

A) KPI's

select * from pizza_sales

The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor contains the command `SELECT * FROM pizza_sales`. The results pane displays a table with 12 columns: `pizza_id`, `order_id`, `pizza_name_id`, `quantity`, `order_date`, `order_time`, `unit_price`, `total_price`, `pizza_size`, `pizza_category`, `pizza_ingredients`, and `pi...`. The status bar at the bottom indicates the query was executed successfully, returning 48,620 rows.

	pizza_id	order_id	pizza_name_id	quantity	order_date	order_time	unit_price	total_price	pizza_size	pizza_category	pizza_ingredients	pi...
1	1	1	hawaiian_m	1	2015-01-01	11:38:36.0000000	13.25	13.25	M	Classic	Sliced Ham, Pineapple, Mozzarella Cheese	TI
2	2	2	classic_dlx_m	1	2015-01-01	11:57:40.0000000	16	16	M	Classic	Pepperoni, Mushrooms, Red Onions, Red Peppers, Ba...	TI
3	3	2	five_cheese_l	1	2015-01-01	11:57:40.0000000	18.5	18.5	L	Veggie	Mozzarella Cheese, Provolone Cheese, Smoked Goud...	TI
4	4	2	ital_supr_l	1	2015-01-01	11:57:40.0000000	20.75	20.75	L	Supreme	Calabrese Salami, Capocollo, Tomatoes, Red Onions, ...	TI
5	5	2	mexicana_m	1	2015-01-01	11:57:40.0000000	16	16	M	Veggie	Tomatoes, Red Peppers, Jalapeno Peppers, Red Onio...	TI
6	6	2	thai_ckn_l	1	2015-01-01	11:57:40.0000000	20.75	20.75	L	Chicken	Chicken, Pineapple, Tomatoes, Red Peppers, Thai Sw...	TI
7	7	3	ital_supr_m	1	2015-01-01	12:12:28.0000000	16.5	16.5	M	Supreme	Calabrese Salami, Capocollo, Tomatoes, Red Onions, ...	TI
8	8	3	prsc_argla_l	1	2015-01-01	12:12:28.0000000	20.75	20.75	L	Supreme	Prosciutto di San Daniele, Arugula, Mozzarella Cheese	TI
9	9	4	ital_supr_m	1	2015-01-01	12:16:31.0000000	16.5	16.5	M	Supreme	Calabrese Salami, Capocollo, Tomatoes, Red Onions, ...	TI
10	10	5	ital_supr_m	1	2015-01-01	12:21:30.0000000	16.5	16.5	M	Supreme	Calabrese Salami, Capocollo, Tomatoes, Red Onions, ...	TI

1. Total Revenue: 817860.05083847

The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor contains the command `SELECT SUM(total_price) As Total_Revenue from pizza_sales`. The results pane displays a single row with the value 817860.05083847 under the column `Total_Revenue`.

	Total_Revenue
1	817860.05083847



2. Average Order Value: 38.3072623343546

The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor contains the command `SELECT SUM(total_price) / COUNT(DISTINCT order_id) as AVG_Order_Value From pizza_sales`. The results pane displays a single row with the value 38.3072623343546 under the column `AVG_Order_Value`.

	AVG_Order_Value
1	38.3072623343546


Since we don't want duplicate results, we used distinct count.


3.Total Pizza Sold: 49574

SQLQuery1.sql - VA...VAIDI\Vaidehi (70)*  

```
SELECT * FROM pizza_sales
```



```
SELECT SUM(quantity) As Total_Pizza_Sold from pizza_sales
```

121 % 

Results  Messages


	Total_Pizza_Sold
1	49574


4. Total Order: 21350

SQLQuery1.sql - VA...VAIDI\Vaidehi (70)*  

```
SELECT * FROM pizza_sales
```



```
SELECT COUNT (DISTINCT order_id) As Total_Order from pizza_sales
```

121 % 

Results  Messages

	Total_Order
1	21350


5. Average Pizzas Per Order: 2.3219672131147


SQLQuery1.sql - VA...VAIDI\Vaidehi (70)*  

```
SELECT * FROM pizza_sales
```

```
SELECT CAST(SUM (quantity) AS DECIMAL(10,2))/
```

```
CAST(COUNT(Distinct order_id) AS DECIMAL(10,2)) AS Avg_Pizza_Per_Order from pizza_sales
```

121 % 

Results  Messages

	Avg_Pizza_Per_Order
1	2.3219672131147

CHART REQUIREMENT

1.Daily Trend for Total Orders:

SQLQuery1.sql - VA...VAIDI\Vaidehi (70))*

```
SELECT * FROM pizza_sales
```

```
SELECT DATENAME(DW, order_date) as order_day, COUNT(DISTINCT order_id)
As Total_orders from pizza_sales
GROUP BY DATENAME(DW,order_date)
```

121 %

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

2.Monthly Trend for Total Orders:

SQLQuery1.sql - VA...VAIDI\Vaidehi (70))*

```
SELECT * FROM pizza_sales
```

```
SELECT DATENAME(MONTH, order_date) AS Month_Name, COUNT(DISTINCT order_id) AS Total_Orders
FROM pizza_sales
GROUP BY DATENAME(MONTH, order_date)
ORDER BY Total_Orders DESC
```

100 %

Results Messages

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

3. Percentage of Sales by Pizza Category:

SQLQuery1.sql - VA...VAIDI\Vaidehi (70))*

```
SELECT * FROM pizza_sales
```

```
SELECT pizza_category, sum(total_price) as Total_Sales, sum(total_price) * 100 /  
(SELECT sum(total_price) from pizza_sales WHERE MONTH (order_date) = 1) AS Percentage_Of_Sales  
from pizza_sales  
WHERE MONTH(order_date)=1  
GROUP BY pizza_category
```

100 %

Results Messages

	pizza_category	Total_Sales	Percentage_Of_Sales
1	Classic	18619.4000015259	26.6779189176038
2	Chicken	16188.75	23.1952780
3	Veggie	17055.4000778198	24.4370162489706
4	Supreme	17929.7499866486	25.6897867985821

4. Percentage of Sales by Pizza Size:

SQLQuery1.sql - VA...VAIDI\Vaidehi (70))*

```
SELECT * FROM pizza_sales
```

```
SELECT pizza_size, sum(total_price) as Total_Sales, sum(total_price) * 100 /  
(SELECT sum(total_price) from pizza_sales) AS Percentage_Of_Sales  
from pizza_sales  
GROUP BY pizza_size  
ORDER BY Percentage_Of_Sales DESC
```

100 %

Results Messages

	pizza_size	Total_Sales	Percentage_Of_Sales
1	L	375318.701004028	45.8903330244889
2	M	249382.25	30.492044420599
3	S	178076.49981308	21.7734684107037
4	XL	14076	1.72107684995364
5	XXL	1006.6000213623	0.123077294254725

5. Total Pizzas Sold By Pizza Category:

SQLQuery1.sql - VA...VAIDI\Vaidehi (70))*

```
SELECT * FROM pizza_sales
```

```
SELECT pizza_category, SUM(quantity) as Total_Quantity_Sold  
FROM pizza_sales  
WHERE MONTH(order_date) = 2  
GROUP BY pizza_category  
ORDER BY Total_Quantity_Sold DESC
```

100 %

Results Messages

	pizza_category	Total_Quantity_Sold
1	Classic	1178
2	Supreme	964
3	Veggie	944
4	Chicken	875

6. Top 5 Best Sellers by Revenue, Total Quantity and Total Orders:

By Revenue:

```
SQLQuery1.sql - VA...VAID(Vaidehi (70)) * X
```

```
SELECT * FROM pizza_sales
```

```
SELECT Top 5 pizza_name, SUM(total_price) AS Total_Revenue
```

```
FROM pizza_sales
```

```
GROUP BY pizza_name
```

```
ORDER BY Total_Revenue DESC
```

100 %

Results Messages

	pizza_name	Total_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 Spicy Italian Pizza	34831.25

By Quantity:

```
SQLQuery1.sql - VA...VAID(Vaidehi (70)) * X
```

```
SELECT * FROM pizza_sales
```

```
SELECT Top 5 pizza_name, SUM(quantity) AS Total_Pizza_Sold
```

```
FROM pizza_sales
```

```
GROUP BY pizza_name
```

```
ORDER BY Total_Pizza_Sold DESC
```

100 %

Results Messages

	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

By Orders:

```
SQLQuery1.sql - VA...VAID(Vaidehi (70)) * X
```

```
SELECT * FROM pizza_sales
```

```
SELECT Top 5 pizza_name, COUNT(DISTINCT order_id) AS Total_Orders
```

```
FROM pizza_sales
```

```
GROUP BY pizza_name
```

```
ORDER BY Total_Orders DESC
```

100 %

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

7. Bottom 5 Best Sellers by Revenue, Total Quantity and Total Orders:

By Revenue:

```
SQLQuery1.sql - VA...VAIDI\Vaidehi (70))* - X
```

```
SELECT * FROM pizza_sales
```

```
SELECT Top 5 pizza_name, SUM(total_price) AS Total_Revenue
FROM pizza_sales
GROUP BY pizza_name
ORDER BY Total_Revenue ASC
```

100 %

Results Messages

	pizza_name	Total_Revenue
1	The Brie Carre Pizza	11588.4998130798
2	The Green Garden Pizza	13955.75
3	The Spinach Supreme Pizza	15277.75
4	The Mediterranean Pizza	15360.5
5	The Spinach Pesto Pizza	15596

By Quantity :

```
SQLQuery1.sql - VA...VAIDI\Vaidehi (70))* - X
```

```
SELECT * FROM pizza_sales
```

```
SELECT TOP 5 pizza_name, SUM(quantity) AS Total_Pizza_Sold
FROM pizza_sales
GROUP BY pizza_name
ORDER BY Total_Pizza_Sold ASC
```

100 %

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

By Orders :

```
SQLQuery1.sql - VA...VAIDI\Vaidehi (70))* - X
```

```
SELECT * FROM pizza_sales
```

```
SELECT Top 5 pizza_name, COUNT(DISTINCT order_id) AS Total_Orders
FROM pizza_sales
GROUP BY pizza_name
ORDER BY Total_Orders ASC
```

100 %

Results Messages

	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