



PIZZA

SALES ANALYSIS



Home

About

Contact





Retrieve the total number of orders placed.

```
-- Retrieve the total number of orders placed.  
select count(order_id) as total_orders from orders;
```

Result Grid

	total_orders
▶	21350



Calculate the total revenue generated from pizza sales.

[Home](#)[About](#)[Contact](#)

```
1      -- Calculate the total revenue generated from pizza sales.  
2  
3 •  SELECT  
4     ROUND(SUM(order_details.quantity * pizzas.price),  
5            2) AS total_sales  
6  FROM  
7    order_details  
8    JOIN  
9    pizzas ON pizzas.pizza_id = order_details.pizza_id  
10  
11  
12  
13
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	total_sales			
▶	817860.05			



Identify the highest-priced pizza.

[Home](#)[About](#)[Contact](#)

```
1  -- Identify the highest-priced pizza.  
2  
3 • select pizza_types.name, pizzas.price  
4   from pizza_types join pizzas  
5     on pizza_types.pizza_type_id= pizzas.pizza_type_id  
6   order by pizzas.price desc limit 1;
```

\$30

< []

Result Grid | Filter Rows: Export: Wrap Cell Content: Fetch rows:

	name	price
▶	The Greek Pizza	35.95

\$30



Identify the most common pizza size ordered.

[Home](#)[About](#)[Contact](#)

```
1  -- Identify the most common pizza size ordered.  
2  
3 • select pizzas.size, count(order_details.order_details_id) as order_count  
4  from pizzas join order_details  
5  on pizzas.pizza_id = order_details.pizza_id  
6  group by pizzas.size  
7  order by order_count desc limit 1;  
8
```

The screenshot shows a database query results grid. The grid has two columns: 'size' and 'order_count'. There is one row of data with the value 'L' in the 'size' column and '18526' in the 'order_count' column. The grid includes standard SQL navigation buttons (first, previous, next, last) and various export options (Result Grid, Filter Rows, Export, Wrap Cell Content, Fetch rows).

	size	order_count
▶	L	18526



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

[Home](#)[About](#)[Contact](#)

```
1 -- List the top 5 most ordered pizza types along with their quantities.  
2  
3 • select pizza_types.name ,  
4 sum(order_details.quantity) as quantity  
5 from pizza_types join pizzas  
6 on pizza_types.pizza_type_id = pizzas.pizza_type_id  
7 join order_details  
8 on order_details.pizza_id = pizzas.pizza_id  
9 group by pizza_types.name order by quantity desc limit 5;
```

[Result Grid](#)

Filter Rows:

Export:



Wrap Cell Content:



Fetch rows:



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



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

[Home](#) [About](#) [Contact](#)

```
1      -- Join the necessary tables to find the total quantity of each pizza category ordered.  
2  
3 •  select pizza_types.category,  
4      sum(order_details.quantity) as quantity  
5      from pizza_types join pizzas  
6      on pizza_types.pizza_type_id= pizzas.pizza_type_id  
7      join order_details  
8      on order_details.pizza_id  = pizzas.pizza_id  
9      group by pizza_types.category order by quantity desc;
```

< [refresh] [grid] Filter Rows: [] Export: [] Wrap Cell Content: []

	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050



Determine the distribution of orders by hour of the day.

[About](#)[Contact](#)

```
1      -- Determine the distribution of orders by hour of the day.  
2  
3 • SELECT  
4      HOUR(order_time) AS hour, COUNT(order_id) AS order_count  
5      FROM  
6      orders  
7      GROUP BY HOUR(order_time);  
8  
9  
10
```

Result Grid | Filter Rows: Export: Wrap Cell Content:

	hour	order_count
▶	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



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

Home About Contact

```
1 -- Group the orders by date and calculate the average number of pizzas ordered per day.  
2 • select round(avg(quantity),0) from  
3   (select orders.order_date , sum(order_details.quantity) as quantity  
4     from orders join order_details  
5       on orders.order_id = order_details.order_id  
6       group by orders.order_date) as order_quantity;  
7  
8
```

< |

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

	round(avg(quantity),0)
▶	138

\$30





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

```
1      -- Determine the top 3 most ordered pizza types based on revenue.  
2  
3 •  select pizza_types.name,  
4      sum(order_details.quantity*pizzas.price) as revenue  
5      from pizza_types join pizzas  
6      on pizza_types.pizza_type_id = pizzas.pizza_type_id  
7      join order_details  
8      on order_details.pizza_id = pizzas.pizza_id  
9      group by pizza_types.name  
10     order by revenue desc limit 3 ;  
11
```

About

Contact

< |

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



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

[Home](#)[About](#)[Contact](#)

```
1  -- Calculate the percentage contribution of each pizza type to total revenue.  
2  
3  ● select pizza_types.category,  
4  ○ concat(round(sum(order_details.quantity*pizzas.price) / (select round(sum(order_details.quantity*pizzas.price),2) as total_sales  
5  from order_details join pizzas  
6  on pizzas.pizza_id= order_details.pizza_id)*100,2) , '%' )as revenue  
7  from pizza_types join pizzas  
8  on pizza_types.pizza_type_id = pizzas.pizza_type_id  
9  join order_details  
10 on order_details.pizza_id = pizzas.pizza_id  
11 group by pizza_types.category  
12 order by revenue desc;  
13
```

<

Result Grid | Filter Rows: Export: Wrap Cell Content:

category	revenue
Classic	26.91%
Supreme	25.46%
Chicken	23.96%
Veggie	23.68%



Analyze the cumulative revenue generated over time.

[Home](#)[About](#)[Contact](#)

```
1  -- Analyze the cumulative revenue generated over time.  
2 • select order_date,  
3   sum(revenue) over(order by order_date) as cum_revenue  
4   from  
5   (select orders.order_date ,  
6    sum(order_details.quantity* pizzas.price) as revenue  
7    from order_details join pizzas  
8    on order_details.pizza_id = pizzas.pizza_id  
9    join orders  
10   on orders.order_id = order_details.order_id  
11   group by orders.order_date) as sales ;  
12  
13
```

Result Grid | Filter Rows: Export: Wrap Cell Content:

	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

Result 2 ×

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



```
1      -- Determine the top 3 most ordered pizza types based on revenue for each pizza category.
2 •  select category, name, revenue
3   from
4   (select category, name, revenue,
5    rank() over( partition by category order by revenue desc) as rn
6   from
7   (select pizza_types.category, pizza_types.name,
8    sum((order_details.quantity)*pizzas.price) as revenue
9   from pizza_types join pizzas
10  on pizza_types.pizza_type_id= pizzas.pizza_type_id
11  join order_details
12  on order_details.pizza_id = pizzas.pizza_id
13  group by pizza_types.category, pizza_types.name) as a) as b
14  where rn<=3 ;
15
```

Result Grid | Filter Rows: | Export: Wrap Cell Content:

	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	32265.70000000065

About

Contact