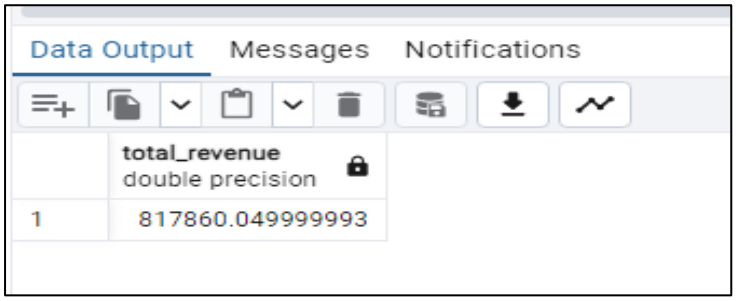
PIZZAS SALES SQL QUERIES

A. KPIs

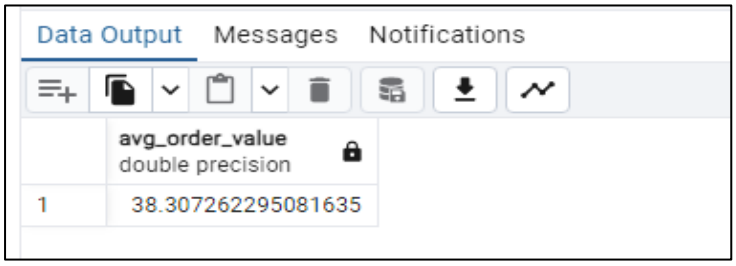
1. Total Revenue

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



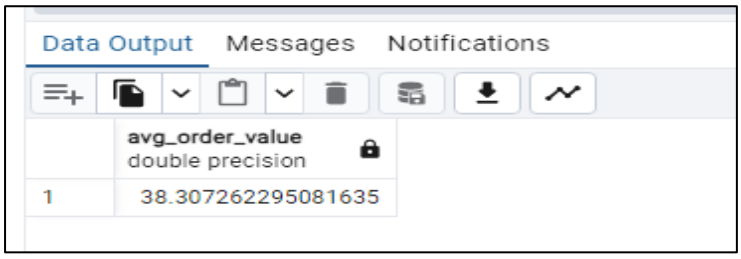
2. Average Order Value

```
SELECT SUM(total_price)/COUNT(DISTINCT order_id) AS Avg_order_value FROM sales
```



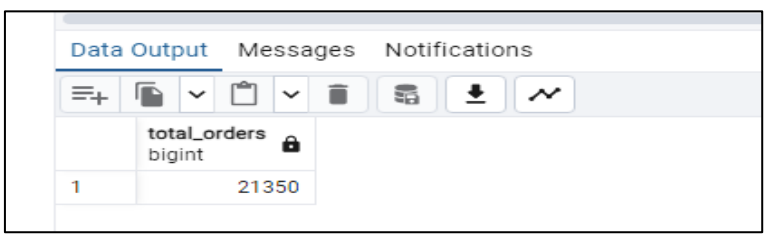
3. Total Pizzas sold

```
SELECT SUM(quantity) AS Total_pizzas_sold FROM sales
```



4. Total orders placed

```
SELECT COUNT(DISTINCT order_id) AS Total_orders FROM sales
```

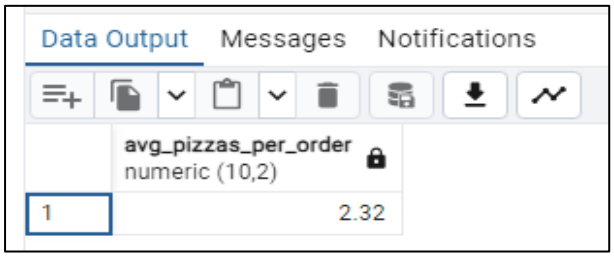


5. **Average Pizzas per order**

```
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 sales
```

AS DECIMAL (10,2) to obtain our result up to 2 decimal place.

10 signifies the no. of digits after decimal and 2 signifies the rounding off digits.

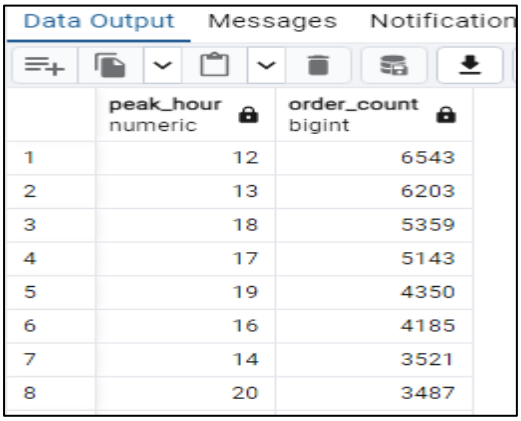


Data Output | Messages | Notifications

	avg_pizzas_per_order numeric (10,2)
1	2.32

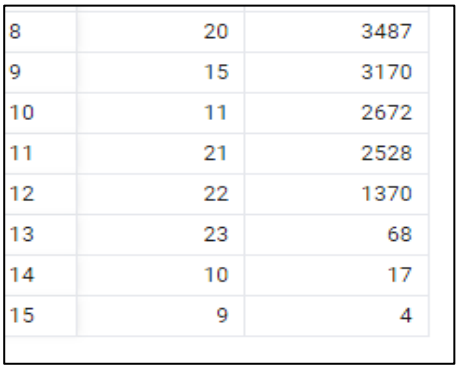
6. **Peak Hours of Pizza Sales**

```
SELECT EXTRACT(HOUR FROM order_time) AS peak_hour, COUNT(*) AS order_count
FROM sales
GROUP BY EXTRACT(HOUR FROM order_time)
ORDER BY order_count DESC;
```



Data Output | Messages | Notification

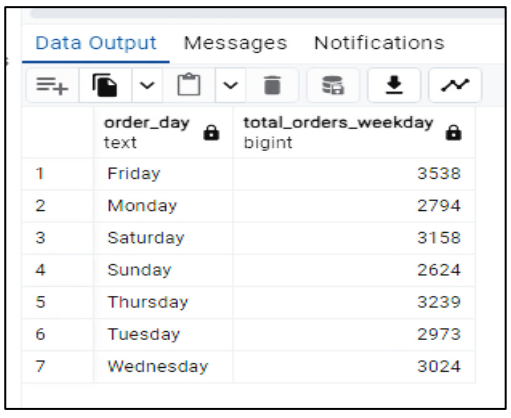
	peak_hour numeric	order_count bigint
1	12	6543
2	13	6203
3	18	5359
4	17	5143
5	19	4350
6	16	4185
7	14	3521
8	20	3487



8	20	3487
9	15	3170
10	11	2672
11	21	2528
12	22	1370
13	23	68
14	10	17
15	9	4

B. Daily Trends of Orders placed

```
SELECT TO_CHAR(order_date, 'Day') AS order_day, COUNT(DISTINCT order_id) AS
Total_orders FROM sales
GROUP BY order_day
```

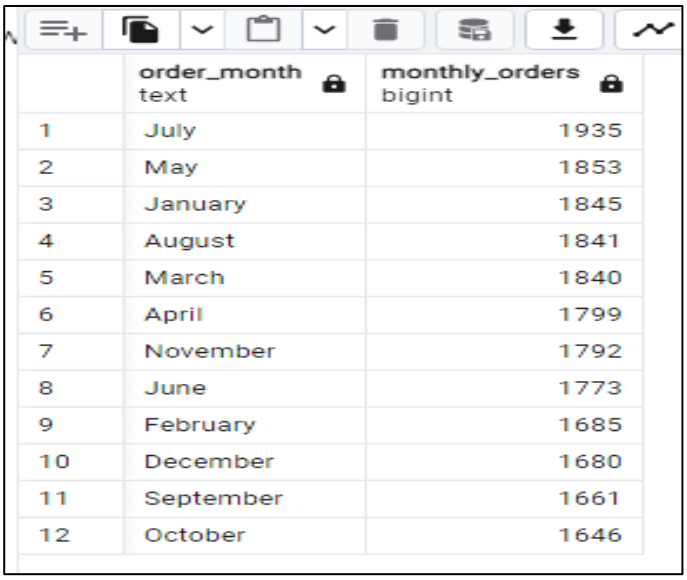


Data Output | Messages | Notifications

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

C. Monthly Trends of Orders placed

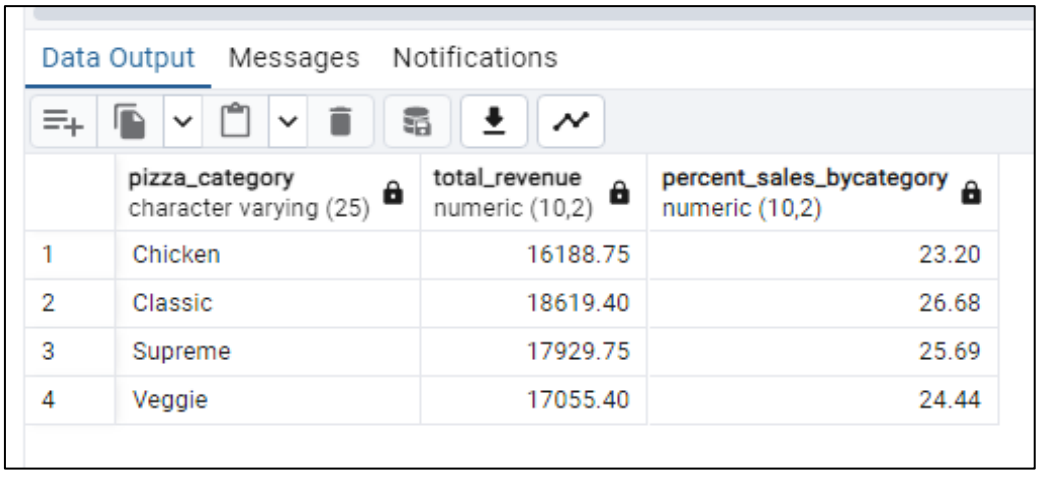
```
SELECT TRIM(TO_CHAR(order_date, 'Month')) AS order_month, COUNT(DISTINCT order_id)
AS monthly_orders FROM sales
GROUP BY order_month
ORDER BY monthly_orders DESC
```



	order_month text	monthly_orders bigint
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

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 sales WHERE
EXTRACT(Month FROM order_date) = 1) AS DECIMAL(10, 2)) AS percent_sales_bycategory
FROM  sales
WHERE EXTRACT (Month FROM order_date) = 1 -- 1 corresponds to the first quarter
GROUP BY pizza_category;
-- in place of EXTRACT(Month From...) we can also use
TRIM(TO_CHAR(column_name,'Month')=1)
```

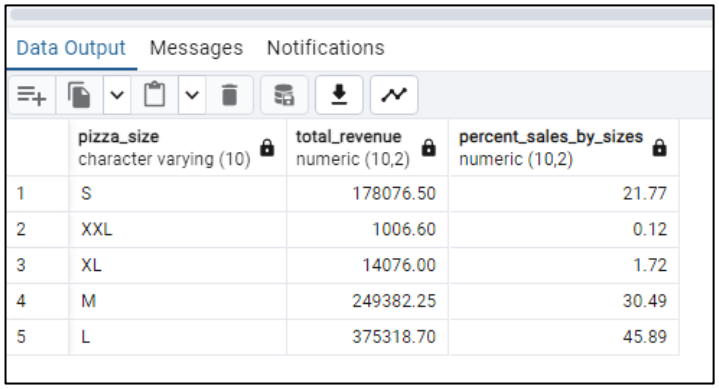


Data Output | Messages | Notifications

	pizza_category character varying (25)	total_revenue numeric (10,2)	percent_sales_bycategory numeric (10,2)
1	Chicken	16188.75	23.20
2	Classic	18619.40	26.68
3	Supreme	17929.75	25.69
4	Veggie	17055.40	24.44

E. % of Sales by pizza sales(pizza sizes)

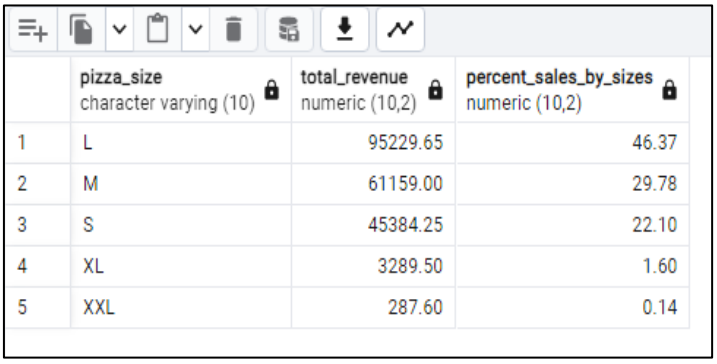
```
SELECT pizza_size,
CAST(SUM(total_price) AS DECIMAL(10,2)) AS total_revenue,
CAST(SUM(total_price)*100/(SELECT SUM(total_price) FROM sales WHERE
EXTRACT(QUARTER FROM order_date)=1) AS DECIMAL(10,2)) AS percent_sales_by_sizes
FROM sales
WHERE EXTRACT(QUARTER FROM order_date)=1    -- filtering data on the basis of month
and quarters
GROUP BY pizza_size
ORDER BY percent_sales_by_sizes DESC ;
--to get the entire data we can remove the WHERE clause.
```



Data Output | Messages | Notifications

	pizza_size character varying (10)	total_revenue numeric (10,2)	percent_sales_by_sizes numeric (10,2)
1	S	178076.50	21.77
2	XXL	1006.60	0.12
3	XL	14076.00	1.72
4	M	249382.25	30.49
5	L	375318.70	45.89

Result for entire data.

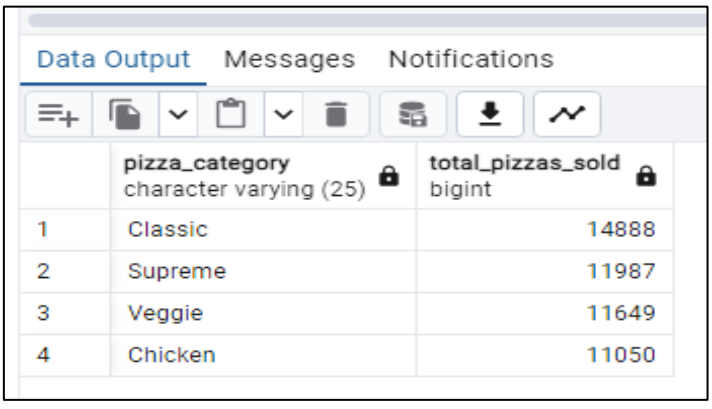


	pizza_size character varying (10)	total_revenue numeric (10,2)	percent_sales_by_sizes numeric (10,2)
1	L	95229.65	46.37
2	M	61159.00	29.78
3	S	45384.25	22.10
4	XL	3289.50	1.60
5	XXL	287.60	0.14

Result for WHERE clause

F. Total pizzas sold by Pizza Category

```
SELECT pizza_category, SUM(quantity) AS total_pizzas_sold FROM sales
--WHERE EXTRACT(Month FROM order_date)=2
GROUP BY pizza_category
ORDER BY total_pizzas_sold DESC;
```

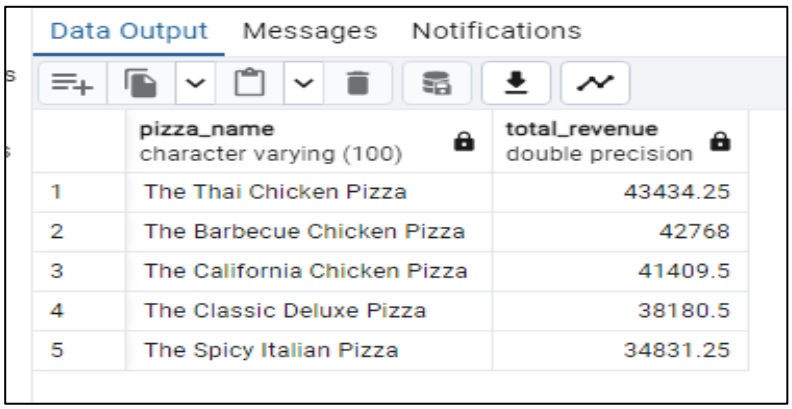


Data Output | Messages | Notifications

	pizza_category character varying (25)	total_pizzas_sold bigint
1	Classic	14888
2	Supreme	11987
3	Veggie	11649
4	Chicken	11050

G. Top 5 Best Selling Pizzas by Revenue

```
SELECT pizza_name, SUM(total_price) AS total_revenue FROM sales
GROUP BY pizza_name
ORDER BY total_revenue DESC
LIMIT 5
```



Data Output | Messages | Notifications

	pizza_name character varying (100)	total_revenue double precision
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

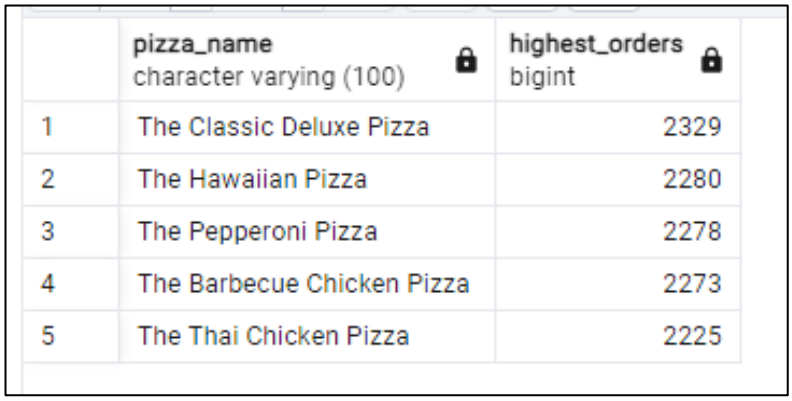
```
SELECT pizza_name, SUM(quantity) AS highest_quantity FROM sales
GROUP BY pizza_name
ORDER BY highest_quantity DESC
LIMIT 5
```

Data Output | Messages | Notifications

	pizza_name character varying (100)	highest_quantity 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

by Count of order_ID

```
SELECT pizza_name, COUNT(DISTINCT order_id) AS highest_orders FROM sales
GROUP BY pizza_name
ORDER BY highest_orders DESC
LIMIT 5
```



	pizza_name character varying (100)	highest_orders bigint
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

H. Bottom 5 Worst Selling Pizzas by Revenue

```
SELECT pizza_name, CAST(SUM(total_price) AS DECIMAL(10,2)) AS total_revenue_worst
FROM sales
GROUP BY pizza_name
ORDER BY total_revenue ASC
LIMIT 5
```

Data Output | Messages | Notifications

	pizza_name character varying (100)	total_revenue_worst numeric (10,2)
1	The Brie Carre Pizza	11588.50
2	The Green Garden Pizza	13955.75
3	The Spinach Supreme Pizza	15277.75
4	The Mediterranean Pizza	15360.50
5	The Spinach Pesto Pizza	15596.00

by Quantity

```
SELECT pizza_name, SUM(quantity) AS lowest_quantity FROM sales
GROUP BY pizza_name
ORDER BY lowest_quantity ASC
LIMIT 5
```

	pizza_name character varying (100)	lowest_quantity 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

by Count of Order ID

```
SELECT pizza_name, COUNT(DISTINCT order_id) AS lowest_orders FROM sales
GROUP BY pizza_name
ORDER BY lowest_orders ASC
LIMIT 5
```

Data Output | Messages | Notifications

	pizza_name character varying (100)	lowest_orders bigint
1	The Brie Carre Pizza	480
2	The Mediterranean Pizza	912
3	The Calabrese Pizza	918
4	The Spinach Supreme Pizza	918
5	The Chicken Pesto Pizza	938

- **Report by**

  Hafsha Wahab