PIZZA HUT

# SQL PROJECTS ON PIZZA SALES

YUMMY

# HELLO !
# WELCOME TO
# SQL PROJECT

Hello My Name Is Ankar Goswami, a data enthusiast with a passion for turning raw data into meaningful insights. Recently, I worked on this SQL-based project analyzing Pizza Hut sales data to uncover trends, customer preferences, and key performance factors.
Using SQL queries, I explored:

- Sales performance by pizza type, size, and time period
- Customer ordering patterns and peak demand hours
- Revenue drivers and best-selling products

This project helped me sharpen my data analysis and SQL skills while providing actionable business insights. Let’s connect if you’d like to discuss data, SQL, or pizza! 🍕📊

# THE TOTAL NUMBER OF ORDERS PLACED

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

Result Grid

| | TOTAL_ORDERS |
|---|---|
| ▶ | 21350 |

# TOTAL REVENUE GENERATED FROM PIZZA SALES

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

Result Grid

| TOTAL_SALES |
| --- |
| 817860.05 |

# THE HIGHEST PRICE PIZZA

```sql
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;
```

Result Grid | Filter Rows:

| NAME | PRICE |
|---|---|
| The Greek Pizza | 35.95 |

# THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY.

```sql
SELECT
    ROUND(AVG(QUANTITY), 0) AS AVG_ORDER_PERDAY
FROM
    (SELECT
        O.ORDER_DATE AS ORDER_PER_DAY, SUM(OD.QUANTITY) AS QUANTITY
    FROM
        ORDERS O
    JOIN ORDERS_DETAILS OD ON O.ORDER_ID = OD.ORDER_ID
    GROUP BY ORDER_PER_DAY) AS ORDER_QUNATITY;
```

Result Grid | Filter Rows:

| AVG_ORDER_PERDAY |
| --- |
| 138 |

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

```sql
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
    ORDERS_DETAILS OD ON OD.PIZZA_ID = P.PIZZA_ID
GROUP BY PT.NAME
ORDER BY QUANTITY DESC
LIMIT 5;
```

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 |

# THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED.

```sql
SELECT
    PT.CATEGORY, SUM(OD.QUANTITY) AS TOTAL_QUANTITY
FROM
    PIZZA_TYPES PT
        JOIN
    PIZZAS P ON PT.PIZZA_TYPE_ID = P.PIZZA_TYPE_ID
        JOIN
    ORDERS_DETAILS OD ON OD.PIZZA_ID = P.PIZZA_ID
GROUP BY PT.CATEGORY
ORDER BY TOTAL_QUANTITY DESC;
```

Result Grid  Filter Rows:

| CATEGORY | TOTAL_QUANTITY |
|---|---|
| Classic | 14888 |
| Supreme | 11987 |
| Veggie | 11649 |
| Chicken | 11050 |

# TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.

```sql
SELECT PT.NAME AS NAME, SUM(P.PRICE*OD.QUANTITY) AS REVENUE
FROM PIZZA_TYPES PT
JOIN PIZZAS P
ON PT.PIZZA_TYPE_ID = P.PIZZA_TYPE_ID
JOIN ORDERS_DETAILS OD
ON OD.PIZZA_ID = P.PIZZA_ID
GROUP BY NAME
ORDER BY REVENUE DESC
LIMIT 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 CATEGORY-WISE DISTRIBUTION OF PIZZAS.

```sql
SELECT
    PT.CATEGORY PIZZA_CATEGORY, COUNT(PT.NAME) AS NO_OF_PIZZAS
FROM
    PIZZA_TYPES PT
GROUP BY PIZZA_CATEGORY;
```

Result Grid | Filter Rows:

| PIZZA_CATEGORY | NO_OF_PIZZAS |
|---|---|
| Chicken | 6 |
| Classic | 8 |
| Supreme | 9 |
| Veggie | 9 |

# THE MOST COMMON PIZZA SIZE ORDERED

```sql
SELECT
    P.SIZE, COUNT(OD.ORDER_DETAILS_ID) AS NO_OF_ORDER
FROM
    PIZZAS P
        JOIN
    ORDERS_DETAILS OD ON P.PIZZA_ID = OD.PIZZA_ID
GROUP BY P.SIZE
ORDER BY NO_OF_ORDER DESC;
```

Result Grid | Filter Rows:

| SIZE | NO_OF_ORDER |
|---|---|
| L | 18526 |
| M | 15385 |
| S | 14137 |
| XL | 544 |
| XXL | 28 |

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

```sql
SELECT NAME, REVENUE FROM
(SELECT CATEGORY, NAME, REVENUE,
RANK() OVER( PARTITION BY CATEGORY ORDER BY REVENUE DESC) AS RANKING
FROM
(SELECT PT.CATEGORY, PT.NAME, SUM(OD.QUANTITY * P.PRICE) AS REVENUE
FROM ORDERS_DETAILS OD
JOIN PIZZAS P
ON OD.PIZZA_ID = P.PIZZA_ID
JOIN PIZZA_TYPES PT
ON PT.PIZZA_TYPE_ID = P.PIZZA_TYPE_ID
GROUP BY PT.CATEGORY, PT.NAME) AS TABLE1) AS TABLE2
WHERE RANKING<= 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 |

# THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

```sql
SELECT
    HOUR(O.ORDER_TIME) AS HOURS,
    COUNT(O.ORDER_ID) AS ORDERS_COUNT
FROM
    ORDERS O
GROUP BY HOURS;
```

Result Grid | Filter Rows:

| HOURS | ORDERS_COUNT |
|---|---|
| 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 |
| 10 | 8 |
| 9 | 1 |

# PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE.

```sql
SELECT
    PT.CATEGORY,
    ROUND((SUM(OD.QUANTITY * P.PRICE) / (SELECT
            ROUND(SUM(OD.QUANTITY * P.PRICE), 2) AS TOTAL_SALES
        FROM
            ORDERS_DETAILS OD
                JOIN
            PIZZAS P ON OD.PIZZA_ID = P.PIZZA_ID)),2) * 100 AS REVENUE
FROM
    PIZZA_TYPES PT
        JOIN
    PIZZAS P ON PT.PIZZA_TYPE_ID = P.PIZZA_TYPE_ID
        JOIN
    ORDERS_DETAILS OD ON OD.PIZZA_ID = P.PIZZA_ID
GROUP BY PT.CATEGORY
ORDER BY REVENUE DESC;
```

Result Grid | Filter Rows

| CATEGORY | REVENUE |
|---|---|
| Classic | 27 |
| Supreme | 25 |
| Veggie | 24 |
| Chicken | 24 |

# THE CUMULATIVE REVENUE GENERATED OVER TIME.

```sql
SELECT ORDER_DATE,
ROUND(SUM(REVENUE) OVER (ORDER BY ORDER_DATE),2) AS CUM_REVENUE
FROM
(SELECT O.ORDER_DATE,
SUM(OD.QUANTITY * P.PRICE) AS REVENUE
FROM ORDERS_DETAILS OD
JOIN PIZZAS P
ON OD.PIZZA_ID = P.PIZZA_ID
JOIN ORDERS O
ON O.ORDER_ID = OD.ORDER_ID
GROUP BY O.ORDER_DATE) AS SALES;
```

Result Grid | Filter Rows:

| ORDER_DATE | CUM_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 |
| 2015-01-10 | 23990.35 |
| 2015-01-11 | 25862.65 |

PIZZA HUT

# THANK YOU!