



PIZZA SALES ANALYSIS



**50%
OFF**

A large orange circular badge with a dashed white border and a white sunburst graphic above it, containing the text '50% OFF' in bold black letters.

Project by - Prince Kumar Yadav





Retrieve the total number of orders placed.

```
• SELECT  
    COUNT(order_id) AS total_orders  
FROM  
orders;
```

Result Grid		Filter Rows:
	total_orders	
▶	21350	





Calculate the total revenue generated from pizza sales.

```
• SELECT
  ROUND(SUM(order_details.quantity * pizzas.price),
        2) AS Total_Sales
FROM
  order_details
  JOIN
  pizzas ON pizzas.pizza_id = order_details.pizza_id;
```

Result Grid	
	Total_Sales
	817860.05



Identify the highest-priced pizza.

```
• SELECT
    pizza_types.name, pizzas.price
FROM
    pizza_types
    JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
ORDER BY pizzas.price DESC
LIMIT 1;
```

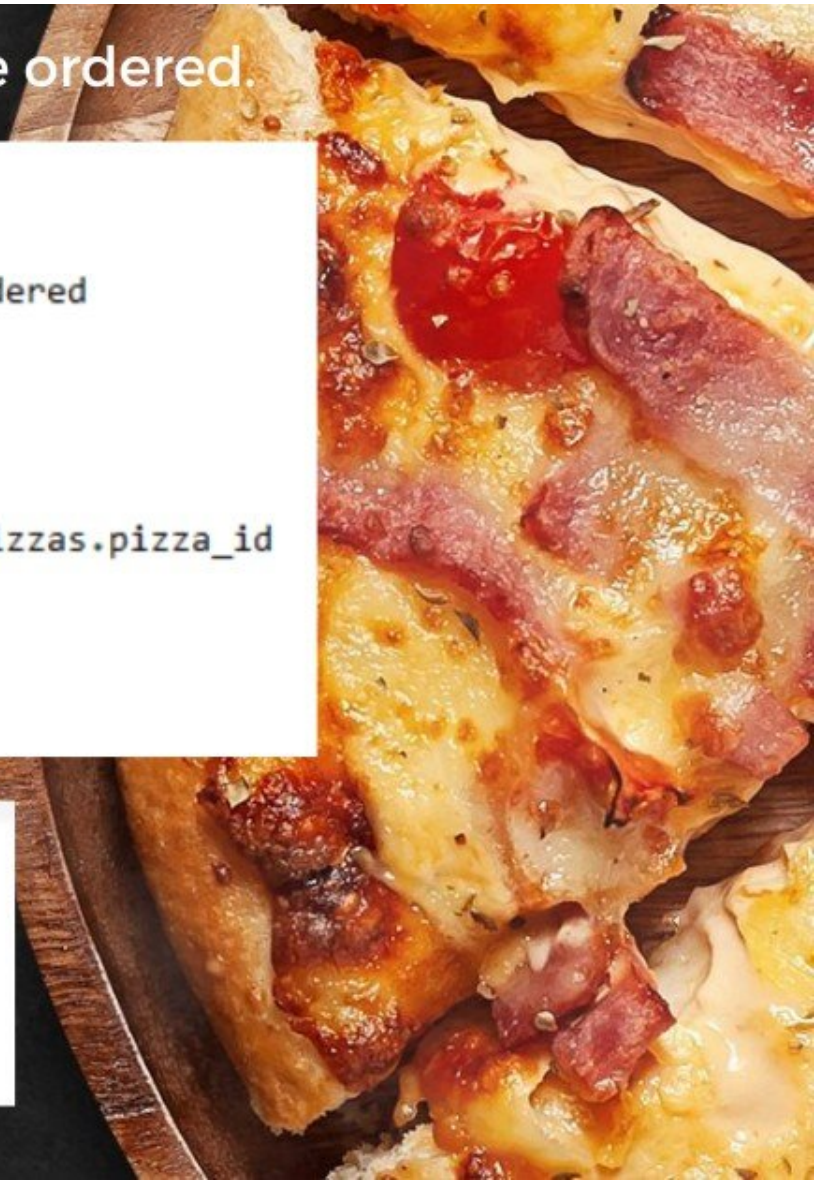
	name	price
▶	The Greek Pizza	35.95



Identify the most common pizza size ordered.

```
• SELECT
    pizzas.size,
    COUNT(order_details.order_details_id) AS ordered
FROM
    pizzas
    JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizzas.size
ORDER BY ordered DESC
LIMIT 1;
```

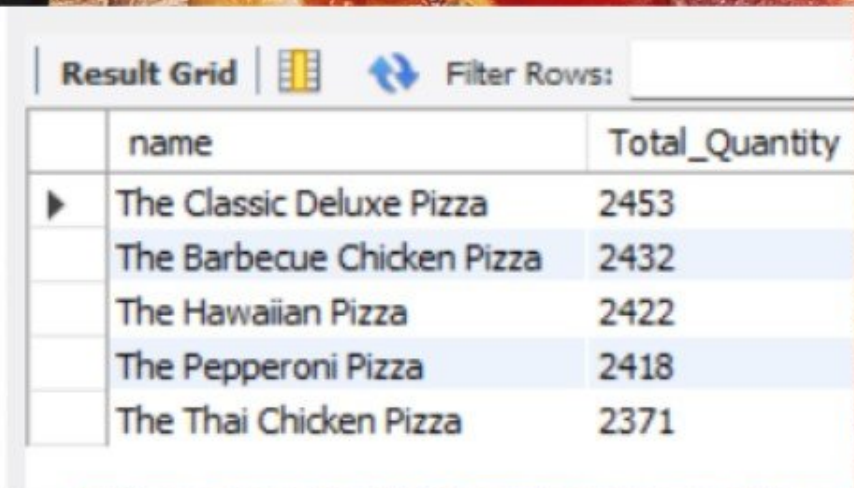
	size	ordered
▶	L	18526





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

```
• SELECT
    pizza_types.name,
    SUM(order_details.quantity) AS Total_Quantity
FROM
    pizza_types
    JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
    JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY total_quantity DESC
LIMIT 5;
```

A screenshot of a database query result interface. It features a 'Result Grid' tab, a grid icon, a refresh icon, and a 'Filter Rows:' input field. Below these controls is a table with two columns: 'name' and 'Total_Quantity'. The table contains five rows of data, with the first row highlighted in blue. The background of the slide shows a close-up of a pizza with toppings like pepperoni and mushrooms.

	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



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

- **SELECT**

```
    pizza_types.category,  
    SUM(order_details.quantity) AS Total_Qty
```

FROM

```
    pizza_types
```

JOIN

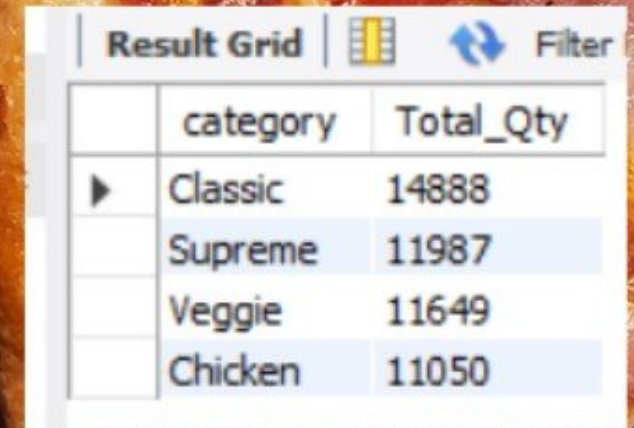
```
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
```

JOIN

```
    order_details ON pizzas.pizza_id = order_details.pizza_id
```

```
GROUP BY pizza_types.category
```

```
ORDER BY Total_Qty DESC;
```

A screenshot of a database query result grid. The grid has a header row with 'category' and 'Total_Qty'. Below the header, there are four rows of data: 'Classic' with '14888', 'Supreme' with '11987', 'Veggie' with '11649', and 'Chicken' with '11050'. The grid is titled 'Result Grid' and has a 'Filter' button.

	category	Total_Qty
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050



Determine the distribution of orders by hour of the day

```
• SELECT
    HOUR(order_time) AS Hour, COUNT(order_id) AS Total_order
FROM
    orders
GROUP BY hour
ORDER BY Total_order DESC;
```

	Hour	Total_order
▶	12	2520
	13	2455
	18	2399
	17	2336
	19	2009
	16	1920
	20	1642
	14	1472
	15	1468
	11	1231
	21	1198
	22	663
	23	28
	10	8
	9	1



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

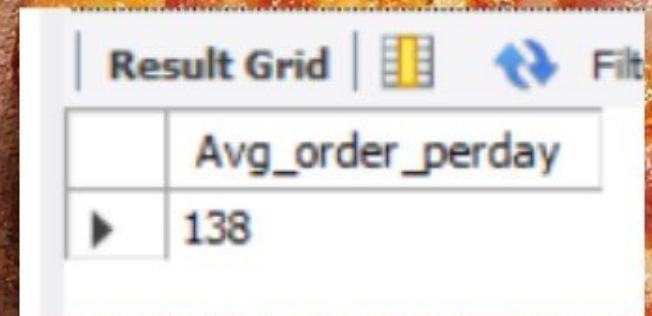
```
• SELECT
    category, COUNT(name)
FROM
    pizza_types
GROUP BY category;
```

	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.

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

A screenshot of a database query result grid. The grid has a header row with the column name 'Avg_order_perday' and a data row with the value '138'. The grid is part of a software interface with buttons for 'Result Grid', a filter icon, and a 'Filter' label.

Result Grid		Filter
	Avg_order_perday	
▶	138	



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

```
• SELECT
    pizza_types.name,
    SUM(pizzas.price * order_details.quantity) AS Revenue
FROM
    pizza_types
    JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
    JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY Revenue DESC
LIMIT 3;
```

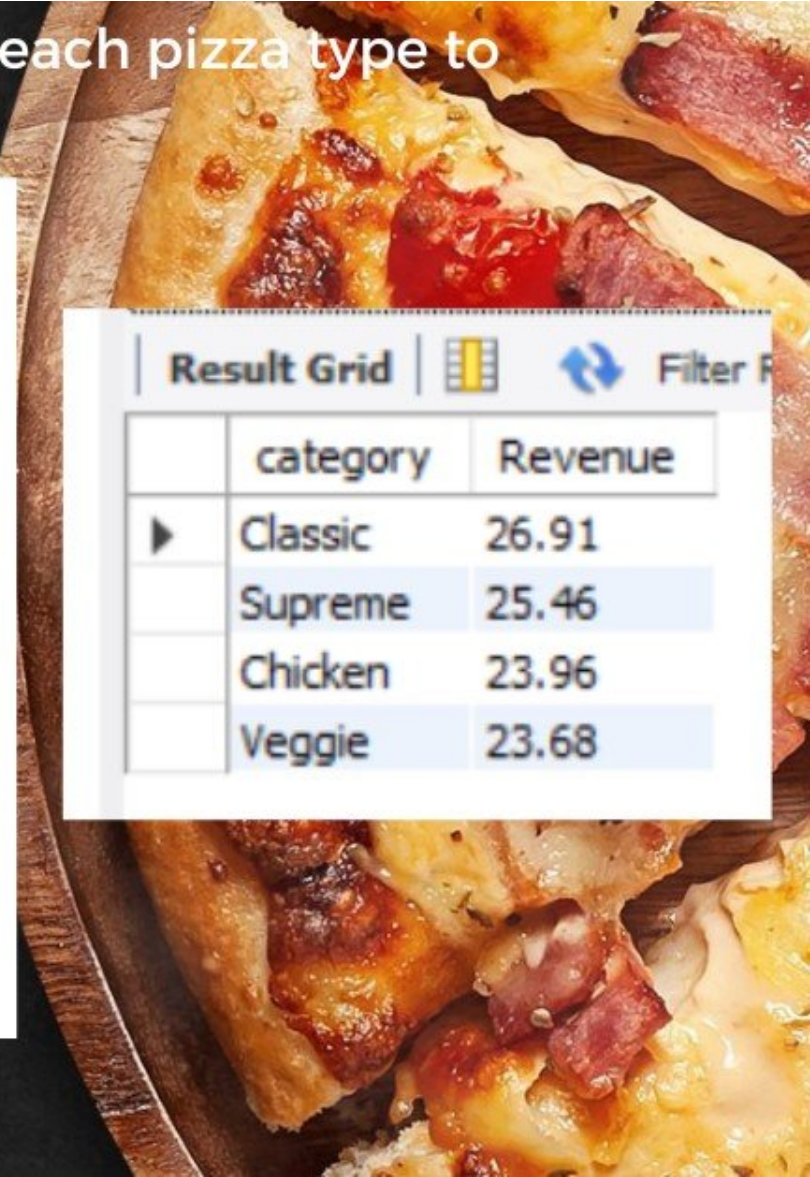
	name	Revenue
•	The Thai Chicken Pizza	43434.25
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41409.5





Calculate the percentage contribution of each pizza type to total revenue

```
• SELECT
    pizza_types.category,
    ROUND((SUM(order_details.quantity * pizzas.price) / (SELECT
        SUM(order_details.quantity * Pizzas.price) AS total_sales
    FROM
        order_details
        JOIN
        pizzas ON pizzas.pizza_id = order_details.pizza_id)),
    4) * 100 AS Revenue
FROM
    pizza_types
    JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
    JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizza_types.category
ORDER BY Revenue DESC;
```



	category	Revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68



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 orders join order_details on orders.order_id = order_details.order_id
join pizzas on pizzas.pizza_id = order_details.pizza_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	
	2015-01-10	23990.350000000002	
	2015-01-11	25862.65	



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

```
• select name , Revenue
  from
  (select category , name, Revenue ,
   Rank() over(partition by category order by revenue desc) as Rn
   from
   (select pizza_types.category , pizza_types.name ,sum(order_details.quantity*pizzas.price ) as Revenue
    from pizza_types join pizzas on pizza_types.pizza_type_id = pizzas.pizza_type_id join order_details
    on pizzas.pizza_id = order_details.pizza_id
    group by pizza_types.category , pizza_types.name) as a ) as b
 where Rn <= 3;
```

	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
	The Italian Supreme Pizza	33476.75