

TASK 6

We are using the sales sample dataset.

This dataset contains sales transaction records from various regions, sales representatives, and product categories. It includes details such as sale dates, unit prices, discounts, quantities sold, and customer types. The data is useful for analyzing revenue trends, regional performance, and the impact of discounts on sales.

Here is a simple SQL using `EXTRACT(MONTH FROM Sale_Date)` in BigQuery to extract the month from a `Sale_Date` column:

```
SELECT
  Sale_Date,
  EXTRACT(MONTH FROM Sale_Date) AS Sale_Month
FROM
  `imposing-vista-456007-p4.sales.data`
LIMIT 20;
```

Row	Sale_Date	Sale_Month
1	2023-07-22	7
2	2023-01-26	1
3	2023-01-13	1
4	2023-02-11	2
5	2023-11-24	11
6	2023-12-19	12
7	2023-01-05	1
8	2023-08-07	8
9	2023-11-20	11
10	2023-08-14	8
11	2023-03-22	3
12	2023-04-25	4

13	2023-08-17	8
14	2023-10-24	10
15	2023-06-01	6
16	2023-07-14	7
17	2023-09-24	9
18	2023-11-07	11
19	2023-05-22	5
20	2023-01-11	1

Sale_Date is in DATE or DATETIME format.

If you want to GROUP BY year and month from a Sale_Date, you can do it like this in BigQuery:

```
SELECT
  EXTRACT(YEAR FROM Sale_Date) AS Sale_Year,
  EXTRACT(MONTH FROM Sale_Date) AS Sale_Month,
  COUNT(*) AS Sales_Count
FROM
  `imposing-vista-456007-p4.sales.data`
GROUP BY
  Sale_Year,
  Sale_Month
ORDER BY
  Sale_Year,
  Sale_Month
LIMIT 20;
```

Row	Sale_Year	Sale_Month	Sales_Count
1	2023	1	100
2	2023	2	75
3	2023	3	80
4	2023	4	81
5	2023	5	72
6	2023	6	92

7	2023	7	68
8	2023	8	93
9	2023	9	68
10	2023	10	88
11	2023	11	95
12	2023	12	85
13	2024	1	3

This query:

- Extracts year and month from each sale,
- Groups the sales by year and month,
- Counts how many sales happened in each group,
- Orders results by year then month.

If you want Year-Month together as one field, you can format it like this:

```
SELECT
  FORMAT_DATE('%Y-%m', DATE(Sale_Date)) AS Sale_YearMonth,
  COUNT(*) AS Sales_Count
FROM
  `imposing-vista-456007-p4.sales.data`
GROUP BY
  Sale_YearMonth
ORDER BY
  Sale_YearMonth
LIMIT 20;
```

Row	Sale_YearMonth	Sales_Count
1	2023-01	100
2	2023-02	75
3	2023-03	80
4	2023-04	81
5	2023-05	72
6	2023-06	92

7	2023-07	68
8	2023-08	93
9	2023-09	68
10	2023-10	88
11	2023-11	95
12	2023-12	85
13	2024-01	3

You can SUM(Sales_Amount) for total revenue per month.

Here's the correct BigQuery query:

```
SELECT
  EXTRACT(YEAR FROM Sale_Date) AS Sale_Year,
  EXTRACT(MONTH FROM Sale_Date) AS Sale_Month,
  SUM(Sales_Amount) AS Total_Sales_Amount
FROM
  `imposing-vista-456007-p4.sales.data`
GROUP BY
  Sale_Year,
  Sale_Month
ORDER BY
  Sale_Year,
  Sale_Month
LIMIT 20;
```

Row	Sale_Year	Sale_Month	Total_Sales_Amount
1	2023	1	476092.36000000001
2	2023	2	368919.35999999993
3	2023	3	402638.76999999996
4	2023	4	438992.60999999998
5	2023	5	389078.75999999998
6	2023	6	418458.34000000002
7	2023	7	374242.88

8	2023	8	443171.27999999997
9	2023	9	367837.60000000003
10	2023	10	460378.780000000014
11	2023	11	467482.89999999991
12	2023	12	392643.58
13	2024	1	19328.01

This will give you total Sales_Amount per year-month

Alternatively, if you prefer Year-Month together like 2025-04:

```
SELECT
  FORMAT_DATE('%Y-%m', DATE(Sale_Date)) AS Sale_YearMonth,
  SUM(Sales_Amount) AS Total_Sales_Amount
FROM
  `imposing-vista-456007-p4.sales.data`
GROUP BY
  Sale_YearMonth
ORDER BY
  Sale_YearMonth
LIMIT 20;
```

Row	Sale_YearMonth	Total_Sales_Amount
1	2023-01	476092.36000000001
2	2023-02	368919.35999999993
3	2023-03	402638.76999999996
4	2023-04	438992.609999999987
5	2023-05	389078.759999999983
6	2023-06	418458.34000000002
7	2023-07	374242.88
8	2023-08	443171.27999999997
9	2023-09	367837.60000000003
10	2023-10	460378.780000000014

11	2023-11	467482.89999999991
12	2023-12	392643.58
13	2024-01	19328.01

if you want to calculate sales volume as the number of distinct product (product_id) per year and month, here's the updated BigQuery query:

Year and Month in separate columns

```
SELECT
  EXTRACT(YEAR FROM Sale_Date) AS Sale_Year,
  EXTRACT(MONTH FROM Sale_Date) AS Sale_Month,
  COUNT(DISTINCT product_id) AS Sales_Volume
FROM
  `imposing-vista-456007-p4.sales.data`
GROUP BY
  Sale_Year,
  Sale_Month
ORDER BY
  Sale_Year,
  Sale_Month
LIMIT 20;
```

Row	Sale_Year	Sale_Month	Sales_Volume
1	2023	1	66
2	2023	2	55
3	2023	3	57
4	2023	4	56
5	2023	5	51
6	2023	6	60
7	2023	7	44
8	2023	8	61
9	2023	9	46
10	2023	10	56

11	2023	11	59
12	2023	12	58
13	2024	1	3

Combined Year-Month format (YYYY-MM)

```
SELECT
  FORMAT_DATE('%Y-%m', DATE(Sale_Date)) AS Sale_YearMonth,
  COUNT(DISTINCT product_id) AS Sales_Volume
FROM
  `imposing-vista-456007-p4.sales.data`
GROUP BY
  Sale_YearMonth
ORDER BY
  Sale_YearMonth
LIMIT 20;
```

Row	Sale_YearMonth	Sales_Volume
1	2023-01	66
2	2023-02	55
3	2023-03	57
4	2023-04	56
5	2023-05	51
6	2023-06	60
7	2023-07	44
8	2023-08	61
9	2023-09	46
10	2023-10	56
11	2023-11	59
12	2023-12	58
13	2024-01	3

This will give you the number of unique orders per month.

Here's a single BigQuery query that gives you monthly:

- SUM(Sales_Amount) → Total revenue
- COUNT(DISTINCT product_id) → Sales volume (unique product)

Total Revenue & Volume Per Month

```
SELECT
  FORMAT_DATE('%Y-%m', DATE(Sale_Date)) AS Sale_YearMonth,
  SUM(Sales_Amount) AS Total_Sales_Amount,
  COUNT(DISTINCT product_id) AS Sales_Volume
FROM
  `imposing-vista-456007-p4.sales.data`
GROUP BY
  Sale_YearMonth
ORDER BY
  Sale_YearMonth
LIMIT 20;
```

Row	Sale_YearMonth	Total_Sales_Amount	Sales_Volume
1	2023-01	476092.36000000001	66
2	2023-02	368919.35999999993	55
3	2023-03	402638.76999999996	57
4	2023-04	438992.60999999998	56
5	2023-05	389078.75999999983	51
6	2023-06	418458.34000000002	60
7	2023-07	374242.88	44
8	2023-08	443171.27999999997	61
9	2023-09	367837.60000000003	46
10	2023-10	460378.78000000001	56
11	2023-11	467482.89999999991	59
12	2023-12	392643.58	58
13	2024-01	19328.01	3

Since you have a Quantity column, we can now include more meaningful metrics like:

- Total Revenue: SUM(Sales_Amount)
- Sales Volume: COUNT(DISTINCT product_id)
- Total Units Sold: SUM(Quantity_Sold)
- Total Profit: (Unit_Price - Unit_Cost) * Quantity_Sold
- Average Discount: AVG(Discount)

Full BigQuery Query with All Metrics

```
SELECT
  FORMAT_DATE('%Y-%m', DATE(Sale_Date)) AS Sale_YearMonth,
  SUM(Sales_Amount) AS Total_Sales_Amount,
  COUNT(DISTINCT product_id) AS Sales_Volume,
  SUM(Quantity_Sold) AS Total_Units_Sold,
  SUM((Unit_Price - Unit_Cost) * Quantity_Sold) AS Total_Profit,
  AVG(Discount) AS Average_Discount
FROM
  `imposing-vista-456007-p4.sales.data`
GROUP BY
  Sale_YearMonth
ORDER BY
  Sale_YearMonth
LIMIT 20;
```

Row	Sale_YearMonth	Total_Sales_Amount	Sales_Volume	Total_Units_Sold	Total_Profit	Average_Discount
1	2023-01	476092.3600000001	66	2472	655202.5900000002	0.14379999999999993
2	2023-02	368919.35999999993	55	2064	478053.08000000007	0.15733333333333338
3	2023-03	402638.76999999996	57	2069	528631.81	0.14887499999999995
4	2023-04	438992.60999999987	56	1977	496006.82000000007	0.16765432098765434
5	2023-05	389078.75999999983	51	1968	489472.15	0.15361111111111117
6	2023-06	418458.3400000002	60	2276	605992.20000000007	0.14478260869565215

7	2023-07	374242.88	44	1603	447678.760000000018	0.15367647058823528
8	2023-08	443171.279999999997	61	2336	558417.690000000018	0.14397849462365597
9	2023-09	367837.600000000003	46	1815	472940.519999999984	0.16499999999999995
10	2023-10	460378.780000000014	56	2386	584679.829999999984	0.14965909090909088
11	2023-11	467482.899999999991	59	2129	557812.059999999994	0.15578947368421056
12	2023-12	392643.58	58	2183	590397.549999999993	0.14964705882352936
13	2024-01	19328.01	3	77	22562.010000000002	0.19666666666666671

This will give you a monthly summary with revenue, volume, units sold, profit, and discount insights.

SORT BY

```

SELECT
  FORMAT_DATE('%Y-%m', DATE(Sale_Date)) AS Sale_YearMonth,
  SUM(Sales_Amount) AS Total_Sales_Amount,
  COUNT(DISTINCT product_id) AS Sales_Volume,
  SUM(Quantity_Sold) AS Total_Units_Sold,
  SUM((Unit_Price - Unit_Cost) * Quantity_Sold) AS Total_Profit,
  AVG(Discount) AS Average_Discount
FROM
  `imposing-vista-456007-p4.sales.data`
GROUP BY
  Sale_YearMonth
ORDER BY
  Sale_YearMonth DESC
LIMIT 20;

```

Row	Sale_YearMonth	Total_Sales_Amount	Sales_Volume	Total_Units_Sold	Total_Profit	Average_Discount
1	2024-01	19328.01	3	77	22562.010000000002	0.19666666666666671

2	2023-12	392643.58	58	2183	590397.54999999993	0.14964705882352936
3	2023-11	467482.89999999991	59	2129	557812.05999999994	0.15578947368421056
4	2023-10	460378.78000000014	56	2386	584679.82999999984	0.14965909090909088
5	2023-09	367837.60000000003	46	1815	472940.51999999984	0.16499999999999995
6	2023-08	443171.27999999997	61	2336	558417.69000000018	0.14397849462365597
7	2023-07	374242.88	44	1603	447678.76000000018	0.15367647058823528
8	2023-06	418458.34000000002	60	2276	605992.20000000007	0.14478260869565215
9	2023-05	389078.75999999983	51	1968	489472.15	0.15361111111111117
10	2023-04	438992.60999999987	56	1977	496006.82000000007	0.16765432098765434
11	2023-03	402638.76999999996	57	2069	528631.81	0.14887499999999995
12	2023-02	368919.35999999993	55	2064	478053.08000000007	0.15733333333333338
13	2023-01	476092.36000000001	66	2472	655202.59000000002	0.14379999999999993

If you want to sort by other metrics, like highest revenue or most units sold, here are a few examples:

Sort by Highest Revenue

ORDER BY

Total_Sales_Amount DESC

Sort by Most Units Sold

ORDER BY

Total_Units_Sold DESC

Here's the updated query that includes all key metrics, and sorts the results by Total Revenue (Total_Sales_Amount) descending, then by month ascending for ties:

```
SELECT
  FORMAT_DATE('%Y-%m', DATE(Sale_Date)) AS Sale_YearMonth,
  SUM(Sales_Amount) AS Total_Sales_Amount,
  COUNT(DISTINCT product_id) AS Sales_Volume,
  SUM(Quantity_Sold) AS Total_Units_Sold,
  SUM((Unit_Price - Unit_Cost) * Quantity_Sold) AS Total_Profit,
  AVG(Discount) AS Average_Discount
FROM
  `imposing-vista-456007-p4.sales.data`
GROUP BY
  Sale_YearMonth
ORDER BY
  Total_Sales_Amount DESC,
  Sale_YearMonth ASC
LIMIT 20;
```

Row	Sale_YearMonth	Total_Sales_Amount	Sales_Volume	Total_Units_Sold	Total_Profit	Average_Discount
1	2023-01	476092.3600000001	66	2472	655202.5900000002	0.14379999999999993
2	2023-11	467482.89999999991	59	2129	557812.05999999994	0.15578947368421056
3	2023-10	460378.78000000014	56	2386	584679.82999999984	0.14965909090909088
4	2023-08	443171.27999999997	61	2336	558417.69000000018	0.14397849462365597
5	2023-04	438992.60999999987	56	1977	496006.82000000007	0.16765432098765434
6	2023-06	418458.3400000002	60	2276	605992.20000000007	0.14478260869565215

7	2023-03	402638.76999999996	57	2069	528631.81	0.14887499999999995
8	2023-12	392643.58	58	2183	590397.54999999993	0.14964705882352936
9	2023-05	389078.75999999983	51	1968	489472.15	0.15361111111111117
10	2023-07	374242.88	44	1603	447678.76000000018	0.15367647058823528
11	2023-02	368919.35999999993	55	2064	478053.08000000007	0.15733333333333338
12	2023-09	367837.60000000003	46	1815	472940.51999999984	0.16499999999999995
13	2024-01	19328.01	3	77	22562.010000000002	0.19666666666666671

This query:

- Shows all key metrics per month,
- Sorts primarily by highest revenue (Total_Sales_Amount DESC),
- Breaks ties by sorting chronologically (Sale_YearMonth ASC).

Revenue based on Sales Amount:

If Sales_Amount already reflects revenue after discount, just sum it:

```
SELECT SUM(Sales_Amount) AS Total_Revenue
FROM imposing-vista-456007-p4.sales.data;
```

Row	Total_Revenue
1	5019265.2299999986

Revenue calculated from Unit Price and Discount:

If Unit_Price and Discount are given, and the discount is a percentage:

```
SELECT SUM(Unit_Price * (1 - Discount)) AS Total_Revenue
FROM imposing-vista-456007-p4.sales.data;
```

Row Total_Revenue

1	2314758.4560999987
---	--------------------

Revenue calculated from Unit Price, Discount, and quantity:

If you also have a Quantity_Sold column use:

```
SELECT SUM(Unit_Price * Quantity_Sold * (1 - Discount)) AS Total_Revenue
FROM imposing-vista-456007-p4.sales.data;
```

Row	Total_Revenue
1	59686173.283700034

Revenue by Region

```
SELECT Region,
       SUM(Unit_Price * Quantity_Sold * (1 - Discount)) AS Total_Revenue
FROM imposing-vista-456007-p4.sales.data
GROUP BY Region;
```

Row Region Total_Revenue

1	East	15202282.224599998
2	North	15570342.181999987
3	South	13894660.279499998

Revenue by Sales Rep

```
SELECT Sales_Rep,
       SUM(Unit_Price * Quantity_Sold * (1 - Discount)) AS Total_Revenue
FROM imposing-vista-456007-p4.sales.data
GROUP BY Sales_Rep;
```

Row Sales_Rep Total_Revenue

1	Alice	11476645.5633
2	Bob	11768703.708700005
3	Charlie	9827690.8507000022

Monthly Revenue

If Sale_Date is a string or date, extract the month:

```
SELECT FORMAT_DATE('%Y-%m', DATE(Sale_Date)) AS Sale_Month,
```

```
SUM(Unit_Price * Quantity_Sold * (1 - Discount)) AS Total_Revenue
FROM imposing-vista-456007-p4.sales.data
GROUP BY Sale_Month
ORDER BY Sale_Month;
```

Row	Sale_Month	Total_Revenue
1	2023-01	6375602.7685
2	2023-02	4662249.4973999988
3	2023-03	5111669.9625999993
4	2023-04	4518199.637000001
5	2023-05	4878261.7867000019
6	2023-06	5296111.9989000009
7	2023-07	4003580.3107999992
8	2023-08	5724989.7415000005
9	2023-09	3835193.0910000009
10	2023-10	5678169.618999999
11	2023-11	4157584.6568999989
12	2023-12	5398212.0220000008
13	2024-01	46348.1914

To limit results for a specific time period in BigQuery, you add a WHERE clause that filters on the Sale_Date column.

Assuming Sale_Date is a string in 'YYYY-MM-DD' format (which it appears to be), convert it to DATE for filtering.

Revenue by Region for January to March 2023

```
SELECT Region,
SUM(Unit_Price * Quantity_Sold * (1 - Discount)) AS Total_Revenue
FROM imposing-vista-456007-p4.sales.data
WHERE DATE(Sale_Date) BETWEEN '2023-01-01' AND '2023-03-31'
GROUP BY Region
LIMIT 20;
```

Row	Region	Total_Revenue
1	East	5333339.9491999978
2	North	3729817.1374000004
3	South	3355137.2315000012
4	West	3731227.9103999995

Monthly Revenue for Q2 2023 (April–June)

```
SELECT FORMAT_DATE('%Y-%m', DATE(Sale_Date)) AS Sale_Month,
SUM(Unit_Price * Quantity_Sold * (1 - Discount)) AS Total_Revenue
FROM imposing-vista-456007-p4.sales.data
```

```

WHERE DATE(Sale_Date) BETWEEN '2023-04-01' AND '2023-06-30'
GROUP BY Sale_Month
ORDER BY Sale_Month
LIMIT 20;

```

Row	Sale_Month	Total_Revenue
1	2023-04	4518199.637000001
2	2023-05	4878261.7867000019
3	2023-06	5296111.9989000009

Compare Revenue by Region for Q1 and Q2 of 2023

```

SELECT Region,
       CONCAT('Q', CAST(EXTRACT(QUARTER FROM DATE(Sale_Date)) AS STRING)) AS
Quarter,
       SUM(Unit_Price * Quantity_Sold * (1 - Discount)) AS Total_Revenue
FROM imposing-vista-456007-p4.sales.data
WHERE EXTRACT(YEAR FROM DATE(Sale_Date)) = 2023
      AND EXTRACT(QUARTER FROM DATE(Sale_Date)) IN (1, 2)
GROUP BY Region, Quarter
ORDER BY Region, Quarter
LIMIT 20;

```

Row	Region	Quarter	Total_Revenue
1	East	Q1	5333339.9491999978
2	East	Q2	3216331.0129
3	North	Q1	3729817.1374000004
4	North	Q2	4219541.5009
5	South	Q1	3355137.2315000012
6	South	Q2	3931833.7466
7	West	Q1	3731227.9103999995
8	West	Q2	3324867.1621999997

This query gives you total revenue per region for Q1 and Q2 of 2023.