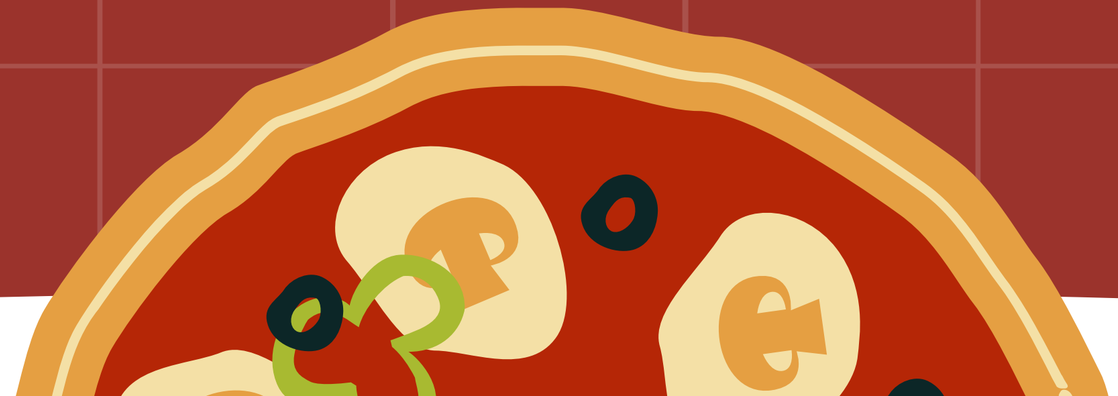
PIZZA SALES PROJECT

LIST OF CONTENTS

INTRODUCTION

My name is Alberto Sanchez and for this project, I used MYSQL Server 2019 to run the queries along with Excel to retrieve all the information.

Let's dive in!



RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

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

=

total_orders
21350

CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

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

=

total_sales
817860.05

IDENTIFY THE HIGHEST-PRICED PIZZA.

```sql
SELECT
    TOP 1 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
```

=

	name	price
1	The Greek Pizza	35.95

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

```sql
SELECT pizzas.size, COUNT(order_details.order_details_id) AS order_count
FROM pizzas JOIN order_details
    ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY order_count DESC
```

=

	size	order_count
1	L	18526
2	M	15385
3	S	14137
4	XL	544
5	XXL	28

LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES.

```sql
SELECT TOP 5
    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
```

=

	name	quantity
1	The Classic Deluxe Pizza	2453
2	The Barbecue Chicken Pizza	2432
3	The Hawaiian Pizza	2422
4	The Pepperoni Pizza	2418
5	The Thai Chicken Pizza	2371

JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED.

```sql
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
```

=

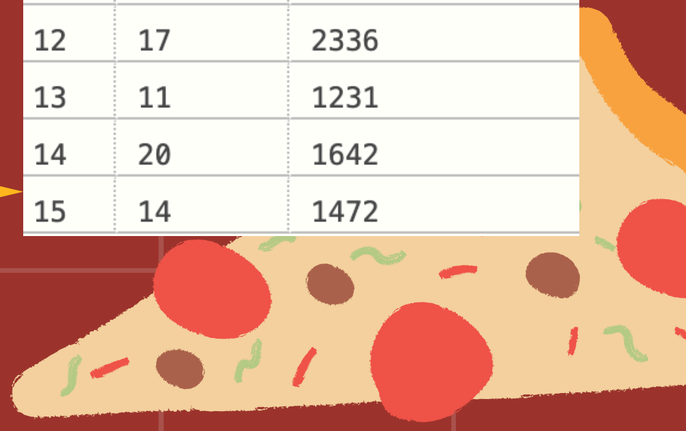
	category	quantity
1	Classic	14888
2	Supreme	11987
3	Veggie	11649
4	Chicken	11050

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

```sql
SELECT DATEPART(HH, [time]) AS hour, COUNT(order_id) AS order_count
FROM orders
GROUP BY DATEPART(HH, [time])
```

=

	hour	order_count
1	23	28
2	15	1468
3	9	1
4	12	2520
5	21	1198
6	18	2399
7	10	8
8	19	2009
9	13	2455
10	22	663
11	16	1920
12	17	2336
13	11	1231
14	20	1642
15	14	1472



JOIN RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS.

```sql
SELECT category, COUNT(name) AS count_category
FROM pizza_types
GROUP BY category
```

=

	category	count_category
1	Chicken	6
2	Classic	8
3	Supreme	9
4	Veggie	9

GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY.

```sql
SELECT AVG(quantity) AS avg_pizza_ordered_per_day
FROM
    (SELECT orders.date, SUM(order_details.quantity) AS quantity
    FROM orders JOIN order_details
        ON orders.order_id = order_details.order_id
    GROUP BY orders.date) AS order_quantity
```

=

	avg_pizza_ordered_per_day
1	138

DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.

```sql
SELECT TOP 3
    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
```

=

	name	revenue
1	The Thai Chicken Pizza	43434.25
2	The Barbecue Chicken Pizza	42768
3	The California Chicken Pizza	41409.5

CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE.

```sql
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 ON pizzas.pizza_id = order_details.pizza_id), 2) *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
```

=

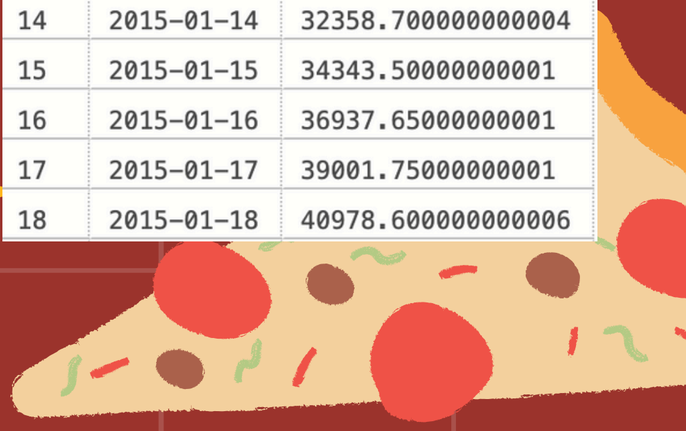
	category	revenue
1	Classic	27
2	Supreme	25
3	Veggie	24
4	Chicken	24

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

```sql
SELECT date, SUM(revenue) OVER (ORDER BY date) AS cum_revenue
FROM
    (SELECT orders.[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.[date]) AS sales
```

=

	date	cum_revenue
1	2015-01-01	2713.8500000000004
2	2015-01-02	5445.75
3	2015-01-03	8108.15
4	2015-01-04	9863.6
5	2015-01-05	11929.55
6	2015-01-06	14358.5
7	2015-01-07	16560.7
8	2015-01-08	19399.05
9	2015-01-09	21526.4
10	2015-01-10	23990.350000000002
11	2015-01-11	25862.65
12	2015-01-12	27781.7
13	2015-01-13	29831.300000000003
14	2015-01-14	32358.700000000004
15	2015-01-15	34343.50000000001
16	2015-01-16	36937.65000000001
17	2015-01-17	39001.75000000001
18	2015-01-18	40978.600000000006



DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY.

```sql
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
```

=

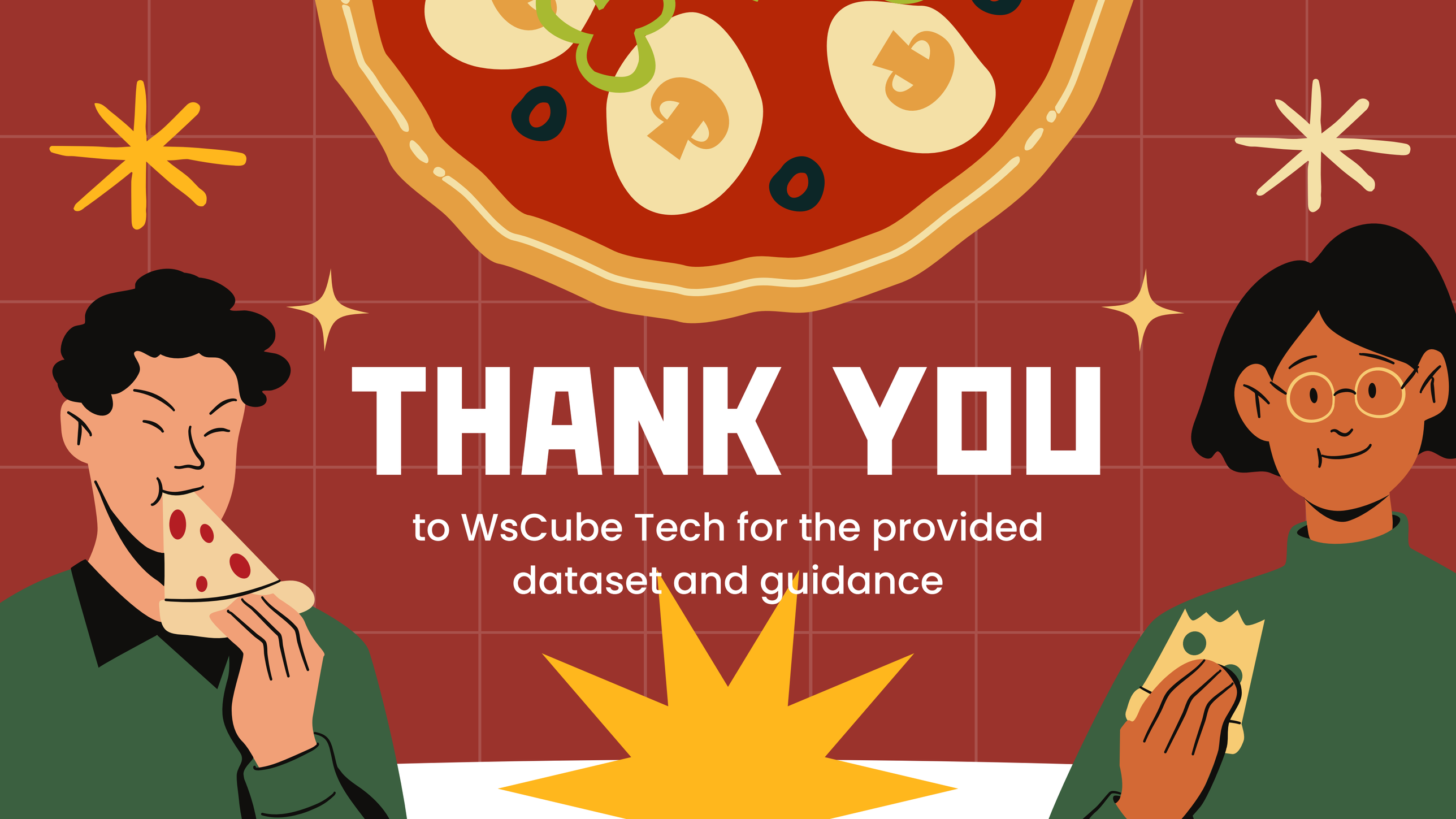
	name	revenue
1	The Thai Chicken Pizza	43434.25
2	The Barbecue Chicken Pizza	42768
3	The California Chicken Pizza	41409.5

CONCLUSION

THE RESULTS OF OUR SQL QUERIES HAVE PROVIDED VALUABLE INSIGHTS INTO OUR PIZZA SALES, DEMONSTRATING POSITIVE TRENDS IN OUR BUSINESS PERFORMANCE. BY ANALYZING THE TOTAL NUMBER OF ORDERS, REVENUE GENERATION, AND POPULAR PIZZA CHOICES, WE HAVE IDENTIFIED KEY AREAS OF SUCCESS AND POTENTIAL OPPORTUNITIES FOR GROWTH.

THESE QUERIES HAVE ENABLED US TO:

- Measure overall sales performance and revenue.
- Identify best-selling pizzas and their sizes.
- Understand customer preferences and ordering patterns.
- Gain insights into hourly and daily sales distributions.


THANK YOU
to WsCube Tech for the provided
dataset and guidance