

Pizza Sales SQL Queries

A. KPI

1. Total Revenue:

The screenshot shows a SQL Server Management Studio (SSMS) interface. The top bar displays the connection information: 'SQLQuery1.s...abdul (52)*' with icons for refresh, close, and other operations. The main area contains a script window with three numbered lines of SQL code:

```
1 select * from pizza_sales
2
3 Select sum (total_price) AS total_revenue from pizza_sales
```

Below the script window, the status bar indicates '100 %' and 'No issues found'. The bottom pane is divided into 'Results' and 'Messages' tabs, with the 'Results' tab selected. It displays a single row of data in a table format:

	total_revenue
1	817860.05083847

2. Average Order Value

The screenshot shows a SQL query window in SQL Server Management Studio. The query is:

```
1  Select * FROM pizza_sales
2
3  Select SUM (total_price) / COUNT (DISTINCT order_id) AS avg_order_value from pizza_sales
```

The results pane shows one row of data:

avg_order_value
1 38.3072623343546

Metadata at the bottom of the results pane indicates: 100 %, No issues found, Ln: 3, Ch: 89 | (88 chars, 1 lines) | TABS.

3. Total Pizza Sold

The screenshot shows a SQL Server Management Studio (SSMS) interface. At the top, there is a title bar with the text "SQLQuery1.s...abdul (52)*" and three icons: a downward arrow, a refresh symbol, and a close (X) button. Below the title bar is a code editor window containing the following SQL query:

```
1 Select * FROM pizza_sales
2
3 SELECT SUM(quantity) AS total_pizza_sold from pizza_sales
```

Below the code editor is a status bar with the text "100 %" and a green checkmark icon followed by the message "No issues found". Underneath the status bar are two tabs: "Results" and "Messages". The "Results" tab is selected and displays a single row of data in a table:

	total_pizza_sold
1	49574

4. Total Order Placed

The screenshot shows a SQL Server Management Studio (SSMS) interface. In the top-left corner, there's a tab labeled "SQLQuery1.s...abdul (52)*". Below the tabs, the code editor contains the following T-SQL script:

```
1      Select * FROM pizza_sales
2
3      SELECT COUNT (DISTINCT order_id) AS Total_orders from pizza_sales |
```

In the bottom right corner of the code editor, there are status indicators: "Ln: 3, Ch: 67 | (". Below the code editor is a toolbar with a "Results" tab selected and a "Messages" tab next to it. The results pane displays a single row of data:

Total_orders
21350

5. Average Pizza Per Order

The screenshot shows a SQL query window titled "SQLQuery1.s...abdul (52)*". The query calculates the average quantity of pizzas per order from the "pizza_sales" table. The results show a single row with the value 2.32.

```
SQLQuery1.s...abdul (52)*  X
1   Select * FROM pizza_sales
2
3   Select CAST(CAST (SUM(quantity) AS DECIMAL(10,2))/(
4       CAST (COUNT(DISTINCT order_id) AS DECIMAL(10,2)) AS DECIMAL (10,2))
5   AS Average_pizza_per_order from pizza_sales

100 %    No issues found      Ln: 3, Ch: 1
Results  Messages
Average_pizza_per_order
1  2.32
```

CHARTS REQUIREMENT

B. Daily Trend

SQLQuery1.s...abdul (52)* ✎ X

```
1      Select * FROM pizza_sales
2
3      --Daily Trend
4      SELECT DATENAME(DW, order_date) as order_day, COUNT(DISTINCT order_id)
5      AS Total_orders from pizza_sales
6      GROUP BY DATENAME(DW, order_date)
7
```

100 % No issues found

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

```
SQLQuery1.s...abdul (52)* ⌂ X
1 Select * FROM pizza_sales
2
3 --Daily Trend
4 SELECT DATENAME(DW, order_date) AS order_day, COUNT(DISTINCT order_id)
5 AS Total_orders FROM pizza_sales
6 GROUP BY DATENAME(DW, order_date)
7
8 --Hourly Trend
9 SELECT DATEPART(HOUR, order_time) AS order_hours, COUNT(DISTINCT order_id)
10 AS Total_orders FROM pizza_sales
11 GROUP BY DATEPART(HOUR, order_time)
12 ORDER BY DATEPART(HOUR, order_time)
```

100 % ▾ ● No issues found Ln: 9, Ch: 1 | (183 chars, 4 lines)

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. Percentage of sales by pizza category

The screenshot shows a SQL Server Management Studio (SSMS) interface. The top pane is a query editor window titled "SQLQuery1.s...abdul (52)*". It contains the following T-SQL code:

```
1 Select * FROM pizza_sales
2
3 SELECT pizza_category, sum(total_price) as Total_Sales, sum(total_price) * 100 /
4 (SELECT sum(total_price) from pizza_sales WHERE MONTH(order_date) = 1) AS PCT
5 from pizza_sales
6 WHERE MONTH(order_date) = 1
7 GROUP BY pizza_category |
```

The bottom pane is a results grid titled "Results". It displays the output of the query, showing the percentage of total sales for each pizza category in January:

	pizza_category	Total_Sales	PCT
1	Classic	18619.4000015259	26.6779189176038
2	Chicken	16188.75	23.1952780348435
3	Veggie	17055.4000778198	24.4370162489706
4	Supreme	17929.7499866486	25.6897867985821

E. TOTAL % OF PIZZA BY SIZE

```
SQLQuery1.s...abdul (52)* ✘ ×
1   Select * FROM pizza_sales
2
3   SELECT pizza_size, cast(sum(total_price) AS DECIMAL(10,2))
4       as Total_Sales, CAST(sum(total_price) * 100 /
5           (SELECT sum(total_price) from pizza_sales WHERE DATEPART (quarter, order_date)=1)
6           AS DECIMAL(10,2)) AS PCT
7   from pizza_sales
8   WHERE DATEPART (quarter, order_date)=1
9   GROUP BY pizza_size
10  ORDER BY PCT DESC |
```

100 % ▾ No issues found Ln: 10, Ch: 19 TABS

Results Messages

	pizza_size	Total_Sales	PCT
1	L	95229.65	46.37
2	M	61159.00	29.78
3	S	45384.25	22.10
4	XL	3289.50	1.60
5	XXL	287.60	0.14

F. TOTAL NUMBER OF PIZZA SOLD BY CATEGORY

```
SQLQuery1.s...abdul (52)* ✘ X
1   Select * FROM pizza_sales
2
3   --SELECT pizza_size, cast(sum(total_price) AS DECIMAL(10,2))
4   --as Total_Sales, CAST(sum(total_price) * 100 /
5   --(SELECT sum(total_price) from pizza_sales WHERE DATEPART (quarter, order_date)=1)
6   --AS DECIMAL(10,2)) AS PCT
7   --from pizza_sales
8   --WHERE DATEPART (quarter, order_date)=1
9   --GROUP BY pizza_size
10  --ORDER BY PCT DESC
11
12  SELECT pizza_category, sum(quantity) as total_pizzas_sold
13  from pizza_sales
14  group by pizza_category
```

100 % ✓ No issues found Ln: 12, Ch: 1 (100 chars, 3 lines) TABS

Results Messages

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

G. TOP 5 BEST SELLER PIZZAS

```
SQLQuery1.s...abdul (52)* ✎ X
1   Select * FROM pizza_sales
2
3   --SELECT pizza_size, cast(sum(total_price) AS DECIMAL(10,2))
4   --as Total_Sales, CAST(sum(total_price) * 100 /
5   --(SELECT sum(total_price) from pizza_sales WHERE DATEPART (quarter, order_date)=1)
6   --AS DECIMAL(10,2)) AS PCT
7   --from pizza_sales
8   --WHERE DATEPART (quarter, order_date)=1
9   --GROUP BY pizza_size
10  --ORDER BY PCT DESC
11
12  SELECT TOP 5 pizza_name, sum (quantity) as total_pizzas_sold
13  from pizza_sales
14  group by pizza_name
15  order by sum(quantity) desc
```

100 % ▾ 0 No issues found Ln: 15, Ch: 29

Results Messages

	pizza_name	total_pizzas_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 pizza's

```
SQLQuery1.s...abdul (52)* ✘ ×
1   Select * FROM pizza_sales
2
3   --SELECT pizza_size, cast(sum(total_price) AS DECIMAL(10,2))
4   --as Total_Sales, CAST(sum(total_price) * 100 /
5   --(SELECT sum(total_price) from pizza_sales WHERE DATEPART (quarter, order_date)=1)
6   --AS DECIMAL(10,2)) AS PCT
7   --from pizza_sales
8   --WHERE DATEPART (quarter, order_date)=1
9   --GROUP BY pizza_size
10  --ORDER BY PCT DESC
11
12  SELECT TOP 5 pizza_name, sum (quantity) as total_pizzas_sold
13  from pizza_sales
14  group by pizza_name
15  order by sum(quantity) asc
```

100 % ✓ No issues found Ln: 15, Ch: 28 TA

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

I. Worst selling pizza done by month (worst 5)

```
SQLQuery1.s...abdul (52)* ✘ ×
1   Select * FROM pizza_sales
2
3   --SELECT pizza_size, cast(sum(total_price) AS DECIMAL(10,2))
4   --as Total_Sales, CAST(sum(total_price) * 100 /
5   --(SELECT sum(total_price) from pizza_sales WHERE DATEPART (quarter, order_date)=1)
6   --AS DECIMAL(10,2)) AS PCT
7   --from pizza_sales
8   --WHERE DATEPART (quarter, order_date)=1
9   --GROUP BY pizza_size
10  --ORDER BY PCT DESC
11
12  SELECT TOP 5 pizza_name, sum (quantity) as total_pizzas_sold
13  from pizza_sales
14  where month (order_date) = 1
15  group by pizza_name
16  order by sum(quantity) asc
```

100 % ▾ ● No issues found

Ln: 12, Ch: 1 | (160 chars, 5 lines) | T

Results Messages

	pizza_name	total_pizzas_sold
1	The Brie Carre Pizza	35
2	The Calabrese Pizza	67
3	The Mediterranean Pizza	68
4	The Green Garden Pizza	75
5	The Chicken Pesto Pizza	77