



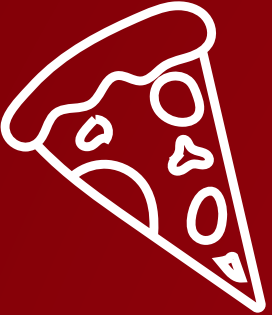
Pizza Sales Data Analysis Using SQL

End-to-End SQL Project

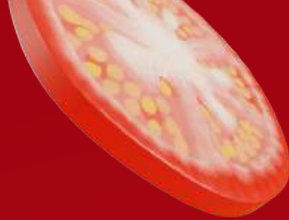
NAME : Yash Yadav

GMAIL : yashkosalia@gmail.com

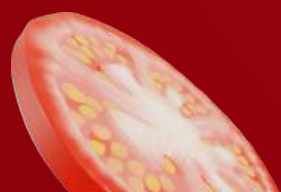




PROJECT OVERVIEW



- Dataset : Orders, Order Details, Pizzas, Pizza Types
- Objective : Derive sales, customer behavior, and revenue insights using SQL.
- Tools : MySQL Workbench



QUERIES COVERED



Basic:

Orders, Revenue, Highest-priced pizza,
Most common size, Top 5 pizzas

Intermediate:

Category analysis, Hourly trends, Daily
averages, Revenue ranking

Advanced:

Revenue contribution %, Cumulative
growth, Top pizzas per category



Basic Queries

1. Retrieve the total number of orders placed.

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

Result Grid		Filter Rows
	total_orders	
▶	21350	

Basic Queries

2. Calculate the total revenue generated from pizza sales.

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

Result Grid	
	total_revenue
▶	817860.05

Basic Queries

3. 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	

Basic Queries

4. Identify the most common pizza size ordered.

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

Result Grid			Filter Rows:
	size	total_orders	
▶	L	18526	

Basic Queries

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

```
SELECT
    pizza_types.name, SUM(order_details.quantity) AS quantity
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.name
ORDER BY quantity DESC
LIMIT 5;
```

Result Grid			Filter Rows:
	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	

Intermediate Queries

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

```
SELECT
    pizza_types.category,
    SUM(order_details.quantity) AS quantity
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 quantity DESC;
```

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

Intermediate Queries

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

```
SELECT
    HOUR(order_time) AS o_time, COUNT(order_id) AS order_count
FROM
    orders
GROUP BY o_time;
```

Result Grid			Filter Rows:
	o_time	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	

Intermediate Queries

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

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

Result Grid			Filter Rows:
	category	COUNT(name)	
▶	Chicken	6	
	Classic	8	
	Supreme	9	
	Veggie	9	

Intermediate Queries

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

```
SELECT
    ROUND(AVG(quantity), 0) as avg_orders_per_day
FROM
    (SELECT
        orders.order_date, SUM(order_details.quantity) AS quantity
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY orders.order_date) AS order_quantity;
```

Result Grid		Filter Rows:
	avg_orders_per_day	
▶	138	

Intermediate Queries

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

```
SELECT
    pizza_types.name,
    SUM(order_details.quantity * pizzas.price) AS 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_types.name
ORDER BY revenue DESC
LIMIT 3;
```

Result Grid			Filter Rows:
	name	revenue	
▶	The Greek Pizza	5450661.3000048855	
	The Italian Vegetables Pizza	2503487	
	The Thai Chicken Pizza	2491093.5	

Advanced Queries

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

```
SELECT
    pizza_types.category,
    SUM(order_details.quantity * pizzas.price) / (SELECT
        SUM(order_details.quantity * pizzas.price) AS total_sales
    FROM
        order_details
        JOIN
            pizzas ON order_details.pizza_id = pizzas.pizza_id) * 100 AS 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_types.category
ORDER BY revenue DESC;
```

	category	revenue
▶	Classic	26.905960255669903
	Supreme	25.45631126009884
	Chicken	23.955137556847493
	Veggie	23.682590927384783

Advanced Queries

12. 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(order_details.quantity*pizzas.price) as revenue
from
order_details join pizzas
on order_details.pizza_id=pizzas.pizza_id
join orders on
orders.order_id=order_details.order_id
group by orders.order_date) as sales;
```

Result Grid	Filter Rows:
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
2015-01-13	29831.300000000003
2015-01-14	32358.700000000004
2015-01-15	34343.50000000001
2015-01-16	36937.65000000001
2015-01-17	39001.75000000001
2015-01-18	40978.600000000006
2015-01-19	43365.75000000001
2015-01-20	45763.65000000001
2015-01-21	47804.20000000001
2015-01-22	50300.90000000001

Advanced Queries

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

```
select name, revenue, rnk, category from
(select category, name, revenue, rank() over
(partition by category order by revenue desc) as rnk
from
(select pizza_types.category, pizza_types.name,
sum(order_details.quantity*pizzas.price) as 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_types.category, pizza_types.name) as a) as b
where rnk <=3;
```

Result Grid

Filter Rows:

Export:

Wrap

	name	revenue	rnk	category
▶	The Thai Chicken Pizza	43434.25	1	Chicken
	The Barbecue Chicken Pizza	42768	2	Chicken
	The California Chicken Pizza	41409.5	3	Chicken
	The Classic Deluxe Pizza	38180.5	1	Classic
	The Hawaiian Pizza	32273.25	2	Classic
	The Pepperoni Pizza	30161.75	3	Classic
	The Spicy Italian Pizza	34831.25	1	Supreme
	The Italian Supreme Pizza	33476.75	2	Supreme
	The Sicilian Pizza	30940.5	3	Supreme
	The Four Cheese Pizza	32265.70000000065	1	Veggie
	The Mexicana Pizza	26780.75	2	Veggie
	The Five Cheese Pizza	26066.5	3	Veggie



Overview

- Date range: 2015-01-01 → 2015-12-31
(365 days)
- Total orders: 21,350
- Total revenue: \$817,860.05
- Average pizzas per day: 138
- Most common size: (L)
- Highest-priced pizza: The Greek (XXL)
— \$35.95



Top 5 pizzas by quantity

1. The Classic Deluxe — 2,453
2. The Barbecue Chicken — 2,432
3. The Hawaiian — 2,422
4. The Pepperoni — 2,418
5. The Thai Chicken — 2,371

Category totals (by quantity).

- Classic: 14,888
- Supreme: 11,987
- Veggie: 11,649
- Chicken: 11,050



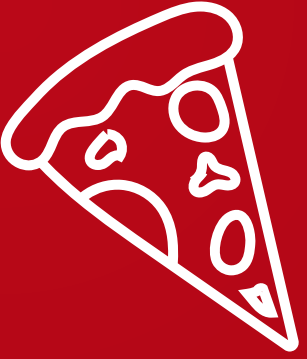
Peak order hours (by order count)

- 12:00 → 2,520 (highest)
- 13:00 → 2,455
- 18:00 → 2,399
- (Lowest: 23:00 → 28)

Top 3 pizzas by revenue

- The Thai Chicken — \$43,434.25
- The Barbecue Chicken — \$42,768.00
- The California Chicken — \$41,409.50





Business Recommendations

Staffing & Ops

- Focus staff at lunch (12–14h) & dinner (18h)
- Reduce late-night shifts

Menu & Inventory

- Stock more L-size pizzas
- Ensure Classic SKUs always available
- Maintain quality for Chicken pizzas (high revenue)

Pricing & Promotions

- Bundle Chicken bestsellers (Thai/BBQ/California)
- Launch lunch combos to boost ticket size
- Offer off-peak discounts (after 20h)

Merchandising

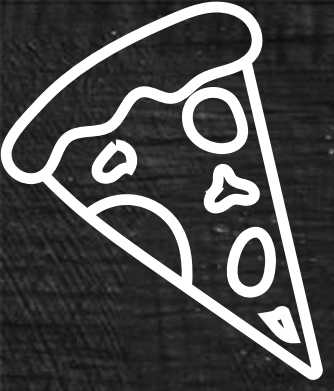
- Position The Greek (XXL) as premium item
- Highlight Classic Deluxe, Pepperoni, Hawaiian as fast movers



LEARNINGS & TOOLS

- **SQL concepts:** Joins, Aggregations, Window Functions, CTEs
- **Tools used:** MySQL Workbench
- **Outcome:** Turned raw data into business-driven insights





THANK YOU!

