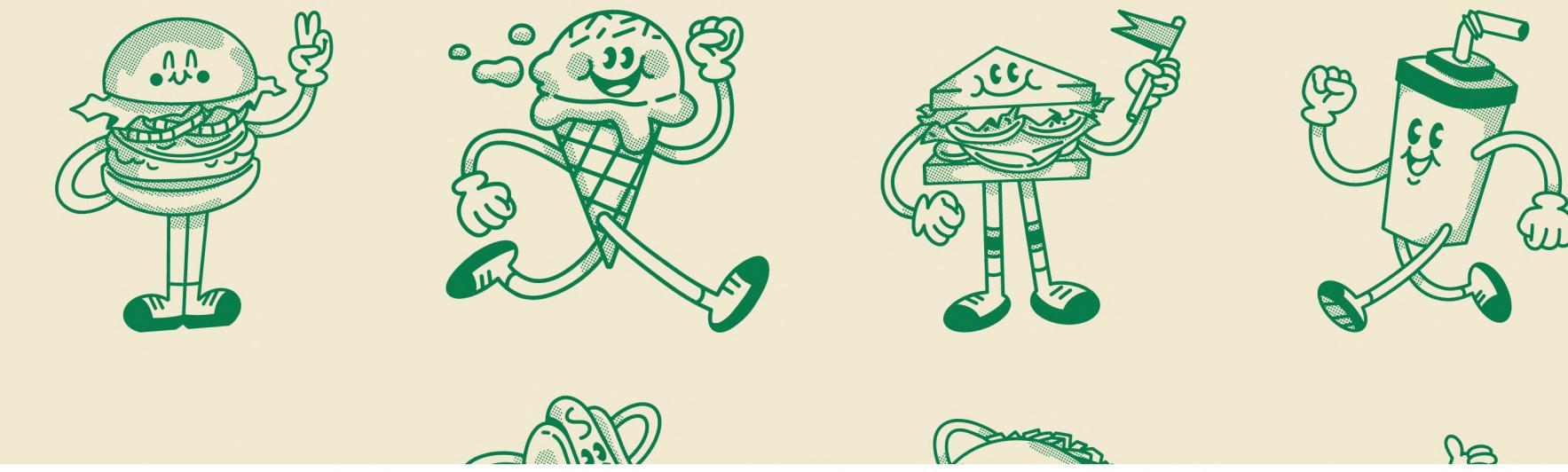
Problem

Retrieve the total number of orders placed.

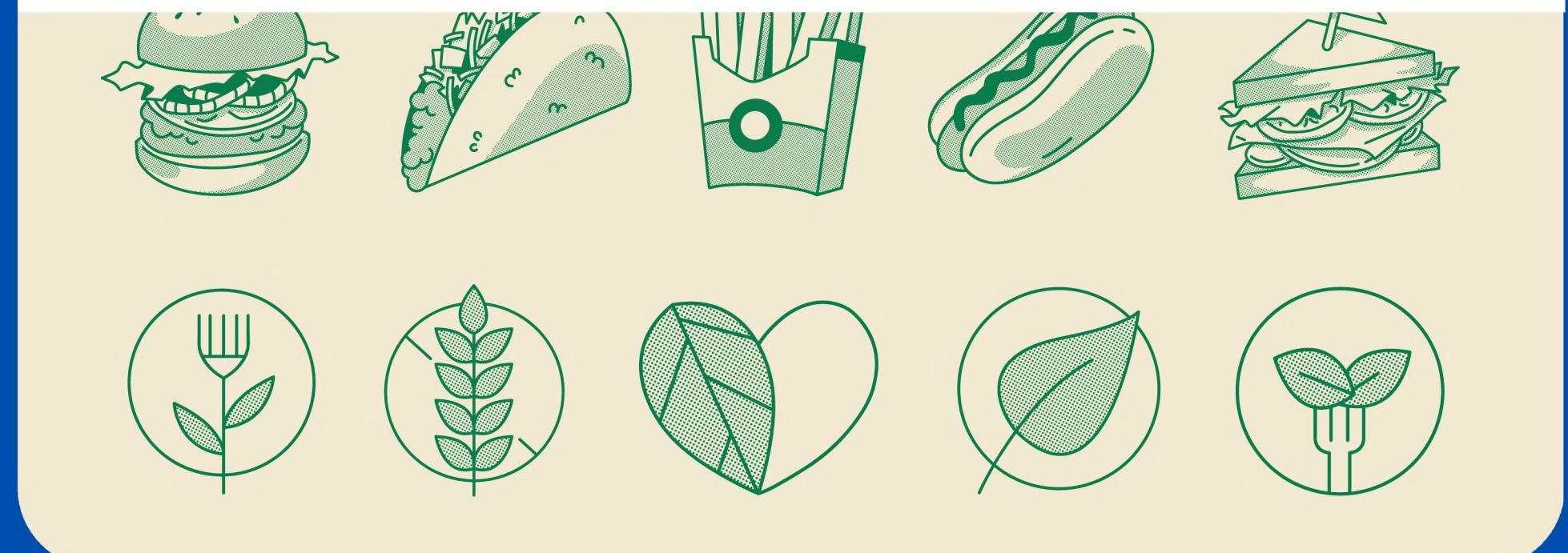
Output

Result Grid	
	Total_orders
▶	21350

SQL Query



```
1 -- Retrieve the total number of orders placed  
2 • Select Count(*) as Total_orders  
3 from orders;
```



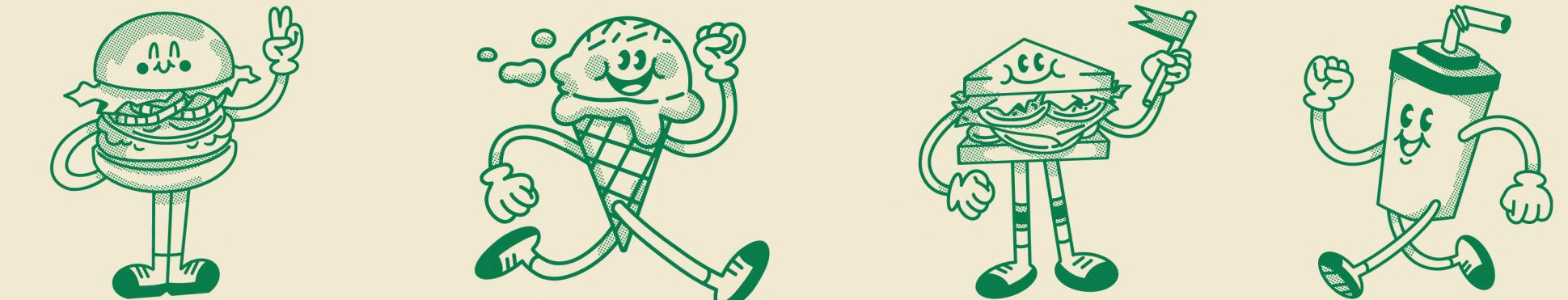
Problem

Calculate the total revenue generated from pizza sales.

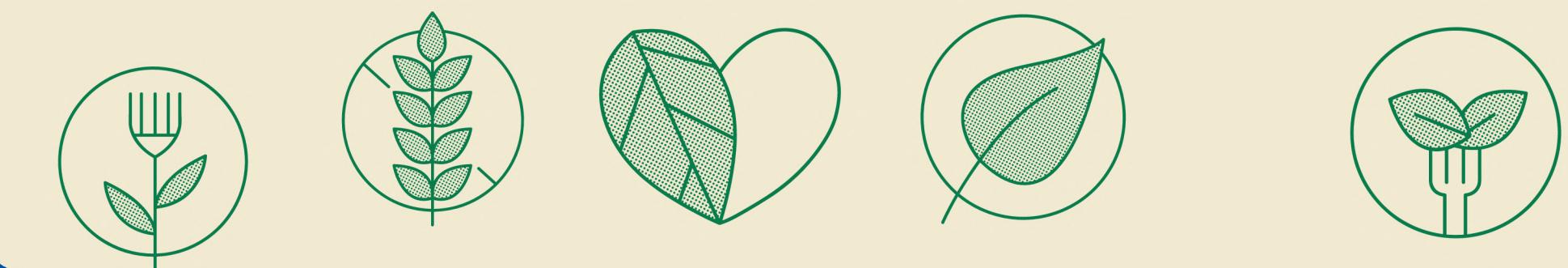
Output

Result Grid	
	total_revenue
▶	817860.05

SQL Query



```
1  -- Calculate the total revenue generated from pizza sales.  
2  
3 • SELECT  
4      ROUND(SUM(Quantity * price), 2) AS total_revenue  
5  FROM  
6      order_details AS od  
7      LEFT JOIN  
8      pizzas AS p ON od.pizza_id = p.pizza_id;  
9
```



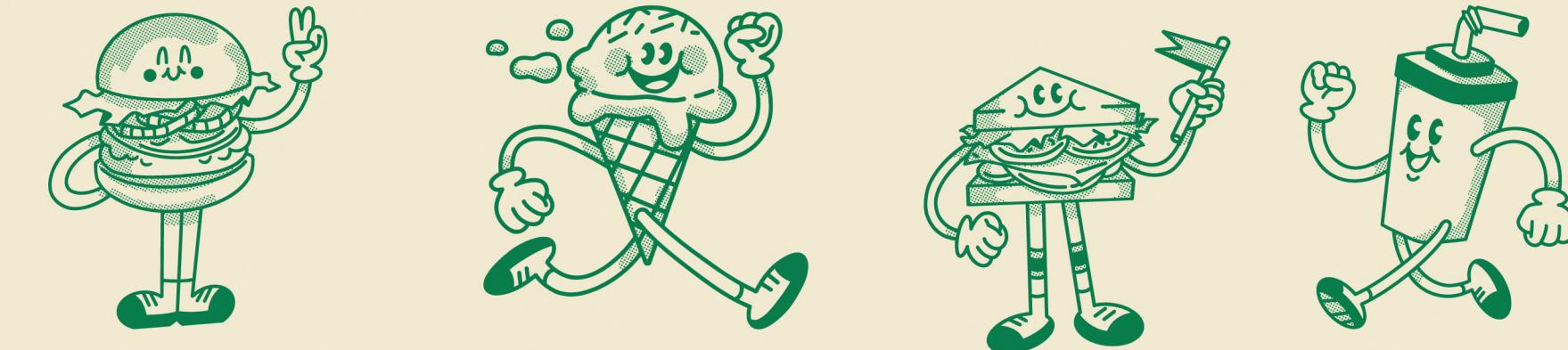
Problem

Calculate the total revenue generated from pizza sales.

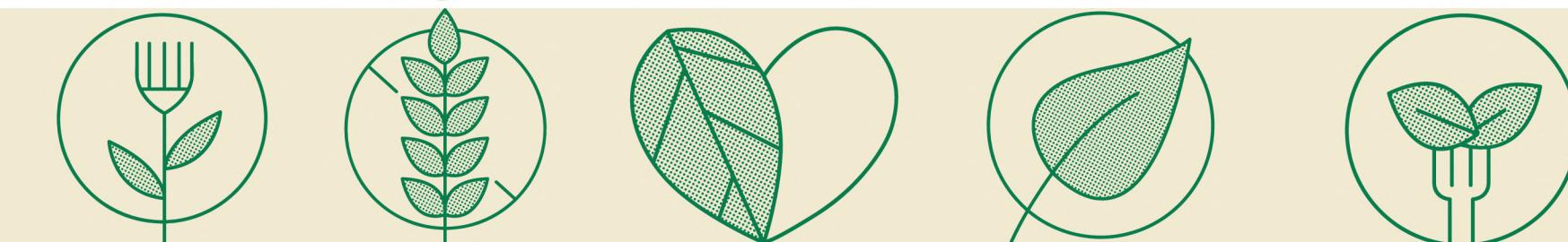
Output

Result Grid	
	total_revenue
▶	817860.05

SQL Query



```
1 -- Identify the highest-priced pizza.  
2  
3 • Select name as Pizza_name , price  
4 from pizza_types as pt  
5 Join pizzas as p  
6     ON pt.pizza_type_id = p.pizza_type_id  
7 Order by price Desc  
8 Limit 1;
```



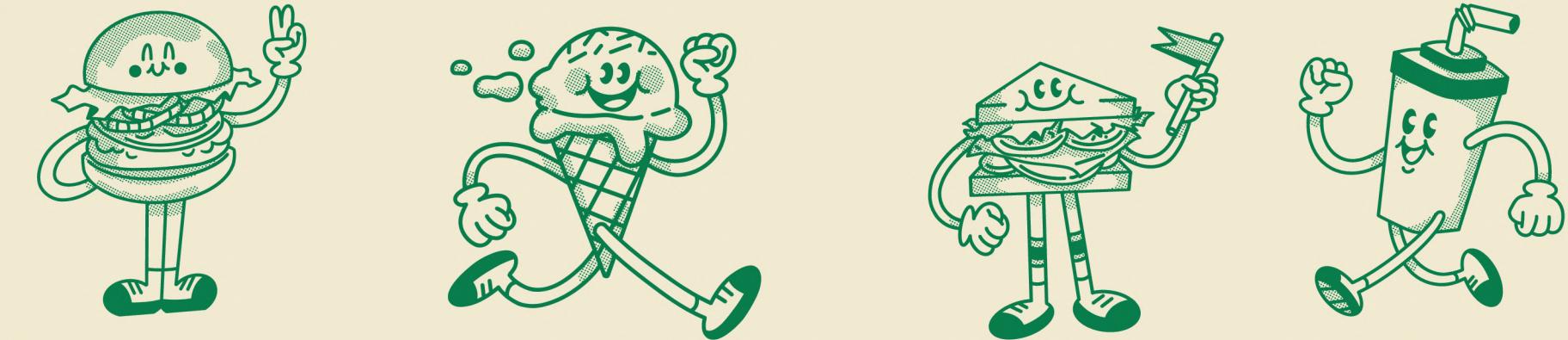
Problem

Identify the highest-priced pizza.

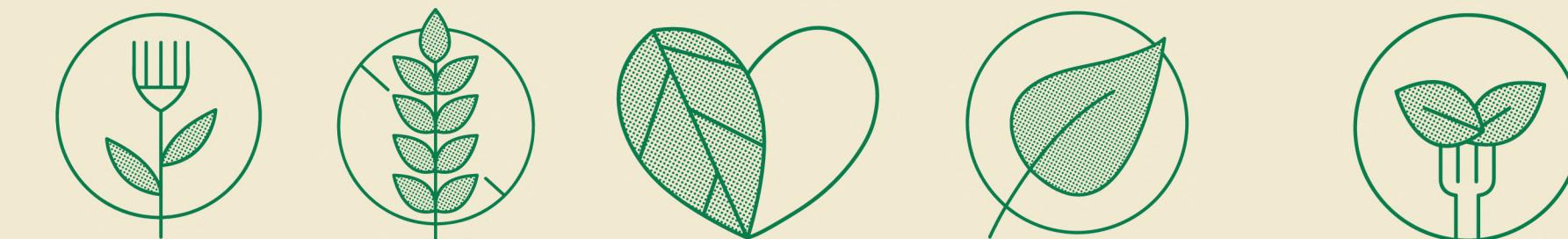
Output

Result Grid		
	qty	size
▶	18526	L
	15385	M
	14137	S
	544	XL
	28	XXL

SQL Query



```
1 -- Identify the most common pizza size ordered.  
2  
3 • Select count(quantity) as qty, size  
4   from order_details as od  
5   Join pizzas as p  
6     ON od.pizza_id=p.pizza_id  
7   group by size  
8   order by qty Desc;
```



Problem

Identify the most common pizza size ordered.

Output

Result Grid		Filter Rows:
	Pizza_names	qty
▶	The Classic Deluxe Pizza	2453
	The Barbecue Chicken Pizza	2432
	The Hawaiian Pizza	2422
	The Pepperoni Pizza	2418
	The Thai Chicken Pizza	2371

SQL Query



```
1 -- List the top 5 most ordered pizza types
2 -- along with their quantities.
3
4 • Select name as Pizza_names, sum(quantity) as qty
5   from order_details as od
6   Join pizzas as p
7     ON od.pizza_id= p.pizza_id
8   Join pizza_types as pt
9     ON p.pizza_type_id=pt.pizza_type_id
10  group by Pizza_names
11  order by qty Desc
12  Limit 5
```



Problem

Identify the most common pizza size ordered.

Output

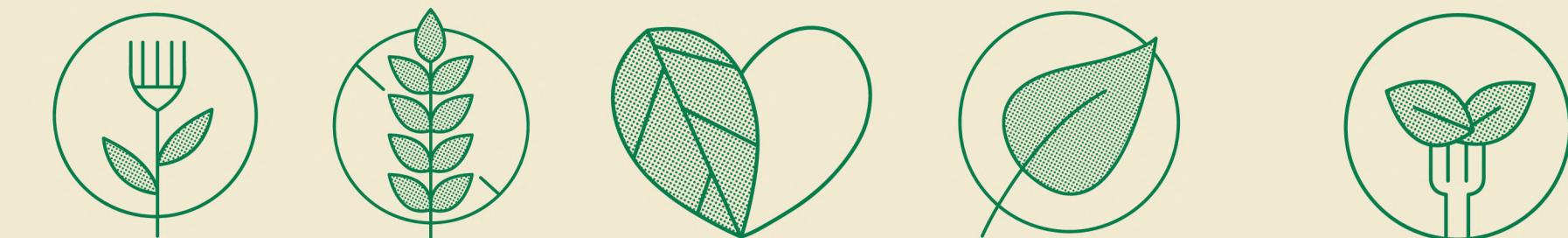
Result Grid

Category	total
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

SQL Query



```
1 -- Join the necessary tables to find
2 -- the total quantity of each pizza category ordered.--
3
4 • Select distinct Category, sum(quantity)as total
5 from pizza_types as pty
6 join pizzas as p
7     ON pty.pizza_type_id=p.pizza_type_id
8 Join order_details as od
9     ON p.pizza_id=od.pizza_id
10 group by Category
11 order by total Desc ;
```



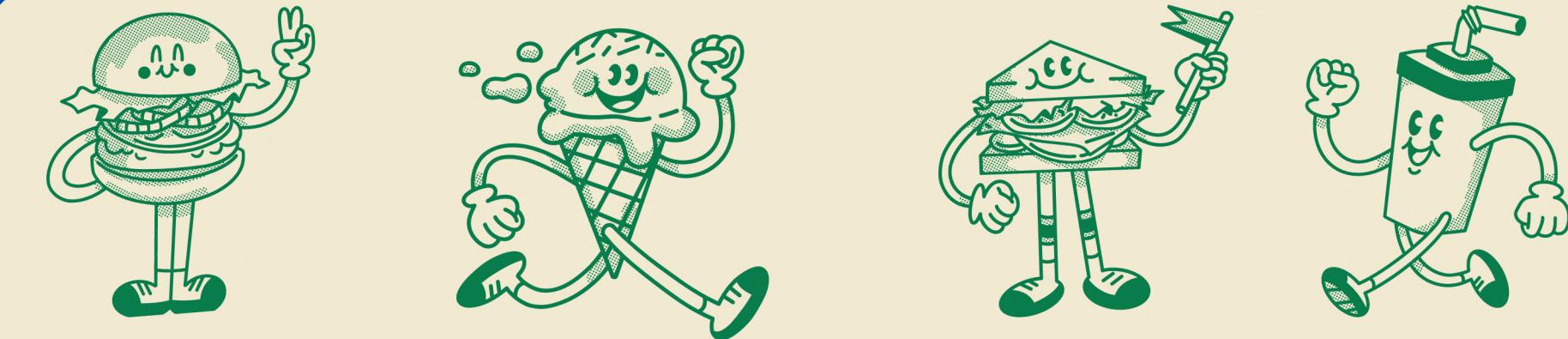
Problem

Identify the most common pizza size ordered.

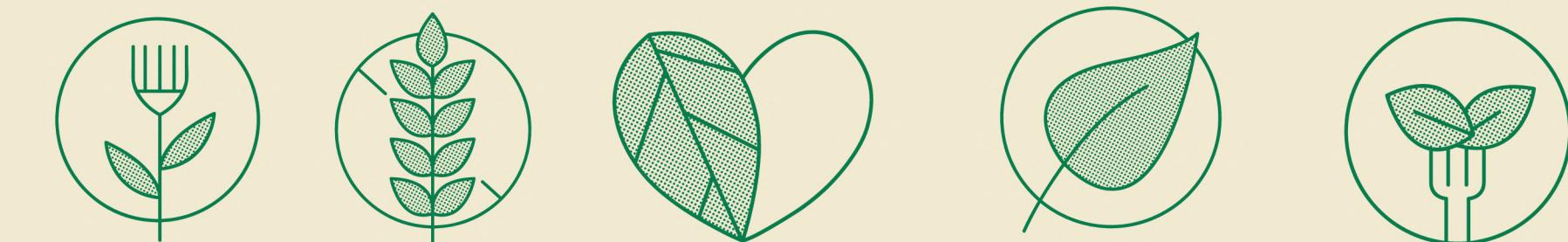
Output

Result Grid	
odrs	total_orders
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009

SQL Query



```
1 -- Determine the distribution of orders
2 -- by hour of the day.
3
4 • Select Hour(order_time) as odrs,
5 Count(order_id) as total_orders
6 from orders
7 group by odrs;
```



Problem

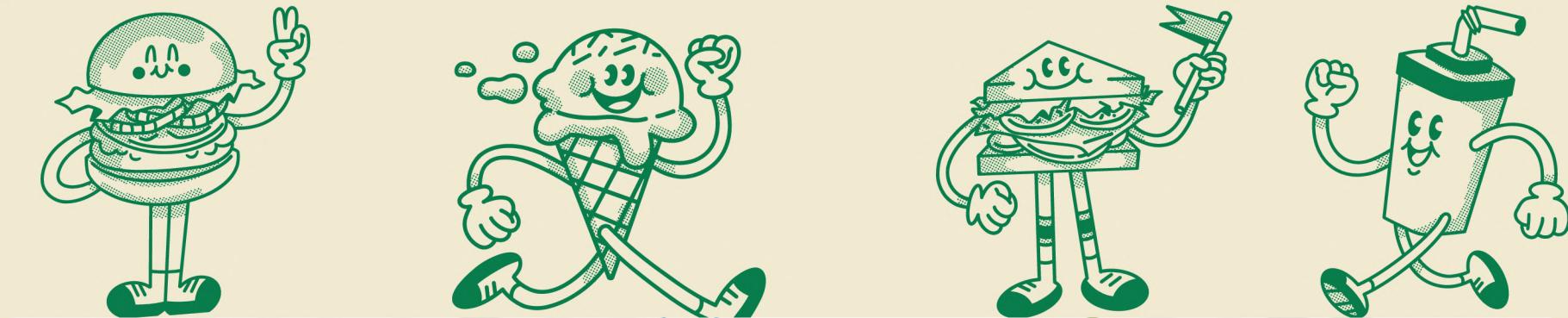
Identify the most common pizza size ordered.

Output

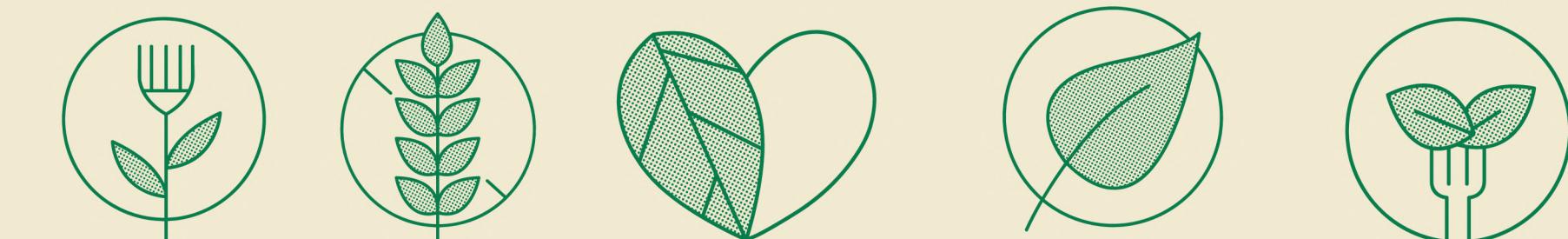
Result Grid | Filter Rows:

Category	Pizza_names
Chicken	6
Classic	8
Supreme	9
Veggie	9

SQL Query



```
1 -- Join relevant tables to find  
2 -- the category-wise distribution of pizzas.  
3  
4 • SELECT  
5     Category, COUNT(name) AS Pizza_names  
6 FROM  
7     pizza_types  
8 GROUP BY Category;
```



Problem

Identify the most common pizza size ordered.

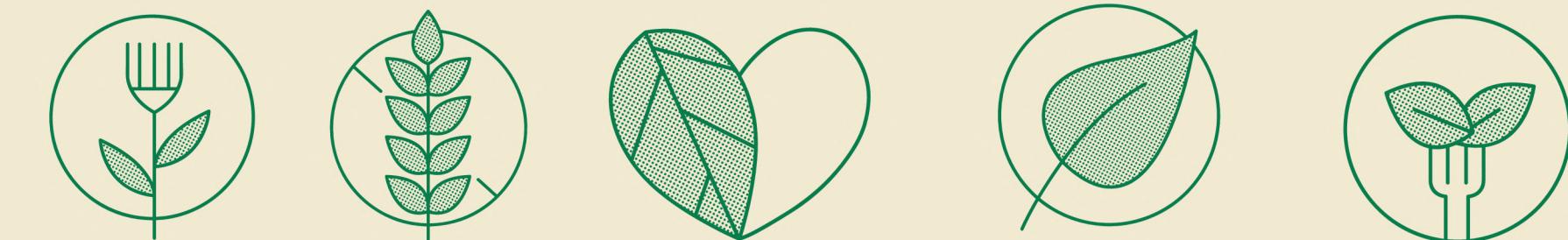
Output

Result Grid	
	avg_no_pizzas
▶	138

SQL Query



```
1 -- Group the orders by date
2 --
3 -- calculate the average number of pizzas ordered per day.
4
5 • Select Round(avg(quantity), 0) as avg_no_pizzas From
6   (Select Date(order_date) as orders , sum(quantity) as quantity
7   From orders as o
8   Join order_details as od
9   ON o.order_id=od.order_id
10  group by orders)as order_quantity;
```



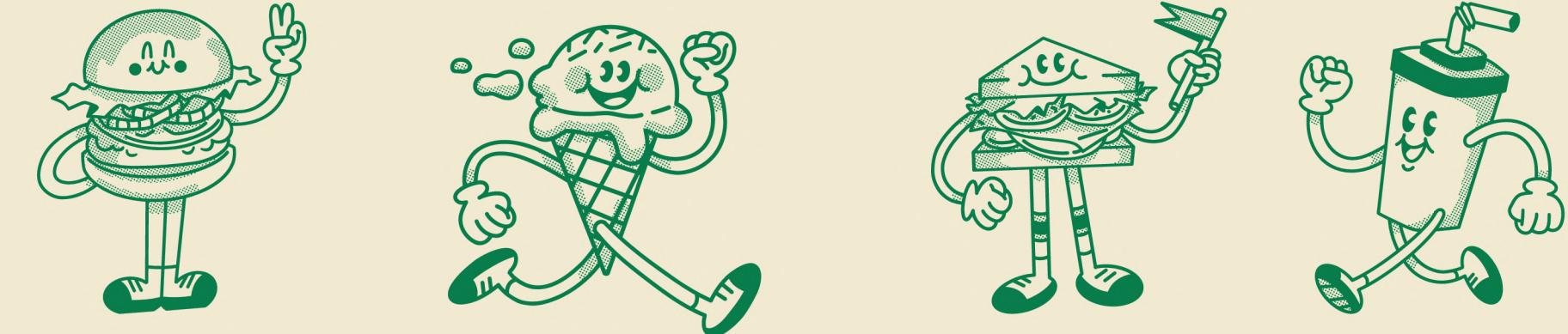
Problem

Identify the most common pizza size ordered.

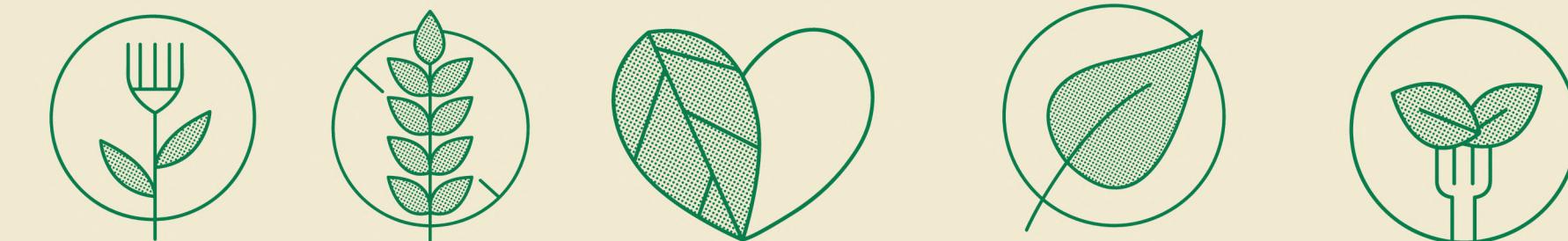
Output

Result Grid		Filter Rows:
Pizza_name	Revenue	
The Thai Chicken Pizza	43434.25	
The Barbecue Chicken Pizza	42768	
The California Chicken Pizza	41409.5	

SQL Query



```
1 -- Determine the top 3 most ordered pizza types based on revenue.  
2  
3 • Select name as Pizza_name, Sum(quantity*price) as Revenue  
4 From pizza_types as pty  
5 Join pizzas as p  
6 On pty.pizza_type_id=p.pizza_type_id  
7 Join order_details as od  
8 ON p.pizza_id=od.pizza_id  
9 Group by Pizza_name  
10 Order by Revenue Desc  
11 Limit 3;
```



Problem

Identify the most common pizza size ordered.

Output

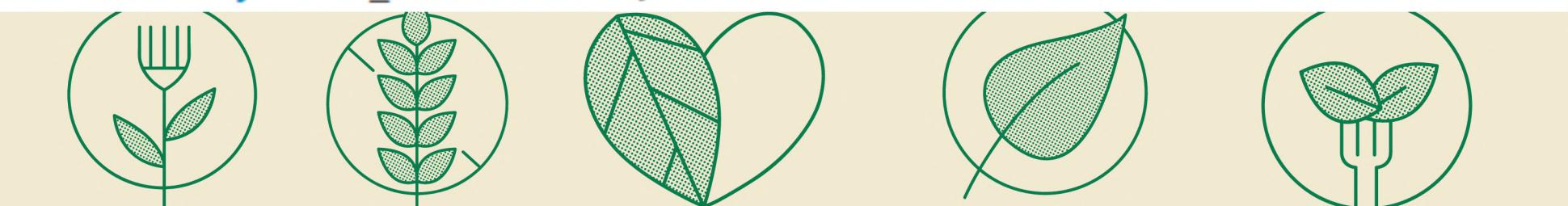
Result Grid | Filter Rows:

Category	total_revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

SQL Query



```
1 -- Calculate the percentage contribution of each pizza type to total revenue.
2 • Select Category,
3   Round((Sum(price*quantity) / (SELECT
4     ROUND(SUM(Quantity * price), 2) AS total_revenue
5   FROM
6     order_details AS od
7     LEFT JOIN
8       pizzas AS p ON od.pizza_id = p.pizza_id) )*100, 2) as total_revenue
9   From pizza_types as pty
10  Join pizzas as p
11    On pty.pizza_type_id=p.pizza_type_id
12  Join order_details as od
13    On p.pizza_id=od.pizza_id
14  group by Category
15  order by total_revenue Desc ;
```



Problem

Identify the most common pizza size ordered.

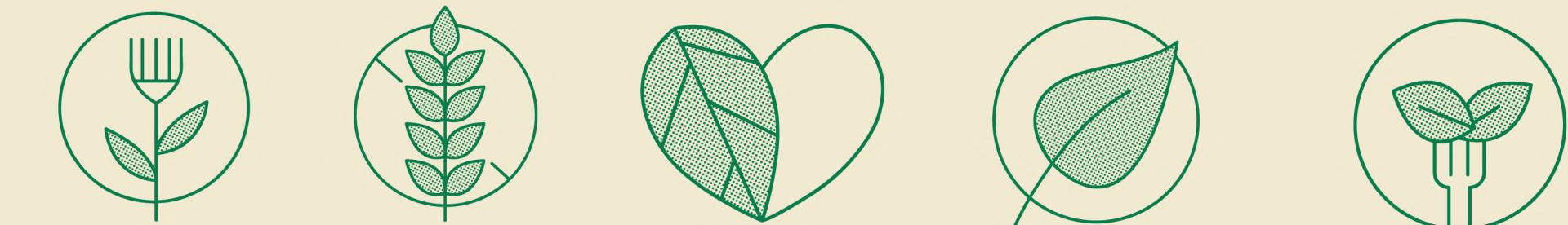
Output

	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
	2015-01-06	14358.5
	2015-01-07	16560.7
	2015-01-08	18000.0

SQL Query



```
1  -- Analyze the cumulative revenue generated over time.  
2 • Select order_date, sum(revenue) over (Order by order_date) as Cum_Revenue  
3  From  
4  ⊕ (Select order_date, Sum(price*quantity) as revenue  
5    from Orders as o  
6    Join order_details as od  
7      ON o.order_id = od.order_id  
8    Join pizzas as p  
9      ON p.pizza_id=od.pizza_id  
10   group by order_date) as Sales;
```



Problem

Identify the most common pizza size ordered.

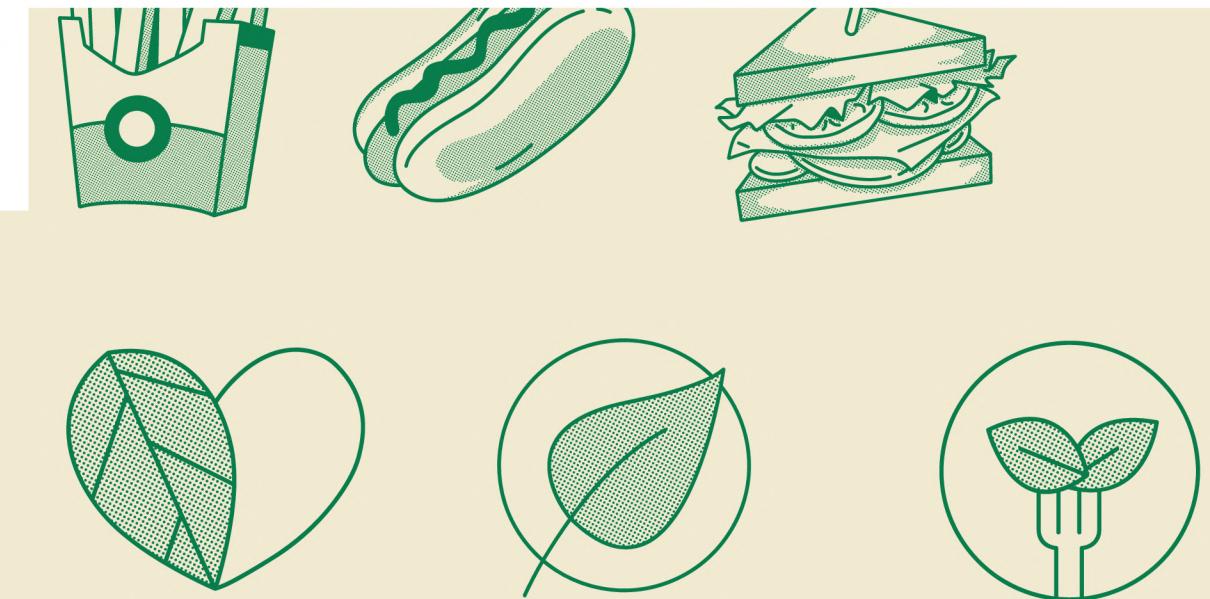
Output

Pizza_names	revenue
The Thai Chicken Pizza	43434
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41410
The Classic Deluxe Pizza	38180
The Hawaiian Pizza	32273
The Pepperoni Pizza	30162
The Spicy Italian Pizza	34831
The Italian Supreme Pizza	33477
The Sicilian Pizza	30940
The Four Cheese Pizza	32266
The Mexicana Pizza	26781
The Five Cheese Pizza	26066

SQL Query



```
1  #Determine the top 3 most ordered pizza types
2  #based on revenue for each pizza category.
3
4 • Select Pizza_names, revenue From
5   (Select category, Pizza_names, revenue,
6    Rank() over(partition by Category order by revenue desc) as Rn
7    From
8    (Select Distinct Category, name as Pizza_names , Round(Sum(price*quantity),0) as revenue
9     from pizza_types as pty
10    join pizzas as p
11      ON pty.pizza_type_id=p.pizza_type_id
12    Join order_details as od
13      ON p.pizza_id= od.pizza_id
14    Group by Category, Pizza_names
15   ) as P) as B
16  where Rn<=3;
```



Conclusion & Takeaways

PROJECT WRAP-UP

In this project, I analyzed a pizza sales dataset using SQL to uncover real business insights — like best-selling pizzas, peak ordering hours, and category-wise revenue.

I used SQL techniques such as:

- JOIN, GROUP BY, ORDER BY
- RANK() and DENSE_RANK() for performance comparison
- WINDOW FUNCTIONS for rolling totals and rankings

Learning Outcome

This project sharpened my skills in:

- Writing clean, optimized SQL queries
- Thinking like a data analyst
- Translating data into business impact

Thank you for your time — and I hope this presentation was as satisfying as a loaded pizza slice!

SIMILAR FONTS

- ITC Souvenir
- Recolecta

SUGGESTED FONT PAIRING

- ITC Franklin Gothic
- Gladiola

BONUS INSIGHTS

- CLASSIC & VEG PIZZAS TOPPED THE SALES CHART
- MEDIUM SIZE WAS THE MOST PREFERRED BY CUSTOMERS
- EVENING HOURS SAW THE HIGHEST ORDER TRAFFIC
- ROLLING REVENUE GAVE GREAT CLARITY ON DAILY GROWTH

LEARNING OUTCOME

THIS PROJECT SHARPENED MY SKILLS IN:

- WRITING CLEAN, OPTIMIZED SQL QUERIES
- THINKING LIKE A DATA ANALYST
- TRANSLATING DATA INTO BUSINESS IMPACT