



DATASETS

order_details

- order_details_id
- order_id
- pizza_id
- quantity

pizza_types

- pizza_type_id
- name
- category
- ingredients

orders

- order_id
- date
- time

pizzas

- pizza_id
- pizz_type_id
- size
- price



Retrieve the total number of orders placed. **SELECT** COUNT(*) AS Total_orders_placed **FROM** orders; Total_orders_placed 21350

Calculate the total revenue generated from pizza sales.



	Total_Sales	
•	817860.05	

```
Identify the highest-priced pizza.
SELECT
    pizza_types.name, pizzas.price
FROM
    pizza_types
        INNER JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
ORDER BY price DESC
LIMIT 1;
                                     price
                        name
                       The Greek Pizza
                                     35.95
```

Identify the most common pizza size ordered.

SELECT

```
pizzas.size,
```

COUNT(order_details.order_detail_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;

	size	order_count
•	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
        INNER 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;

	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 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 category
ORDER BY quantity DESC;
                                category
                                         quantity
                               Classic
                                         14888
                               Supreme
                                        11987
```

Veggie

Chicken

11649

11050

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);



	hour	order_count
•	11	1231
	12	2520
	13	2455
	14	1472
	15	1468

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

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

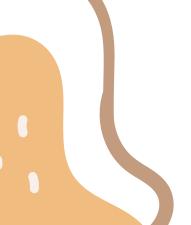


	category	category_count
•	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9



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

```
SELECT
    ROUND(AVG(quantity),0) AS average_pizza_ordered_per_day
FROM
    (SELECT
        orders.order_date, SUM(order_details.quantity) AS quantity
    FROM
        orders
    INNER JOIN order_details ON order_details.order_id = orders.order_id
    GROUP BY orders.order_date) AS order_quantity;
```



average_pizza_ordered_per_day

138

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

```
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;

	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.

```
pizza_types.category,
    ROUND((SUM(order_details.quantity * pizzas.price) / (SELECT
                    ROUND(SUM(order_details.quantity * pizzas.price),
                                2) AS Total_Sales
                FROM
                    order_details
                        INNER JOIN
                    pizzas ON order_details.pizza_id = pizzas.pizza_id)) * 100,
            2) 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
ORDER BY revenue DESC;
```

	category	revenue
•	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

Analyze the cumulative revenue generated over time.

2015-01-01

2015-01-04

2015-01-02 5445.75

2015-01-03 8108.15

9863.6

2713.85000000000004

```
select order date,
sum(revenue) over(order by order_date) 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
```

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;

The California Chicken Pizza
 The Southwest Chicken Pizza
 The Chicken Alfredo Pizza

The Chicken Pesto Pizza

revenue

41409.5

34705,75

16900.25

16701.75

