PIZZA Sales

Delicious pizza for Everyone!

Hello !

In this project, I used SQL queries creatively to address pizza sales-related inquiries, offering unique insights into customer behaviors and operational strategies.

Retrieve the total number of orders placed.

```sql
-- Retrieve the total number of orders placed--
select count(order_id) as total_orders from orders;
```

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

total_orders
21350

Calculate the total revenue generated from 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

Identify the highest-priced pizza.

```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

Identify the most common pizza size ordered.

```
3    -- Identify the most common pizza size ordered--
4    select pizzas.size,count(order_details.order_details_id) as order_count
5    from pizzas
6    join order_details
7    on pizzas.pizza_id = order_details.pizza_id
8    group by pizzas.size
9    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 ordered pizza types along with their quantities.

```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

Join the necessary tables to find the total quantity of each pizza category ordered.

```sql
-- Join the necessary tables to find the total quantity of each pizza category ordered--
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;
```

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

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Determine the distribution of orders by hour of the day.

```sql
-- Determine the distribution of orders by hour of the day--
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

Join relevant tables to find the category-wise distribution of pizzas.

```
6    -- Join relevant tables to find the category-wise distribution of pizzas--
7 •  select category,count(name) from pizza_types
8    group by category;
```

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

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

Group the orders by date and calculate the average number of pizzas ordered per day.

```
10    -- Group the orders by date and calculate the average number of pizzas ordered per day--
11 •  select round(avg(quantity),0) as avg_pizza_order_per_day from
12    (select orders.order_date,sum(order_details.quantity) as quantity
13    from orders join order_details
14    on orders.order_id=order_details.order_id
15    group by orders.order_date) as order_quantity;
16
```

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

avg_pizza_order_per_day
138

Result Grid

Form Editor

Determine the top 3 most ordered pizza types based on 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

Calculate the percentage contribution of each pizza type to total revenue.

```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_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
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

Analyze the cumulative revenue generated over time.

```
26    -- Analyze the cumulative revenue generated over time--
27    select order_date,
28    sum(revenue) over (order by order_date) as cum_revenue
29    from
30    (select orders.order_date,
31    sum(order_details.quantity*pizzas.price) as revenue
32    from order_details join pizzas
33    on order_details.pizza_id=pizzas.pizza_id
34    join orders on
35    orders.order_id=order_details.order_id
36    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 top 3 most ordered pizza types based on revenue for each pizza category.

```sql
-- 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 pt.category, pt.name,
               SUM(od.quantity * p.price) AS revenue
        FROM pizza_types pt
        JOIN pizzas p ON pt.pizza_type_id = p.pizza_type_id
        JOIN order_details od ON p.pizza_id = od.pizza_id
        GROUP BY pt.category, pt.name
    ) AS a
) AS b
```

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

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

THANK You!