

# PIZZA SALES ANALYSIS



Q1. Retrieve the total number of orders placed.

```
SELECT  
    COUNT(order_id) AS Total_orders  
FROM  
    orders;
```

Total_orders
21350

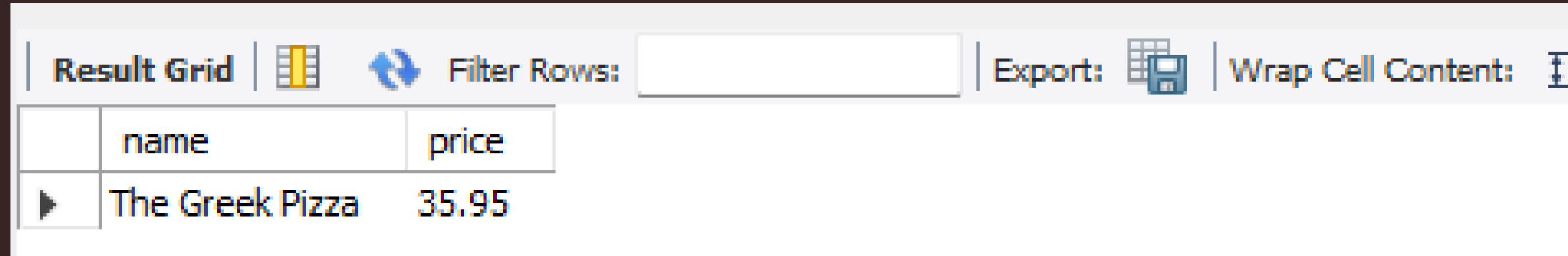
## Q2. Calculate the total revenue generated from pizza sales?

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

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
revenue				
▶	817860.05			

### Q3. Identify the highest priced pizza.

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



The screenshot shows a MySQL Workbench interface with a result grid. The grid has two columns: 'name' and 'price'. A single row is displayed, showing 'The Greek Pizza' with a price of '35.95'. The result grid tab is selected at the top left. There are also tabs for 'Filter Rows:' and 'Export:'.

	name	price
▶	The Greek Pizza	35.95

# Q4. Identify the most common pizza size ordered?

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

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

## Q5. List the top 5 most ordered pizza types along with their quantities.

```
SELECT pt.name, SUM(od.quantity) AS quantity
FROM pizza_types pt
    JOIN pizzas p ON pt.pizza_type_id = p.pizza_type_id
    JOIN order_details od ON p.pizza_id = od.pizza_id
GROUP BY pt.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

Q6. Join the necessary tables to find the total quantity of each pizza category ordered.

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

	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

## Q7. 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 BY order_count DESC;
```

	hour	order_count
▶	12	2520
	13	2455
	18	2399
	17	2336
	19	2009
	16	1920
	20	1642
	14	1470

# Q8. Join relevant tables to find the category-wise distribution of pizzas?

```
SELECT  
    category, COUNT(name) AS pizza_name  
FROM  
    pizza_types  
GROUP BY category  
ORDER BY pizza_name DESC;
```

	category	pizza_name
▶	Supreme	9
	Veggie	9
	Classic	8
	Chicken	6

Q9. Group the orders by date and calculate the average number of pizzas ordered per day?

```
SELECT
    ROUND(AVG(quantity), 0) AS avg_num_of_pizzas_per_day
FROM
    (SELECT
        o.order_date, SUM(od.quantity) AS quantity
    FROM
        orders o
    JOIN order_details od ON o.order_id = od.order_id
    GROUP BY o.order_date) AS order_quantity
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	avg_num_of_pizzas_per_day			
▶	138			

# Q10. Determine the top 3 most ordered pizza types based on revenue?

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

Result Grid | Filter Rows:  Export: Wrap Cell Content: Fetch rows:

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

# Q11. Calculate the percentage contribution of each pizza type to total revenue?

```
SELECT
    pt.category,
    round((SUM(od.quantity * p.price) / (SELECT
        SUM(od.quantity * p.price)
    FROM
        pizzas p
    JOIN
        order_details od ON p.pizza_id = od.pizza_id)) * 100,2) AS pct_contribution
FROM
    pizza_types pt
    JOIN
        pizzas p ON pt.pizza_type_id = p.pizza_type_id
    JOIN
        order_details od ON p.pizza_id = od.pizza_id
GROUP BY pt.category
ORDER BY pct_contribution DESC;
```

Result Grid | Filter Rows:  Export: Wrap Cell Content:

	category	pct_contribution
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

## Q12. Analyze the cumulative revenue generated over time.

```
select
order_date, sum(revenue) OVER (ORDER BY order_date) AS cum_revenue
FROM
(SELECT
    o.order_date, SUM(od.quantity * p.price) AS revenue
FROM
    orders o
        JOIN
    order_details od ON o.order_id = od.order_id
        JOIN
    pizzas p ON od.pizza_id = p.pizza_id
GROUP BY o.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

# Q13. Determine the top 3 most ordered pizza types based on revenue for each pizza category.

```
SELECT category, name, revenue
FROM
(SELECT category, name, revenue, RANK() OVER(PARTITION BY category ORDER BY revenue DESC) AS rank_
FROM
(SELECT
    pt.category, pt.name, SUM(od.quantity * p.price) AS revenue
FROM
    pizza_types pt
        JOIN
    pizzas p ON pt.pizza_type_id = p.pizza_type_id
        JOIN
    order_details od ON od.pizza_id = p.pizza_id
GROUP BY pt.category , pt.name) AS a) AS b
WHERE rank_ <= 3;
```

	category	name	revenue
▶	Chicken	The Thai Chicken Pizza	43434.25
	Chicken	The Barbecue Chicken Pizza	42768
	Chicken	The California Chicken Pizza	41409.5
	Classic	The Classic Deluxe Pizza	38180.5
	Classic	The Hawaiian Pizza	32273.25

**THANK  
YOU**

