

SQL PROJECT ON PIZZA-SALES



HELLO !



My name is ASHWNI. In this project, I worked on solving SQL queries related to pizza sales. Using SQL, I answered various questions such as finding total sales, identifying the best-selling pizzas, and checking customer orders. This helped in organizing and understanding the data better.

DATABASE

Pizzahut database include following tables:

order_details	
▼	Columns
◆	order_details_id
◆	order_id
◆	pizza_id

orders	
▼	Columns
◆	order_id
◆	order_date
◆	order_time

pizzas	
▼	Columns
◆	pizza_id
◆	pizza_type_id
◆	size
◆	price

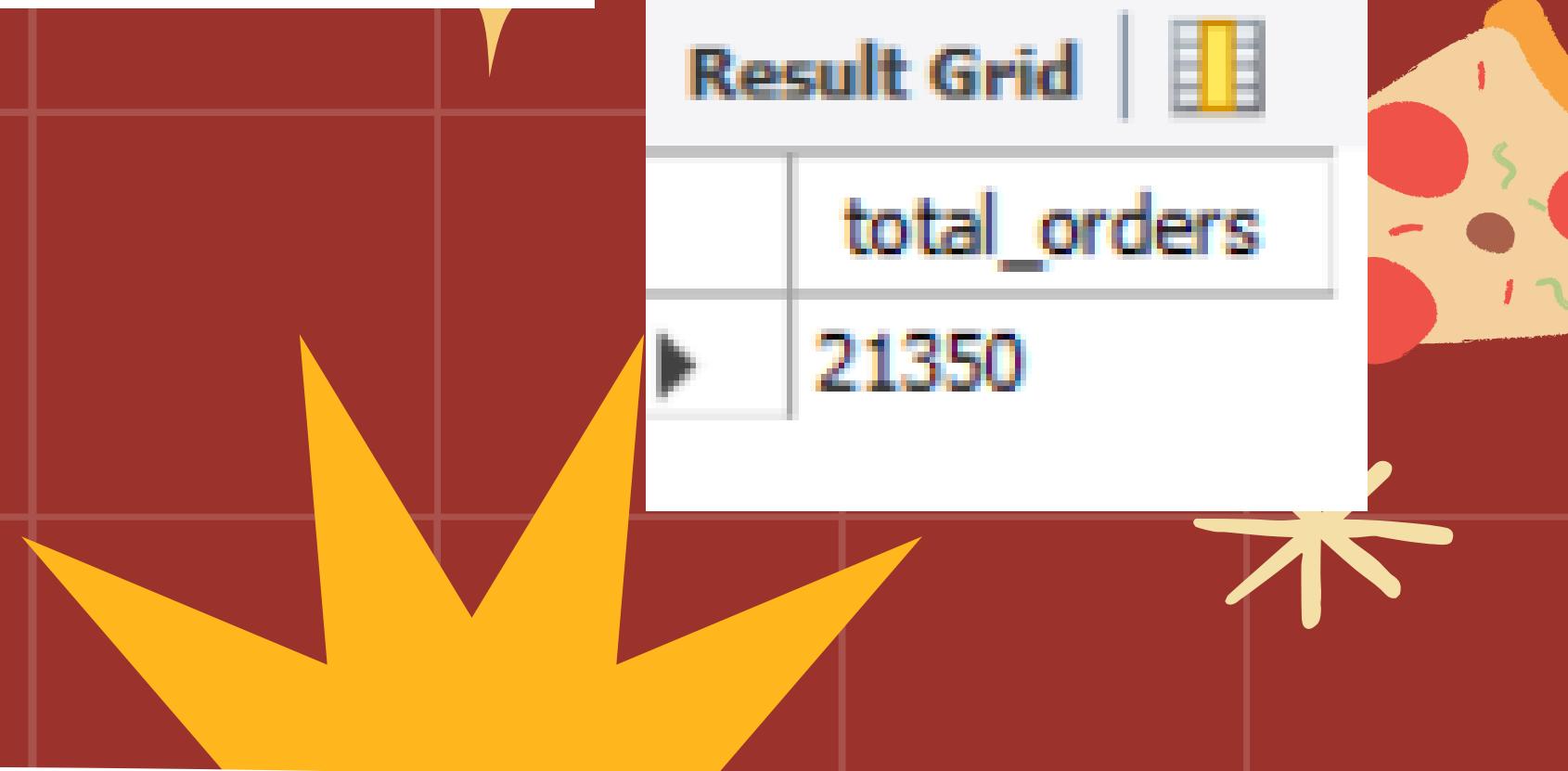
pizza_types	
▼	Columns
◆	pizza_type_id
◆	name
◆	category
◆	ingredients

1) RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

```
use pizzahut;  
  
SELECT  
    COUNT(order_id) AS total_orders  
FROM  
    orders;
```

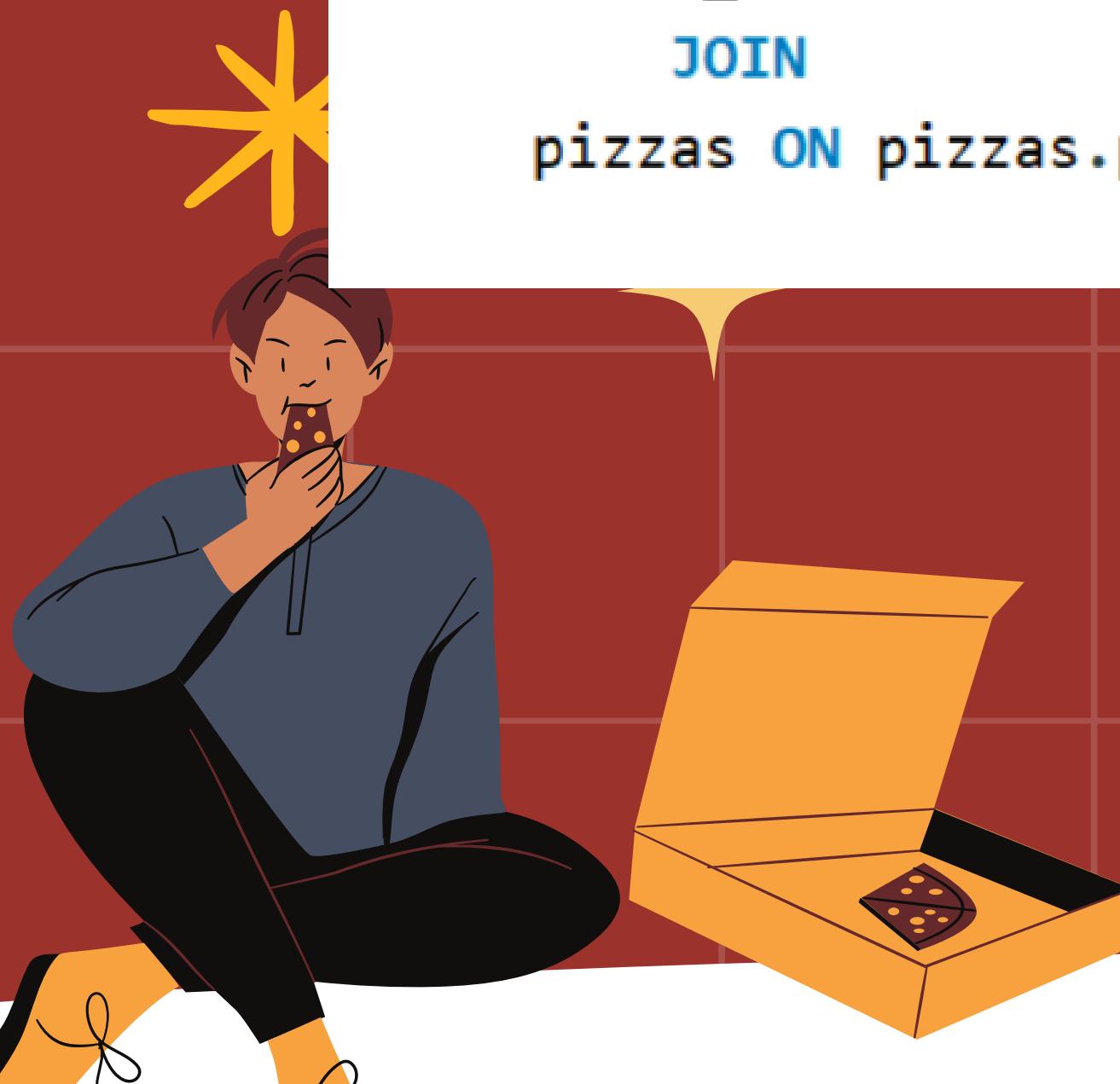


Result Grid	
	total_orders
→	21350

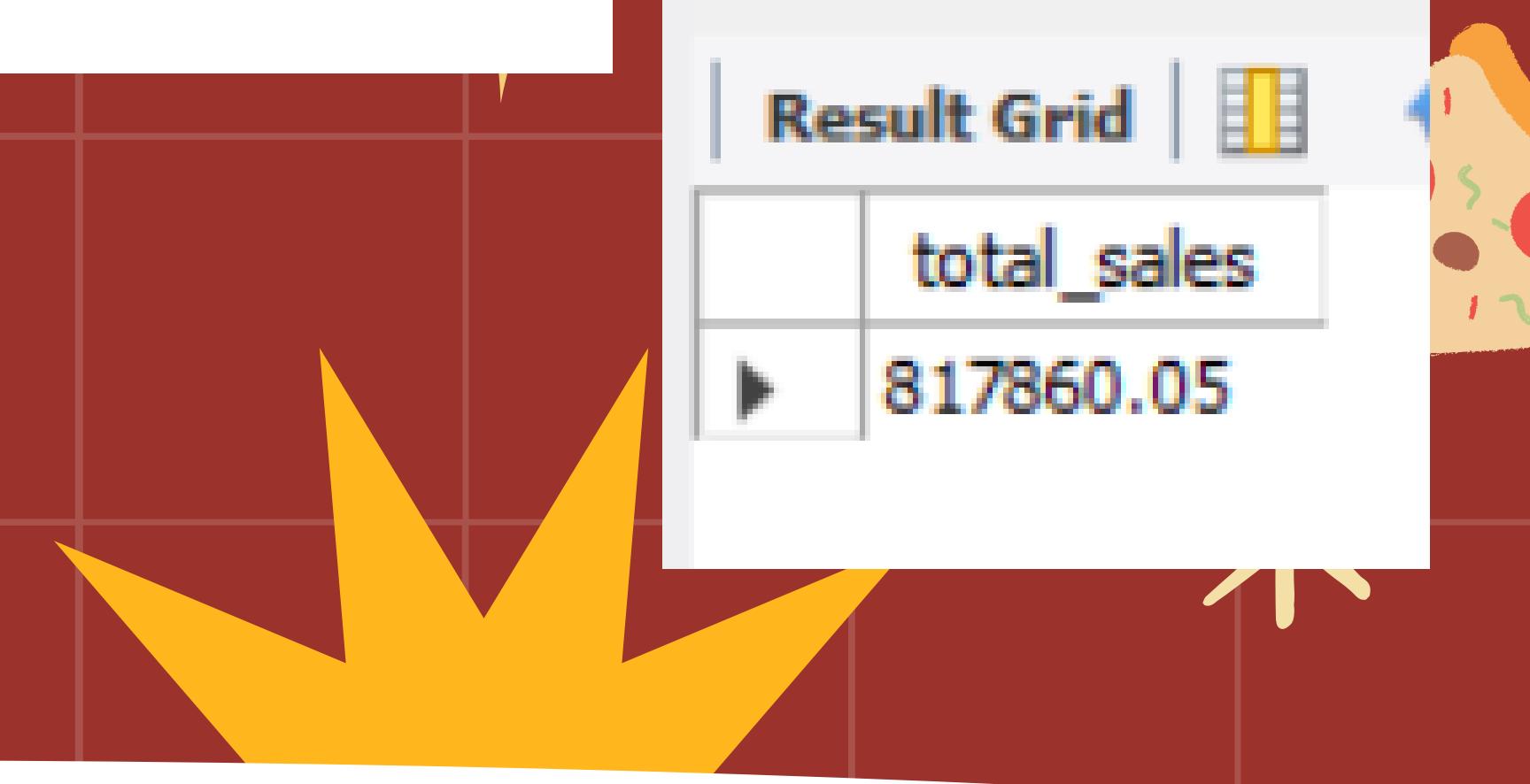


2) CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES

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



Result Grid	
	total_sales
▶	817860.05



3) IDENTIFY THE HIGHEST-PRICED PIZZA.

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



Result Grid | Filter Rows:

	name	price
▶	The Greek Pizza	35.95

4) IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

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

Result Grid | Filter Rows

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



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

```
SELECT  
    pizza_types.name, SUM(order_details.quantity) AS quantity  
FROM  
    pizza_types  
        JOIN  
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id  
        JOIN  
    order_details ON order_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



6) JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED.

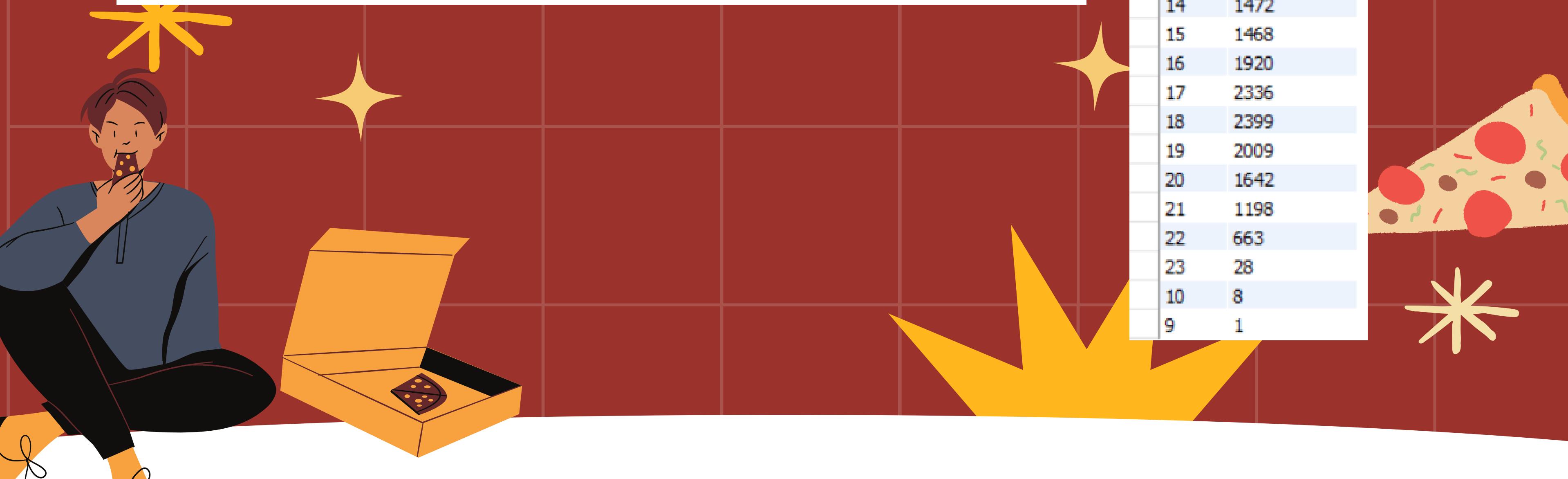
```
SELECT  
    pizza_types.category,  
    SUM(order_details.quantity) AS quantity  
FROM  
    pizza_types  
        JOIN  
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id  
        JOIN  
    order_details ON order_details.pizza_id = pizzas.pizza_id  
GROUP BY pizza_types.category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

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

7) DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

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

hour	order_count
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642
21	1198
22	663
23	28
10	8
9	1



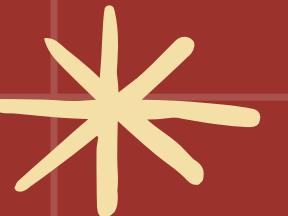
8) JOIN RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS.

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



Result Grid | Filter Rows:

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



9) GROUP THE ORDERS BY DATE CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY.

```
SELECT  
    ROUND(AVG(quantity), 0) as avg_pizza_ordered_perday  
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_quantity;
```

Result Grid	Filter Rows:
avg_pizza_ordered_perday	
138	

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

SELECT

```
    pizza_types.name,  
    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  
ORDER BY revenue DESC  
LIMIT 3;
```

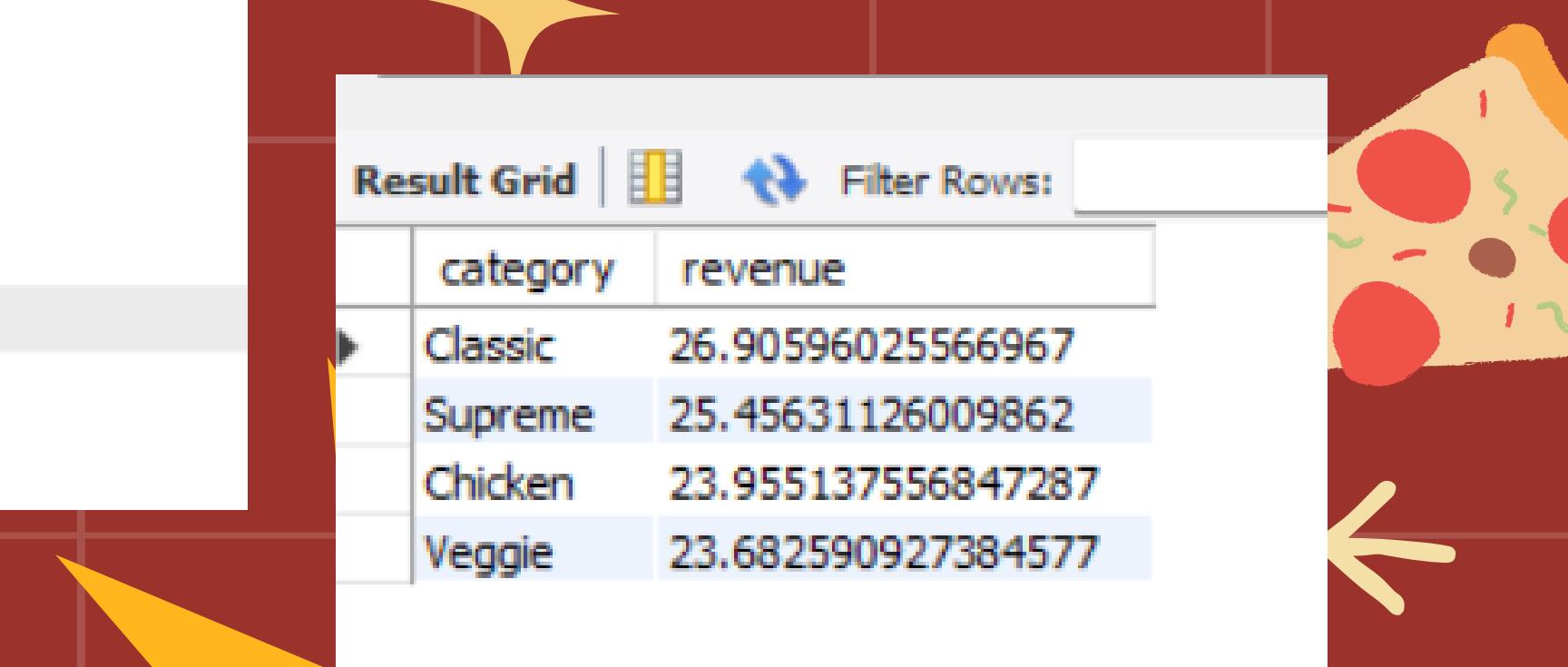
Result Grid | Filter Rows:

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

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

```
SELECT
    pizza_types.category,
    (SUM(order_details.quantity * pizzas.price) / (SELECT
        ROUND(SUM(order_details.quantity * pizzas.price),
        2) AS total_sales
    )
    FROM
        order_details
        JOIN
            pizzas ON pizzas.pizza_id = order_details.pizza_id)) * 100 AS revenue
FROM
    pizza_types
    JOIN
        pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
    JOIN
        order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY revenue DESC;
```

category	revenue
Classic	26.90596025566967
Supreme	25.45631126009862
Chicken	23.955137556847287
Veggie	23.682590927384577



12) 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 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



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

```
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(order_details.quantity* pizzas.price) as revenue
  from pizza_types join pizzas
  on pizza_types.pizza_type_id=pizzas.pizza_type_id
  join order_details
  on order_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	32265.7000000065
	The Mexicana Pizza	26780.75
	The Five Cheese Pizza	26066.5



THANK YOU

“Have fun making your own pizza
and enjoy every bite”