# PIZZA SALES PROJECT - SQL

HI, I'M SHANAWAZ, AND I HAVE COMPLETED AN SQL PROJECT ANALYZING PIZZA SALES DATA. THIS PROJECT INVOLVED SOLVING VARIOUS QUERIES TO UNCOVER INSIGHTS, FROM BASIC ORDER TOTALS AND REVENUE CALCULATIONS TO ADVANCED ANALYSES LIKE REVENUE CONTRIBUTION AND CATEGORY-BASED PERFORMANCE, SHOWCASING MY SQL EXPERTISE.

# RETRIVE THE TOTAL NUMBER OF ORDERS PLACED

```
SELECT
    COUNT(ORDER_ID) AS TOTAL_ORDERS
FROM
    ORDERS;
```

# TOTAL REVENUE GENEREATED FROM PIZZA SALES

```sql
SELECT
    ROUND(SUM(PRICE * QUANTITY), 2) AS TOTAL_REVENUE
FROM
    PIZZAS
        JOIN
    ORDER_DETAILS ON PIZZAS.PIZZA_ID = ORDER_DETAILS.PIZZA_ID;
```

# IDENTIFY THE HIGHEST-PRICED PIZZA.

```sql
SELECT
    NAME, PRICE
FROM
    PIZZAS
        JOIN
    PIZZA_TYPES ON PIZZAS.PIZZA_TYPE_ID = PIZZA_TYPES.PIZZA_TYPE_ID
ORDER BY PRICE DESC
LIMIT 1;
```

# IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

```sql
SELECT
    SIZE AS COMMON_SIZE,
    COUNT(ORDER_DETAILS_ID) AS QUANTITY_ORDERED
FROM
    PIZZAS
        JOIN
    ORDER_DETAILS ON PIZZAS.PIZZA_ID = ORDER_DETAILS.PIZZA_ID
GROUP BY SIZE
ORDER BY QUANTITY_ORDERED DESC
LIMIT 1;
```

# LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES.

```sql
SELECT
    PIZZA_TYPES.PIZZA_TYPE_ID,
    NAME,
    SUM(ORDER_DETAILS.QUANTITY) AS QUANTITY
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.PIZZA_TYPE_ID, NAME
ORDER BY QUANTITY DESC
LIMIT 5;
```

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

```
SELECT
    CATEGORY AS PIZZA_CATEGORY, SUM(QUANTITY) AS QUANTITY
FROM
    PIZZA_TYPES
        JOIN
    PIZZAS ON PIZZA_TYPES.PIZZA_TYPE_ID = PIZZAS.PIZZA_TYPE_ID
        JOIN
    ORDER_DETAILS ON PIZZAS.PIZZA_ID = ORDER_DETAILS.PIZZA_ID
GROUP BY CATEGORY;
```

# DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

```sql
SELECT
    HOUR(TIME) AS ORDER_HOUR, COUNT(ORDER_ID) AS ORDER_COUNT
FROM
    ORDERS
GROUP BY ORDER_HOUR;
```

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

```
SELECT
    CATEGORY, COUNT(PIZZA_TYPE_ID)
FROM
    PIZZA_TYPES
GROUP BY CATEGORY;
```

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

```sql
SELECT
    ROUND(AVG(QUANTITY), 0)
FROM
    (SELECT
        DATE(ORDERS.DATE) AS ORDER_DATE, SUM(QUANTITY) AS QUANTITY
    FROM
        ORDERS
    JOIN ORDER_DETAILS ON ORDERS.ORDER_ID = ORDER_DETAILS.ORDER_ID
    GROUP BY ORDER_DATE) AS ORDER_QUANTITY;
```

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

```sql
SELECT
    PIZZA_TYPES.PIZZA_TYPE_ID,
    PIZZA_TYPES.NAME,
    SUM(QUANTITY * PRICE) AS REVENUE
FROM
    PIZZA_TYPES
        JOIN
    PIZZAS ON PIZZAS.PIZZA_TYPE_ID = PIZZA_TYPES.PIZZA_TYPE_ID
        JOIN
    ORDER_DETAILS ON PIZZAS.PIZZA_ID = ORDER_DETAILS.PIZZA_ID
GROUP BY PIZZA_TYPE_ID , PIZZA_TYPES.NAME
ORDER BY REVENUE DESC
LIMIT 3;
```

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

```sql
SELECT
    PIZZA_TYPES.PIZZA_TYPE_ID,
    ROUND(SUM(QUANTITY * PRICE) / (SELECT
                ROUND(SUM(PRICE * QUANTITY), 2) AS TOTAL_REVENUE
            FROM
                PIZZAS
                    JOIN
                ORDER_DETAILS ON PIZZAS.PIZZA_ID = ORDER_DETAILS.PIZZA_ID) * 100,
        0) AS REVENUE_PERCENTAGE
FROM
    PIZZA_TYPES
        JOIN
    PIZZAS 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.PIZZA_TYPE_ID;
```

# ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

```
SELECT DATE,  ROUND(SUM(REVENUE) OVER(ORDER BY DATE))AS CUM_REVENUE FROM
(SELECT DATE(DATE) AS DATE, SUM(QUANTITY * PRICE) AS REVENUE
FROM ORDERS
JOIN ORDER_DETAILS ON
ORDER_DETAILS.ORDER_ID = ORDERS.ORDER_ID
JOIN PIZZAS ON
ORDER_DETAILS.PIZZA_ID = PIZZAS.PIZZA_ID
GROUP BY DATE) AS SALES;
```

# DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY.

```sql
SELECT CATEGORY, NAME, REVENUE, RN FROM
(SELECT CATEGORY, NAME, REVENUE,
RANK() OVER(PARTITION BY CATEGORY ORDER BY REVENUE DESC) AS RN FROM
(SELECT PIZZA_TYPES.PIZZA_TYPE_ID, PIZZA_TYPES.CATEGORY, PIZZA_TYPES.NAME, SUM(QUANTITY * PRICE) AS REVENUE
FROM ORDERS
JOIN ORDER_DETAILS ON
ORDER_DETAILS.ORDER_ID = ORDERS.ORDER_ID
JOIN PIZZAS ON
ORDER_DETAILS.PIZZA_ID = PIZZAS.PIZZA_ID
JOIN PIZZA_TYPES ON
PIZZAS.PIZZA_TYPE_ID = PIZZA_TYPES.PIZZA_TYPE_ID
GROUP BY PIZZA_TYPES.PIZZA_TYPE_ID, PIZZA_TYPES.CATEGORY, PIZZA_TYPES.NAME)AS A)AS B
WHERE RN <=3 ;
```

## CONCLUSION AND REFLECTION

THIS PROJECT PROVIDED VALUABLE INSIGHTS INTO PIZZA SALES, INCLUDING CUSTOMER PREFERENCES, PEAK SALES PERIODS, AND THE MOST POPULAR PIZZA TYPES. CHALLENGES SUCH AS HANDLING LARGE DATASETS AND OPTIMIZING COMPLEX QUERIES WERE EFFECTIVELY MANAGED, ENHANCING MY SQL SKILLS.

## FUTURE ENHANCEMENTS:

EXPAND THE DATASET WITH CUSTOMER FEEDBACK TO ANALYZE SATISFACTION.
INTEGRATE EXTERNAL DATA SOURCES FOR A MORE COMPREHENSIVE MARKET ANALYSIS.

ACKNOWLEDGMENTS: SPECIAL THANKS TO WS CUBETECH FOR PROVIDING THE DATASET AND RESOURCES.

CONTACT INFORMATION: FOR MORE DETAILS OR COLLABORATION OPPORTUNITIES, FEEL FREE TO CONNECT WITH ME:

LINKEDIN
GITHUB
EMAIL