sql project on pizza sales

-by menavath pavan kumar-

"Hello, my name is Menavath Pavan Kumar. I have utilized SQL (Structured Query Language) for performing data analysis on pizza sales. Through SQL, I was able to extract, filter, and manipulate the sales data effectively. This allowed me to gain valuable insights into various aspects of the business, such as identifying the best-selling pizzas, analyzing revenue trends, monitoring order frequencies, and understanding customer preferences. My analysis helped in uncovering patterns that could support data-driven decision-making for improving sales strategies and overall business performance."

CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

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

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;
```

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED

```
SELECT
    pizzas.size, COUNT(order_details.pizza_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 ;
```

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 5;
```

JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED.

```sql
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;
```

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

```sql
SELECT
    HOUR(order_time), COUNT(order_id)
FROM
    ordered
GROUP BY HOUR(order_time);
```

JOIN RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS.

```
SELECT
    category, COUNT(name)
FROM
    PIZZA_TYPES
GROUP BY category;
```

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

```
SELECT ROUND(avg(quantity),0) FROM

   (SELECT ordered.order_date, SUM(order_details.quantity) AS quantity
FROM
    ordered
        JOIN
    order_details ON ordered.order_id = order_details.order_id
GROUP BY ordered.order_date) AS order_quantity;
```

DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.

```sql
SELECT
    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.name
ORDER BY revenue DESC
LIMIT 3;
```

RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

```sql
SELECT
    COUNT(order_id) AS total_orders
FROM
    ordered;
```

CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE.

```sql
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 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
```

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

```
SELECT
order_date ,sum(revenue) over (ORDER BY order_date) AS cum_revenue
FROM
 (SELECT ordered.order_date,sum(order_details.quantity * pizzas.price) AS revenue
FROM
 order_details
 JOIN pizzas
 ON order_details.pizza_id=pizzas.pizza_id
JOIN
 ordered ON ordered.order_id=order_details.order_id
 GROUP BY ordered.order_date
 ORDER BY revenue DESC) AS sales;
```

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 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;
```

CONCLUSION

Based on the SQL analysis of pizza sales data, it was observed that non-vegetarian pizzas consistently outperformed vegetarian pizzas in terms of total sales volume and revenue. This indicates a stronger customer preference for non-veg options, suggesting that enhancing the variety and availability of non-vegetarian pizzas could further boost overall sales performance.

Thank you