PIZZA SALES PROJECT WITH SQL.

We would like to offer you a stylish and reasonable presentation that will help you to promote your business

Hello, My name is Tausiful Haque.

I am Driven and curious data analyst working with Python, SQL, Power BI, Excel and several more programming languages.

CONTENTS

Solved Basic,

Intermediate and

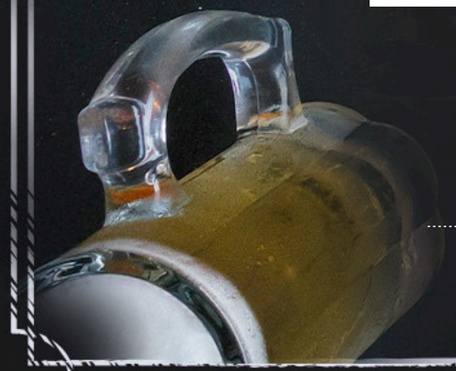
Advance questions from

SQL Language.

BASIC.

1. Retrieve the total number of orders placed.

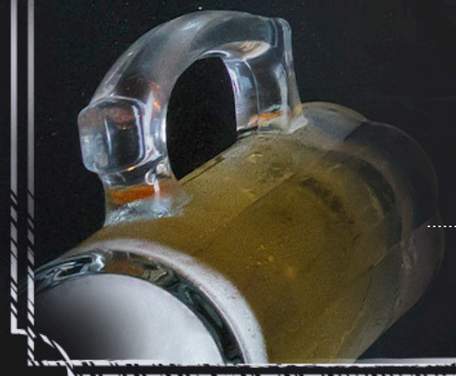
```
SELECT
    COUNT(order_id)
FROM
    orders;
```



	COUNT(order_id)
▶	21350

2. Calculate the total revenue generated from pizza sales.

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



total_sales
817860.05

3. Identify the highest-priced pizza.

```sql
SELECT
    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;
```

	name	price
▶	The Greek Pizza	35.95

4. Identify the most common pizza size ordered.

```sql
SELECT
    pizzas.size,
    SUM(order_details.order_details_id) AS pizza_count
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY pizza_count DESC
LIMIT 1;
```

	size	pizza_count
▶	L	449383379

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

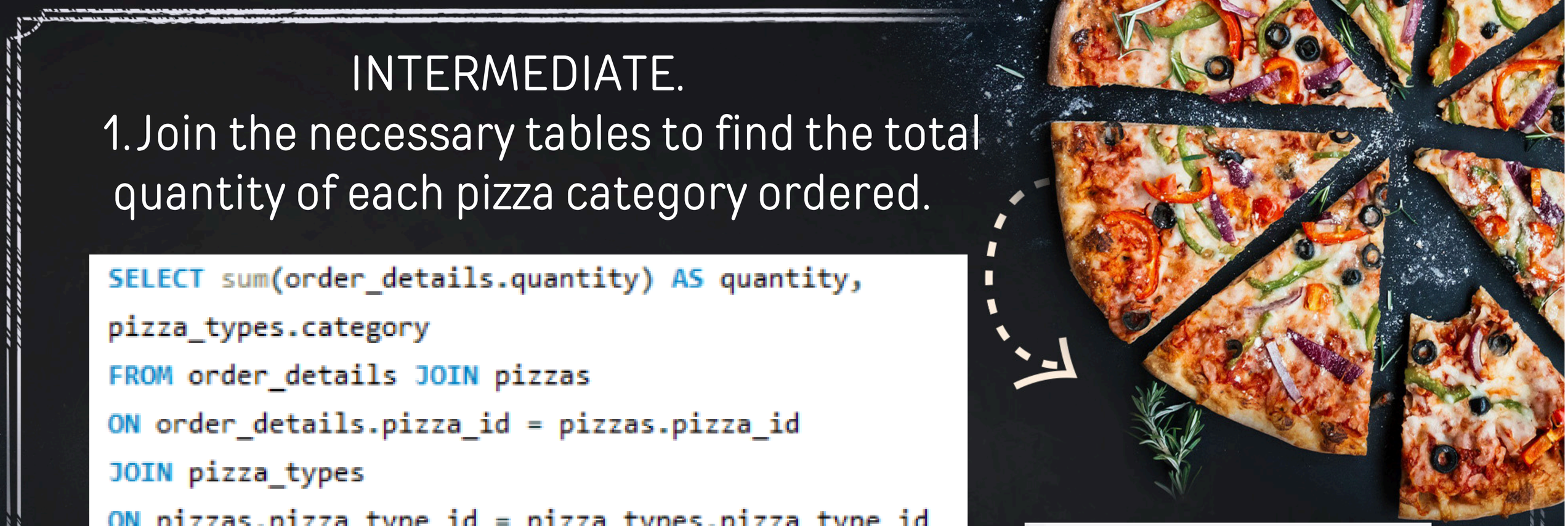
```sql
SELECT
    pizza_types.name,
    SUM(order_details.quantity) AS total_quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizza_types.name
ORDER BY total_quantity 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

INTERMEDIATE.

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

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

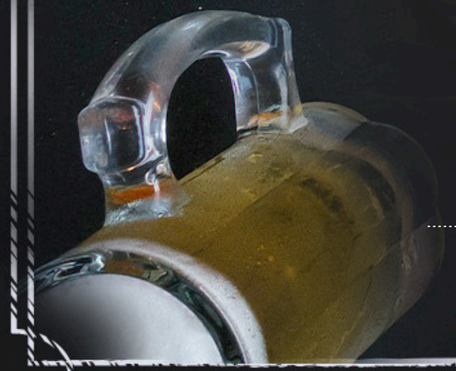


	quantity	category
▶	14888	Classic
	11987	Supreme
	11649	Veggie
	11050	Chicken

2. Determine the distribution of orders by hour of the day.

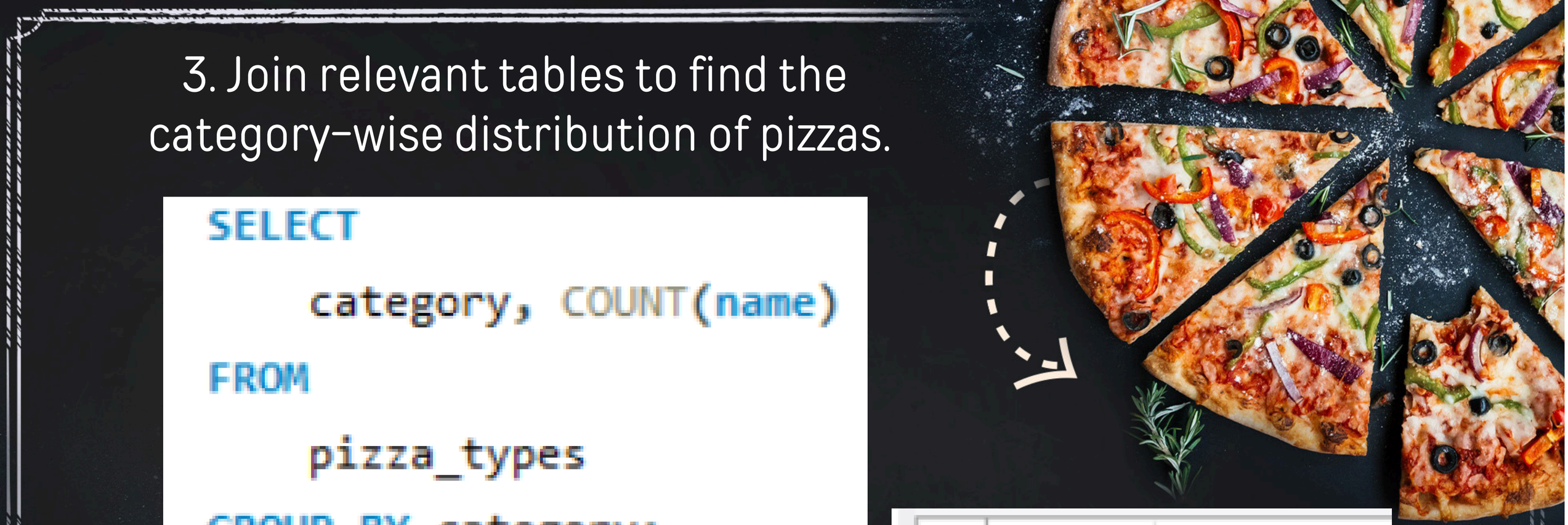
```sql
SELECT
    HOUR(order_time) AS hour, COUNT(order_id) AS distribution
FROM
    orders
GROUP BY hour;
```

hour	distribution
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642

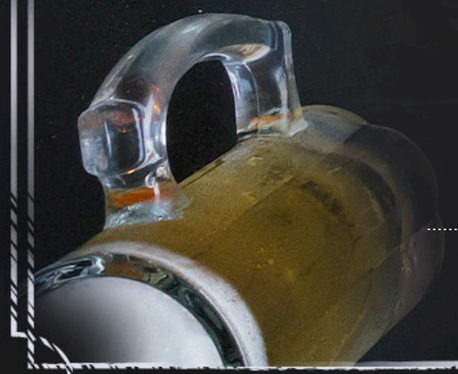


3. Join relevant tables to find the category-wise distribution of pizzas.

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

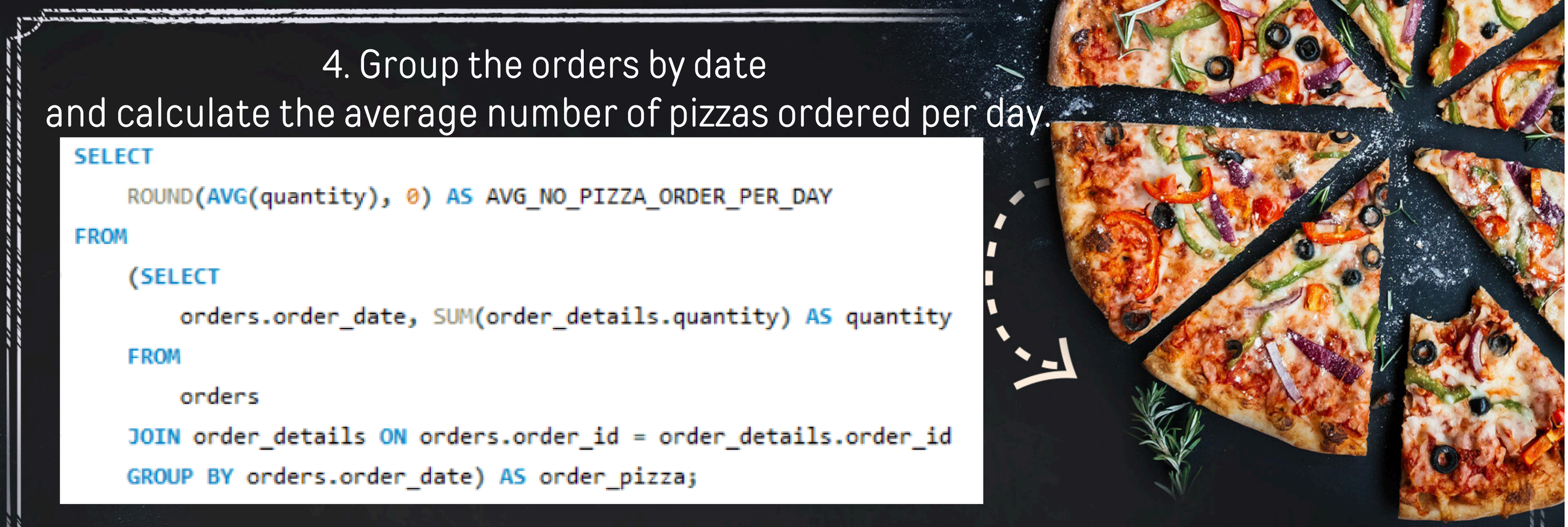


	category	COUNT(name)
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

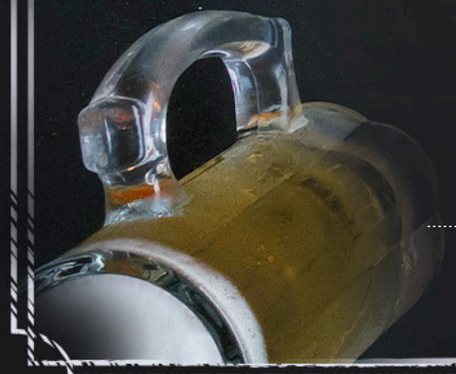


4. Group the orders by date and calculate the average number of pizzas ordered per day.

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

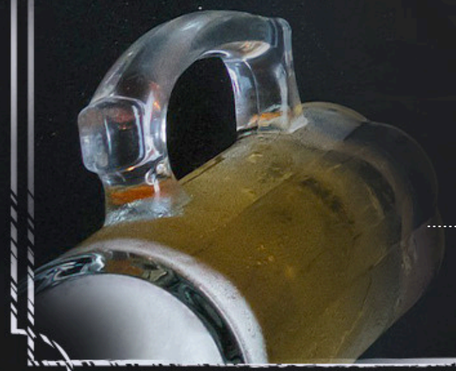


AVG_NO_PIZZA_ORDER_PER_DAY
138



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

```sql
SELECT
    pizza_types.name,
    SUM(pizzas.price * order_details.quantity) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON pizzas.pizza_id = order_details.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

ADVANCE.

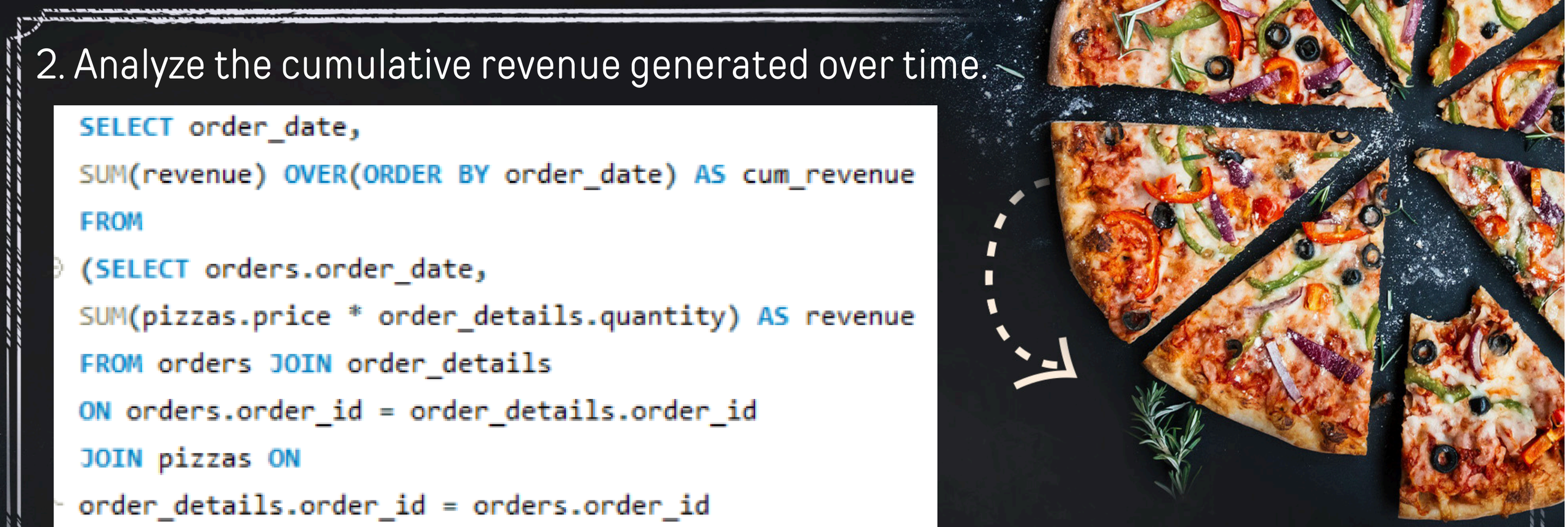
1. Calculate the percentage contribution of each pizza type to total revenue.

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

	category	revenue
▶	Classic	26.91
	Veggie	23.68
	Supreme	25.46
	Chicken	23.96

2. 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(pizzas.price * order_details.quantity) AS revenue
FROM orders JOIN order_details
ON orders.order_id = order_details.order_id
JOIN pizzas ON
order_details.order_id = orders.order_id
GROUP BY orders.order_date) AS sales;
```



order_date	cum_revenue
2015-01-01	255684.60000000085
2015-01-02	516104.1000000017
2015-01-03	765475.5000000023
2015-01-04	932775.3000000021
2015-01-05	1130062.8000000024


THANK
YOU
9905251964
tausiful540@gmail.com