

SQL Project: World Layoffs











Exploratory Data Analysis

The project leverages MySQL to dive into the data on worldwide employee layoffs across different industries.






The project is aimed at:

Identifying trends and patterns in the dataset.

Presenting data-driven insights.





Limit to 1000 rows





```
5      -- Finding maximum number of layoffs that occurred at an within the timeframe covered in the dataset.
6 •    SELECT MAX(total_laid_off) FROM layoffs_staging2;
7
8      -- Finding the number of employees laid-off when lay off % is 100%
9 •    SELECT MAX(total_laid_off)
10     FROM layoffs_staging2
11     WHERE percentage_laid_off = 1;
12
```

Result Grid



 Filter Rows:

Export: 

Wrap Cell Content: 

	MAX(total_laid_off)
▶	2434



```
18 -- Total number of employees laid off at each company
19 • SELECT company, SUM(total_laid_off)
20 FROM layoffs_staging2
21 GROUP BY company
22 ORDER BY 2 DESC;
23
```

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
	Philips	10000
	Ericsson	8500
	Uber	7585
	Dell	6650
	Booking.com	4601
	Cisco	4100
	Peloton	4084

Limit to 1000 rows

24

-- Finding layoffs by date range

25

• SELECT MIN(`date`), MAX(`date`)

26

FROM layoffs_staging2;

27

28

-- Finding layoffs by industry

29

• SELECT industry, SUM(total_laid_off)

30

FROM layoffs_staging2

31

GROUP BY industry

32

ORDER BY 2 DESC;

33

Result Grid

Filter Rows:

Export:

Wrap Cell Content:
















	industry	SUM(total_laid_off)
▶	Consumer	45182
	Retail	43613
	Other	36209
	Transportation	33548
	Finance	28344
	Healthcare	25894
	Food	22855
	Real Estate	17565

Toolbar icons: Folder, Save, Undo, Redo, Find, Stop, Run, Limit to 1000 rows, Star, Erase, Zoom, Split, Refresh.





```
34  -- Layoffs by year
35  •  SELECT year(`date`), SUM(total_laid_off)
36     FROM layoffs_staging2
37     GROUP BY year(`date`)
38     ORDER BY 1 DESC;
```

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

	year(`date`)	SUM(total_laid_off)
▶	2023	125677
	2022	160322
	2021	15823
	2020	80998
	NULL	500

Limit to 1000 rows

```
40  -- Layoffs by country
41  •  SELECT country, SUM(total_laid_off)
42     FROM layoffs_staging2
43     GROUP BY country
44     ORDER BY 2 DESC;
```

Result GridFilter Rows:Export:Wrap Cell Content:

	country	SUM(total_laid_off)
▶	United States	256420
	India	35793
	Netherlands	17220
	Sweden	11264
	Brazil	10391
	Germany	8701
	United Kingdom	6398
	Canada	6319
	Singapore	5995
	China	5905
	Israel	3638
	Indonesia	3521
	Australia	2324
	Nigeria	1882

-- Per month layoffs, by extracting the substring from date column by slicing

• SELECT substring(`date`,1,7) AS `Month`, SUM(total_laid_off)

FROM layoffs_staging2

WHERE substring(`date`,1,7) IS NOT NULL

GROUP BY `Month`

ORDER BY 1 ASC;






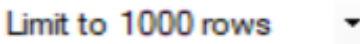










Result Grid

Filter Rows:






Export:

Wrap Cell Content:











	Month	SUM(total_laid_off)
▶	2020-03	9628
	2020-04	26710
	2020-05	25804
	2020-06	7627
	2020-07	7112
	2020-08	1969
	2020-09	609
	2020-10	450
	2020-11	237
	2020-12	852
	2021-01	6813
	2021-02	868
	2021-03	47








```
72 -- Company layoffs by particular year
73 • SELECT company, year(`date`), SUM(total_laid_off)
74 FROM layoffs_staging2
75 GROUP BY company, year(`date`)
76 ORDER BY 3 DESC;
77
```

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

	company	year(`date`)	SUM(total_laid_off)
	Google	2023	12000
	Meta	2022	11000
	Amazon	2022	10150
	Microsoft	2023	10000
	Ericsson	2023	8500
	Amazon	2023	8000
	Salesforce	2023	8000
	Uber	2020	7525
	Dell	2023	6650
	Philips	2023	6000
	Booking.com	2020	4375
	Cisco	2022	4100
	Peloton	2022	4084





Limit to 1000 rows




```
59  -- Layoffs with rolling total i.e. the accumulation of layoff numbers over the months
60  •  WITH Rolling_Total AS
61      (
62      SELECT substring(`date`,1,7) AS `Month`, SUM(total_laid_off) AS total_off
63      FROM layoffs_staging2
64      WHERE substring(`date`,1,7) IS NOT NULL
65      GROUP BY `Month`
66      ORDER BY 1 ASC
67      )
68      SELECT `Month`, total_off,
69      SUM(total_off) OVER(ORDER BY `Month`) AS rolling_total
70      FROM Rolling_Total;
```

Result Grid

 Filter Rows:

Export: 

Wrap Cell Content: 

	Month	total_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
	2020-09	609	79459
	2020-10	450	79909
	2020-11	237	80146
	2020-12	852	80998

79 -- Ranking top 5 companies by layoffs with year
 80 • WITH Company_Year (company, years, total_laid_off) AS
 81 (
 82 SELECT company, year(`date`), SUM(total_laid_off)
 83 FROM layoffs_staging2
 84 GROUP BY company, year(`date`)
 85), Company_Year_Rank AS
 86 (
 87 SELECT *,
 88 DENSE_RANK() OVER (PARTITION BY years ORDER BY total_laid_off DESC) AS Ranking
 89 FROM Company_Year
 90 WHERE years IS NOT NULL
 91)
 92 SELECT *
 93 FROM Company_Year_Rank
 94 WHERE Ranking <=5;

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

	company	years	total_laid_off	Ranking
►	Uber	2020	7525	1
	Booking.com	2020	4375	2
	Groupon	2020	2800	3
	Swiggy	2020	2250	4
	Airbnb	2020	1900	5
	Bytedance	2021	3600	1