Pizza Sales Analysis Using MY SQL

Introduction

- This MySQL project focuses on analyzing a pizza sal es dataset to gain insights into various aspects of the business. The dataset includes information about ord ers, pizza types, prices, and order timings. By using SQL queries, the project aims to retrieve meaningful data that can help in understanding customer prefere nces, optimizing the menu, and improving overall bu siness strategies.

Objective

The objective of this pizza sales project is to analyze the sales data to extract actionable insights that can inform business decisions. By utilizing SQL, the project aims to:

1.Understand Sales Performance: Calculate total orders, revenue, and identify the best-performing pizzas in terms of sales volume and revenue.

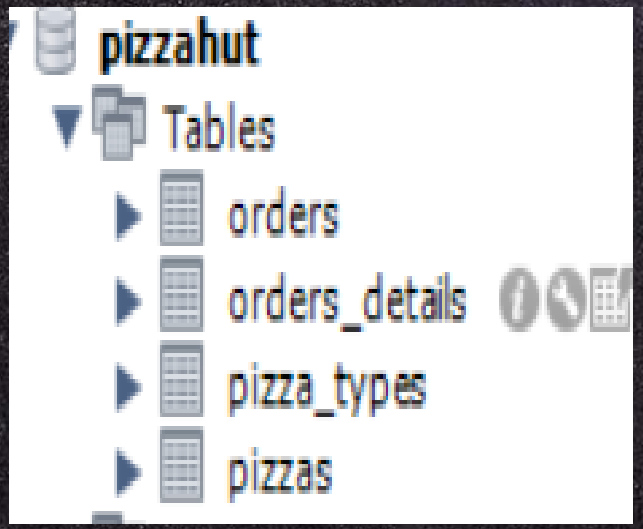
2.Customer Preferences: Identify the most popular pizza sizes and types to understand customer preferences and demand patterns.

3.Optimize Inventory and Menu: Analyze the distribution of orders by time and category to optimize inventory management and tailor the menu to meet customer demand effectively.

4.Revenue Analysis: Assess the contribution of different pizza types to total revenue and evaluate revenue trends over time to support strategic planning and forecasting.

Through this analysis, the project seeks to provide a comprehensive understanding of sales dynamics, enabling better decision-making and improved business performance.

PIZZA SALES DATABASE




```sql
create database pizzahut;

create table orders (order_id int not null,
order_date date not null,
order_time time not null,
primary key (order_id));

create table orders_details (
order_details_id int not null,
order_id int not null,
pizza_id text not null,
quantity int not null,
primary key (order_details_id));
```

Retrieve the total number of orders placed

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

Result Grid

	total_orders
▶	21350

Calculate the total revenue generated from pizza sales.

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

Result Grid

total_sales
817860.05

Identify the highest-priced pizza

```sql
SELECT
    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
LIMIT 1;
```

Result Grid | Filter Rows:

name	price
The Greek Pizza	35.95

Identify the most common pizza size ordered

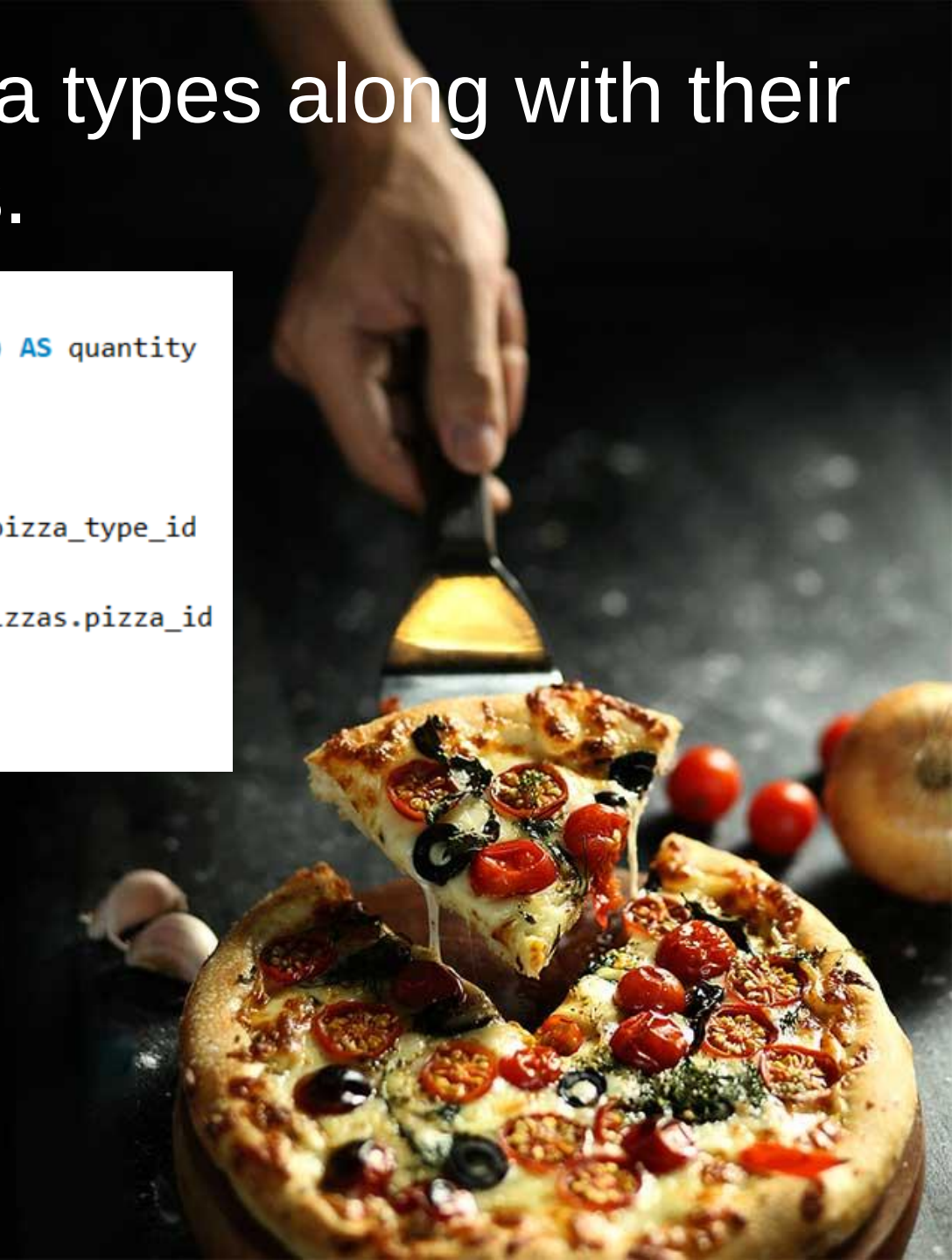
```sql
SELECT
    pizzas.size,
    COUNT(orders_details.order_details_id) AS order_count
FROM
    pizzas
        JOIN
    orders_details ON pizzas.pizza_id = orders_details.pizza_id
GROUP BY pizzas.size
ORDER BY order_count DESC;
```

size	order_count
L	18526
M	15385
S	14137
XL	544
XXL	28

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

```sql
SELECT
    pizza_types.name, SUM(orders_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY quantity DESC
LIMIT 5;
```

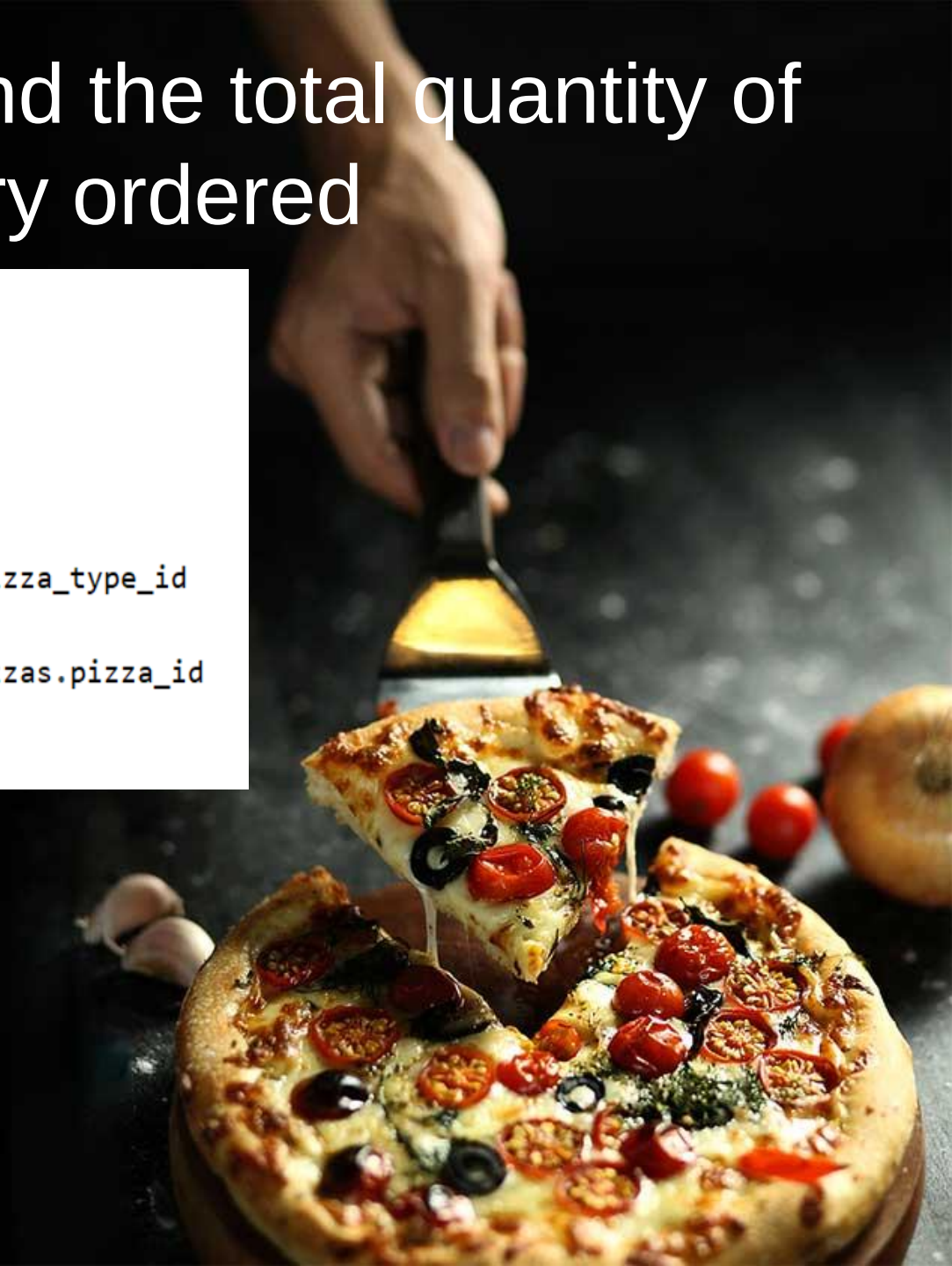
name	quantity
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371



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

```sql
SELECT
    pizza_types.category,
    SUM(orders_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY quantity DESC;
```

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050



Determine the distribution of orders by hour of the day.

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

HOUR(order_time)	order_count
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

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

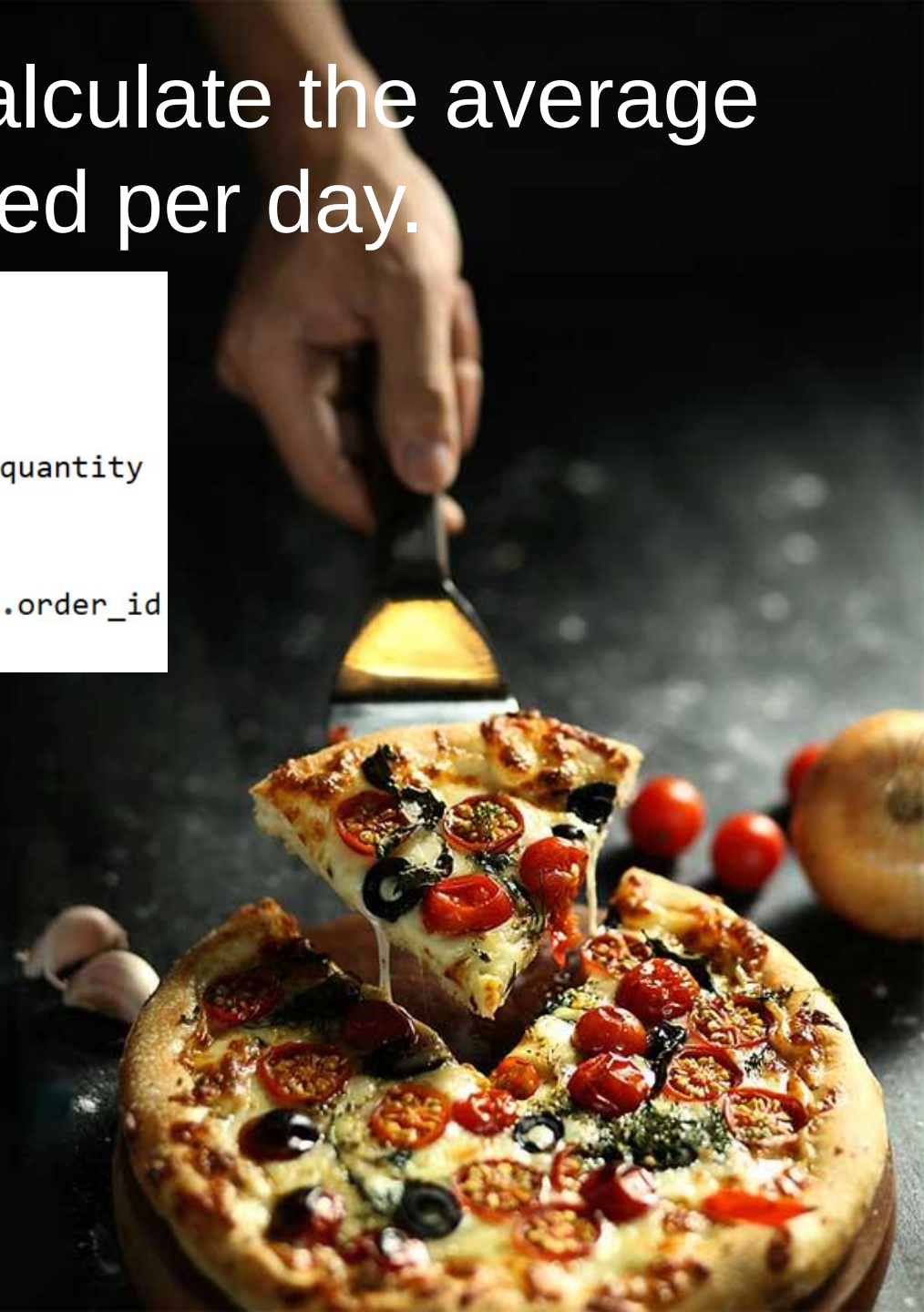
```
select category,count(name) from pizza_types
group by category;
```

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

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

```sql
SELECT
    ROUND(AVG(quantity), 0) as avg_pizza_ordered_per_day
FROM
    (SELECT
        orders.order_date, SUM(orders_details.quantity) AS quantity
    FROM
        orders
    JOIN orders_details ON orders.order_id = orders_details.order_id
    GROUP BY orders.order_date) AS order_quantity;
```

avg_pizza_ordered_per_day
138



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

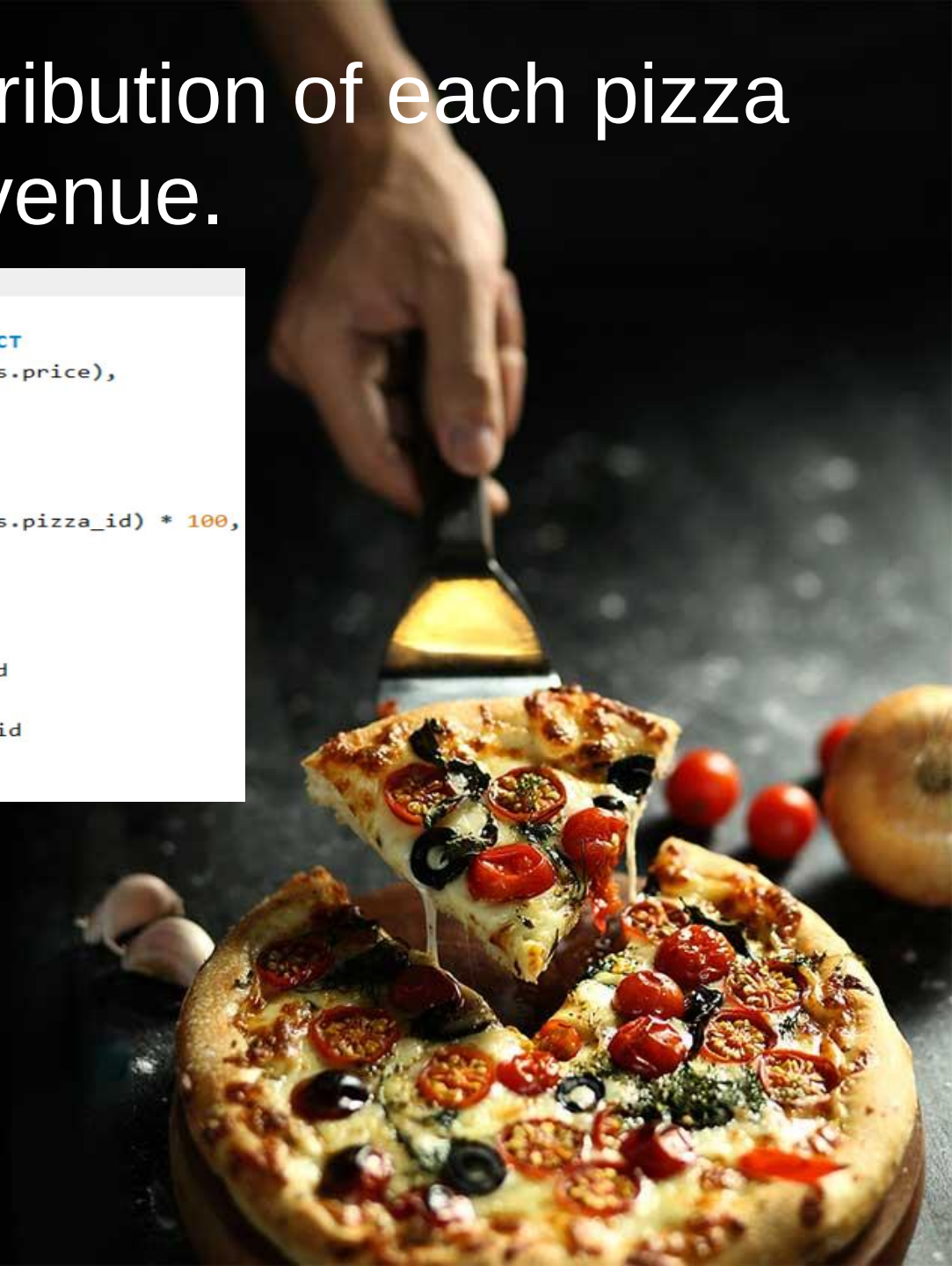
```sql
SELECT
    pizza_types.name,
    SUM(orders_details.quantity * pizzas.price) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY revenue DESC
LIMIT 3;
```

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

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

```sql
SELECT
    pizza_types.category,
    ROUND(SUM(orders_details.quantity * pizzas.price) / (SELECT
                    ROUND(SUM(orders_details.quantity * pizzas.price),
                                2) AS total_sales
                FROM
                    orders_details
                        JOIN
                    pizzas ON pizzas.pizza_id = orders_details.pizza_id) * 100,
            2) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    orders_details ON orders_details.pizza_id = pizzas.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



Analyze the cumulative revenue generated over time

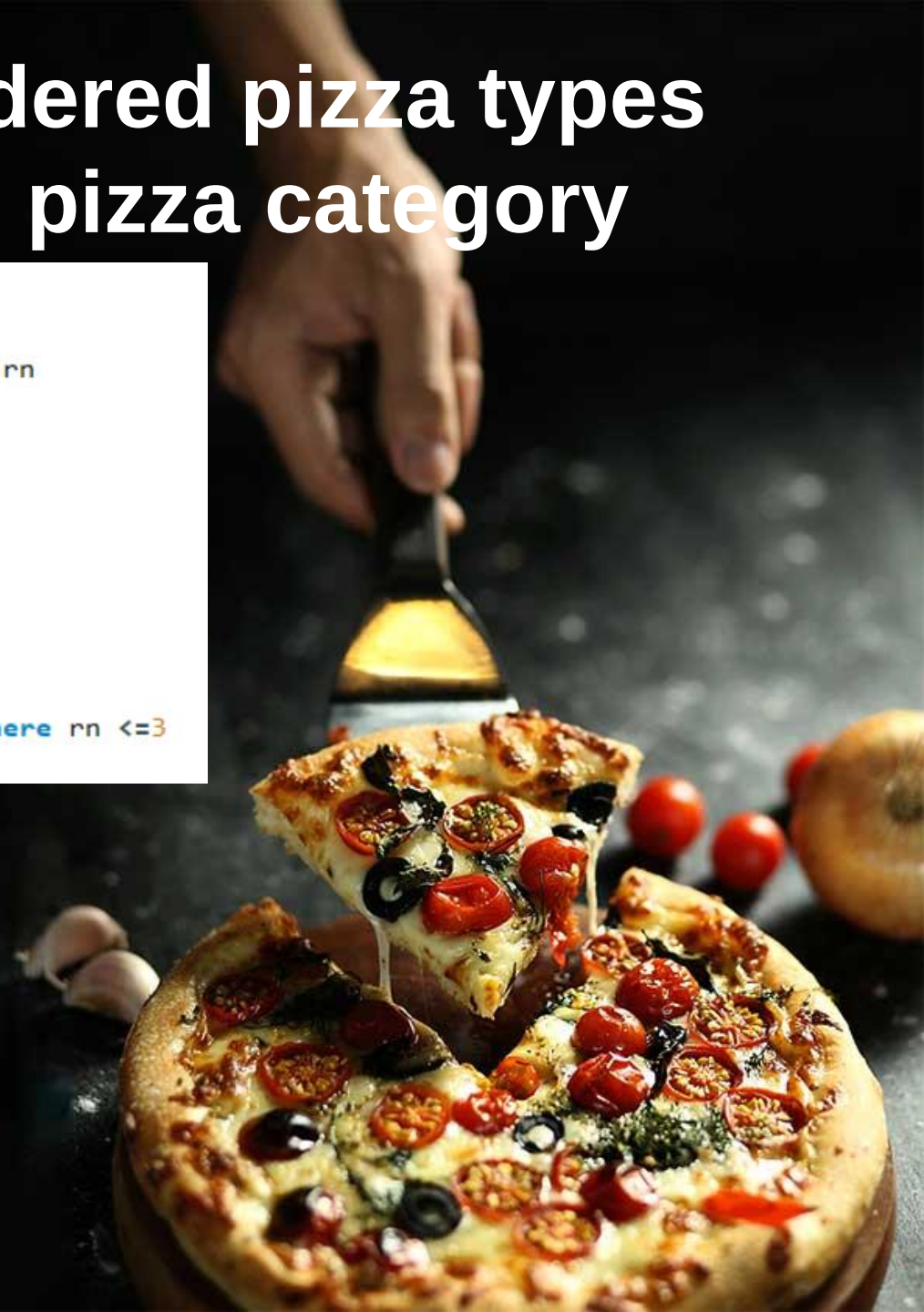
```sql
select order_date,
sum(revenue) over(order by order_date) as cum_revenue
from
(select orders.order_date,
sum(orders_details.quantity * pizzas.price) as revenue
from orders_details join pizzas
on orders_details.pizza_id  = pizzas.pizza_id
join orders
on orders.order_id  = orders_details.order_id
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
2015-01-12	27781.7
2015-01-13	29831.300000000003
2015-01-14	32358.700000000004
2015-01-15	34343.50000000001
2015-01-16	36937.65000000001

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(orders_details.quantity * pizzas.price) as revenue
from pizza_types join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
join orders_details
on orders_details.pizza_id = pizzas.pizza_id
group by pizza_types.category,pizza_types.name ) 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



Suggestion and solution

- **Promote Top-Selling Pizzas:** Focus on marketing and promotions for the top 3 revenue-generating pizza types in each category.

- **Use Revenue Trends:** Analyze revenue trends over time to optimize inventory, staffing, and marketing efforts during peak and off-peak times.

- **Optimize Pizza Sizes:** Streamline the menu by focusing on the most popular pizza sizes, potentially reducing costs and enhancing customer satisfaction.
- **Enhance Operational Efficiency:** Align staffing and preparation with peak order times to reduce wait times and improve service.

- **Refine Product Mix:** Focus on high-margin, high-demand pizzas while phasing out less popular options.

- **Maximize Customer Engagement:** Create loyalty programs or promotions based on the top 5 most ordered pizzas to encourage repeat purchases.

- **Strategic Pricing:** Introduce premium versions of popular pizzas and assess pricing strategies to maximize revenue.
- **Data-Driven Decision Making:** Regularly update data analysis to monitor sales performance and customer preferences for informed decision-making.


THANK YOU