Data Analysis - MySQL

An in-depth analysis of sales report of a Pizza company

Pizza Sales Analysis using SQL

As part of this project, I have used a sales report of a pizza company as part of my independent learning on my journey to becoming a Data Analyst/Data Engineer. Here is my effort from a basic to advanced approach.

Q1. Calculate the total revenue generated from pizza sales

```sql
-- calculate the total revenue generated from pizza sales (using two different tables)
SELECT
    ROUND(SUM(order_details.quantity * pizzas.price)) AS total_sales
FROM
    order_details
        JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id;

```

Result Grid

total_sales
817860

Q2. Identify the highest-priced pizza.

```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;
```

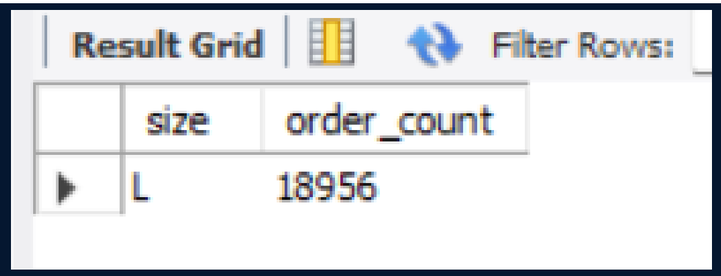
Result Grid | Filter Rows:

	name	price
▶	The Greek Pizza	35.95

Q3. Identify the most common pizza size ordered.

```sql
-- identify the most common size ordered

SELECT
    p.size, sum(o.quantity) AS order_count
FROM
    pizzas AS p
        JOIN
    order_details AS o ON p.pizza_id = o.pizza_id
GROUP BY p.size
ORDER BY order_count DESC limit 1;
```



Q4. List the top 5 most ordered pizza types along with their quantities.

```sql
-- select top 5 most ordered pizza types by their quantities
SELECT
    pt.name, SUM(o.quantity) AS quantities
FROM
    pizza_types AS pt
        JOIN
    pizzas AS p ON pt.pizza_type_id = p.pizza_type_id
        JOIN
    order_details AS o ON p.pizza_id = o.pizza_id
GROUP BY pt.name
ORDER BY quantities DESC
LIMIT 5;
```

Result Grid | Filter Rows:

name	quantities
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

Q5. Join the necessary tables to find the total quantity of each pizza category ordered.

```sql
-- Join necessaery tables to find the total quantity of each pizza category ordered

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

Result Grid | Filter Rows:

category	SUM(o.quantity)
Classic	14888
Veggie	11649
Supreme	11987
Chicken	11050

Q6.Determine the distribution of orders by hour of the day.

```sql
-- determine the distribution of orders by hour of the day

SELECT
    HOUR(order_time), COUNT(order_id)
FROM
    orders
GROUP BY HOUR(order_time) order by COUNT(order_id) desc;
```

Result Grid | Filter Rows:

HOUR(order_time)	COUNT(order_id)
12	2520
13	2455
18	2399
17	2336
19	2009
16	1920
20	1642
14	1472
15	1468
11	1231
21	1198
22	663
23	28
10	8
9	1

Q7. Group the orders by date and calculate the average number of pizzas ordered per day.

```sql
-- group the orders by date and calculate the average number of pizzas ordered per day
use pizzahut;
SELECT
    ROUND(AVG(quantity), 0) AS avg_pizzas
FROM
    (SELECT
        o.order_date, SUM(od.quantity) AS quantity
    FROM
        orders AS o
    JOIN order_details AS od ON o.order_id = od.order_id
    GROUP BY o.order_date) AS order_quantity;
```

Result Grid | Filter Rows:

	avg_pizzas
▶	138

Q8. Determine the top 3 most ordered pizza types based on revenue.

```sql
-- Determine the top 3 most ordered pizza types based on revenue.
SELECT
    pt.category, ROUND(SUM(od.quantity * p.price), 2) AS revenue
FROM
    pizza_types AS pt
        JOIN
    pizzas AS p ON p.pizza_type_id = pt.pizza_type_id
        JOIN
    order_details AS od ON p.pizza_id = od.pizza_id
GROUP BY pt.category
ORDER BY revenue DESC
LIMIT 3;
```

Result Grid | Filter Row

category	revenue
Classic	220053.1
Supreme	208197
Chicken	195919.5

Q9. Calculate the percentage contribution of each pizza type to total revenue.

```sql
-- calculate percentage of contribution of every pizza to the total revenue
SELECT
    pt.name,
    ROUND(SUM(p.price * od.quantity) / (SELECT
                    ROUND(SUM(od.quantity * p.price), 2) AS total_sales
                FROM
                    order_details AS od
                        JOIN
                    pizzas AS p ON p.pizza_id = od.pizza_id) * 100,
            2) AS perc_contri
FROM
    pizza_types AS pt
        JOIN
    pizzas AS p ON p.pizza_type_id = pt.pizza_type_id
        JOIN
    order_details AS od ON od.pizza_id = p.pizza_id
GROUP BY pt.name;
```

Result Grid | Filter Rows:

name	perc_contri
The Hawaiian Pizza	3.95
The Classic Deluxe Pizza	4.67
The Five Cheese Pizza	3.19
The Italian Supreme Pizza	4.09
The Mexicana Pizza	3.27
The Thai Chicken Pizza	5.31
The Prosciutto and Arugula Pizza	2.96
The Barbecue Chicken Pizza	5.23
The Greek Pizza	3.48
The Spinach Supreme Pizza	1.87
The Green Garden Pizza	1.71
The Italian Capocollo Pizza	3.07
The Spicy Italian Pizza	4.26
The Spinach Pesto Pizza	1.91
The Vegetables + Vegetables Pi...	2.98
The Southwest Chicken Pizza	4.24
The California Chicken Pizza	5.06
The Pepperoni Pizza	3.69
The Chicken Pesto Pizza	2.04
The Big Meat Pizza	2.81
The Soppressata Pizza	2.01
The Four Cheese Pizza	3.95
The Napolitana Pizza	2.95
The Calabrese Pizza	1.95
The Italian Vegetables Pizza	1.96

Q10. Analyze the cumulative revenue generated over time.

```sql
-- analyze the cumulative revenue generated over time

select order_date, round(sum(cum_revenue) over(order by order_date),2) as cumulative_running_revenue
from (select o.order_date,round(sum(od.quantity * p.price),2) as cum_revenue
    from
        orders as o join order_details as od on o.order_id=od.order_id
        join pizzas as p on od.pizza_id=p.pizza_id group by o.order_date) as total_sales;
```

Result Grid | Filter Rows:

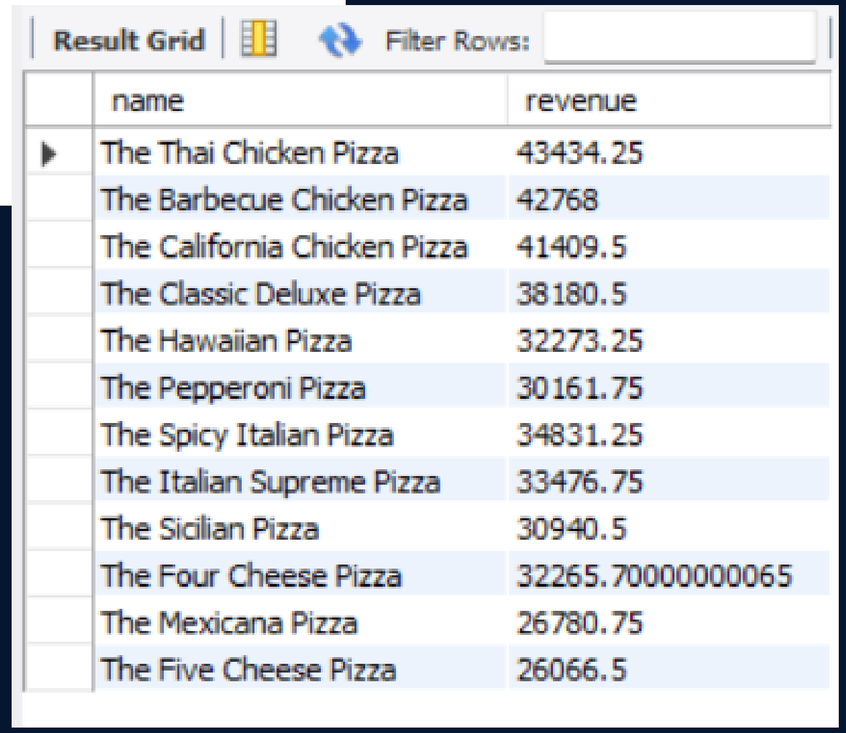
order_date	revenue_per_day
2015-01-01	2713.85
2015-01-02	2731.9
2015-01-03	2662.4
2015-01-04	1755.45
2015-01-05	2065.95
2015-01-06	2428.95
2015-01-07	2202.2

Result Grid | Filter Rows:

order_date	cumulative_running_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
2015-01-09	21526.4

Q11. Determine the top 3 most ordered pizza types based on revenue for each pizza category.

```sql
select name,revenue from
    (select name, category, revenue, rank() over(partition by category order by revenue desc)
    as rn from (select pt.name,pt.category,sum(p.price * od.quantity) as revenue from
    pizza_types as pt
      join
        pizzas as p on p.pizza_type_id=pt.pizza_type_id
      join
        order_details as od on od.pizza_id=p.pizza_id
        group by pt.name,pt.category) as a)
        as ab where rn <=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	32265.70000000065
The Mexicana Pizza	26780.75
The Five Cheese Pizza	26066.5

Thank you