Did you know that data can reveal the secrets to pizza perfection?

'Pizza Sales Data Analysis using SQL' delves deep into pizza order data, uncovering hidden trends and serving up a slice of actionable insights that can revolutionize the way you think about pizza.

Pizza TIME

Q1. Retrieve the total number of orders placed :-

Query

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

Result

Result Grid

total_orders
21350

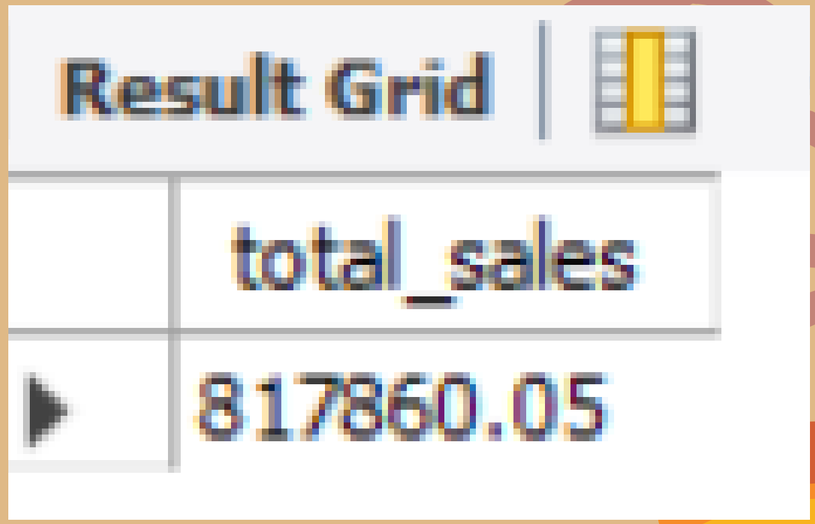
Pizza TIME

Q2. Calculate the total revenue generated from pizza sales:-

Query

```
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



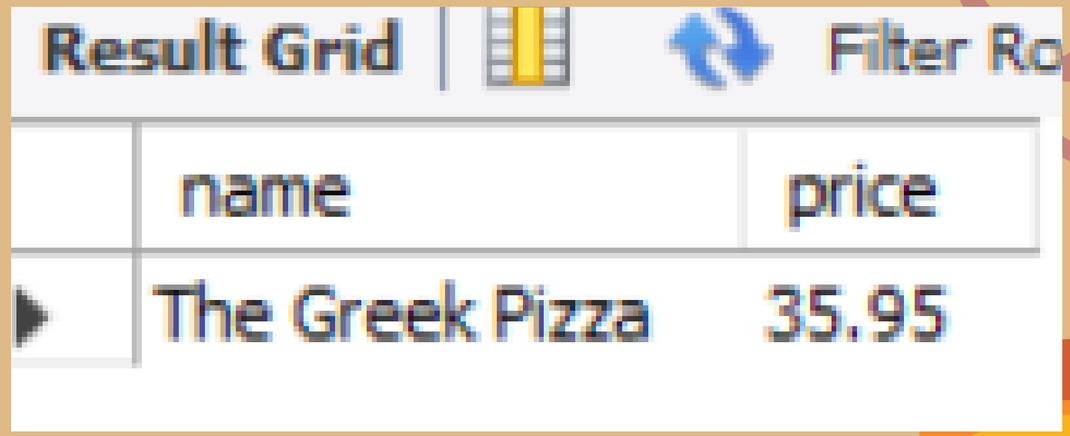
Pizza TIME

Q3. Identify the highest priced pizza :-

Query

```
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



name	price
The Greek Pizza	35.95

Pizza TIME

Q4. Identify the most common pizza size ordered :-

Query

```
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

Result Grid | Filter Rows

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

Pizza TIME

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

Query

```
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;
```

Result

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

Pizza TIME

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

Query

```
SELECT
    pizza_types.category,
    SUM(order_details.quantity) AS total_quantity_ordered
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.category
ORDER BY total_quantity_ordered DESC;
```

Result

Result Grid | Filter Rows:

category	total_quantity_ordered
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Pizza TIME

Q7. Determine the distribution of orders by hour of the day :-

Query

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

Result

Result Grid | Filter Rows: | Export:

hour	order_count
9	1
10	8
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

Pizza TIME

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

Query

```
SELECT
    category AS Category, COUNT(name) AS Distribution_of_Pizzas
FROM
    pizza_types
GROUP BY category
```

Result

Result Grid | Filter Rows:

Category	Distribution_of_Pizzas
Chicken	6
Classic	8
Supreme	9
Veggie	9

Pizza TIME

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

Query

```
SELECT
    ROUND(AVG(pizzas_ordered_per_day)) AS Avg_Pizzas_ordered
FROM
    (SELECT
        orders.order_date,
            SUM(order_details.quantity) AS pizzas_ordered_per_day
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY orders.order_date) AS order_quantity;
```

Result

Result Grid | Filter

Avg_Pizzas_ordered
138

Pizza TIME

Q10. Determine the top 3 most ordered pizza types based on revenue :-

Query

```sql
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

Result Grid | Filter Rows:

name	Revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

Pizza TIME

Q11. Calculate the percentage contribution of each pizza type to total revenue :-

Query

```
SELECT
    pizza_types.category,
    ROUND(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,
            2) 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.category
ORDER BY Revenue DESC;
```

Result

Result Grid | Filter Rows:

category	Revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

Pizza TIME

Q12. Analyze the cumulative revenue generated over time :-

Query

```
select order_date,
round(sum(revenue) over(order by order_date),2) as cumulative_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;
```

Result

Result Grid | Filter Rows:

order_date	cumulative_revenue
2015-01-01	2713.85
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
2015-01-08	19399.05

Pizza TIME

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

Query

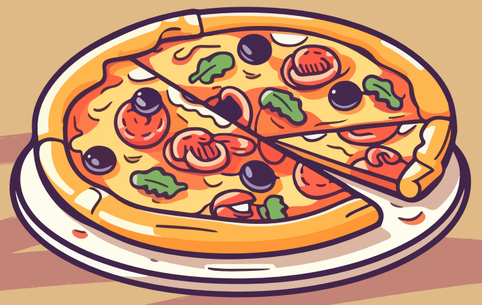
```
select category, name, revenue from

(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;
```

Result

Result Grid | Filter Rows: | Export: | Wra

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
Classic	The Pepperoni Pizza	30161.75
Supreme	The Spicy Italian Pizza	34831.25
Supreme	The Italian Supreme Pizza	33476.75
Supreme	The Sicilian Pizza	30940.5
Veggie	The Four Cheese Pizza	32265.70000000065
Veggie	The Mexicana Pizza	26780.75
Veggie	The Five Cheese Pizza	26066.5



Thank you !