

PIZZERIA COMPANY

Pizza Sales Growth Project

EXPLORING SALES TRENDS USING SQL
-BY SMITHA NAIR



Project Brief

The Pizza Sales Analysis project aims to provide insights into the sales data of a pizza restaurant chain using SQL

Objective

Analyze sales trends and identify top-selling pizzas

Evaluate customer preferences based on pizza categories and sizes

Calculating key performance metric like average pizzas orders per day

Schema

Database Name : Pizzeria

Table: orders

Columns:

<u>order_id</u>	int PK
order_date	datetime
order_time	time

Table: order_details

Columns:

<u>order_details_id</u>	int PK
order_id	int
pizza_id	text
quantity	int

Table: pizzas

Columns:

pizza_id	text
pizza_type_id	text
size	text
price	double


Table: pizza_types

Columns:

pizza_type_id	text
name	text
category	text
ingredients	text

Retrieve the total number of orders placed.

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

Result Grid 	
	Total_orders
▶	21350

Calculate the total revenue generated from pizza sales.

```
• SELECT
    ROUND(SUM(order_details.quantity * pizzas.price),
          2) AS Total_price
FROM
    order_details
    JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id;
```

Result Grid	
	Total_price
▶	817860.05

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(order_details.order_details_id) AS total_count
FROM
    pizzas
    JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY total_count DESC;
```

Result Grid				Fit
	size	total_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 as Pizza_Name, SUM(order_details.quantity) AS Total_count
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_Name
ORDER BY Total_count DESC limit 5;
```

Result Grid			Filter Rows:
	Pizza_Name	Total_count	
▶	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(order_details.quantity) AS total_count
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 total_count desc;
```

Result Grid			Filter Rows
	category	total_count	
▶	Classic	14888	
	Supreme	11987	
	Veggie	11649	
	Chicken	11050	

Determine the distribution of orders by hour of the day

```
SELECT
    hour(orders.order_time) AS order_hour,
    COUNT(orders.order_id) AS order_count
FROM orders
GROUP BY hour(order_time)
order by order_count;
```

Result Grid  Filter Rows		
	order_hour	order_count
	9	1
	10	8
	23	28
	22	663
	21	1198
	11	1231
	15	1468
	14	1472
	20	1642
	16	1920
	19	2009
	17	2336
	18	2399
	13	2455
	12	2520

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

```
SELECT
    pizza_types.category, COUNT(order_details.order_id) as Total_Count
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;
```

Result Grid			Filter Rows
	category	Total_Count	
▶	Classic	14579	
	Veggie	11449	
	Supreme	11777	
	Chicken	10815	

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

```
SELECT
    DATE(orders.order_date) AS Order_Date,
    ROUND(AVG(order_details.quantity), 2)
FROM
    orders
    JOIN
    order_details ON orders.order_id = order_details.order_id
GROUP BY Order_Date;
```

Result Grid			Filter Rows:	Es
	Order_Date	ROUND(AVG(order_details.quantity), 2)		
▶	2015-01-01	1.01		
	2015-01-02	1.03		
	2015-01-03	1.03		
	2015-01-04	1.00		
	2015-01-05	1.03		
	2015-01-06	1.02		
	2015-01-07	1.04		
	2015-01-08	1.01		
	2015-01-09	1.03		
	2015-01-10	1.01		
	2015-01-11	1.02		
	2015-01-12	1.01		
	2015-01-13	1.03		
	2015-01-14	1.04		
	2015-01-15	1.00		
	2015-01-16	1.02		
	2015-01-17	1.02		
	2015-01-18	1.03		
	2015-01-19	1.02		

Determine the top 3 most ordered pizza types based on revenue

```
SELECT
    pizza_types.name as Pizza_Name,
    SUM(order_details.quantity * pizzas.price) AS total_price
FROM
    order_details
    JOIN
    pizzas ON order_details.pizza_id=pizzas.pizza_id
    JOIN
    pizza_types ON pizza_types.pizza_type_id =pizzas.pizza_type_id
group by Pizza_Name
order by total_price desc
LIMIT 3;
```

Result Grid			Filter Rows:
	Pizza_Name	total_price	
▶	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 as Pizza_Category,
    round(SUM(order_details.quantity * pizzas.price)/ (SELECT
    ROUND(SUM(order_details.quantity * pizzas.price),
        2) AS Total_price
FROM
    order_details
    JOIN
        pizzas ON order_details.pizza_id = pizzas.pizza_id)*100,2) as percent_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_Category
    order by percent_revenue desc
;
```

Result Grid			Filter Rows:
	Pizza_Category	percent_revenue	
▶	Classic	26.91	
	Supreme	25.46	
	Chicken	23.96	
	Veggie	23.68	

Analyze the cumulative revenue generated over time

```
Select order_date, sum(Total_Revenue)
over(order by order_date) as Cumulative_Revenue from
(SELECT
    orders.order_date,
    SUM(order_details.quantity * pizzas.price) as Total_Revenue
FROM
    order_details JOIN
    orders ON order_details.order_id=orders.order_id
    JOIN pizzas ON
    pizzas.pizza_id=order_details.pizza_id
group by orders.order_date) as sales;
```

Result Grid	Filter Rows:
order_date	Cumulative_Revenue
2015-01-01 00:00:00	2713.8500000000004
2015-01-02 00:00:00	5445.75
2015-01-03 00:00:00	8108.15
2015-01-04 00:00:00	9863.6
2015-01-05 00:00:00	11929.55
2015-01-06 00:00:00	14358.5
2015-01-07 00:00:00	16560.7
2015-01-08 00:00:00	19399.05
2015-01-09 00:00:00	21526.4
2015-01-10 00:00:00	23990.350000000002
2015-01-11 00:00:00	25862.65
2015-01-12 00:00:00	27781.7
2015-01-13 00:00:00	29831.300000000003
2015-01-14 00:00:00	32358.700000000004
2015-01-15 00:00:00	34343.500000000001
2015-01-16 00:00:00	36937.650000000001
2015-01-17 00:00:00	39001.750000000001
2015-01-18 00:00:00	40978.600000000006
2015-01-19 00:00:00	43365.750000000001

Result Grid	Filter Rows:
order_date	Cumulative_Revenue
2015-12-12 00:00:00	780971.8
2015-12-13 00:00:00	783216.9500000001
2015-12-14 00:00:00	785389.55
2015-12-15 00:00:00	787777
2015-12-16 00:00:00	790011.8
2015-12-17 00:00:00	791892.55
2015-12-18 00:00:00	794778.8500000001
2015-12-19 00:00:00	797083.05
2015-12-20 00:00:00	799187.9500000001
2015-12-21 00:00:00	801288.65
2015-12-22 00:00:00	803171.6
2015-12-23 00:00:00	805415.9
2015-12-24 00:00:00	807553.75
2015-12-26 00:00:00	809196.8
2015-12-27 00:00:00	810615.8
2015-12-28 00:00:00	812253
2015-12-29 00:00:00	813606.25
2015-12-30 00:00:00	814944.05
2015-12-31 00:00:00	817860.05

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

```
select category,name,Total_Price from
(select category,name,Total_price,
rank() over(partition by category order by Total_price desc) as Table2
FROM
(SELECT
    pizza_types.category,
    pizza_types.name,
    SUM(order_details.quantity * pizzas.price) AS Total_price
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 Table1) as Ranking
where Table2<=3;
```

category	name	Total_Price
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

