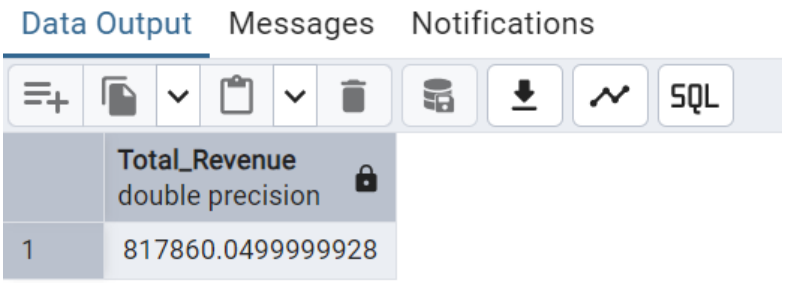
PIZZA SALES SQL QUERIES

A. KPI's

1. Write a SQL query to calculate the total revenue.

```
SELECT SUM(total_price) AS Total_Revenue FROM Pizza_Sales;
```

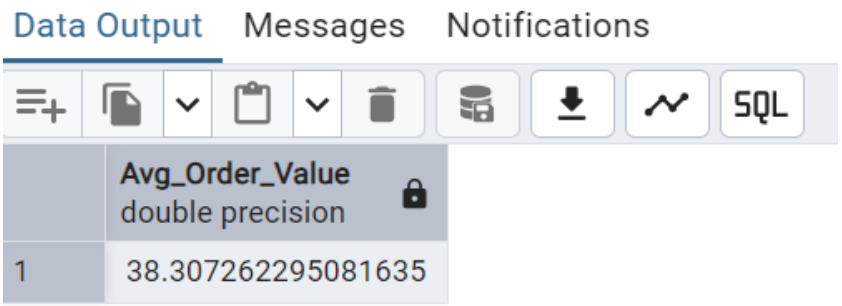
Output:



2. Write a SQL query to find the Average order value.

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

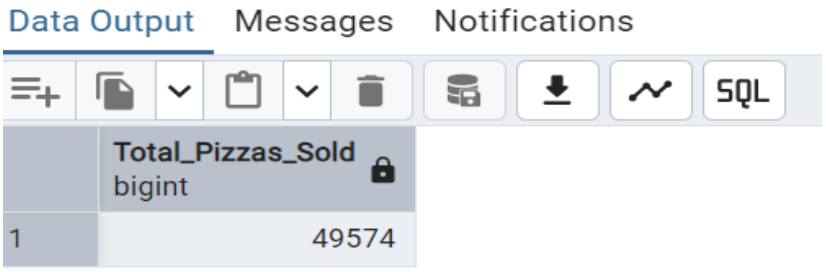
Output:



3. Write a SQL query to find the total pizzas sold.

```
SELECT SUM(quantity) AS Total_pizza_sold FROM pizza_sales;
```

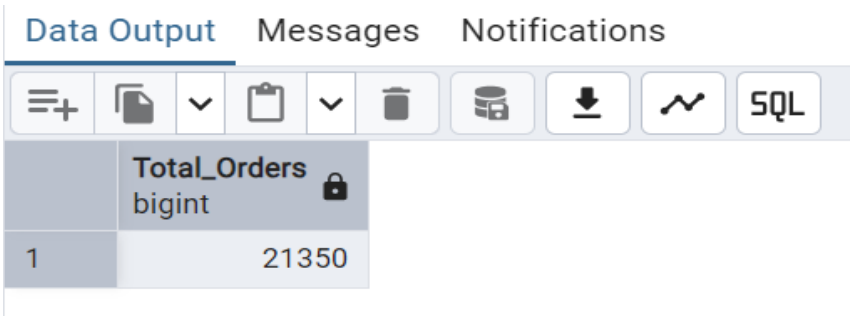
Output:



4. Write a SQL query to find the total number of orders placed.

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

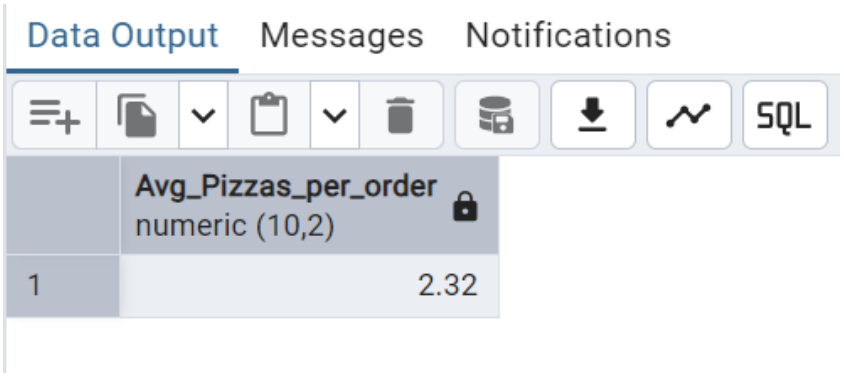
Output:



5. Write a SQL query to find the 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;
```

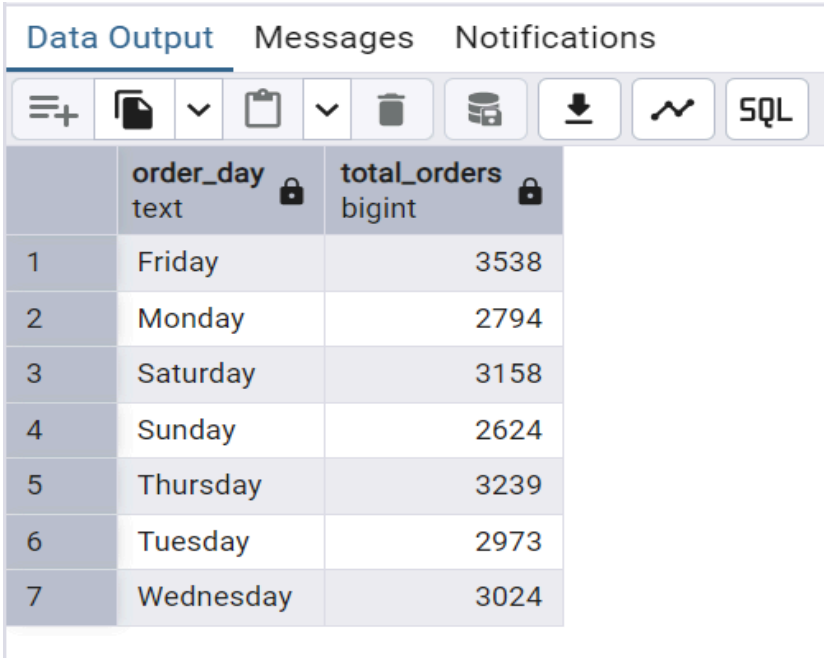
Output:



B. Daily Trend for Total Orders

```
SELECT TO_CHAR(order_date, 'Day') AS order_day, COUNT(DISTINCT order_id)
AS total_orders
FROM pizza_sales
GROUP BY order_date;
```

Output:



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

C. Hourly Trend for Orders

```
SELECT EXTRACT(HOUR FROM order_time)as “Order_hours”, COUNT(DISTINCT
order_id) as “Total_Orders”
from pizza_sales
group by “Order_hours”;
```

Output:

Data Output | Messages | Notifications

	Order_hours numeric	Total_Orders bigint
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

```
SELECT pizza_category, CAST(SUM(total_price) AS DECIMAL(10,2)) as
“Total_revenue”,
CAST(SUM(total_price) * 100 / (SELECT SUM(total_price) from pizza_sales)
AS DECIMAL(10,2)) AS % of sales
FROM pizza_sales
GROUP BY pizza_category;
```

Output:

Data Output | Messages | Notifications

	pizza_category character varying (50)	Total_revenue numeric (10,2)	% of sales numeric (10,2)
1	Supreme	208197.00	25.46
2	Chicken	195919.50	23.96
3	Veggie	193690.45	23.68
4	Classic	220053.10	26.91

E. % of Sales by Pizza Size

```
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)
AS DECIMAL(10,2)) AS % of sales
FROM pizza_sales
GROUP BY pizza_size
ORDER BY pizza_size;
```

Output:

Data Output | Messages | Notifications

	pizza_size character varying (50)	Total_revenue numeric (10,2)	% of sales numeric (10,2)
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

```
SELECT pizza_category, SUM(quantity) as Total_Quantity_Sold
FROM pizza_sales
GROUP BY pizza_category
ORDER BY Total_Quantity_Sold DESC;
```

Output:

Data Output | Messages | Notifications

	pizza_category character varying (50)	Total_Quantity_Sold bigint
1	Classic	14888
2	Supreme	11987
3	Veggie	11649
4	Chicken	11050

G. Top 5 Best Sellers by Total Pizzas Sold

```
SELECT pizza_name, SUM(quantity) AS Total_Pizza_Sold
FROM pizza_sales
GROUP BY pizza_name
ORDER BY Total_Pizza_Sold DESC Limt 5;
```

Output:

Data Output | Messages | Notifications

	pizza_name character varying (50)	Total_Pizza_Sold bigint
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 worst Sellers by Total Pizzas Sold

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

Output:

Data Output | Messages | Notifications

	pizza_name character varying (50)	Total_Pizza_Sold bigint
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

- KASULA SAITEJACHARY