Retrieve the total number of orders placed.

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

total_orders
▶ 21350

Result Grid

Calculate the total revenue generated from pizza sales.

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

Result Grid

	total_sales
▶	817860.05

Identify the highest-priced pizza.

```sql
SELECT
    p.name, pp.price
FROM
    pizza_types p
        JOIN
    pizzas pp USING (pizza_type_id)
ORDER BY pp.price DESC
LIMIT 1;
```

	name	price
▶	The Greek Pizza	35.95

Result Grid | Filter

Identify the most common pizza size ordered.

```sql
SELECT
    p.size, COUNT(o.order_details_id) AS order_count
FROM
    pizzas p
        JOIN
    order_details o USING (pizza_id)
GROUP BY p.size
ORDER BY order_count DESC
LIMIT 1
```

Result Grid	
size	order_count
L	18526

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

```sql
SELECT
    pt.name, SUM(o.quantity) AS qty
FROM
    pizza_types pt
        JOIN
    pizzas p USING (pizza_type_id)
        JOIN
    order_details o USING (pizza_id)
GROUP BY pt.name
ORDER BY qty DESC
LIMIT 5;
```

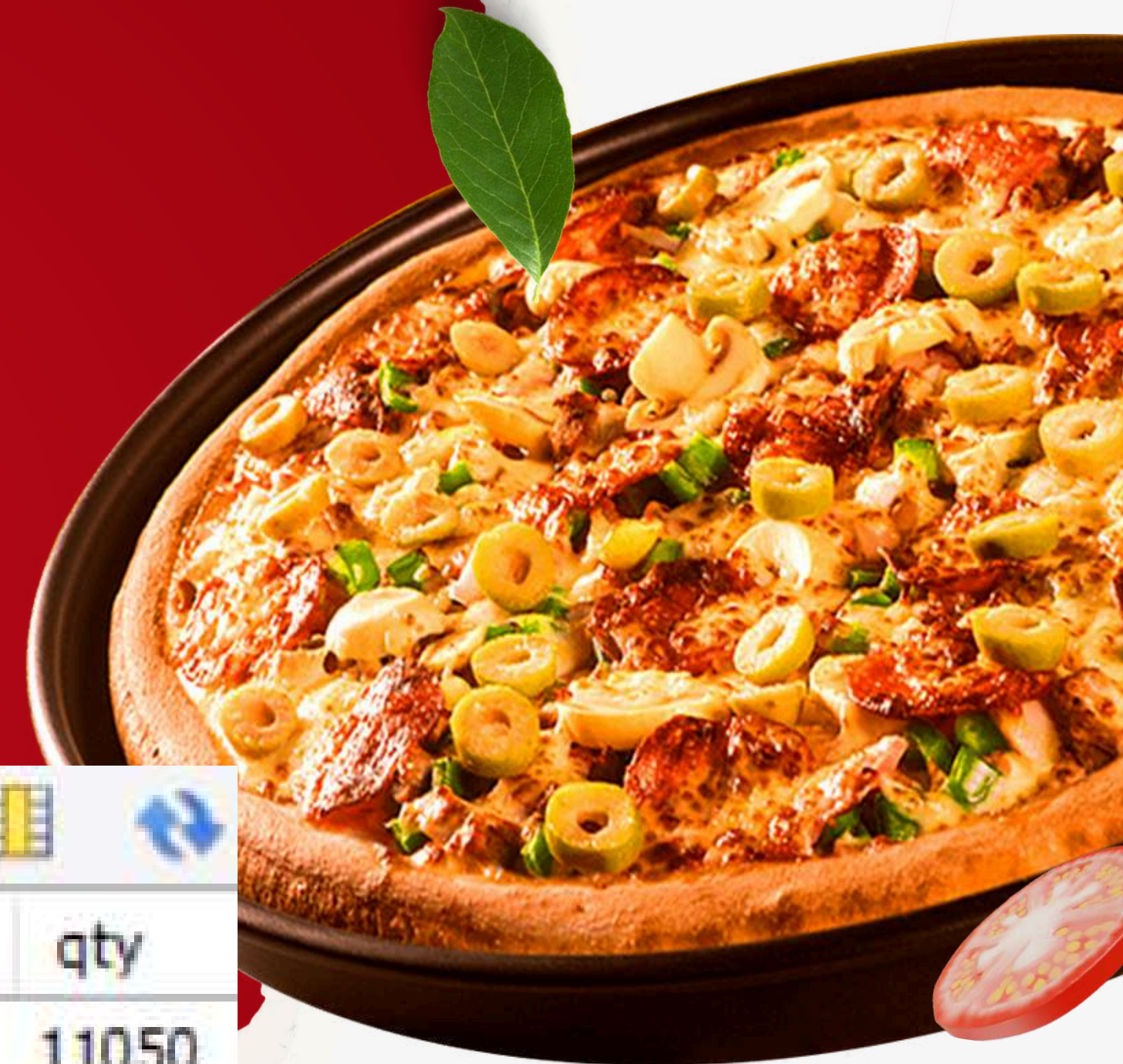|	name	qty
▶	The Classic Deluxe Pizza	2453
	The Barbecue Chicken Pizza	2432
	The Hawaiian Pizza	2422
	The Pepperoni Pizza	2418
	The Thai Chicken Pizza	2371

Result Grid | Filter Rows:

Find the total quantity of each pizza category ordered.

```sql
SELECT
    pt.category, SUM(o.quantity) AS qty
FROM
    pizza_types pt
        JOIN
    pizzas p USING (pizza_type_id)
        JOIN
    order_details o USING (pizza_id)
GROUP BY pt.category
ORDER BY qty;
```

Result Grid

category	qty
Chicken	11050
Veggie	11649
Supreme	11987
Classic	14888

Determine the distribution of orders by hour of the day.

```sql
select hour(order_time) as hour,
count(order_id) as order_count
from orders
group by hour(order_time);
```

Result Grid

hour	order_count
18	2399
19	2009
20	1642
21	1198
22	663
23	28
10	8
9	1

Result Grid

hour	order_count
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399

Find the category-wise distribution of pizzas.

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

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

find the total orders of each pizza category.

```sql
SELECT
    pt.category, COUNT(o.order_details_id) AS order_count
FROM
    pizza_types pt
        JOIN
    pizzas USING (pizza_type_id)
        JOIN
    order_details o USING (pizza_id)
GROUP BY pt.category;
```

Result Grid | Filter

category	order_count
Classic	14579
Veggie	11449
Supreme	11777
Chicken	10815

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

```sql
SELECT
    ROUND(AVG(qty), 0) AS avg_pizza
FROM
    (SELECT
        o.order_date, SUM(od.quantity) AS qty
    FROM
        orders o
    JOIN order_details od USING (order_id)
    GROUP BY o.order_date) AS order_qty;
```

Result Grid	
	avg_pizza
▶	138

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

```sql
SELECT
    pt.name, SUM(od.quantity * p.price) AS revenue
FROM
    pizza_types pt
        JOIN
    pizzas p USING (pizza_type_id)
        JOIN
    order_details od USING (pizza_id)
GROUP BY pt.name
ORDER BY revenue DESC
LIMIT 3;
```

Result Grid | Filter Rows:

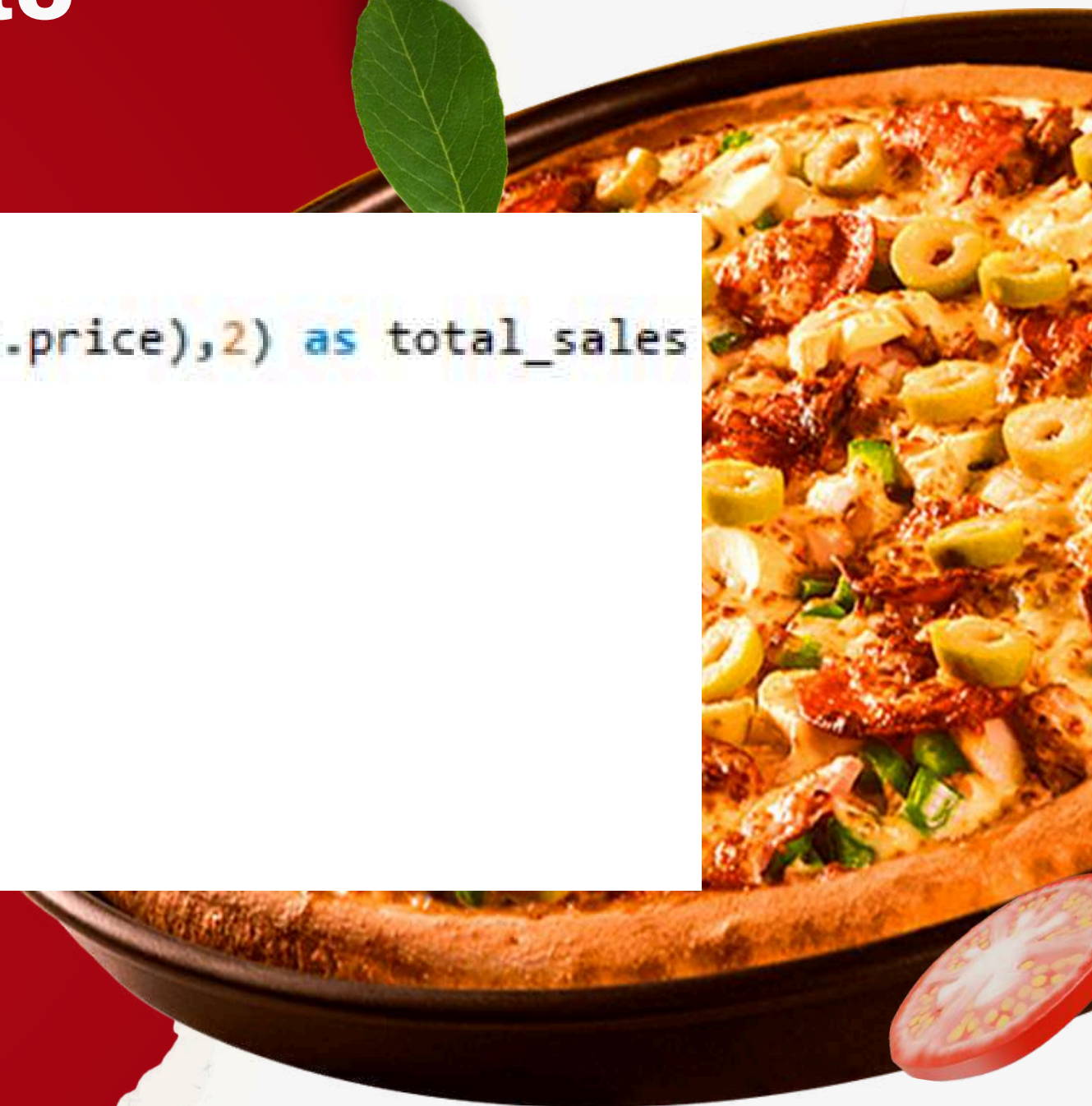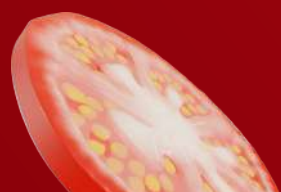| name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

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

```sql
select pt.category,
round(sum(od.quantity * p.price) / (select round(sum(o.quantity * p.price),2) as total_sales
from order_details o
join pizzas p using(pizza_id)) * 100, 2) as revenue
from pizza_types pt
join pizzas p using(pizza_type_id)
join order_details od using(pizza_id)
group by pt.category order by revenue desc;
```

Result Grid

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_revnue
from
(select o.order_date, sum(od.quantity * p.price)
as revenue from orders o
join order_details od using(order_id)
join pizzas p using(pizza_id)
group by o.order_date) as sales;
```

Result Grid	Filter Rows:
order_date	cum_revnue
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

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 rev
from
(select pt.category, pt.name, sum(od.quantity * p.price)
as revenue from pizza_types pt
join pizzas p using(pizza_type_id)
join order_details od using(pizza_id)
group by pt.category, pt.name) as a) as b
where rev <=3;
```

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

Result Grid | Filter Rows:

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

Result Grid | Filter Rows:

name	revenue
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

thank you!