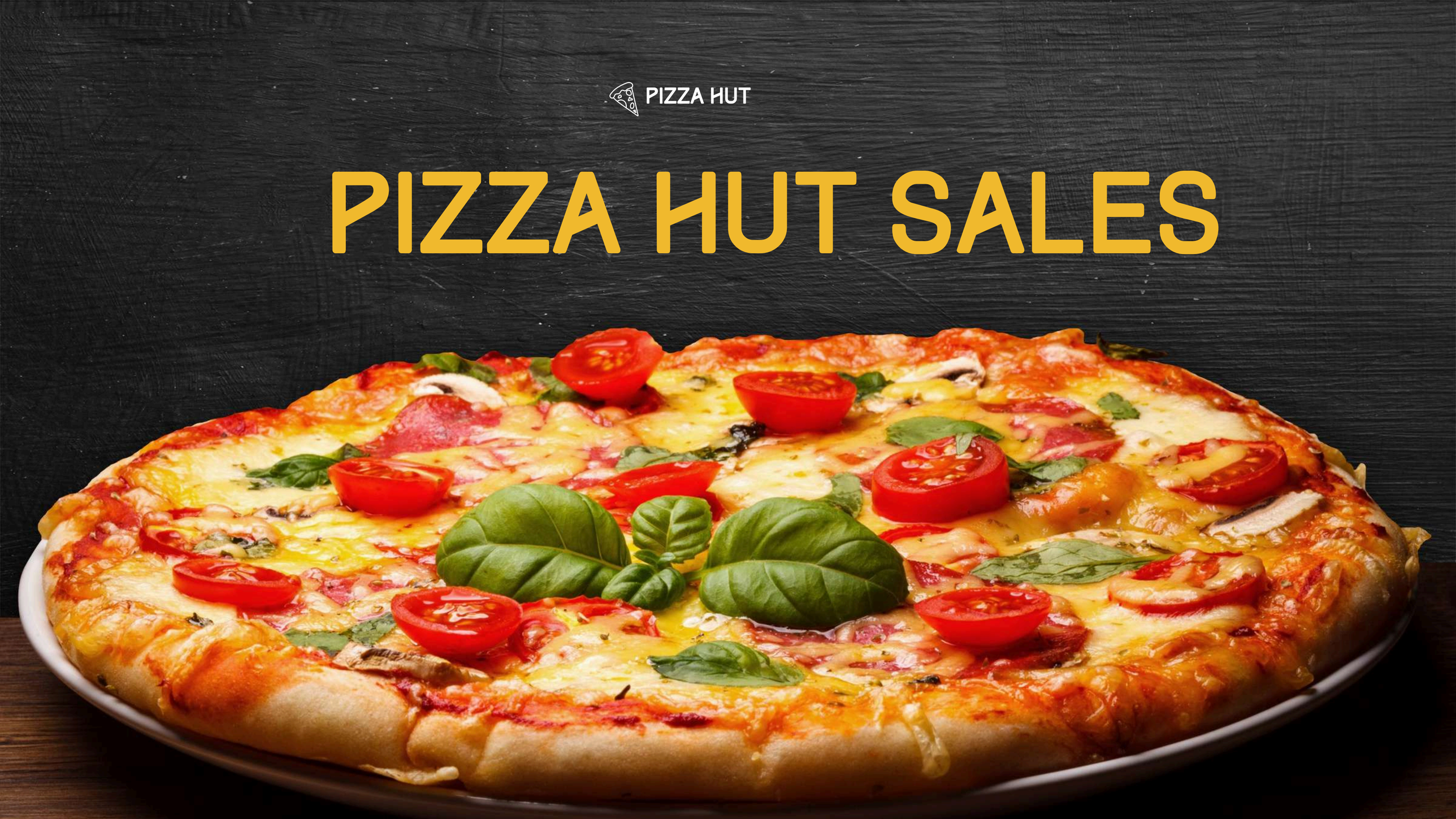
PIZZA HUT

# PIZZA HUT SALES



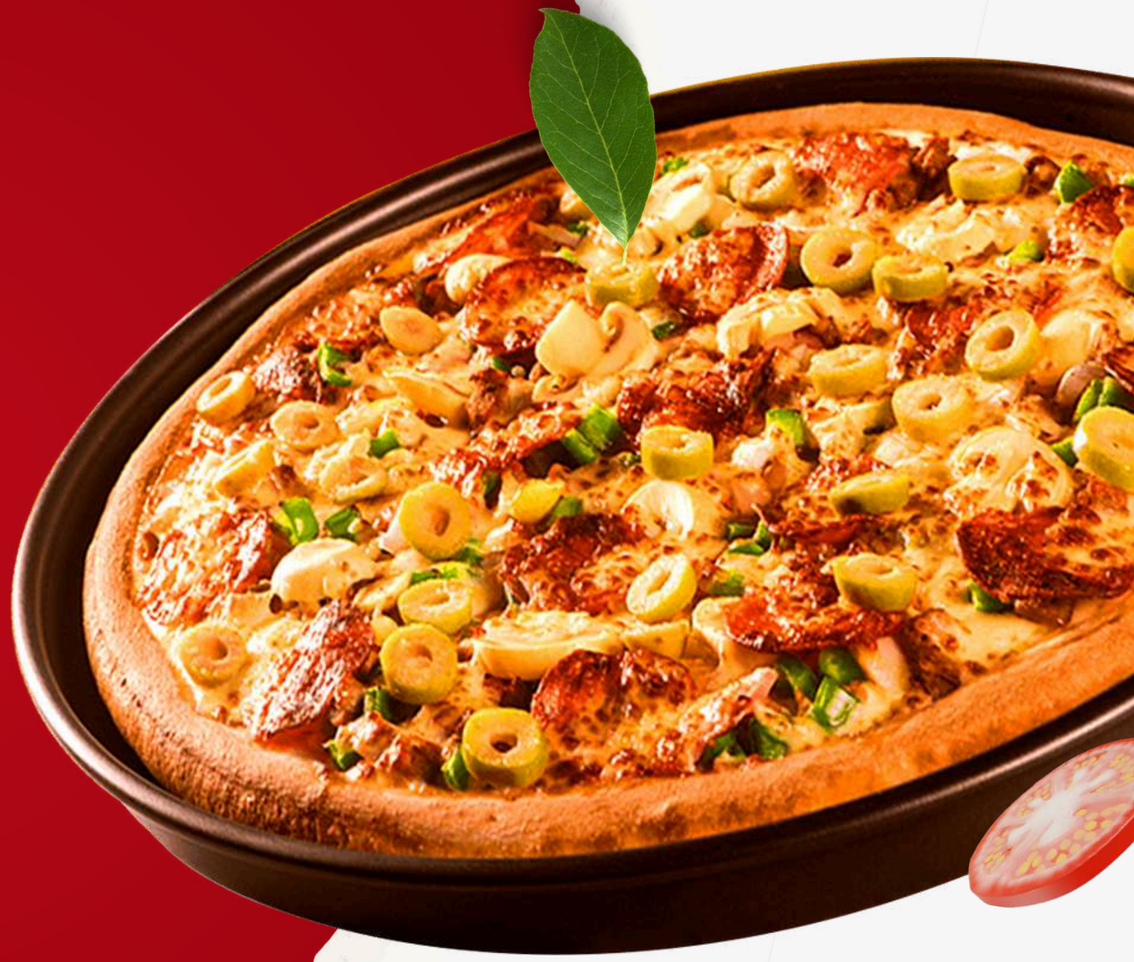
PIZZA HUT

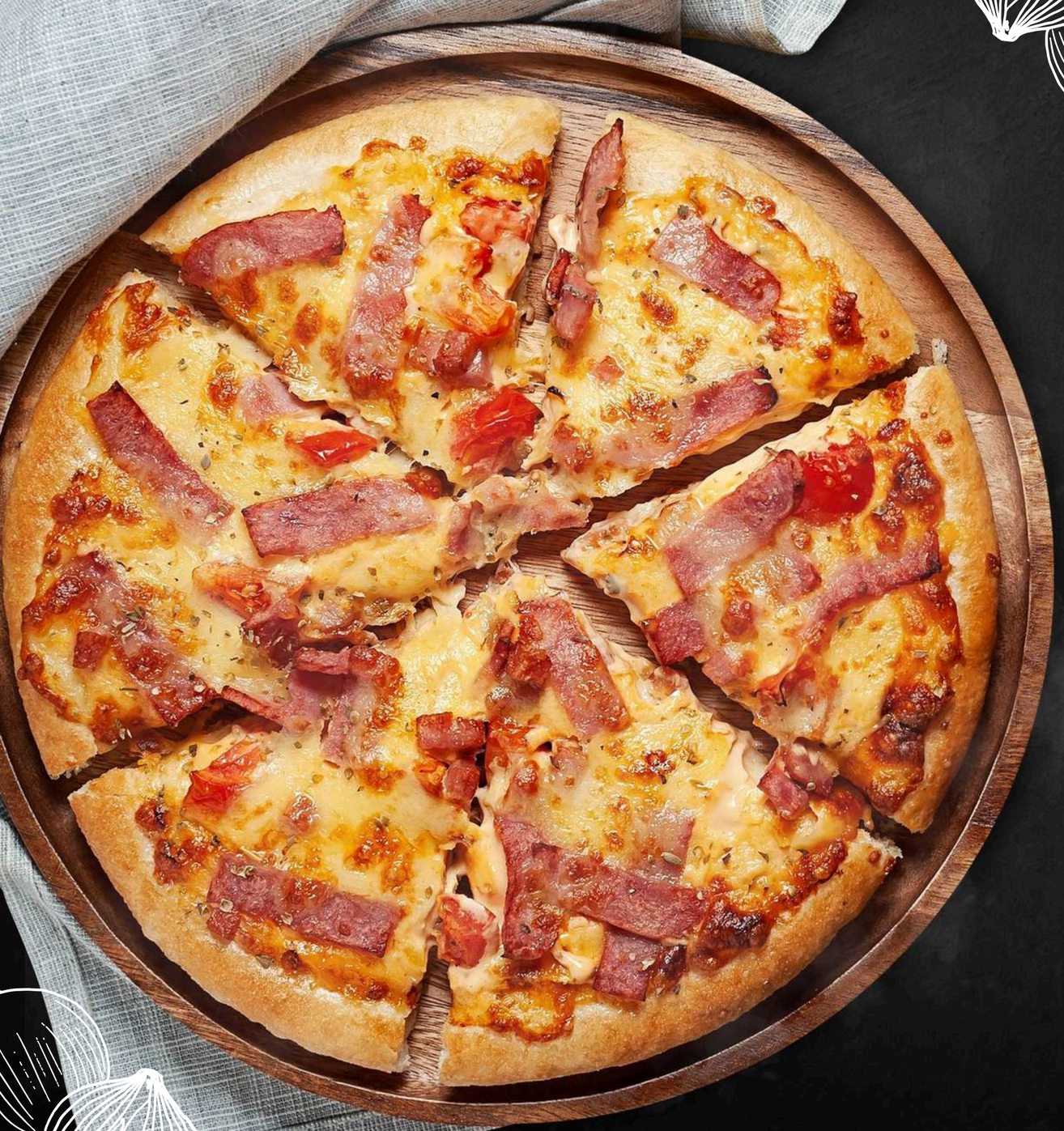
# WELCOME TO

# PIZZA HUT SALES PROJECT

My name is **Mohsin Saifi**, and I have created this project on Pizza Hut Sales Analysis using SQL. The project is based on a dataset containing information about pizza categories, sizes, prices, order quantities, and customer details.

Queries Used: Joins, Window Functions, Aggregations, Group By, Order By, Subquerie





PIZZA HUT

# VISSION & MISSION

## VISSION

To build a data-driven sales analysis system that helps Pizza Hut gain clear visibility into their sales performance, customer preferences, and market trends, enabling smarter decisions for growth and customer satisfaction.

## MISSION

- To analyze Pizza Hut sales data using SQL and uncover meaningful insights.
- To provide actionable reports on revenue, product performance, and customer behavior.
- optimization, pricing strategies, and promotional offers.

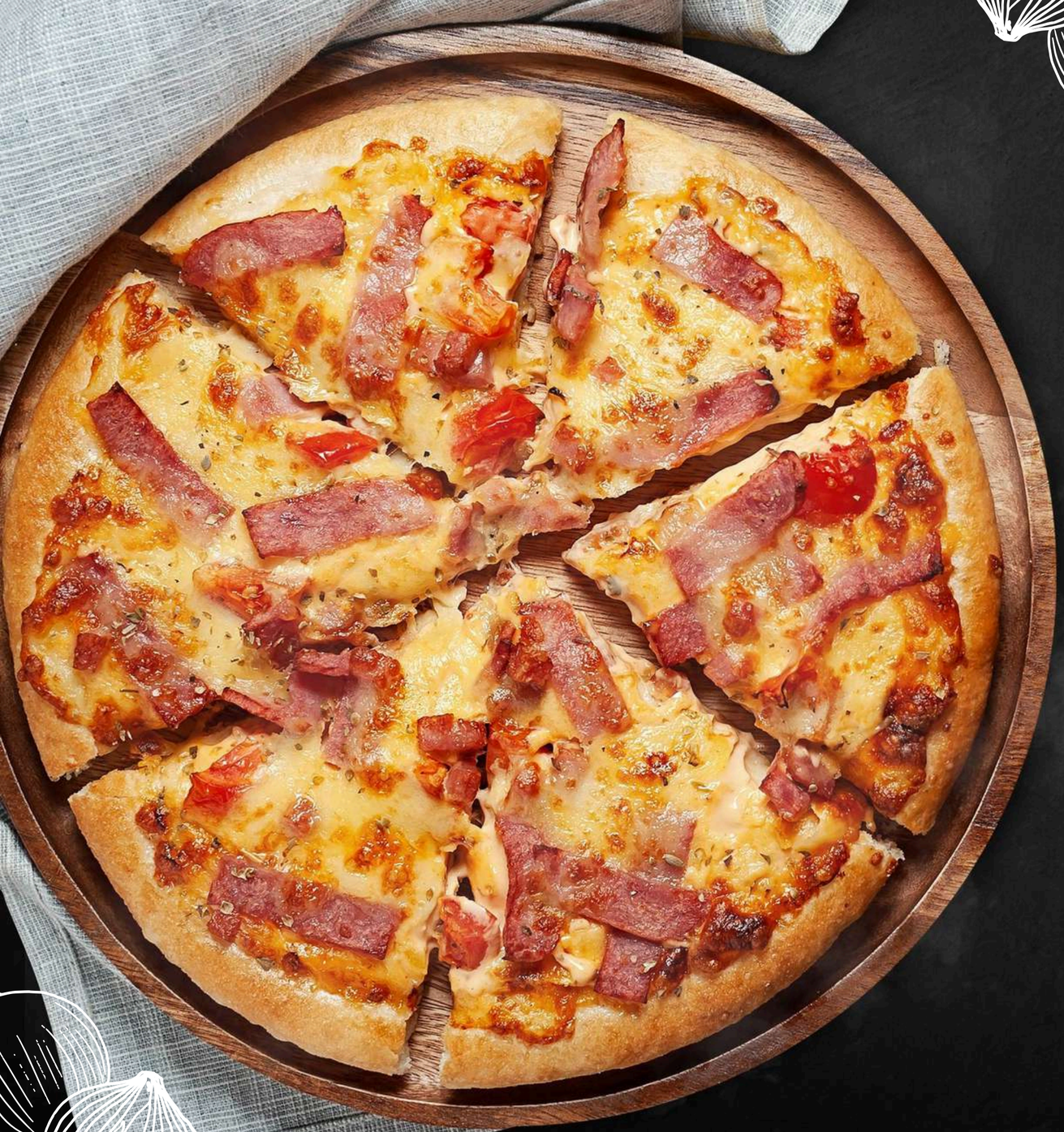
LARANA PIZZA

# PIZZA HUT CATEGORY

'CHICKEN'

'CLASSIC'

'SUPREME'
'VEGGIE'



PIZZA HUT

# QUESTION

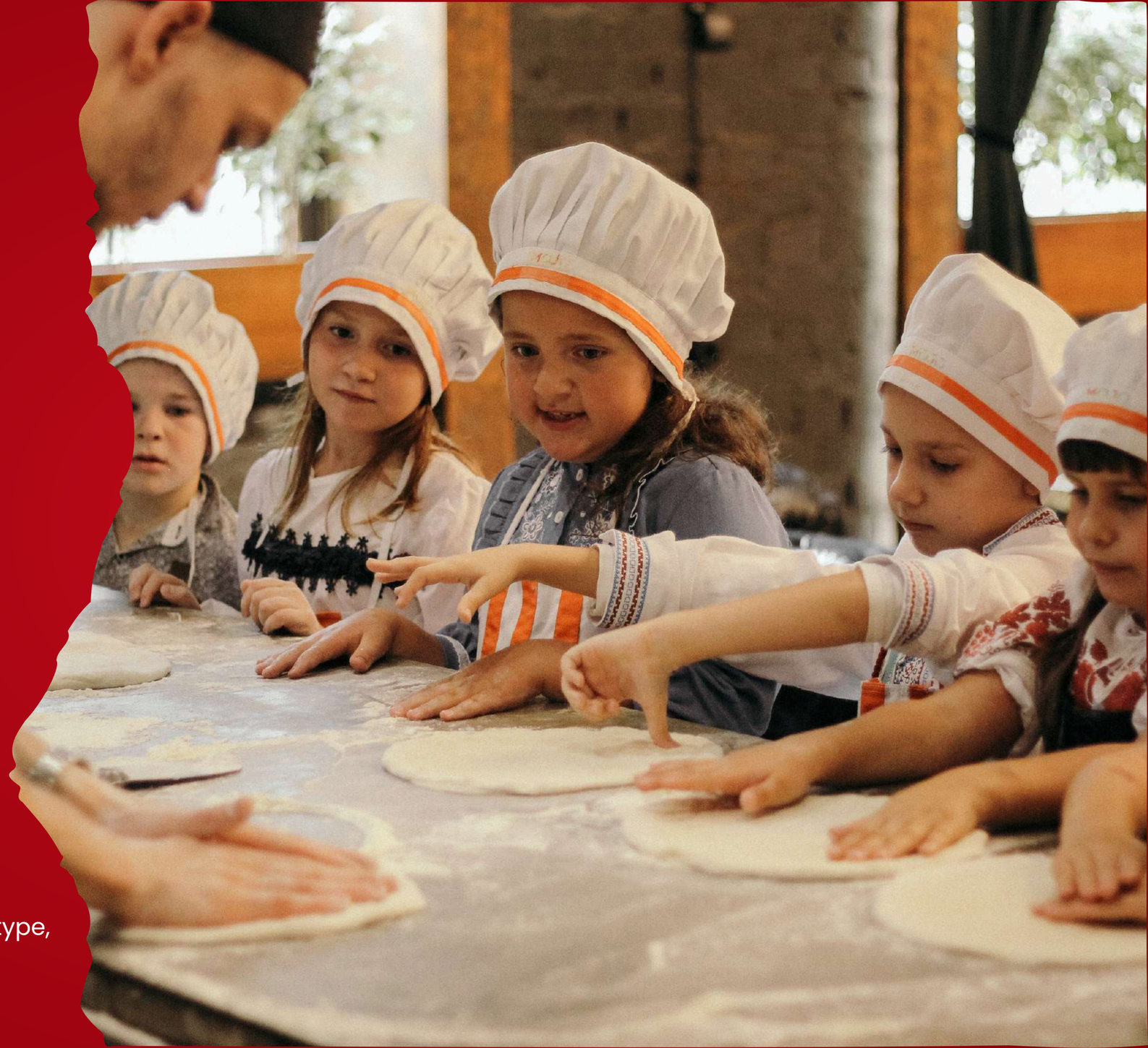
- Retrieve the total number of orders placed.
- Calculate the total revenue generated from pizza sales.
- Identify the highest-priced pizza.
- Identify the most common pizza size ordered.
- List the top 5 most ordered pizza types along with their quantities.
- Join the necessary tables to find the total quantity of each pizza category ordered.
- Determine the distribution of orders by hour of the day.
- Join relevant tables to find the category-wise distribution of pizzas.
- Group the orders by date and calculate the average number of pizzas ordered per day.
- Determine the top 3 most ordered pizza types based on revenue.
- Calculate the percentage contribution of each pizza type to total revenue.
- Analyze the cumulative revenue generated over time.
- Determine the top 3 most ordered pizza types based on revenue for each pizza category.

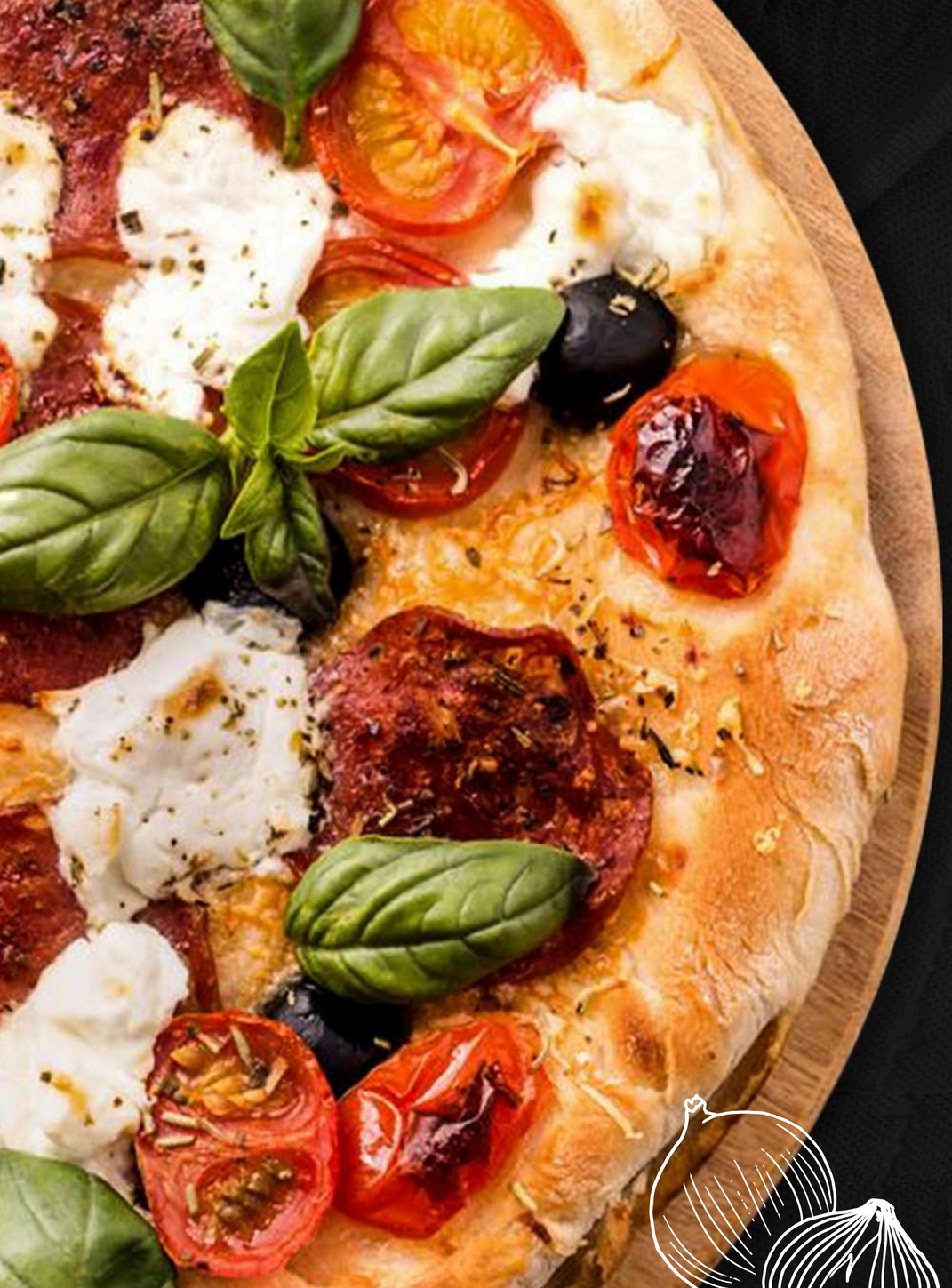
PIZZA HUT

# USING SQL FOR INSIGHTS ANALYSIS

"Using SQL to analyze Pizza Hut sales data and deliver business-desired insights for better decision-making."

4 Types of data sets - pizzas, order, pizza_type, order_details





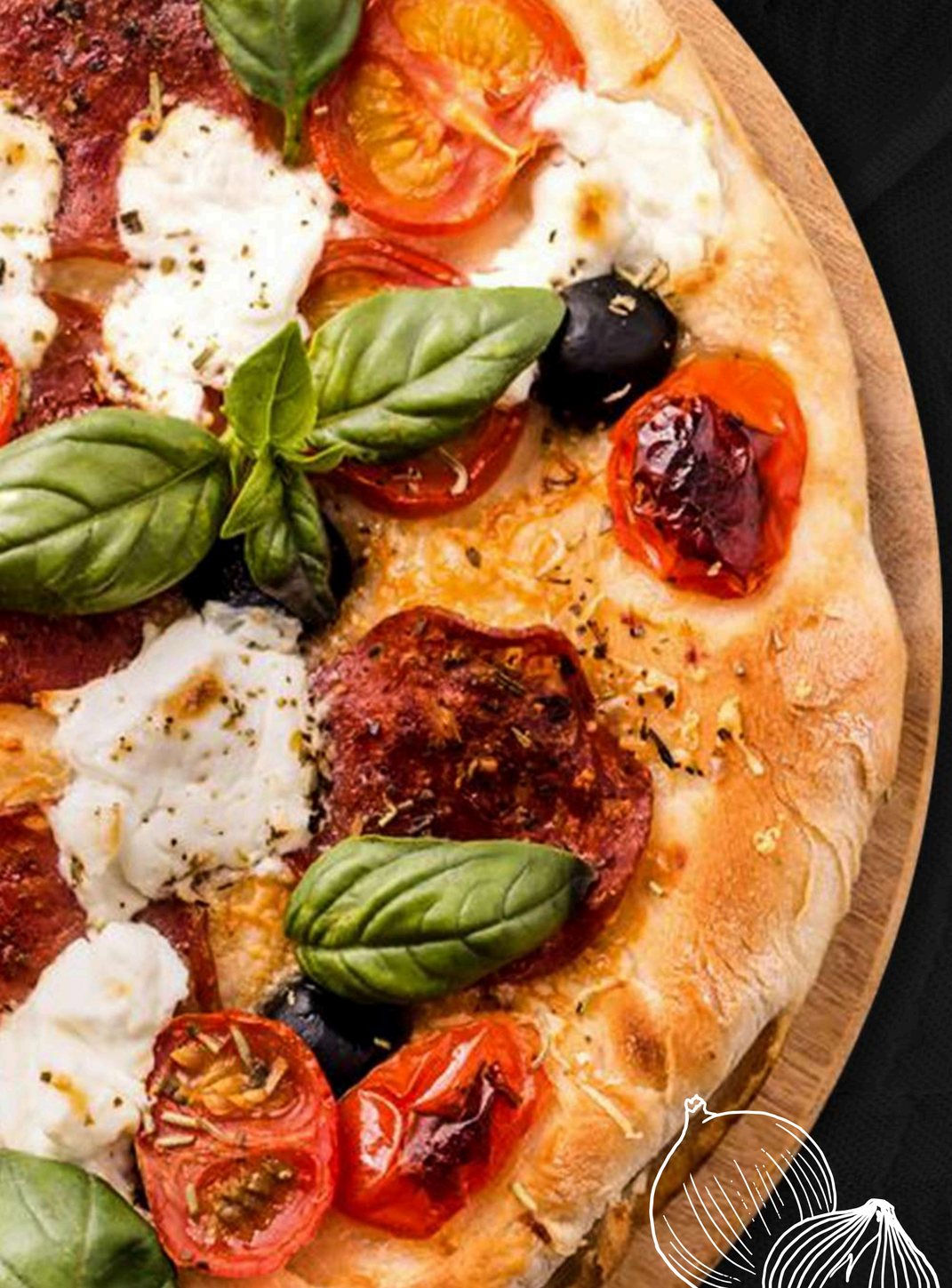
PIZZA HUT

# RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

```
SELECT
    COUNT(order_id) AS total_no_of_orders_placed
FROM
    pizzahut.orders;
```

Result Grid | Filter Rows:

| total_no_of_orders_placed |
| --- |
| 21350 |



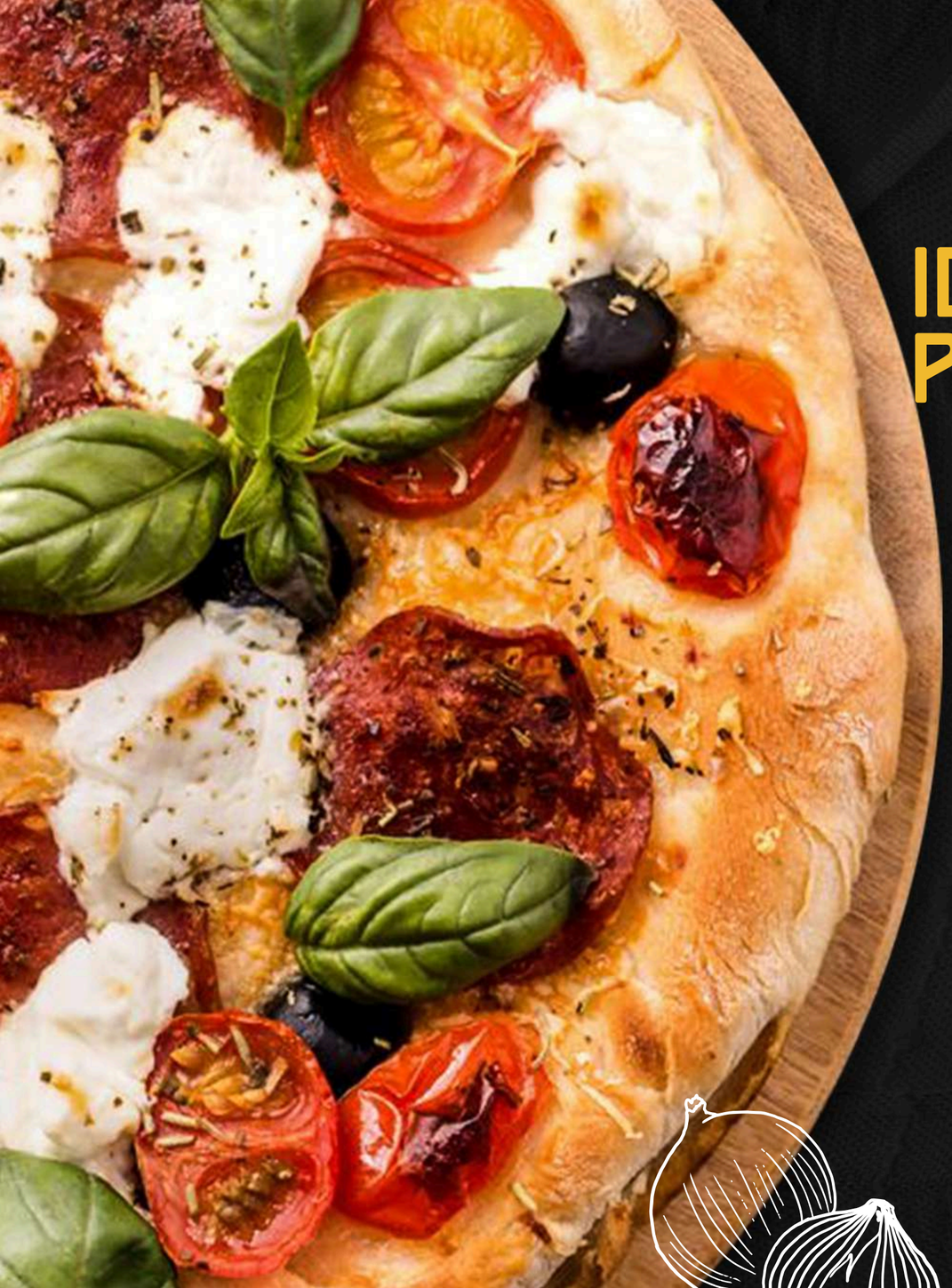
PIZZA HUT

# CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

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

Result Grid

| | total_revenue |
|---|---|
| ▶ | 817860.05 |

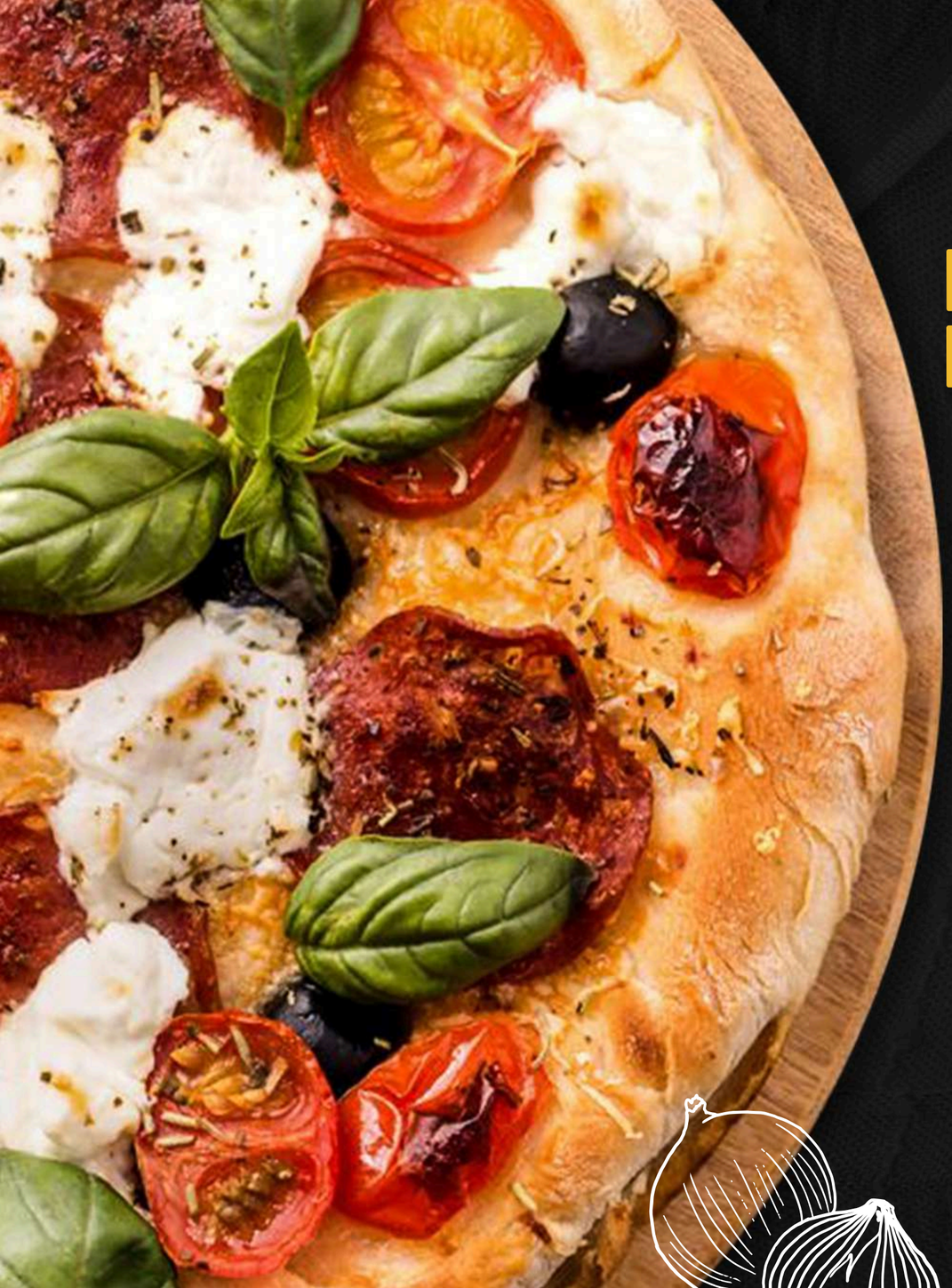


PIZZA HUT

# IDENTIFY THE HIGHEST-PRICED PIZZA.

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

| | pizza_type_id | name | price |
|---|---|---|---|
| ▶ | the_greek | The Greek Pizza | 35.95 |

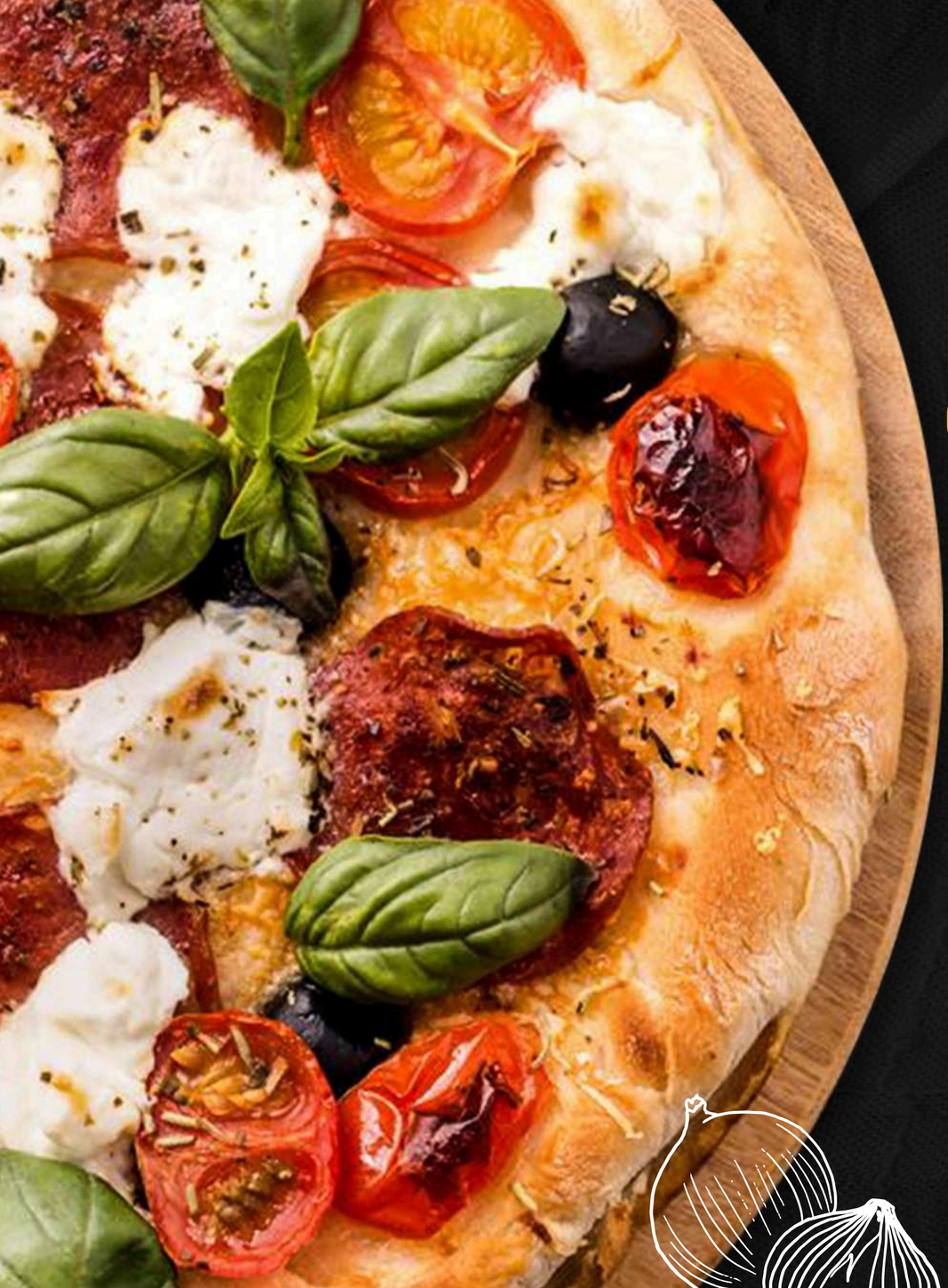


PIZZA HUT

# IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

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

| | size | orders_counts |
|---|---|---|
| ▶ | L | 18526 |

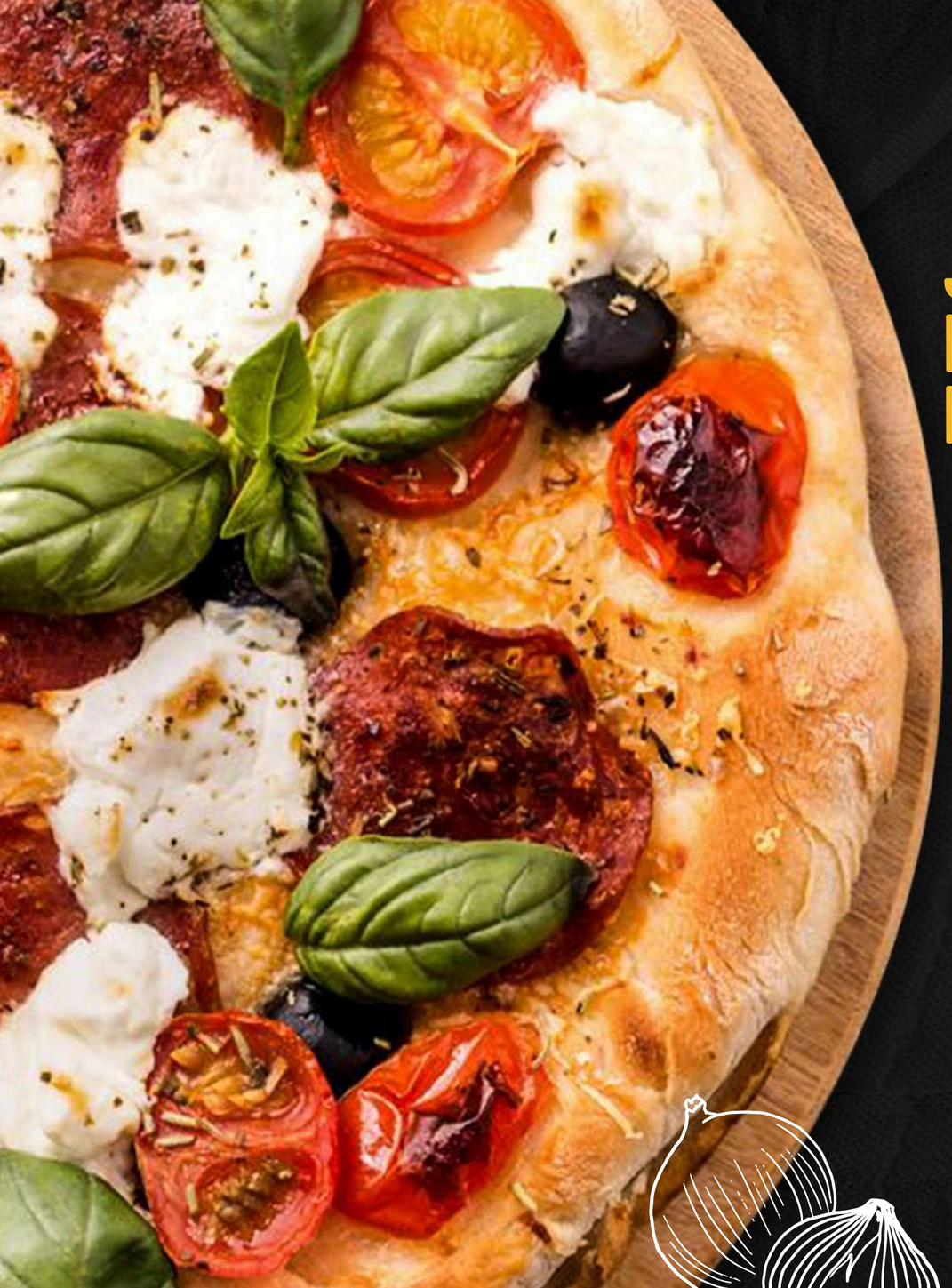


PIZZA HUT

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

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

| name | total_quantity |
|---|---|
| The Classic Deluxe Pizza | 2453 |
| The Barbecue Chicken Pizza | 2432 |
| The Hawaiian Pizza | 2422 |
| The Pepperoni Pizza | 2418 |
| The Thai Chicken Pizza | 2371 |

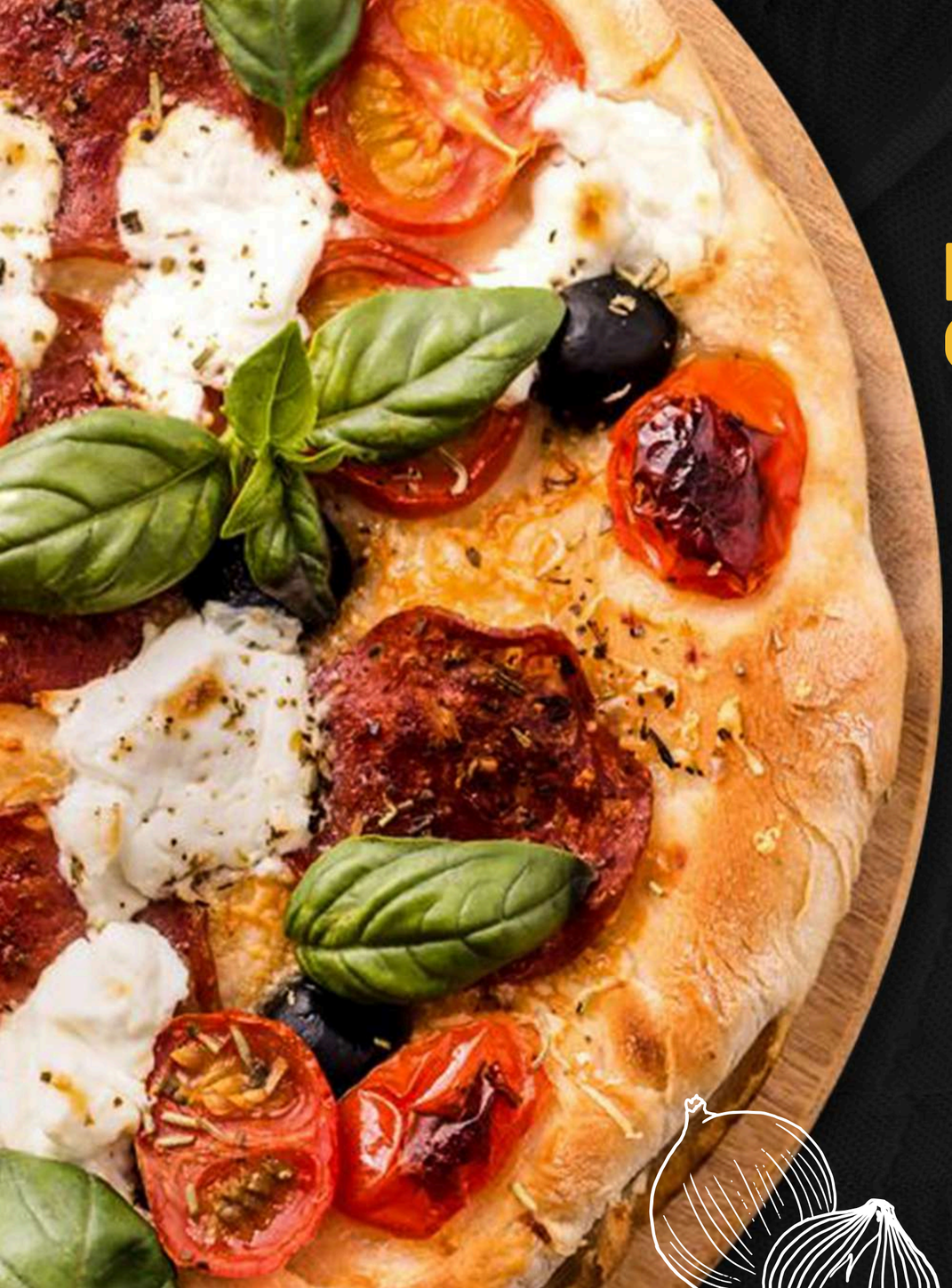


PIZZA HUT

# 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 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;
```

| | category | quantity |
|---|---|---|
| ▶ | Classic | 14888 |
| | Veggie | 11649 |
| | Supreme | 11987 |
| | Chicken | 11050 |

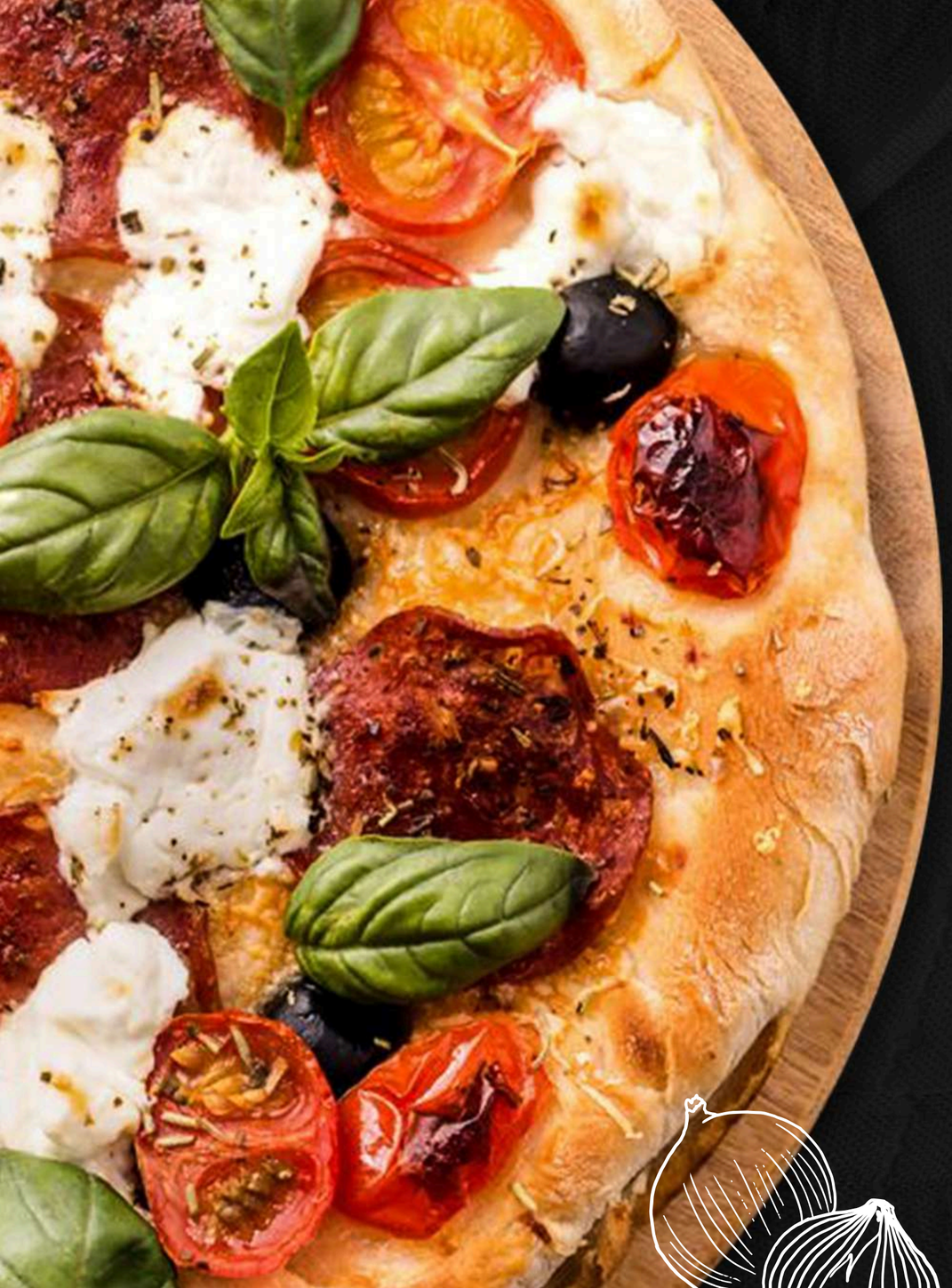


PIZZA HUT

# DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

```sql
SELECT
    HOUR(order_time) AS hour,
    COUNT(order_id) AS count_of_distribution
FROM
    orders
GROUP BY hour;
```

| hour | count_of_distribution |
|---|---|
| 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 |

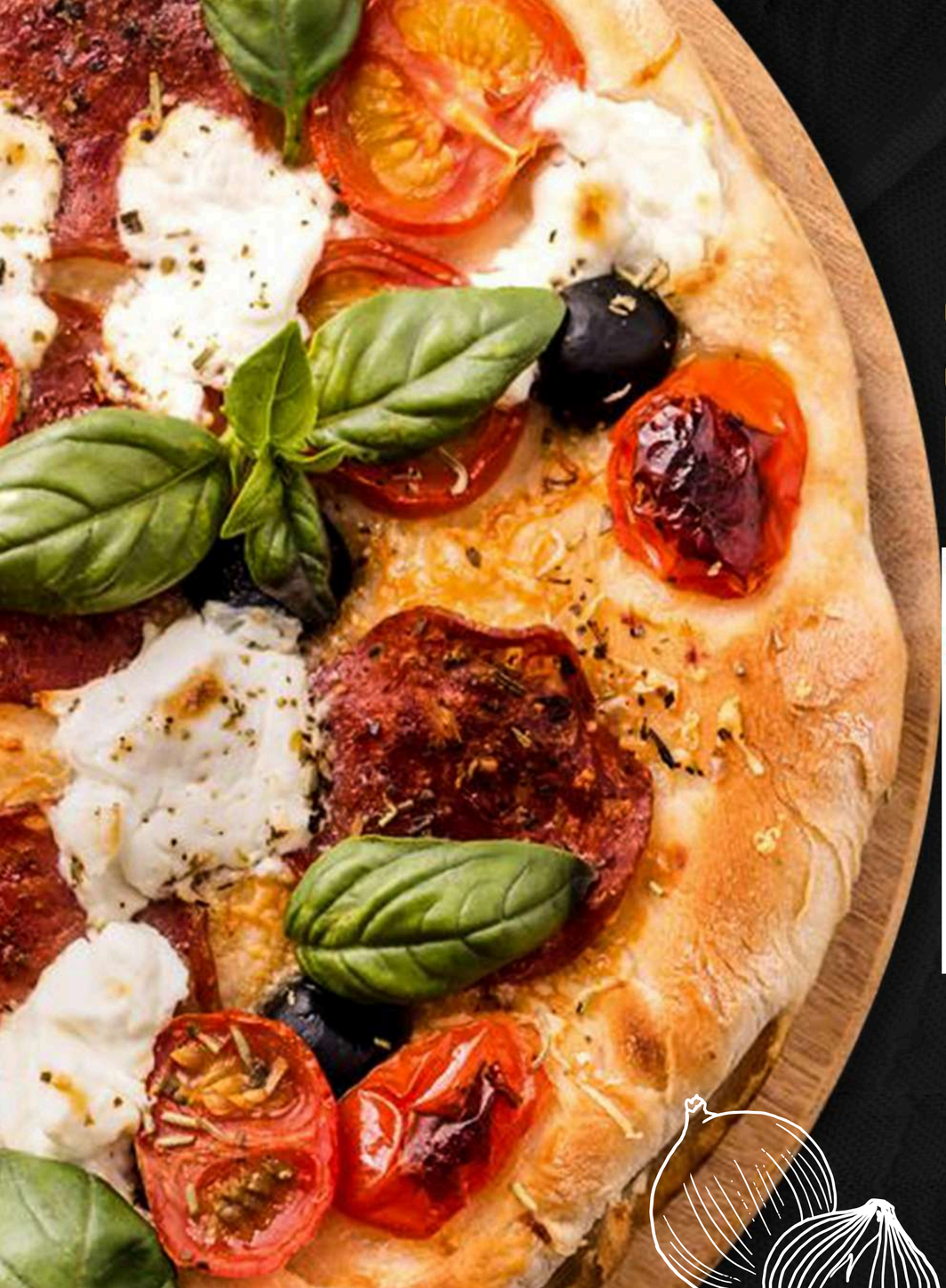


PIZZA HUT

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

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

| category | counts |
|---|---|
| Chicken | 6 |
| Classic | 8 |
| Supreme | 9 |
| Veggie | 9 |



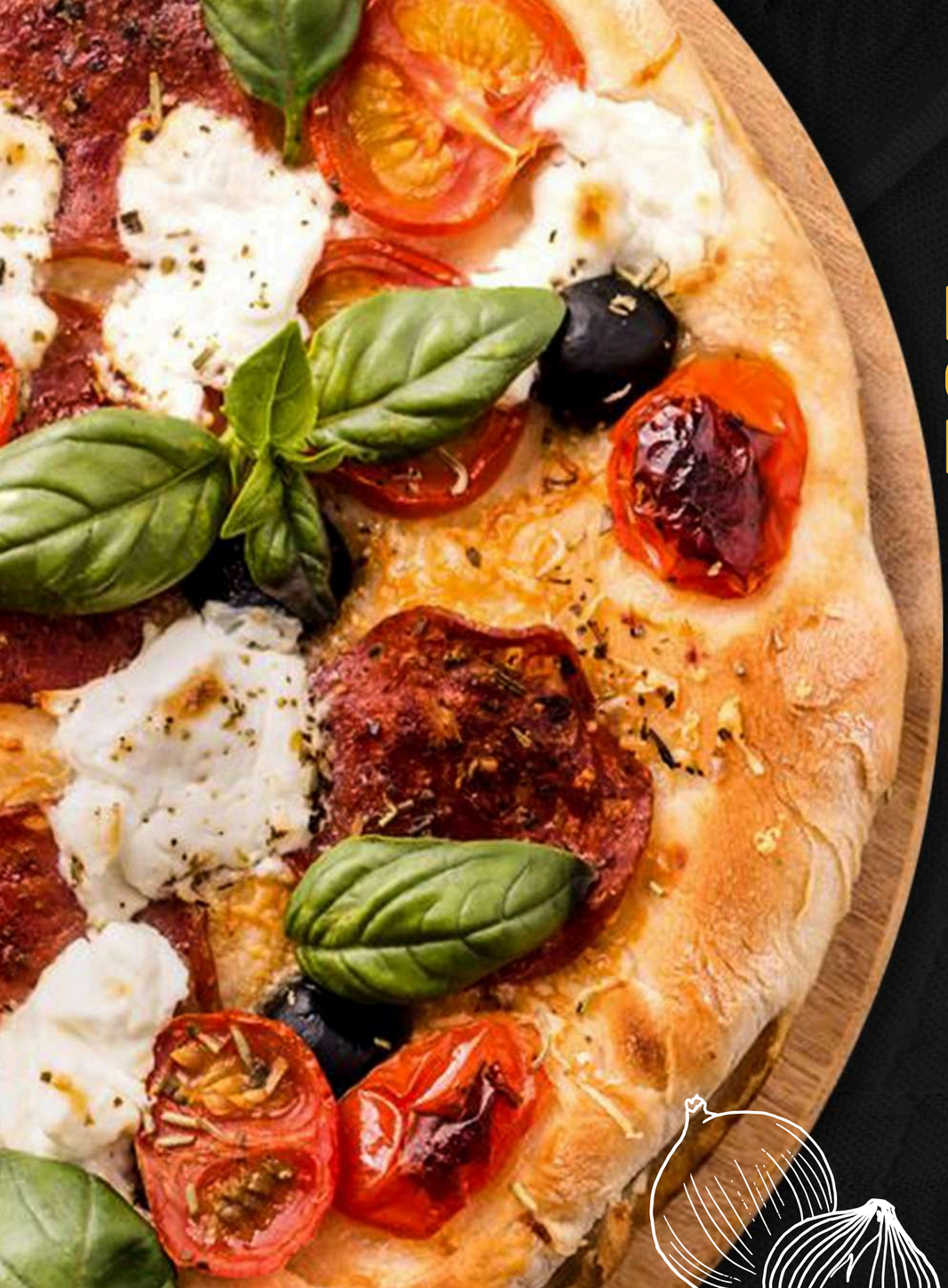
PIZZA HUT

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

```sql
SELECT
    ROUND(AVG(total_quantity), 0) AS per_day_avg_orders
FROM
    (SELECT
        orders.order_date,
            SUM(order_details.quantity) AS total_quantity
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY orders.order_date) AS date_by_quantity_order;
```

Result Grid | Filte

| per_day_avg_orders |
| --- |
| 138 |

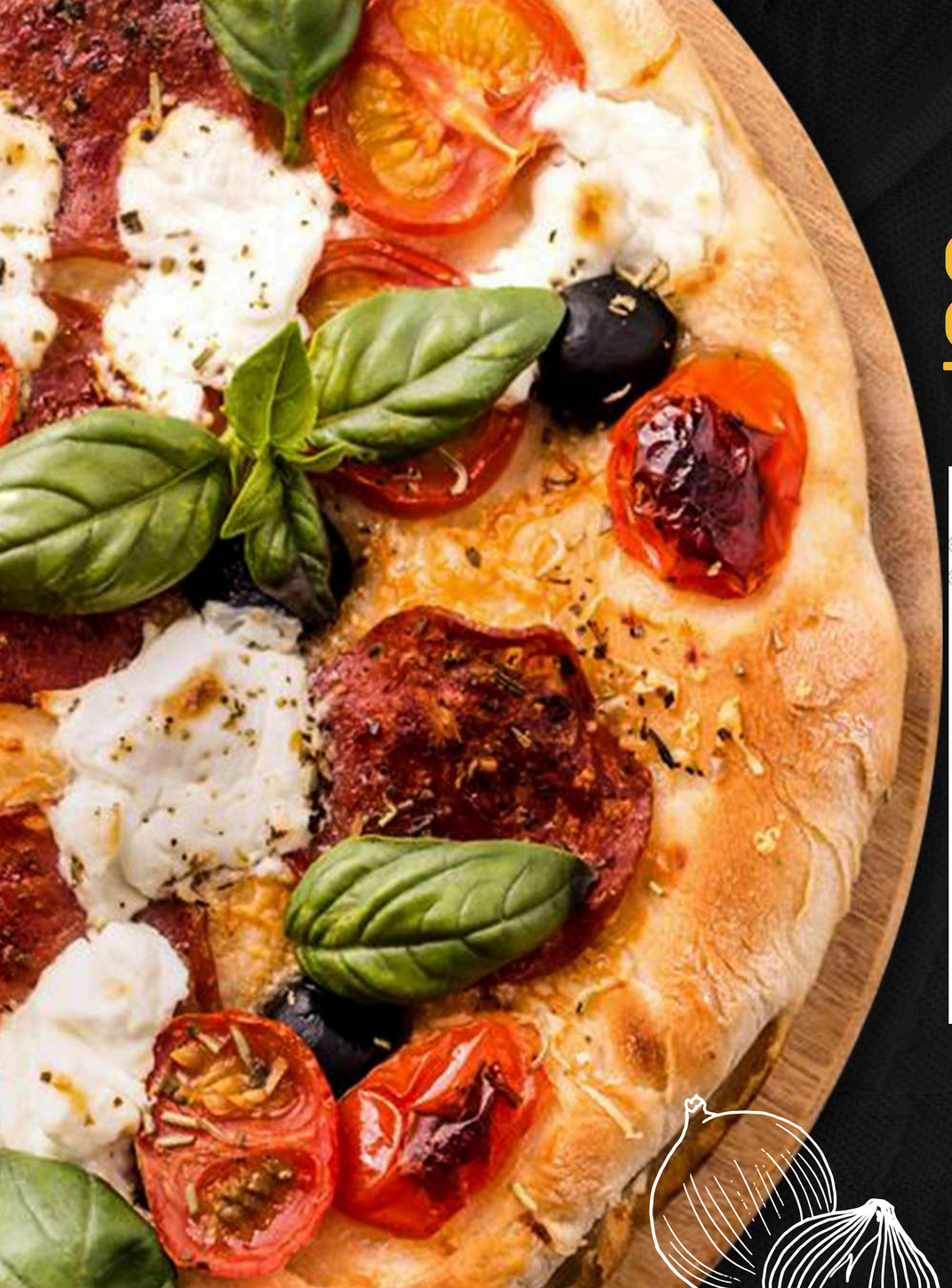


PIZZA HUT

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

```
SELECT
    pizza_types.name,
    SUM(pizzas.price * order_details.quantity) 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;
```

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

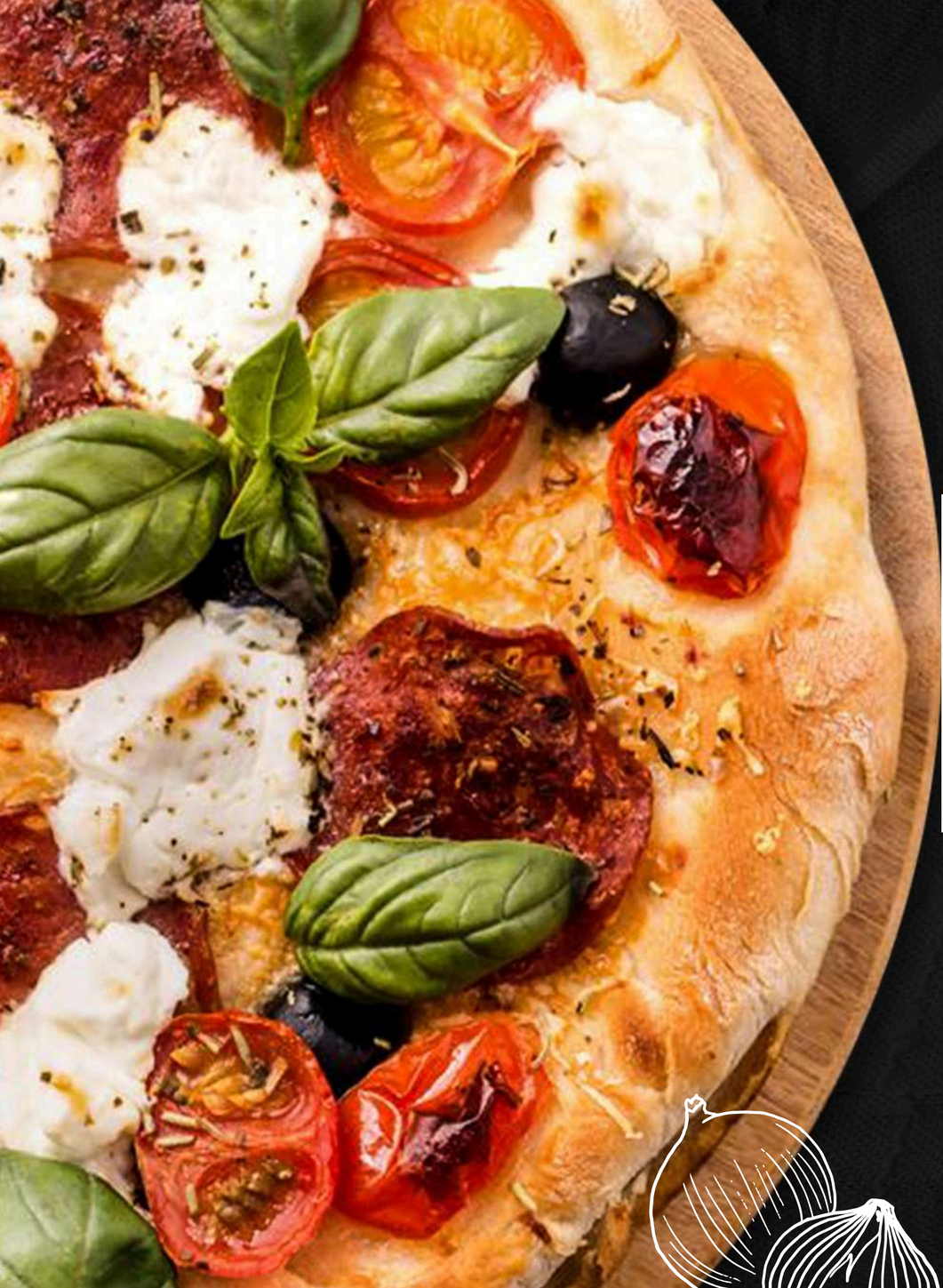


PIZZA HUT

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

```
SELECT
    pizza_types.category,
    ROUND(SUM(pizzas.price * order_details.quantity) / (SELECT
                    ROUND(SUM(pizzas.price * order_details.quantity),
                                2)
                FROM
                    order_details
                        JOIN
                    pizzas ON pizzas.pizza_id = order_details.pizza_id) * 100,
            2) AS revenue_by_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 category;
```

| category | revenue_by_percentage |
|---|---|
| Classic | 26.91 |
| Veggie | 23.68 |
| Supreme | 25.46 |
| Chicken | 23.96 |

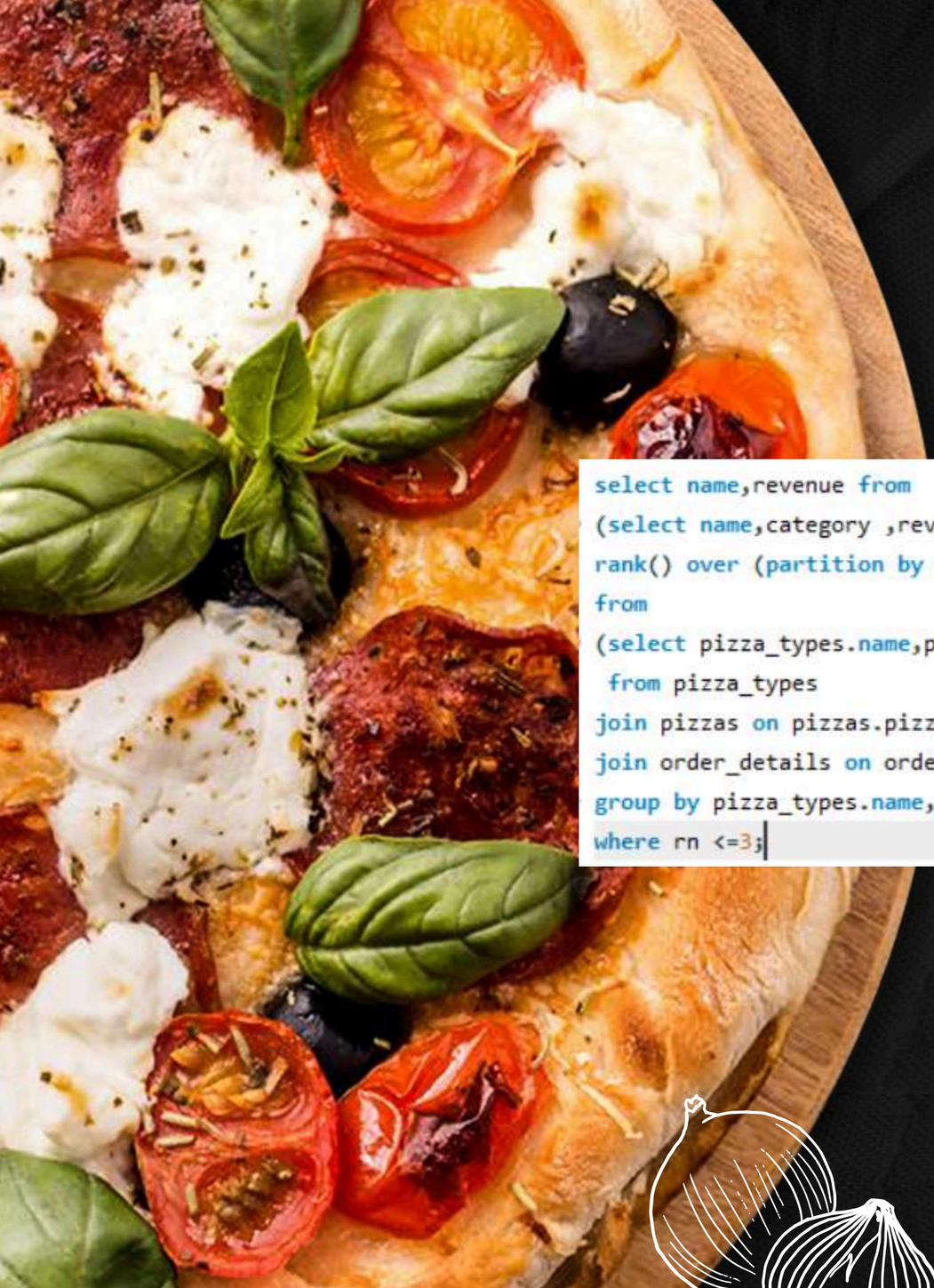


PIZZA HUT

# 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 sale;
```

| 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 |



PIZZA HUT

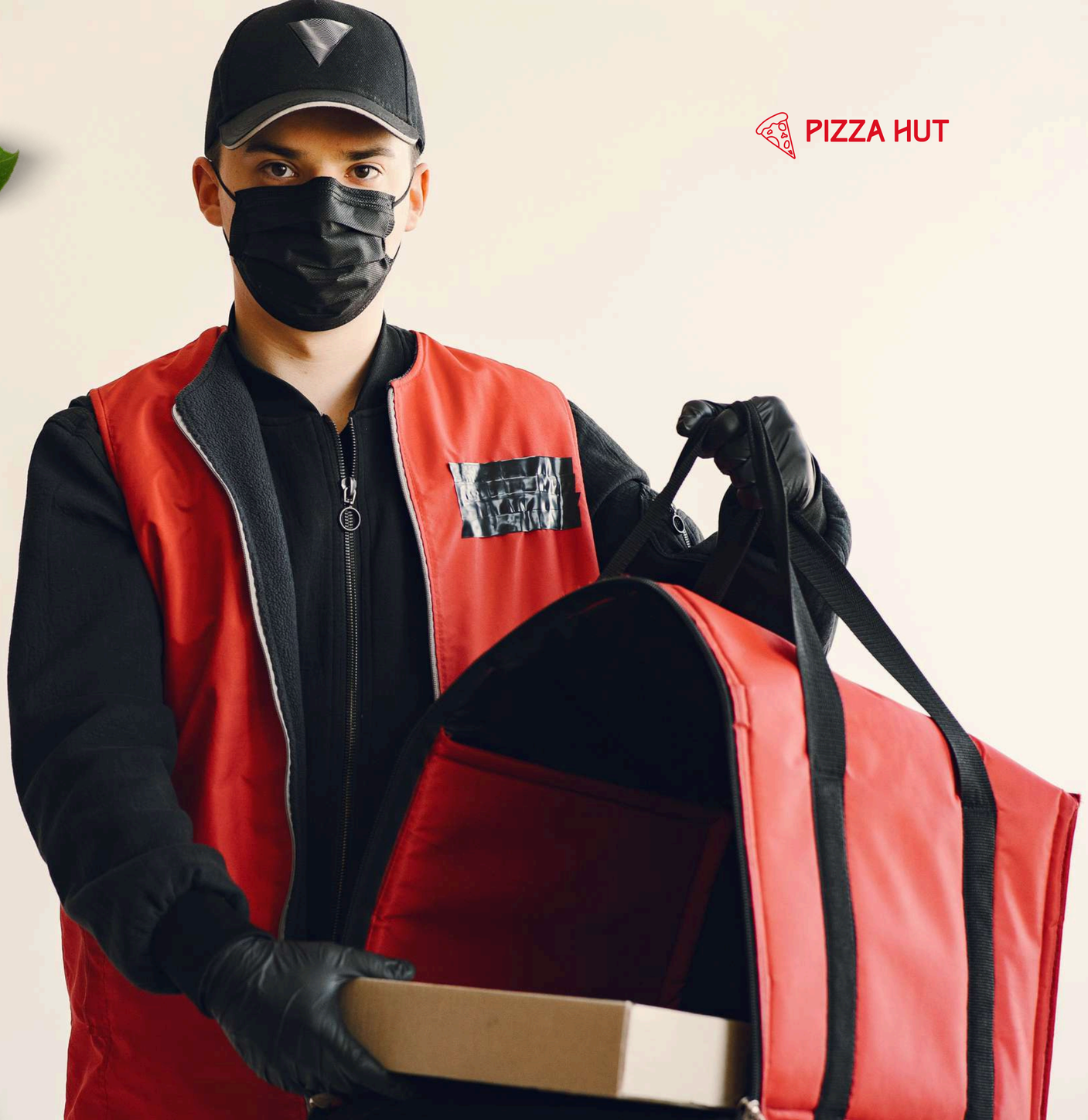
# DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY.

```sql
select name,revenue from
(select name,category ,revenue ,
rank() over (partition by category order by revenue desc)  as rn
from
(select pizza_types.name,pizza_types.category,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,pizza_types.category ) as a) as b
where rn <=3;
```

| 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 | 32265.70000000065 |
| The Mexicana Pizza | 26780.75 |
| The Five Cheese Pizza | 26066.5 |

PIZZA HUT

# FREE DELIVERY SERVICE



PIZZA HUT

# THANK YOU!

MOHSIN SAIFI