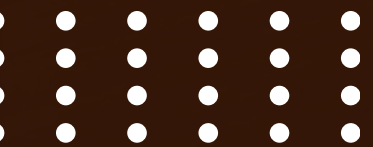


PIZZA SALES



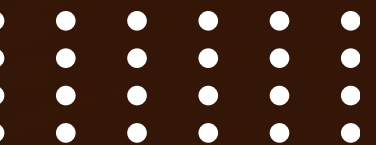
Analyze pizza sales data to derive insights on orders, revenue, popular pizzas, and ordering patterns.





DATABASE SCHEMA OVERVIEW

- **orders:** Contains order details like order ID and time.
- **orders_details:** Contains specific order items with quantity.
- **pizzas:** Contains pizza information like price and size.
- **pizza_types:** Details on pizza types (name and category).



-- 1) Retrieve the total number of orders placed.

SELECT

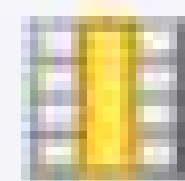
COUNT(order_id) AS total_orders

FROM

orders;

Result Grid	
	total_orders
▶	21350

Result Grid



	total_sales
▶	817860.05

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

• SELECT

ROUND(SUM(orders_details.quantity * pizzas.price),

2) AS total_sales

FROM

orders_details

JOIN

pizzas ON orders_details.pizza_id = pizzas.pizza_id;

-- 3) Identify the highest-priced pizza.

SELECT

 pizza_types.name AS pizza_name, pizzas.price AS price

FROM

 pizza_types

 JOIN

 pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id

ORDER BY price DESC

LIMIT 1;

Result Grid



Filter Rows

	pizza_name	price
▶	The Greek Pizza	35.95

-- 4) Identify the most common pizza size ordered.

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

LIMIT 3;

Result Grid					Filter
	size	order_count			
▶	L	18526			
	M	15385			
	S	14137			

```
-- 5) List the top 5 most ordered pizza types along with their quantities.
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;
```

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	



```
-- 6) Intermediate:
```

```
-- Join the necessary tables to find the
```

```
-- total quantity of each pizza category ordered.
```

```
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;
```

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			


```
-- 6) Intermediate:
```

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

```
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;
```

Result Grid



Filter

	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

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



SELECT

HOUR(order_time) AS hours, COUNT(order_id) AS order_count

FROM

orders

GROUP BY hours;

Result Grid				 Filter R
	hours	order_count		
▶	11	1231		
	12	2520		
	13	2455		
	14	1472		
	15	1468		
	16	1920		


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

```
SELECT
```

```
    category, COUNT(name) AS Pizzas_Count
```

```
FROM
```

```
    pizza_types
```

```
GROUP BY category
```



```
ORDER BY Pizzas_Count DESC;
```

Result Grid

	category	Pizzas_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(quantity), 0) AS avg_pizzas_per_day  
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 order_quantity;
```

Result Grid				F
	avg_pizzas_per_day			
▶	138			


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

```
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 orders_details.pizza_id = pizzas.pizza_id
```

```
GROUP BY pizza_types.name
```

```
ORDER BY revenue DESC
```

```
LIMIT 3;
```

Result Grid



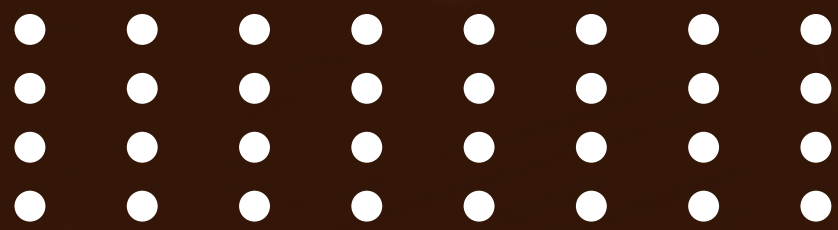
Filter Rows:

	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, ROUND(SUM(orders_details.quantity * pizzas.price)/  
(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) * 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;
```

Result Grid				Filter
	category	revenue		
▶	Classic	26.91		
	Supreme	25.46		
	Chicken	23.96		
	Veggie	23.68		



-- 12) Analyze the cumulative revenue generated over time.

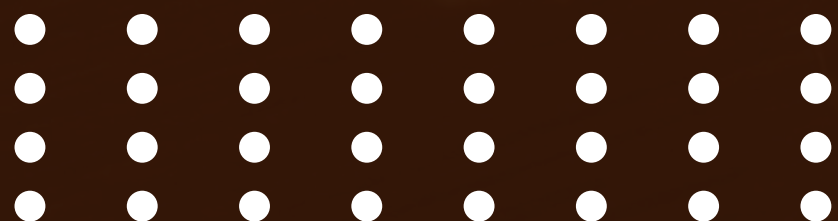
```
select order_date, revenue as daily_revenue, sum(revenue) over (order by order_date) as cum_revenue
from
(select orders.order_date, round(sum(orders_details.quantity * pizzas.price),0) as revenue
from orders_details join pizzas
on orders_details.pizza_id = pizzas.pizza_id join orders
on orders.order_id = orders_details.order_id
group by orders.order_date) as sales;
```

Result Grid



Filter Rows:


	order_date	daily_revenue	cum_revenue
▶	2015-01-01	2714	2714
	2015-01-02	2732	5446
	2015-01-03	2662	8108
	2015-01-04	1755	9863
	2015-01-05	2066	11929
	2015-01-06	2429	14358




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

```
select category, name, revenue from  
(select category, name, revenue, rank() over  
(partition by category order by revenue desc) as ranking  
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 pizzas.pizza_id = orders_details.pizza_id  
group by pizza_types.category, pizza_types.name) as Table_A) as Table_B  
where ranking<=3;
```

Result Grid





Filter Rows:

	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

**THANK YOU
FOR ATTENTION**

