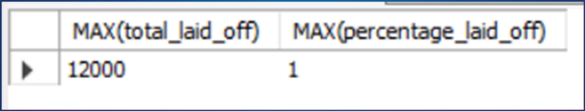
EXPLORATORY DATA ANALYSIS USING SQL

Zaeem Farooq

INTRODUCTION

Following SQL queries perform an exploratory data analysis (EDA) on layoff's cleaned data, uncovering insights like maximum layoffs, trends over time, and company-specific impacts.The analysis reveals patterns across industries, countries, and companies, providing a clear view of the layoff landscape.

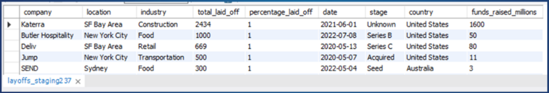
```sql
SELECT *
FROM layoffs_staging2;
SELECT MAX(total_laid_off),
MAX(percentage_laid_off)
FROM layoffs_staging2;
```



	MAX(total_laid_off)	MAX(percentage_laid_off)
▶	12000	1

This query retrieves all data from the `layoffs_staging2` table and identifies the maximum total layoffs and the highest percentage of layoffs in the dataset.

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

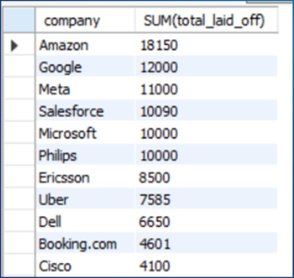


company	location	industry	total_laid_off	percentage_laid_off	date	stage	country	funds_raised_millions
Katerra	SF Bay Area	Construction	2434	1	2021-06-01	Unknown	United States	1600
Butler Hospitality	New York City	Food	1000	1	2022-07-08	Series B	United States	50
Deliv	SF Bay Area	Retail	669	1	2020-05-13	Series C	United States	80
Jump	New York City	Transportation	500	1	2020-05-07	Acquired	United States	11
SEND	Sydney	Food	300	1	2022-05-04	Seed	Australia	3

layoffs_staging2 37 ×

This query retrieves all rows from the `layoffs_staging2` table where 1 of the workforce was laid off, ordering the results by the total number of layoffs in descending order.

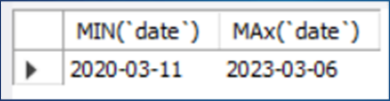
```sql
SELECT company,SUM(total_laid_off)
FROM layoffs_staging2
GROUP BY company
ORDER BY SUM(total_laid_off) DESC;
```



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

This query calculates the total number of layoffs per company from the `layoffs_staging2` table, groups the data by company, and orders the results by total layoffs in descending order.

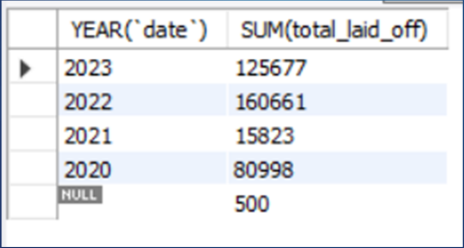
```
SELECT MIN(`date`),MAx(`date`)
FROM layoffs_staging2;
```



	MIN(`date`)	MAx(`date`)
▶	2020-03-11	2023-03-06

This query retrieves the earliest (`MIN`) and latest (`MAX`) dates from the `layoffs_staging2` table.

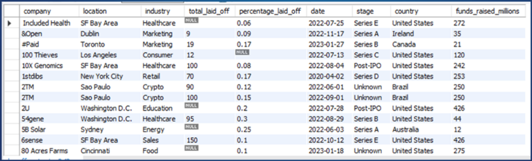
```
SELECT YEAR(`date`),SUM(total_laid_off)
FROM layoffs_staging2
GROUP BY YEAR(`date`)
ORDER BY 1 DESC;
```



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

This query calculates the total number of layoffs for each year in the `layoffs_staging2` table, grouping the data by year and ordering the results in descending order by year.

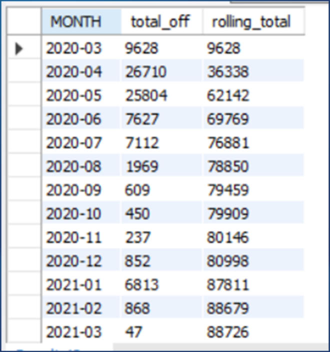
```sql
SELECT * FROM layoffs_staging2;
SELECT SUBSTRING(`date`,1,7) AS `MONTH`,
SUM(total_laid_off)
FROM layoffs_staging2
WHERE SUBSTRING(`date`,1,7) IS NOT NULL
GROUP BY `MONTH`
ORDER BY 1 ASC;
```



company	location	industry	total_laid_off	percentage_laid_off	date	stage	country	funds_raised_millions
Included Health	SF Bay Area	Healthcare	NULL	0.06	2022-07-25	Series E	United States	272
&Open	Dublin	Marketing	9	0.09	2022-11-17	Series A	Ireland	35
#Paid	Toronto	Marketing	19	0.17	2023-01-27	Series B	Canada	21
100 Thieves	Los Angeles	Consumer	12	NULL	2022-07-13	Series C	United States	120
10X Genomics	SF Bay Area	Healthcare	100	0.08	2022-08-04	Post-IPO	United States	242
1stdibs	New York City	Retail	70	0.17	2020-04-02	Series D	United States	253
2TM	Sao Paulo	Crypto	90	0.12	2022-06-01	Unknown	Brazil	250
2TM	Sao Paulo	Crypto	100	0.15	2022-09-01	Unknown	Brazil	250
2U	Washington D.C.	Education	NULL	0.2	2022-07-28	Post-IPO	United States	426
54gene	Washington D.C.	Healthcare	95	0.3	2022-08-29	Series B	United States	44
5B Solar	Sydney	Energy	NULL	0.25	2022-06-03	Series A	Australia	12
6sense	SF Bay Area	Sales	150	0.1	2022-10-12	Series E	United States	426
80 Acres Farms	Cincinnati	Food	NULL	0.1	2023-01-18	Unknown	United States	275

calculates the total layoffs for each month (in YYYY-MM format) by extracting the first 7 characters of the date field, grouping by month, and ordering the results in ascending order by month.

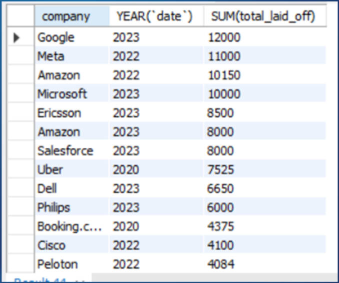
```
WITH rolling_total AS (
SELECT SUBSTRING(`date`,1,7) AS `MONTH`,
SUM(total_laid_off) AS total_off
FROM layoffs_staging2
WHERE SUBSTRING(`date`,1,7) IS NOT NULL
GROUP BY `MONTH`
ORDER BY 1 ASC)
SELECT `MONTH`, total_off, SUM(total_off) OVER(
ORDER BY `MONTH`) AS rolling_total
FROM rolling_total;
```



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
2021-01	6813	87811
2021-02	868	88679
2021-03	47	88726

This query calculates the total layoffs per month from the `layoffs_staging2` table and then computes a cumulative (rolling) total of layoffs over time, ordered by month.

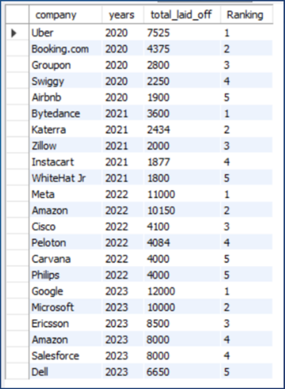
```
SELECT company, YEAR(`date`),
SUM(total_laid_off)
FROM layoffs_staging2
GROUP BY company, YEAR(`date`)
ORDER BY 3 DESC;
```



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.c...	2020	4375
Cisco	2022	4100
Peloton	2022	4084

This query sums the total layoffs by company and year from the `layoffs_staging2` table, grouping by company and year, and orders the results by the total layoffs in descending order.

```
WITH company_Year (company,years,total_laid_off) AS (SELECT
company, YEAR(`date`),SUM(total_laid_off)
FROM layoffs_staging2
GROUP BY company,
YEAR(`date`)),
company_year_rank AS
(
SELECT *,
DENSE_RANK()  OVER(
PARTITION BY years
ORDER BY
total_laid_off DESC)
AS Ranking
FROM company_Year
WHERE years IS NOT NULL
)
SELECT *
FROM company_year_rank
WHERE Ranking <=5;
```



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
Katerra	2021	2434	2
Zillow	2021	2000	3
Instacart	2021	1877	4
WhiteHat Jr	2021	1800	5
Meta	2022	11000	1
Amazon	2022	10150	2
Cisco	2022	4100	3
Peloton	2022	4084	4
Carvana	2022	4000	5
Philips	2022	4000	5
Google	2023	12000	1
Microsoft	2023	10000	2
Ericsson	2023	8500	3
Amazon	2023	8000	4
Salesforce	2023	8000	4
Dell	2023	6650	5

This query first calculates the total layoffs by company and year from the `layoffs_staging2` table. It then ranks companies within each year by the total layoffs using a dense rank, and retrieves the top 5 companies with the highest layoffs per year.