

# PIZZA SALES

Quarterly

All

Pizza Category

All

Pizza Size

All

Total Revenue

\$817.86K

Total Orders

21.35K

Total Pizzas Sold

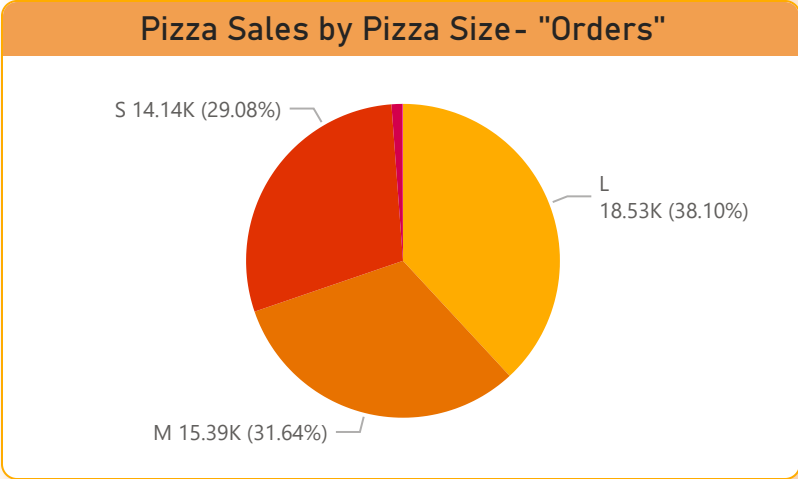
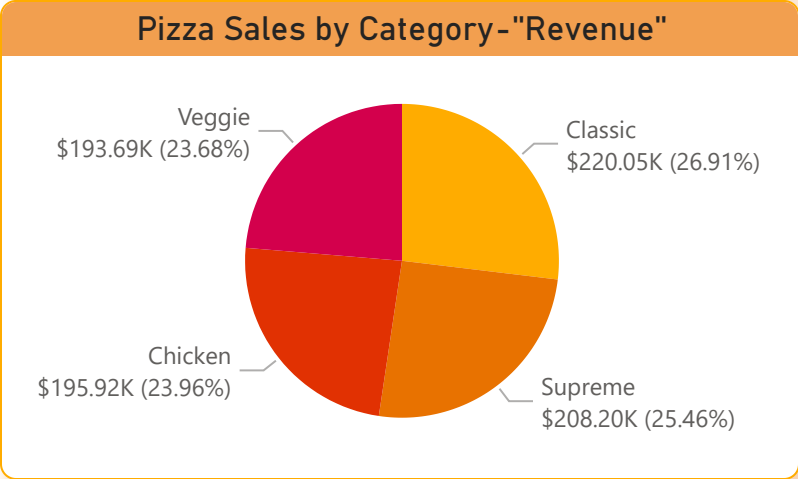
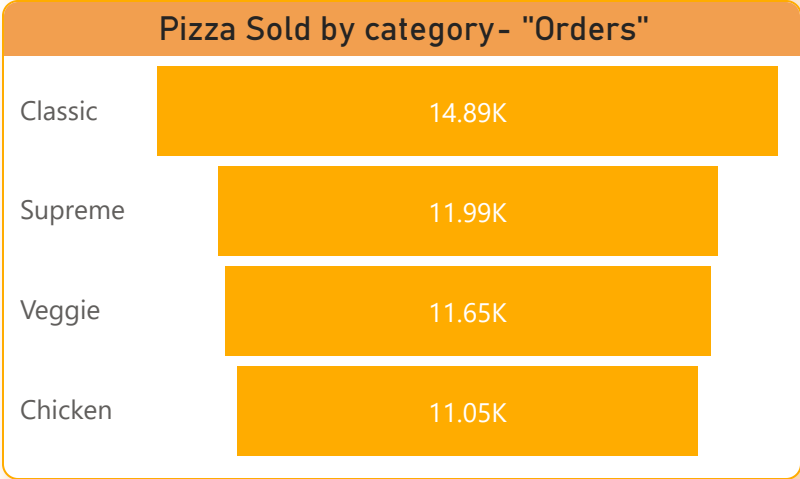
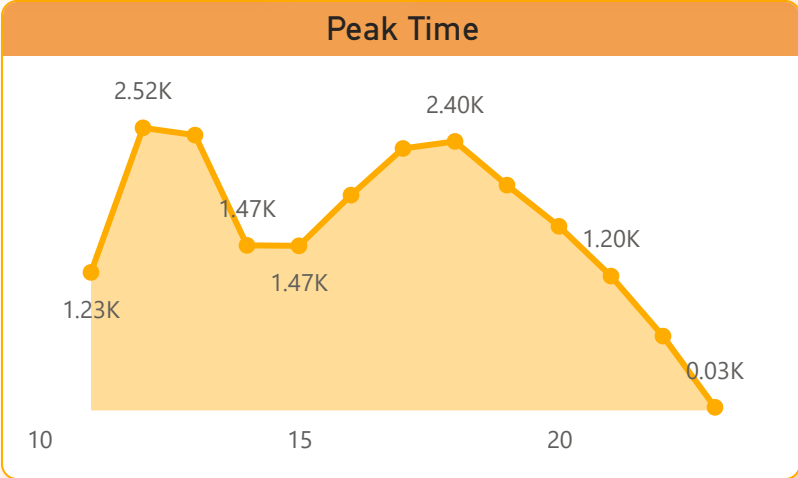
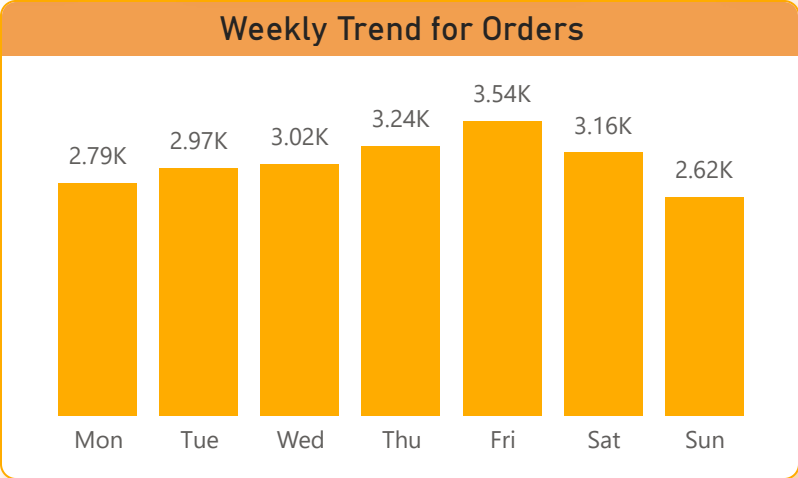
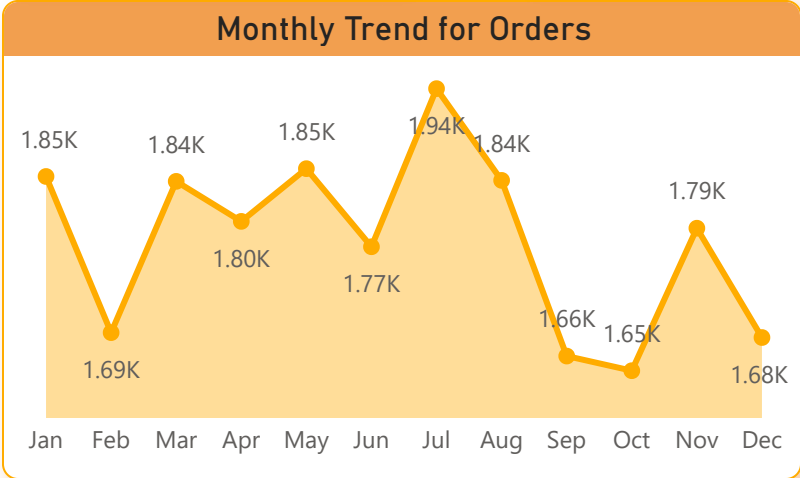
49.6K

Expensive Pizza

The Greek Pizza

Cheap Pizza

The Pepperoni Pizza



# PIZZA SALES

Quarterly

All

Pizza Category

All

Pizza Size

All

Total Revenue

\$817.86K

Total Orders

21.35K

Total Pizzas Sold

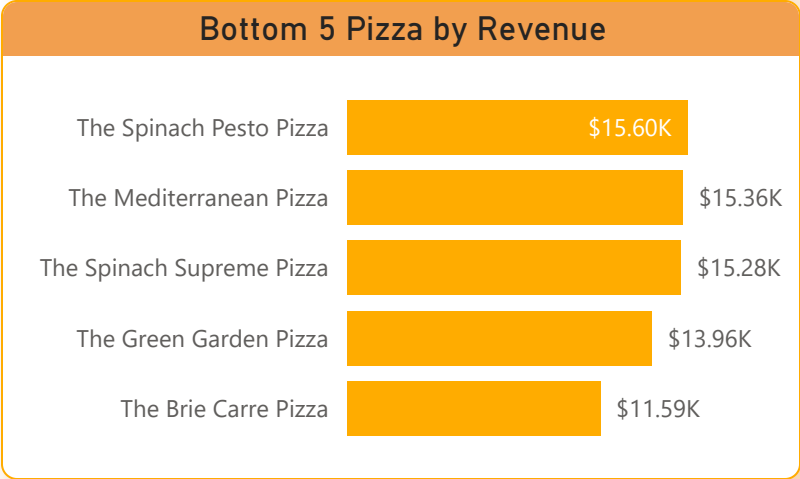
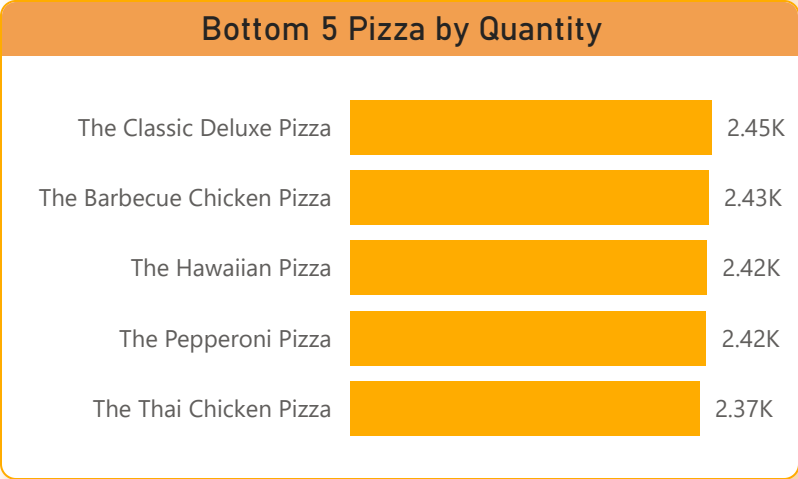
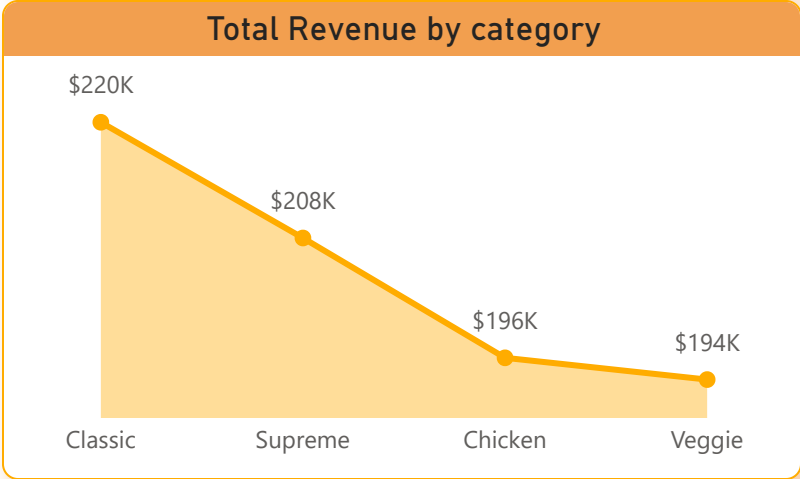
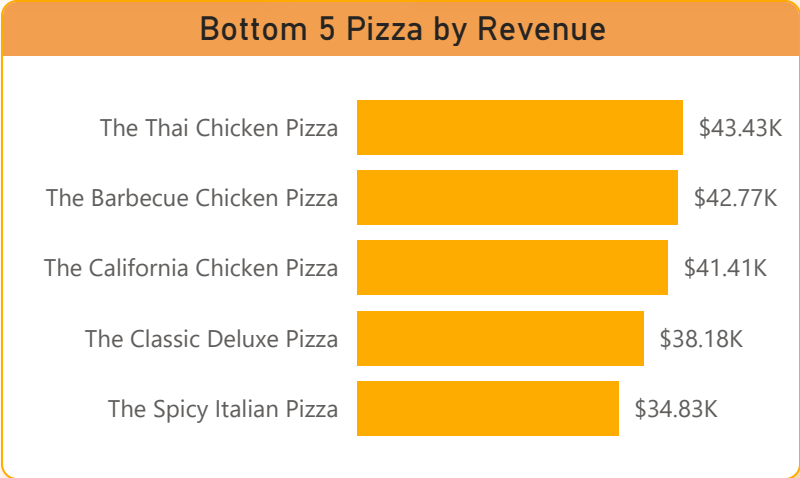
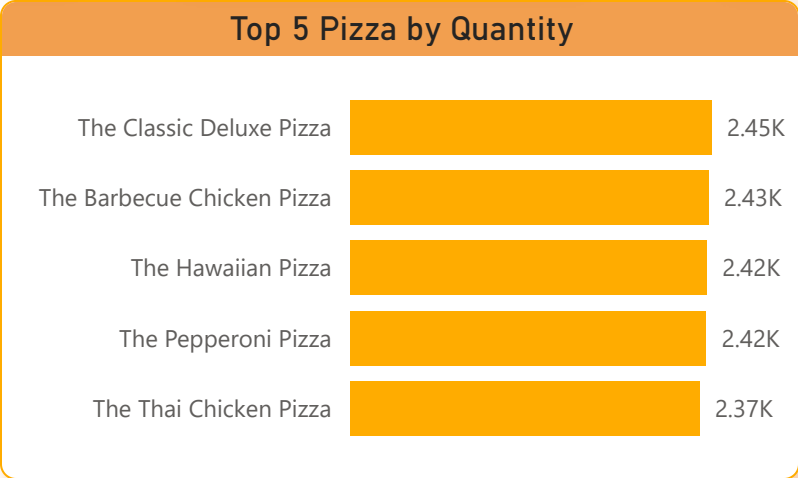
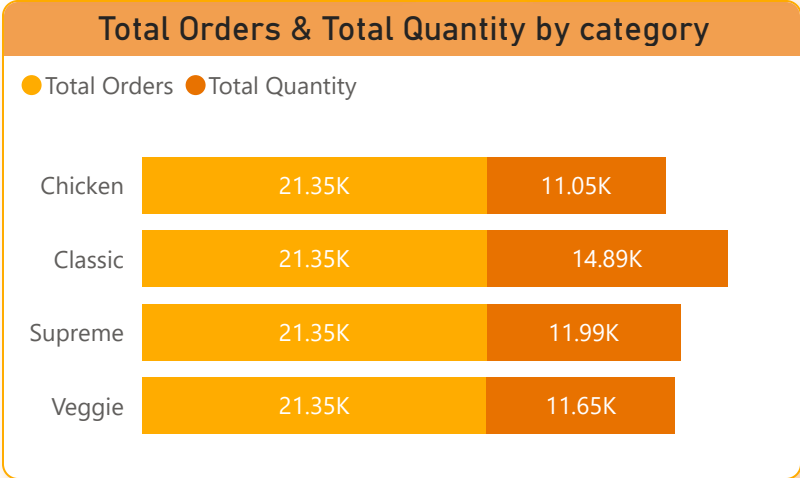
49.6K

Expensive Pizza

The Greek Pizza

Cheap Pizza

The Pepperoni Pizza



<div><div>Report</div><div>PIZZA SALES</div></div>				Quarterly <div>All</div>		Pizza Category <div>All</div>		Pizza Size <div>All</div>	
Total Revenue <div>\$64.18K</div>		Total Orders <div>1.66K</div>		Total Pizzas Sold <div>3.9K</div>		Expensive Pizza <div>The Greek Pizza</div>		Cheap Pizza <div>The Pepperoni Pizza</div>	
Year	Month	Name	Category	Name	Size	Total Pizza Sold		Total Revenue	
2015	Sep		Chicken	The Barbecue Chicken Pizza	L	82		\$1,701.50	
2015	Sep		Chicken	The Barbecue Chicken Pizza	M	69		\$1,155.75	
2015	Sep		Chicken	The Barbecue Chicken Pizza	S	41		\$522.75	
2015	Sep		Classic	The Big Meat Pizza	S	142		\$1,704.00	
2015	Sep		Supreme	The Brie Carre Pizza	S	49		\$1,158.85	
2015	Sep		Supreme	The Calabrese Pizza	L	15		\$303.75	
2015	Sep		Supreme	The Calabrese Pizza	M	44		\$715.00	
2015	Sep		Supreme	The Calabrese Pizza	S	7		\$85.75	
Total						3890		\$64,180.05	

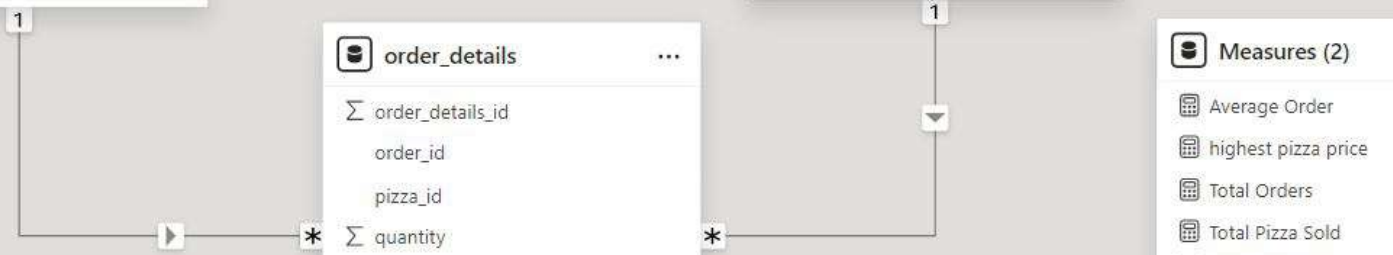
orders	...
date	
Day Name	
Σ Hour	
Collapse ^	

order_details	...
Σ order_details_id	
order_id	
pizza_id	
Σ quantity	
Collapse ^	

pizzas	...
pizza_id	
pizza_type_id	
Σ price	
size	
Collapse ^	

pizza_types	...
category	
ingredients	
name	
pizza_type_id	
Collapse ^	

Measures (2)	...
Average Order	
highest pizza price	
Total Orders	
Total Pizza Sold	
Total Quantity	
Total Revenue	
Collapse ^	



# SQL PROJECT- PIZZA SALES DATA ANALYSIS

1. Retrieve the total number of orders placed.

select count(order\_id) as total\_orders from orders

```
1 • select count(order_id) as total_orders from orders
```

total_orders
21350

2. Calculate the total revenue generated from pizza sales.

select round(sum(price\*quantity),2) as total\_revenue from pizzas

join order\_details

on pizzas.pizza\_id = order\_details.pizza\_id

```
1 select round(sum(price*quantity),2) as total_revenue from pizzas
2 join order_details
3 on pizzas.pizza_id = order_details.pizza_id
4
```

total_revenue
817860.05

### 3. Identify the highest-priced pizza.

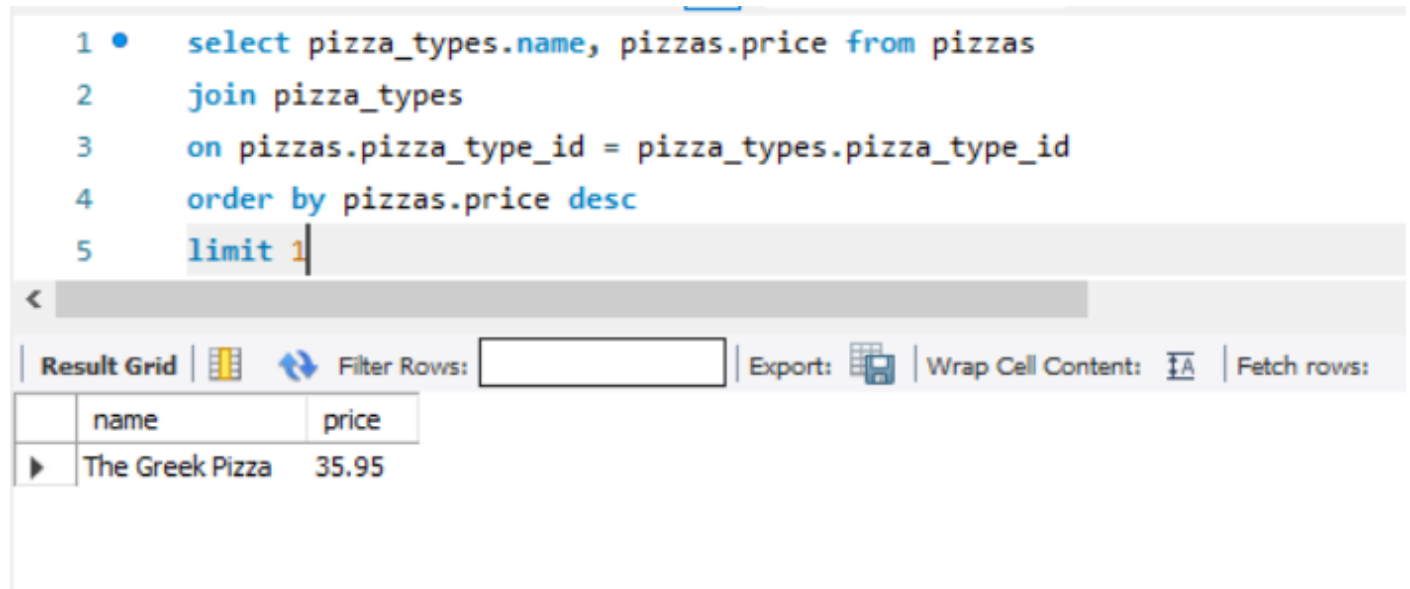
```
select pizza_types.name, pizzas.price from pizzas
```

```
join pizza_types
```

```
on pizzas.pizza_type_id = pizza_types.pizza_type_id
```

```
order by pizzas.price desc
```

```
limit 1
```



The screenshot shows a SQL query editor with a query that selects the highest-priced pizza. The query is as follows:

```
1 • select pizza_types.name, pizzas.price from pizzas
2   join pizza_types
3   on pizzas.pizza_type_id = pizza_types.pizza_type_id
4   order by pizzas.price desc
5   limit 1
```

Below the query editor, there is a toolbar with options: "Result Grid" (selected), "Filter Rows" (with a search box), "Export" (with a download icon), "Wrap Cell Content" (with a text icon), and "Fetch rows".

The results are displayed in a table with two columns: "name" and "price". The first row shows "The Greek Pizza" with a price of "35.95".

	name	price
▶	The Greek Pizza	35.95

#### 4. Identify the most common pizza size ordered.

select pizzas.size, count(order\_details.order\_details\_id) as order\_count

from pizzas

join order\_details



on pizzas.pizza\_id = order\_details.pizza\_id

group by pizzas.size

order by order\_count desc

.

```
1 • select pizzas.size, count(order_details.order_details_id) as order_count
2   from pizzas
3   join order_details
4   on pizzas.pizza_id = order_details.pizza_id
5   group by pizzas.size
6   order by order_count desc
```

<		
Result Grid		
Filter Rows: <input type="text"/>		
Export:  Wrap Cell Content: 		
	size	order_count
▶	L	18526
	M	15385
	S	14137
	XL	544
	XXL	28

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

```
select pizza_types.name, sum(order_details.quantity) as order_quantity
```

```
from pizzas
```

```
join pizza_types
```

```
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.name
```

```
order by order_quantity desc
```

```
limit 5
```

```
1 • select pizza_types.name, sum(order_details.quantity) as order_quantity
2   from pizzas
3  join pizza_types
4  on pizzas.pizza_type_id = pizza_types.pizza_type_id
5  join order_details
6  on pizzas.pizza_id = order_details.pizza_id
7  group by pizza_types.name
8  order by order_quantity desc
9  limit 5
```

<		
Result Grid		
Filter Rows: <input type="text"/>		
Export: <input type="button" value="Export"/>		
Wrap Cell Content: <input type="button" value="Wrap"/>		
Fetch rows: <input type="button" value="Fetch"/>		
	name	order_quantity
▶	The Classic Deluxe Pizza	2453
	The Barbecue Chicken Pizza	2432
	The Hawaiian Pizza	2422
	The Pepperoni Pizza	2418
	The Thai Chicken Pizza	2371



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

```
select pizza_types.category, sum(order_details.quantity) as order_quantity
```

```
from pizzas
```

```
join pizza_types
```

```
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 order_quantity desc
```

```
1 • select pizza_types.category, sum(order_details.quantity) as order_quantity
2   from pizzas
3   join pizza_types
4   on pizzas.pizza_type_id = pizza_types.pizza_type_id
5   join order_details
6   on pizzas.pizza_id = order_details.pizza_id
7   group by pizza_types.category
8   order by order_quantity desc
```

<		
Result Grid		
Filter Rows: <input type="text"/>		
Export:  Wrap Cell Content: 		
	category	order_quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

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

`select hour(order_time), count(order_id) 'No._of_order' from orders`

`group by hour(order_time)`

```
1 • select hour(order_time), count(order_id) 'No._of_order' from orders
2   group by hour(order_time)
3
```

<		
Result Grid		
Filter Rows:		
Export:		
Wrap Cell Content:		
	hour(order_time)	No._of_order
▶	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



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

```
select category, count(name) as count from pizza_types
```

```
group by category
```

```
order by count desc
```

```
1 • select category, count(name) as count from pizza_types
2   group by category
3   order by count desc
```

<		
Result Grid		
Filter Rows: <input type="text"/>		
Export:  Wrap Cell Content: 		
	category	count
▶	Supreme	9
	Veggie	9
	Classic	8
	Chicken	6

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

```
select Round(avg(total_order_quantity),0) from  
  
(select orders.order_date, sum(order_details.quantity) as total_order_quantity  
  
from orders  
  
join order_details  
  
on orders.order_id = order_details.order_id  
  
group by orders.order_date)  
  
as order_per_day  
  
.
```

```
1 • select Round(avg(total_order_quantity),0) from  
2 (select orders.order_date, sum(order_details.quantity) as total_order_quantity  
3 from orders  
4 join order_details  
5 on orders.order_id = order_details.order_id  
6 group by orders.order_date)  
7 as order_per_day  
8
```

<	
Result Grid	Filter Rows: <input type="text"/> Export:  Wrap Cell Content:
	Round(avg(total_order_quantity),0)
▶	138

## 10. Determine the top 5 most ordered pizza types based on revenue.

```
select pizza_types.name, sum(order_details.quantity*pizzas.price) as revenue
```

```
from pizzas
```

```
join pizza_types
```

```
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.name
```

```
order by revenue desc
```


```
limit 5
```

```
1 • select pizza_types.name, sum(order_details.quantity*pizzas.price) as revenue
2   from pizzas
3  join pizza_types
4  on pizzas.pizza_type_id = pizza_types.pizza_type_id
5  join order_details
6  on pizzas.pizza_id = order_details.pizza_id
7  group by pizza_types.name
8  order by revenue desc
9  limit 5
```

Result Grid			Filter Rows:	Export:	Wrap Cell Content:	Fetch rows:
	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 Spicy Italian Pizza	34831.25				

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

```
select pizza_types.category,  
round(sum(order_details.quantity*pizzas.price) / (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) * 100,2) as revenue  
from pizzas  
join pizza_types  
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 revenue desc
```



The screenshot shows a SQL query editor with the following query:

```
1 • select pizza_types.category,  
2 round(sum(order_details.quantity*pizzas.price) / (select round(sum(order_details.quantity*pizzas.price),2) as total_sales  
3 from order_details  
4 join pizzas  
5 on pizzas.pizza_id = order_details.pizza_id) * 100,2) as revenue  
6 from pizzas  
7 join pizza_types  
8 on pizzas.pizza_type_id = pizza_types.pizza_type_id  
9 join order_details  
10 on pizzas.pizza_id = order_details.pizza_id  
11 group by pizza_types.category  
12 order by revenue desc
```

Below the query editor, the results are displayed in a table with the following data:

category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

## 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 pizzas.pizza_id = order_details.pizza_id
```

```
join orders
```

```
on orders.order_id = order_details.order_id
```

```
group by orders.order_date) as sales
```

```
1 • select order_date, sum(revenue) over(order by order_date) as cum_revenue
2   from
3   (select orders.order_date, sum(order_details.quantity * pizzas.price) as revenue
4     from order_details
5     join pizzas
6     on pizzas.pizza_id = order_details.pizza_id
7     join orders
8     on orders.order_id = order_details.order_id
9     group by orders.order_date) as sales
```

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
2015-01-13	29831.300000000003
2015-01-14	32358.700000000004
2015-01-15	34343.50000000001
2015-01-16	36937.65000000001
2015-01-17	39001.75000000001
2015-01-18	40978.600000000006
2015-01-19	43365.75000000001
2015-01-20	45763.65000000001

Result 20

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

```
select name, revenue, ranking_by_category from
(select name, category, revenue,
rank() over(partition by category order by revenue desc) as ranking_by_category
from
(
select pizza_types.name, pizza_types.category, 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 order_details.pizza_id = pizzas.pizza_id
group by pizza_types.category, pizza_types.name
) as order_table ) as b
where ranking_by_category <=3
```

```
1 • select name, revenue, ranking_by_category from
2 (select name, category, revenue,
3 rank() over(partition by category order by revenue desc) as ranking_by_category
4 from
5 (
6 select pizza_types.name, pizza_types.category, sum((order_details.quantity) * pizzas.price) as revenue
7 from pizza_types
8 join pizzas
9 on pizza_types.pizza_type_id = pizzas.pizza_type_id
10 join order_details
11 on order_details.pizza_id = pizzas.pizza_id
12 group by pizza_types.category, pizza_types.name
13 ) as order_table ) as b
14 where ranking_by_category <=3
```

name	revenue	ranking_by_category
The Thai Chicken Pizza	43434.25	1
The Barbecue Chicken Pizza	42768	2
The California Chicken Pizza	41409.5	3
The Classic Deluxe Pizza	38180.5	1
The Hawaiian Pizza	32273.25	2
The Pepperoni Pizza	30161.75	3
The Spicy Italian Pizza	34831.25	1
The Italian Supreme Pizza	33476.75	2
The Sicilian Pizza	30940.5	3
The Four Cheese Pizza	32265.70000000065	1
The Mexicana Pizza	26780.75	2
The Five Cheese Pizza	26066.5	3