

SALES QUERIES(PostgreSQL)

A. KPI's

1. **Total Revenue:** SELECT SUM(total_price) AS Total_Revenue FROM Pizza_Sales;

Data Output		Messages
total_revenue	numeric	

1	817860.05
---	-----------

2. **Average Order Value:** SELECT SUM(total_price) / COUNT(DISTINCT(order_id)) AS Avg_Order_Value FROM Pizza_Sales;

Data Output		Messages	Notifications
avg_order_value	numeric		
1	38.3072622950819672		

3. **Total Pizza Sold:** SELECT SUM(quantity) AS Total_pizzas_sold FROM Pizza_Sales;

Data Output		Messages
total_pizzas_sold	bigint	
1	49574	

4. **Total Orders:** SELECT COUNT(DISTINCT(order_id)) AS Total_Orders FROM Pizza_Sales;

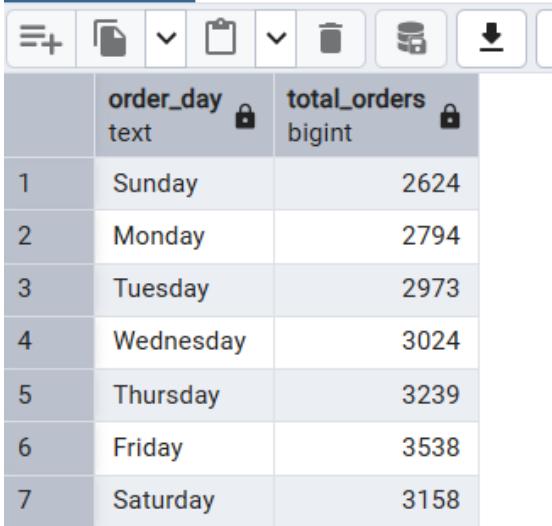
Data Output		Messages
total_orders	bigint	
1	21350	

5. **Average Pizza Per Order:** SELECT CAST (CAST(SUM(quantity) AS NUMERIC(10,2))/COUNT(DISTINCT(order_id)) AS NUMERIC(10,2)) AS Avg_Pizzas_Per_Order FROM Pizza_Sales;

Data Output		Messages	Noti
avg_pizzas_per_order	numeric (10,2)		
1	2.32		

6. **Daily Trend:** SELECT TO_CHAR(order_date,'Day') AS order_day , COUNT(DISTINCT(order_id)) AS total_orders FROM Pizza_Sales
 GROUP BY TO_CHAR(order_date,'Day'), TO_CHAR(order_date,'D')
 ORDER BY TO_CHAR(order_date,'D');

Data Output Messages Notifications

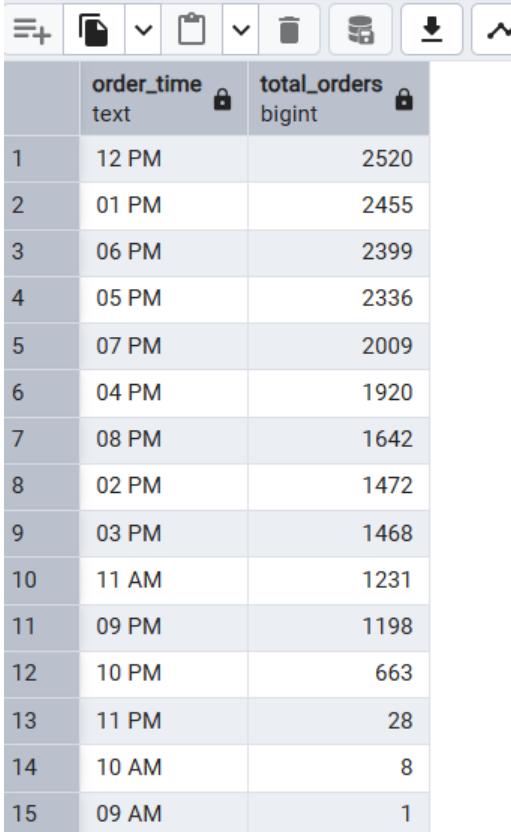


A screenshot of a database query results interface. At the top, there are tabs for "Data Output", "Messages", and "Notifications", with "Data Output" being the active tab. Below the tabs is a toolbar with various icons for file operations like copy, paste, and save. The main area is a table with two columns: "order_day" and "total_orders". The "order_day" column contains values 1 through 7, corresponding to the days of the week: Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, and Saturday. The "total_orders" column shows the count of distinct orders for each day: 2624, 2794, 2973, 3024, 3239, 3538, and 3158 respectively.

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

7. **Hourly Trend:** SELECT TO_CHAR(order_time,'HH12 AM') AS order_Time ,
 COUNT(DISTINCT(order_id)) AS total_orders
 FROM Pizza_Sales
 GROUP BY TO_CHAR(order_time,'HH12 AM')
 ORDER BY total_orders DESC;

Data Output Messages Notifications



A screenshot of a database query results interface. At the top, there are tabs for "Data Output", "Messages", and "Notifications", with "Data Output" being the active tab. Below the tabs is a toolbar with various icons for file operations like copy, paste, and save. The main area is a table with two columns: "order_time" and "total_orders". The "order_time" column contains values from 1 to 15, representing hours from 12 PM down to 09 AM. The "total_orders" column shows the count of distinct orders for each hour: 2520, 2455, 2399, 2336, 2009, 1920, 1642, 1472, 1468, 1231, 1198, 663, 28, 8, and 1 respectively.

	order_time	total_orders
	text	bigint
1	12 PM	2520
2	01 PM	2455
3	06 PM	2399
4	05 PM	2336
5	07 PM	2009
6	04 PM	1920
7	08 PM	1642
8	02 PM	1472
9	03 PM	1468
10	11 AM	1231
11	09 PM	1198
12	10 PM	663
13	11 PM	28
14	10 AM	8
15	09 AM	1

8. Percentage of Sales by Pizza Category (eg: Month=1)

```
SELECT
    pizza_category, SUM(total_price) AS total_sales, SUM(total_price)*100 / (SELECT
        SUM(total_price) FROM Pizza_sales WHERE
        EXTRACT(Month From order_date)=1) AS Percentage_of_sales
    FROM Pizza_Sales
    WHERE
        EXTRACT(MONTH FROM order_date)=1
    GROUP BY pizza_category
    ORDER BY Percentage_of_sales DESC;
```

Data Output Messages Notifications

The screenshot shows a database interface with tabs for Data Output, Messages, and Notifications. The Data Output tab is selected. Below it is a toolbar with various icons. The main area displays a table with four columns: pizza_category, total_sales, and percentage_of_sales. The data shows four rows for Classic, Supreme, Veggie, and Chicken pizzas.

	pizza_category character varying (50)	total_sales numeric	percentage_of_sales numeric
1	Classic	18619.40	26.6779189406432996
2	Supreme	17929.75	25.6897868420034588
3	Veggie	17055.40	24.4370161605770181
4	Chicken	16188.75	23.1952780567762235

9. Percentage Of Sale by Pizza Size:

```
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;
```

Data Output Messages Notifications

The screenshot shows a database interface with tabs for Data Output, Messages, and Notifications. The Data Output tab is selected. Below it is a toolbar with various icons. The main area displays a table with four columns: pizza_size, total_sales, and percentage_of_sales. The data shows five rows for L, M, S, XL, and XXL pizza sizes.

	pizza_size character varying (50)	total_sales numeric	percentage_of_sales numeric
1	L	375318.70	45.8903329487728371
2	M	249382.25	30.4920444518594593
3	S	178076.50	21.7734684558806852
4	XL	14076.00	1.7210768517180904
5	XXL	1006.60	0.12307729176892794800

10. Total Pizza Sold Category Wise:

```
SELECT pizza_category, SUM(quantity) AS Total_sales
    FROM Pizza_Sales
    GROUP BY pizza_category
    ORDER BY Total_sales DESC;
```

Data Output Messages Notifications

The screenshot shows a database interface with tabs for Data Output, Messages, and Notifications. The Data Output tab is selected. Below it is a toolbar with various icons. The main area displays a table with two columns: pizza_category and total_sales. The data shows four rows for Classic, Supreme, Veggie, and Chicken pizzas.

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

11. Top 5 Seller: SELECT pizza_name , SUM(quantity) AS Sales_price FROM Pizza_Sales
GROUP BY pizza_name
ORDER BY Sales_price DESC
LIMIT 5;

Data Output Messages Notifications

	pizza_name character varying (50)	sales_price 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

12. Worst 5 Seller : SELECT pizza_name , SUM(quantity) AS Sales_price FROM Pizza_Sales
GROUP BY pizza_name
ORDER BY Sales_price DESC
LIMIT 5;

Data Output Messages Notifications

	pizza_name character varying (50)	sales_price 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