

PIZZA SALES



INTRODUCTION

Hello my self ajay

Let's start our adventure in the world of pizza!

-- Retrieve the total number of orders placed.

SELECT

COUNT(orders_id)

FROM

orders;

COUNT(orders_id)

21350



-- Calculate the total revenue generated from pizza sales.

SELECT

ROUND(SUM(quantity * price), 2) AS total_sales

FROM

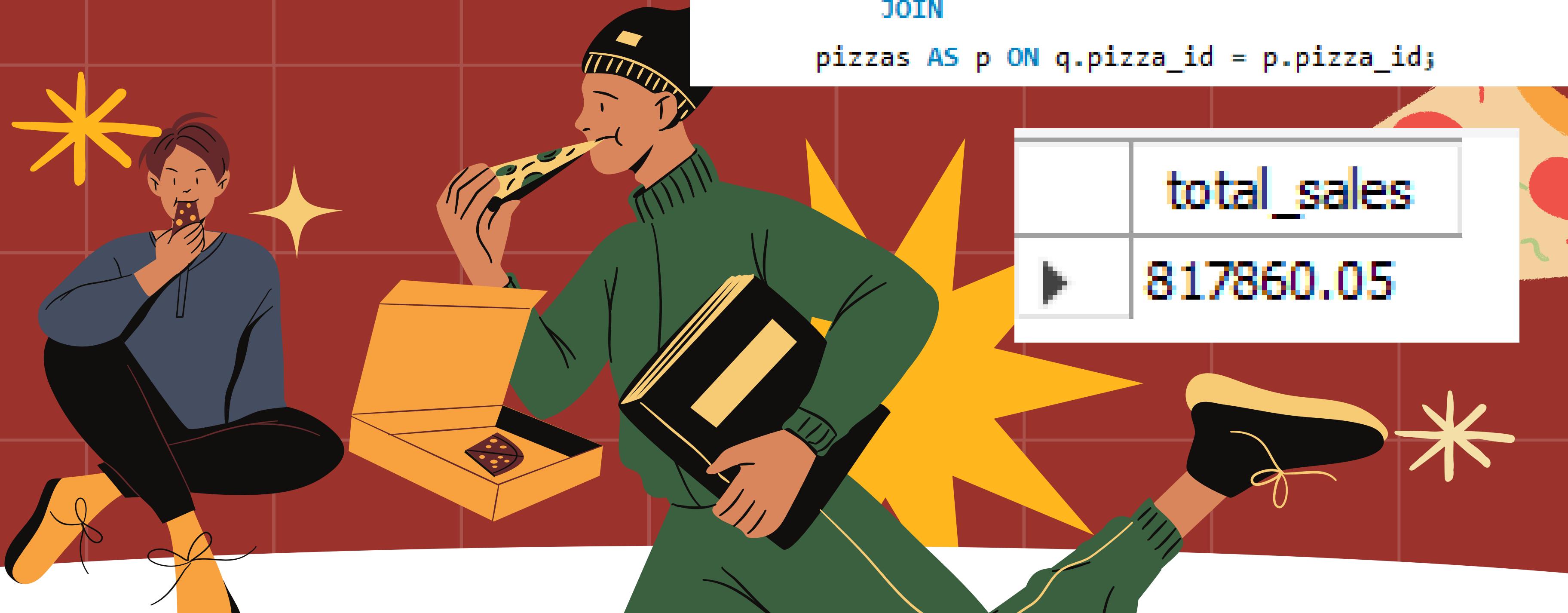
order_details AS q

JOIN

pizzas AS p ON q.pizza_id = p.pizza_id;

total_sales

817860.05



-- Identify the most common pizza size ordered.

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

j

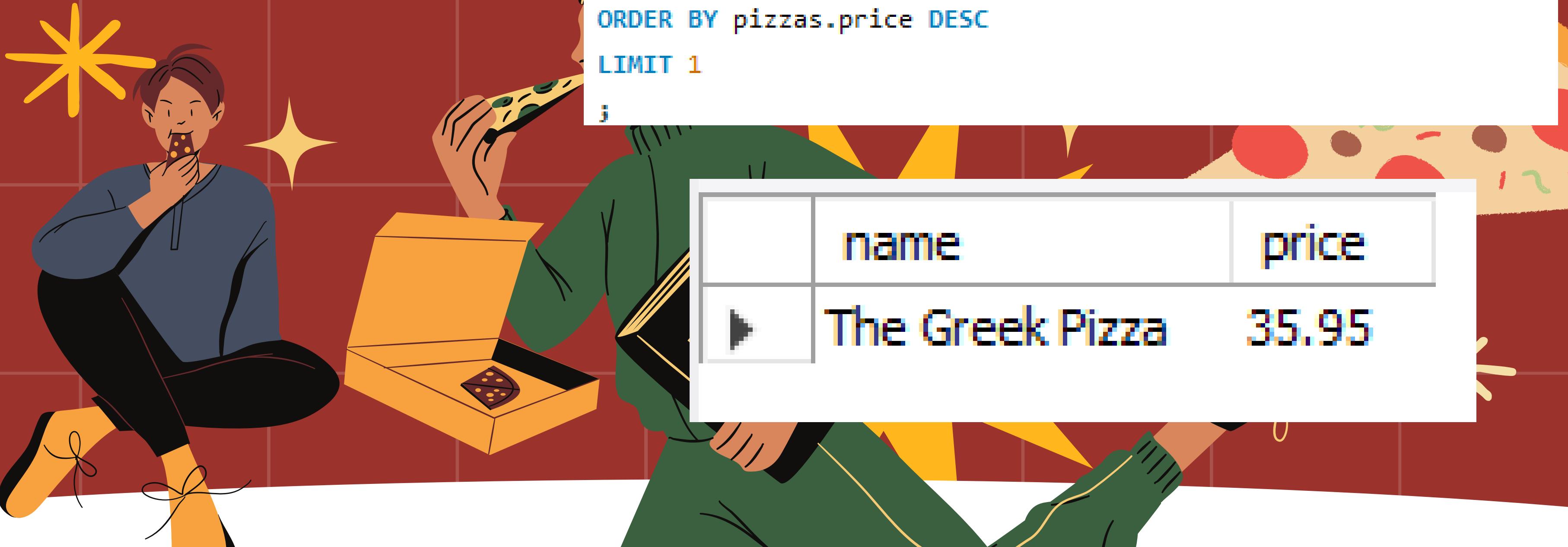
| | size | quantity |
|---|------|----------|
| ↓ | L | 18526 |



-- Identify the highest-priced pizza.

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

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



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

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

| | name | qua |
|---|----------------------------|------|
| ▶ | 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.

SELECT

pizza_types.category, SUM(order_details.quantity) qua

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

ORDER BY qua DESC;

| Category | qua |
|----------|-------|
| Classic | 14888 |
| Supreme | 11987 |
| Veggie | 11649 |
| Chicken | 11050 |

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

SELECT

HOUR(orders_time), COUNT(orders_id)

FROM

orders

GROUP BY HOUR(orders_time);



| HOUR(orders_time) | COUNT(orders_id) |
|-------------------|------------------|
| 11 | 1231 |
| 12 | 2520 |
| 13 | 2455 |
| 14 | 1472 |
| 15 | 1468 |
| 16 | 1920 |
| 17 | 2336 |
| 18 | 2399 |
| 19 | 2009 |
| 20 | 1647 |

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

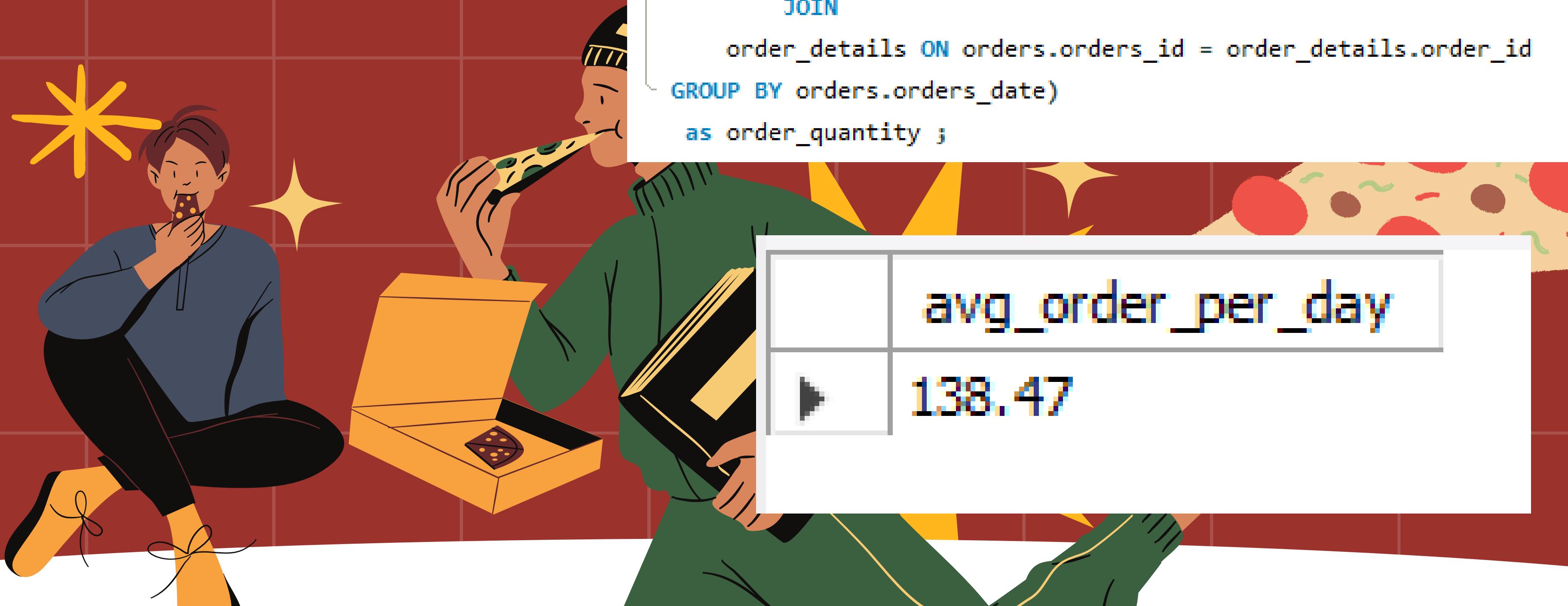


```
SELECT  
    pizza_types.category, COUNT(name)  
FROM  
    pizza_types  
GROUP BY category;
```

| | 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.

```
select round(avg(quantity),2) as avg_order_per_day from  
  (SELECT  
    orders.orders_date, SUM(order_details.quantity) AS quantity  
  FROM  
    orders  
  JOIN  
    order_details ON orders.orders_id = order_details.order_id  
  GROUP BY orders.orders_date)  
  as order_quantity ;
```



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

```
SELECT  
    pizza_types.name,  
    SUM(order_details.quantity * pizzas.price) AS revanu  
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 revanu DESC  
LIMIT 3;
```

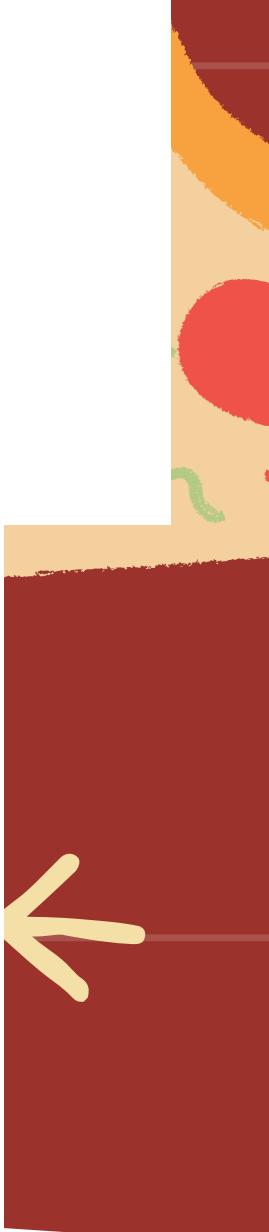
| | name | revanu |
|---|------------------------------|----------|
| ▶ | The Thai Chicken Pizza | 43434.25 |
| ▶ | The Barbecue Chicken Pizza | 42768 |
| ▶ | The California Chicken Pizza | 41409.5 |

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

```
• SELECT  
    pizza_types.category,  
    round( SUM(order_details.quantity * pizzas.price) / (SELECT  
        ROUND(SUM(quantity * price), 2) AS total_sales  
    )  
    FROM  
        order_details AS q  
        JOIN  
            pizzas AS p ON q.pizza_id = p.pizza_id) * 100,2) as revanu  
    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 revanu DESC;
```



| | category | revanu |
|---|----------|--------|
| ▶ | Classic | 26.91 |
| | Supreme | 25.46 |
| | Chicken | 23.96 |
| | Veggie | 23.68 |



--- Analyze the cumulative revenue generated over time

```
select orders_date, round(sum(revanu) over(order by orders_date),2)as cum_revamu from  
@ (SELECT  
    orders.orders_date,  
    SUM(order_details.quantity * pizzas.price) AS revanu  
FROM  
    orders  
    JOIN  
    order_details ON orders.orders_id = order_details.order_id  
    JOIN  
    pizzas ON pizzas.pizza_id = order_details.pizza_id  
GROUP BY orders.orders_date ) as sale ;
```

| orders_date | cum_revamu |
|-------------|------------|
| 2015-01-01 | 2713.85 |
| 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 | 23000.0 |



-- Determine the top 3 most ordered pizza types based on revenue for each pizza category.

```
select pizza_types.category, pizza_types.name, revanu, rank()
over(partition by pizza_types.category order by revanu desc)as rn from
(SELECT
pizza_types.category,
pizza_types.name,
SUM(order_details.quantity * pizzas.price) AS revanu
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 , pizza_types.name) as bb
```

