

Pizza Sales Analysis

This presentation delves into the pizza sales data using MySQL queries. We'll analyze key metrics such as order volume, revenue, pizza popularity, and ordering patterns. This data will provide valuable insights into customer preferences and sales trends.

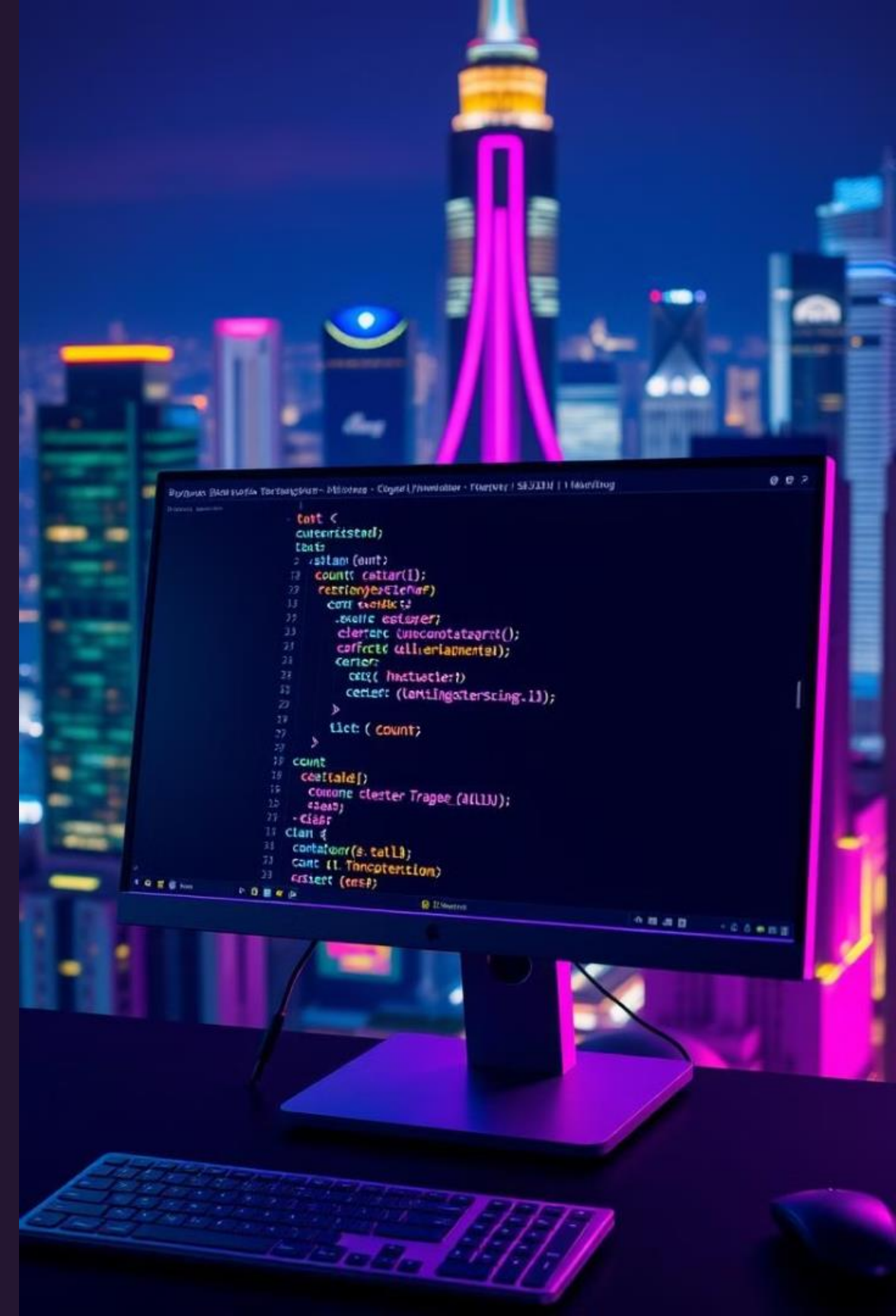


Retrieve the Total Number of Orders

This query calculates the total number of orders placed. This metric is crucial for understanding the overall activity and demand for pizzas.

```
1  -- Retrieve the total number of orders placed
2
3  • select count(order_id) as total_orders from orders;
```

total_orders
21350



Calculate Total Revenue from Pizza Sales

This query determines the total revenue generated from pizza sales. It considers the quantity of each pizza ordered and its corresponding price.

1

-- Calculate the total revenue generated from pizza sales.

2

3

• SELECT

4

ROUND(SUM(order_details.quantity * pizzas.price),

5

2) AS total_sales

6

FROM

7

order_details

8

JOIN

9

pizzas ON pizzas.pizza_id = order_details.pizza_id

Result Grid

total_sales

▶

817860.05

Filter Rows:

Export:

Wrap Cell Content:



Identify the Highest-Priced Pizza

This query pinpoints the pizza with the highest price. This information helps in understanding the pricing strategies and identifying potential high-margin items.

```
1  -- Identify the highest-priced pizza.
2
3  • SELECT
4      pizza_types.name, pizzas.price
5  FROM
6      pizza_types
7      JOIN
8      pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
9  ORDER BY pizzas.price DESC
10 LIMIT 1;
```

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



Most Common Pizza Size Ordered

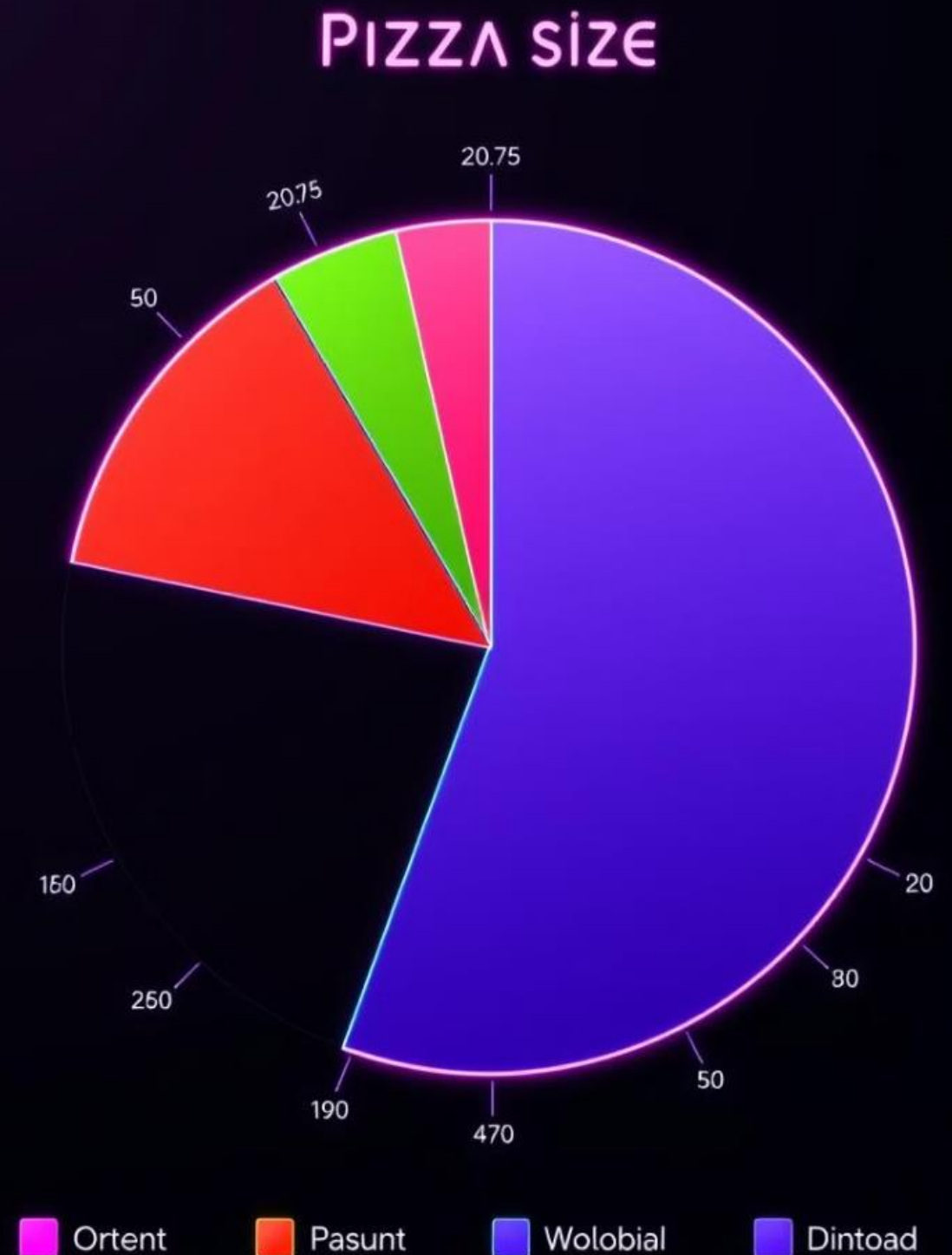
This query reveals the most popular pizza size ordered by customers. This insight is valuable for inventory management and optimizing pizza production.

```
1  -- Identify the most common pizza size ordered.
2
3  • SELECT
4      pizzas.size,
5      COUNT(order_details.order_details_id) AS order_count
6  FROM
7      pizzas
8      JOIN
9      order_details ON pizzas.pizza_id = order_details.pizza_id
10 GROUP BY pizzas.size
11 ORDER BY order_count DESC limit 1;
12
```

Result Grid

	size	order_count
▶	L	18526

Filter Rows: Export: Wrap Cell Content: Fetch rows:



Top 5 Most Ordered Pizza Types

This query identifies the top 5 most ordered pizza types. This information allows us to understand customer preferences and tailor marketing strategies accordingly.

1

-- List the top 5 most ordered pizza types

2

-- along with their quantities.

3

4

• SELECT

5

pizza_types.name, SUM(order_details.quantity) AS quantity

6

FROM

7

pizza_types

8

JOIN

9

pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id

10

JOIN

11

order_details ON order_details.pizza_id = pizzas.pizza_id

12

GROUP BY pizza_types.name

13

ORDER BY quantity DESC

14

LIMIT 5

15

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

Fetch rows:

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

PIZZA						
	Pizza	Scarts	Stader	Shory	Plyill	Stide
Bdnedtize	3333		2660		1244	8660
Bdnedtize	1699		3007		1199	.1660
Bdnoddues	1863		1090		1100	.1960
Bdnedtize	3373		3670		1419	.3650
Bdnedtize	1550		5550		1670	.9980
Bdnedtize	3990		3450		1690	.3860
Bdnedtize	2380		3991		1223	.9870
Sdnedtize	7259		2552		1140	.3640
Bdnedtize	7290		3002		3169	.3640
Bdneddues	5677		3207		1827	.3360
Sdnedtize	5691		3430		1605	.3860
Bdnedtize	3331		3301		1488	3980
Sdneddues	7323		3606		1825	.9500
Bdneodues	6820		3670		1677	.1680
Bdnedtize	2560		5650		1817	.3480
Bdneodues	2225		3370		1574	.7640
Bdneddues	2329		5350		1721	..1300

Pizza Category-Wise Distribution

This query analyzes the distribution of pizzas across different categories. This insight helps in understanding the overall product mix and customer preferences.

```
1  -- Join the necessary tables to find the total quantity
2  -- of each pizza category ordered.
3
4  •  SELECT
5      pizza_types.category,
6      SUM(order_details.quantity) AS quantity
7  FROM
8      pizza_types
9      JOIN
10     pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
11     JOIN
12     order_details ON order_details.pizza_id = pizzas.pizza_id
13  GROUP BY pizza_types.category
14  ORDER BY quantity DESC
```

Result Grid

Filter Rows:

Export:

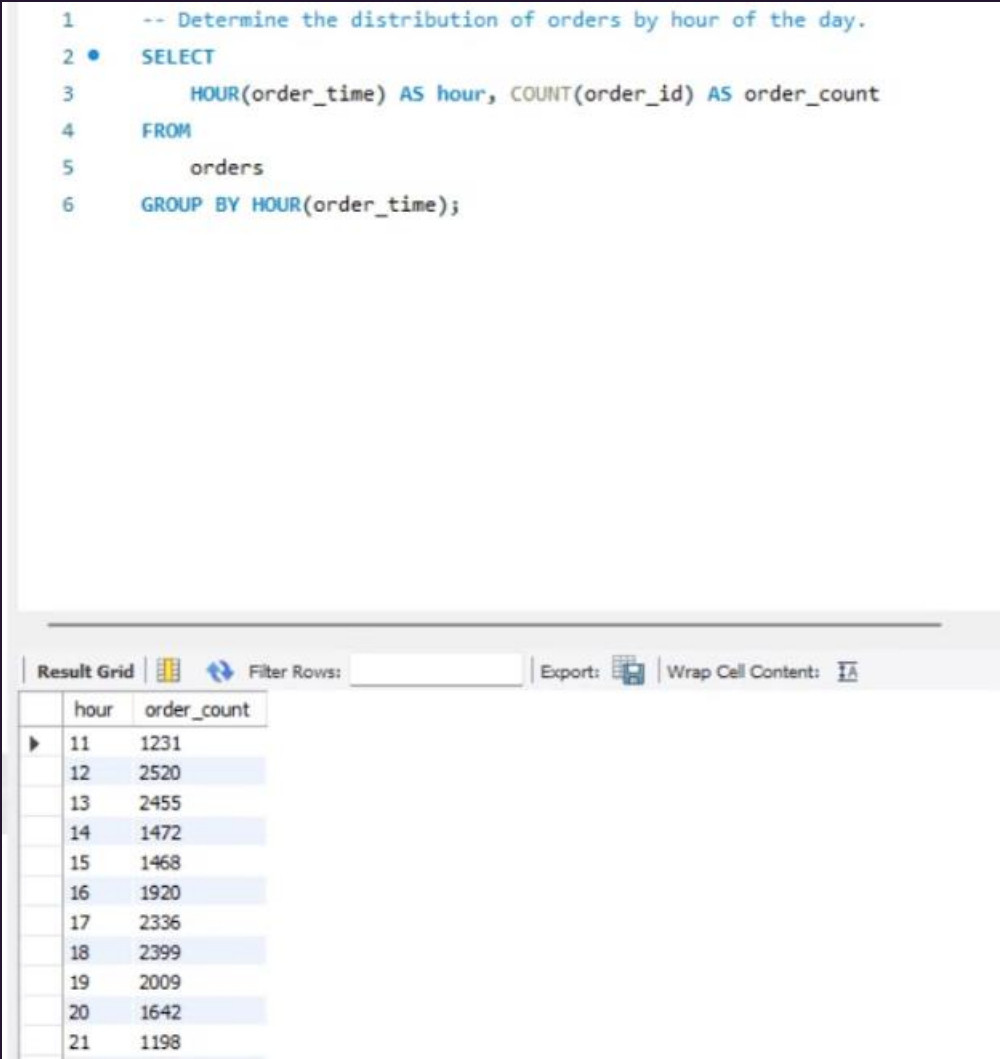
Wrap Cell Content:

	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050



Order Distribution by Hour of the Day

This query reveals the distribution of orders by the hour of the day. This information allows us to understand peak ordering times and optimize staffing levels.



Category-Wise Pizza Distribution

This query provides a breakdown of pizzas by category, indicating the relative popularity of different pizza types.

1

-- Join relevant tables to find the category-wise

2

-- distribution of pizzas.

3

4

• select category, count(name) from pizza_types

5

group by category

6

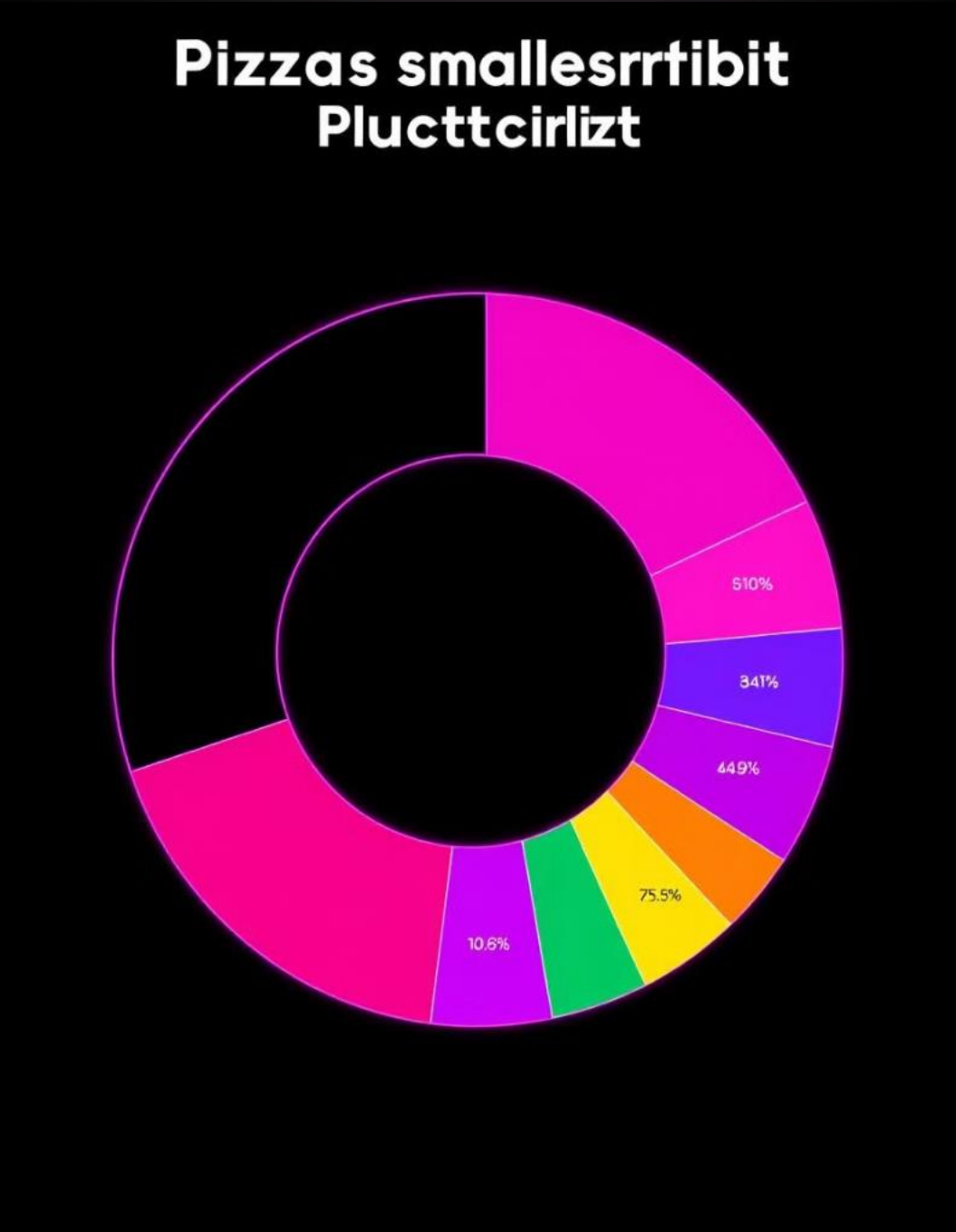
Result Grid

Filter Rows:

Export:

Wrap Cell Content:

	category	count(name)
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9



Average Pizzas Ordered Per Day

This query calculates the average number of pizzas ordered per day. This metric helps in understanding the overall sales volume and identifying any trends or fluctuations.

1

-- Group the orders by date and calculate the

2

-- average number of pizzas ordered per day.

3

4

• SELECT

5

orders.order_date, SUM(order_details.quantity)

6

FROM

7

orders

8

JOIN

9

order_details ON orders.order_id = order_details.order_id

10

GROUP BY order_date;

11

Result Grid

Filter Rows:

Export:


Wrap Cell Content:

	order_date	SUM(order_details.quantity)
▶	2015-01-01	162
	2015-01-02	165
	2015-01-03	158
	2015-01-04	106
	2015-01-05	125
	2015-01-06	147
	2015-01-07	138
	2015-01-08	173
	2015-01-09	127
	2015-01-10	146
	2015-01-11	116



Top 3 Most Ordered Pizza Types by Revenue

```
1  -- Determine the top 3 most ordered pizza types
2  -- based on revenue.
3
4  • SELECT
5      pizza_types.name,
6      sum(order_details.quantity * pizzas.price) AS revenue
7  FROM
8      pizza_types
9      JOIN
10     pizzas
11     ON pizza_types.pizza_type_id = pizzas.pizza_type_id
12     JOIN
13     order_details
14     ON order_details.pizza_id = pizzas.pizza_id
15     group by pizza_types.name order by revenue desc limit 3;
```

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

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



percentage contribution of each pizza type to total revenue.

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

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

	category	revenue
▶	Classic	220053.1000000001
	Supreme	208196.99999999822
	Chicken	195919.5
	Veggie	193690.45000000298



Analyze the cumulative revenue generated over time.

```
1  -- Analyze the cumulative revenue generated over time.
2
3  • select order_date,
4     sum(revenue) over(order by order_date) as cum_revenue
5     from
6     (select orders.order_date,
7        sum(order_details.quantity * pizzas.price) as revenue
8        from order_details join pizzas
9        on order_details.pizza_id = pizzas.pizza_id
10       join orders
11       on orders.order_id = order_details.order_id
12       group by orders.order_date) as sales;
13
```

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

	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	19399.05
	2015-01-09	21526.4
	2015-01-10	23990.350000000002
	2015-01-11	25862.65



Recommendations

1. **Focus on High-Revenue Pizzas:** From the analysis, identifying the top pizza types by revenue can inform which items to promote, both for sales and marketing campaigns.
2. **Optimize Pizza Sizes:** Understanding the most common pizza sizes ordered can help streamline inventory management and production efficiency, ensuring you have the right sizes in stock.
3. **Peak Ordering Times:** The distribution of orders by hour could be valuable for staffing and preparing for peak times, enhancing customer experience.
4. **Category Diversification:** The category-wise distribution can highlight underperforming pizza categories, allowing for targeted promotions or adjustments in the menu.
5. **Revenue Strategy:** The percentage contribution analysis can guide promotional strategies, ensuring that higher-margin pizzas or popular ones are emphasized.



Conclusion

By analyzing the total orders, revenue, and customer preferences across various dimensions (pizza type, category, size, etc.), we gain a deeper understanding of customer behavior. This data-driven insight enables better decision-making in terms of inventory management, staffing, marketing strategies, and menu optimization. With a focus on high-revenue pizzas and peak ordering times, businesses can enhance profitability and customer satisfaction.

