

Pizza Insights

Pizza Sales Data Overview



Delicious Pizza Sales Analysis Report

By

Saurabh Jamnare

Introduction:

The pizza sales insights are derived from a comprehensive analysis of sales data using MySQL. Various SQL queries have been utilized to extract meaningful information and provide valuable insights into pizza sales trends, customer preferences, revenue generation, and more. Below is an overview of the queries, clauses, and functions used in this analysis.

Tools Used: MySQL-SQL Queries

Clauses/Function Used: SELECT, FROM, JOINS, GROUP BY, ORDER BY, LIMIT, ROUND, SUM, COUNT, AVG, WINDOW FUNCTIONS, WHERE, DATE FUNCTIONS, SUBQUERY, CTE.

PIZZA
PARTY

Agenda

Overview of Pizza Sales Report

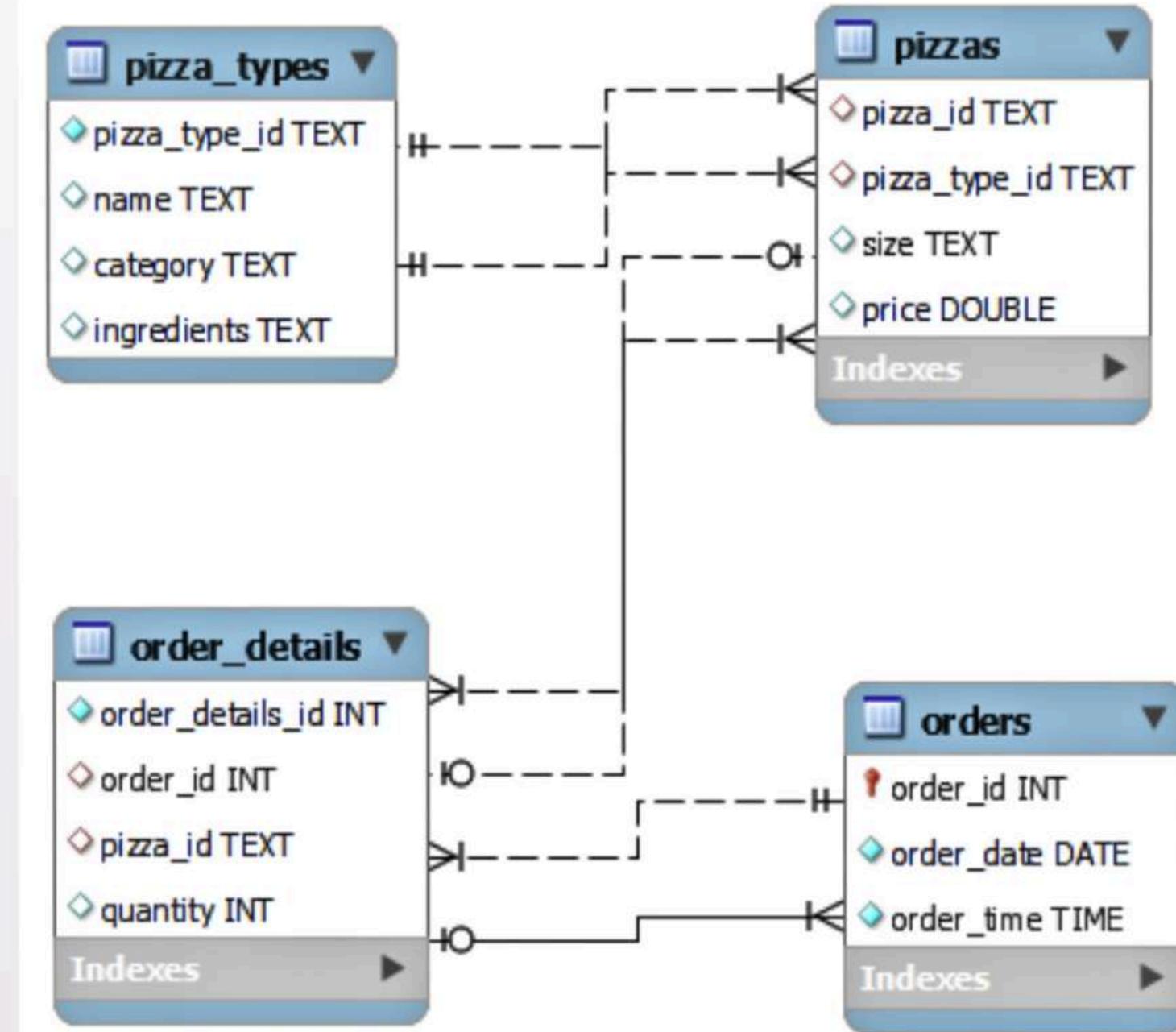
- 01 Introduction to Sales Analysis
- 02 Total ordered Placed
- 03 Cutotal revenue generated from pizza sales
- 04 Highest-priced pizza.
- 05 Most common pizza size ordered.
- 06 Total quantity of each pizza category ordered
- 07 Distribution of orders by hour of the day.
- 08 Category-wise distribution of pizzas
- 09 Group the orders by date and calculating the Category-wise distribution of pizzas
- 10 Top 3 most ordered pizza types based on revenue.
- 11 Percentage contribution of each pizza type to total revenue.
- 12 Cumulative revenue generated over time.
- 13 Top 3 most ordered pizza types based on revenue for each pizza category.
- 14 Thank You

Pizza Insights

Analyzing pizza sales data to drive business decisions effectively and efficiently

KEY METRICS FOR PIZZA SALES ANALYSIS

Explore the latest trends and performance in the pizza sales industry to optimize strategies and boost profitability



Total ordered Placed

```
1 -- Q.1 Retrieve the total number of orders placed.  
2 • SELECT COUNT(order_id) AS Total_orders_placed  
3 FROM orders
```

A screenshot of a MySQL query results window. The window has a toolbar at the top with buttons for 'Result Grid' (selected), 'Filter Rows', 'Export', and 'Wrap Cell Content'. The main area shows a single row of data:

Total_orders_placed
21350



Order_placed

Total order Placed till now

sales data reveals the total orders that has been placed over a period is 21350

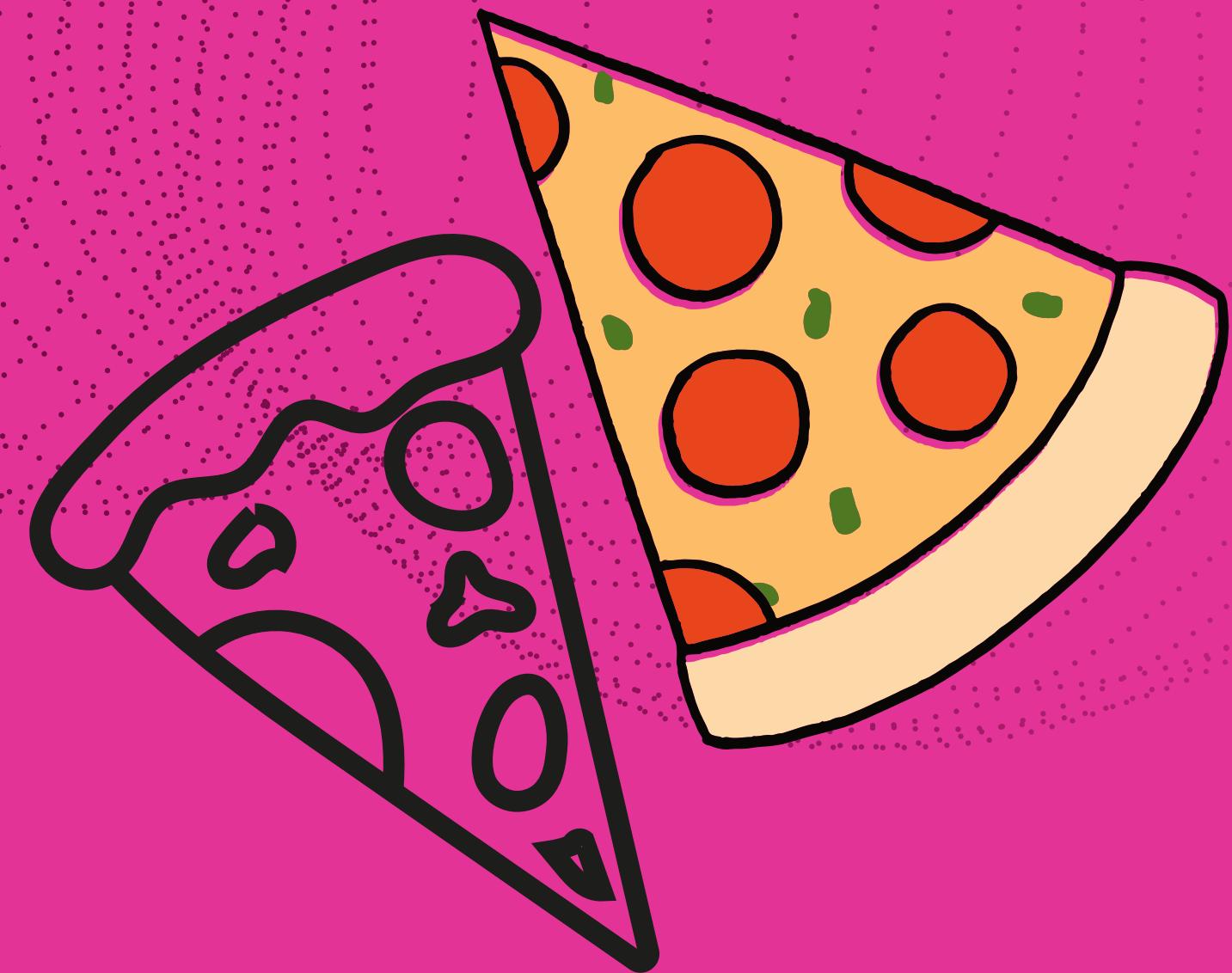
Total Revenue

Revenue is the most important factor to make sure your business remains profitable and sustainable. It directly reflects the financial health of your company and its ability to cover costs, invest in growth opportunities, and generate profits.

```
1 -- Q.2 Calculate the total revenue generated from pizza sales.  
2 • SELECT ROUND(SUM(o.quantity*p.price), 2) AS Revenue  
3   FROM pizzas p  
4   JOIN order_details o ON p.pizza_id = o.pizza_id  
5
```

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

Revenue
817860.05



Revenue

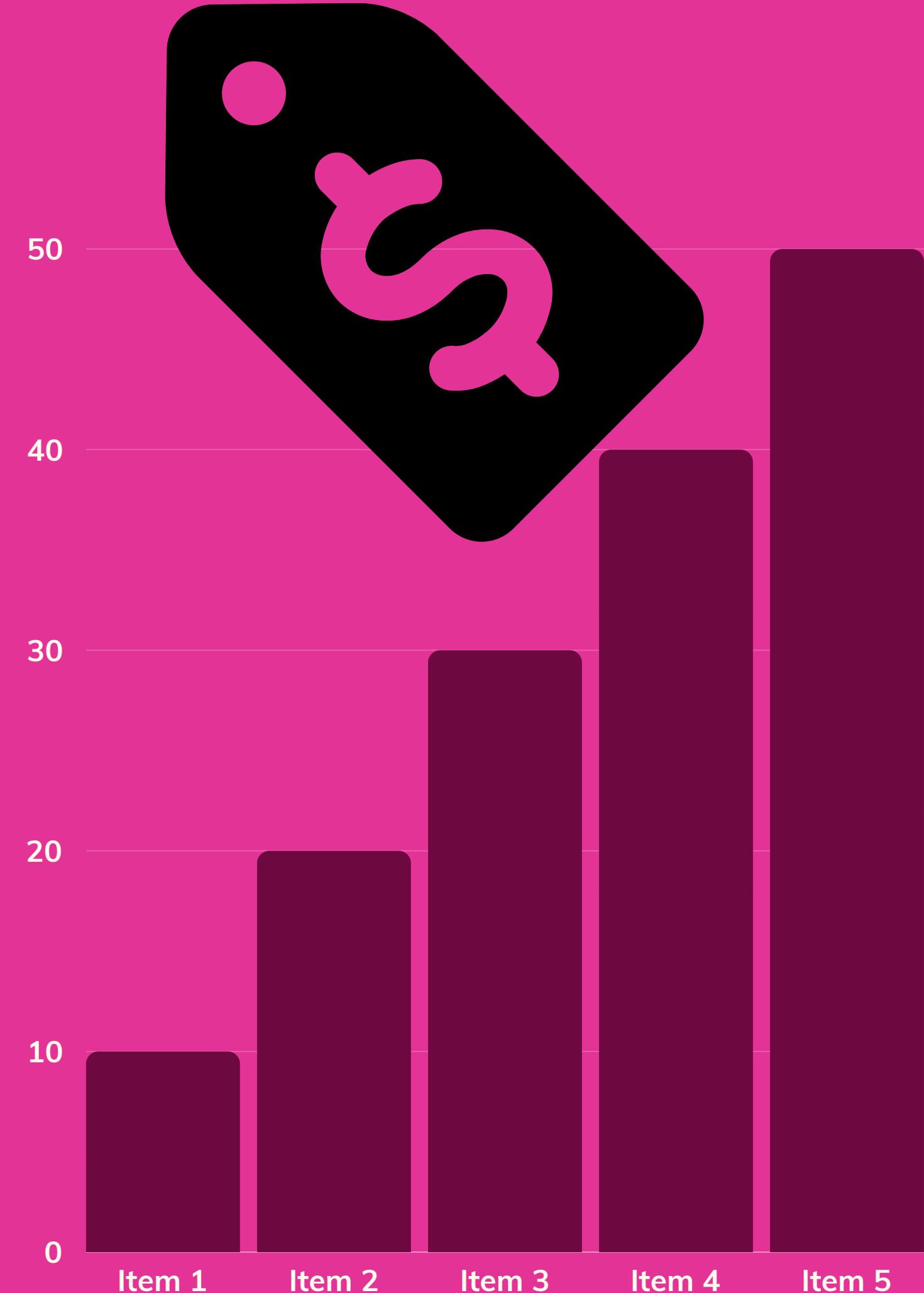
Total Revenue generated from pizzas is 817860.05 Rs.

Highest Price Pizza

After calculating the highest pizza price, you will obtain the specific pizza type (name) along with its corresponding price. This information is valuable for various purposes, such as understanding pricing tiers within your product offerings, identifying premium or specialty items, analyzing profit margins, and making pricing strategy decisions.

```
1 -- Q3. Identify the highest-priced pizza.  
2 • SELECT t.name, p.price  
3   From pizzas p  
4   JOIN pizza_types t ON p.pizza_type_id = t.pizza_type_id  
5   order By p.price desc  
6   limit 1
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
name	price			
The Greek Pizza	35.95			



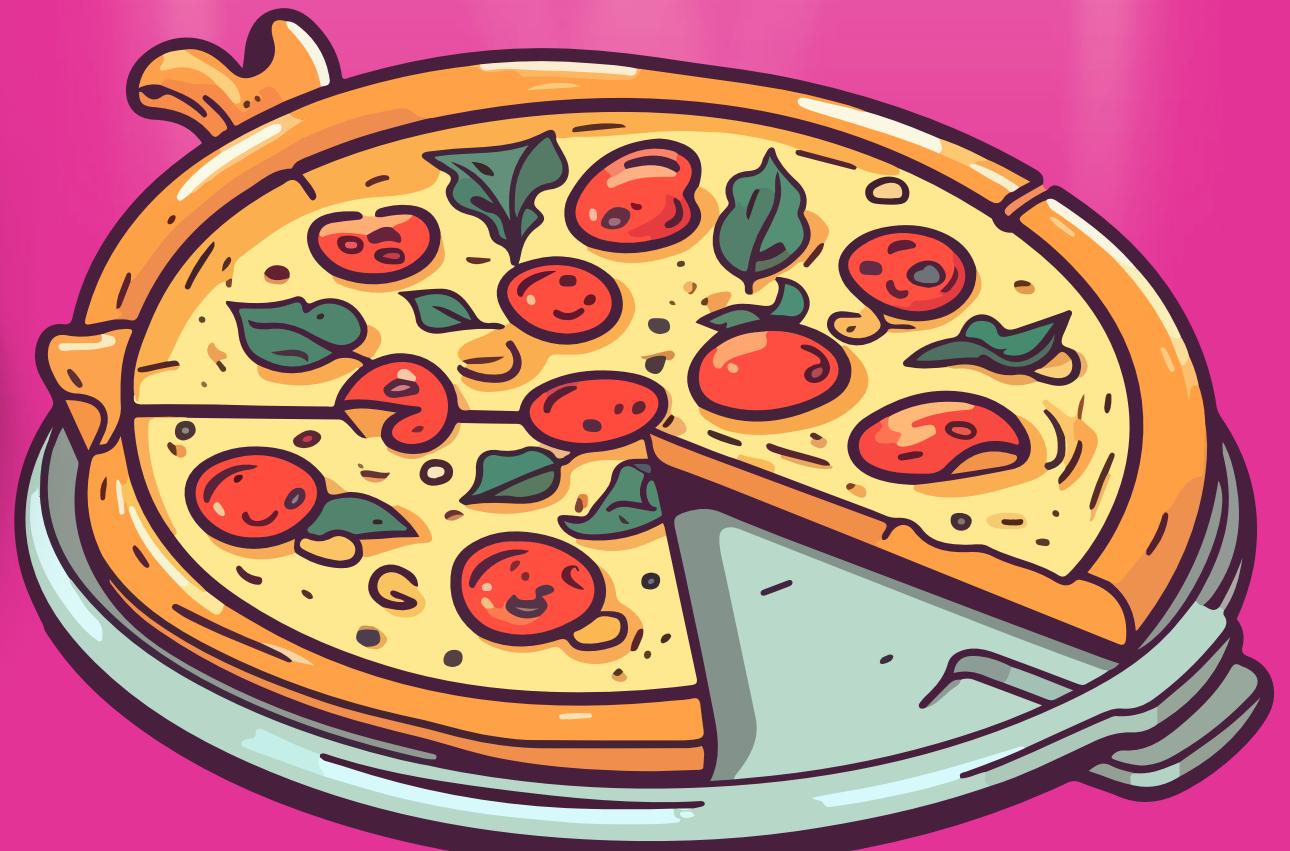
Most Common Ordered Pizzas

Calculating most common ordered pizza

```
1      -- Q.4 Identify the most common pizza size ordered.  
2 •  SELECT p.size, Count(order_details_id) as Total_Count  
3   FROM pizzas p  
4   JOIN order_details o ON p.pizza_id = o.pizza_id  
5   Group By p.size  
6   Order by Total_count Desc
```

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

size	Total_Count
L	18526
M	15385
S	14137
XL	544
XXL	28



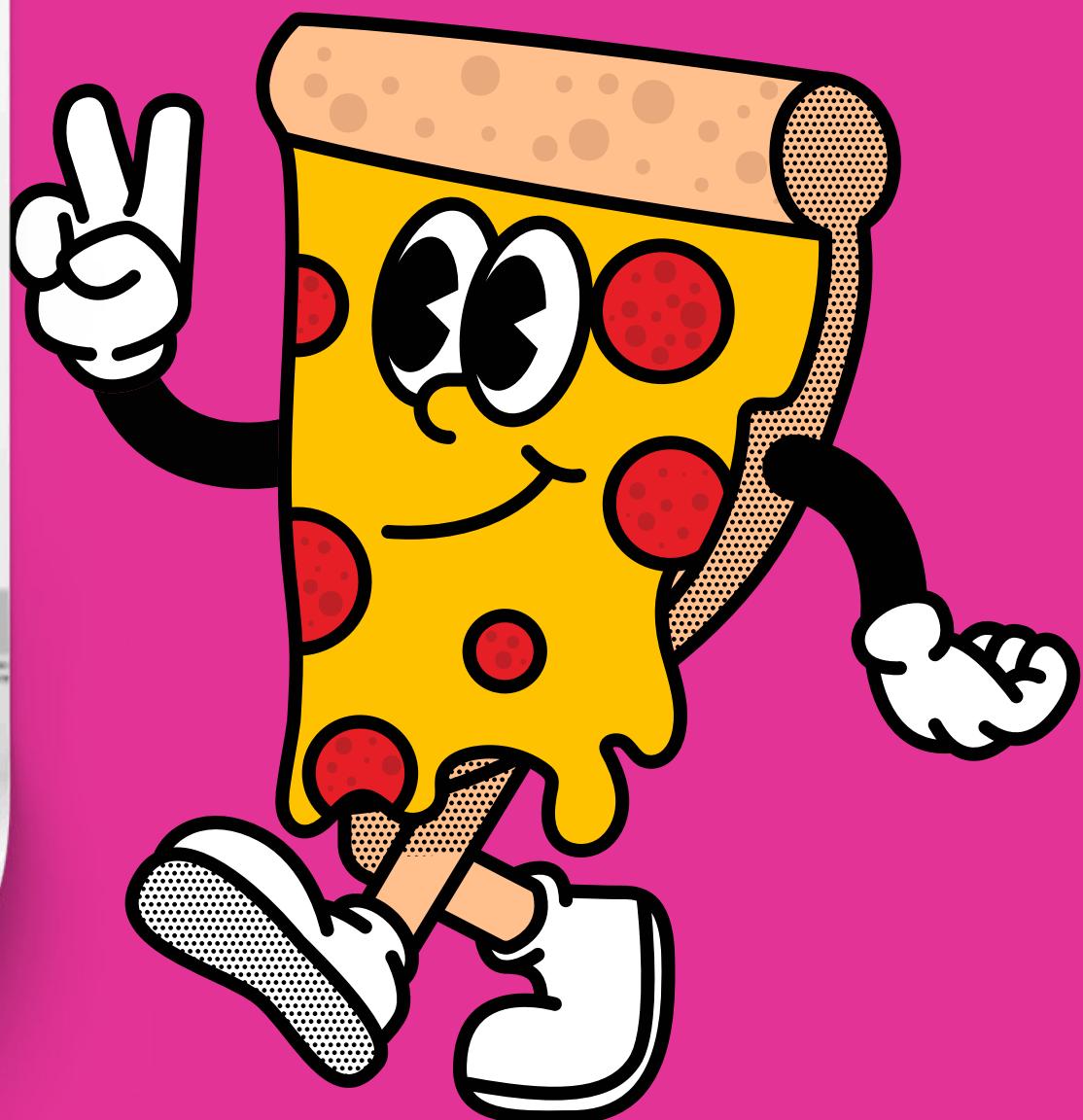
5 MOST ORDERED PIZZASS

Top 5 Most Ordered Pizzas are :

```
1 -- Q.5 List the top 5 most ordered pizza types along with their quantities.  
2 • SELECT t.name, SUM(o.quantity) as total_quantity  
3   FROM pizza_types t  
4   JOIN pizzas p on t.pizza_type_id = p.pizza_type_id  
5   JOIN order_details o On p.pizza_id = o.pizza_id  
6   GROUP BY t.name  
7   ORDER BY total_quantity DESC  
8   LIMIT 5;
```

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

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



TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED

```
1      -- Q.6 Join the necessary tables to find the total quantity of each pizza category ordered.  
2 •  SELECT t.category, SUM(o.quantity) as Total_quantity  
3   FROM pizza_types t  
4   JOIN pizzas p on t.pizza_type_id = p.pizza_type_id  
5   JOIN order_details o On p.pizza_id = o.pizza_id  
6   GROUP BY t.category
```

The screenshot shows a database query results window. At the top, there are buttons for 'Result Grid' (selected), 'Filter Rows', 'Export', and 'Wrap Cell Content'. The main area displays a table with the following data:

	category	Total_quantity
▶	Classic	14888
	Veggie	11649
	Supreme	11987
	Chicken	11050

Total quantity of each pizza category that has been ordered is mainly in 4 categories i.e 1. classic
2.Veggie
3.Supreme,
4.Chicken

1 ----- Q.7 Determine the distribution of orders by hour of the day.-----

2 • SELECT Hour(order_time) as Hours, Count(order_id) as Count

3 FROM orders

4 Group By Hour(order_time)

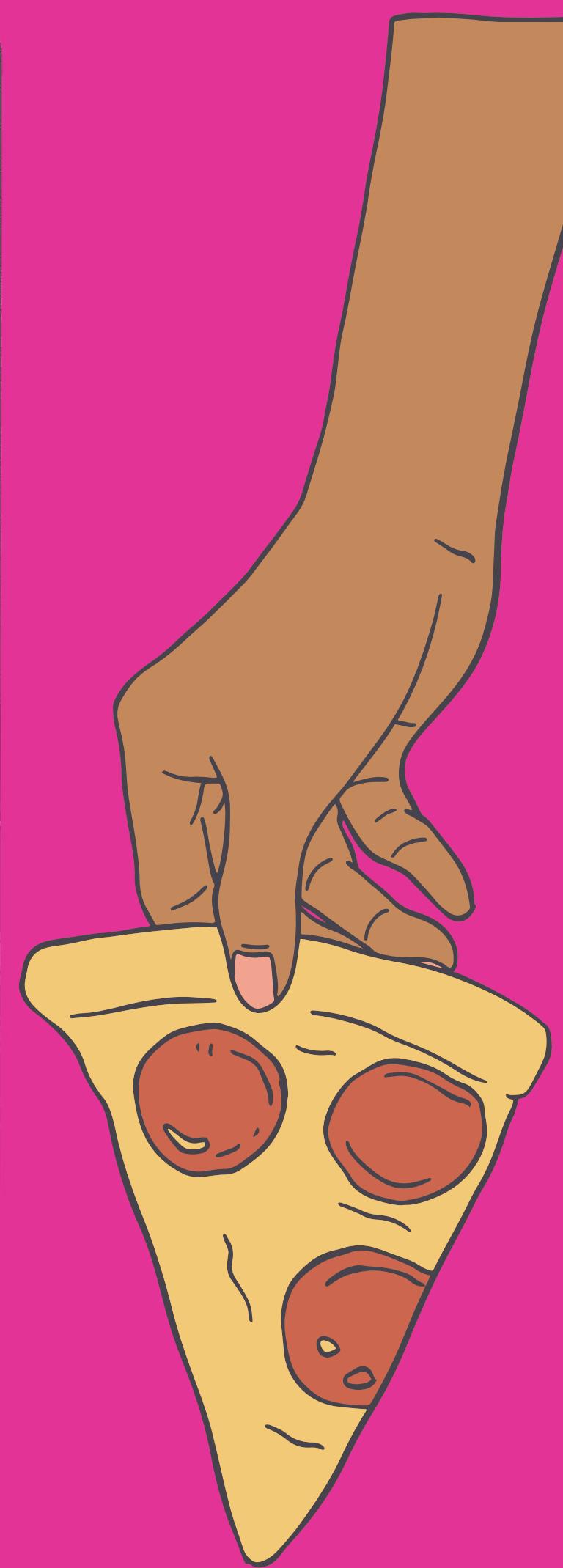
5 Order By Hour(order_time) ASC

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

	Hours	Count
▶	9	1
	10	8
	11	1231
	12	2520
	13	2455
	14	1472
	15	1468
	16	1920



Hourly Sales



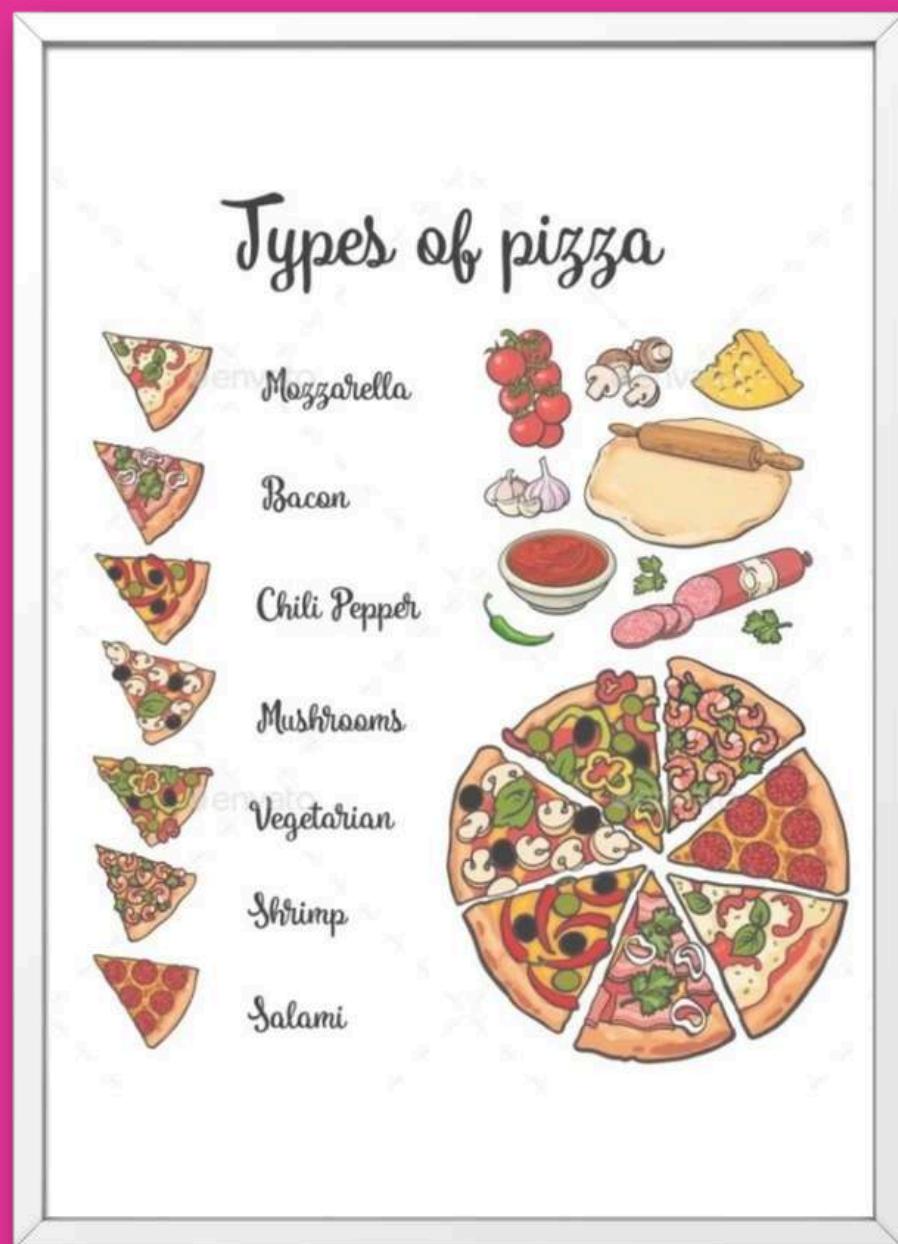
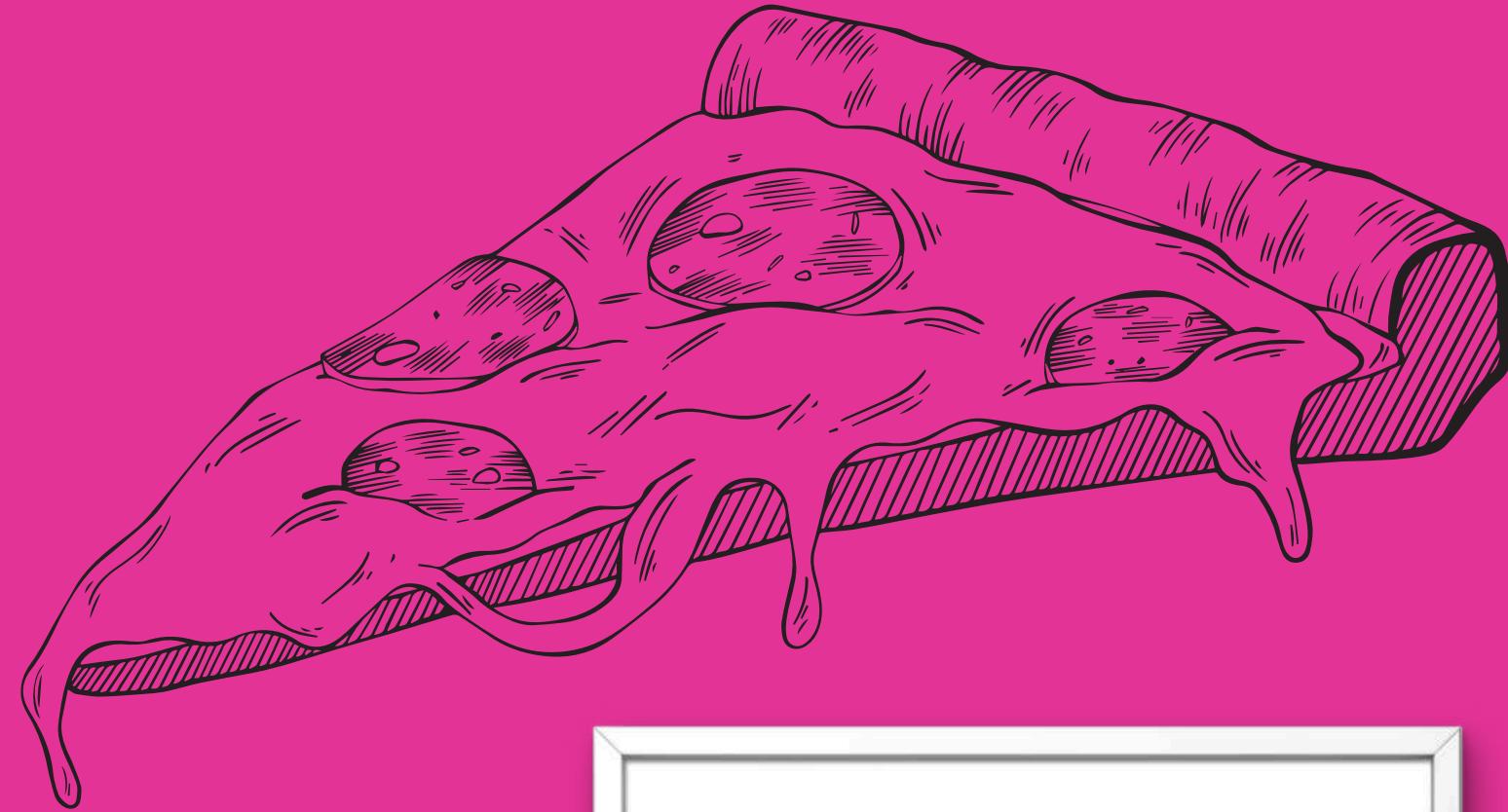
Types of Pizza

Total varieties of pizzas that every person loves and should try

```
1 -- Q.8 Join relevant tables to find the category-wise di
2 • SELECT category, COUNT(name) as Unique_count
3   FROM pizza_types
4   Group By category
```

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

category	Unique_count
Chicken	6
Classic	8
Supreme	9
Veggie	9



Average Pizza Sold per Day

```
1      -- Q.9 Group the orders by date and calculate the average
2 •      SELECT ROUND(AVG(Avg_quantity),2) as Avg_num_pizza_sold
3      FROM (
4          SELECT o.order_date, SUM(od.quantity) as Avg_quantity
5          FROM orders o
6              JOIN order_details od ON o.order_id = od.order_id
7              GROUP BY order_date
8      ) as order_quantity
```

Result Grid		Filter Rows:	Export:	Wrap Cell
Avg_num_pizza_sold				
138.47				



Top 3 Most Ordered Pizzas

These 3 Pizzas are most loved pizzas in our hand

```
1 -- Q.10 Determine the top 3 most ordered pizza types based on revenue.
2 • SELECT t.name, SUM(p.price*o.quantity) as revune
3   FROM pizza_types t
4   JOIN pizzas p ON t.pizza_type_id = p.pizza_type_id
5   JOIN order_details o ON p.pizza_id = o.pizza_id
6   GROUP BY t.name
7   ORDER BY revune DESC
8   LIMIT 3
```

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

	name	revune
▶	The Thai Chicken Pizza	43434.25
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41409.5



```
1 -- Q.11 Calculate the percentage contribution of  
2 -- each pizza type to total revenue.--  
3 • SELECT t.category, ROUND(SUM(p.price * o.quantity) / (  
4     SELECT SUM(p.price * o.quantity)  
5     FROM order_details o  
6     JOIN pizzas p ON p.pizza_id = o.pizza_id  
7 ) * 100, 2) AS revenue  
8     FROM pizza_types t  
9     JOIN pizzas p ON t.pizza_type_id = p.pizza_type_id  
10    JOIN order_details o ON p.pizza_id = o.pizza_id  
11    GROUP BY t.category  
12    ORDER BY revenue DESC
```

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

category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

Percent Wise Revenue of Each pizza Category



Cumulative Revenue Over Time

```
1 -- Q.12 Analyze the cumulative revenue generated over time.  
2 • SELECT order_date, ROUND(SUM(revenue) OVER (ORDER BY order_date),2) as Cumulative  
3 FROM  
4 (SELECT o.order_date, ROUND(SUM(od.quantity*p.price),2) as revenue  
5 FROM order_details od  
6 JOIN pizzas p ON od.pizza_id = p.pizza_id  
7 JOIN orders o ON o.order_id = od.order_id  
8 GROUP BY o.order_date ) AS Sales
```

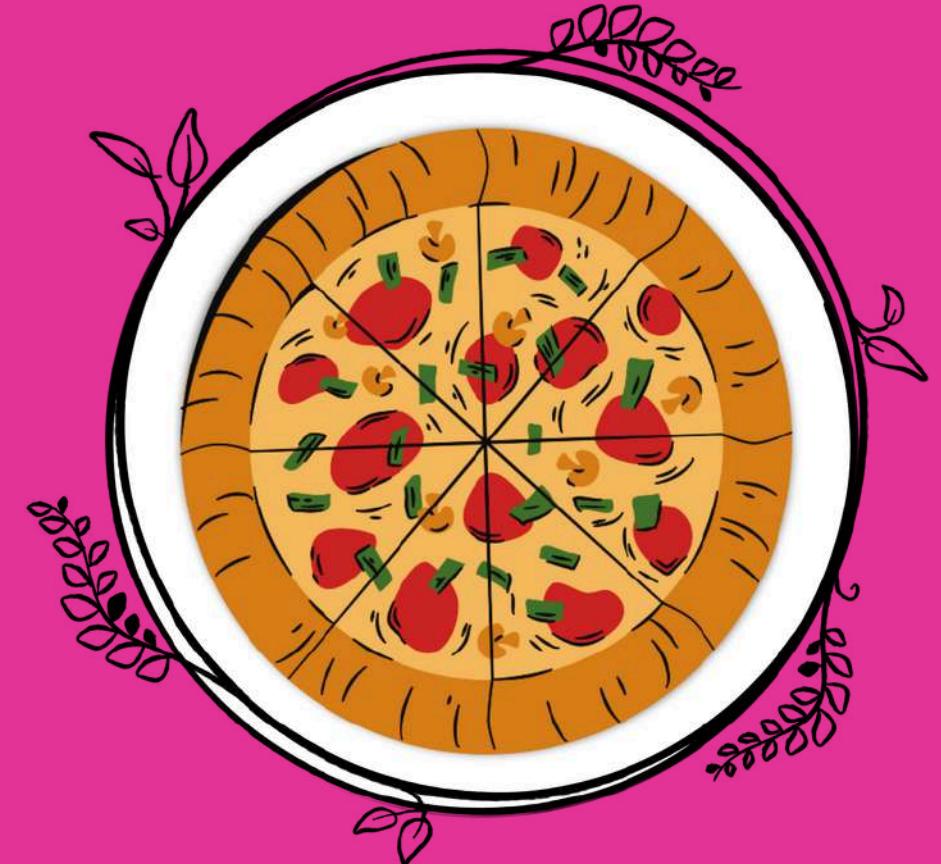
Result Grid | Filter Rows: Export: Wrap Cell Content:

order_date	Cumulative
2015-01-01	2713.85
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	19399.05
2015-01-09	21526.4
2015-01-10	23990.35
2015-01-11	25862.65
2015-01-12	27781.7
2015-01-13	29831.3
2015-01-14	32358.7
2015-01-15	34343.5
2015-01-16	36937.65



```
1 -- Q.13 Determine the top 3 most ordered pizza types based on revenue for each pizza category.  
2 • SELECT category, name, revune, Ranking  
3   FROM  
4   (SELECT category, name, revune, RANK() OVER (PARTITION BY category order by revune DESC) as Ranking  
5     FROM  
6   (SELECT t.category,t.name, ROUND(SUM((p.price)*o.quantity),2) as revune  
7     FROM pizza_types t  
8   JOIN pizzas p ON t.pizza_type_id = p.pizza_type_id  
9   JOIN order_details o ON p.pizza_id = o.pizza_id  
10  GROUP BY t.category,t.name  
11  Order By revune DESC) a) as b  
12 WHERE Ranking <= 3
```

Top Revenue Pizza Picks by Category



	category	name	revune	Ranking
▶	Chicken	The Thai Chicken Pizza	43434.25	1
	Chicken	The Barbecue Chicken Pizza	42768	2
	Chicken	The California Chicken Pizza	41409.5	3
	Classic	The Classic Deluxe Pizza	38180.5	1
	Classic	The Hawaiian Pizza	32273.25	2
	Classic	The Pepperoni Pizza	30161.75	3
	Supreme	The Spicy Italian Pizza	34831.25	1
	Supreme	The Italian Supreme Pizza	33476.75	2
	Supreme	The Sicilian Pizza	30940.5	3
	Veggie	The Four Cheese Pizza	32265.7	1
	Veggie	The Mexicana Pizza	26780.75	2
	Veggie	The Five Cheese Pizza	26066.5	3

THANK YOU
thank you
so much!