

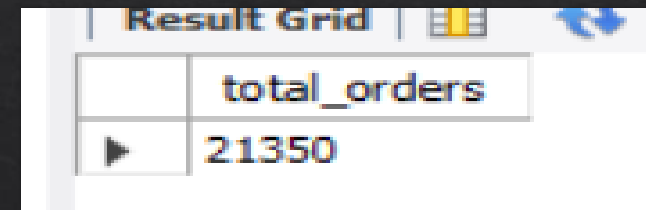
# SQL PROJECT FOR PIZZA SALES ANALYSIS

# The total number of orders placed.

CODE

```
select count(order_id) as total_orders from orders;
```

OUTPUT



The screenshot shows a 'Result Grid' window with a table containing one row. The column is labeled 'total\_orders' and the value is '21350'. There are icons for grid, table, and refresh at the top right.

Result Grid	
	total_orders
▶	21350

# The total revenue generated from pizza sales.

CODE

```
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
```

OUTPUT

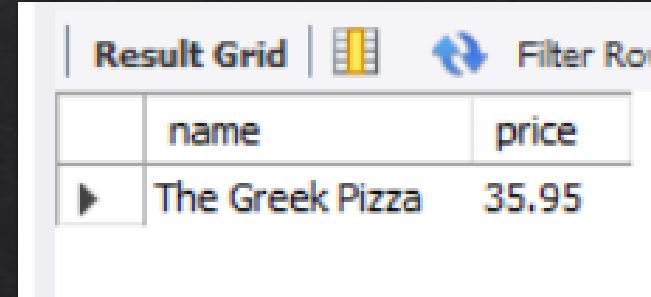
Result Grid	
	total_sales
▶	817860.05

# The highest-priced pizza.

## CODE

```
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



The screenshot shows a database interface with a 'Result Grid' tab. The grid contains two columns: 'name' and 'price'. The first row of data shows 'The Greek Pizza' with a price of '35.95'. Above the grid, there are icons for a grid view, a refresh button, and a 'Filter Rows' label.

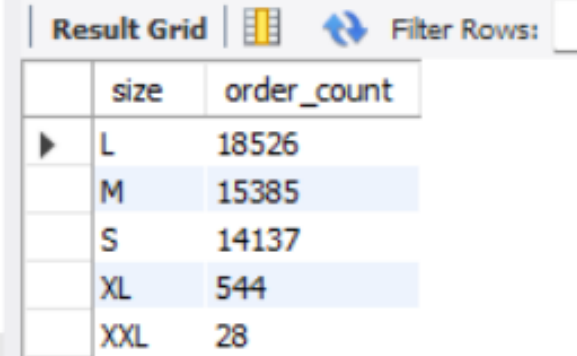
	name	price
▶	The Greek Pizza	35.95

# The most common pizza size ordered.

## CODE

```
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 ;
```

## OUTPUT



The screenshot shows a database interface with a 'Result Grid' tab. It displays a table with two columns: 'size' and 'order\_count'. The data is sorted in descending order of 'order\_count'. The rows are: L (18526), M (15385), S (14137), XL (544), and XXL (28). There are also icons for a grid view, a refresh button, and a 'Filter Rows' input field.

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




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

## CODE

```
select pizza_types.name, sum(orders_details.quantity) as quantity
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 quantity desc
limit 5;
```

## OUTPUT



The screenshot shows a database interface with a 'Result Grid' tab selected. The grid displays the results of the SQL query, showing the top 5 most ordered pizza types by quantity. The columns are 'name' and 'quantity'. The rows are: 'The Classic Deluxe Pizza' (2453), 'The Barbecue Chicken Pizza' (2432), 'The Hawaiian Pizza' (2422), 'The Pepperoni Pizza' (2418), and 'The Thai Chicken Pizza' (2371). The first row is highlighted with a mouse cursor.

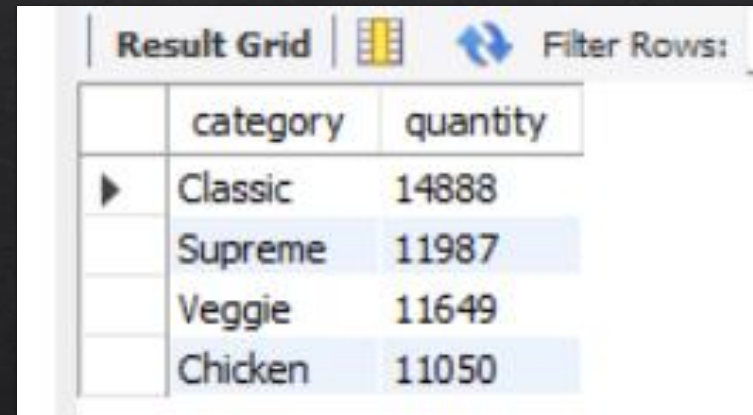
	name	quantity
▶	The Classic Deluxe Pizza	2453
	The Barbecue Chicken Pizza	2432
	The Hawaiian Pizza	2422
	The Pepperoni Pizza	2418
	The Thai Chicken Pizza	2371

the total quantity of each pizza category ordered.

## CODE

```
select pizza_types.category, sum(orders_details.quantity) as quantity
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 quantity desc;
```

## OUTPUT



The screenshot shows a database interface with a 'Result Grid' tab. It displays a table with two columns: 'category' and 'quantity'. The data is sorted in descending order of quantity. The categories listed are Classic, Supreme, Veggie, and Chicken.

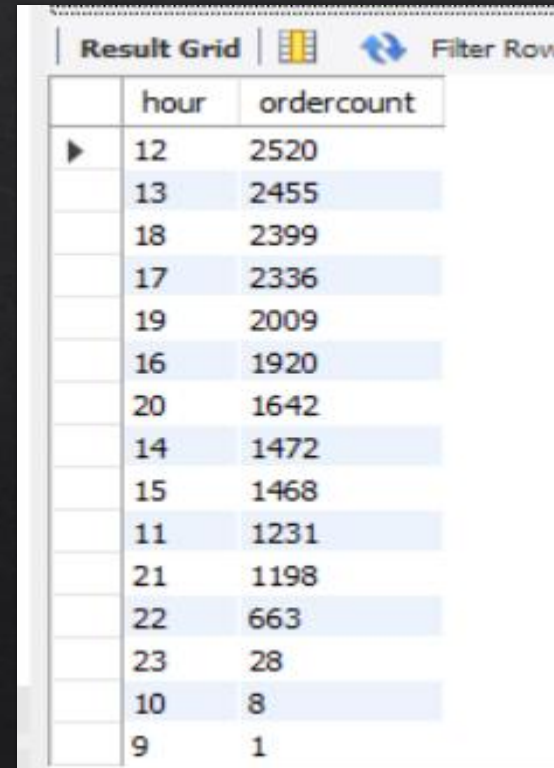
	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

# The distribution of orders by hour of the day.

## CODE

```
select hour(order_time) as hour , count(order_id) as ordercount
from orders
group by hour(order_time)
order by ordercount desc;
```

## OUTPUT



The screenshot shows a database interface with a 'Result Grid' tab. The grid displays the results of a SQL query, showing the hour of the day and the corresponding order count, sorted in descending order of count. The interface includes a 'Filter Rows' button and a grid icon.

	hour	ordercount
▶	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

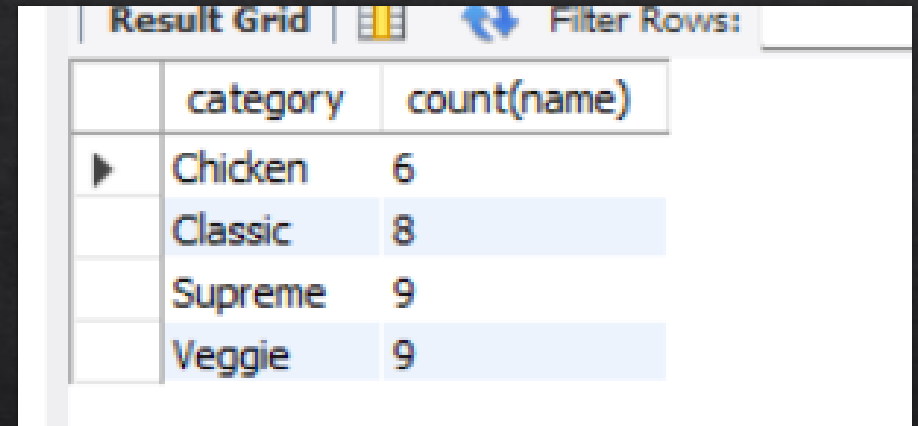


# The category-wise distribution of pizzas.

## CODE

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

## OUTPUT



The screenshot shows a database interface with a 'Result Grid' tab. It displays the results of a SQL query. The grid has two columns: 'category' and 'count(name)'. There are four rows of data, each with a small triangle icon in the first column. The rows are: Chicken (6), Classic (8), Supreme (9), and Veggie (9). The rows for Classic, Supreme, and Veggie are highlighted in light blue.

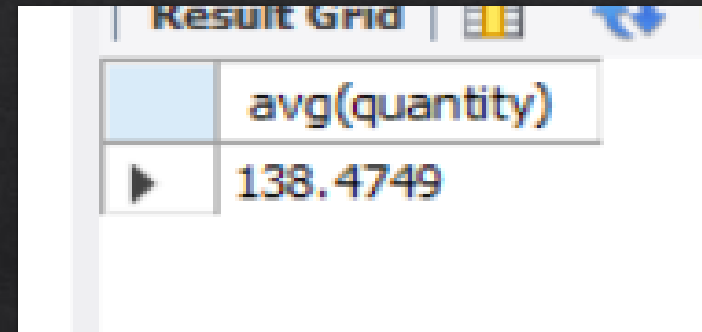
	category	count(name)
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

# The orders by date and the average number of pizzas ordered per day.

## CODE

```
select avg(quantity) 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 orderquantity;
```

## OUTPUT



The screenshot shows a 'Result Grid' window with a single row of data. The column header is 'avg(quantity)' and the value is '138.4749'. The window has a title bar and standard window controls.

	avg(quantity)
▶	138.4749

# The top 3 most ordered pizza types based on revenue.

## CODE

```
select 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 pizzas.pizza_id = orders_details.pizza_id
group by pizza_types.name order by revenue desc limit 3;
```

## OUTPUT

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

# The percentage contribution of each pizza type to total revenue.

## CODE

```
select pizza_types.category,  
round(sum(orders_details.quantity*pizzas.price)/(select sum(orders_details.quantity*pizzas.price) as total_sales  
from orders_details join pizzas ON pizzas.pizza_id = orders_details.pizza_id) * 100,2) 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	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68



# The cumulative revenue generated over time.

## CODE

```
select order_date,  
sum(revenue) over(order by order_date ) as cum_sales  
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

order_date	cum_sales
2015-01-01	2713.85000000000004
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	23990.3500000000002
2015-01-11	25862.65
2015-01-12	27781.7
2015-01-13	29831.3000000000003
2015-01-14	32358.7000000000004
2015-01-15	34343.500000000001



# The top 3 most ordered pizza types based on revenue for each pizza category

## CODE

```
select category,name,revenue
from
(select category,name,revenue,
rank() over(partition by category order by revenue desc) as ran
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 ran <=3;
```

## OUTPUT

	category	name	revenue
▶	Chicken	The Thai Chicken Pizza	43434.25
	Chicken	The Barbecue Chicken Pizza	42768
	Chicken	The California Chicken Pizza	41409.5
	Classic	The Classic Deluxe Pizza	38180.5
	Classic	The Hawaiian Pizza	32273.25
	Classic	The Pepperoni Pizza	30161.75
	Supreme	The Spicy Italian Pizza	34831.25
	Supreme	The Italian Supreme Pizza	33476.75
	Supreme	The Sicilian Pizza	30940.5
	Veggie	The Four Cheese Pizza	32265.700000000065
	Veggie	The Mexicana Pizza	26780.75
	Veggie	The Five Cheese Pizza	26066.5