

PIZZA SALES





PIZZA

-- 1 Retrieve the total number of orders placed.

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

	Result Grid			Filter Rows:
	total_orders			
▶	21350			



PIZZA

-- 2 Calculate the total revenue generated from pizza sales.

```
select
round(sum(order_details.quantity * pizzas.price), 2) as total_revenue
from order_details
Join pizzas
on pizzas.pizza_id = order_details.pizza_id;
```

Result Grid

	total_revenue
▶	817860.05



-- 3 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;
```

Result Grid | Filter Rows:

	name	price
▶	The Greek Pizza	35.95



-- 4 Identify the most common pizza size ordered.

```
select quantity, count(order_details_id)
from order_details group by quantity;

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;
```

Result Grid | Filter

	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.



PIZZA

```
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 order_details.pizza_id = pizzas.pizza_id  
 group by pizza_types.name  
 order by quantity desc  
 limit 5;
```

Result Grid | Filter 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



PIZZA

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

```
select pizza_types.category,  
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 order_details.pizza_id = pizzas.pizza_id  
group by pizza_types.category  
order by quantity desc;
```

Result Grid | Filter

	category	quantity
▶	Classic	14388
	Supreme	11987
	Veggie	11649
	Chicken	11050



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

```
select hour(order_time) as hour, count(order_id) as order_count from orders  
group by hour(order_time);
```

Result Grid | Filter

	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



PIZZA

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

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

Result Grid | Filter Rows

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



PIZZA

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

```
SELECT
```

```
    ROUND(AVG(quantity), 0) as avg_pizza_ordered_per_day
```

```
FROM
```

```
(SELECT
```

```
    orders.order_date, SUM(order_details.quantity) as quantity
```

```
FROM
```

```
orders
```

```
JOIN order_details ON orders.order_id = order_details.order_id
```

```
GROUP BY orders.order_date) AS order_quantity;
```

Result Grid | Filter Rows:

	avg_pizza_ordered_per_day
▶	138

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



PIZZA

```
select pizza_types.name,  
sum(order_details.quantity * pizzas.price) as revenue  
from pizza_types  
join pizzas  
on pizzas.pizza_type_id = pizza_types.pizza_type_id  
join order_details  
on order_details.pizza_id = pizzas.pizza_id  
group by pizza_types.name  
order by revenue desc  
limit 3;
```

Result Grid | Filter Rows:

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



```
-- 11 calculate the percentage contribution  
-- of each pizza type to total revenue.
```

```
select 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 category  
order by revenue desc;
```

Result Grid | Filter Rows:

	category	revenue
▶	Classic	220053.1000000001
	Supreme	208196.99999999822
	Chicken	195919.5
	Veggie	193690.45000000298



PIZZA

-- 12 Analyze the cumulative revenue generated over time.

```
select order_date,
round(sum(revenue) over(order by order_date), 0) as cum_revenue
from
(select orders.order_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.order_date) as sales;
```

	order_date	cum_revenue
▶	2015-01-01	2714
	2015-01-02	5446
	2015-01-03	8108
	2015-01-04	9864
	2015-01-05	11930
	2015-01-06	14358
	2015-01-07	16561
	2015-01-08	19399
	2015-01-09	21526
	2015-01-10	23990
	2015-01-11	25863
	2015-01-12	27782
	2015-01-13	29831
	2015-01-14	32359
	2015-01-15	34344