



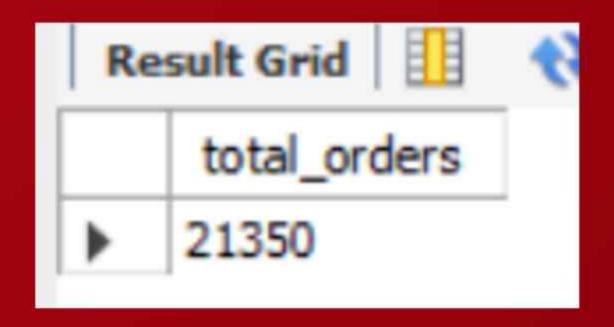
# THE TOTAL NUMBER OF ORDER PLACED

```
SELECT

COUNT(order_id) AS total_orders

FROM

orders;
```



# CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES

```
SELECT

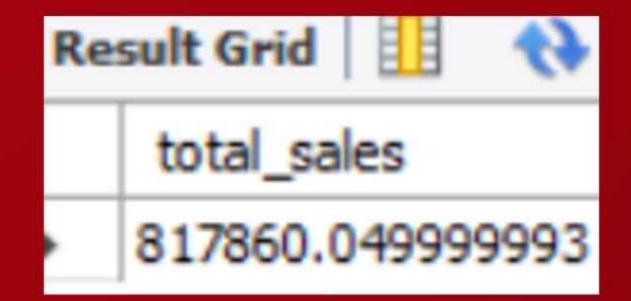
SUM(order_details.quantity * pizzas.price) AS total_sales

FROM

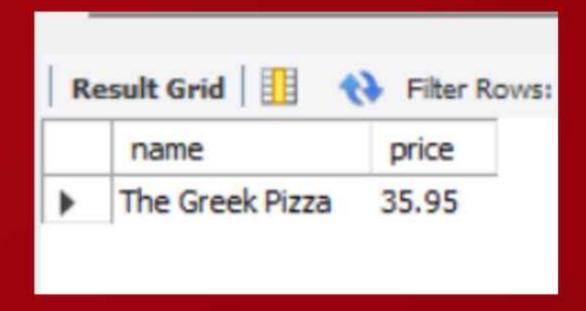
order_details

JOIN

pizzas ON pizzas.pizza_id = order_details.pizza_id
```



#### IDENTIFY THE HIGHEST-PRICED PIZZAS.



#### IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

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

#### LIST THE TOP 5 MOST ORDERED PIZZA TYPE ALONG WITH THEIR QUANTITIES.

Result Grid Filter Rows:				
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			
- 21				

## 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;
```

Re	esult Grid	Filter I
	category	quantity
<b>&gt;</b>	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

## DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY

```
SELECT

HOUR(order_time), COUNT(order_id) AS order_count

FROM

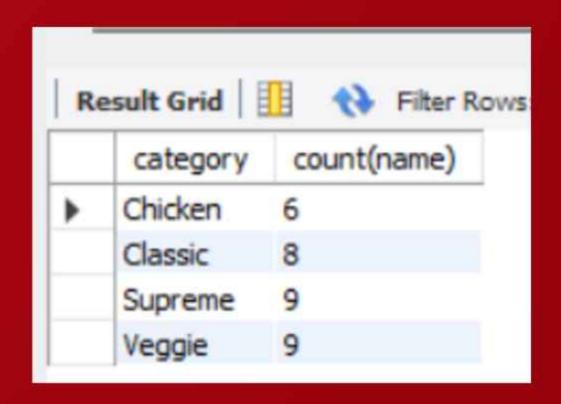
orders

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;
```



### GROUP THR ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY.

```
AVG(quantity) as avg_pizza_ordered_per_day

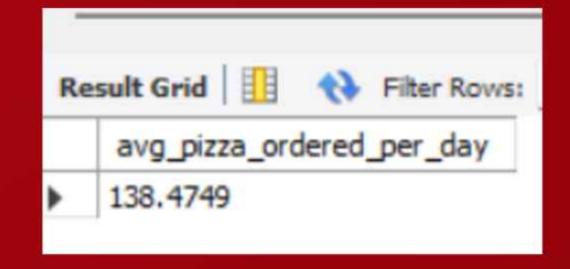
FROM

(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_quantity;
```



### 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;
```



## DETERMINE THR 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
```

	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

#### Summary of this topic:

- To solve the pizza sales use the mysql workbench.
- Use select to find the total orders, identify orders.
- And also use sum, cout, avg etc to count pizza and sum the quantity of pizza.
- Use group by, order by, where to find the quantity of pizza sales.
- use join to identify most common pizza order and many more.

