

# DATA COLLECTION & ANALYSIS



FOR PIZZA SALES By PIZZABOX

BY-SANKALP SAMANT

# PROJECT OVERVIEW

## SALES DATA ANALYSIS

The comprehensive analysis of Pizza Box's sales data offers valuable insights into customer behavior, ordering patterns, and menu performance. Leveraging this data, we aim to drive business growth and make data-driven decisions.



# PROJECT OVERVIEW

## DATA INSIGHTS

The dataset includes detailed information about each transaction, allowing us to identify trends in pizza sales, such as which pizzas are most popular and when pizza sales are highest.



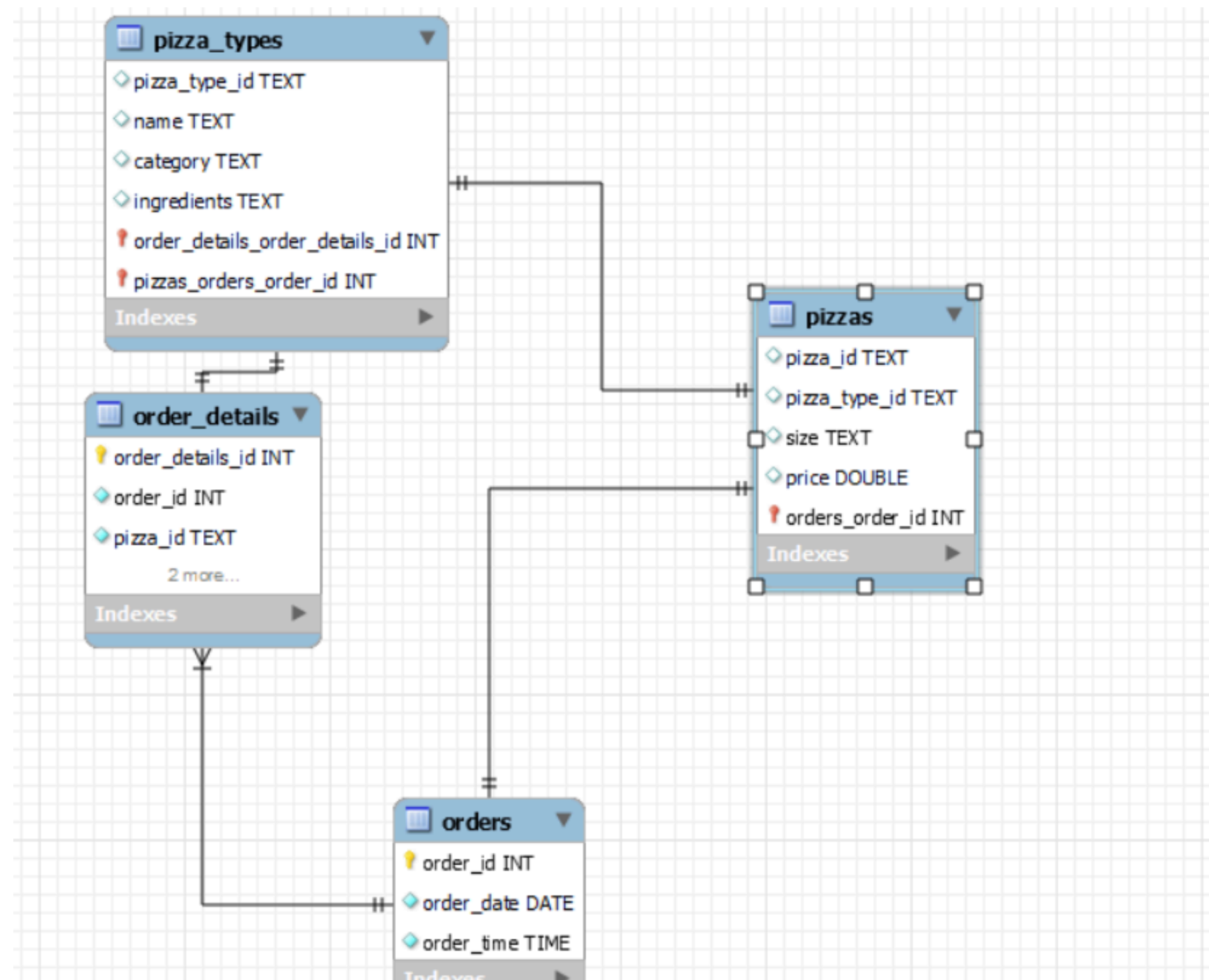
# PROJECT OVERVIEW

## BUSINESS OBJECTIVES

Our objective is to present the findings in a clear and concise manner, enabling the audience to grasp the key insights easily.



# SCHEMA OF THE DATABASE



# EDA: USING MYSQL

Retrieve the total number of orders placed.

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

OUTPUT:

Result Grid	
	total_orders
▶	21350

# EDA: USING MYSQL

Calculate the total revenue generated from pizza sales.

```
select round(sum(p.price* od.quantity),2) as revenue from pizzas p  
join order_details od on p.pizza_id=od.pizza_id;
```

OUTPUT:

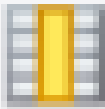

Result Grid	
	revenue
▶	817860.05

# EDA: USING MYSQL

Identify the highest-priced pizza.

```
select pt.name,p.price  
from pizza_types pt join pizzas p  
on pt.pizza_type_id=p.pizza_type_id  
order by p.price desc limit 1;
```

OUTPUT:

Result Grid     Filter Rows		
	name	price
▶	The Greek Pizza	35.95



# EDA: USING MYSQL

Identify the most common pizza size ordered.

```
select p.size, count(od.order_details_id) as order_count  
from pizzas p join order_details od  
on p.pizza_id=od.pizza_id  
group by p.size  
order by order_count desc  
limit 1;
```

OUTPUT:

Result Grid			Filter R
	size	order_count	
▶	L	18526	

# EDA: USING MYSQL

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;
```

OUTPUT:

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	

# EDA: USING MYSQL

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

```
select pt.category as category , sum(od.quantity)
from pizza_types pt join pizzas p
on pt.pizza_type_id=p.pizza_type_id
join order_details od
on od.pizza_id=p.pizza_id
group by category;
```

OUTPUT:

Result Grid			Filter Rows:
	category	sum(od.quantity)	
▶	Classic	14888	
	Veggie	11649	
	Supreme	11987	
	Chicken	11050	

# EDA: USING MYSQL

Determine the distribution of orders by hour of the day.

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

OUTPUT:

Result Grid			Filter Rows:
	hour	order_count	
	19	2009	
	16	1920	
	20	1642	
	14	1472	
	15	1468	
	11	1231	
	21	1198	
	22	663	
	23	28	
	10	8	
	9	1	

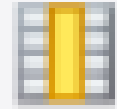


Result 4 ×

# EDA: USING MYSQL

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

```
select round(avg(quantity),0) as average_per_day_sales from
(select orders.order_date,
 sum(order_details.quantity) quantity
from orders join order_details
on orders.order_id = order_details.order_id
group by orders.order_date) as order_quantity;
```

OUTPUT:

Result Grid			 Filter Rows:
	average_per_day_sales		
	138		

# EDA: USING MYSQL

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 pizzas.pizza_type_id = pizza_types.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;
```

OUTPUT:



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

# EDA: USING MYSQL

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

```
SELECT pizza_types.category, ROUND(SUM(order_details.quantity * pizzas.price), 2) AS revenue,  
ROUND((SUM(order_details.quantity * pizzas.price) / SUM(SUM(order_details.quantity * pizzas.price)) OVER()), 2) * 100 AS percentage  
FROM pizza_types  
JOIN pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id  
JOIN order_details ON order_details.pizza_id = pizzas.pizza_id  
GROUP BY pizza_types.category  
ORDER BY revenue DESC;
```

OUTPUT:


Result Grid     Filter Rows: <input type="text"/>			
	category	revenue	percentage
▶	Classic	220053.1	27
	Supreme	208197	25
	Chicken	195919.5	24
	Veggie	193690.45	24

# EDA: USING MYSQL

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;
```

OUTPUT:

Result Grid    Filter Rows: <input type="text"/>		
	order_date	cum_revenue
▶	2015-01-01	2713.85000000000004
	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	23990.3500000000002
	2015-01-11	25862.65
	2015-01-12	27781.7
	2015-01-13	29831.3000000000003
	2015-01-14	32358.7000000000004





# EDA: USING MYSQL

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

```
select name,revenue from
(select category , name , revenue,
rank() over(partition by category order by revenue desc) as rn
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 rn<=3;
```

OUTPUT:

Result Grid     Filter Rows: <input type="text"/>   Export:		
	name	revenue
▶	The Thai Chicken Pizza	43434.25
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41409.5
	The Classic Deluxe Pizza	38180.5
	The Hawaiian Pizza	32273.25
	The Pepperoni Pizza	30161.75
	The Spicy Italian Pizza	34831.25
	The Italian Supreme Pizza	33476.75
	The Sicilian Pizza	30940.5
	The Four Cheese Pizza	32265.700000000065
	The Mexicana Pizza	26780.75
	The Five Cheese Pizza	26066.5

# KEY FINDINGS AND INSIGHTS

## POPULAR PIZZA CATEGORIES

### a. CUSTOMER PREFERENCES

Analysis of the data reveals the most popular pizza categories, shedding light on customer preferences and ordering patterns.

### b. MENU OPTIMIZATION

Understanding the popularity of specific pizza categories enables Pizza Box to optimize the menu, ensuring that it resonates with customer preferences.

# KEY FINDINGS AND INSIGHTS

## PEAK SALES TIME

### a. RESOURCE ALLOCATION

Identifying peak sales times allows Pizza Box to allocate resources effectively, ensuring optimal service during high-demand periods.

### b. OPERATIONAL EFFICIENCY

Understanding sales patterns throughout the day enables Pizza Box to streamline operations and enhance efficiency.



thank  
you

