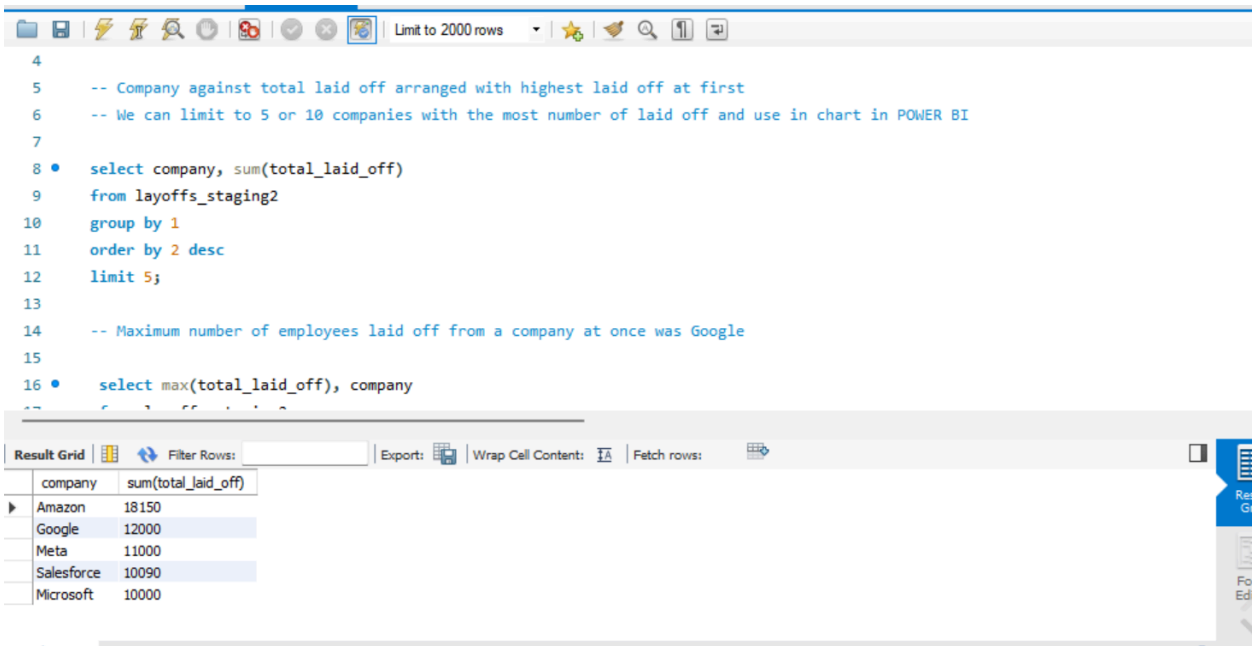
Exploratory Data Analysis - SQL

We use the data that we cleaned from layoffs_staging2 to further analyze and draw meaningful insights from it using SQL queries.

1. Top companies that laid off the highest number of its staff.



```
4
5     -- Company against total laid off arranged with highest laid off at first
6     -- We can limit to 5 or 10 companies with the most number of laid off and use in chart in POWER BI
7
8     select company, sum(total_laid_off)
9     from layoffs_staging2
10    group by 1
11    order by 2 desc
12    limit 5;
13
14    -- Maximum number of employees laid off from a company at once was Google
15
16    select max(total_laid_off), company
```

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

company	sum(total_laid_off)
Amazon	18150
Google	12000
Meta	11000
Salesforce	10090
Microsoft	10000

2. Maximum number of employees laid off at once

```
10    group by 1
11    order by 2 desc;
12
13    -- Maximum number of employees laid off from a company at once was Google
14
15    select max(total_laid_off), company
16    from layoffs_staging2
17    group by company
18    order by 1 desc
19    limit 1;
20
21    -- Maximum percentage of people laid off from a company
22    -- Percentage of 1 represents, 100% of its employees were laid off
```

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

max(total_laid_off)	company
12000	Google

3. How many companies laid off 100% of its employees

116 companies lost all of their employees, mostly start-ups who went out of business during the pandemic.

```
22    -- Percentage of 1 represents, 100% of its employees were laid off
23
24    select max(percentage_laid_off)
25    from layoffs_staging2;
26
27    -- 116 companies laid off all of it's staff
28
29    select count(*)
30    from layoffs_staging2 where percentage_laid_off = 1;
31
32    -- Date range of this data to find out between what years these layoffs have taken place
33    -- It's between March 2020 and March 2023
34    -- Pandemic behaviour might show in data
```

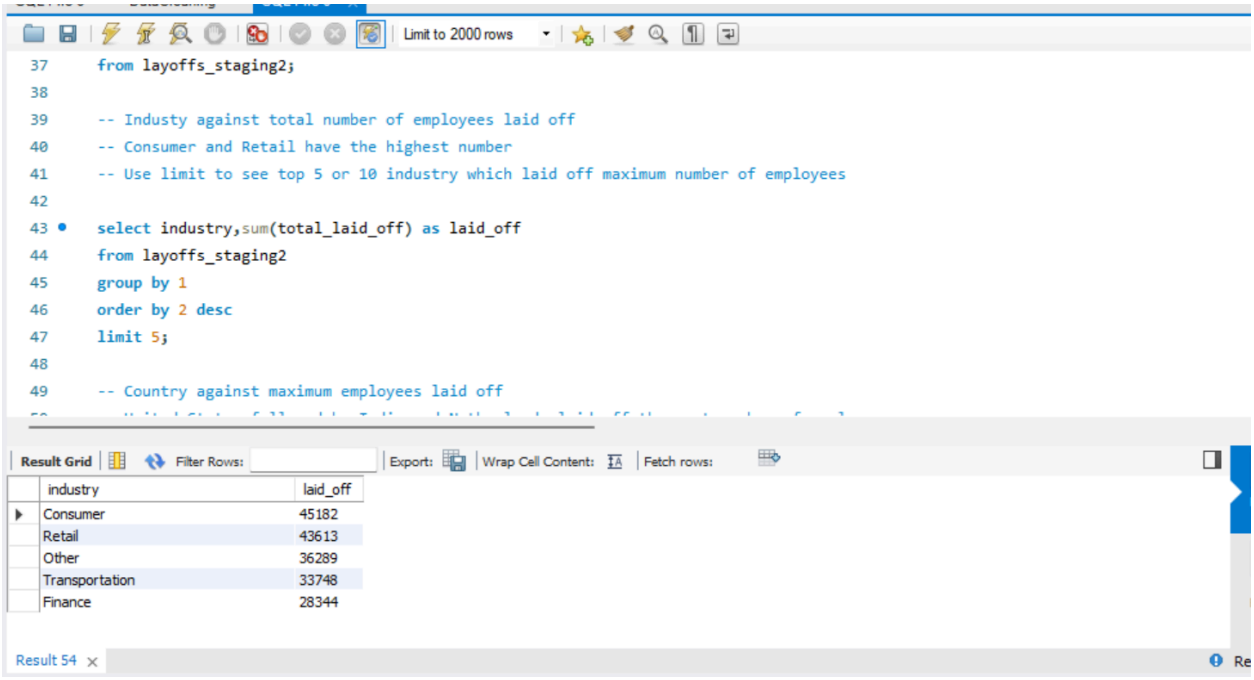
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

count(*)
116

Result 53

4. Industry which saw the highest layoffs

   ‘Consumer’ and ‘Retail’ industries suffered the highest blow of job cuts.



```
37  from layoffs_staging2;
38
39  -- Industy against total number of employees laid off
40  -- Consumer and Retail have the highest number
41  -- Use limit to see top 5 or 10 industry which laid off maximum number of employees
42
43  select industry,sum(total_laid_off) as laid_off
44  from layoffs_staging2
45  group by 1
46  order by 2 desc
47  limit 5;
48
49  -- Country against maximum employees laid off
```

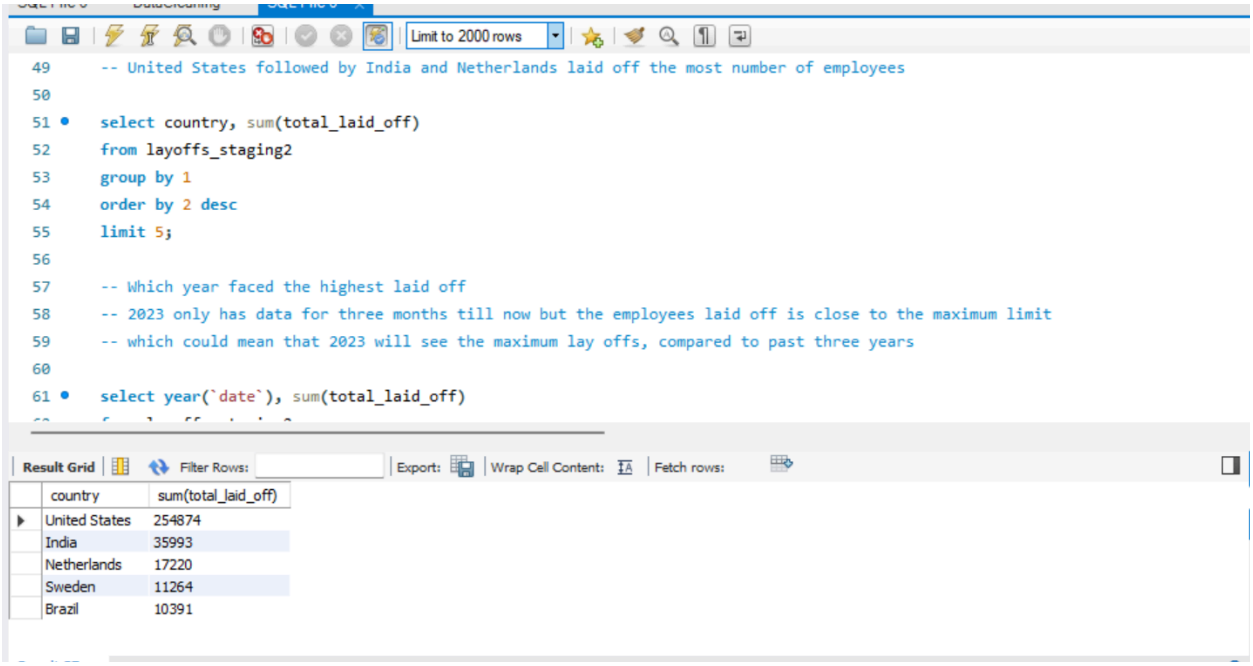
Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

industry	laid_off
Consumer	45182
Retail	43613
Other	36289
Transportation	33748
Finance	28344

Result 54

5. Top countries that made the highest jobs cuts

   US and India being the top contenders, followed by Netherlands



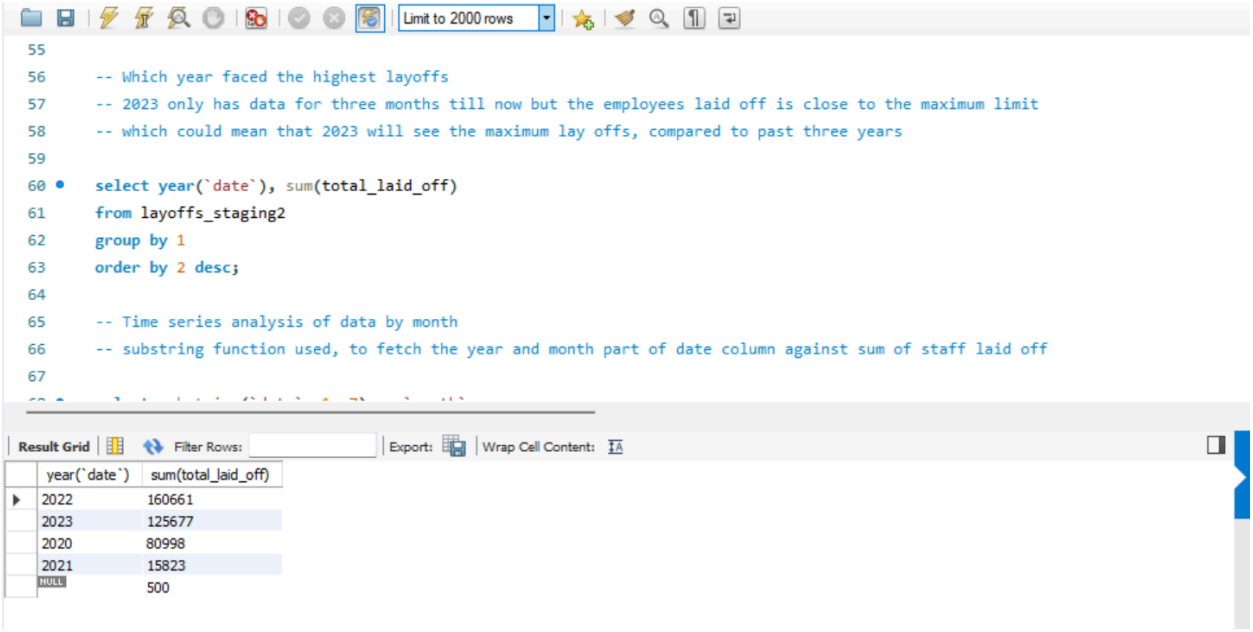
```
Limit to 2000 rows

49  -- United States followed by India and Netherlands laid off the most number of employees
50
51  select country, sum(total_laid_off)
52  from layoffs_staging2
53  group by 1
54  order by 2 desc
55  limit 5;
56
57  -- Which year faced the highest laid off
58  -- 2023 only has data for three months till now but the employees laid off is close to the maximum limit
59  -- which could mean that 2023 will see the maximum lay offs, compared to past three years
60
61  select year(`date`), sum(total_laid_off)
```

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

country	sum(total_laid_off)
United States	254874
India	35993
Netherlands	17220
Sweden	11264
Brazil	10391

6. Comparison of total number of employees laid off in 4 years (2020 - 2023)

2023 will probably see the highest layoffs as it has data till March and it's already close to the highest value which was in 2022.

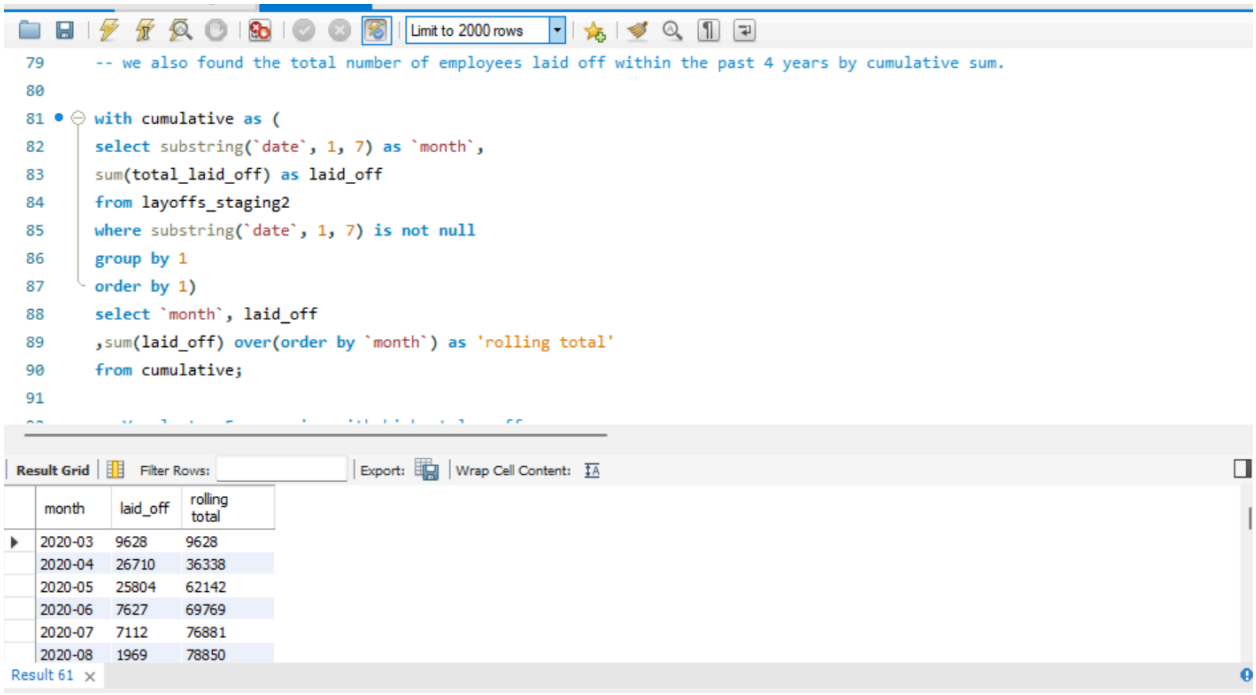


```
Limit to 2000 rows

55
56  -- Which year faced the highest layoffs
57  -- 2023 only has data for three months till now but the employees laid off is close to the maximum limit
58  -- which could mean that 2023 will see the maximum lay offs, compared to past three years
59
60  select year(`date`), sum(total_laid_off)
61  from layoffs_staging2
62  group by 1
63  order by 2 desc;
64
65  -- Time series analysis of data by month
66  -- substring function used, to fetch the year and month part of date column against sum of staff laid off
67
```

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

year(`date`)	sum(total_laid_off)
2022	160661
2023	125677
2020	80998
2021	15823
NULL	500

7. Time series analysis of layoffs against each month



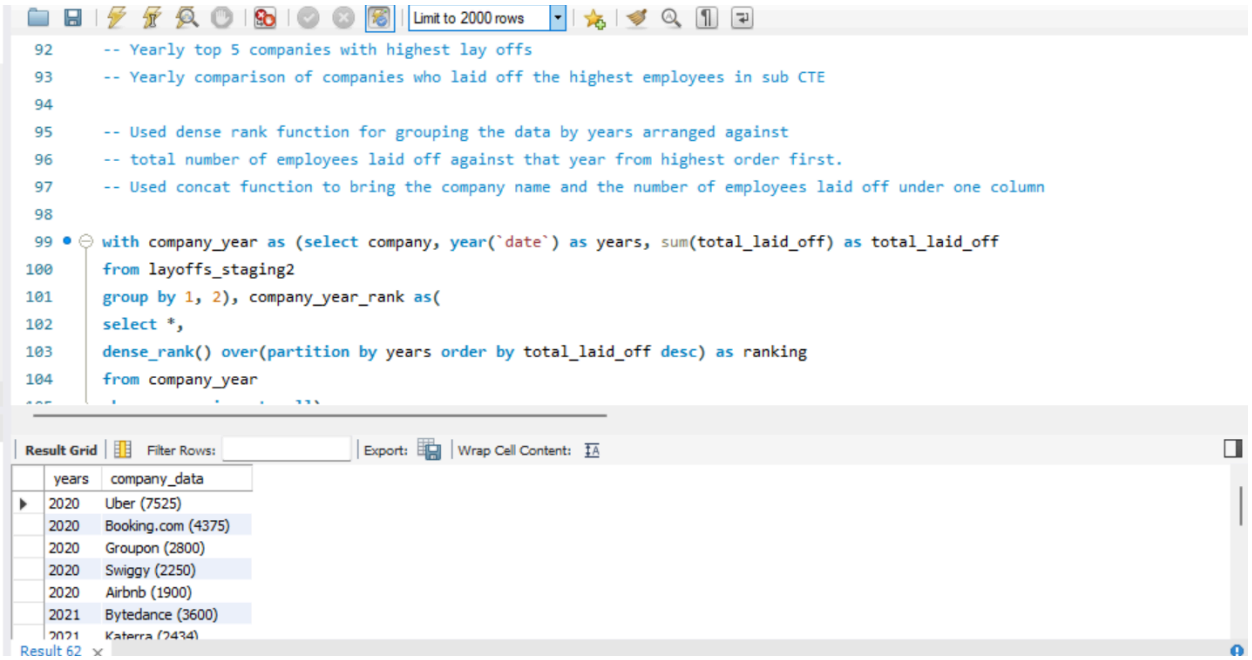
```
79    -- we also found the total number of employees laid off within the past 4 years by cumulative sum.
80
81    with cumulative as (
82    select substring(`date`, 1, 7) as `month`,
83    sum(total_laid_off) as laid_off
84    from layoffs_staging2
85    where substring(`date`, 1, 7) is not null
86    group by 1
87    order by 1)
88    select `month`, laid_off
89    ,sum(laid_off) over(order by `month`) as 'rolling total'
90    from cumulative;
91
```

Result Grid

month	laid_off	rolling total
2020-03	9628	9628
2020-04	26710	36338
2020-05	25804	62142
2020-06	7627	69769
2020-07	7112	76881
2020-08	1969	78850

Result 61

8. Top companies with highest layoffs of each year



```
92    -- Yearly top 5 companies with highest lay offs
93    -- Yearly comparison of companies who laid off the highest employees in sub CTE
94
95    -- Used dense rank function for grouping the data by years arranged against
96    -- total number of employees laid off against that year from highest order first.
97    -- Used concat function to bring the company name and the number of employees laid off under one column
98
99    with company_year as (select company, year(`date`) as years, sum(total_laid_off) as total_laid_off
100   from layoffs_staging2
101   group by 1, 2), company_year_rank as(
102   select *,
103   dense_rank() over(partition by years order by total_laid_off desc) as ranking
104   from company_year
```

Result Grid

years	company_data
2020	Uber (7525)
2020	Booking.com (4375)
2020	Groupon (2800)
2020	Swiggy (2250)
2020	Airbnb (1900)
2021	Bytedance (3600)
2021	Katerra (2434)

Result 62