

# Data Analysis

Pizzahut

# Objectives:

Understand customer preferences.

Identify best-selling pizzas and slow-moving items.

Analyze seasonal trends and geographical sales performance.

Stakeholders: Pizza Hut management, sales teams, marketing teams, data analytics team.

# About the Company

- Pizza Hut is a pizza chain that was founded in 1958 by two brothers in Wichita, Kansas.



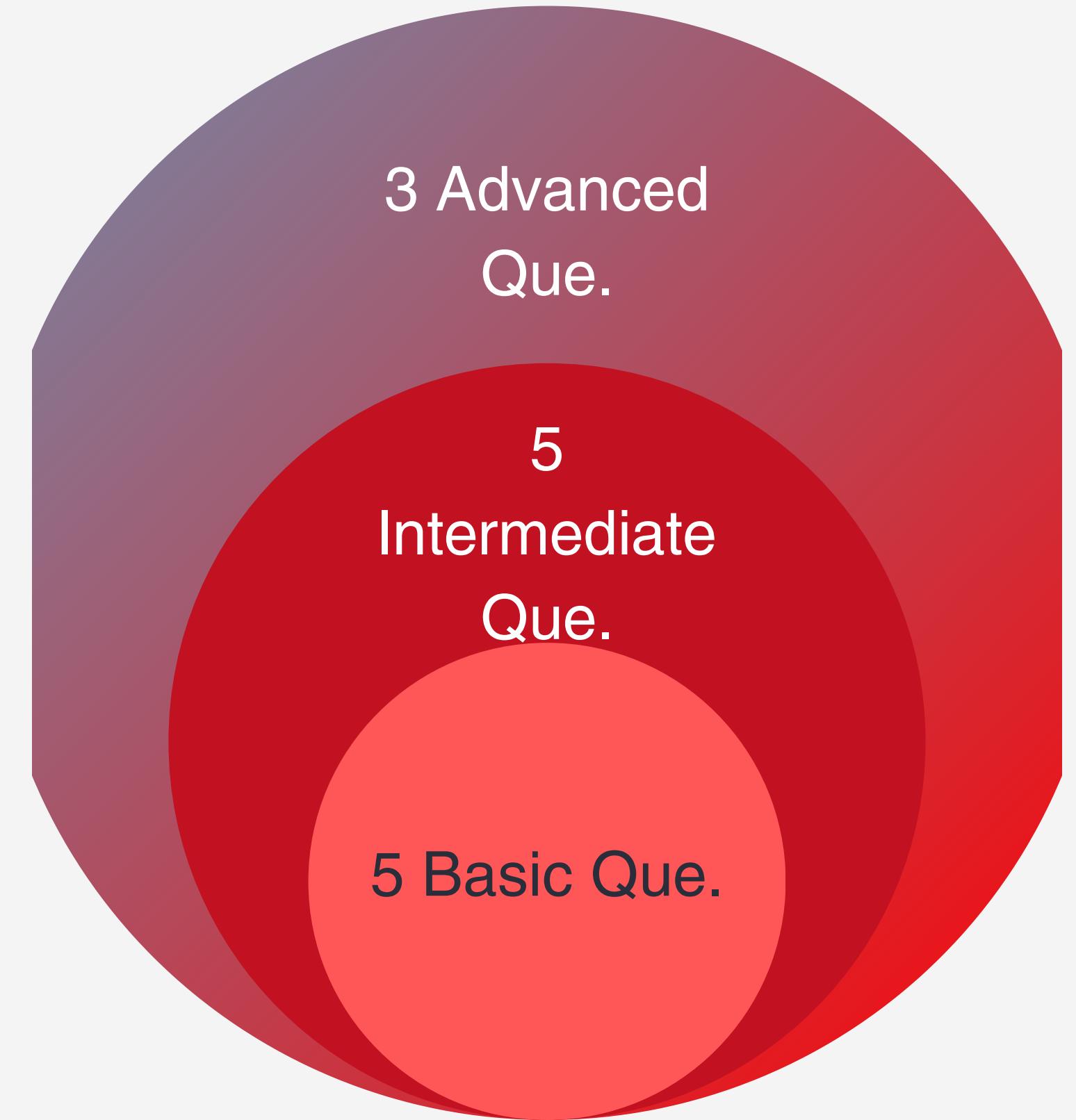
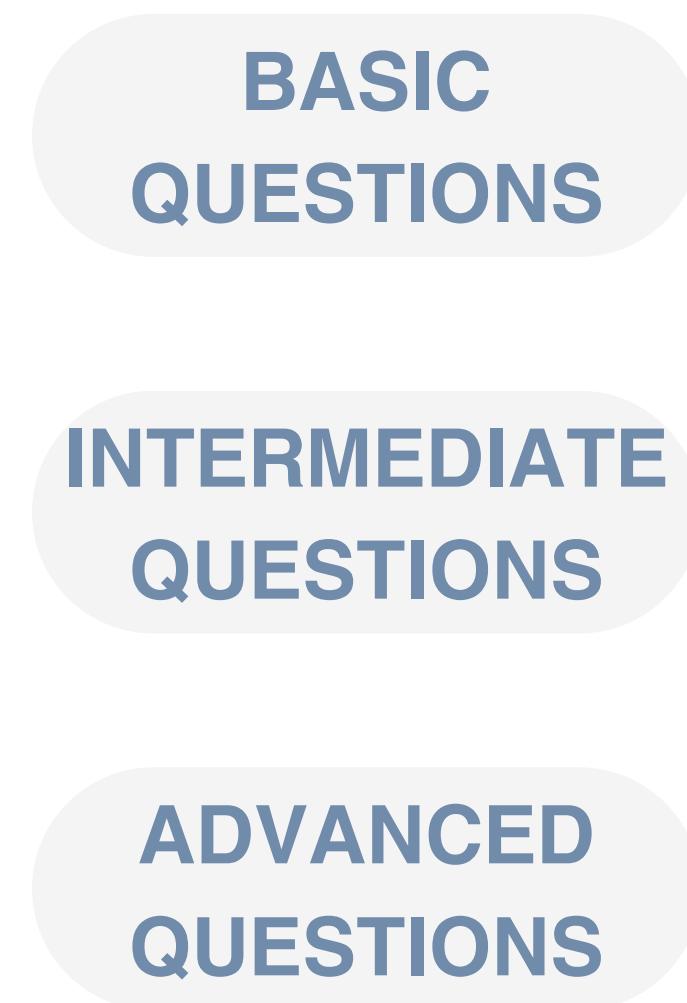
[Back to Agenda](#)

# Data Analytics

- Data analysis is the process of systematically examining, transforming, and modeling data to uncover useful information, support decision-making, and drive actionable insights. It involves various techniques such as descriptive statistics, data cleaning, data visualization, and the use of statistical tools to identify trends, patterns, and relationships within datasets. By employing methodologies such as regression analysis, clustering, and hypothesis testing, data analysts can extract meaningful conclusions from raw data.



# Levels of Question



# Our Goals

Briefly elaborate on what you want to discuss.



**# 1**

Collection of Data



**# 2**

Solving Question Using the SQL Queries



**# 3**

Conclusions Over Data

[Back to Agenda](#)

# Basic Questions

Briefly elaborate on what you want to discuss.

1

Retrieve the total number of orders placed.

2

Calculate the total revenue generated from pizza sales.

3

Identify the highest-priced pizza.

4

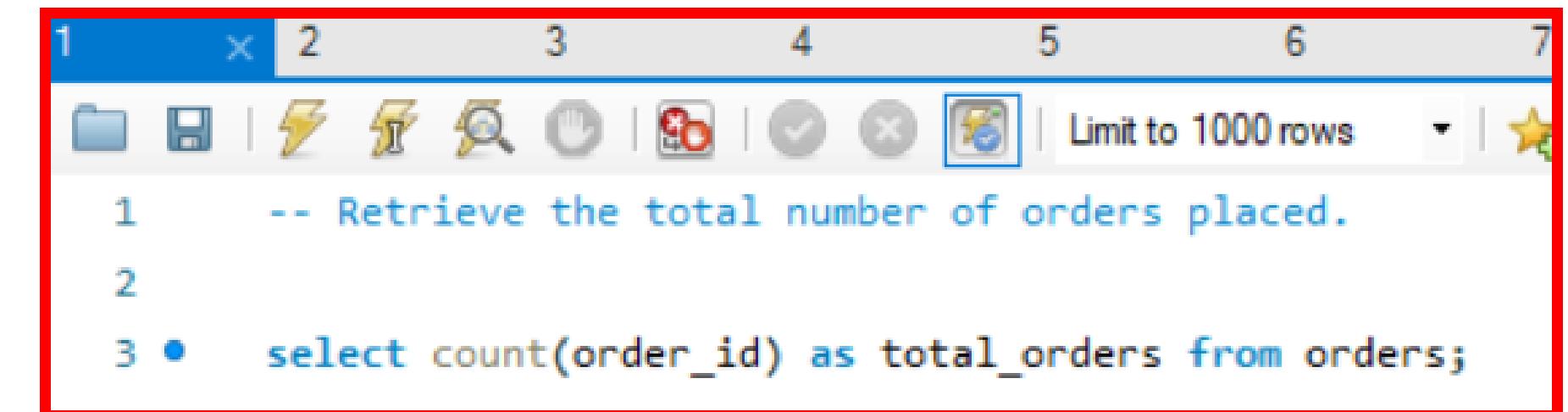
Identify the most common pizza size ordered.

5

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

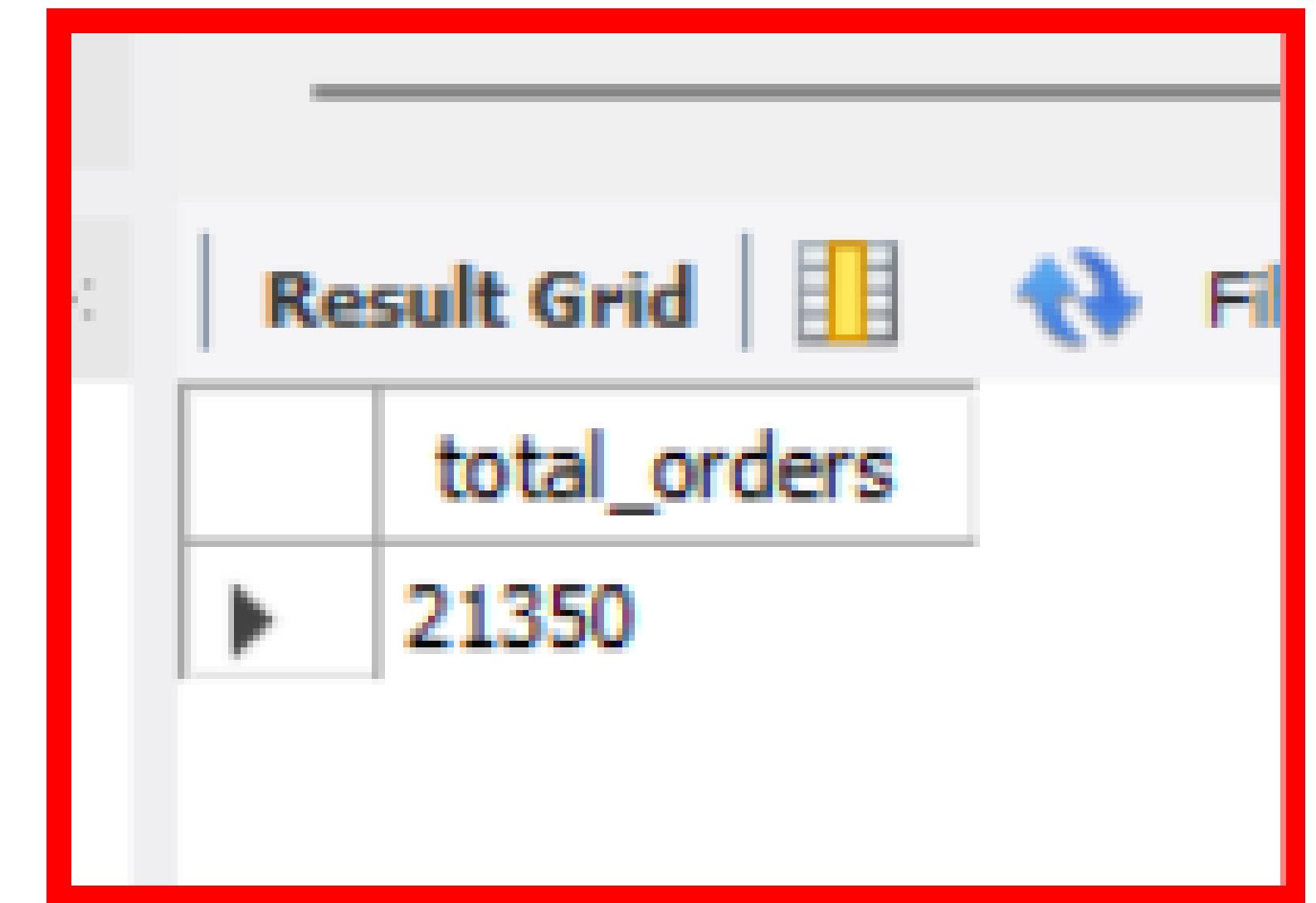
# Question 1

Retrieve the total number of orders placed.



The screenshot shows a MySQL query editor window. The top bar has buttons for file operations and a toolbar with icons for copy, paste, search, and refresh. A dropdown menu says "Limit to 1000 rows". The main area contains the following SQL code:

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



The screenshot shows the results of the executed query. The title "Result Grid" is visible above the table. The table has one column labeled "total\_orders" and one row containing the value "21350".

	total_orders
▶	21350

# Question 2

Calculate the total revenue generated from pizza sales.

```
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
7 FROM
8   order_details
9   JOIN
10  pizzas ON pizzas.pizza_id = order_details.pizza_id
```

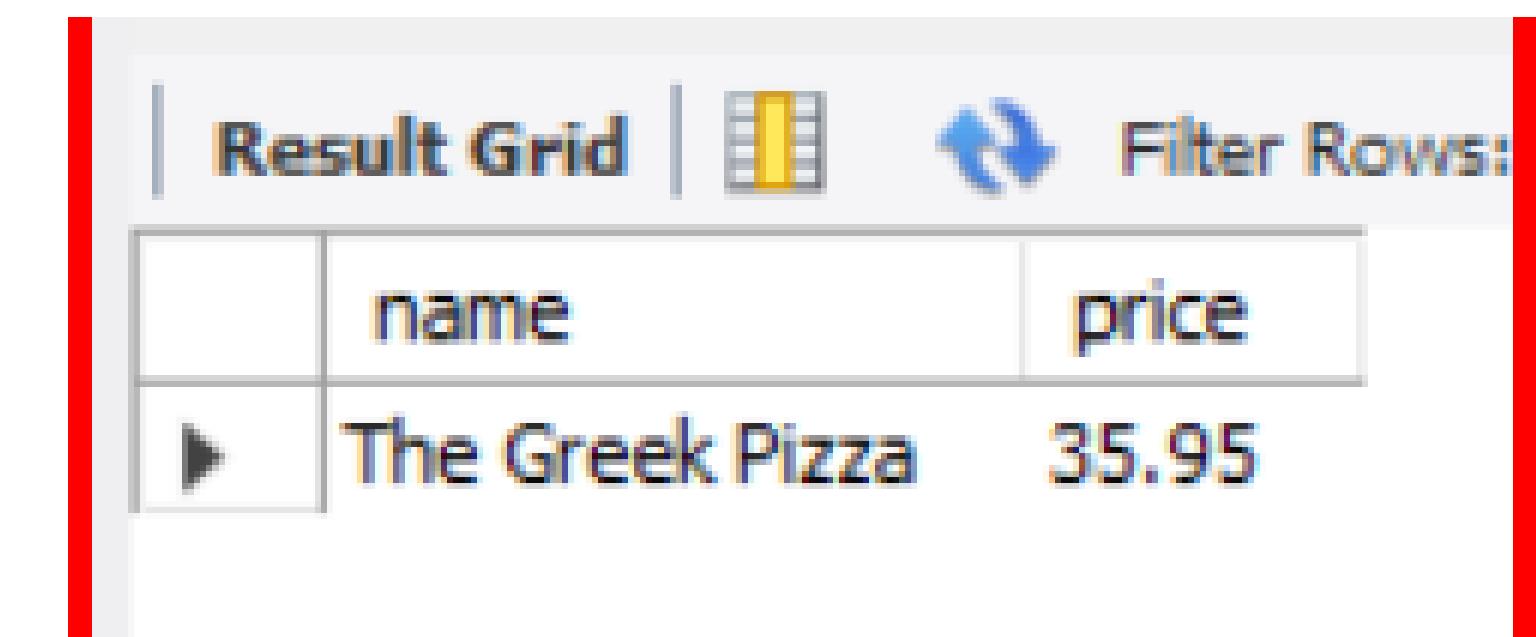
The screenshot shows a database query results interface. At the top, there is a header bar with a search icon, a 'Result Grid' button, and a refresh/circular arrow icon. Below the header, the results are displayed in a grid. The first column is labeled 'total\_sales' and contains a single row with the value '817860.05'. The grid has a light gray background with alternating row colors.

total_sales
817860.05

# Question 3

Identify the highest-priced pizza.

```
1 -- Identify the highest-priced pizza.  
2  
3 • select pizza_types.name, pizzas.price  
4   from pizza_types join pizzas  
5     on pizza_types.pizza_type_id = pizzas.pizza_type_id  
6   order by pizzas.price desc limit 1;
```



The screenshot shows a MySQL Workbench interface with a result grid. The grid has two columns: 'name' and 'price'. A single row is displayed, showing 'The Greek Pizza' with a price of '35.95'. The 'Result Grid' tab is selected at the top. There are also 'Filter Rows' and other navigation buttons.

	name	price
▶	The Greek Pizza	35.95

# Question 4

```
5  
6 •   SELECT  
7     pizzas.size,  
8     COUNT(order_details.order_details_id) AS order_count  
9   FROM  
10    pizzas  
11      JOIN  
12        order_details ON pizzas.pizza_id = order_details.pizza_id  
13    GROUP BY pizzas.size  
14    ORDER BY order_count DESC;
```

Identify the most common pizza size ordered.

Result Grid | Filter

	size	order_count
▶	L	18526
	M	15385
	S	14137
	XL	544
	XXL	28

# Question 5

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

```
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;
```

	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

# Intermediate Questions

Briefly elaborate on what you want to discuss.

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.

# Question 6

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

```
1 -- Join the necessary tables to find the total quantity of each pizza category ordered.  
2  
3 SELECT  
4     pizza_types.category,  
5     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.category  
13 ORDER BY quantity DESC;
```

Result Grid | Filter Rows

	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

# Question 7

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), COUNT(order_id) as order_count  
5  FROM  
6      orders  
7  GROUP BY HOUR(order_time);
```

Result Grid | Filter Rows:

	hour(order_time)	count(order_id)
▶	11	1231
	12	2520
	13	2455
	14	1472
	15	1468
	16	1920
	17	2336

Result 1 ×

# Question 8

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

- `select category, count(name) from pizza_types  
group by category`

Result Grid | Filter Row

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

# Question 9

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

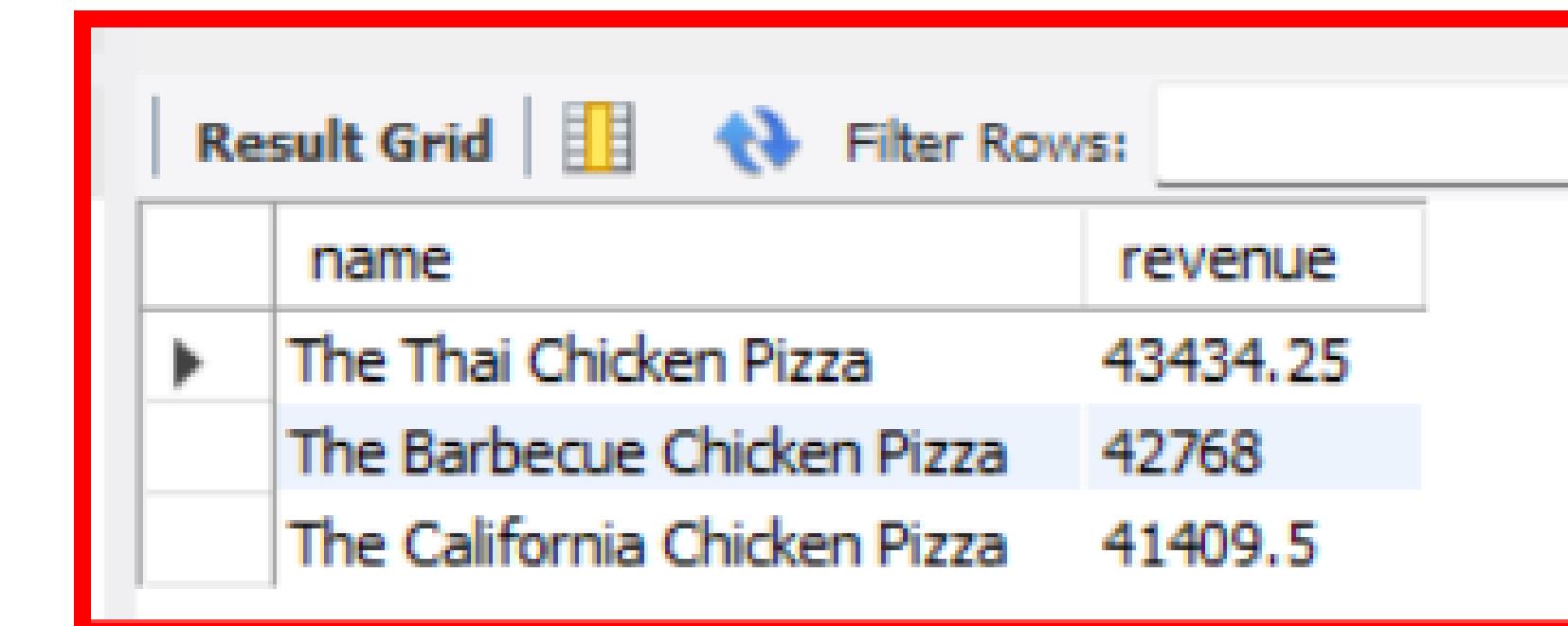
```
SELECT  
    ROUND(AVG(quantity), 0)  
FROM  
    (SELECT  
        orders.order_date, SUM(order_details.quantity) AS quantity  
    FROM  
        orders  
    JOIN order_details ON orders.order_id = order_details.order_id  
    GROUP BY orders.order_date) AS order_quantity;
```

Result Grid	
	ROUND(AVG(quantity), 0)
▶	138

# Question 10

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

```
1  -- Determine the top 3 most ordered pizza types based on revenue.  
2  
3 • select pizza_types.name,  
4   sum(order_details.quantity * pizzas.price) as revenue  
5   from pizza_types join pizzas  
6   on pizzas.pizza_type_id = pizza_types.pizza_type_id  
7   join order_details  
8   on order_details.pizza_id = pizzas.pizza_id  
9   group by pizza_types.name order by revenue desc limit 3;  
.0
```



The screenshot shows a MySQL Workbench interface with a result grid. The grid has two columns: 'name' and 'revenue'. The data is sorted by revenue in descending order. The top three rows are highlighted in light blue. The first row contains 'The Thai Chicken Pizza' with a revenue of 43434.25. The second row contains 'The Barbecue Chicken Pizza' with a revenue of 42768. The third row contains 'The California Chicken Pizza' with a revenue of 41409.5.

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

# Advanced Questions

Briefly elaborate on what you want to discuss.

11

Write an achievement,  
successful project or  
milestone here.

12

Write an achievement,  
successful project or  
milestone here.

13

Write an achievement,  
successful project or  
milestone here.

# Question 11

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

```
3   SELECT
4       pizza_types.category,
5           (SUM(order_details.quantity * pizzas.price) / (SELECT
6               ROUND(SUM(order_details.quantity * pizzas.price),
7                   2) AS total_sales
8           FROM
9               order_details
10              JOIN
11                  pizzas ON pizzas.pizza_id = order_details.pizza_id)) * 100 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 ON order_details.pizza_id = pizzas.pizza_id
18      GROUP BY pizza_types.category
19      ORDER BY revenue DESC;
```

Result Grid | Filter Rows:

	category	revenue
▶	Classic	26.90596025566967
	Supreme	25.45631126009862
	Chicken	23.955137556847287
	Veggie	23.682590927384577

# Question 12

Analyze the cumulative revenue generated over time.

```
•  
select order_date,  
sum(revenue) over(order by order_date) as cum_revenue  
from  
(select orders.order_date,  
sum(order_details.quantity * pizzas.price) as revenue  
from order_details join pizzas  
on order_details.pizza_id = pizzas.pizza_id  
join orders  
on orders.order_id = order_details.order_id  
group by orders.order_date) as sales
```

Result Grid | Filter Rows:

	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

Result 1 ×

# Question 13

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  
3 • select name, revenue from  
4  
5   (select category, name, revenue,  
6    rank() over(partition by category order by revenue desc) as rn  
7    from  
8   (select pizza_types.category, pizza_types.name,  
9    sum((order_details.quantity)* pizzas.price) as revenue  
10   from pizza_types 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
```

Result Grid		Filter Rows:	Exp
	name	revenue	
▶	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	

Thank You.



# Connect with me.



Email

**shrutisonone24@gmail.com**



LinkedIn

**Shruti Sonone**



Mobile

**7498110792**

