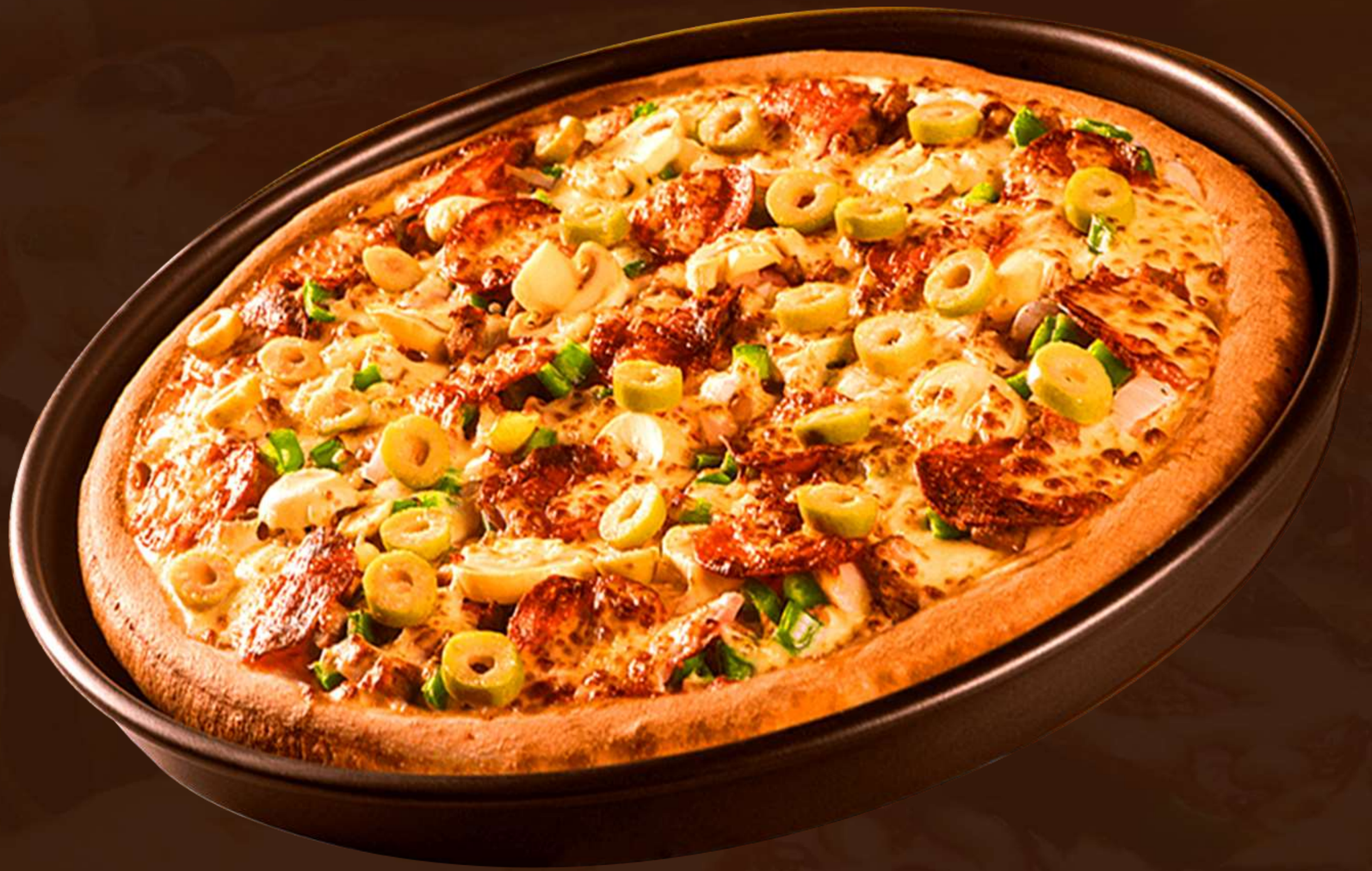
Where Every Slice is a Taste of Perfection

WELCOME ! !



ABSTRACT

This is purely designed to showcase the analysis of pizza sales records along with its distribution and many more details related to orders .
Here queries are been solved from basic to advance and the output has been analyzed to know more about the sales.

Language used for this analysis purpose is SQL

- SQL is primarily used for managing and manipulating relational databases. It allows users to query data.
- SQL also facilitates the creation and modification of database schemas, ensuring data integrity through rules and constraints.

SCHEMAS USED FOR SALES RECORDS

PIZZAS

PIZZA TYPES

ORDERS

ORDER DETAILS

THE TABLES IN THE FOLLOWING SCHEMAS ARE

- PIZZAS

Pizza_id,Pizza_type_id,size,price

- PIZZA TYPES

Pizza_type_id,name,category,ingredient

- ORDERS

Order_id,Order_date,Order_time

- ORDER DETAILS

Order_details_id,Pizza_id,Order_id,quantity

-- CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES
-- IDENTIFY THE HIGHEST PRICE PIZZA

```sql
-- Calculate the total revenue generated from pizza sales

SELECT
    ROUND(SUM(orders_details.quantity * pizzas.price),
            2) AS total_sales
FROM
    orders_details
        JOIN
    pizzas ON pizzas.pizza_id = orders_details.pizza_id
```

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

total_sales
817860.05

```sql
SELECT
    pizzas.size,
    COUNT(orders_details.order_details_id) AS order_count
FROM
    pizzas
        JOIN
    orders_details ON pizzas.pizza_id = orders_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

-- LIST THE TOP 5 MOST ODERED PIZZA TYPES ALONG WITH THEIR QUANTITIES
-- JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANNTITY OF EACH PIZZA CATEGORY ORDERED.

```sql
SELECT
    pizza_types.name, SUM(orders_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    orders_details ON orders_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

```sql
SELECT
    pizza_types.category,
    SUM(orders_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    orders_details ON pizzas.pizza_id = orders_details.pizza_id
GROUP BY pizza_types.category
ORDER BY quantity DESC;
```

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

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

-- GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY
-- DETEMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE

```sql
SELECT
    ROUND(AVG(quantity))
FROM
    (SELECT
        orders.order_date, SUM(orders_details.quantity) AS quantity
    FROM
        orders
    JOIN orders_details ON orders.order_id = orders_details.order_id
    GROUP BY orders.order_date) AS order_quantity;
```

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

ROUND(AVG(quantity))
138

```sql
SELECT
    pizza_types.name,
    SUM(orders_details.quantity * pizzas.price) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    orders_details ON orders_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

-- CALCULATE THE PERCENTAGE OONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE

```sql
SELECT
    pizza_types.category,
    round( (SUM(orders_details.quantity * pizzas.price) / (SELECT
            ROUND(SUM(orders_details.quantity * pizzas.price),
                    2) AS total_sales
        FROM
            orders_details
                JOIN
            pizzas ON pizzas.pizza_id = orders_details.pizza_id)) * 100,2) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY revenue DESC;--
```

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

category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

-- ANALYSE THE CUMALATIVE REVENUE GENERATED OVER TIME.

```sql
select
order_date, sum(revenue) over(order by order_date) as cum_revenue
 from
 (select orders.order_date,
   sum(orders_details.quantity * pizzas.price) as revenue
   from orders_details join pizzas
on orders_details.pizza_id=pizzas.pizza_id
join orders
on orders.order_id=orders_details.order_id
group by orders.order_date) as sales;
```

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

order_date	cum_revenue
2015-01-01	2713.8500000000004
2015-01-02	5445.75
2015-01-03	8108.15
2015-01-04	9863.6
2015-01-05	11929.55

-- DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY

-- JOIN RELEVANT TABLE TO FIND THE CATEGORY WISE DISTRBUUTION OF PIZZAS

```sql
SELECT
    HOUR(order_time) AS hour, COUNT(order_id) AS order_count
FROM
    orders
GROUP BY HOUR(order_time);
```

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

hour	order_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

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

Result Grid | Filter Rows: | Export: | Wra

category	COUNT(name)
Chicken	6
Classic	8
Supreme	9
Veggie	9

-- DETERMINE 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 rn
 from
(select pizza_types.category,pizza_types.name,
sum((orders_details.quantity) * pizzas.price) as revenue
 from
 pizza_types join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
join orders_details on orders_details.pizza_id=pizzas.pizza_id
group by  pizza_types.category,pizza_types.name) as a) as b
 where rn>=3;
```

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

name	revenue
The California Chicken Pizza	41409.5
The Southwest Chicken Pizza	34705.75
The Chicken Alfredo Pizza	16900.25
The Chicken Pesto Pizza	16701.75
The Pepperoni Pizza	30161.75
The Greek Pizza	28454.100000000013
The Italian Capocollo Pizza	25094
The Napolitana Pizza	24087
The Big Meat Pizza	22968
The Pepperoni, Mushroom, and Peppers Pizza	18834.5

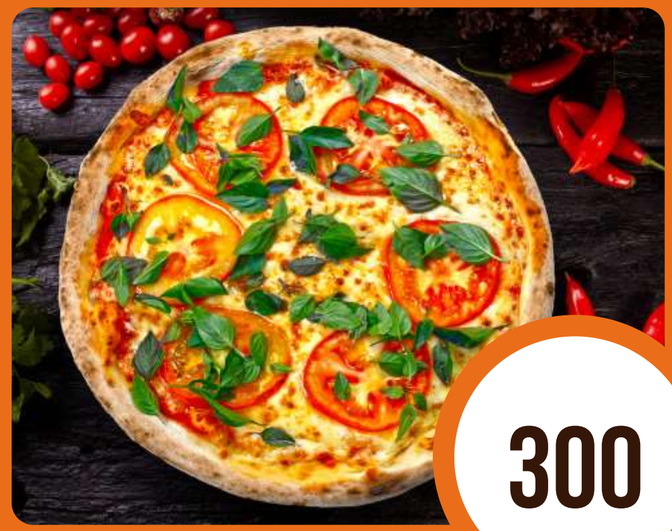
name	revenue
The Pepperoni, Mushroom, and Peppers Pizza	18834.5
The Sicilian Pizza	30940.5
The Pepper Salami Pizza	25529
The Prosciutto and Arugula Pizza	24193.25
The Soppressata Pizza	16425.75
The Calabrese Pizza	15934.25
The Spinach Supreme Pizza	15277.75
The Brie Carre Pizza	11588.4999999999
The Five Cheese Pizza	26066.5
The Vegetables + Vegetables Pizza	24374.75

The Spinach Supreme Pizza	15277.75
The Brie Carre Pizza	11588.4999999999
The Five Cheese Pizza	26066.5
The Vegetables + Vegetables Pizza	24374.75
The Spinach and Feta Pizza	23271.25
The Italian Vegetables Pizza	16019.25
The Spinach Pesto Pizza	15596
The Mediterranean Pizza	15360.5
The Green Garden Pizza	13955.75

OUR SIGNATURE PIZZAS

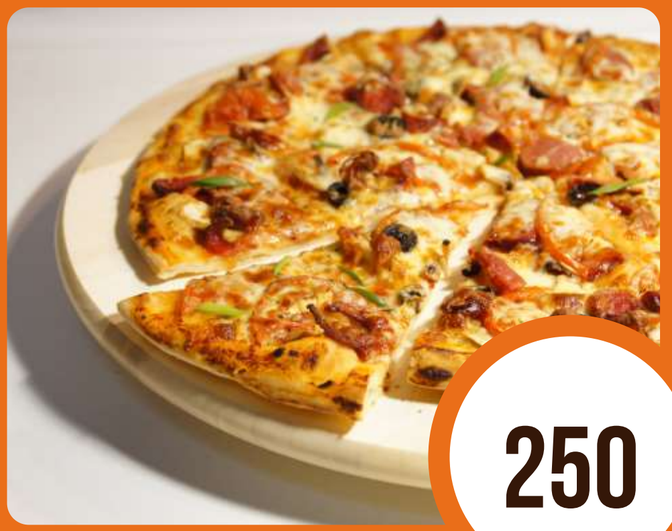
200

Margherita Pizza



300

Veggie Delight



250

Pepperoni Pizza

THANK YOU FOR ATTENTION

See You Next