

An Exploratory Dive into Global Layoffs

🔍 Project Summary:

Using the layoffs dataset, this exploratory data analysis (EDA) project investigates trends in global layoffs. The approach focuses on open-ended exploration, examining metrics like total and percentage layoffs, outliers, funding levels, and company lifecycle stages.

Let's understand the query execution

1. Check the dataset after data cleaning.

```
SELECT*FROM layoffs_staging2;
```

company	location	industry	total_laid_off	percentage_laid_off	date	stage	country	funds_raised_millions	row_num
Included Health	SF Bay Area	Healthcare	0	0.06	2022-07-25	Series E	United States	272	1
&Open	Dublin	Marketing	9	0.09	2022-11-17	Series A	Ireland	35	1
#Paid	Toronto	Marketing	19	0.17	2023-01-27	Series B	Canada	21	1
100 Thieves	Los Angeles	Consumer	12	0	2022-07-13	Series C	United States	120	1
10X Genomics	SF Bay Area	Los Angeles	100	0.08	2022-08-04	Post-IPO	United States	242	1
1stDibs	New York City	Retail	70	0.17	2020-04-07	Series D	United States	253	1

```
SELECT MAX(total_laid_off)
FROM layoffs_staging2;
```

MAX(total_laid_off)
99

2. Looking at Percentage to see how big these layoffs were.

```
SELECT MAX(percentage_laid_off),
MIN(percentage_laid_off)
FROM layoffs_staging2
WHERE percentage_laid_off IS NOT NULL;
```

MAX(percentage_laid_off)	MIN(percentage_laid_off)
1	0

```
SELECT *
FROM layoffs_staging2
WHERE percentage_laid_off = 1
ORDER BY funds_raised_millions DESC;
```

company	location	industry	total_laid_off	percentage_laid_off	date	stage	country	funds_raised_millions	row_num
Britishvolt	London	Transportation	206	1	2023-01-17	Unknown	United Kingdom	2400	1
Quibi	Los Angeles	Media	0	1	2020-10-21	Private Equity	United States	1800	1
Deliveroo Australia	Melbourne	Food	120	1	2022-11-15	Post-IPO	Australia	1700	1
Katerra	SF Bay Area	Construction	2434	1	2021-06-01	Unknown	United States	1600	1
BlockFi	New York City	Crypto	0	1	2022-11-28	Series E	United States	1000	1
Aura Financial	SF Bay Area	Finance	0	1	2021-01-11	Unknown	United States	584	1
Openpay	Melbourne	Finance	83	1	2023-02-07	Post-IPO	Australia	299	1

3. Companies with the biggest single Layoff.

```
SELECT company, total_laid_off  
FROM layoffs_staging  
ORDER BY 2 DESC  
LIMIT 5;
```

	company	total_laid_off
▶	Google	12000
	Meta	11000
	Amazon	10000
	Microsoft	10000
	Ericsson	8500

4. Companies with the most Total Layoffs.

```
SELECT company, SUM(total_laid_off)  
FROM layoffs_staging2  
GROUP BY company  
ORDER BY 2 DESC  
LIMIT 10;
```

	company	SUM(total_laid_off)
▶	Amazon	18150
	Google	12000
	Meta	11000
	Salesforce	10090
	Philips	10000
	Microsoft	10000
	Ericsson	8500
	Other	7585

5. By Location

```
SELECT location, SUM(total_laid_off)  
FROM layoffs_staging2  
GROUP BY location  
ORDER BY 2 DESC  
LIMIT 10;
```

	location	SUM(total_laid_off)
▶	SF Bay Area	125631
	Seattle	34743
	New York City	29364
	Bengaluru	21787
	Amsterdam	17140
	Stockholm	11217
	Boston	10785

6. By country

```
SELECT country, SUM(total_laid_off)
```

```
FROM layoffs_staging2
GROUP BY country
ORDER BY 2 DESC;
```

country	SUM(total_laid_off)
United States	35993
India	17220
Netherlands	11264
Brazil	10391
Germany	8701
United Kingdom	6398
Canada	6310

7. By Year

```
SELECT YEAR(date), SUM(total_laid_off)
FROM layoffs_staging2
GROUP BY YEAR(date)
ORDER BY 1 ASC;
```

YEAR(date)	SUM(total_laid_off)
NULL	500
2020	80998
2021	15823
2022	160661
2023	125677

8. By Industry

```
SELECT industry, SUM(total_laid_off)
FROM layoffs_staging2
GROUP BY industry
ORDER BY 2 DESC;
```

industry	SUM(total_laid_off)
Consumer	44782
Retail	43613
Other	36289
Transportation	31248
Finance	28344
Healthcare	25953
Food	22855
Real Estate	17565

9. By Stages

```
SELECT stage, SUM(total_laid_off)
FROM layoffs_staging2
GROUP BY stage
ORDER BY 2 DESC;
```

stage	SUM(total_laid_off)
Post-IPO	204132
Unknown	40716
Acquired	27576
Series C	20017
Series D	19225
Series B	15311
Series E	12697
Series F	0022

10. Earlier we looked at Companies with the most Layoffs. Now let's look at that per year.

```
WITH Company_Year AS
(
    SELECT company, YEAR(date) AS years,
    SUM(total_laid_off) AS total_laid_off
    FROM layoffs_staging2
    GROUP BY company, YEAR(date)
)
, Company_Year_Rank AS (
    SELECT company, years, total_laid_off, DENSE_RANK()
    OVER (PARTITION BY years ORDER BY total_laid_off
DESC) AS ranking
    FROM Company_Year
)
SELECT company, years, total_laid_off, ranking
FROM Company_Year_Rank
WHERE ranking <= 3
AND years IS NOT NULL
ORDER BY years ASC, total_laid_off DESC;
```

company	years	total_laid_off	ranking
Uber	2020	7525	1
Booking.com	2020	4375	2
Groupon	2020	2800	3
Bytedance	2021	3600	1
Katerra	2021	2434	2
Zillow	2021	2000	3
Meta	2022	11000	1
Amazon	2022	10150	2

11. Rolling Total of Layoffs Per Month

```
SELECT SUBSTRING(date,1,7) as dates,
SUM(total_laid_off) AS total_laid_off
FROM layoffs_staging2
GROUP BY dates
```

```
ORDER BY dates ASC;
```

	dates	total_laid_off
▶	NULL	500
	2020-03	9628
	2020-04	26710
	2020-05	25804
	2020-06	7627
	2020-07	7112
	2020-08	1969
	2020-09	600

Now use it in a CTE so we can query off of it

```
WITH DATE_CTE AS
(
SELECT SUBSTRING(date,1,7) as dates, SUM(total_laid_off)
AS total_laid_off
FROM layoffs_staging2
GROUP BY dates
ORDER BY dates ASC
)
SELECT dates, SUM(total_laid_off) OVER (ORDER BY dates
ASC) as rolling_total_layoffs
FROM DATE_CTE
ORDER BY dates ASC;
```

	dates	rolling_total_layoffs
▶	NULL	500
	2020-03	10128
	2020-04	36838
	2020-05	62642
	2020-06	70269
	2020-07	77381
	2020-08	79350
	2020-09	70050