Layoff Data Analysis Project

SQL-BASED DATA CLEANING AND EXPLORATORY DATA ANALYSIS

ARIF AMIN

Contents

Introduction

- Project Overview:
  - This project focuses on analyzing global tech layoffs using SQL.
  - Two key phases: Data Cleaning and Exploratory Data Analysis (EDA).

- Purpose:
  - Clean raw layoff data to make it analysis-ready.
  - Extract patterns, trends, and insights on layoffs by time, region, and company.

Data Preparation

- Dataset Overview:
  - Imported data into MySQL environment.
  - Initial Table: layoffs
  - Staging Tables: layoffs staging, layoffs_staging2

- Columns:
  - company, location, industry, total_laid_off, percentage_laid_off, date, stage, country, funds_raised_millions

- Transformations:
  - Created layoffs_staging for cleaning operations.
  - Used layoffs_staging2 to track and delete duplicates using row numbers.

- Record Count: 2500+ rows processed through cleaning and standardization.

Objective

- Primary Goal:
- Identify significant patterns in layoffs from a cleaned dataset.

- Key Focus Areas:
- Which companies, countries, and industries had the most layoffs?
- Monthly and yearly trends in layoffs.
- Ranking of companies by layoffs over time.
- Rolling cumulative layoffs across months.

Methodology

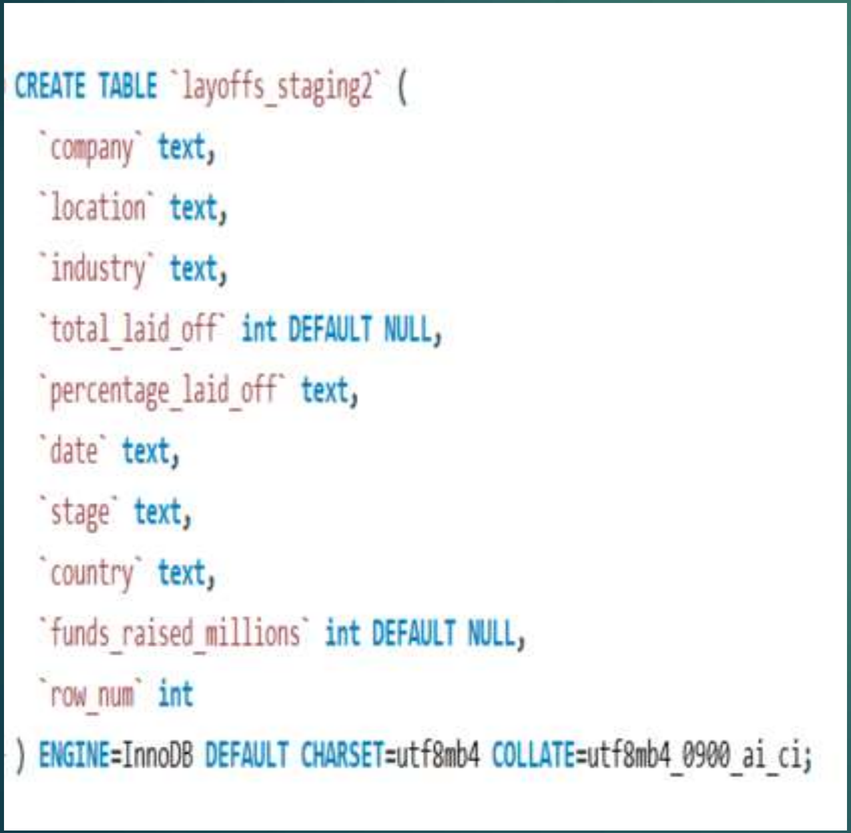
SQL Functions Used:

- Window Functions: ROW_NUMBER(),DENSE_RANK(), SUM() OVER.
- Text Functions: TRIM(), SUBSTRING().
- Date Conversion: STR_TO_DATE().
- Conditional Clauses: WHERE, IS NULL, LIKE.
- Joins: Self-joins for populating missing values.
- CTEs: Used for modular, readable SQL (e.g., ranking and rolling totals).

Data Cleaning

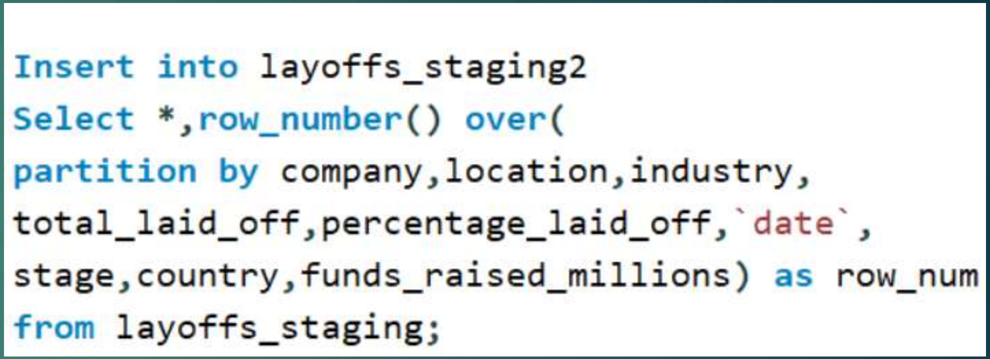
Removed Duplicates

Created table layoff_staging2



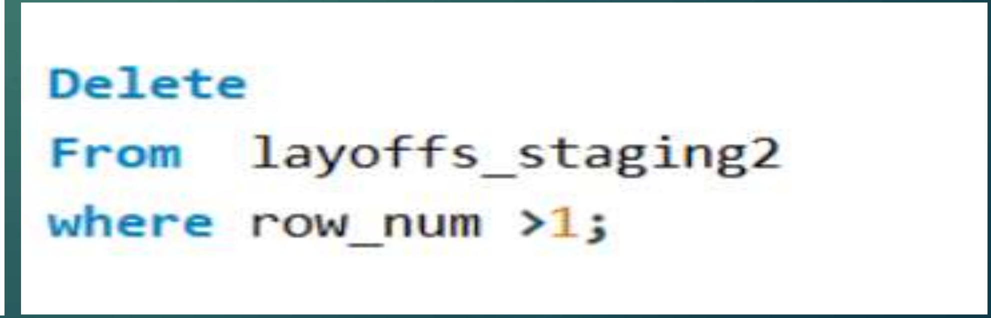
```
CREATE TABLE `layoffs_staging2` (
  `company` text,
  `location` text,
  `industry` text,
  `total_laid_off` int DEFAULT NULL,
  `percentage_laid_off` text,
  `date` text,
  `stage` text,
  `country` text,
  `funds_raised_millions` int DEFAULT NULL,
  `row_num` int
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

Inserted data into layoff_staging 2



```
Insert into layoffs_staging2
Select *,row_number() over(
partition by company,location,industry,
total_laid_off,percentage_laid_off,`date`,
stage,country,funds_raised_millions) as row_num
from layoffs_staging;
```

Deleted Data from layoff_staging 2

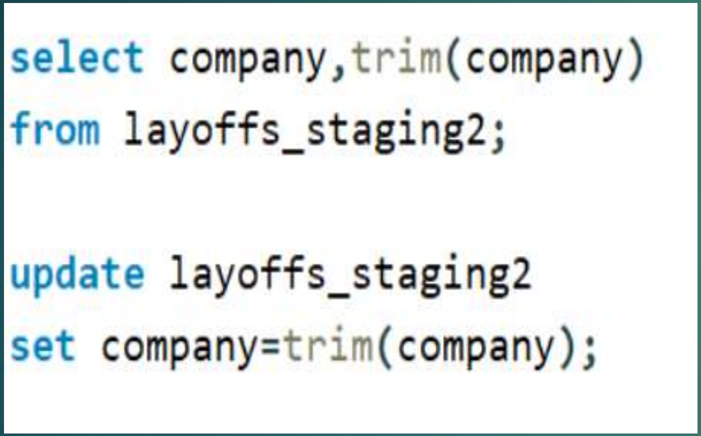


```
Delete
From  layoffs_staging2
where row_num >1;
```

Standardized the data

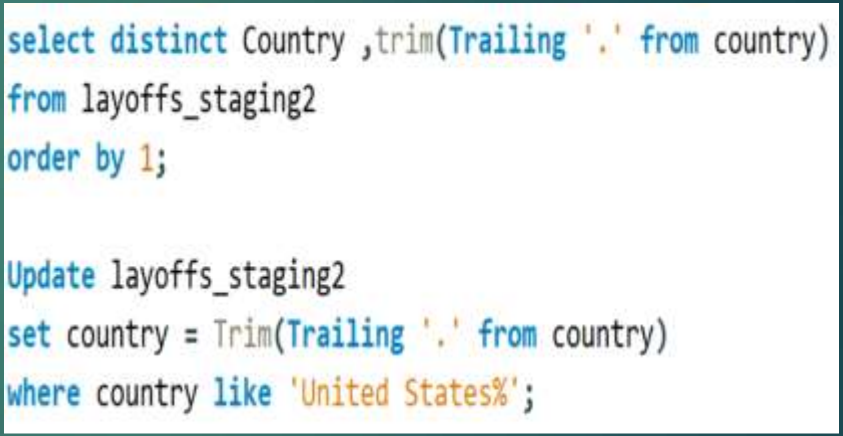
Removed extra-spaces from company column

Removed trailing "." from country column



```sql
select company,trim(company)
from layoffs_staging2;

update layoffs_staging2
set company=trim(company);
```

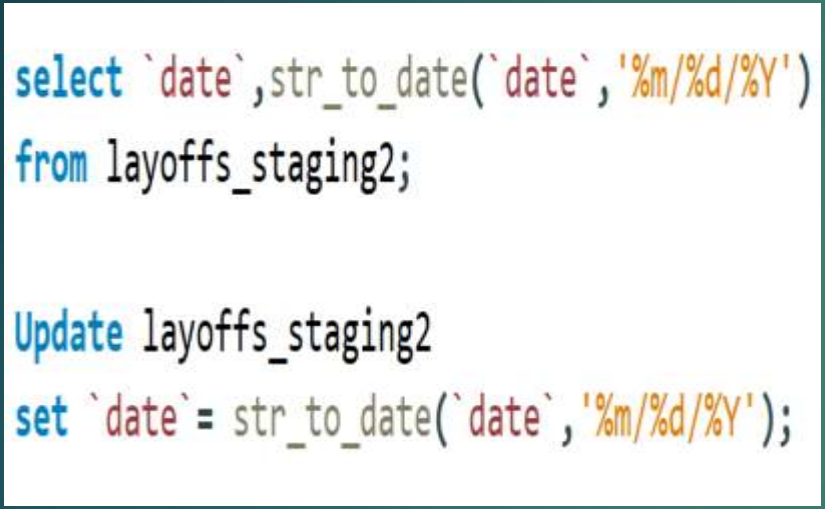


```sql
select distinct Country ,trim(Trailing '.' from country)
from layoffs_staging2
order by 1;

Update layoffs_staging2
set country = Trim(Trailing '.' from country)
where country like 'United States%';
```

Standardized the data

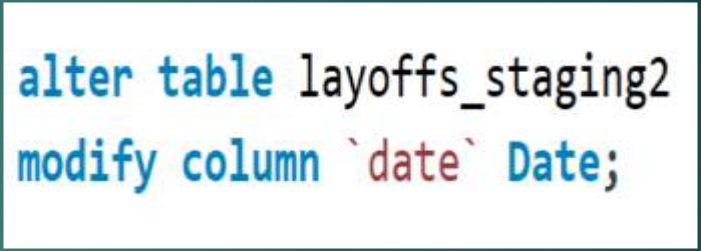
Standardized the date format



```
select `date`,str_to_date(`date`,'%m/%d/%Y')
from layoffs_staging2;

Update layoffs_staging2
set `date`= str_to_date(`date`,'%m/%d/%Y');
```

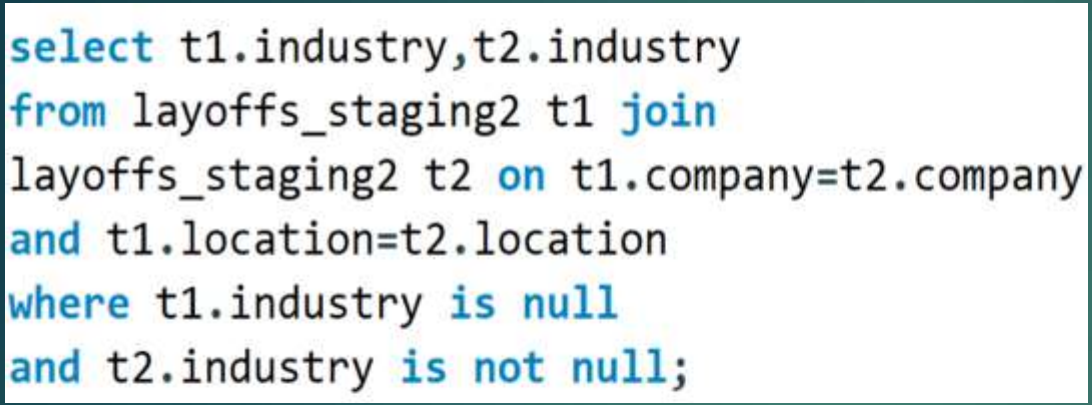
Changed datatype to date



```
alter table layoffs_staging2
modify column `date` Date;
```

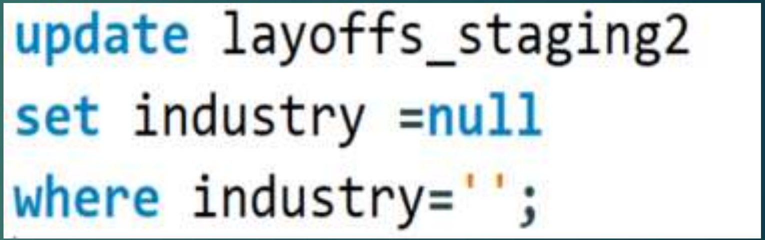
Updated Null & Blank Values

Self-joined the layoffs_staging2 table



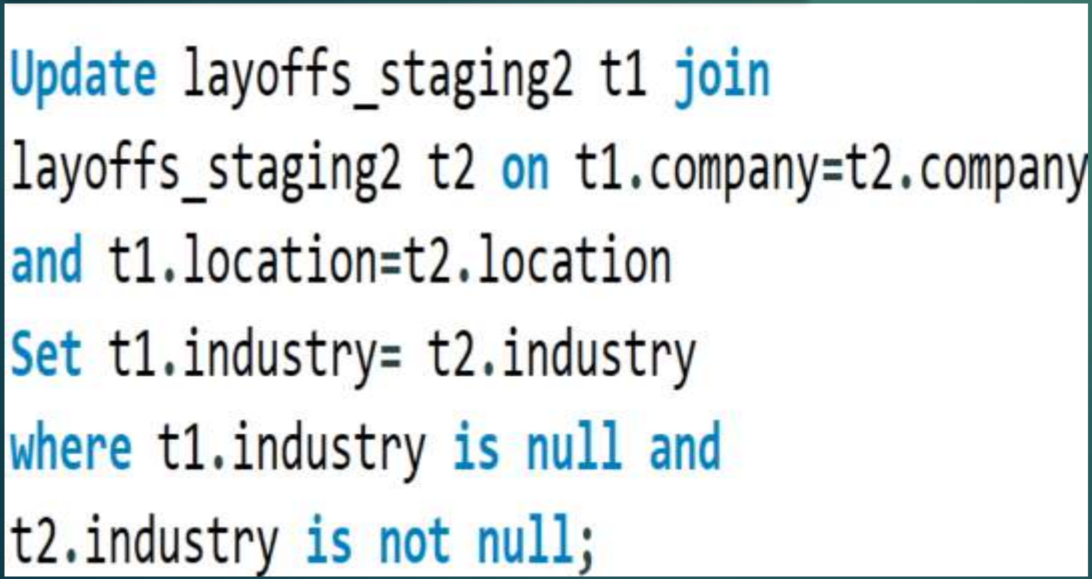
```
select t1.industry,t2.industry
from layoffs_staging2 t1 join
layoffs_staging2 t2 on t1.company=t2.company
and t1.location=t2.location
where t1.industry is null
and t2.industry is not null;
```

Updated blank with null for industry column



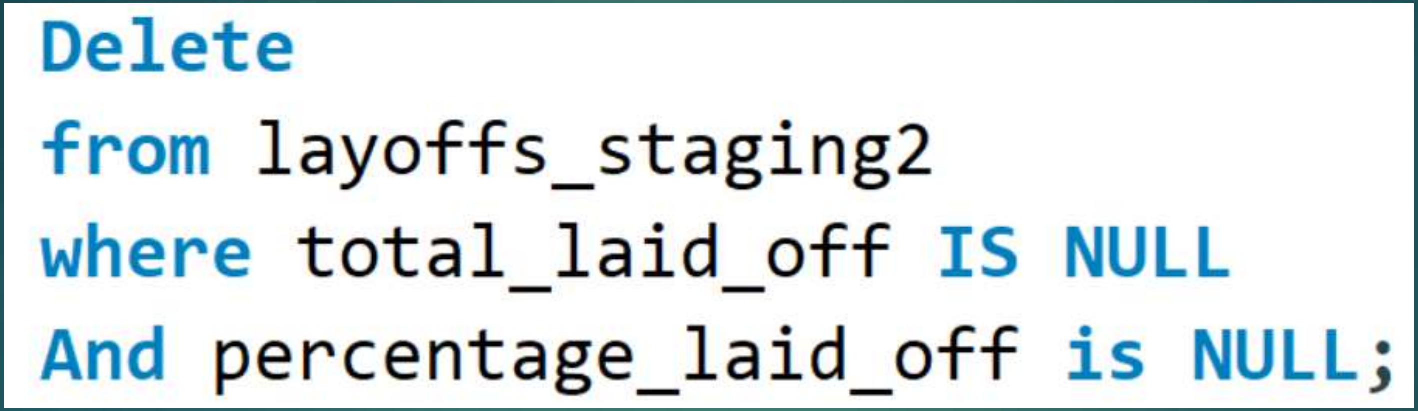
```
update layoffs_staging2
set industry =null
where industry='';
```

Updated the null Industry values



```
Update layoffs_staging2 t1 join
layoffs_staging2 t2 on t1.company=t2.company
and t1.location=t2.location
Set t1.industry= t2.industry
where t1.industry is null and
t2.industry is not null;
```

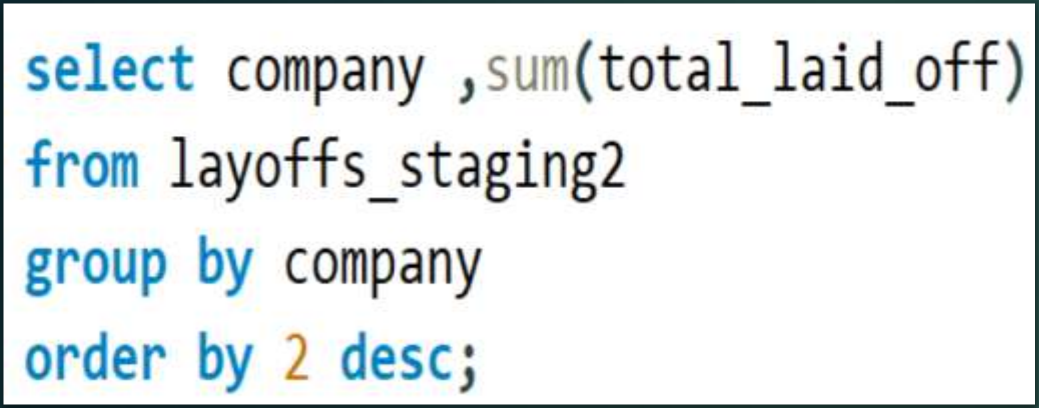
Removed the Null Values



```
Delete
from layoffs_staging2
where total_laid_off IS NULL
And percentage_laid_off is NULL;
```

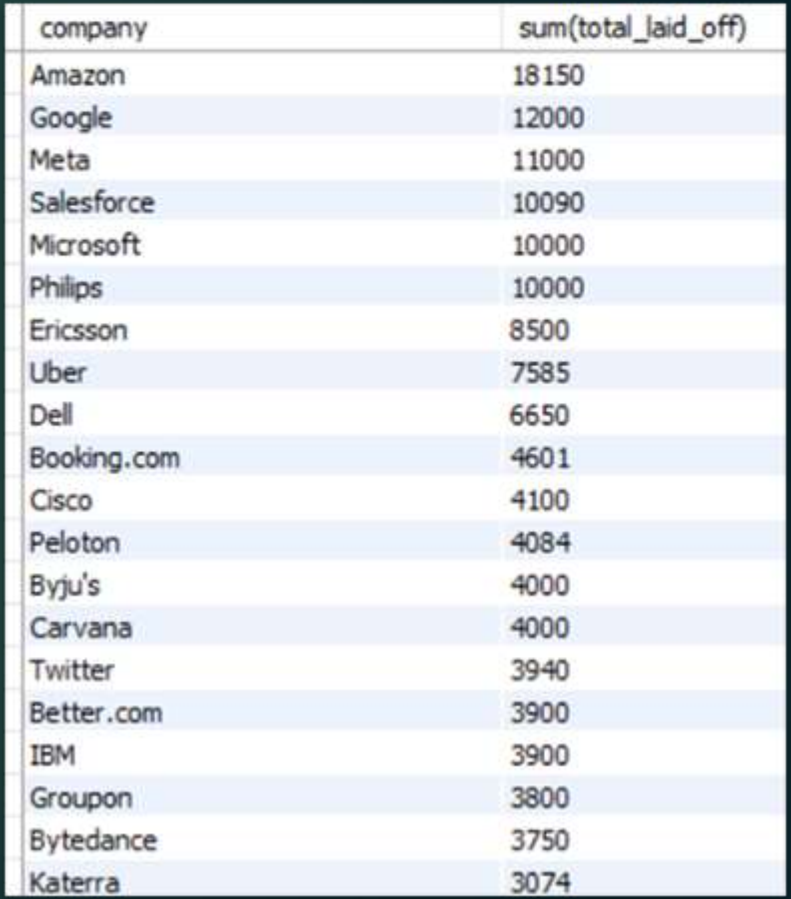
Exploratory Data Analysis

Company-wise highest lay-offs



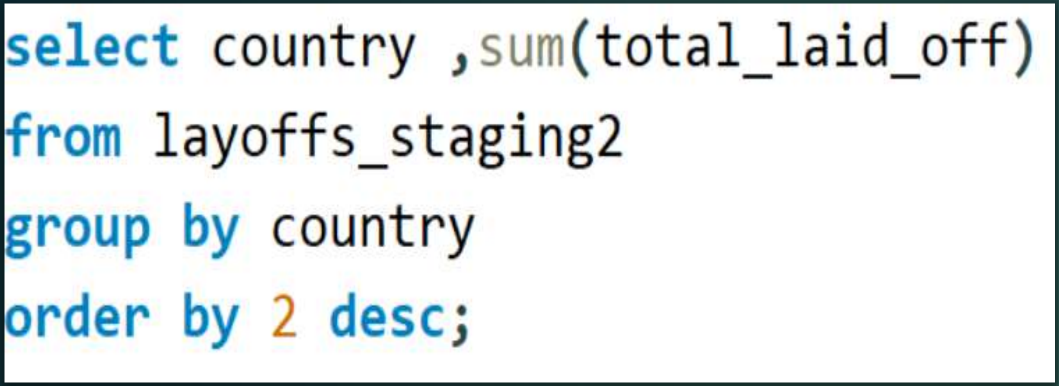
```
select company ,sum(total_laid_off)
from layoffs_staging2
group by company
order by 2 desc;
```

Amazon, Google and Meta had
the highest layoffs.

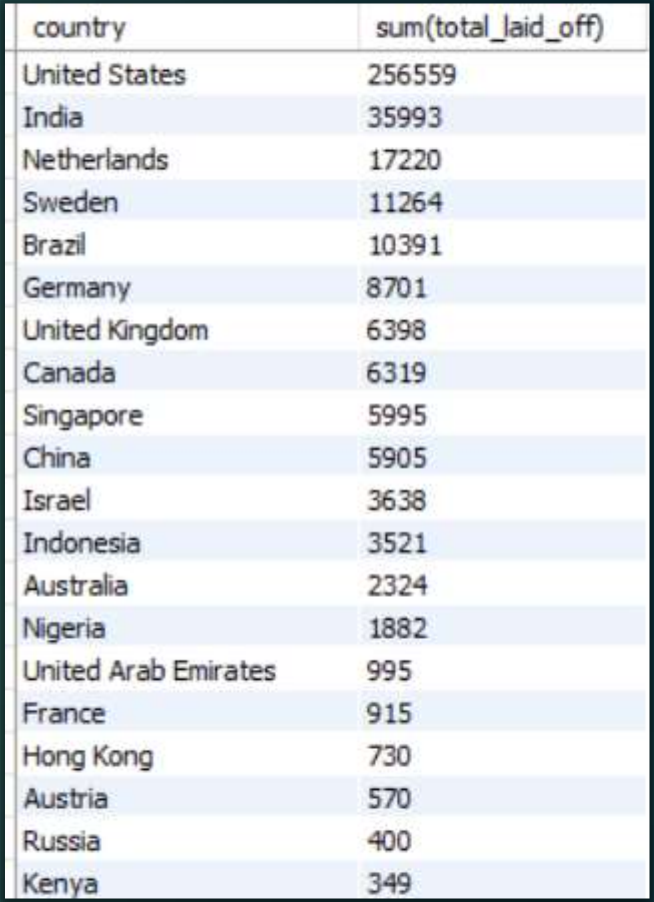


company	sum(total_laid_off)
Amazon	18150
Google	12000
Meta	11000
Salesforce	10090
Microsoft	10000
Philips	10000
Ericsson	8500
Uber	7585
Dell	6650
Booking.com	4601
Cisco	4100
Peloton	4084
Byju's	4000
Carvana	4000
Twitter	3940
Better.com	3900
IBM	3900
Groupon	3800
Bytedance	3750
Katerra	3074

Country-wise highest lay-offs



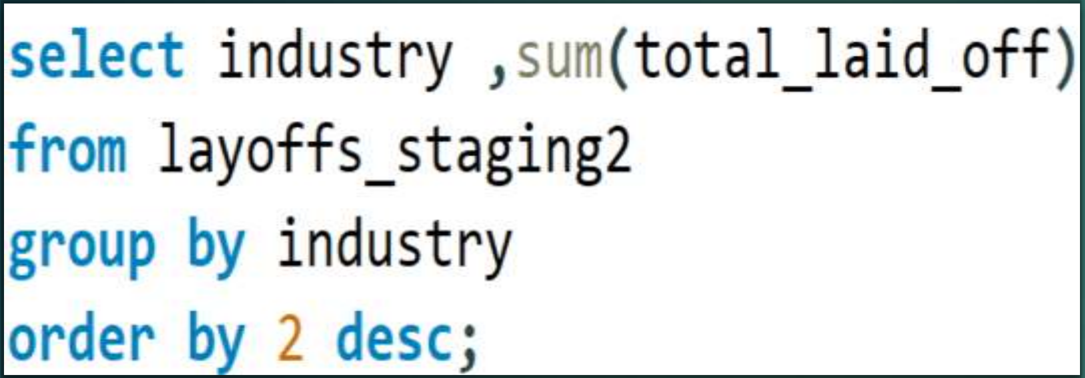
```
select country ,sum(total_laid_off)
from layoffs_staging2
group by country
order by 2 desc;
```



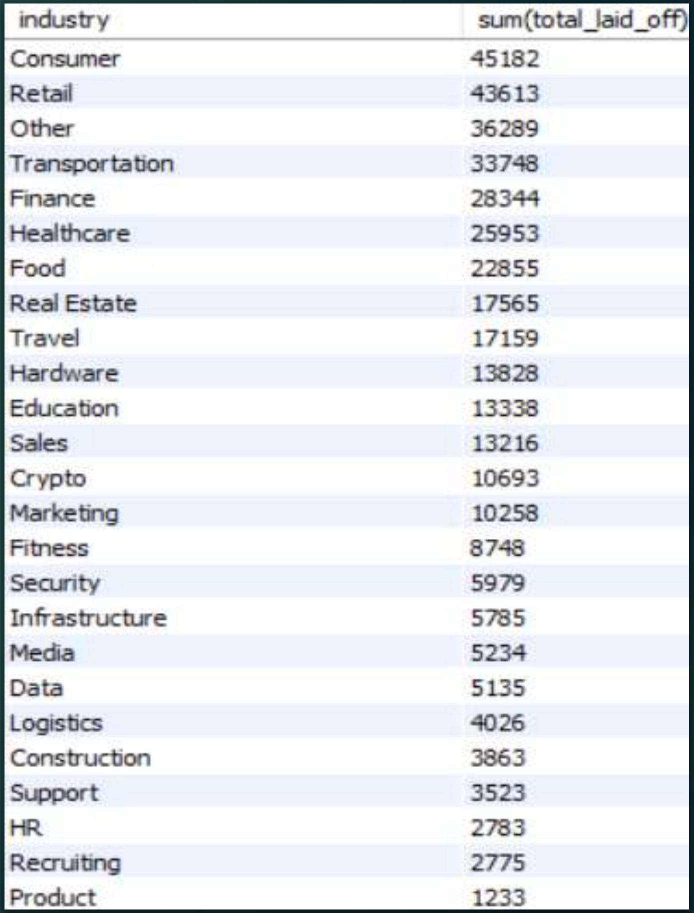
country	sum(total_laid_off)
United States	256559
India	35993
Netherlands	17220
Sweden	11264
Brazil	10391
Germany	8701
United Kingdom	6398
Canada	6319
Singapore	5995
China	5905
Israel	3638
Indonesia	3521
Australia	2324
Nigeria	1882
United Arab Emirates	995
France	915
Hong Kong	730
Austria	570
Russia	400
Kenya	349

The United States saw overwhelming majority of layoffs.

Industry-wise the highest laid off



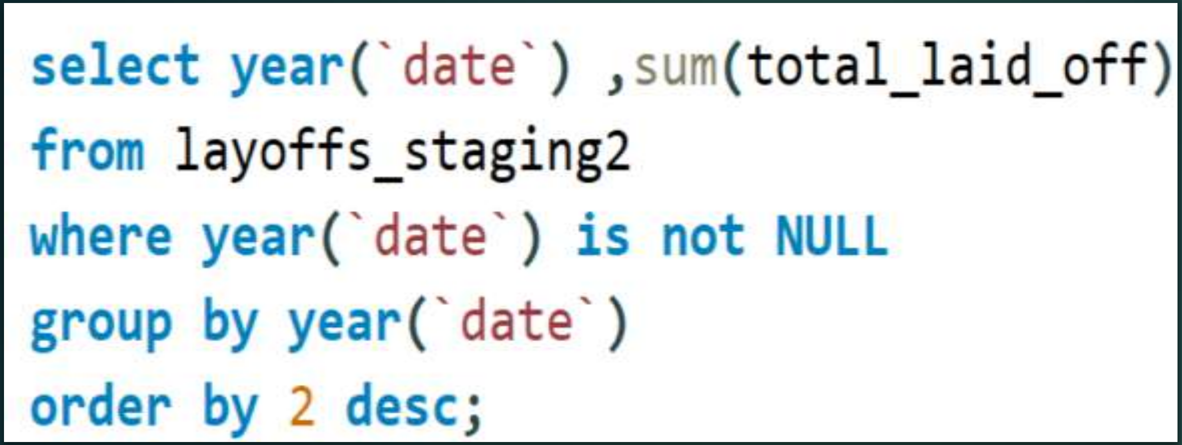
```
select industry ,sum(total_laid_off)
from layoffs_staging2
group by industry
order by 2 desc;
```



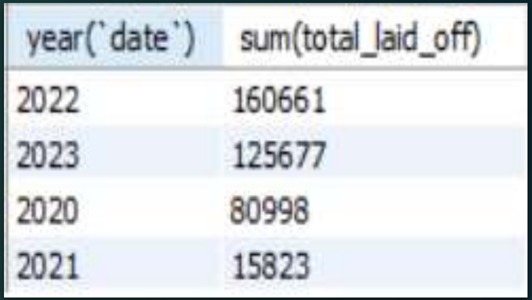
industry	sum(total_laid_off)
Consumer	45182
Retail	43613
Other	36289
Transportation	33748
Finance	28344
Healthcare	25953
Food	22855
Real Estate	17565
Travel	17159
Hardware	13828
Education	13338
Sales	13216
Crypto	10693
Marketing	10258
Fitness	8748
Security	5979
Infrastructure	5785
Media	5234
Data	5135
Logistics	4026
Construction	3863
Support	3523
HR	2783
Recruiting	2775
Product	1233

Consumer , retail and transportation industries were hit hard.

Which year had the highest lay-offs



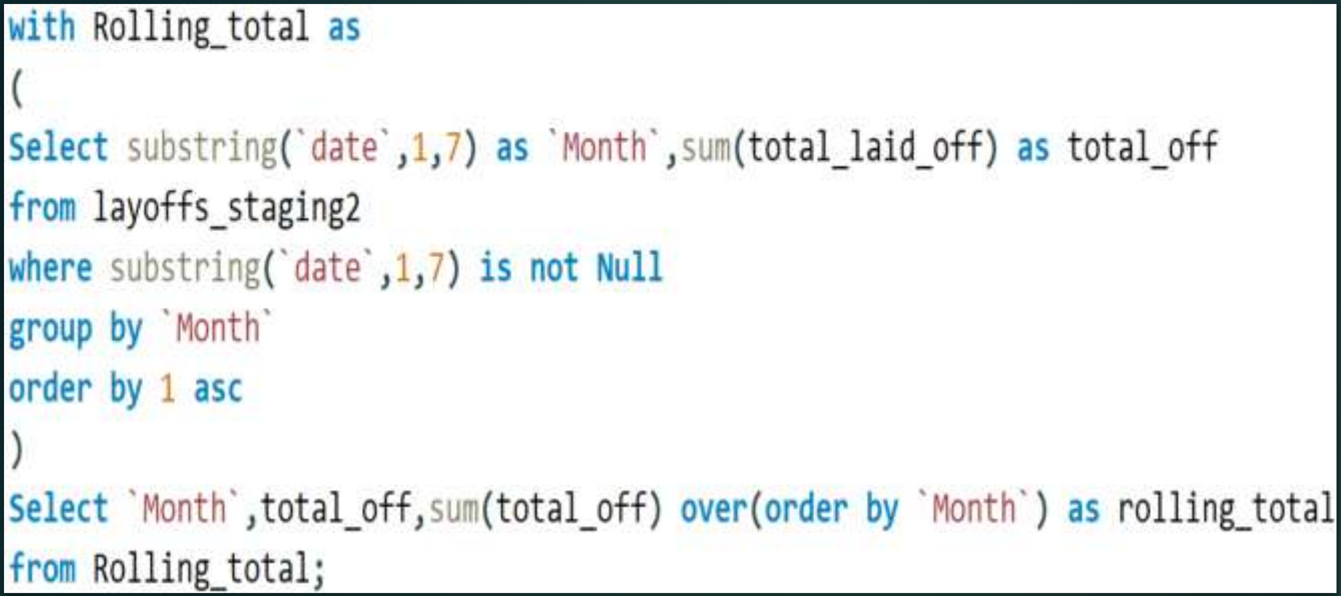
```
select year(`date`) ,sum(total_laid_off)
from layoffs_staging2
where year(`date`) is not NULL
group by year(`date`)
order by 2 desc;
```



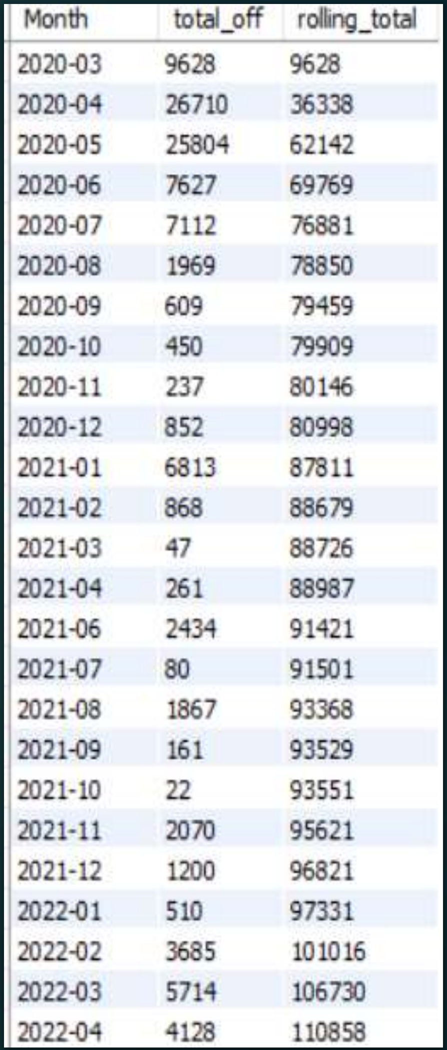
year(`date`)	sum(total_laid_off)
2022	160661
2023	125677
2020	80998
2021	15823

2022 had the highest layoffs and early 2023.

Progression of lay-off, rolling sum until the end



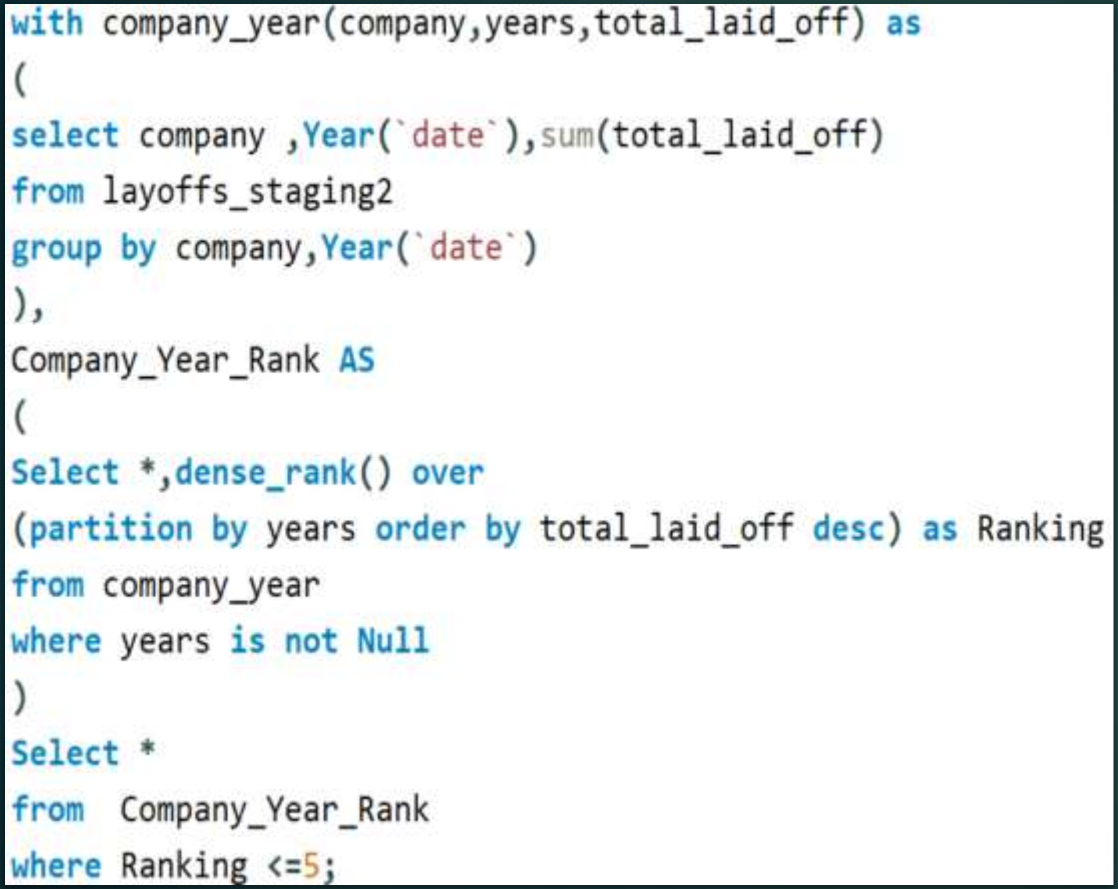
```
with Rolling_total as
(
Select substring(`date`,1,7) as `Month`,sum(total_laid_off) as total_off
from layoffs_staging2
where substring(`date`,1,7) is not Null
group by `Month`
order by 1 asc
)
Select `Month`,total_off,sum(total_off) over(order by `Month`) as rolling_total
from Rolling_total;
```



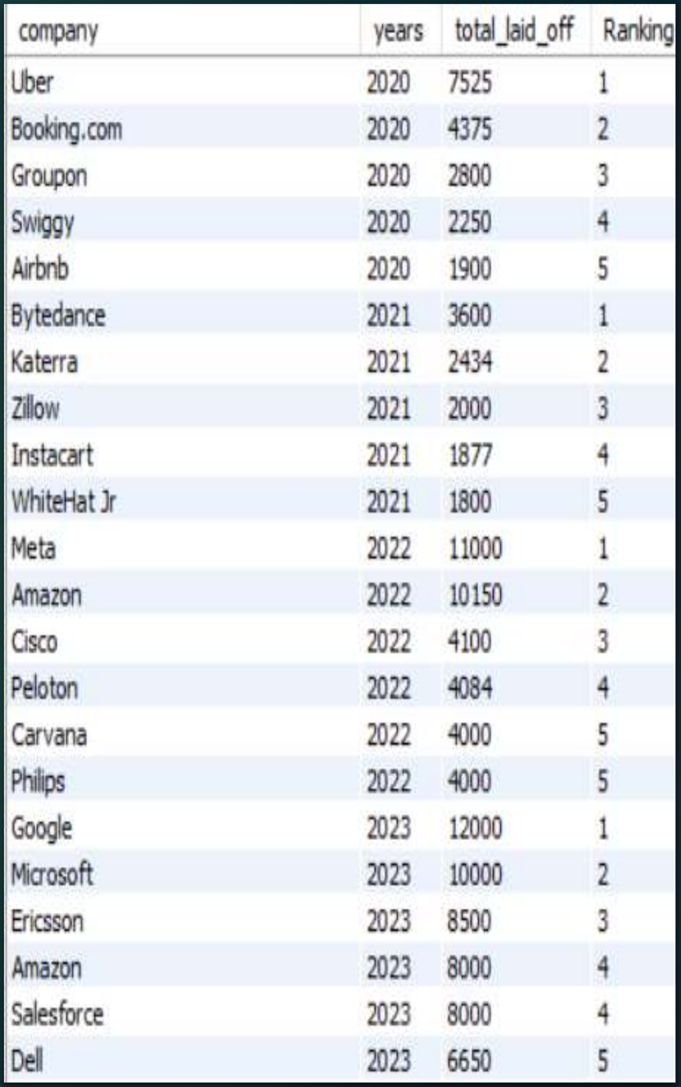
Month	total_off	rolling_total
2020-03	9628	9628
2020-04	26710	36338
2020-05	25804	62142
2020-06	7627	69769
2020-07	7112	76881
2020-08	1969	78850
2020-09	609	79459
2020-10	450	79909
2020-11	237	80146
2020-12	852	80998
2021-01	6813	87811
2021-02	868	88679
2021-03	47	88726
2021-04	261	88987
2021-06	2434	91421
2021-07	80	91501
2021-08	1867	93368
2021-09	161	93529
2021-10	22	93551
2021-11	2070	95621
2021-12	1200	96821
2022-01	510	97331
2022-02	3685	101016
2022-03	5714	106730
2022-04	4128	110858

Layoffs surged in 2022, with rolling cumulative totals highlighting a steep upward curve through late 2022 and early 2023.

Identify the top 5 companies with the highest total layoffs for each year



```
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

Uber, Byte-dance, Meta and Google
lead the lay-offs each year.

Insights

- Findings from SQL Analysis:
- 2022 had the highest layoffs, followed by partial data from early 2023.
- USA led in total layoffs, significantly ahead of other countries.
- **Rolling layoff trends** showed consistent month-over-month increase during peak periods.
- Consumer retail, transportation industries were the most impacted.
- Several companies experienced 100% layoffs.
- Companies with high funding still saw large layoffs.

- Recommendations:
- **Standardize data entry**: Enforce naming conventions for countries, companies, and industries.
- **Add workforce size**: Enables better understanding of impact (not just absolute layoffs).
- **Investigate funding vs. layoffs**: Use additional data to analyze correlation.

Conclusion

- Cleaned and standardized layoff dataset using SQL.
- **Exploratory analysis** revealed trends by time, location, and company scale. **SQL functions (window, CTEs, joins)** enabled effective analysis on structured data.
- Dataset ready for visualization and advanced analytics.

Thank You