PIZZA SALES SQL DATA VALIDATION

MSSQL

A. KPI's

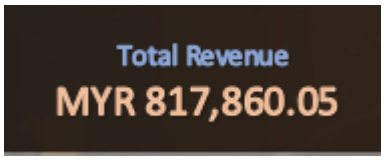
1. Total Revenue: PASSED

```
SELECT SUM(total_price) AS "Total_Revenue" FROM [pizza_sales excel file]
```

Results | Messages

	Total_Revenue
1	817860.05083847

**SQL queries



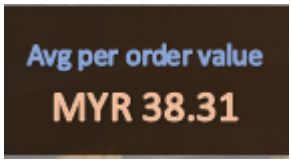
**Dashboard figure

2. Average Order Value PASSED

```
SELECT SUM(total_price)/COUNT(DISTINCT order_id) AS "Avg_order_Value" FROM
[pizza_sales excel file]
```

Results | Messages

	Avg_order_Value
1	38.3072623343546



3. Total Pizzas Sold PASSED

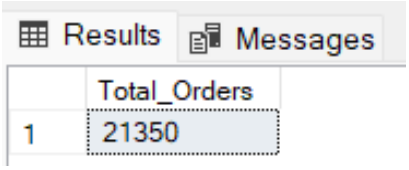
```
SELECT SUM(quantity) AS Total_pizza_sold FROM [pizza_sales excel file]
```

Results | Messages

	Total_pizza_sold
1	49574

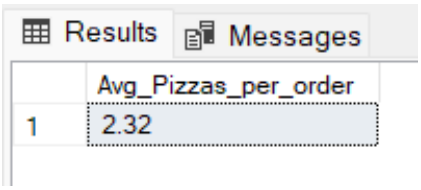
4. Total Orders PASSED

```
SELECT COUNT(DISTINCT order_id) AS Total_order FROM [pizza_sales excel file]
--almost for each and every order more than two pizzas are sold
```



5. Average Pizzas Per Order PASSED

```
SELECT CAST(CAST(SUM(quantity) AS decimal(10,2))/CAST(COUNT(DISTINCT order_id)AS
decimal(10,2)) AS decimal(10,2))
AS Avg_pizzas_per_order
FROM [pizza_sales excel file]
--The use of CAST functions in this query ensures that the calculations
--and the final result are expressed as decimal numbers with two decimal places
```



B. Daily Trend for Total Orders PASSED

```
SELECT DATENAME(DW,order_date) AS order_day, COUNT(DISTINCT order_id) AS total_orders
FROM [pizza_sales excel file]
GROUP BY DATENAME(DW,order_date)
--calculates the total number of distinct orders for each day of the week (e.g.,
Monday, Tuesday) based on the "order_date" column
--The result is grouped by the day of the week using DATENAME(DW, order_date) to
extract the day names, and then counts the number of distinct order IDs for each day.
```

Output:

Results | Messages

	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

C. Hourly Trend for Orders PASSED

```
SELECT DATEPART(HOUR,order_time) AS order_hours, COUNT(DISTINCT order_id) AS
total_orders
FROM [pizza_sales excel file]
GROUP BY DATEPART(HOUR,order_time)
ORDER BY DATEPART(HOUR,order_time)
```

Output

Results | Messages

	order_hours	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

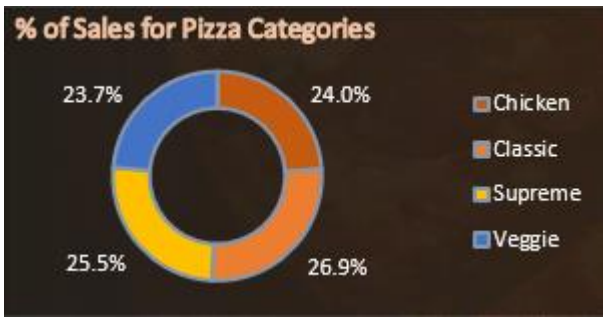
D. % of Sales by Pizza Category PASSED

```
SELECT pizza_category,
       SUM(total_price) AS total_revenue,
       CAST(SUM(total_price)*100/
              (SELECT SUM(total_price) FROM [pizza_sales excel file] ) AS decimal
(10,2))  AS PCT
FROM [pizza_sales excel file]
GROUP BY pizza_category
```

Output

Results Messages

	pizza_category	total_revenue	PCT
1	Classic	220053.10	26.91
2	Chicken	195919.50	23.96
3	Veggie	193690.45	23.68
4	Supreme	208197.00	25.46



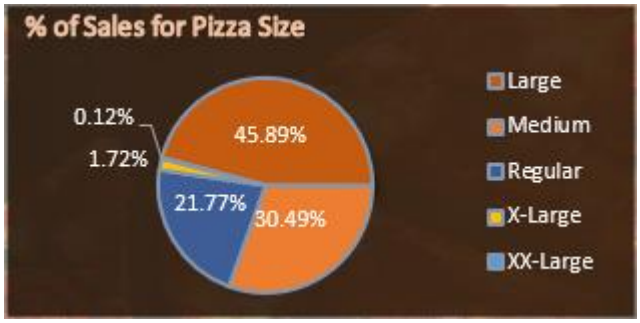
E. % of Sales by Pizza Size PASSED

```
SELECT pizza_size,
       CAST(SUM(total_price) AS decimal(10,2)) AS total_revenue,
       CAST(SUM(total_price)*100/
              (SELECT SUM(total_price) FROM [pizza_sales excel file])  AS decimal
(10,2))  AS PCT
FROM [pizza_sales excel file]
GROUP BY pizza_size
ORDER BY pizza_size
```

Output

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



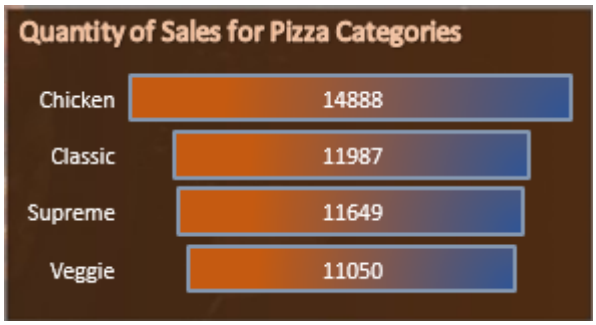
F. Total Pizzas Sold by Pizza Category PASSED

```
SELECT pizza_category,
       SUM(quantity) AS Total_Quantity_Sold
FROM [pizza_sales excel file]
GROUP BY pizza_category
ORDER BY SUM(quantity) DESC
```

Output

Results Messages

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

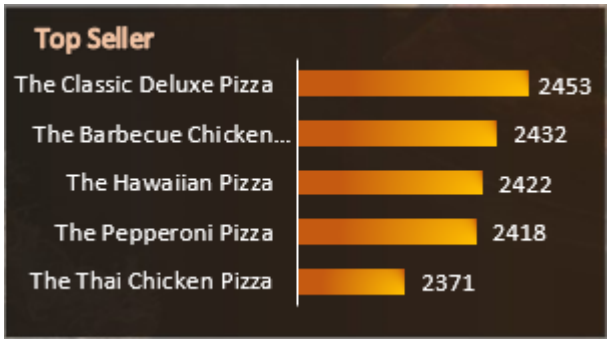


G. Top 5 Best Sellers by Total Pizzas Sold PASSED

```
SELECT TOP 5 pizza_name,
       SUM(quantity) AS Total_Quantity_Sold
FROM [pizza_sales excel file]
GROUP BY pizza_name
ORDER BY SUM(quantity) DESC
```

Output

	pizza_name	Total_Pizza_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



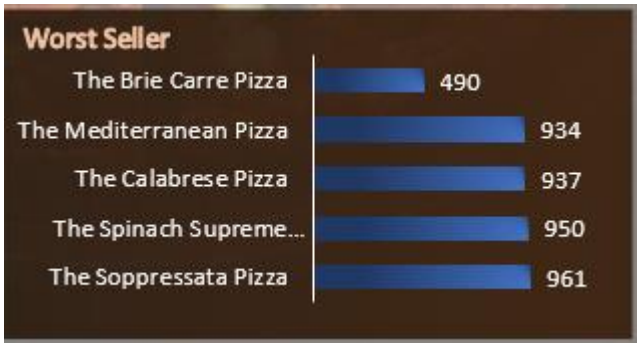
H. Bottom 5 Best Sellers by Total Pizzas Sold PASSED

```
SELECT TOP 5 pizza_name,
       SUM(quantity) AS Total_Quantity_Sold
FROM [pizza_sales excel file]
GROUP BY pizza_name
ORDER BY SUM(quantity) ASC
```

Output

Results Messages

	pizza_name	Total_Pizza_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


Worst Seller
The Brie Carre Pizza
490
The Mediterranean Pizza
934
The Calabrese Pizza
937
The Spinach Supreme...
950
The Soppressata Pizza
961

NOTE

If you want to apply the Month, Quarter, Week filters to the above queries you can use WHERE clause. Follow some of below examples

```
SELECT DATENAME(DW, order_date) AS order_day, COUNT(DISTINCT order_id) AS
total_orders
FROM pizza_sales
WHERE MONTH(order_date) = 1
GROUP BY DATENAME(DW, order_date)
```

**Here MONTH(order_date) = 1 indicates that the output is for the month of January. MONTH(order_date) = 4 indicates output for Month of April.*

```
SELECT DATENAME(DW, order_date) AS order_day, COUNT(DISTINCT order_id) AS
total_orders
FROM pizza_sales
WHERE DATEPART(QUARTER, order_date) = 1
GROUP BY DATENAME(DW, order_date)
```

**Here DATEPART(QUARTER, order_date) = 1 indicates that the output is for the Quarter 1. MONTH(order_date) = 3 indicates output for Quarter 3.*