

# **SQL PIZZA SALES DATA ANALYSIS PROJECT**

**BY: POORTI JAIN**



Retrieve the total number of orders placed.

Query

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

Output

Result Grid | Filter Rows

	total_orders
▶	21350

# Calculate the total revenue generated from pizza sales.

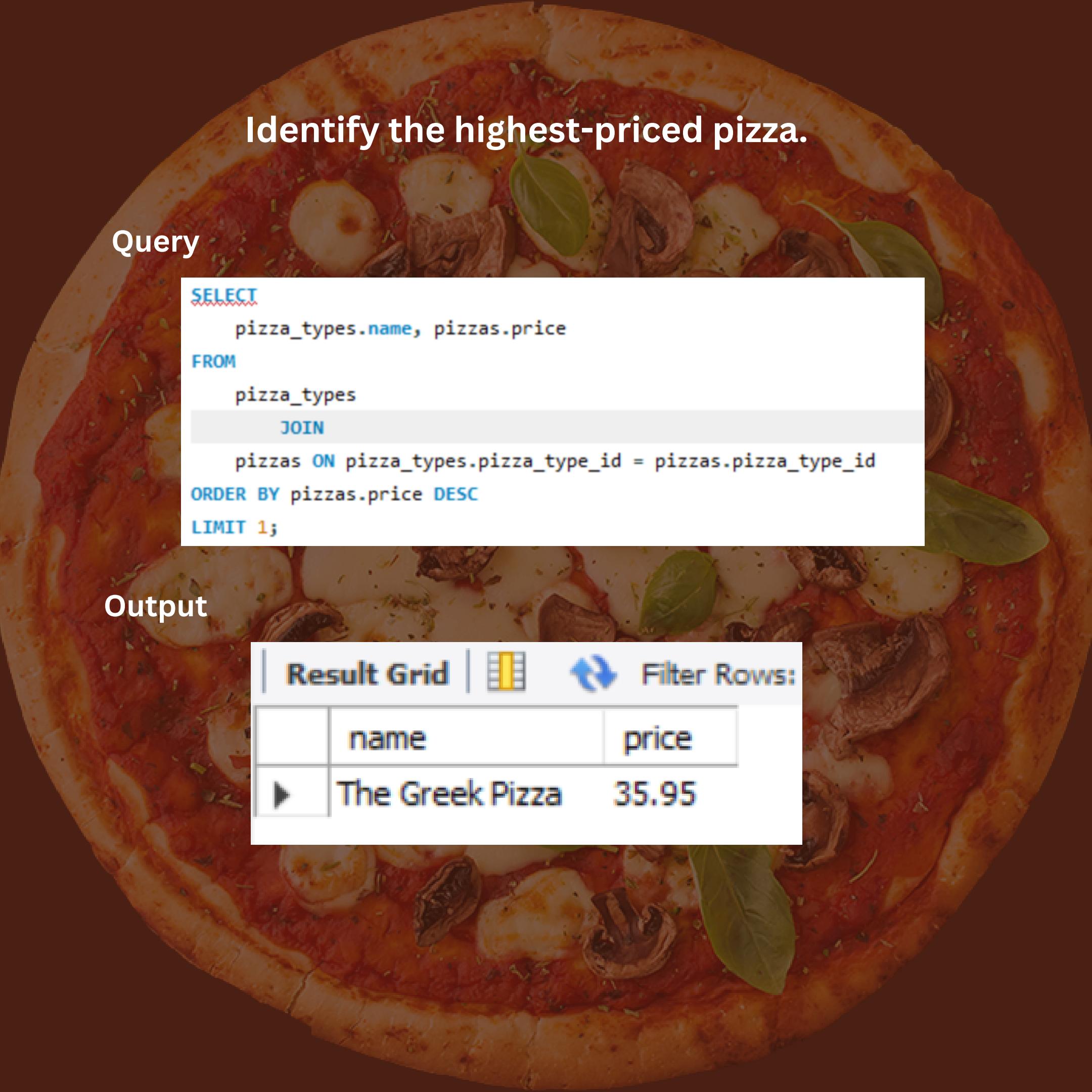
## Query

```
SELECT  
    ROUND(SUM(p.price * q.quantity), 3) AS total_revenue  
FROM  
    pizzas AS p  
    JOIN  
    orders_details AS q ON p.pizza_id = q.pizza_id
```

## Output

Result Grid

▶	total_revenue
	655044.95



# Identify the highest-priced pizza.

## Query

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

## Output

Result Grid |  Filter Rows: 

	name	price
▶	The Greek Pizza	35.95

# Identify the most common pizza size ordered.

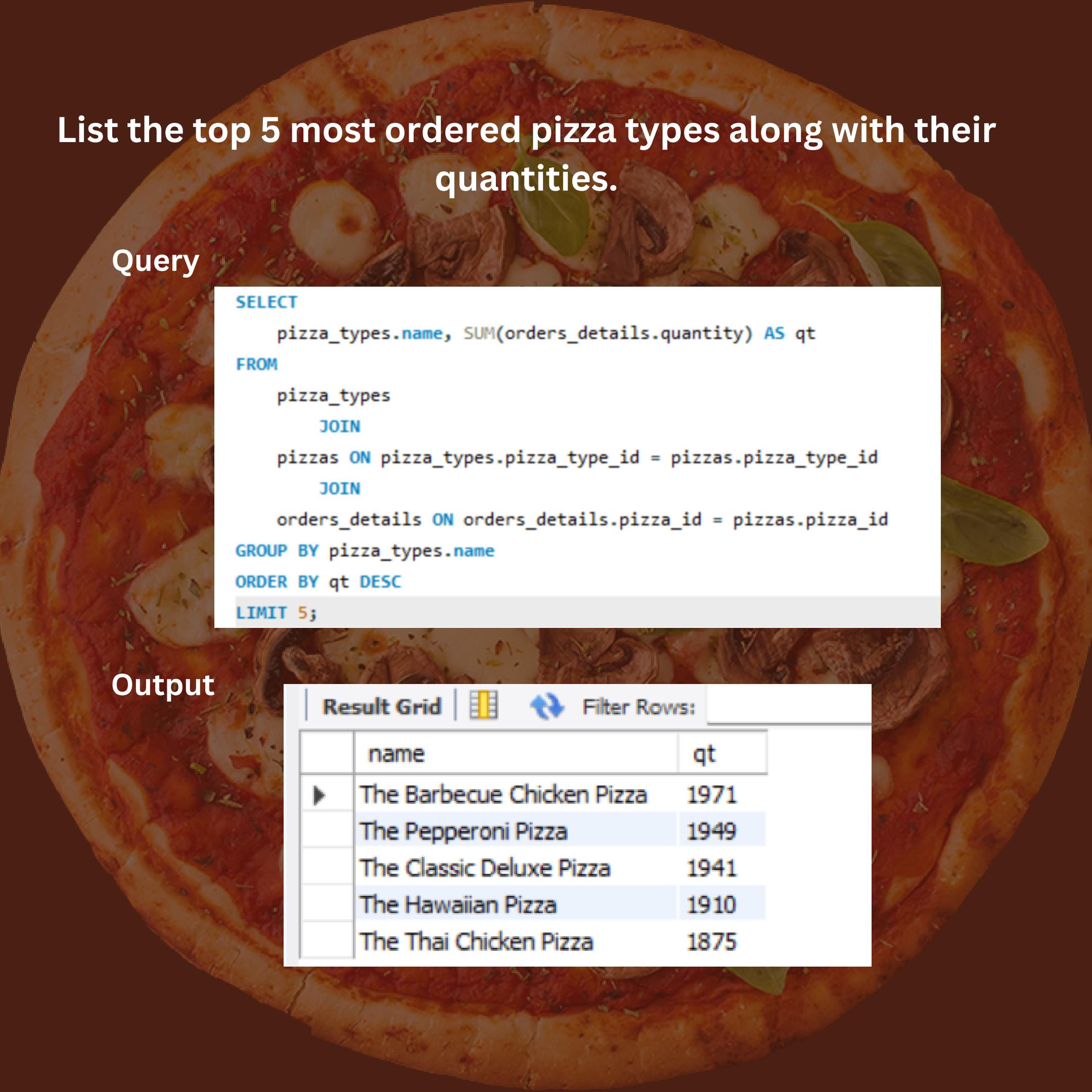
## Query

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

## Output

Result Grid | Filter Rows:

	size	order_count
▶	L	14853



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

## Query

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

## Output

Result Grid | Filter Rows:

	name	qt
▶	The Barbecue Chicken Pizza	1971
	The Pepperoni Pizza	1949
	The Classic Deluxe Pizza	1941
	The Hawaiian Pizza	1910
	The Thai Chicken Pizza	1875

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

## Query

```
SELECT
    pizza_types.category, SUM(orders_details.quantity) AS qt
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category;
```

## Output

	category	qt
▶	Classic	11921
	Veggie	9374

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

Query

```
SELECT
*
FROM
    orders;
SELECT
    HOUR(order_time), COUNT(order_id)
FROM
    orders
GROUP BY HOUR(order_time);
```

Output

	HOUR(order_time)	COUNT(order_id)
▶	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

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

Query

```
SELECT  
    *  
FROM  
    pizza_types;
```

Output

Result Grid | Filter Rows:

	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.

## Query

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

## Output

	Result Grid			Filter Rows:	<input type="text"/>
	avg_pizza_ordered_per_day				
▶					139

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

## Query

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

## Output

	name	revenue
▶	The Barbecue Chicken Pizza	34658.25
	The Thai Chicken Pizza	34286.25
	The California Chicken Pizza	32725.5

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

## Query

```
SELECT
    pizza_types.category,
    ROUND(SUM(orders_details.quantity * pizzas.price),
        2) / (SELECT
            ROUND(SUM(orders_details.quantity * pizzas.price),
                2) AS total_sales
        FROM
            orders_details
            JOIN
            pizzas ON pizzas.pizza_id = orders_details.pizza_id) * 100 AS revenue
FROM
    pizza_types
    JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
    JOIN
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY revenue DESC;
```

## Output

	category	revenue
▶	Classic	26.92049606672031
	Supreme	25.410057737259102
	Chicken	23.869087151958045
	Veggie	23.80035904406255

# Analyze the cumulative revenue generated over time.

## Query

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

## Output

Result Grid | Filter Rows:

	order_date	cum_revenue
	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	27731.7

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

## Query

```
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((orders_details.quantity)*pizzas.price) as revenue
from pizza_types join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
join orders_details
on orders_details.pizza_id = pizzas.pizza_id
group by pizza_types.category, pizza_types.name)as a)
as b
where rn<=3;
```

## Output

	name	revenue
▶	The Barbecue Chicken Pizza	34658.25
	The Thai Chicken Pizza	34286.25
	The California Chicken Pizza	32725.5
	The Classic Deluxe Pizza	30168
	The Hawaiian Pizza	25513.5
	The Pepperoni Pizza	24240.5