**Pizza Sales SQL- quries**

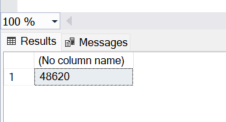
**Check No. of Rows**

Query:

SELECT Count (\*)

FROM pizza;

Result:



**Check number of Columns**

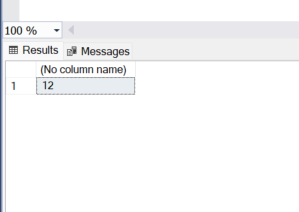
Query:

SELECT COUNT (\*)

FROM INFORMATION\_SCHEMA.COLUMNS

WHERE Table\_Name = 'pizza';

Result:

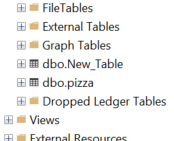


**Create Backup\_Table**

Query:

SELECT \* INTO New\_Table FROM pizza;

Result:



**How to safely Test (Alter Data)**

Query:

BEGIN TRANSACTION;

ALTER TABLE pizza ADD new\_column int;

-- COMMIT ( to apply )

--ROLLBACK; (to Undo)

Result:

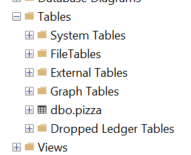


**Rename Table**

Query:

EXEC sp\_rename 'pizza\_sales', 'pizza';

Result:

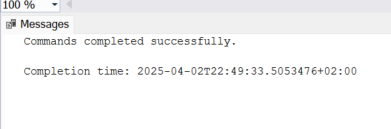


**Delete Table**

Query:

DROP TABLE Table\_1;

Result:

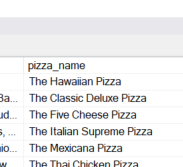


**Rename Column**

Query:

EXEC sp\_rename 'pizza.pizza\_nnme', 'pizza\_name', 'column';

Result:



**Add column**

Query:

ALTER TABLE pizza ADD new\_col int;

Result:



**INSERT values into COLUMN**

Query:

UPDATE pizza SET new\_col = unit\_price \* quantity;

Result:



**Remove Column**

Query:

ALTER TABLE pizza ADD new\_col int;

Result:

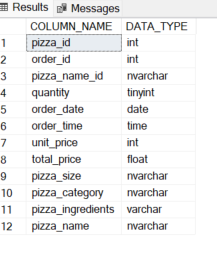


**Change Column Data Type**

Query:

ALTER TABLE pizza ALTER COLUMN unit\_price int;

Result:

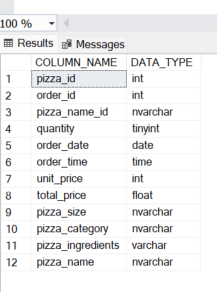


**Check Column names & Data types**

Query:

SELECT COLUMN\_NAME, DATA\_TYPE FROM INFORMATION\_SCHEMA.COLUMNS WHERE TABLE\_NAME = 'pizza';

Result:



**Check NULL value in Column**

Query:

--There is no Null value in column 'quantity'

SELECT \* from pizza

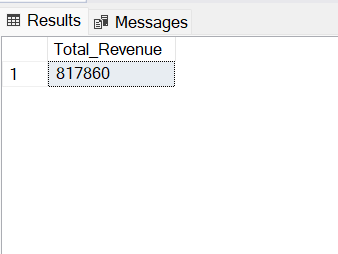
where quantity IS NULL;

**Total Revenue**

Query:

SELECT ROUND(SUM(total\_price),0) AS Total\_Revenue FROM pizza;

Result:



**Average Order Value**

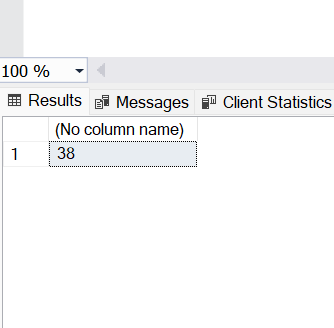
Query:

SELECT

ROUND (SUM(total\_price) / COUNT (DISTINCT order\_id),0) AS average\_order\_value

FROM pizza;

Result:

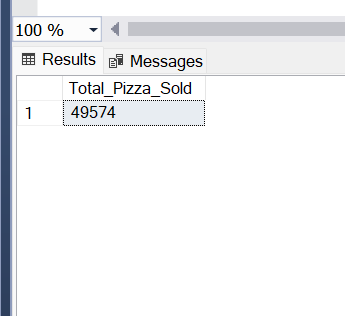


**Total Pizza Sold**

Query:

SELECT SUM(quantity) AS Total\_Pizza\_Sold FROM pizza;

Result:



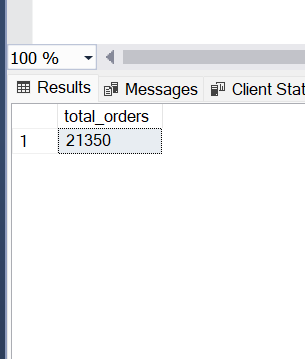
**Total Orders**

Query:

-- Orders are repeating so to just count one order once I used Distinct

SELECT COUNT(DISTINCT order\_id) AS total\_orders FROM pizza;

Result:



**Average Pizza Per Order**

Query:

SELECT

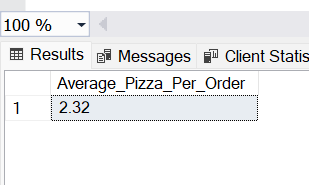
CAST(CAST(SUM(quantity) AS DECIMAL(10,2)) /

CAST(COUNT(DISTINCT order\_id) AS DECIMAL(10,2)) AS DECIMAL(10,2))

AS Average\_Pizza\_Per\_Order

FROM pizza;

Result:



**Total orders by day of week**

Query:

SELECT

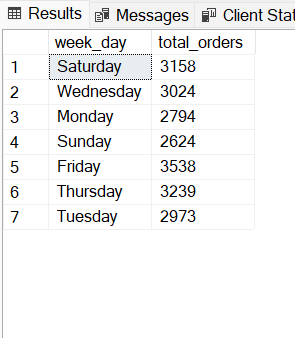
DATENAME(weekday,order\_date) AS week\_day,

COUNT(DISTINCT order\_id) AS total\_orders

FROM pizza

GROUP BY DATENAME(weekday,order\_date);

Result:



**Total Orders by Months**

Query:

SELECT

DATENAME(MONTH, order\_date) AS Name\_OF\_Month,

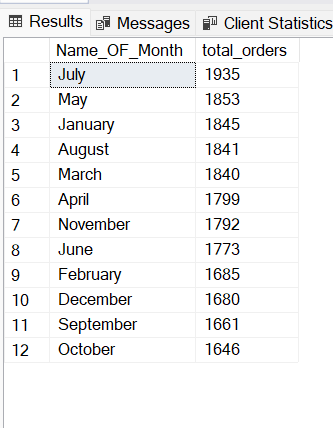
COUNT(DISTINCT order\_id) AS total\_orders

FROM pizza

GROUP BY DATENAME(MONTH, order\_date)

ORDER BY total\_orders DESC;

Result:



**Percentage of sale per pizza category**

Query:

SELECT

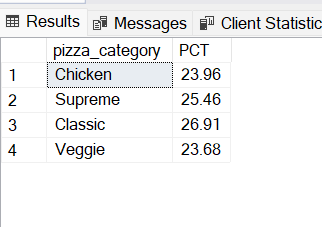
pizza\_category,

ROUND((SUM(total\_price) / (SELECT SUM(total\_price) FROM pizza)) \*100,2) AS PCT

FROM pizza

GROUP BY pizza\_category;

Result:



**Percentage of sale per pizza size**

Query:

SELECT

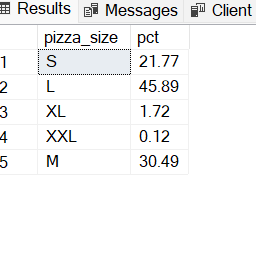
pizza\_size,

ROUND(SUM(total\_price) \*100 / (SELECT SUM(total\_price) FROM pizza), 2) AS pct

FROM pizza

GROUP BY pizza\_size;

Result:



**Total pizza sold by category**

Query:

SELECT

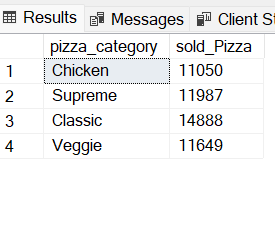
pizza\_category,

SUM(quantity) AS sold\_Pizza

FROM pizza

GROUP BY pizza\_category;

Result:



**TOP 5 pizza by revenue**

Query:

SELECT

TOP 5 pizza\_name,

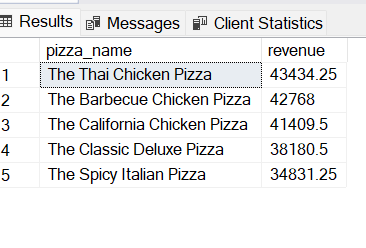
SUM(total\_price) AS revenue

FROM pizza

GROUP BY pizza\_name

ORDER BY revenue DESC;

Result:



**Bottom 5 pizza (name) by revenue**

Query:

SELECT

TOP 5 pizza\_name,

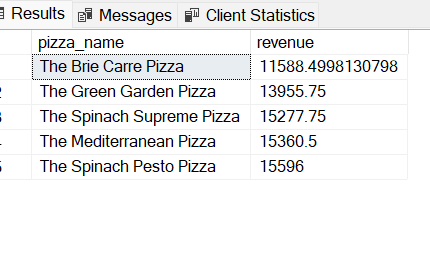
SUM(total\_price) AS revenue

FROM pizza

GROUP BY pizza\_name

ORDER BY revenue DESC;

Result:



**Top 5 pizza by Quantity**

Query:

SELECT

TOP 5 pizza\_name,

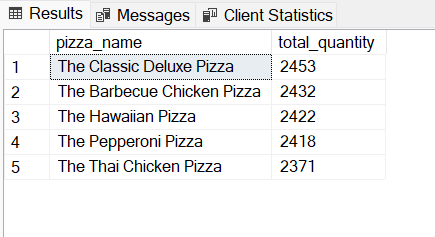
SUM(quantity) AS total\_quantity

FROM pizza

GROUP BY pizza\_name

ORDER BY total\_quantity DESC;

Result:



**Bottom 5 pizza by Quantity**

Query:

SELECT

TOP 5 pizza\_name,

SUM(quantity) AS total\_quantity

FROM pizza

GROUP BY pizza\_name

ORDER BY total\_quantity ASC;

Result:

