DATA ANALYSIS PROJECT – SQL

Retrieve all columns from the Sales_Analysis dataset in ascending order by sale_date.

```
SELECT TOP(5) * FROM Sales_Analysis ORDER BY sale_date ASC;
```

Results | Messages

	transactions_id	sale_date	sale_time	customer_id	gender	age	category	quantity	price_per_unit	cogs	total_sale
1	1269	2022-01-01	08:09:00	71	Male	25	Clothing	4	500.00	145.00	2000.00
2	1366	2022-01-01	11:34:00	25	Male	57	Clothing	2	50.00	23.50	100.00
3	958	2022-01-01	21:14:00	116	Male	62	Electronics	2	25.00	11.00	50.00
4	420	2022-01-02	10:53:00	28	Female	22	Clothing	4	500.00	200.00	2000.00
5	189	2022-01-02	09:44:00	143	Male	63	Beauty	1	50.00	14.50	50.00

----------DATA CLEANING----------

Modify the cogs column in the dataset to display two decimal places.

```
ALTER TABLE Sales_Analysis ALTER COLUMN cogs DECIMAL(10, 2);
```

Modify the price_per_unit column in the dataset to display two decimal places.

```
ALTER TABLE Sales_Analysis ALTER COLUMN price_per_unit DECIMAL(10, 2);
```

Modify the total_sale column in the dataset to display two decimal places.

```
ALTER TABLE Sales_Analysis ALTER COLUMN total_sale DECIMAL(10, 2);
```

Modify the sale_time column in the dataset to display hh:mm:ss format.

```
ALTER TABLE Sales_Analysis ALTER COLUMN sale_time TIME(0);
```

Check if there are any NULL values in the dataset.

```
SELECT * FROM Sales_Analysis
WHERE transactions_id IS NULL
      OR sale_date IS NULL
      OR sale_time IS NULL
         OR customer_id IS NULL
         OR gender IS NULL
         OR age IS NULL
         OR category IS NULL
         OR quantity IS NULL
         OR price_per_unit IS NULL
         OR cogs IS NULL
         OR total_sale IS NULL;
```

Delete rows with NULL values from the dataset.

```
DELETE FROM Sales_Analysis
WHERE transactions_id IS NULL
      OR sale_date IS NULL
      OR sale_time IS NULL
      OR customer_id IS NULL
      OR gender IS NULL
      OR age IS NULL
      OR category IS NULL
      OR quantity IS NULL
      OR price_per_unit IS NULL
      OR cogs IS NULL
      OR total_sale IS NULL;
```

Rename the column 'quantiy' to 'quantity'.

```
EXEC sp_rename 'Sales_Analysis.quantiy',  'quantity', 'COLUMN';
```

----------DATA ANALYSIS----------

Display the total number of transactions.

```
SELECT COUNT(*) AS 'Total Transactions' FROM Sales_Analysis;
```

Results | Messages

	Total Transactions
1	2000

Retrieve unique values from the category column.

```
SELECT DISTINCT category AS 'Category' FROM Sales_Analysis;
```

Results | Messages

	Category
1	Clothing
2	Electronics
3	Beauty

Retrieve the 'Total quantity' and 'Total sale' from the dataset.

```
SELECT COUNT(quantity) AS 'Total Quantity', SUM(total_sale) AS 'Total Sale' FROM
Sales_Analysis;
```

Results Messages

	Total Quantity	Total Sale
1	1997	911720.00

Retrieve 'Category', 'Gender' and 'the number of transactions', grouped by 'Category' and 'Gender'

```
SELECT category AS 'Category', gender AS 'Gender', COUNT(transactions_id) AS
'Transactions' FROM Sales_Analysis
GROUP BY category, gender
ORDER BY COUNT(transactions_id) DESC;
```

Results Messages

	Category	Gender	Transactions
1	Clothing	Male	354
2	Clothing	Female	348
3	Electronics	Male	344
4	Electronics	Female	340
5	Beauty	Female	332
6	Beauty	Male	282

Retrieve 'Category', 'Quantity' and 'the sum of total sales', grouped by 'Category'.

```
SELECT category AS 'Category' ,COUNT(quantity) AS 'Quantity', SUM(total_sale) AS
'Totale Sale' FROM Sales_Analysis
GROUP BY category
ORDER BY COUNT(quantity);
```

Results Messages

	Category	Quantity	Totale Sale
1	Beauty	612	286840.00
2	Electronics	684	313810.00
3	Clothing	701	311070.00

Retrieve the 'Year' and the count of 'Transactions' from the dataset.

```
SELECT YEAR(sale_date) AS 'Year', COUNT(transactions_id) AS 'Transactions' FROM
Sales_Analysis
GROUP BY YEAR(sale_date)
ORDER BY YEAR(sale_date);
```

Results Messages

	Year	Transactions
1	2022	979
2	2023	1021

Retrieve the total number of distinct customers.

```
SELECT COUNT(DISTINCT customer_id) AS 'Total Customers' FROM Sales_Analysis;
```

Results | Messages

	Total Customers
1	155

Retrieve 'Year' and the sum of 'Total sales', grouped by 'Year'.

```
SELECT YEAR(sale_date) AS 'Year', SUM(total_sale) AS 'Total Sales' FROM
Sales_Analysis
GROUP BY YEAR(sale_date)
ORDER BY YEAR(sale_date) ASC;
```

Results | Messages

	Year	Total Sales
1	2022	452825.00
2	2023	458895.00

Compare the total sales between 2022 and 2023 for each month.

```
SELECT MONTH(sale_date) AS 'Month',
COALESCE(SUM(CASE WHEN YEAR(sale_date) = 2022 THEN total_sale END), 0) AS 'Total
Sales 2022',
COALESCE(SUM(CASE WHEN YEAR(sale_date) = 2023 THEN total_sale END), 0) AS 'Total
Sales 2023'
FROM Sales_Analysis
WHERE YEAR(sale_date) IN (2022, 2023)
GROUP BY MONTH(sale_date)
ORDER BY MONTH(sale_date);
```

Results | Messages

	Month	Total Sales 2022	Total Sales 2023
1	1	22635.00	23790.00
2	2	16110.00	25170.00
3	3	24505.00	20530.00
4	4	28705.00	21925.00
5	5	24980.00	27010.00
6	6	20700.00	24555.00
7	7	22195.00	35925.00
8	8	21195.00	28270.00
9	9	61770.00	67560.00
10	10	68235.00	57880.00
11	11	68915.00	57135.00
12	12	72880.00	69145.00

Compare the average sales between 2022 and 2023 for each month.

```
SELECT
MONTH(sale_date) AS 'Month',
COALESCE(CAST(AVG(CASE WHEN YEAR(sale_date) = 2022 THEN total_sale END) AS
DECIMAL(10, 2)), 0.00) AS 'Avg Sales 2022',
COALESCE(CAST(AVG(CASE WHEN YEAR(sale_date) = 2023 THEN total_sale END) AS
DECIMAL(10, 2)), 0.00) AS 'Avg Sales 2023'
FROM Sales_Analysis
WHERE YEAR(sale_date) IN (2022, 2023)
GROUP BY MONTH(sale_date)
ORDER BY MONTH(sale_date);
```

Results | Messages

	Month	Avg Sales 2022	Avg Sales 2023
1	1	397.11	396.50
2	2	366.14	535.53
3	3	521.38	394.81
4	4	486.53	466.49
5	5	480.38	450.17
6	6	481.40	438.48
7	7	541.34	427.68
8	8	385.36	495.96
9	9	478.84	462.74
10	10	467.36	399.17
11	11	472.02	453.45
12	12	464.20	490.39

Retrieve 'Age group', 'Quantity' and 'Total sales', grouped by 'Age group'.

```
SELECT
CASE
	WHEN Age BETWEEN 18 AND 23 THEN '18-23'
	WHEN Age BETWEEN 24 AND 29 THEN '24-29'
	WHEN Age BETWEEN 30 AND 35 THEN '30-35'
	WHEN Age BETWEEN 36 AND 41 THEN '36-41'
	WHEN Age BETWEEN 42 AND 47 THEN '42-47'
	WHEN Age BETWEEN 48 AND 60 THEN '48-60'
ELSE '60+'
END AS 'Age group', SUM(quantity) AS 'Quantity', SUM(total_sale) AS 'Total Sales'
FROM Sales_Analysis
GROUP BY
CASE
	WHEN Age BETWEEN 18 AND 23 THEN '18-23'
	WHEN Age BETWEEN 24 AND 29 THEN '24-29'
	WHEN Age BETWEEN 30 AND 35 THEN '30-35'
	WHEN Age BETWEEN 36 AND 41 THEN '36-41'
	WHEN Age BETWEEN 42 AND 47 THEN '42-47'
	WHEN Age BETWEEN 48 AND 60 THEN '48-60'
ELSE '60+'
```

```
END
ORDER BY 'Age group';
```

Results | Messages

	Age group	Quantity	Total Sales
1	18-23	655	138330.00
2	24-29	590	107840.00
3	30-35	651	119660.00
4	36-41	549	102955.00
5	42-47	734	131900.00
6	48-60	1366	241095.00
7	60+	473	69940.00

Retrieve the 'Time period' and the number of 'Transactions', grouped by 'Time period'.

```
SELECT
  CASE
    WHEN DATEPART(HOUR, sale_time) < 12 THEN 'Morning'
    WHEN DATEPART(HOUR, sale_time) BETWEEN 12 AND 17 THEN 'Afternoon'
    WHEN DATEPART(HOUR, sale_time) BETWEEN 18 AND 21 THEN 'Evening'
    ELSE 'Night'
  END AS 'Time Period', COUNT(transactions_id) AS 'Transactions'
FROM Sales_Analysis
GROUP BY
CASE
    WHEN DATEPART(HOUR, sale_time) < 12 THEN 'Morning'
    WHEN DATEPART(HOUR, sale_time) BETWEEN 12 AND 17 THEN 'Afternoon'
    WHEN DATEPART(HOUR, sale_time) BETWEEN 18 AND 21 THEN 'Evening'
    ELSE 'Night'
  END
ORDER BY 'Transactions' DESC;
```

Results | Messages

	Time Period	Transactions
1	Evening	855
2	Morning	561
3	Afternoon	377
4	Night	207