PIZZAHUB-SQL

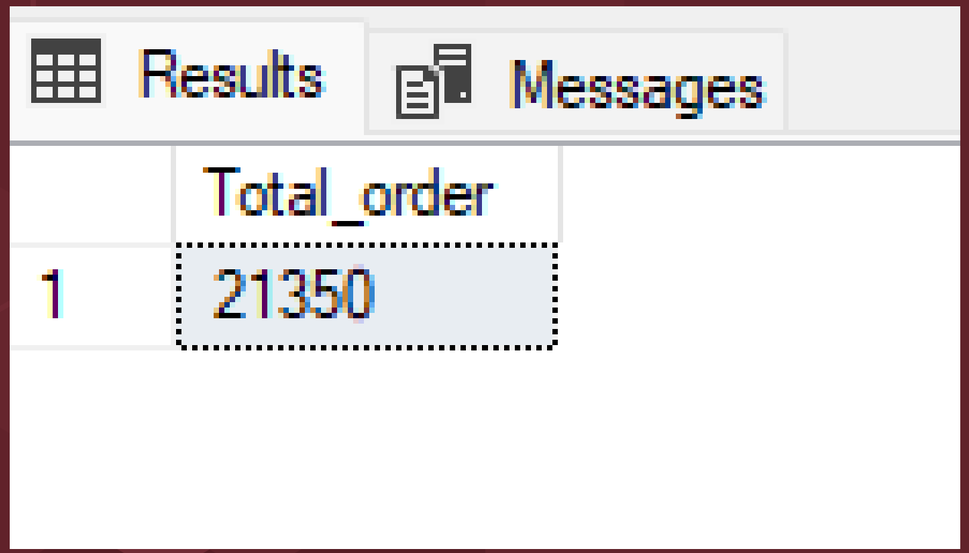
03 July, 2024

INTRODUCTION

Hello i am Monayem islam .
This project involves the design and implementation of a relational database using SQL (Structured Query Language).

- **Q1. RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.**

```
select count(order_id) as Total_order from orders
```



• Q2.CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

```
select round(sum(order_details.quantity * pizzas.price),3) as Total_price
from order_details
inner join pizzas
on order_details.pizza_id=pizzas.pizza_id
```

	Total_price
1	817860.051

• Q3.IDENTIFY THE HIGHEST-PRICED PIZZA.

```
select top 1 pizza_types.name,pizzas.price from pizza_types
inner join pizzas
on pizza_types.pizza_type_id=pizzas.pizza_type_id
order by pizzas.price desc;
```

	name	price
1	The Greek Pizza	35.950000762939453

• Q4.IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

```
select pizzas.size,count(order_details.order_id) from pizzas
inner join order_details
on pizzas.pizza_id=order_details.pizza_id
group by pizzas.size order by count(order_details.order_id) desc
```

	size	(No column name)
1	L	18526
2	M	15385
3	S	14137
4	XL	544
5	XXL	28

• Q5.LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES.

```
select top(5) pizza_types.name,sum(order_details.quantity) as quntity_count
from pizza_types
inner join pizzas
on pizza_types.pizza_type_id=pizzas.pizza_type_id
inner join order_details
on pizzas.pizza_id=order_details.pizza_id
group by pizza_types.name order by quntity_count desc;
```

	name	quntity_count
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

• Q6. JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED.

```
select pizza_types.category,sum(order_details.quantity) from pizza_types
    inner join pizzas
    on pizza_types.pizza_type_id=pizzas.pizza_type_id
    inner join order_details
    on pizzas.pizza_id=order_details.pizza_id
    group by pizza_types.category order by sum(order_details.quantity) desc;
```

	category	(No column name)
1	Classic	14888
2	Supreme	11987
3	Veggie	11649
4	Chicken	11050

• Q7.DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

```
select DATEPART(HOUR,order_time) as DTime,count(order_id) as orderCount
from orders
group by DATEPART(HOUR,order_time) order by orderCount desc;
```

	DTime	orderCount
1	12	2520
2	13	2455
3	18	2399
4	17	2336
5	19	2009
6	16	1920
7	20	1642

- **Q8. JOIN RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS.**

```
select category,count(name) from pizza_types
group by category order by count(name) desc;
```

	category	(No column name)
1	Supreme	9
2	Veggie	9
3	Classic	8
4	Chicken	6

- **Q9.GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY.**

```
select avg(quantity_day) as Avg_perday from
(select orders.order_date,sum(order_details.quantity) as quantity_day
from orders
inner join order_details
on orders.order_id=order_details.order_id
group by orders.order_date) as quntities;
```

Results | Messages

	Avg_perday
1	138

- **Q10.DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.**

```
select top(3) pizza_types.name,sum(order_details.quantity*pizzas.price)
from pizza_types
inner join pizzas
on pizza_types.pizza_type_id=pizzas.pizza_type_id
inner join order_details
on pizzas.pizza_id=order_details.pizza_id
group by pizza_types.name
order by sum(order_details.quantity*pizzas.price) desc;
```

	name	(No column name)
1	The Thai Chicken Pizza	43434.25
2	The Barbecue Chicken Pizza	42768
3	The California Chicken Pizza	41409.5

• Q12. ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

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

	order_date	revenue	cumulative
1	2015-01-01	2713.85000228882	2713.85000228882
2	2015-01-02	2731.90000152588	5445.7500038147
3	2015-01-03	2662.40000343323	8108.15000724792
4	2015-01-04	1755.45000076294	9863.60000801086
5	2015-01-05	2065.95000076294	11929.5500087738
6	2015-01-06	2428.95000267029	14358.5000114441
7	2015-01-07	2202.20000076294	16560.700012207

• Q13.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(order_details.quantity * pizzas.price)
from pizza_types
    inner join pizzas
        on pizza_types.pizza_type_id=pizzas.pizza_type_id
    inner join order_details
        on pizzas.pizza_id=order_details.pizza_id
    group by pizza_types.category,pizza_types.name) as a) as b
    where rn<=3;
```

	name	revenue
1	The Thai Chicken Pizza	43434.25
2	The Barbecue Chicken Pizza	42768
3	The Californa 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....
11	The Mexicana Pizza	26780.75
12	The Five Cheese Pizza	26066.5

THANK YOU

03 July, 2024