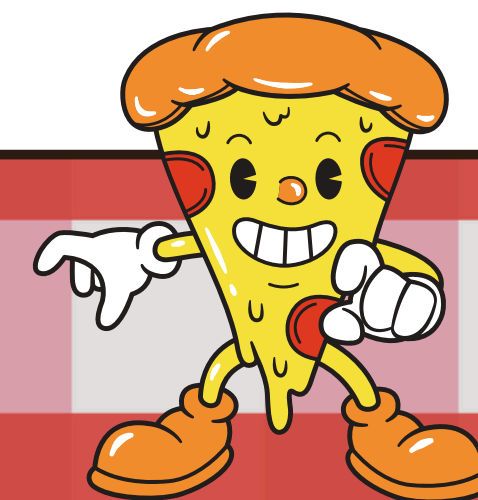
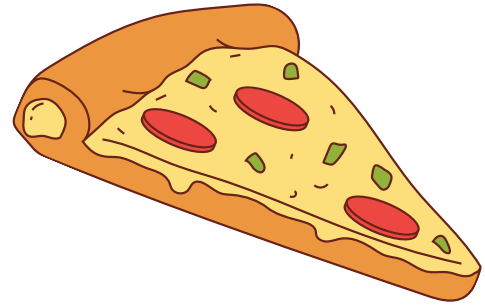


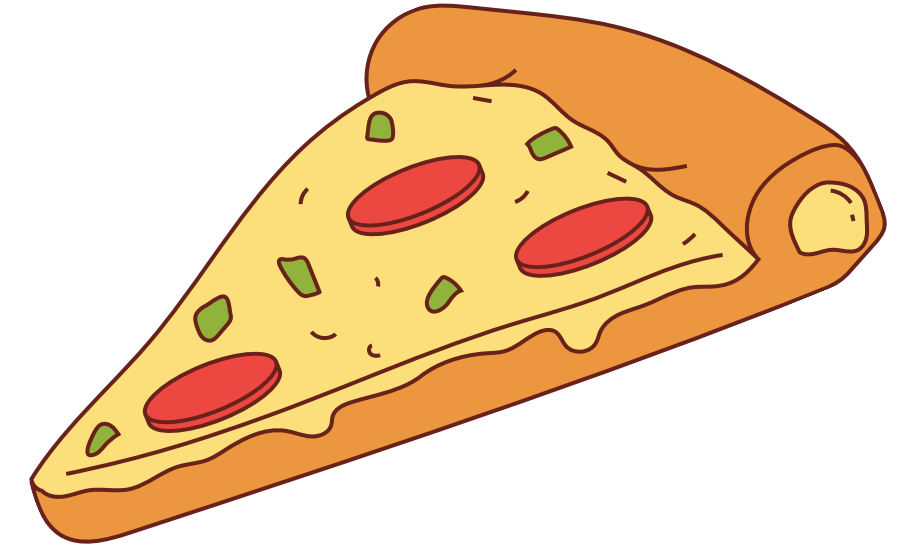
SQL PROJECT

PIZZA COMPANY

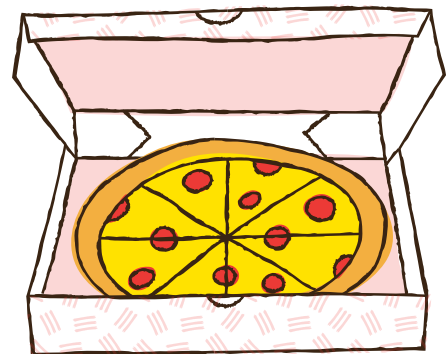




About



Hi, my name is Badullah Ali. This is my SQL project, where I work with a dataset from a pizza company. The dataset consists of four tables, and I have addressed various questions categorized into three levels: basic, intermediate, and advanced.



Retrieve the total number of orders placed.

Code

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

	count(order_id)
▶	21350

Calculate the total revenue generated from pizza sales.

Code

```
SELECT round (sum(quantity * price),2) as Total_Sales  
FROM order_details AS od  
JOIN pizzas AS pi  
ON od.pizza_id = pi.pizza_id;
```

	Total_Sales
▶	817860.05

Identify the highest-priced pizza.

Code

```
SELECT pizza_types.name AS pizza_type, --  
       pizzas.price AS pizza_price  
FROM pizza_types  
JOIN pizzas  
ON pizza_types.pizza_type_id = pizzas.pizza_type_id  
order by pizza_price desc  
limit 1;
```

	pizza_type	pizza_price
▶	The Greek Pizza	35.95

Identify the most common pizza size ordered.

Code

```
SELECT pizzas.size, COUNT(order_details.order_details_id) AS order_count
FROM pizzas
JOIN order_details
ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizzas.size
ORDER BY order_count DESC
LIMIT 1;
```

	size	order_count
▶	L	18526

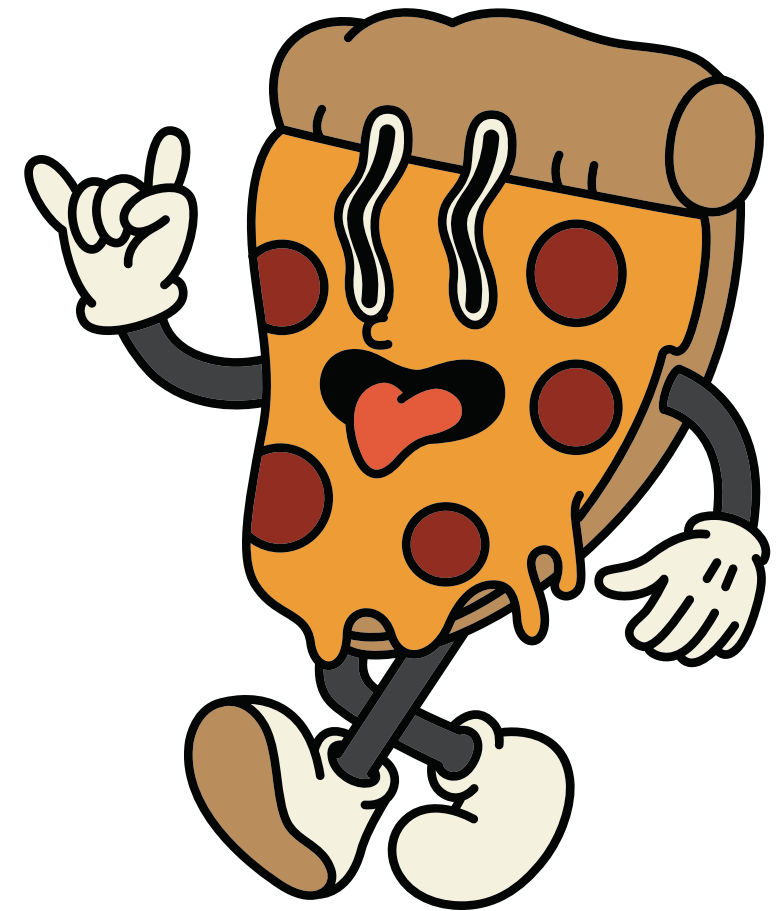
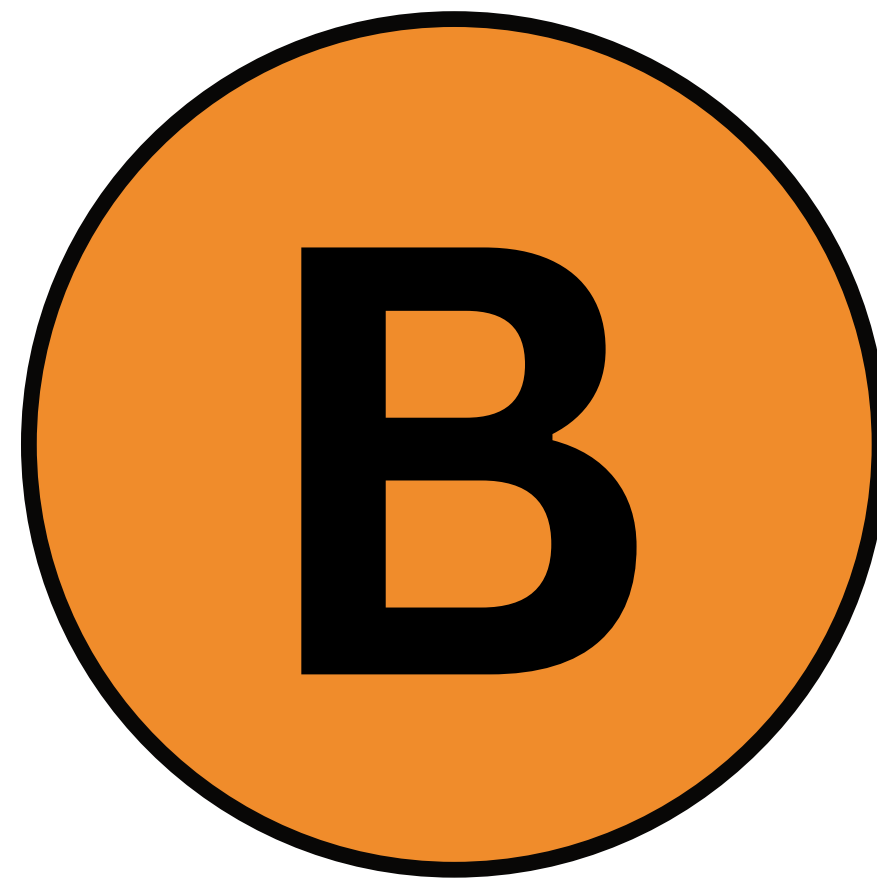
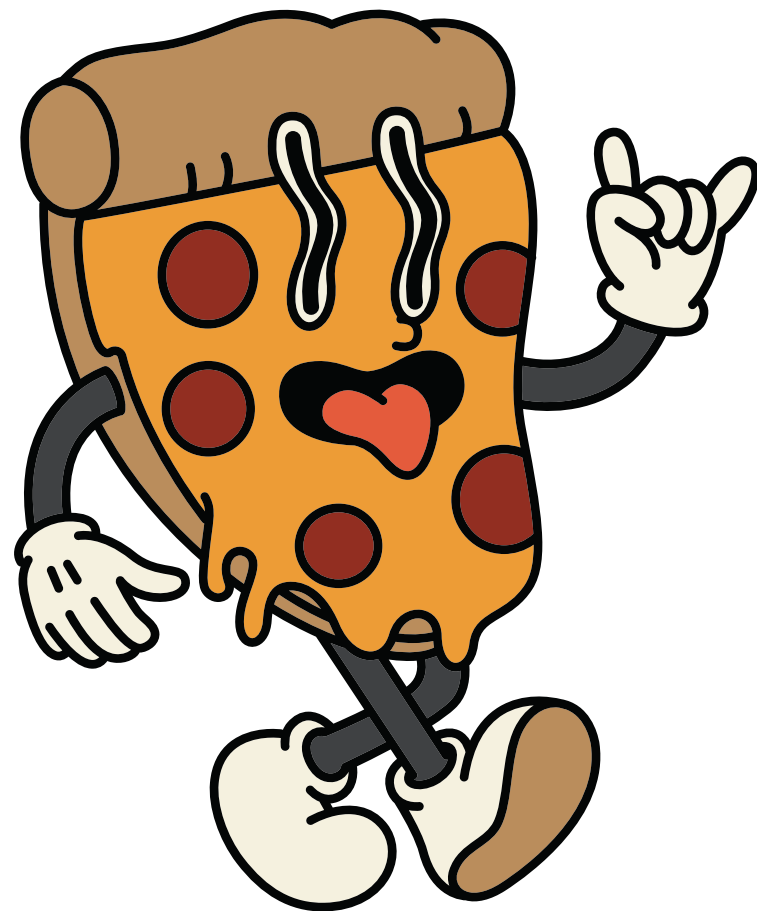
List the top 5 most ordered pizza types along with their quantities.

Code

```
SELECT pizzas.pizza_type_id, pizza_types.name AS pizza_type, COUNT(order_details.quantity) AS order_count
FROM order_details
JOIN pizzas
  ON order_details.pizza_id = pizzas.pizza_id
JOIN pizza_types
  ON pizzas.pizza_type_id = pizza_types.pizza_type_id -- Join based on pizza_type_id
GROUP BY pizzas.pizza_type_id, pizza_types.name
ORDER BY order_count DESC
LIMIT 5;
```

	pizza_type_id	pizza_type	order_count
▶	classic_dlx	The Classic Deluxe Pizza	2416
	bbq_chn	The Barbecue Chicken Pizza	2372
	hawaiian	The Hawaiian Pizza	2370
	pepperoni	The Pepperoni Pizza	2369
	thai_chn	The Thai Chicken Pizza	2315

Intermediate



Determine the distribution of orders by hour of the day.

Code

```
select hour(order_time), count(order_id) as orders
from orders
group by hour(order_time)
order by orders desc
```

	hour(order_time)	orders
▶	12	2520
	13	2455
	18	2399
	17	2336
	19	2009
	16	1920
	20	1642
	14	1472
	15	1468
	11	1231
	21	1198
	22	663
	23	28
	10	8
	9	1

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

Code

```
SELECT pizza_types.category, SUM(order_details.quantity) AS total_orders
FROM pizza_types
JOIN pizzas
  ON pizza_types.pizza_type_id = pizzas.pizza_type_id
JOIN order_details
  ON pizzas.pizza_id = order_details.pizza_id -- Join on pizza ID
GROUP BY pizza_types.category
ORDER BY total_orders DESC;
```

	category	total_orders
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

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

Code

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

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

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

Code

```
SELECT AVG(quantity)
FROM (
    SELECT orders.order_data, sum(order_details.quantity) AS quantity
    FROM orders
    JOIN order_details
    ON order_details.order_id = orders.order_id
    GROUP BY orders.order_data
) AS order_data;
```

	AVG(quantity)
▶	138.4749

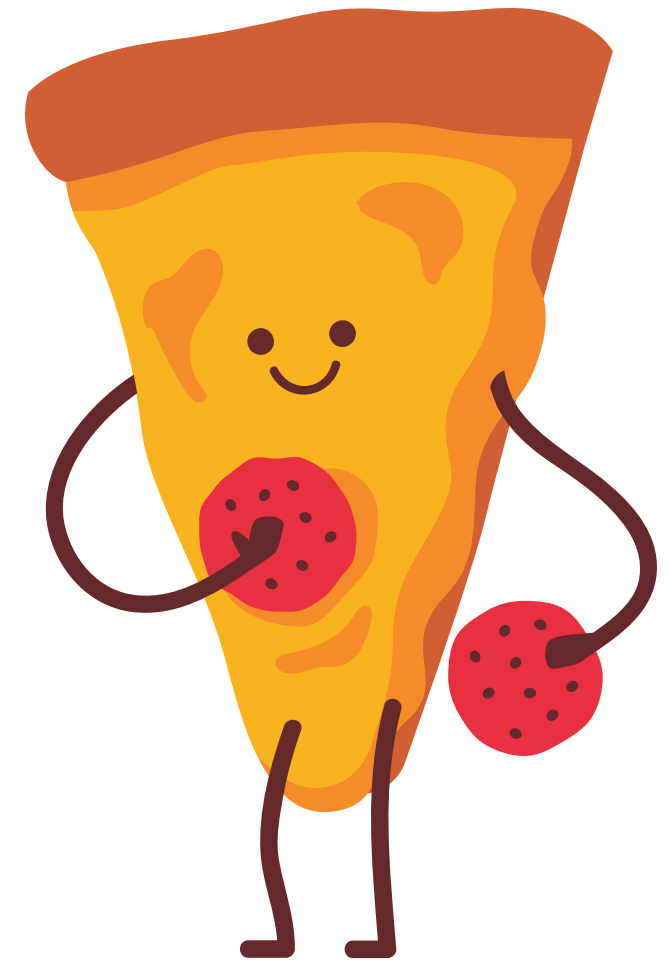
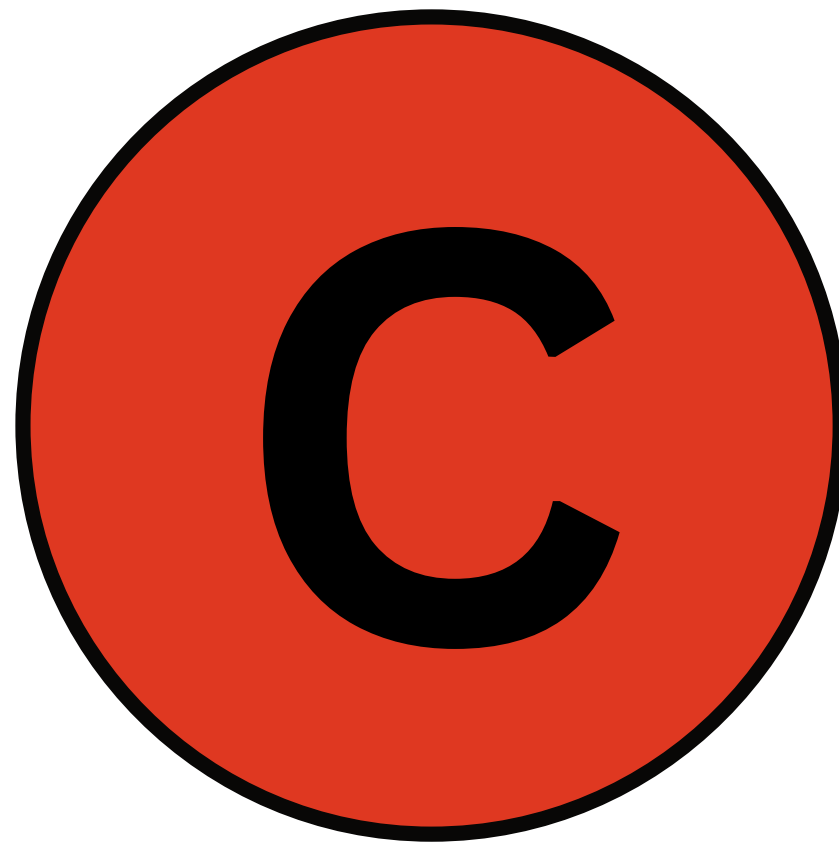
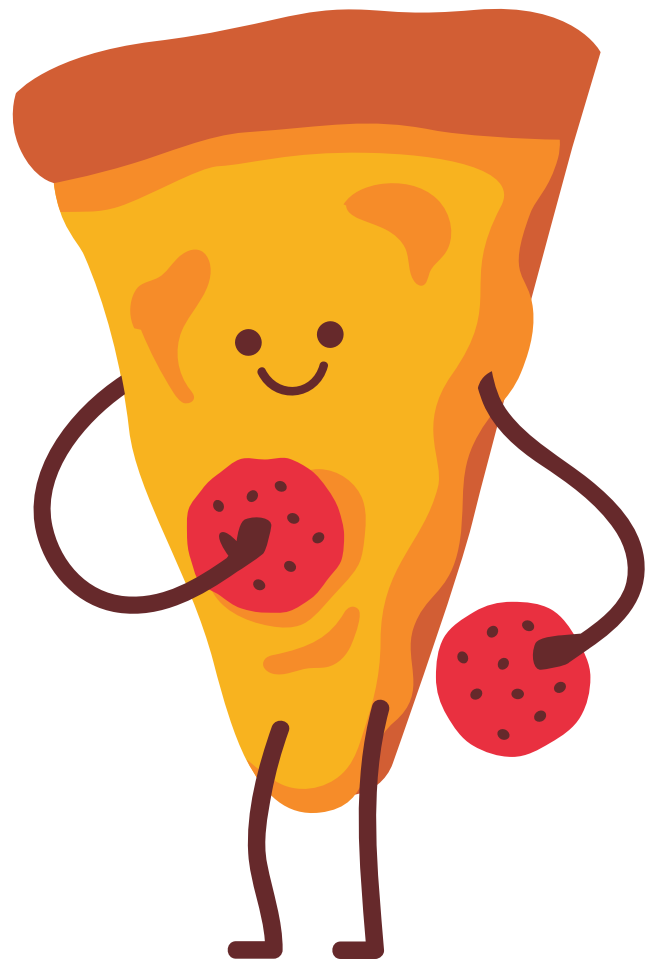
Determine the top 3 most ordered pizza types based on revenue.

Code

```
SELECT pizza_types.name, SUM(pizzas.price * order_details.quantity) AS rev
FROM pizza_types
JOIN pizzas
ON pizza_types.pizza_type_id = pizzas.pizza_type_id
JOIN order_details
ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizza_types.name
ORDER BY rev DESC
LIMIT 3;
```

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

Advanced



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

Code

SELECT

```
    pizza_types.category,  
    ROUND((SUM(pizzas.price * order_details.quantity) / (  
        SELECT SUM(quantity * price)  
        FROM order_details AS od  
        JOIN pizzas AS pi ON od.pizza_id = pi.pizza_id  
    ) * 100), 2) AS rev_percentage
```

FROM

```
    pizza_types
```

JOIN

```
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
```

JOIN

```
    order_details ON pizzas.pizza_id = order_details.pizza_id
```

GROUP BY

```
    pizza_types.category
```

ORDER BY

```
    rev_percentage DESC;
```

	category	rev_percentage
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

Analyze the cumulative revenue generated over time.

Code

```
SELECT
  order_data,
  SUM(rev) OVER (ORDER BY order_data) AS cumulative_rev
FROM
  (SELECT
    orders.order_data,
    SUM(order_details.quantity * pizzas.price) AS rev
  FROM
    orders
  JOIN
    order_details ON orders.order_id = order_details.order_id
  JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id
  GROUP BY
    orders.order_data
  ) AS sales;
```

2015-01-19	43365.750000000001
2015-01-20	45763.650000000001
2015-01-21	47804.200000000001
2015-01-22	50300.900000000001
2015-01-23	52724.600000000006
2015-01-24	55013.850000000006
2015-01-25	56631.400000000001
2015-01-26	58515.800000000001
2015-01-27	61043.850000000001
2015-01-28	63059.850000000001
2015-01-29	65105.150000000016
2015-01-30	67375.450000000001
2015-01-31	69793.300000000002
2015-02-01	72982.500000000001
2015-02-02	75311.100000000002
2015-02-03	77925.900000000002

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

Code

```
SELECT name, rev
FROM (
  SELECT
    name,
    rev,
    RANK() OVER (PARTITION BY category ORDER BY rev DESC) AS `rank`
  FROM (
    SELECT
      pizza_types.category,
      pizza_types.name,
      SUM(order_details.quantity * pizzas.price) AS rev
    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 `rank` >= 3;
```

	name	rev
▶	The California Chicken Pizza	41409.5
	The Southwest Chicken Pizza	34705.75
	The Chicken Alfredo Pizza	16900.25
	The Chicken Pesto Pizza	16701.75
	The Pepperoni Pizza	30161.75
	The Greek Pizza	28454.1000000000013
	The Italian Capocollo Pizza	25094
	The Napolitana Pizza	24087
	The Big Meat Pizza	22968
	The Pepperoni, Mushroom, and Peppers Pizza	18834.5
	The Sicilian Pizza	30940.5
	The Pepper Salami Pizza	25529
	The Prosciutto and Arugula Pizza	24193.25
	The Soppressata Pizza	16425.75
	The Calabrese Pizza	15934.25
	The Spinach Supreme Pizza	15277.75

THANK YOU

