

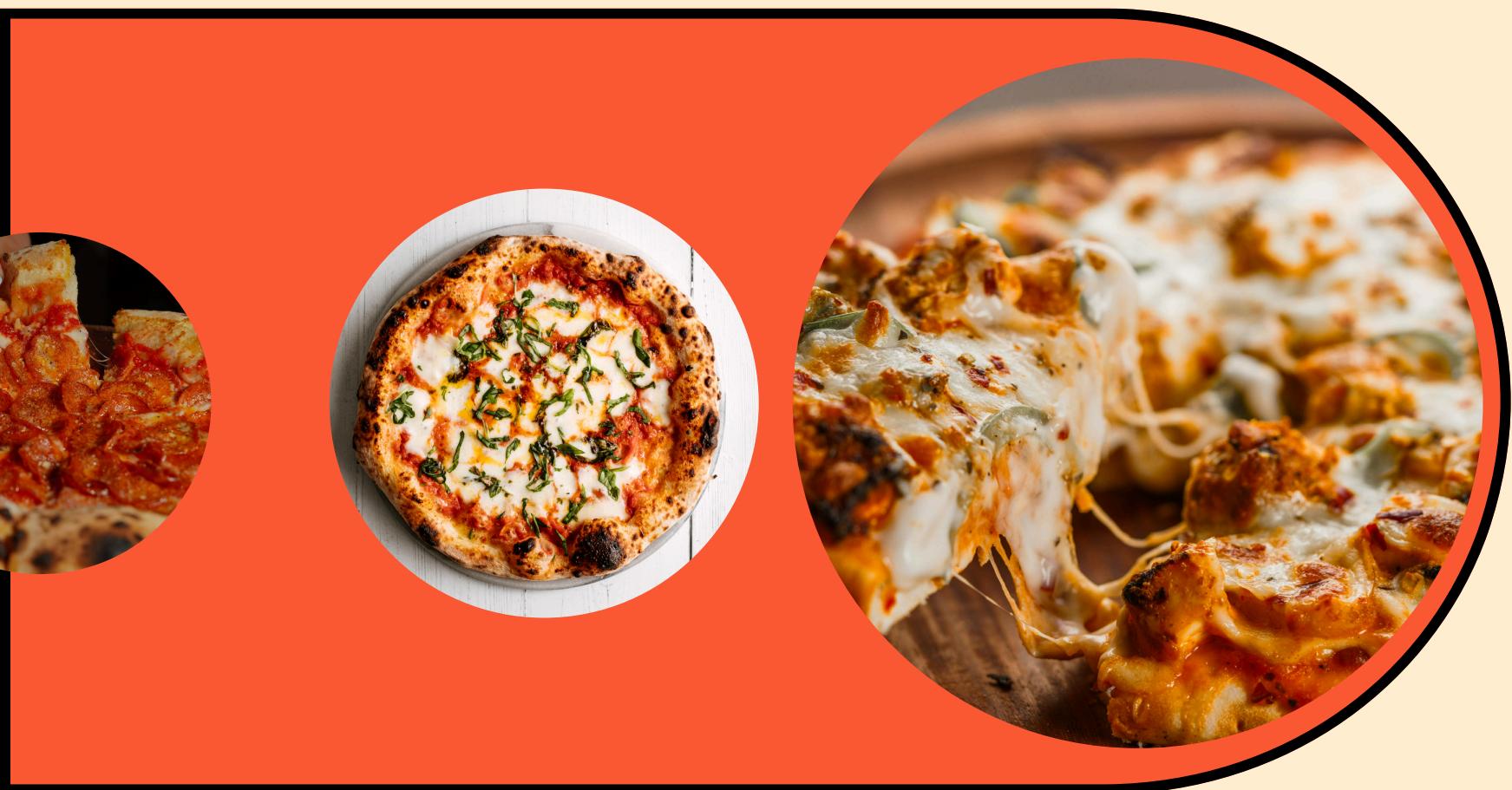
# **Business Insights from PIZZA Sales Data Using SQL**



**V.Harsha vardhan**



# Welcome



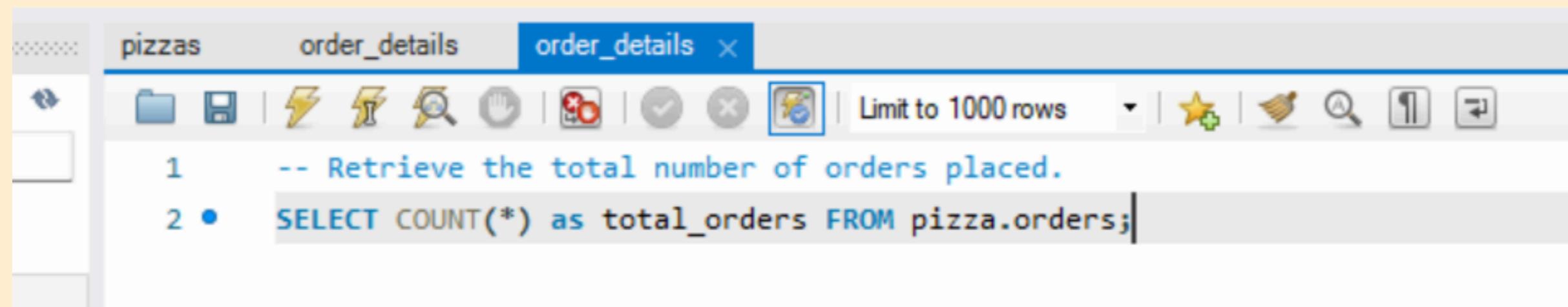
- Objective: Analyze sales and order data
- Dataset: orders, order\_details, pizzas, pizza\_types
- Tool Used: MySQL

# Business Questions

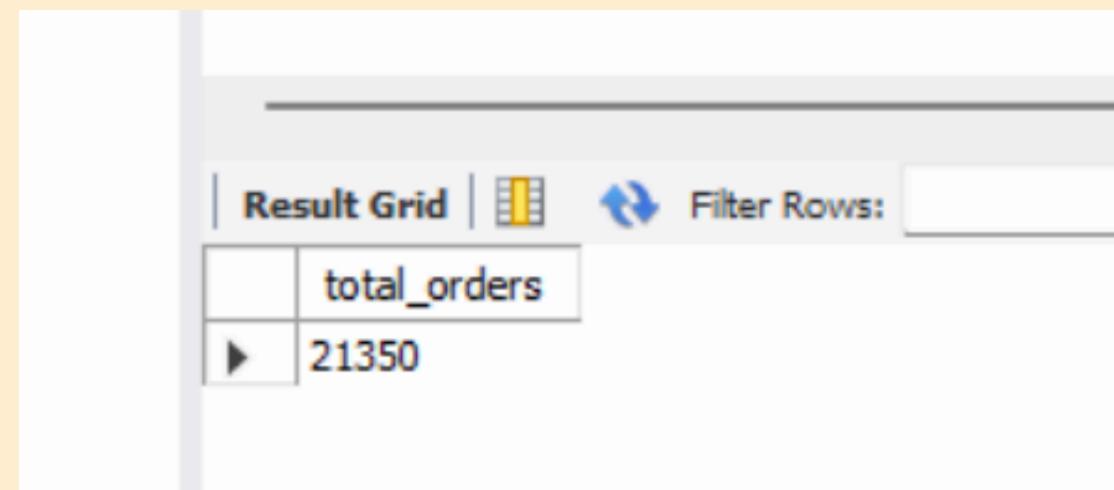


- 1) Retrieve the total number of orders placed.
- 2) Calculate the total revenue generated from pizza sales.
- 3) Identify the highest-priced pizza.
- 4) Identify the most common pizza size ordered.
- 5) List the top 5 most ordered pizza types along with their quantities.
- 6) Join the necessary tables to find the total quantity of each pizza category ordered.
- 7) Determine the distribution of orders by hour of the day.
- 8) Join relevant tables to find the category-wise distribution of pizzas.
- 9) Group the orders by date and calculate the average number of pizzas ordered per day.
- 10) Determine the top 3 most ordered pizza types based on revenue.
- 11) Calculate the percentage contribution of each pizza type to total revenue.
- 12) Analyze the cumulative revenue generated over time.
- 13) Determine the top 3 most ordered pizza types based on revenue for each pizza category.

# Retrieve the total number of orders placed.



```
-- Retrieve the total number of orders placed.  
SELECT COUNT(*) as total_orders FROM pizza.orders;
```



total_orders
21350



# Calculate the total revenue generated from pizza sales.

```
1  -- Calculate the total revenue generated from pizza sales.  
2 • SELECT SUM(order_details.quantity*pizzas.price) as total_sales  
3   from order_details  
4   join pizzas on pizzas.pizza_id = order_details.pizza_id
```

Result Grid	
	total_sales
▶	817860.049999993



# Identify the highest-priced pizza.

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

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

	name	price
▶	The Greek Pizza	35.95





# Identify the most common pizza size ordered.

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

---

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

size	order_count
L	18526

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



```
1 -- List the top 5 most ordered pizza types along with their quantities.
2 • SELECT
3     pizza_types.name, SUM(order_details.quantity) AS quantity
4 FROM
5     pizza_types
6     JOIN
7     pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
8     JOIN
9     order_details ON order_details.pizza_id = pizzas.pizza_id
10    GROUP BY pizza_types.name
11    ORDER BY quantity DESC
12    LIMIT 5;
13
```

The screenshot shows a MySQL query editor window. At the top, there's a toolbar with various icons. Below the toolbar is a code editor containing the SQL query. The code is color-coded: numbers are blue, keywords like 'SELECT' and 'JOIN' are red, and table names and column names are black. The query itself is a standard SQL statement for selecting the top 5 pizza types by quantity. At the bottom of the window is a results grid. The grid has two columns: 'name' and 'quantity'. It displays five rows of data. The first row, 'The Classic Deluxe Pizza', is highlighted with a light blue background. The other four rows are white. The results grid also has a toolbar at the top with buttons for 'Result Grid', 'Filter Rows', 'Export', 'Wrap Cell Content', and '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

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



# Determine the distribution of orders by hour of the day.



```
1  -- Determine the distribution of orders by hour of the day.  
2  
3 • SELECT  
4      HOUR(order_time) AS hour, COUNT(order_id) AS order_count  
5  FROM  
6      orders  
7  GROUP BY HOUR(order_time)
```

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

hour	order_count
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2000

Result 2 X

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



```
1      -- find the category-wise distribution of pizzas.  
2  
3 •  SELECT  
4          category, COUNT(name)  
5      FROM  
6          pizza_types  
7  
8      GROUP BY category
```

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

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

SQL File 4\* SQL File 5\* SQL File 6\* SQL File 7\* SQL File 8\* SQL File 9\* SQL File 10\* SQL File 11\*

Limit to 1000 rows

```
1 -- Group the orders by date and calculate the average number of pizzas ordered per day.
2 • SELECT
3     AVG(quantity)
4 FROM
5     (SELECT
6         orders.order_date, SUM(order_details.quantity) AS quantity
7     FROM
8         orders
9     JOIN order_details ON orders.order_id = order_details.order_id
10    GROUP BY orders.order_date) AS order_q_date;
```

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

	avg(quantity)
▶	138.4749



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



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

```
1 -- Determine the top 3 most ordered pizza types based on revenue.
2 • SELECT
3     pizza_types.name, SUM(order_details.quantity*pizzas.price) AS quantity
4 FROM
5     pizza_types
6     JOIN
7         pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
8     JOIN
9         order_details ON order_details.pizza_id = pizzas.pizza_id
10 GROUP BY pizza_types.name
11 ORDER BY quantity DESC
12 LIMIT 3;
13
```

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

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

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



# Analyze the cumulative revenue generated over time.



```
SQL File 5* SQL File 6* SQL File 7* SQL File 8* SQL File 9* SQL File 10* SQL File
... | Limit to 1000 rows | ... | ...
1   -- Analyze the cumulative revenue generated over time
2
3 •  SELECT
4     order_date,
5     SUM(revenue) OVER (ORDER BY order_date) AS cum_revenue
6   FROM
7   (
8     SELECT
9       orders.order_date,
10      SUM(order_details.quantity * pizzas.price) AS revenue
11    FROM order_details
12    JOIN pizzas
13      ON order_details.pizza_id = pizzas.pizza_id
14    JOIN orders
15      ON orders.order_id = order_details.order_id
16    GROUP BY orders.order_date
17  ) AS sales;
18
```

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

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
2015-01-12	27781.7

Result 1 ×

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

```
1  -- Determine the top 3 most ordered pizza types based on revenue for each pizza category.
2 • select name,revenue from
3   (select name,category,revenue ,
4    rank() over(partition by category order by revenue desc) as rankk from
5   (SELECT
6     pizza_types.category,
7     pizza_types.name,
8     SUM(order_details.quantity * pizzas.price) AS revenue
9   FROM pizza_types
10  JOIN pizzas
11    ON pizza_types.pizza_type_id = pizzas.pizza_type_id
12  JOIN order_details
13    ON order_details.pizza_id = pizzas.pizza_id
14  GROUP BY pizza_types.category, pizza_types.name) as a) as b
15  where rankk <=3;
16
```

Result Grid	
	name
▶	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
	The Spicy Italian Pizza
	34831.25
	The Italian Supreme Pizza
	33476.75
	The Sicilian Pizza
	30940.5
	The Four Cheese Pizza
	32265.70000000065



# THANK YOU!

