DATA ANALYSIS FULL PORTFOLIO PROJECT

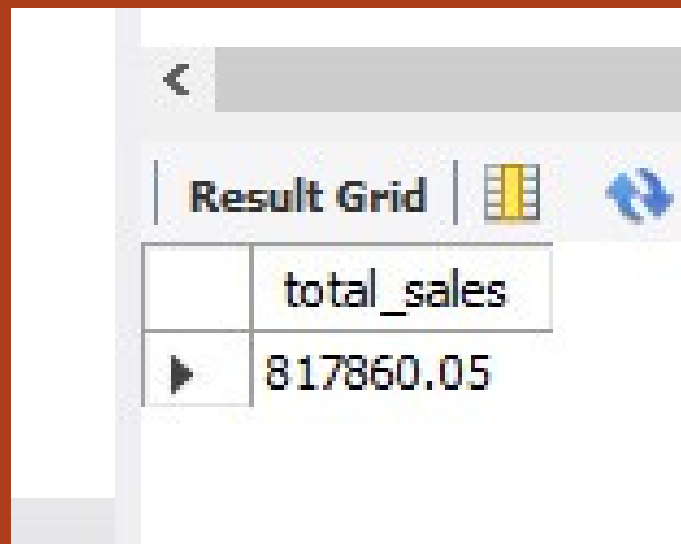# PIZZA

```sql
1    -- Retrieve the total number of orders placed.
2
3 •  select count(order_id)  from orders ;
```

count(order_id)
21350

```sql
-- 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 order_details.pizza_id = pizzas.pizza_id
```

total_sales
817860.05

```sql
-- Identify the highest-priced pizza.
select
pizza_types.name , pizzas.price
from pizza_types join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
order by pizzas.price desc limit 1;
```

| Result Grid | Filter Rows: |
name	price
▶ The Greek Pizza	35.95

```sql
1    -- Identify the most common pizza size ordered.
2 •  select
3    pizzas.size , count(order_details.quantity)
4    from pizzas join order_details
5    on pizzas.pizza_id = order_details.pizza_id
6    group by pizzas.size;
7
```

	size	count(order_details.quantity)
▶	M	15385
	L	18526
	S	14137
	XL	544
	XXL	28

Result Grid · Filter Rows:

```sql
-- 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 4;
```

name	quantity
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422

```sql
-- 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 pizza_types.pizza_type_id = pizzas.pizza_type_id
group by pizza_types.category order by quantity desc limit 4;
```

Result Grid | Filter Rows

quantity	category
14888	Classic
11987	Supreme
11649	Veggie
11050	Chicken

```sql
-- Determine the distribution of orders by hour of the day.
SELECT
    HOUR(order_time) AS order_hour,
    COUNT(order_id) AS order_count
FROM
    orders
GROUP BY
    HOUR(order_time) ORDER BY order_count DESC LIMIT 10;
```

Result Grid | Filter Rows:

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

```
-- 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

Result Grid | Filter Rows:

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

select round(sum(quantity),0) 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:

round(sum(quantity),0)
49574

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

SELECT
pizza_types.name AS pizza_name,
SUM(pizzas.price * order_details.quantity) AS total_revenue
FROM pizzas
JOIN pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id
JOIN order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizza_types.name
ORDER BY total_revenue DESC LIMIT 3;
```

Result Grid | Filter Rows:

pizza_name	total_revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

```sql
-- 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 rm
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 rm <=3;
```

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
The Sicilian Pizza	30940.5
The Four Cheese Pizza	32265.7000000
The Mexicana Pizza	26780.75
The Five Cheese Pizza	26066.5

```sql
-- Analyze the cumulative revenue generated over time.
select order_date,
ROUND(SUM(revenue) OVER (ORDER BY order_date), 2) 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 BY order_date DESC
LIMIT 10;
```

Result Grid | Filter Rows:

order_date	cum_revenue
2015-12-31	817860.05
2015-12-30	814944.05
2015-12-29	813606.25
2015-12-28	812253
2015-12-27	810615.8
2015-12-26	809196.8
2015-12-24	807553.75