

PIZZA SALES SQL QUERIES

A. KPI's

1. Total Revenue:

```
SELECT SUM(total_price) AS Total_Revenue FROM pizza_sales
```

The screenshot shows a SQL query results window. At the top, it displays "133 %" and "No issues found". Below this, there are two tabs: "Results" and "Messages", with "Results" being the active tab. The results table has one column labeled "Total_Revenue" and one row with the value "817860.05083847".

	Total_Revenue
1	817860.05083847

2. Average Order Value:

```
SELECT SUM(total_price) / COUNT(DISTINCT order_id) AS  
Avg_Order_Value FROM pizza_sales
```

The screenshot shows a SQL query results window. At the top, it displays "133 %" and "No issues found". Below this, there are two tabs: "Results" and "Messages", with "Results" being the active tab. The results table has one column labeled "Avg_Order_Value" and one row with the value "38.3072623343546".

	Avg_Order_Value
1	38.3072623343546

3. Total Pizza Sales:

```
SELECT SUM(quantity) AS Total_Pizza_Sales FROM pizza_sales
```

133 % ▼ No issues found

Results Messages

	Total_Pizza_Sales
1	49574

4. Total Orders:

```
SELECT COUNT(DISTINCT order_id) AS Total_Orders FROM
pizza_sales
```

133 % ▼ No issues found

Results Messages

	Total_Orders
1	21350

5. Average Pizzas per Order:

```
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
```

133 % ▼ No issues found

Results Messages

	Avg_Pizzas_per_Order
1	2.32

B. Daily Trend for Total Orders

```
SELECT DATENAME(DW, order_date) AS Order_Day, COUNT(DISTINCT order_id) AS
Total_Orders
```

```
FROM pizza_sales  
GROUP BY DATENAME(DW, order_date)
```

The screenshot shows a SQL Server Management Studio window with the following details:

- Top status bar: 133 %, No issues found.
- Tab bar: Results (selected), Messages.
- Table data:

	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

```
SELECT DATEPART(HOUR, order_time) AS Orders_Hours, COUNT(DISTINCT order_id)  
AS Total_Orders  
FROM pizza_sales  
GROUP BY DATEPART(HOUR, order_time)  
ORDER BY DATEPART(HOUR, order_time)
```

The screenshot shows a SQL Server Management Studio window with the following details:

- Top status bar: 133 %, No issues found.
- Tab bar: Results (selected), Messages.
- Table data:

	Orders_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. Percentage of Sales by Pizza Category

```
SELECT pizza_category, SUM(total_price) AS Total_Sales, sum(total_price) *  
100 / (SELECT SUM(total_price) FROM pizza_sales) AS PCT  
FROM pizza_sales  
GROUP BY pizza_category
```

The screenshot shows a software interface for running SQL queries. At the top, there is a status bar with "133 %", a dropdown arrow, and a green checkmark icon followed by the text "No issues found". Below this is a toolbar with two tabs: "Results" (which is selected and highlighted in blue) and "Messages". The main area is a table titled "Results" with the following data:

	pizza_category	Total_Sales	PCT
1	Classic	220053.100021362	26.9059602306976
2	Chicken	195919.5	23.9551375322885
3	Veggie	193690.451004028	23.6825910258677
4	Supreme	208196.99981308	25.4563112111462

E. Percentage of Sales by Pizza Size

```
SELECT pizza_size, SUM(total_price) AS Total_Sales,  
CAST(SUM(total_price) * 100 /  
(SELECT SUM(total_price) FROM pizza_sales) AS DECIMAL(10,2))  
AS PCT  
FROM pizza_sales  
GROUP BY pizza_size  
ORDER BY PCT DESC
```

The screenshot shows a software interface with a status bar at the top indicating "133 %". Below the status bar is a message "No issues found" with a green checkmark icon. The main area contains two tabs: "Results" and "Messages", with "Results" selected. A table is displayed with the following data:

	pizza_size	Total_Sales	PCT
1	L	375318.701004028	45.89
2	M	249382.25	30.49
3	S	178076.49981308	21.77
4	XL	14076	1.72
5	XXL	1006.6000213623	0.12

F. Total Pizzas Sold by Pizza Category

```
SELECT pizza_category, SUM(quantity) AS Total_Pizzas_Sold
FROM pizza_sales
GROUP BY pizza_category
```

The screenshot shows a software interface with a table titled "Results" and a "Messages" tab. The table displays the following data:

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

G. Top 5 Best Sellers by Total Pizzas Sold

```
SELECT Top 5 pizza_name, SUM(quantity) AS Total_Pizzas_Sold
FROM pizza_sales
GROUP BY pizza_name
ORDER BY Total_Pizzas_Sold
```

Results Messages

	pizza_name	Total_Pizzas_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

H. Bottom 5 Best Sellers by Total Pizzas Sold

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

Results Messages

	pizza_name	Total_Pizzas_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

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.*