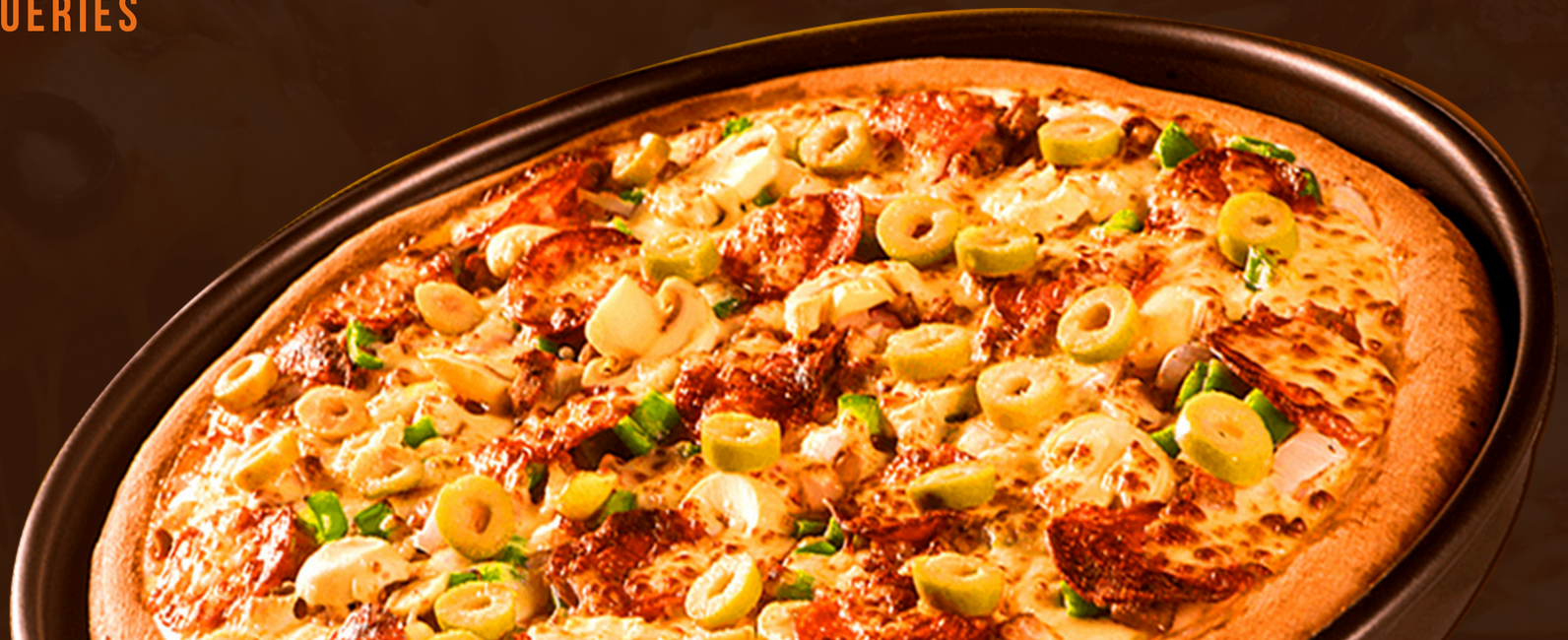
PIZZA HUT SALES SQL ANALYSIS

BUSINESS INSIGHTS USING SQL QUERIES



TOTAL ORDERS PLACED

THIS QUERY COUNTS THE TOTAL NUMBER OF UNIQUE ORDERS IN THE DATASET.

```sql
-- Retrieve the total number of orders placed.
use pizzahut;
SELECT
    COUNT(order_id) AS total_orders
FROM
    orders;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

total_orders
66207

TOTAL REVENUE

CALCULATES THE TOTAL REVENUE GENERATED FROM ALL PIZZA SALES.

```sql
-- Calculate the total revenue generated from pizza sales.
SELECT
    ROUND(SUM(order_details.quantity * pizzas.price),
            2) AS total_revenue
FROM
    order_details
        JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

total_revenue
817860.05

HIGHEST-PRICED PIZZA

IDENTIFIES THE MOST EXPENSIVE PIZZA ON THE MENU.

```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: | Export: | Wrap Cell Content: | Fetch rows:

name	price
The Greek Pizza	35.95

MOST COMMON PIZZA SIZE

Finds the most frequently ordered pizza size.

```sql
-- Identify the most common pizza size ordered.
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 Grid | Filter Rows: | Export: | Wrap Cell Content:

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

TOP 5 PIZZA TYPES BY QUANTITY

LISTS THE FIVE MOST ORDERED PIZZA TYPES BASED ON TOTAL QUANTITY SOLD.

```sql
-- 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;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch 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

TOTAL QUANTITY BY CATEGORY

SUMMARIZES THE TOTAL QUANTITY SOLD FOR EACH PIZZA CATEGORY.

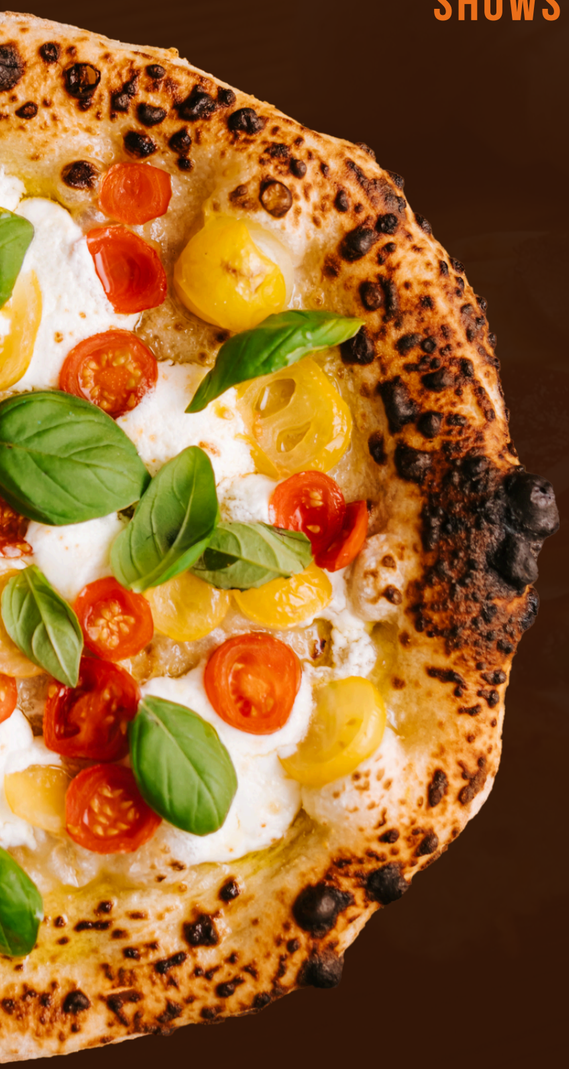
```sql
-- Join the necessary tables to find the total quantity of each pizza category ordered
SELECT
    pizza_types.category,
    SUM(order_details.quantity) AS total_qty
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY total_qty
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

category	total_qty
Chicken	11050
Veggie	11649
Supreme	11987
Classic	14888

CATEGORY-WISE PIZZA DISTRIBUTION

SHOWS HOW PIZZA ORDERS ARE DISTRIBUTED ACROSS DIFFERENT CATEGORIES.



```sql
-- Join relevant tables to find the category-wise distribution of pizzas.
SELECT
    pizza_types.category, COUNT(order_details.order_id)
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;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

category	COUNT(order_details.order_id)
Classic	14579
Veggie	11449
Supreme	11777
Chicken	10815

AVERAGE PIZZAS ORDERED PER DAY

CALCULATES THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY.

```sql
-- Group the orders by date and calculate the average number of pizzas ordered per day.
SELECT
    ROUND(AVG(quantity), 0) as avg_pizza_ordered_per_day
FROM
    (SELECT
        orders.date, SUM(order_details.quantity) AS quantity
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY orders.date) AS order_quantity;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

avg_pizza_ordered_per_day
429

TOP 3 PIZZAS BY REVENUE

DISPLAYS THE THREE PIZZA TYPES THAT GENERATED THE HIGHEST REVENUE.

```sql
-- Determine the top 3 most ordered pizza types based on revenue.
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 Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

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

REVENUE CONTRIBUTION BY CATEGORY

SHOWS WHAT PERCENTAGE OF TOTAL REVENUE EACH CATEGORY CONTRIBUTES.

```sql
-- Calculate the percentage contribution of each pizza type to total revenue.
SELECT
    pizza_types.category,
    ROUND(SUM(order_details.quantity * pizzas.price) / (SELECT
                    ROUND(SUM(order_details.quantity * pizzas.price),
                                2) AS total_revenue
                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 Grid | Filter Rows:

category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

CUMULATIVE REVENUE OVER TIME

ANALYZES HOW REVENUE HAS ACCUMULATED OVER TIME.

```sql
-- Analyze the cumulative revenue generated over time.
SELECT
    orders.date,
    SUM(order_details.quantity * pizzas.price) AS daily_revenue,
    SUM(SUM(order_details.quantity * pizzas.price)) OVER (ORDER BY orders.date)
    AS cumulative_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.date
ORDER BY
    orders.date;
```

Result Grid | Filter Rows: | Export:

date	daily_revenue	cumulative_revenue
2015-01-01	8376.55	8376.55
2015-01-02	8655.199999999999	17031.75
2015-01-03	8052.699999999997	25083.699999999997
2015-01-04	5516.849999999985	30600.549999999996
2015-01-05	6418.349999999985	37018.899999999994
2015-01-06	7637.749999999999	44656.649999999994
2015-01-07	7126.249999999997	51782.899999999994
2015-01-08	8790.05	60572.95
2015-01-09	6634.299999999997	67207.25
2015-01-10	7503.099999999985	74710.35

TOP 3 PIZZAS BY CATEGORY REVENUE

RANKS THE TOP 3 PIZZA TYPES BY REVENUE WITHIN EACH CATEGORY USING WINDOW FUNCTIONS.

```sql
-- Determine the top 3 most ordered pizza types based on revenue for each pizza category.
SELECT
    category,
    name,
    revenue
FROM (
    SELECT
        pt.category,
        pt.name,
        SUM(od.quantity * p.price) AS revenue,
        RANK() OVER (PARTITION BY pt.category ORDER BY SUM(od.quantity * p.price) DESC) AS rank_within_category
    FROM
        pizza_types pt
    JOIN
        pizzas p ON pt.pizza_type_id = p.pizza_type_id
    JOIN
        order_details od ON od.pizza_id = p.pizza_id
    GROUP BY
        pt.category, pt.name
) AS ranked
WHERE
    rank_within_category <= 3
ORDER BY
    category, revenue DESC limit 3;
```

Result Grid | Filter Rows:

category	name	revenue
Chicken	The Thai Chicken Pizza	43434.25
Chicken	The Barbecue Chicken Pizza	42768
Chicken	The California Chicken Pizza	41409.5