

ONLINE SALES ANALYSIS USING SQL

Aim: Analyze monthly revenue and order volume.

1. Total units sold in January (testing it for 1 month)

```
10 SET date = STR_TO_DATE(date, '%d-%m-%Y');
11 #total units sold for one month
12 • SELECT SUM(units_sold) AS TotalSalesJan2024
13 FROM onlinesales
14 WHERE MONTH(date) = 1; # 1 is for January
15 • SELECT
16 YEAR(date) AS SalesYear,
```

Result Grid | Filter Rows: | Export: | Wrap

TotalSalesJan2024
68

2. Calculating units sold for all months.

```
15 • SELECT
16 YEAR(date) AS SalesYear,
17 MONTH(date) AS SalesMonthNumber,
18 SUM(units_sold) AS TotalMonthlyUnits
19 FROM onlinesales
20 GROUP BY SalesYear, SalesMonthNumber
21 ORDER BY TotalMonthlyUnits DESC
```

Result Grid | Filter Rows: | Export: |

SalesYear	SalesMonthNumber	TotalMonthlyUnits
2024	3	82
2024	2	77
2024	1	68
2024	4	65
2024	6	61

```
15 • SELECT
16 YEAR(date) AS SalesYear,
17 MONTH(date) AS SalesMonthNumber,
18 SUM(units_sold) AS TotalMonthlyUnits
19 FROM onlinesales
20 GROUP BY SalesYear, SalesMonthNumber
21 ORDER BY TotalMonthlyUnits DESC
```

Result Grid | Filter Rows: | Export: |

SalesYear	SalesMonthNumber	TotalMonthlyUnits
2024	4	65
2024	6	61
2024	5	60
2024	7	53
2024	8	52

The first quarter has higher sales, meanwhile sales tend to drop in the second quarter.

Highest sales: March,

Lowest sales: August

3. Total revenue by month

```
23 • SELECT
24 YEAR(date) AS SalesYear,
25 MONTH(date) AS SalesMonthNumber,
26 ROUND(SUM(total_revenue),2) AS TotalMonthlyRevenue
27 FROM onlinesales
28 GROUP BY SalesYear, SalesMonthNumber
29 ORDER BY TotalMonthlyRevenue DESC;
```

	SalesYear	SalesMonthNumber	TotalMonthlyRevenue
▶	2024	1	14548.32
	2024	3	12849.24
	2024	4	12451.69
	2024	2	10803.37
	2024	5	8455.49

	SalesYear	SalesMonthNumber	TotalMonthlyRevenue
	2024	2	10803.37
	2024	5	8455.49
	2024	6	7384.55
	2024	8	7278.11
	2024	7	6797.08

Again, the 1st quarter generates higher revenue and the 2nd quarter seems to generate way less.

Highest revenue generating: January

Lowest revenue generating: July

```
30 • SELECT SUM(total_revenue)
31 from ONLINESALES
32 WHERE MONTH(date) < 5;
33 • SELECT SUM(total_revenue)
34 from ONLINESALES
35 WHERE MONTH(date) > 4;
36
```

	SUM(total_revenue)
▶	50652.62000000003

```
30 • SELECT SUM(total_revenue)
31 from ONLINESALES
32 WHERE MONTH(date) < 5;
33 • SELECT SUM(total_revenue)
34 from ONLINESALES
35 WHERE MONTH(date) > 4;
36
```

	SUM(total_revenue)
▶	29915.23000000001

They have a difference of around \$20,000. Company should look into this disadvantageous drop.

4. Most purchased category for the 2 quarters

```
39 • SELECT category, SUM(units_sold) AS total_units
40 FROM onlinesales
41 WHERE MONTH(date) < 5
42 GROUP BY category
43 ORDER BY total_units DESC
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	category	total_units
▶	Clothing	79
	Books	62
	Sports	56
	Electronics	41
	Home Appliances	30
	Beauty Products	24

```
44 • SELECT category, SUM(units_sold) AS total_units
45 FROM onlinesales
46 WHERE MONTH(date) > 4
47 GROUP BY category
48 ORDER BY total_units DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	category	total_units
▶	Clothing	66
	Books	52
	Sports	32
	Home Appliances	29
	Electronics	25
	Beauty Products	22

Clothing was the most popular category for both the quarters. We see a reduction of every category in the 2nd quarter.

5. Most sold product in the quarters

```
50 • SELECT product, SUM(units_sold) AS total_units
51 FROM onlinesales
52 WHERE MONTH(date) <5
53 GROUP BY product
54 ORDER BY total_units DESC
55 LIMIT 5;
```

product	total_units
Hanes ComfortSoft T-Shirt	10
Gap Essential Crewneck T-Shirt	6
Nike Air Force 1	6
Yeti Rambler Tumbler	6
Spalding NBA Street Basketball	6

```
56 • SELECT product, SUM(units_sold) AS total_units
57 FROM onlinesales
58 WHERE MONTH(date) >4
59 GROUP BY product
60 ORDER BY total_units DESC
61 LIMIT 5;
```

product	total_units
Forever 21 Graphic Tee	5
Adidas 3-Stripes Shorts	5
The Girl with the Dragon Tattoo by Stieg Larsson	5
Adidas Originals Trefoil Hoodie	4
Amazon Echo Dot (4th Gen)	4