

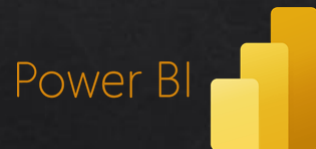
PIZZA SALES REPORT

— *Life's short, eat more pizza*

BY SPOORTHY M



&



CONTENTS

- Problem Statement
- Tables
- SQL Codes



PROBLEM STATEMENT

- 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.
- Join the relevant tables to find the total quantity of each pizza category ordered.
- Determine the distribution of orders by hour of the day.



PROBLEM STATEMENT

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



TABLES

▼	orders
▼	Columns
◆	order_id
◆	order_date
◆	order_time

▼	pizza_types
▼	Columns
◆	pizza_type_id
◆	name
◆	category
◆	ingredients

▼	orders_details
▼	Columns
◆	order_details_id
◆	order_id
◆	pizza_id
◆	quantity

▼	pizzas
▼	Columns
◆	pizza_id
◆	pizza_type_id
◆	size
◆	price



1. Retrieve the total number of orders placed.

```
1  -- Retrieve the total number of orders placed.  
2  
3  •  select count(distinct order_id) as total_orders from orders;
```

Result Grid



Filter Rows:

Export:



Wrap Cell Content:



	total_orders
▶	21350



2. Calculate the total revenue generated from pizza sales.

```
1  -- Calculate the total revenue generated from pizza sales.  
2  
3  • select  
4  round(sum(orders_details.quantity * pizzas.price), 2) as total_sales  
5  from orders_details join pizzas  
6  on pizzas.pizza_id = orders_details.pizza_id;  
7
```

Result Grid



Filter Rows:

Export:



Wrap Cell Content:



	total_sales
▶	817860.05



3. Identify the highest-priced pizza.

```
1  -- Identify the highest-priced pizza.  
2  
3  • select pizza_types.name, pizzas.price  
4    from pizza_types join pizzas  
5    on pizza_types.pizza_type_id = pizzas.pizza_type_id  
6    order by pizzas.price desc limit 1;
```

Result Grid			Filter Rows: <input type="text"/>	Export: 	Wrap Cell Content: 
	name	price			
▶	The Greek Pizza	35.95			



4. Identify the most common pizza size ordered.

```
1  -- Identify the most common pizza size ordered.  
2  
3  • select pizzas.size, count(orders_details.order_details_id) as order_count  
4  from pizzas join orders_details  
5  on pizzas.pizza_id = orders_details.pizza_id  
6  group by pizzas.size order by order_count desc limit 1;
```

Result Grid |   Filter Rows: | Export:  | Wrap Cell Content: 

	size	order_count
▶	L	18526



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
3  • select pizza_types.name,
4     sum(orders_details.quantity) as quantity
5  from pizza_types join pizzas
6  on pizza_types.pizza_type_id = pizzas.pizza_type_id
7  join orders_details
8  on orders_details.pizza_id = pizzas.pizza_id
9  group by pizza_types.name order by quantity desc limit 5;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	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. Join the relevant 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
3  • select pizza_types.category,
4     sum(orders_details.quantity) as quantity
5  from pizza_types join pizzas
6  on pizza_types.pizza_type_id = pizzas.pizza_type_id
7  join orders_details
8  on orders_details.pizza_id = pizzas.pizza_id
9  group by pizza_types.category order by quantity desc;
```

Result Grid   Filter Rows: Export:  Wrap Cell Content: 

	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050



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 hour(order_time) as hour, count(order_id) as order_count from orders
4  group by hour(order_time);
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

	hour	order_count
▶	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



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 category, count(distinct pizza_type_id) as no_of_pizzas  
4    from pizza_types  
5    group by category order by no_of_pizzas;
```

Result Grid



Filter Rows:

Export:



Wrap Cell Content:



	category	no_of_pizzas
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9



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
3  with cte as (
4      select orders.order_date as date,
5             sum(orders_details.quantity) as total_pizzas_ordered_per_day
6      from orders_details join orders
7      on orders_details.order_id = orders.order_id
8      group by orders.order_date
9  )
10 select avg(total_pizzas_ordered_per_day) as average_pizzas_ordered_per_day from cte;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	average_pizzas_ordered_per_day
▶	138.4749



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

```
1  -- Determine the top 3 most ordered pizza types based on revenue.
2
3  • select pizza_types.name,
4     sum(orders_details.quantity * pizzas.price) as revenue
5  from pizza_types join pizzas
6  on pizza_types.pizza_type_id = pizzas.pizza_type_id
7  join orders_details
8  on orders_details.pizza_id = pizzas.pizza_id
9  group by pizza_types.name order by revenue desc limit 3;
```

Result Grid



Filter Rows:

Export:



Wrap Cell Content:



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



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

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

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	category	revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68



Result
Grid



Form
Editor



12. Analyze the cumulative revenue generated over time.

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

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

	order_date	cumulative_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
	2015-01-13	29831.300000000003
	2015-01-14	32358.700000000004
	2015-01-15	34343.50000000001
	2015-01-16	36937.65000000001



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
3  • select category, name, revenue from
4  (select category, name, revenue,
5   rank() over(partition by category order by revenue desc) as rn
6   from
7   (select pizza_types.category, pizza_types.name,
8    sum(orders_details.quantity * pizzas.price) as revenue
9    from pizza_types join pizzas
10   on pizza_types.pizza_type_id = pizzas.pizza_type_id
11   join orders_details
12   on orders_details.pizza_id = pizzas.pizza_id
13   group by pizza_types.category, pizza_types.name) as a) as b
14  where rn <= 3;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

	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
	Supreme	The Spicy Italian Pizza	34831.25
	Supreme	The Italian Supreme Pizza	33476.75
	Supreme	The Sicilian Pizza	30940.5
	Veggie	The Four Cheese Pizza	32265.70000000065
	Veggie	The Mexicana Pizza	26780.75
	Veggie	The Five Cheese Pizza	26066.5



THANK YOU

