Pizza Sales Queries

A. Create New Database

1. Create the Schema

```
Create Database "Normalized Pizza DB"
USE [Normalized Pizza DB]
GO

SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[pizza_sales](
        [pizza_id] [int] NOT NULL,
        [order_id] [int] NOT NULL,
        [pizza_name_id] [varchar](50) NOT NULL,
        [quantity] [tinyint] NOT NULL,
        [order_date] [date] NOT NULL,
        [order_time] [time](7) NOT NULL,
        [unit_price] [float] NOT NULL,
        [total_price] [float] NOT NULL,
        [pizza_size] [varchar](50) NOT NULL,
        [pizza_category] [varchar](50) NOT NULL,
        [pizza_ingredients] [varchar](200) NOT NULL,
        [pizza_name] [varchar](50) NOT NULLf
) ON [PRIMARY]
GO
```

1. Connect with the source file

```
EXEC sp_configure "show advanced options", 1;
RECONFIGURE;
EXEC sp_configure "Ad Hoc Distributed Queries", 1;
RECONFIGURE;

BULK INSERT pizza_sales
FROM 'D:\Data Science Project\Pizza Sales\pizza_sales.csv'
WITH (
   FIRSTROW = 2,
   FIELDTERMINATOR = ',',
   ROWTERMINATOR = '\n',
   TABLOCK);
```

B. Normalization

```
select * from New_pizza_sales;

select distinct(pizza_name_id), pizza_size,pizza_name,
pizza_category,pizza_ingredients,unit_price
into Pizza_Menu
from New_pizza_sales

Alter Table New_pizza_sales
drop column pizza_id, pizza_size,pizza_name, pizza_category
,pizza_ingredients,unit_price,total_price
```

C. Tables Relations

```
Alter Table New_pizza_sales
Add constraint PK_New_pizza_sales primary Key (order_id,Pizza_name_id)

Alter Table Pizza_Menu
Add constraint PK_Pizza_Menu primary Key (pizza_name_id)

Alter table New_pizza_sales
add constraint FK_New_pizza_sales_Pizza_Menu
Foreign key (pizza_name_id) references Pizza_Menu(pizza_name_id)
```

D. Key Performance indicators (KPIs)

1.Total Revenue

```
select cast(sum(s.quantity * m.unit_Price) as decimal(10,0)) as Total_Revenu
from New_pizza_sales as s , pizza_menu as m
where s.pizza_name_id = m.pizza_name_id
```

	Total_Revenu
1	817860

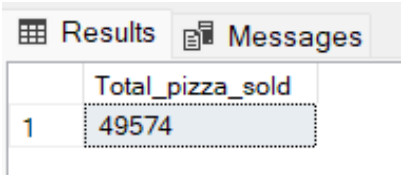
2.Average order value

```
select
cast(
cast(sum(s.quantity * m.unit_Price) as decimal(10,2)) /
cast( count( distinct(s.order_id) ) as decimal(10,2) )
as decimal(10,2)) as Avg_Order_Value
from New_pizza_sales as s , pizza_menu as m
where s.pizza_name_id = m.pizza_name_id
```

	Avg_Order_Value
1	38.31

3.Total pizza sold

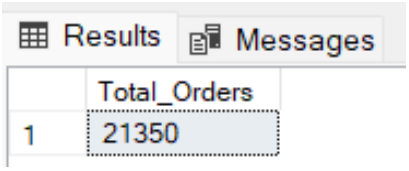
```
select sum(quantity) as Total_Pizza_Sold from New_pizza_sales
```



	Total_pizza_sold
1	49574

4.Total orders

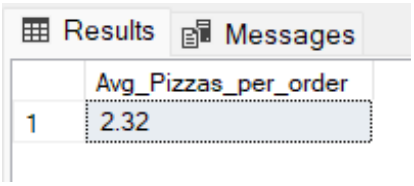
```
select count(distinct(order_id)) Total_Order from New_pizza_sales
```



	Total_Orders
1	21350

5.Average pizza per order

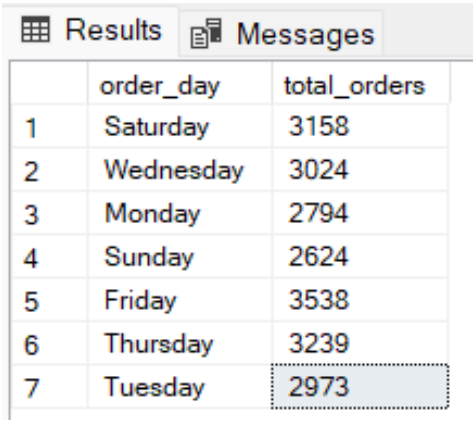
```
select
cast(
cast(sum(quantity) as decimal(10,2))
/ count(distinct(order_id))
as decimal(10,2)) as Avg_Pizza_Per_Order
from New_pizza_sales
```



	Avg_Pizzas_per_order
1	2.32

E. Daily Trend for Total Orders

```
SELECT DATENAME(DW, order_date) AS order_day
, count(distinct(order_id)) as Total_Orders
from New_pizza_sales
group by DATENAME(DW, order_date)
```



	order_day	total_orders
1	Saturday	3158
2	Wednesday	3024
3	Monday	2794
4	Sunday	2624
5	Friday	3538
6	Thursday	3239
7	Tuesday	2973

F. Monthly Trend for Total Orders

```
SELECT DATENAME(MONTH, order_date) AS order_Month
, count(distinct(order_id)) as Total_Orders
from New_pizza_sales
group by DATENAME(MONTH, order_date)
```

	order_Month	Total_Orders
1	February	1685
2	June	1773
3	August	1841
4	April	1799
5	May	1853
6	December	1680
7	January	1845
8	September	1661
9	October	1646
10	July	1935
11	November	1792
12	March	1840

G. Percentage of sales for pizza category

```
select pizza_category
, cast(sum(s.quantity * m.unit_Price) as decimal(10,0)) as Total_Revenu ,
cast(
cast(sum(s.quantity * m.unit_Price) as decimal(10,2) ) * 100 /
(select sum(s.quantity * m.unit_Price)
from New_pizza_sales as s , pizza_menu as m
where s.pizza_name_id = m.pizza_name_id)
 as decimal(10,2) ) as PCT
from New_pizza_sales as s , pizza_menu as m
where s.pizza_name_id = m.pizza_name_id
group by pizza_category
order by Total_Revenu desc
```

	pizza_category	Total_Revenu	PCT
1	Classic	220053	26.91
2	Supreme	208197	25.46
3	Chicken	195920	23.96
4	Veggie	193690	23.68

H. Percentage of sales for pizza size

```
select pizza_size
, cast(sum(s.quantity * m.unit_Price) as decimal(10,0)) as Total_Revenu ,
cast(
cast(sum(s.quantity * m.unit_Price) as decimal(10,2) ) * 100 /
(select sum(s.quantity * m.unit_Price)
from New_pizza_sales as s , pizza_menu as m
where s.pizza_name_id = m.pizza_name_id)
 as decimal(10,2) ) as PCT
from New_pizza_sales as s , pizza_menu as m
where s.pizza_name_id = m.pizza_name_id
group by pizza_size
order by Total_Revenu desc
```

Results Messages

	pizza_size	total_revenue	PCT
1	L	375318.70	45.89
2	M	249382.25	30.49
3	S	178076.50	21.77
4	XL	14076.00	1.72
5	XXL	1006.60	0.12

I. Total pizza sold by pizza category

```
select m.pizza_category , sum(s.quantity) as Total_Quantity_Sold
from New_pizza_sales as s , pizza_menu as m
where s.pizza_name_id = m.pizza_name_id
group by m.pizza_category
order by Total_Quantity_Sold desc
```

Results Messages

	pizza_category	Total_Quantity_Sold
1	Classic	14888
2	Supreme	11987
3	Veggie	11649
4	Chicken	11050

J. Top 5 pizza by Revenue

```
select top 5 pizza_name
, cast(sum(s.quantity * m.unit_Price) as decimal(10,0)) as Total_Revenu
from New_pizza_sales as s , pizza_menu as m
where s.pizza_name_id = m.pizza_name_id
group by pizza_name
order by Total_Revenu desc
```

	pizza_name	Total_Revenu
1	The Thai Chicken Pizza	43434
2	The Barbecue Chicken Pizza	42768
3	The California Chicken Pizza	41410
4	The Classic Deluxe Pizza	38181
5	The Spicy Italian Pizza	34831

K. Bottom 5 pizza by Revenue

```
select top 5 pizza_name
, cast(sum(s.quantity * m.unit_Price) as decimal(10,0)) as Total_Revenu
from New_pizza_sales as s , pizza_menu as m
where s.pizza_name_id = m.pizza_name_id
group by pizza_name
order by Total_Revenu asc
```

	pizza_name	Total_Revenu
1	The Brie Carre Pizza	11588
2	The Green Garden Pizza	13956
3	The Spinach Supreme Pizza	15278
4	The Mediterranean Pizza	15361
5	The Spinach Pesto Pizza	15596

L. Top 5 pizza by Quantity

```
select top 5 pizza_name ,sum(quantity) Quantity_Sold
from New_pizza_sales as s , pizza_menu as m
where s.pizza_name_id = m.pizza_name_id
group by pizza_name
order by Quantity_Sold desc
```

	pizza_name	Quantity_Sold
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

M. Bottom 5 pizza by Quantity

```
select top 5 pizza_name ,sum(quantity) Quantity_Sold
from New_pizza_sales as s , pizza_menu as m
where s.pizza_name_id = m.pizza_name_id
group by pizza_name
order by Quantity_Sold asc
```

	pizza_name	Quantity_Sold
1	The Brie Carre Pizza	490
2	The Mediterranean Pizza	934
3	The Calabrese Pizza	937
4	The Spinach Supreme Pizza	950
5	The Soppressata Pizza	961

N. Top 5 pizza by Total orders

```
select top 5 pizza_name ,count(distinct(order_id)) Total_Orders
from New_pizza_sales as s , pizza_menu as m
where s.pizza_name_id = m.pizza_name_id
group by pizza_name
order by Total_Orders desc
```

Results Messages

	pizza_name	Total_Orders
1	The Classic Deluxe Pizza	2329
2	The Hawaiian Pizza	2280
3	The Pepperoni Pizza	2278
4	The Barbecue Chicken Pizza	2273
5	The Thai Chicken Pizza	2225

O. Bottom 5 pizza by Total orders

```
select top 5 pizza_name ,count(distinct(order_id)) Total_Orders
from New_pizza_sales as s , pizza_menu as m
where s.pizza_name_id = m.pizza_name_id
group by pizza_name
order by Total_Orders asc
```

	pizza_name	Total_Orders
1	The Brie Carre Pizza	480
2	The Mediterranean Pizza	912
3	The Spinach Supreme Pizza	918
4	The Calabrese Pizza	918
5	The Chicken Pesto Pizza	938