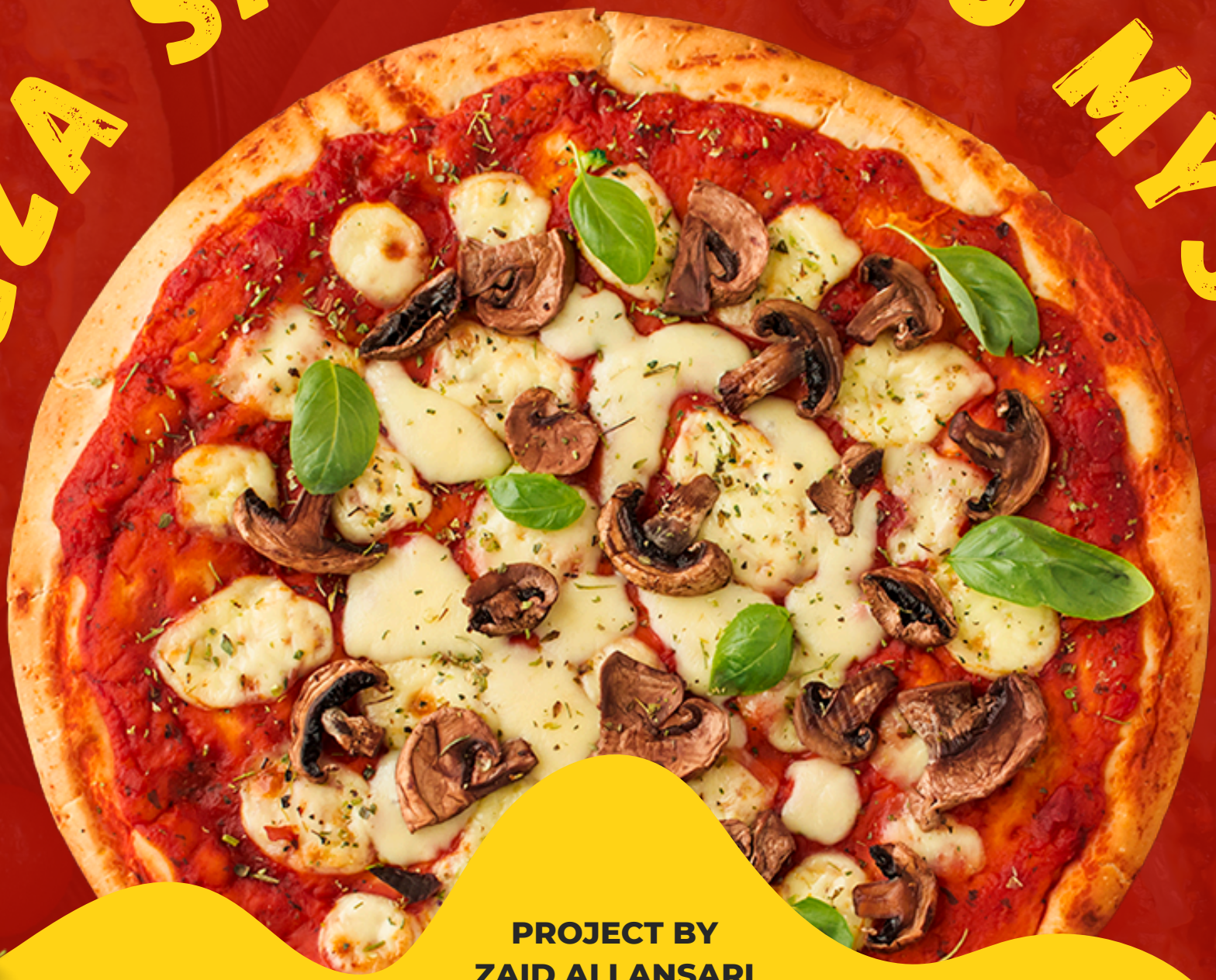
PIZZA SALES USING MYSQL



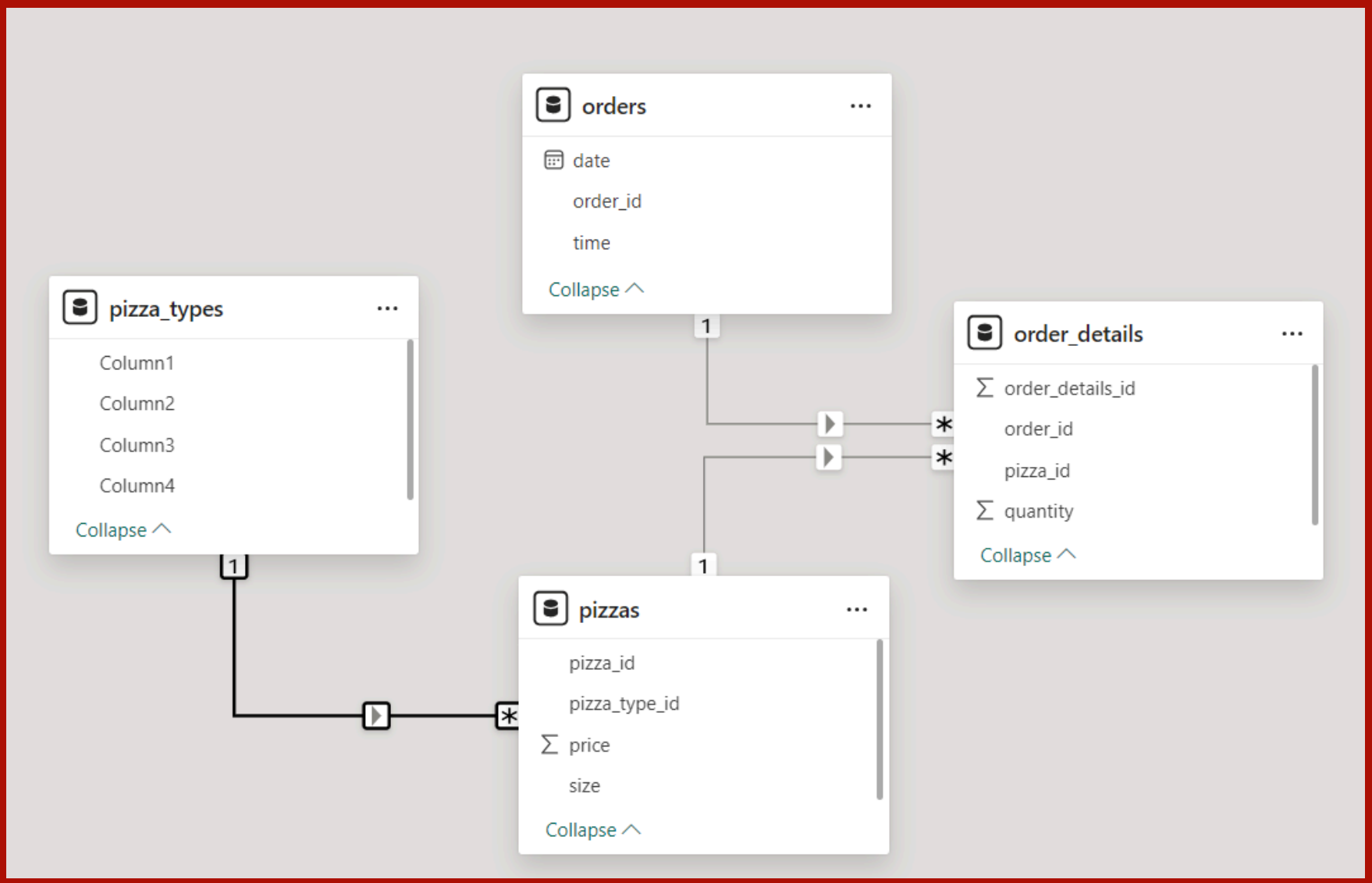
PROJECT BY
ZAID ALI ANSARI

This presentation contains pizza sales insights using MYSQL queries

Project Details

IN THIS ANALYSIS, WE UTILIZED MYSQL TO EXTRACT AND ANALYZE KEY INSIGHTS FROM OUR PIZZA SALES DATA. BY QUERYING THE SALES DATABASE, WE WERE ABLE TO IDENTIFY PATTERNS AND TRENDS THAT ARE CRUCIAL FOR OPTIMIZING OUR OPERATIONS AND BOOSTING REVENUE

Data model Relationships



Queries with output

1

```sql
Retrieve the total number of orders placed
ect count(order_id) as total_orders from orders;
```

	total_orders
▶	21350

Queries with output

2

```sql
-- Calculate the total revenue generated from pizza sales.

SELECT
    ROUND(SUM(order_details.quantity * pizzas.price),
            2) AS total_sales
FROM
    order_details
        JOIN
    pizzas USING (pizza_id);
```

	total_sales
▶	817860.05

Queries with output

3

```sql
-- Identify the highest-priced pizza.

select pizza_types.name,pizzas.price
 from pizza_types join pizzas using(pizza_type_id)
order by pizzas.price desc limit 1 ;
```

	name	price
▶	The Greek Pizza	35.95

Queries with output

4

```sql
-- Identify the most common pizza size ordered.

select pizzas.size,count(order_details.order_details_id) as order_count
from pizzas join order_details using(pizza_id)
group by  pizzas.size order by order_count desc limit 1 ;

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

	size	order_count
▶	L	18526

Queries with output

5

```sql
-- 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 using(pizza_type_id)
join order_details using(pizza_id)
group by pizza_types.name order by quantity  desc limit 5;
```

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

Queries with output

6

```sql
-- Join the necessary tables to find the total quantity of each pizza category ordered.
select  pizza_types.category , sum(order_details.quantity) as Total_quantity
from order_details join pizzas using(pizza_id)
join pizza_types using(pizza_type_id)
group by pizza_types.category order by total_quantity desc ;
```

category	Total_quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Queries with output

7

```sql
-- Determine the distribution of orders by hour of the day.
select hour(order_time) as Hour, count(order_id) as Orders from orders
group by  hour(order_time);
```

	Hour	Orders
▶	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

Queries with output

8

```sql
-- Join relevant tables to find the category-wise distribution of pizzas.
select category , count(name) from pizza_types
group by category;
```

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

Queries with output

9

```sql
-- Group the orders by date and calculate the average number of pizzas ordered per day.
select round(avg(quantity),0) from
(select orders.order_date , sum(order_details.quantity) as quantity
from orders join order_details using(order_id) group by order_date) as ordered_quantity ;
```

	round(avg(quantity),0)
▶	138

Queries with output

10

```sql
-- 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 using(pizza_type_id) join
order_details  using(pizza_id) group by pizza_types.name
order by revenue desc limit 3;
```

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

Queries with output

11

```sql
-- Calculate the percentage contribution of each pizza type to total revenue.
select pizza_types.category,round((sum(order_details.quantity*pizzas.price) /
 (select round(sum(order_details.quantity*pizzas.price),2) as
total_sales from order_details join pizzas using(pizza_id)))*100,2) as revenue
from pizza_types join pizzas using(pizza_type_id)
join order_details using(pizza_id)
group by pizza_types.category order by revenue desc;
```

	category	revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

Queries with output

12

```sql
-- 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 using(pizza_id) join
orders using(order_id)
group by orders.order_date) as sales;
```

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

Queries with output

13

```sql
-- Determine the top 3 most ordered pizza types based on revenue for each pizza category.
select category,name,revenue from
(select category,name,revenue,
rank() over(partition by category order by revenue desc) as ranking from
(select pizza_types.category,pizza_types.name,
sum(order_details.quantity*pizzas.price) as revenue from
pizza_types join pizzas using(pizza_type_id) join order_details using(pizza_id)
group by pizza_types.category,pizza_types.name) as a) as b
where ranking<= 3;
```

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

Conclusion

These findings are crucial for decision-making and can help the business optimize inventory management, tailor marketing strategies, and improve customer satisfaction. For example, understanding which pizzas are most popular can inform menu design, while identifying peak hours can assist in staffing decisions. Moreover, by recognizing patterns in customer behavior, the business can design targeted promotions and loyalty programs to drive sales during slower periods.

Overall, this project demonstrated the power of SQL in analyzing large datasets and extracting actionable insights. The conclusions drawn from this analysis will be instrumental in enhancing business operations and driving growth in a competitive market.

End of Project
Thank you