PIZZA SALES ANALYSIS

using MySQL

DATA ANALYST: PUNIT PAL

Tools Used: MySQL Workbench

Techniques: Joins, Aggregations, Window Functions, Grouping, Ordering, Time-Series Analysis

punitpalofficial@gmail.com

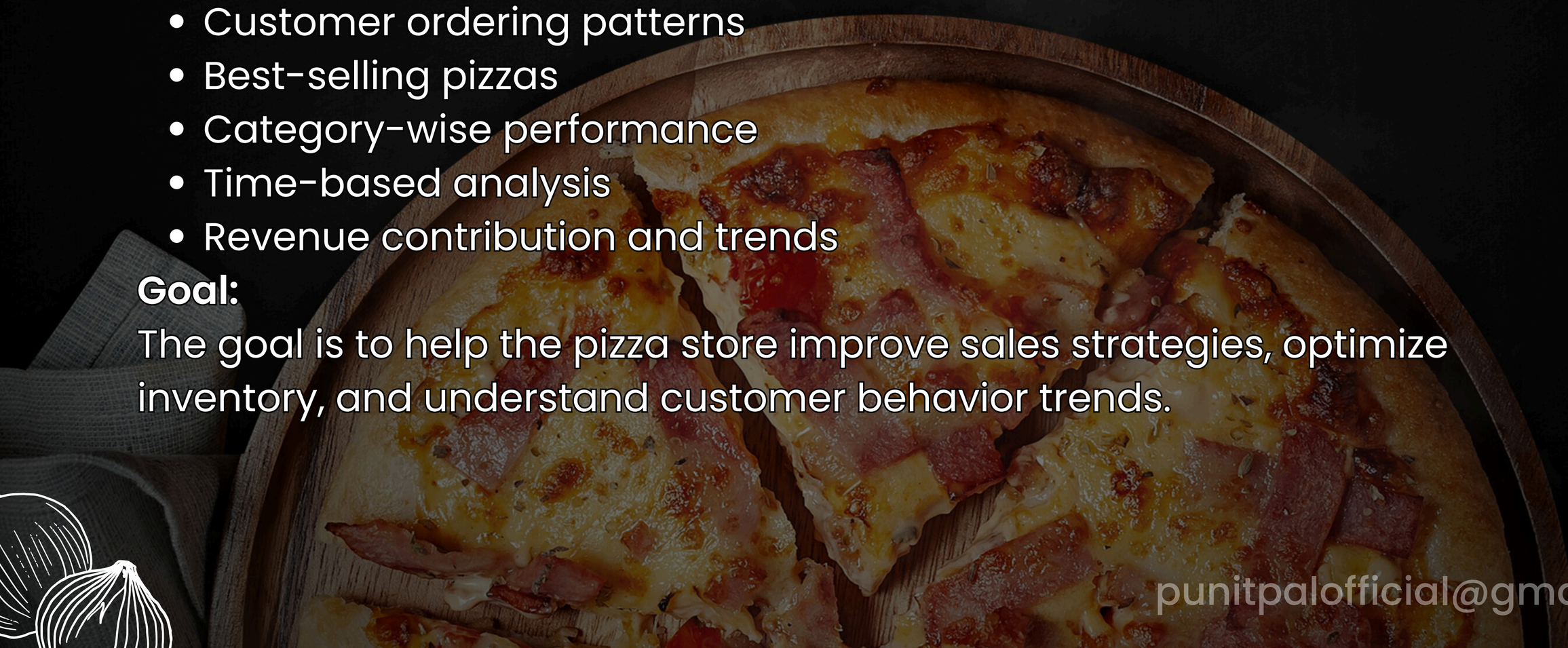
🔍 Project Overview:

This project analyzes real-world pizza sales data using MySQL.
It answers key business questions related to:

- Total orders
- Revenue
- Customer ordering patterns
- Best-selling pizzas
- Category-wise performance
- Time-based analysis
- Revenue contribution and trends

Goal:

The goal is to help the pizza store improve sales strategies, optimize inventory, and understand customer behavior trends.



punitpalofficial@gmail.com

Dataset Description:

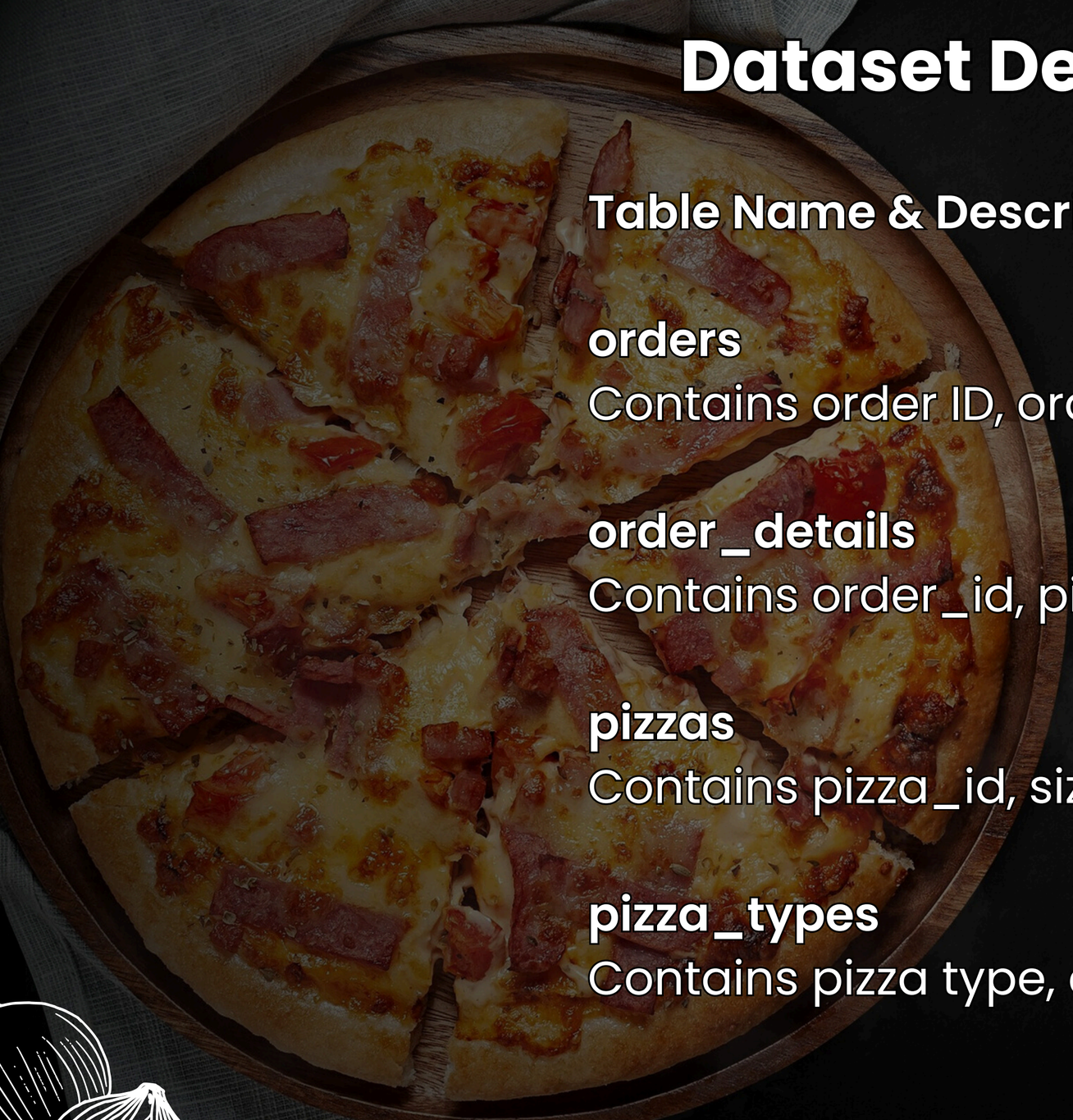


Table Name & Description:

orders
Contains order ID, order time/date

order_details
Contains order_id, pizza_id, quantity

pizzas
Contains pizza_id, size, price

pizza_types
Contains pizza type, category, name

punitpalofficial@gmail.com

🎯 Project Objectives:

This project aims to answer 3 levels of questions:

Basic Analysis

- Total orders
- Total revenue
- Most expensive pizza
- Most ordered pizza size
- Top 5 pizza types

Intermediate Analysis

- Category-wise quantity
- Hour-wise order distribution
- Category distribution
- Daily average orders
- Top 3 pizzas by revenue

Advanced Analysis

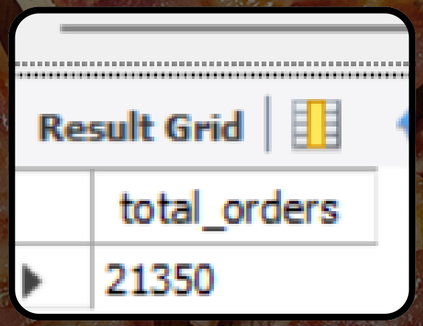
- Percentage contribution of each pizza type
- Cumulative revenue (time-series)
- Revenue-based top pizzas per category

punitpalofficial@gmail.com

Basic:

(1). Retrieve the total number of orders placed.

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



punitpalofficial@gmail.com

Basic:

(2).Calculate the total revenue generated from pizza sales.

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

Result Grid

	total_sales
▶	817860.05

punitpalofficial@gmail.com

Basic:

(3). Identify the highest-priced pizza.

```sql
SELECT
    name, price
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
ORDER BY pizzas.price DESC
LIMIT 1;
```

Result Grid | Filter Rows:

	name	price
▶	The Greek Pizza	35.95

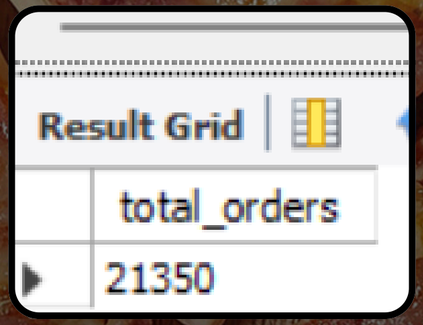
punitpalofficial@gmail.com

Basic:

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

★★

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



punitpalofficial@gmail.com

Basic:

(5). Identify the most common pizza size ordered.

```
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
LIMIT 1;
```

Result Grid | Filter Rows:

size	order_count
L	18526

punitpalofficial@gmail.com

Intermediate:

(6). 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 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 quantity DESC;
```

Result Grid | Filter Rows:

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

punitpalofficial@gmail.com

Intermediate:

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

```sql
SELECT
    HOUR(order_time) AS hours, COUNT(order_id) AS order_count
FROM
    orders
GROUP BY hours
ORDER BY order_count DESC;
```

Result Grid | Filter Rov

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

punitpalofficial@gmail.com

Intermediate:

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

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

Result Grid | Filter Rows:

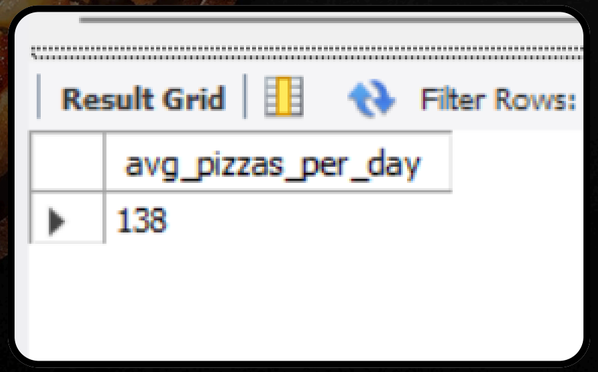
category	COUNT(pizza_type_id)
Chicken	6
Classic	8
Supreme	9
Veggie	9

punitpalofficial@gmail.com

Intermediate:

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

```sql
SELECT
    ROUND(AVG(quantity), 0) AS avg_pizzas_per_day
FROM
    (SELECT
        DATE(order_date) AS order_date, SUM(quantity) AS quantity
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY order_date
    ORDER BY quantity DESC) AS order_quantity;
```



punitpalofficial@gmail.com

Intermediate:

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

```sql
SELECT
    pizza_types.name AS 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 name
ORDER BY revenue DESC
LIMIT 3;
```

Result Grid | Filter Rows:

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

punitpalofficial@gmail.com

Advanced:

(11). 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
                    pizzas
                        JOIN
                    order_details ON order_details.pizza_id = pizzas.pizza_id)) * 100,
            2) 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 category
ORDER BY revenue DESC);
```

Result Grid | Filter Rows:

category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

punitpalofficial@gmail.com

Advanced

(12). Analyze the cumulative revenue generated over time.

```sql
with a as(select date(orders.order_date) as dates,sum(order_details.quantity*pizzas.price) as revenue
from order_details join pizzas
on pizzas.pizza_id=order_details.pizza_id
join orders
on orders.order_id=order_details.order_id
group by dates
order by dates asc)
select dates, revenue, sum(revenue) over(order by dates) as previous_day_payment from a;
```

Result Grid | Filter Rows: | Export: | Wrap

dates	revenue	previous_day_payment
2015-01-01	2713.8500000000004	2713.8500000000004
2015-01-02	2731.8999999999996	5445.75
2015-01-03	2662.3999999999996	8108.15
2015-01-04	1755.4500000000003	9863.6
2015-01-05	2065.95	11929.55

punitpalofficial@gmail.com

Advanced

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

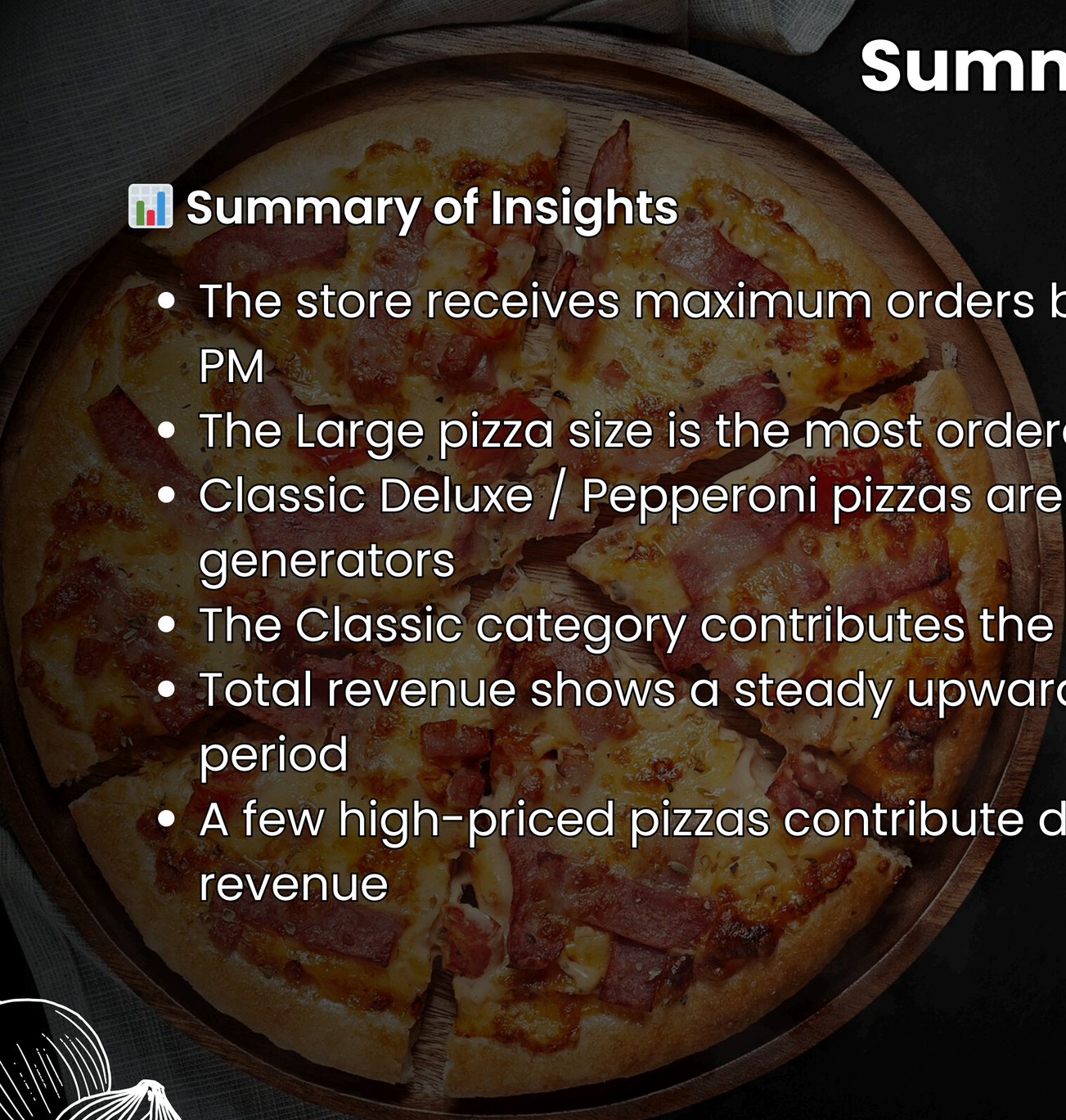
```sql
select category,name,revenue
from
(select category,name,revenue,
rank() over(partition by category order by revenue desc) as ranks
from
(select pizza_types.category,pizza_types.name,
sum(order_details.quantity*pizzas.price) as revenue
from pizza_types join pizzas
on pizza_types.pizza_type_id=pizzas.pizza_type_id
join order_details
on order_details.pizza_id=pizzas.pizza_id
group by pizza_types.name,pizza_types.category
order by revenue desc) as a) as b
where ranks<=3;
```

Result Grid | Filter Rows: | Export:

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

punitpalofficial@gmail.com

Summary:



📊 Summary of Insights

- The store receives maximum orders between 12 PM – 2 PM
- The Large pizza size is the most ordered
- Classic Deluxe / Pepperoni pizzas are top revenue generators
- The Classic category contributes the highest revenue
- Total revenue shows a steady upward trend over the period
- A few high-priced pizzas contribute disproportionally to revenue

punitpalofficial@gmail.com

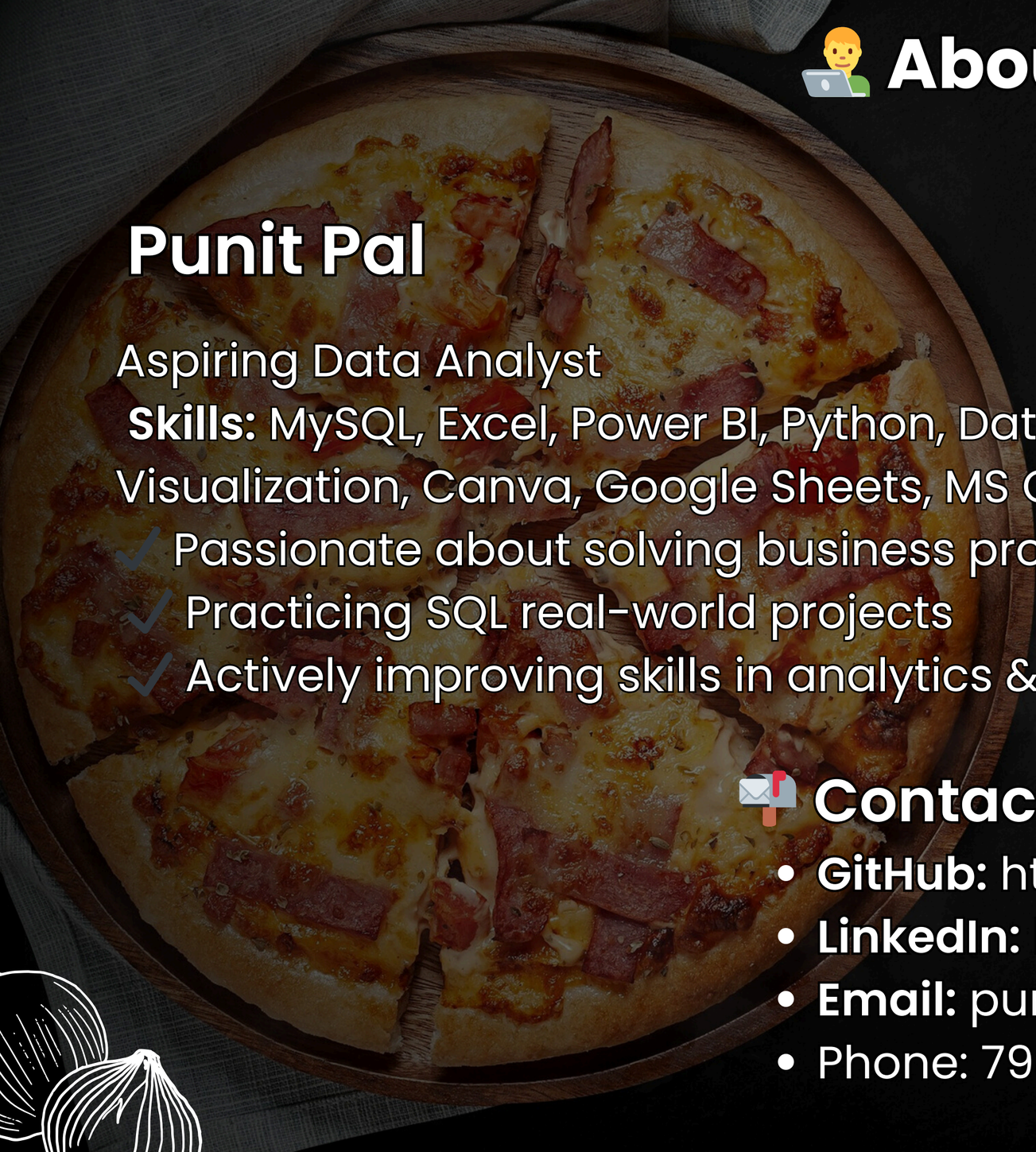
Conclusion:

This MySQL project helped identify key business insights using SQL techniques such as:

- **Joins**
- **Aggregations**
- **Group By**
- **Window Functions**
- **CTEs**
- **Time-based functions**

These insights can help the business improve marketing, plan stock, optimize staffing, and grow sales.

punitpalofficial@gmail.com



👨‍💻 About Me:

Punit Pal

Aspiring Data Analyst

Skills: MySQL, Excel, Power BI, Python, Data Cleaning, Data Visualization, Canva, Google Sheets, MS Office, Google Suite.

- ✓ Passionate about solving business problems using data
- ✓ Practicing SQL real-world projects
- ✓ Actively improving skills in analytics & visualization

📬 Contact

- **GitHub:** https://github.com/punitpalofficial
- **LinkedIn:** https://www.linkedin.com/in/punit-pal/
- **Email:** punitpalofficial@gmail.com
- Phone: 7983555927

punitpalofficial@gmail.com

THANK YOU!

punitpalofficial@gmail.com