Where Every Slice is a Taste of Perfection

WELCOME TO PIZZA RESTO

ORDER NOW

Start Your Slide

HELLO

The Pizza Sales Management Project, developed using SQL queries by Awtar Kisan Yadav, is designed to streamline and analyze pizza business operations. This project enables efficient data handling by storing customer details, tracking orders, managing inventory, and generating sales reports.

Questions that I covered in this project

1. Retrieve the total number of orders placed.
2. Calculate the total revenue generated from pizza sales.
3. Identify the highest-priced pizza.
4. Identify the most common pizza size ordered.
5. List the top 5 most ordered pizza types along with their quantities.
6. Join the necessary tables to find the total quantity of each pizza category ordered.
7. Determine the distribution of orders by hour of the day.
8. Join relevant tables to find the category-wise distribution of pizzas.
9. Group the orders by date and calculate the average number of pizzas ordered per day.
10. Determine the top 3 most ordered pizza types based on revenue.
11. Calculate the percentage contribution of each pizza type to total revenue.
12. Analyze the cumulative revenue generated over time.
13. Determine the top 3 most ordered pizza types based on revenue for each pizza category.

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

Results | Messages

	Total_order_placed
1	21350

```sql
-- Calculate the total revenue generated from pizza sales.
select round(SUM (order_details.quantity*pizzas.price),2) as revenue
from order_details join pizzas
on order_details.pizza_id = pizzas.pizza_id;
```

Results | Messages

	revenue
1	817860.05

```sql
-- Identify the highest-priced pizza.
select top 1 pizza_types.[name] , round(pizzas.price,2) as costliest_pizza
from pizza_types join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
order by pizzas.price desc ;
```

Results | Messages

	name	costliest_pizza
1	The Greek Pizza	35.95

```sql
-- Identify the most common pizza size ordered.A
select top 1 pizzas.size, count(order_details.order_details_id) as common_size
from pizzas join order_details
on pizzas.pizza_id = order_details.pizza_id
group by pizzas.size
order by common_size desc  ;
```

Results | Messages

	size	common_size
1	L	18526

```sql
-- List the top 5 most ordered pizza types along with their quantities.

select top 5 pizza_types.[name]  , sum (order_details.quantity ) as top_selling
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 pizza_types.[name]
order by top_selling desc ;
```

Results | Messages

	name	top_selling
1	The Classic Deluxe Pizza	2453
2	The Barbecue Chicken Pizza	2432
3	The Hawaiian Pizza	2422
4	The Pepperoni Pizza	2418
5	The Thai Chicken Pizza	2371

```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_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 pizza_types.category
order by total_quantity desc
```

Results | Messages

	category	total_quantity
1	Classic	14888
2	Supreme	11987
3	Veggie	11649
4	Chicken	11050

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

select DATEPART( hour,[time] ) as hour_of_day, count (order_id) as total_orders
from orders
group by DATEPART( hour,[time] )
order by DATEPART( hour,[time] );
```

Results | Messages

	hour_of_day	total_orders
1	9	1
2	10	8
3	11	1231
4	12	2520
5	13	2455
6	14	1472
7	15	1468
8	16	1920
9	17	2336
10	18	2399
11	19	2009
12	20	1642
13	21	1198
14	22	663
15	23	28

```sql
-- Join relevant tables to find the category-wise distribution of pizzas.
select category ,count ([name]) as total_pizza
from pizza_types
group by category ;
```

Results | Messages

	category	total_pizza
1	Chicken	6
2	Classic	8
3	Supreme	9
4	Veggie	9

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

select round (avg(total_orders),0) as avg_orders_per_day from
(select  orders.[date] as [date], SUM( order_details.quantity) as total_orders
from orders join order_details
on orders.order_id = order_details.order_id
group by orders.[date]
) as order_quantity
 ;
```

Results | Messages

	avg_orders_per_day
1	138

```sql
-- Determine the top 3 most ordered pizza types based on revenue.

select top 3 pizza_types.[name] as top_pizzas, sum(order_details.quantity*pizzas.price) 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.[name]
order by revenue desc ;
```

Results | Messages

	top_pizzas	revenue
1	The Thai Chicken Pizza	43434.25
2	The Barbecue Chicken Pizza	42768
3	The California Chicken Pizza	41409.5

```sql
-- Calculate the percentage contribution of each ordepizza category  to total revenue.

select  pizza_types.category as top_pizzas, round (sum (order_details.quantity*pizzas.price) / (select SUM (order_details.quantity*pizzas.price)
from order_details join pizzas
on order_details.pizza_id = pizzas.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 ;
```

Results | Messages

	top_pizzas	revenue
1	Classic	26.91
2	Supreme	25.46
3	Chicken	23.96
4	Veggie	23.68

```sql
-- Analyze the cumulative revenue generated over time.
select [date], SUM ( revenue) over (order by [date]) as cummalative_revenue
from
(select orders.[date] , SUM(pizzas.price*order_details.quantity) as revenue
from orders join order_details
on orders.order_id = order_details. order_id
join pizzas
on pizzas.pizza_id = order_details.pizza_id
group by orders.[date]) as sales ;
```

	date	cummalative_revenue
1	2015-01-01	2713.85000228882
2	2015-01-02	5445.7500038147
3	2015-01-03	8108.15000724792
4	2015-01-04	9863.60000801086
5	2015-01-05	11929.5500087738
6	2015-01-06	14358.5000114441
7	2015-01-07	16560.700012207
8	2015-01-08	19399.0500183105
9	2015-01-09	21526.4000225067
10	2015-01-10	23990.350025177
11	2015-01-11	25862.6500263214
12	2015-01-12	27781.7000274658
13	2015-01-13	29831.3000278473
14	2015-01-14	32358.7000293732
15	2015-01-15	34343.5000324249
16	2015-01-16	36937.6500339508

```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 TOP_PIZZA
from
(select pizza_types.category,pizza_types.[name], SUM(pizzas.price * order_details.quantity) as revenue
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 pizza_types.category,pizza_types.[name]) as a) as b
where TOP_PIZZA <= 3;
```

	name	revenue
1	The Thai Chicken Pizza	43434.25
2	The Barbecue Chicken Pizza	42768
3	The California Chicken Pizza	41409.5
4	The Classic Deluxe Pizza	38180.5
5	The Hawaiian Pizza	32273.25
6	The Pepperoni Pizza	30161.75
7	The Spicy Italian Pizza	34831.25
8	The Italian Supreme Pizza	33476.75
9	The Sicilian Pizza	30940.5
10	The Four Cheese Pizza	32265.7010040283
11	The Mexicana Pizza	26780.75
12	The Five Cheese Pizza	26066.5

THANK YOU