



Delicious **PIZZA**

35%
off

SALES PROJECT

ORDER NOW

HELLO!

My name is Subham Kumar Singh, and I completed a project utilizing SQL queries to analyze pizza sales data. This project aimed to identify best-selling pizzas, peak sales times, customer preferences, and operational efficiencies. By extracting and interpreting data from a comprehensive sales database, I provided actionable insights into sales trends and customer behaviors, which can inform strategic decisions and optimize business processes for pizza sales operations.

Questions:



- Retrieve the total number of orders placed.
- Calculate the total revenue generated from pizza sales.
- Identify the highest-priced pizza.
- Identify the most common pizza size ordered.
- List the top 5 most ordered pizza types along with their quantities.
- text

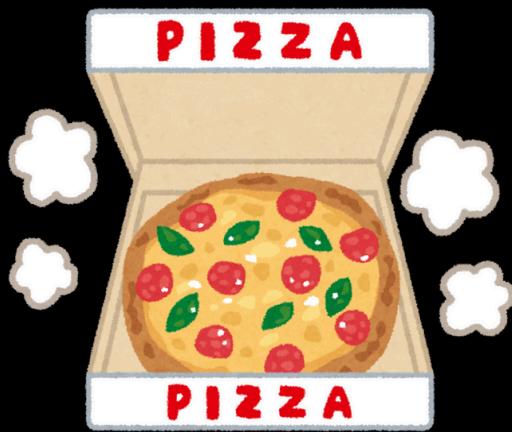
Intermediate:

- Join the necessary tables to find the total quantity of each pizza category ordered.
- Determine the distribution of orders by hour of the day.
- Join relevant tables to find the category-wise distribution of pizzas.
- Group the orders by date and calculate the average number of pizzas ordered per day.
- Determine the top 3 most ordered pizza types based on revenue.

Advanced:

- Calculate the percentage contribution of each pizza type to total revenue.
- Analyze the cumulative revenue generated over time.
- Determine the top 3 most ordered pizza types based on revenue for each pizza category.

Calculate the total revenue generated from pizza sales.



SELECT

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

```
2) AS total_sales
```

FROM

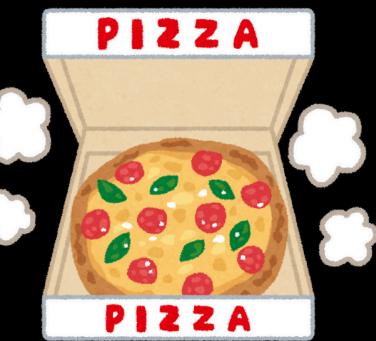
```
orders_details
```

```
JOIN
```

```
pizzas ON pizzas.pizza_id = orders_details.pizza_id;
```

Result Grid	
	total_sales
▶	817860.05

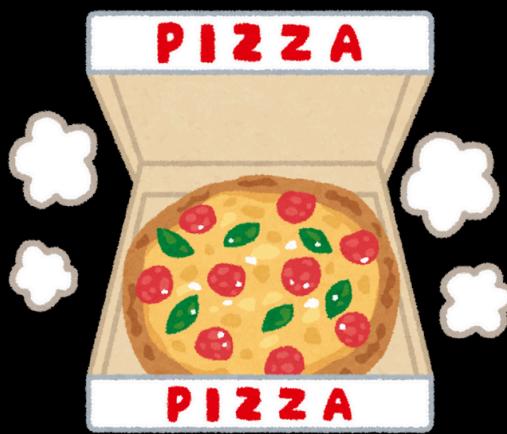
Retrieve the total number of orders placed.



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

Result Grid

	total_orders
▶	21350



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



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;
```

Result Grid | Filter Rows:

	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(orders_details.quantity) AS quantity_order
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_order DESC
LIMIT 5;
```

Result Grid | Filter Rows:

	name	quantity_order
▶	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.

```
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:

	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

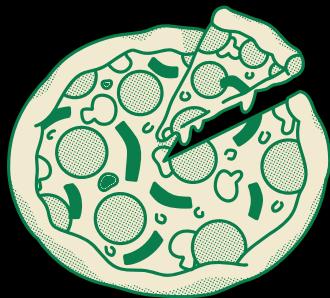


Determine the distribution of orders by hour of the day.



SELECT

```
HOUR(order_time) AS hour,  
COUNT(order_id) AS quantity_ordered  
FROM  
orders  
GROUP BY HOUR(order_time);
```



	hour	quantity_ordered
▶	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



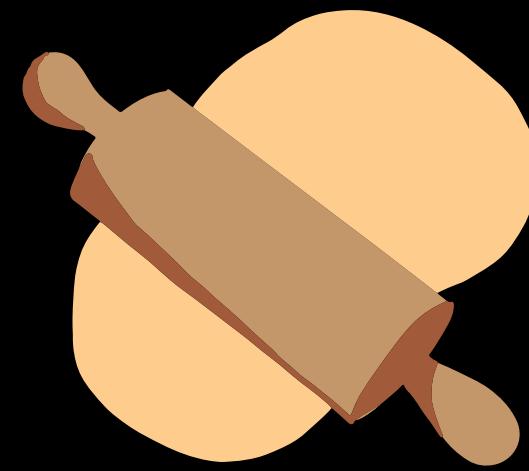
Join relevant tables to find the category-wise distribution of pizzas.

```
SELECT  
    category, COUNT(name) AS count  
FROM  
    pizza_types  
GROUP BY category;
```

Result Grid | Filter Rows:

	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 avg_pizza_ordered_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 quantity_ordered;
```

Result Grid | Filter Rows:

	avg_pizza_ordered_per_day
▶	138

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 pizzas.pizza_type_id = pizza_types.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 |

	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.

```
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 pizzas.pizza_type_id = pizza_types.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

Analyze the cumulative revenue generated over time.



```
SELECT order_date,  
       SUM(revenue) OVER(ORDER BY order_date) AS cum_revenue  
  FROM  
    (SELECT orders.order_date,  
           SUM(orders_details.quantity * pizzas.price) 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;
```

	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

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(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.category, pizza_types.name) AS a) AS b
  WHERE rn <=3;
```

	name	revenue
▶	The Thai Chicken Pizza	43434.25
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41409.5
	The Classic Deluxe Pizza	38180.5
	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	30940.5
	The Four Cheese Pizza	32265.70000000065