



PIZZA STORE

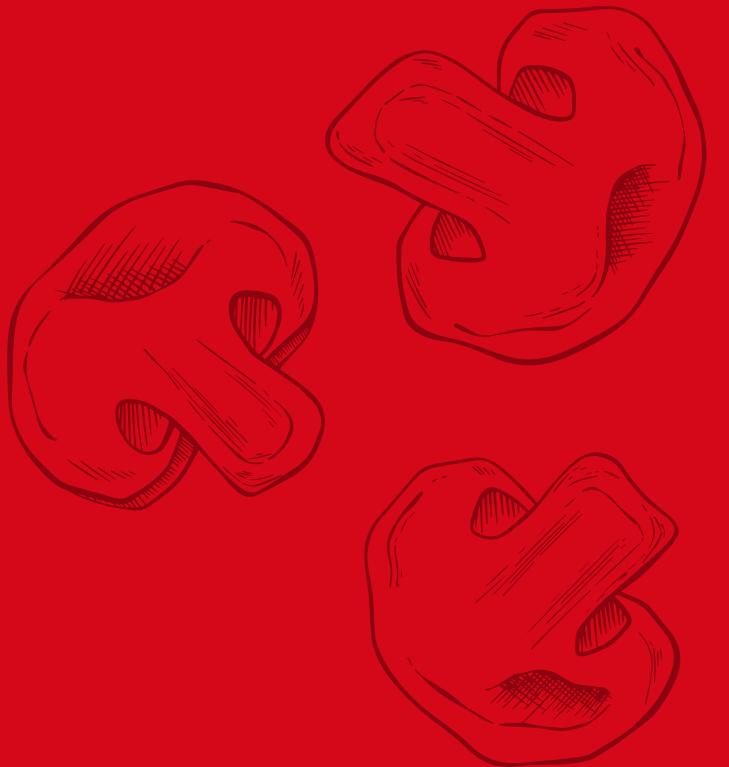
HOME

ABOUT

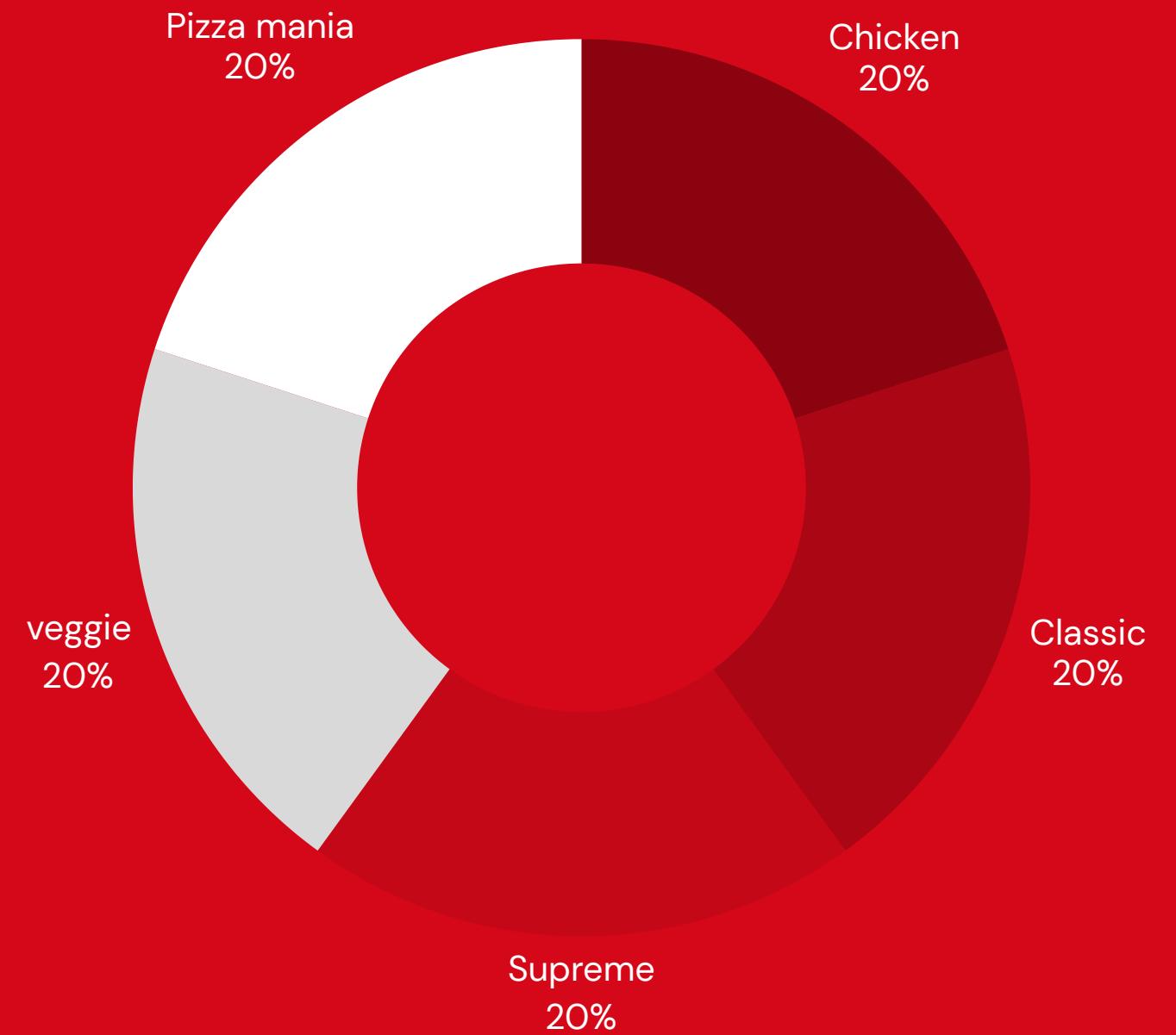
MORE

# PROJECT

## PIZZA SALES REVIEW



Presented by  
VIJAYESH KUMAR





# Introduction

My name is vijayesh kumar. I have developed a project on pizza sales, utilizing sql commands to analyse key methods aimed at maximising sales and driving revenue growth.

The objectives of this project is to analyse total revenue generated, identify opportunities to increase revenue, optimise product costs, improve product quality and quantity, introduce new product categories and boost the number of orders.



# Agenda

- **Problem Statement**

Task 1: Retrieve the total number of orders placed.

Task 2: Calculate the total revenue generated from pizza sales.

Task 3: Identify the highest-priced pizza.

Task 4: Identify the most common pizza size ordered.

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

Task 6: Join the necessary tables to find the total quantity of each pizza category ordered.

Task 7: Determine the distribution of orders by hour of the day.

Task 8: Join relevant tables to find the category-wise distribution of pizzas.

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

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

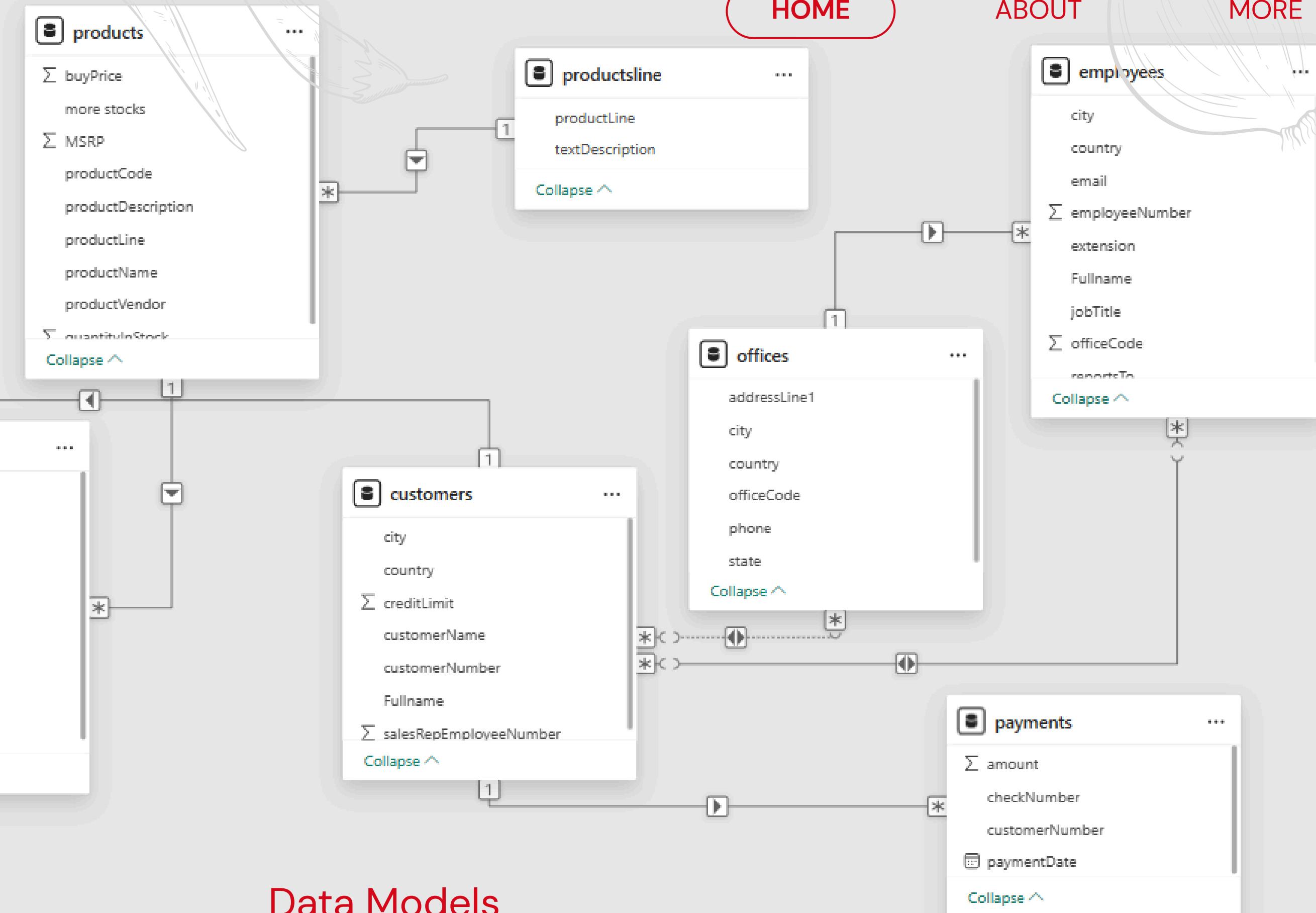
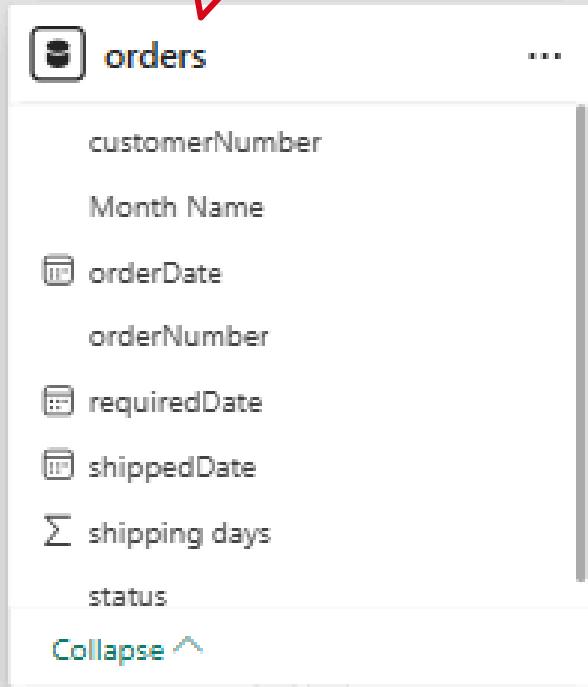
Task 11: Calculate the percentage contribution of each pizza type to total revenue.

Task 12: Analyze the cumulative revenue generated over time.

Task 13: Determine the top 3 most ordered pizza types based on revenue for each pizza category.



PIZZA STORE



HOME

ABOUT

MORE

Data Models



## Task 1: Retrieve the total number of orders placed.

```
1      -- Retrieve the total number of orders placed.  
2  
3 •  SELECT  
4      COUNT(order_id) AS total_orders  
5  FROM  
6      orders;
```

Result Grid	
	total_orders
▶	21350



## Task 2: Calculate the total revenue generated from pizza sales.

```
1  -- Calculate the total revenue generated from pizza sales.  
2 •  SELECT  
3   ROUND(SUM(order_details.quantity * pizzas.price),  
4        2) AS total_revenue  
5  FROM  
6    order_details  
7  JOIN  
8    pizzas ON order_details.pizza_id = pizzas.pizza_id;
```

	total_revenue
▶	817860.05



## Task 3: Identify the highest-priced pizza.

```
1  -- Identify the highest-priced pizza.  
2  
3 • SELECT  
4      pizza_types.name, pizzas.price  
5  FROM  
6      pizza_types  
7      JOIN  
8      pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id  
9  ORDER BY pizzas.price DESC  
10 LIMIT 1;
```

Result Grid | Filter Rows:

	name	price
▶	The Greek Pizza	35.95



## Task 4: Identify the most common pizza size ordered.

```
1 -- Identify the most common pizza size ordered.  
2  
3 • SELECT  
4     pizzas.size,  
5     COUNT(order_details.order_details_id) AS order_count  
6   FROM  
7     pizzas  
8   JOIN  
9     order_details ON pizzas.pizza_id = order_details.Pizza_id  
10  GROUP BY pizzas.size  
11  ORDER BY Order_count DESC;
```

Result Grid		
	size	order_count
▶	L	18526
	M	15385
	S	14137
	XL	544
	XXL	28



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

```
1  -- List the top 5 most ordered pizza types along with their quantities.
2 • SELECT
3      pizza_types.name, SUM(order_details.quantity) AS quantity
4  FROM
5      pizza_types
6      JOIN
7      pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
8      JOIN
9      order_details ON order_details.pizza_id = pizzas.pizza_id
10 GROUP BY pizza_types.name
11 ORDER BY quantity DESC
12 LIMIT 5;
13
```

Result Grid		
	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



## Task 6: Join the necessary tables to find the total quantity of each pizza category ordered.

```
1  -- Join the necessary tables to find the total quantity of each pizza category ordered
2 • SELECT
3      SUM(order_details.quantity) AS total_quantity,
4      pizza_types.category
5  FROM
6      order_details
7      JOIN
8      pizzas ON order_details.pizza_id = pizzas.pizza_id
9      JOIN
10     pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_id
11    GROUP BY pizza_types.category
12   ORDER BY total_quantity DESC;
```

Result Grid | Filter Rows:

	total_quantity	category
▶	14888	Classic
	11987	Supreme
	11649	Veggie
	11050	Chicken



## Task 7: Determine the distribution of orders by hour of the day.

```
1    -- Determine the distribution of orders by hour of the day.  
2  
3 • SELECT  
4      HOUR(order_time), COUNT(order_id)  
5  FROM  
6    orders  
7 GROUP BY HOUR(order_time);
```

hour (order_time)	count(order_id)
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



## Task 8: Join relevant tables to find the category-wise distribution of pizzas.

```
1  -- Join relevant tables to find the category-wise distribution of pizzas.  
2  
3  SELECT  
4      category, COUNT(name)  
5  FROM  
6      pizza_types  
7  GROUP BY category;
```

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



## Task 9 : Group the orders by date and calculate the average number of pizzas ordered per day.

```
1  -- Group the orders by date and calculate the average number of pizzas ordered per day.  
2 • SELECT  
3      ROUND(AVG(quantity), 0) AS avg_pizzas_ordered_per_day  
4  FROM  
5  (SELECT  
6      orders.order_date, SUM(order_details.quantity) AS quantity  
7  FROM  
8      orders  
9  JOIN order_details ON orders.order_id = order_details.order_id  
10 GROUP BY orders.order_date) AS order_quantity;  
11
```

Result Grid	
	avg_pizzas_ordered_per_day
▶	138



## Task 10: Determine the top 3 most ordered pizza types based on revenue.

```
1 -- Determine the top 3 most ordered pizza types based on revenue
2 • SELECT
3     pizza_types.name,
4     SUM(order_details.quantity * pizzas.price) AS revenue
5 FROM
6     pizza_types
7     JOIN
8     pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
9     JOIN
10    order_details ON order_details.pizza_id = pizzas.pizza_id
11 GROUP BY pizza_types.name
12 ORDER BY revenue DESC
13 LIMIT 3;
```

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



## Task 11: Calculate the percentage contribution of each pizza type to total revenue.

```
1  -- Calculate the percentage contribution of each pizza type to total revenue
2 •  select pizza_types.category,
3   round(sum(order_details.quantity * pizzas.price) / (select round(sum(order_details.quantity * pizzas.price), 2) as total_sales
4   from order_details join pizzas
5   on pizzas.pizza_id = order_details.pizza_id) * 100,2) as revenue
6   from pizza_types join pizzas
7   on pizza_types.pizza_type_id = pizzas.pizza_type_id
8   join order_details on
9   order_details.pizza_id = pizzas.pizza_id
10  group by pizza_types.category order by revenue desc;
11
```

Result Grid | Filter

	category	revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68



## Task 12: Analyze the cumulative revenue generated over time.

```
1  -- Analyze the cumulative revenue generated over time
2 • select order_date, sum(revenue) over(order by order_date) as cum_revenue
3  from
4  (select orders.order_date,
5    sum(order_details.quantity * pizzas.price) as revenue
6    from order_details join pizzas
7    on order_details.pizza_id = pizzas.pizza_id
8    join orders
9    on orders.order_id = order_details.order_id
10   group by orders.order_date) as sales;
```

Result Grid		
	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



## Task 13: Determine the top 3 most ordered pizza types based on revenue for each pizza category.

```
1  -- Determine the top 3 most ordered pizza types based on revenue for each pizza category
2 • select category, name, revenue,
3   rank() over(partition by category order by revenue desc) as rn
4   from
5   (select pizza_types.category, pizza_types.name,
6    sum(order_details.quantity * pizzas.price) as revenue
7    from pizza_types join pizzas
8    on pizza_types.pizza_type_id = pizzas.pizza_type_id
9    join order_details on
10   order_details.pizza_id = pizzas.pizza_id
11   group by pizza_types.category, pizza_types.name) as a
12   order by revenue desc
13   limit 3;
```

	category	name	revenue	rn
▶	Chicken	The Thai Chicken Pizza	43434.25	1
	Chicken	The Barbecue Chicken Pizza	42768	2
	Chicken	The California Chicken Pizza	41409.5	3



PIZZA STORE

HOME

ABOUT

MORE

# Thank you!

vijayeshkumar.github.io