

Basic:

Retrieve the total number of orders placed.

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

Your SQL query has been executed successfully.

```
-- Retrieve the total number of orders placed. SELECT COUNT(order_id) as total_orders FROM orders;
```

[Edit inline] [Edit] [Create PHP code]

Extra options

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 order_details.pizza_id = pizzas.pizza_id;
```

Showing rows 0 - 0 (1 total, Query took 0.3407 seconds.)

```
-- 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 order_details.pizza_id = pizzas.pizza_id;
```

[Edit inline] [Edit] [Create PHP code]

☐ Show all | Number of rows: 25 | Filter rows: Search this table

Extra options

total_sales
817860.05

Identify the highest-priced pizza.

```
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;
```

Showing rows 0 - 0 (1 total, Query took 0.0007 seconds.) [price: 35.95... - 35.95...]

```
-- Identify the highest-priced pizza. 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;
```

[Edit inline] [Edit] [Create PHP code]

Extra options

name	price
The Greek Pizza	35.95

Identify the most common pizza size ordered.

```
SELECT pizzas.size, COUNT(order_details.order_details_id) as common_size from pizzas JOIN order_details on pizzas.pizza_id=order_details.pizza_id GROUP BY pizzas.size;
```

Showing rows 0 - 4 (5 total, Query took 0.7348 seconds.)

```
SELECT pizzas.size, COUNT(order_details.order_details_id) as common_size from pizzas JOIN order_details on pizzas.pizza_id=order_details.pizza_id GROUP BY pizzas.size;
```

☐ Profiling [\[Edit inline \]](#) [\[Edit \]](#) [\[Explain SQL \]](#) [\[Create PHP code \]](#) [\[Refresh \]](#)

☐ Show all | Number of rows: 25 | Filter rows: Search this table

Extra options

size	common_size
L	18526
M	15385
S	14137
XL	544
XXL	28

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

```
SELECT pizza_types.name, sum(order_details.quantity) as quantity 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.name ORDER by quantity desc LIMIT 5;
```

Showing rows 0 - 4 (5 total, Query took 0.5551 seconds.)

```
SELECT pizza_types.name, sum(order_details.quantity) as quantity 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.name ORDER by quantity desc LIMIT 5;
```

☐ Profiling [\[Edit inline \]](#) [\[Edit \]](#) [\[Explain SQL \]](#) [\[Create PHP code \]](#) [\[Refresh \]](#)

Extra options

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:

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

```
SELECT pizza_types.category, SUM(order_details.quantity) AS totalprice_category 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 quantity DESC;
```

Showing rows 0 - 3 (4 total, Query took 0.6731 seconds.)

```
SELECT pizza_types.category, SUM(order_details.quantity) AS totalprice_category 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 quantity DESC;
```

☐ Profiling [\[Edit inline \]](#) [\[Edit \]](#) [\[Explain SQL \]](#) [\[Create PHP code \]](#) [\[Refresh \]](#)

☐ Show all | Number of rows: 25 | Filter rows: Search this table

Extra options

category	totalprice_category
Classic	14888
Veggie	11649
Supreme	11987
Chicken	11050

Determine the distribution of orders by hour of the day.

```
SELECT HOUR(time) AS HOUR , COUNT(order_id) AS HOUR_cOUNT FROM orders GROUP BY HOUR(time);
```

Showing rows 0 - 14 (15 total, Query took 0.0088 seconds.)

```
SELECT HOUR(time) AS HOUR , COUNT(order_id) AS HOUR_cOUNT FROM orders GROUP BY HOUR(time);
```

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

☐ Show all | Number of rows: 25 | Filter rows: Search this table

Extra options

HOUR	HOUR_cOUNT
9	1
10	8
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

Console

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

```
SELECT pizza_types.category, COUNT(pizzas.pizza_id) AS total_pizzas FROM pizzas JOIN pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_id GROUP BY pizza_types.category;
```

Showing rows 0 - 3 (4 total, Query took 0.0009 seconds.)

```
SELECT pizza_types.category, COUNT(pizzas.pizza_id) AS total_pizzas FROM pizzas JOIN pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_id GROUP BY pizza_types.category;
```

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

☐ Show all | Number of rows: 25 | Filter rows: Search this table

Extra options

category	total_pizzas
Chicken	18
Classic	26
Supreme	25
Veggie	27

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

```
SELECT round(AVG(daily_total),0) AS avg_pizzas_per_day FROM ( SELECT orders.date, SUM(order_details.quantity) AS daily_total FROM orders JOIN order_details ON orders.order_id = order_details.order_id GROUP BY orders.date ) AS daily_summary;
```

⚠ Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available. ⓘ

✓ Showing rows 0 - 0 (1 total, Query took 0.0653 seconds.)

```
SELECT round(AVG(daily_total),0) AS avg_pizzas_per_day FROM ( SELECT orders.date, SUM(order_details.quantity) AS daily_total FROM orders JOIN order_details ON orders.order_id = order_details.order_id GROUP BY orders.date ) AS daily_summary;
```

☐ Profiling [\[Edit inline \]](#) [\[Edit \]](#) [\[Explain SQL \]](#) [\[Create PHP code \]](#) [\[Refresh \]](#)

☐ Show all | Number of rows: 25 | Filter rows:

Extra options

avg_pizzas_per_day
138

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

```
SELECT pizza_types.name, SUM(pizzas.price * order_details.quantity) as total_Sales from order_details JOIN pizzas on order_details.pizza_id = pizzas.pizza_id JOIN pizza_types on pizzas.pizza_type_id = pizza_types.pizza_type_id GROUP by pizza_types.name ORDER by total_sales DESC LIMIT 3;
```

✓ Showing rows 0 - 2 (3 total, Query took 0.7583 seconds.)

```
SELECT pizza_types.name, SUM(pizzas.price * order_details.quantity) as total_Sales from order_details JOIN pizzas on order_details.pizza_id = pizzas.pizza_id JOIN pizza_types on pizzas.pizza_type_id = pizza_types.pizza_type_id GROUP by pizza_types.name ORDER by total_sales DESC LIMIT 3;
```

☐ Profiling [\[Edit inline \]](#) [\[Edit \]](#) [\[Explain SQL \]](#) [\[Create PHP code \]](#) [\[Refresh \]](#)

Extra options

name	total_Sales
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768.00
The California Chicken Pizza	41409.50

Query results operations

Advanced:

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

```
SELECT pizza_types.category, ROUND(SUM(pizzas.price * order_details.quantity), 2) AS category_sales, ROUND( 100.0 * SUM(pizzas.price * order_details.quantity) / (SELECT SUM(order_details.quantity * pizzas.price) FROM order_details JOIN pizzas ON order_details.pizza_id = pizzas.pizza_id), 2) AS percentage_of_total_sales FROM order_details JOIN pizzas ON order_details.pizza_id = pizzas.pizza_id JOIN pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id GROUP BY pizza_types.category ORDER BY percentage_of_total_sales DESC;
```

✓ Showing rows 0 - 3 (4 total, Query took 1.0272 seconds.)

```
SELECT pizza_types.category, ROUND(SUM(pizzas.price * order_details.quantity), 2) AS category_sales, ROUND( 100.0 * SUM(pizzas.price * order_details.quantity) / (SELECT SUM(order_details.quantity * pizzas.price) FROM order_details JOIN pizzas ON order_details.pizza_id = pizzas.pizza_id), 2) AS percentage_of_total_sales FROM order_details JOIN pizzas ON order_details.pizza_id = pizzas.pizza_id JOIN pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id GROUP BY pizza_types.category ORDER BY percentage_of_total_sales DESC;
```

☐ Profiling [\[Edit inline \]](#) [\[Edit \]](#) [\[Explain SQL \]](#) [\[Create PHP code \]](#) [\[Refresh \]](#)

☐ Show all | Number of rows: 25 | Filter rows:

Extra options

category	category_sales	percentage_of_total_sales
Classic	220053.10	26.91
Supreme	208197.00	25.46
Chicken	195919.50	23.96
Veggie	193690.45	23.68

Analyze the cumulative revenue generated over time.

```
SELECT date, SUM(revenue) over (ORDER by date) as cum_revenue FROM (SELECT orders.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.date) as sales;
```

✓ Showing rows 0 - 24 (358 total, Query took 0.6756 seconds.)

```
SELECT date, SUM(revenue) over (ORDER by date) as cum_revenue FROM (SELECT orders.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.date) as sales;
```

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

1 ▾

> >>

☐ Show all

Number of rows: 25 ▾

Extra options

date	cum_revenue
2015-01-01	2713.85
2015-01-02	5445.75
2015-01-03	8108.15
2015-01-04	9863.60
2015-01-05	11929.55
2015-01-06	14358.50
2015-01-07	16560.70
2015-01-08	19399.05
2015-01-09	21526.40
2015-01-10	23990.35
2015-01-11	25862.65
2015-01-12	27781.70
2015-01-13	29831.30
2015-01-14	32358.70
2015-01-15	34343.50
2015-01-16	36937.65

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 order_details.pizza_id = pizzas.pizza_id GROUP BY pizza_types.category, pizza_types.name ) AS a ) AS b WHERE rn <= 3;
```

✓ Showing rows 0 - 11 (12 total, Query took 0.7174 seconds.)

```
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 order_details.pizza_id =
pizzas.pizza_id GROUP BY pizza_types.category, pizza_types.name ) AS a ) AS b WHERE rn <= 3;
```

☐ Profiling [[Edit inline](#)] [[Edit](#)] [[Explain SQL](#)] [[Create PHP code](#)] [[Refresh](#)]

☐ Show all | Number of rows: 25 ▼

Extra options

name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768.00
The California Chicken Pizza	41409.50
The Classic Deluxe Pizza	38180.50
The Hawaiian Pizza	32273.25
The Pepperoni Pizza	30161.75
The Spicy Italian Pizza	34831.25
The Italian Supreme Pizza	33476.75
The Sicilian Pizza	30940.50
The Four Cheese Pizza	32265.70
The Mexicana Pizza	26780.75
The Five Cheese Pizza	26066.50