



PIZZA SALES REPORT

20 24

presented by
Gaurav Gupta





INTRODUCTION

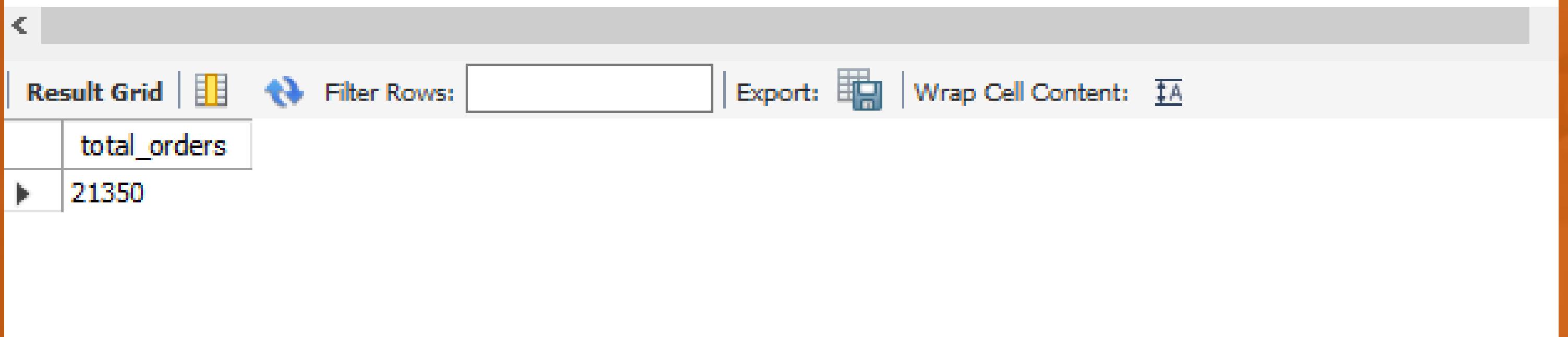
Welcome to our sales report presentation. In this report, we will analyze and review our company PIZZA MANIA's sales performance over a specific period. By examining key metrics and trends, we aim to provide valuable insights that inform decision-making and drive business growth.



presented by
Gaurav Gupta

QUERY 01

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



The screenshot shows a MySQL query editor interface. At the top, there is a code editor with the following SQL query:

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

Below the code editor is a toolbar with the following buttons:

- Result Grid (highlighted)
- Filter Rows:
- Export:
- Wrap Cell Content:

Underneath the toolbar is a table with one row of data:

	total_orders
▶	21350

QUERY 02



```
1      -- Calculate the total revenue generated from pizza sales.  
2      SELECT  
3          ROUND(SUM(order_details.quantity * pizzas.price),  
4                  2) AS total_sales_revenue  
5      FROM  
6          order_details  
7      JOIN  
8          pizzas ON pizzas.pizza_id = order_details.pizza_id
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	total_sales_revenue			
▶	1635720.1			

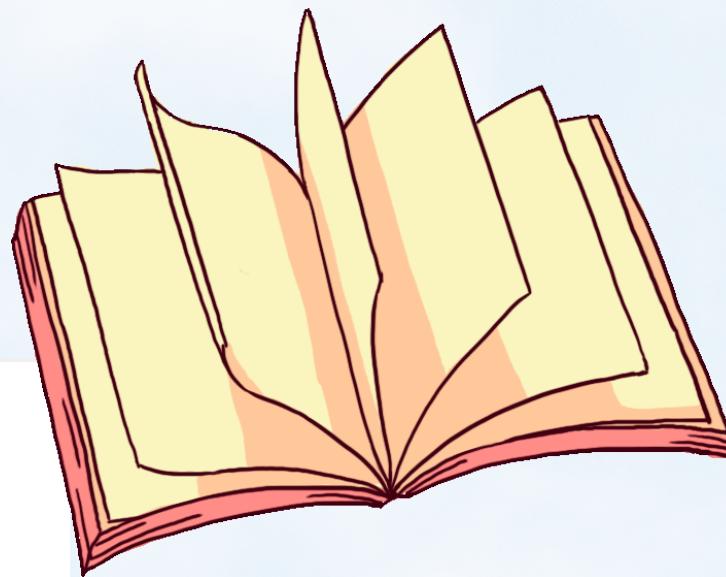
QUERY 03

```
1  -- Identify the highest-priced pizza.  
2  
3  SELECT  
4      pizza_types.name, pizzas.price  
5  FROM  
6      pizza_types  
7          JOIN  
8      pizzas USING (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

QUERY 04



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

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

	order_count	size
▶	37052	L
	30770	M
	28274	S
	1088	XL
	56	XXL

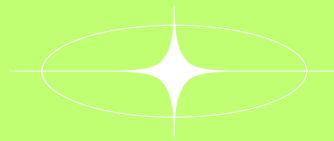
QUERY 05

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

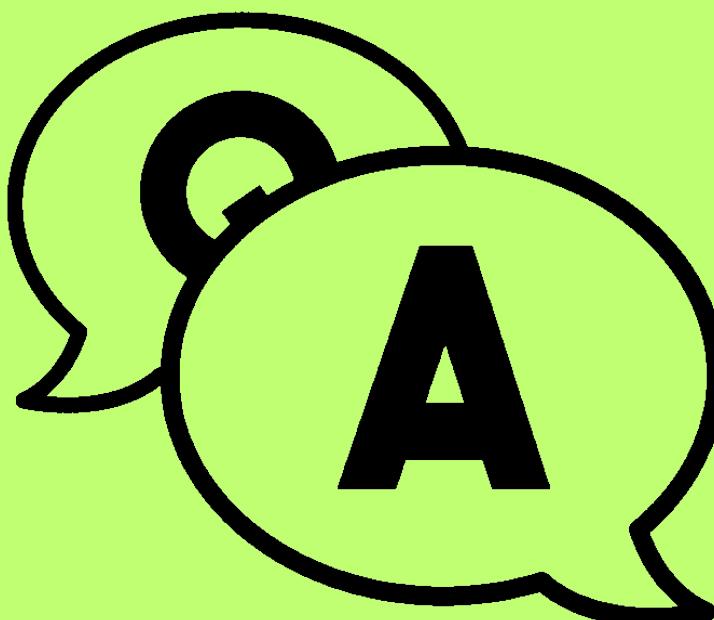


Result Grid | Filter Rows: Export: Wrap Cell Content: Fetch rows:

quantity	name
4906	The Classic Deluxe Pizza
4864	The Barbecue Chicken Pizza
4844	The Hawaiian Pizza
4836	The Pepperoni Pizza
4742	The Thai Chicken Pizza



QUERY 06



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

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

	quantity	category
▶	29776	Classic
	23974	Supreme
	23298	Veggie
	22100	Chicken



QUERY 07

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

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

	hours	order_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





QUERY 08

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

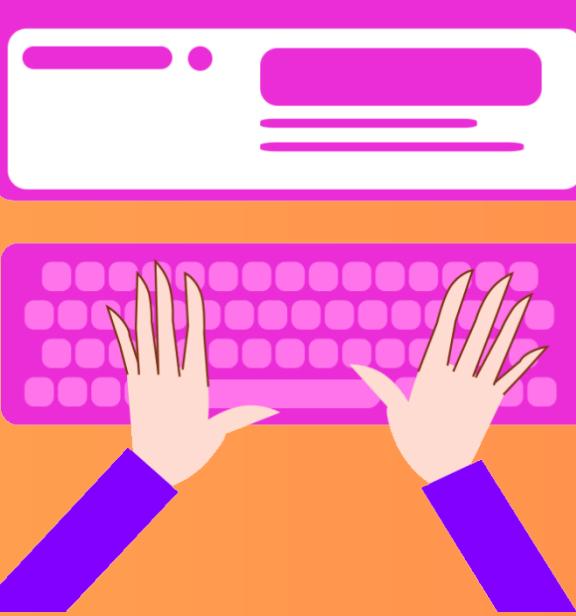


Result Grid |  Filter Rows: Export:  Wrap Cell Content: 

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

QUERY 09

```
1  -- Group the orders by date and calculate the average number of pizzas ordered per day.  
2  SELECT  
3      ROUND(AVG(quantity), 0) AS Average_number_of_pizzas_ordered_per_day  
4  FROM  
5  (SELECT  
6      orders.order_date, SUM(order_details.quantity) AS quantity  
7  FROM  
8      orders  
9  JOIN order_details USING (order_id)  
10     GROUP BY orders.order_date) AS order_quantity;
```



Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	Average_number_of_pizzas_ordered_per_day			
▶	277			

QUERY 10



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

< []

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

	name	revenue
▶	The Thai Chicken Pizza	86868.5
	The Barbecue Chicken Pizza	85536
	The California Chicken Pizza	82819

QUERY 11

```
1      -- Calculate the percentage contribution of each pizza type to total revenue.
2 •  SELECT
3      pizza_types.category,
4      ROUND(SUM(order_details.quantity * pizzas.price) / (SELECT
5          ROUND(SUM(order_details.quantity * pizzas.price),
6          2) AS total_sales
7      FROM
8          order_details
9      JOIN
10         pizzas ON pizzas.pizza_id = order_details.pizza_id) * 100,
11      2) AS revenue
12  FROM
13      pizza_types
14      JOIN
15         pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
16      JOIN
17         order_details USING (pizza_id) GROUP BY pizza_types.category 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

QUERY 12



```
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  
9  JOIN pizzas  
10 USING(pizza_id)  
11 JOIN orders  
12 USING(order_id)  
13 GROUP BY order_date) AS sales;
```

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

	order_date	cum_revenue
▶	2015-01-01	5427.7
	2015-01-02	10891.5
	2015-01-03	16216.3
	2015-01-04	19727.2
	2015-01-05	23859.1
	2015-01-06	28717
	2015-01-07	33121.4
	2015-01-08	38798.1
	2015-01-09	43052.799999999996

QUERY 13

```
1  -- Determine the top 3 most ordered pizza types based on revenue for each pizza category.
2 •  SELECT name, revenue, category
3   FROM
4   (SELECT category, name, revenue,
5    RANK() OVER(PARTITION BY category ORDER BY revenue DESC) AS rn
6   FROM
7   (SELECT pizza_types.category, pizza_types.name,
8    SUM((order_details.quantity)*pizzas.price) AS revenue
9   FROM pizza_types
10  JOIN pizzas
11  USING(pizza_type_id)
12  JOIN order_details
13  USING(pizza_id)
14  GROUP BY pizza_types.category, pizza_types.name) AS a) AS b
15 WHERE rn<= 3;
```

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

	name	revenue	category
▶	The Thai Chicken Pizza	86868.5	Chicken
	The Barbecue Chicken Pizza	85536	Chicken
	The California Chicken Pizza	82819	Chicken
	The Classic Deluxe Pizza	76361	Classic
	The Hawaiian Pizza	64546.5	Classic
	The Pepperoni Pizza	60323.5	Classic
	The Spicy Italian Pizza	69662.5	Supreme



+91 7014797072

guptagaurav061@gmail.com

THANK YOU!



Thank you for your attention to our sales report presentation. If you have any questions or would like to discuss the findings in more detail, please don't hesitate to reach out to our sales team. We appreciate your continued support and partnership.