LAYOFFS DATA CLEANING AND EXPLORATORY ANALYSIS

BY MOHAMMAD JUNAID

+91 9030816488

github.com/junaid876/layoffs-data-analysis

Table of Contents

Introduction

In this project, I worked on cleaning and analyzing a dataset containing layoffs data from multiple companies across different industries and regions. Real-world datasets are often messy, with missing values, inconsistent formatting, and duplicate records. Through structured SQL techniques, I cleaned the dataset, standardized entries, handled missing values, and performed exploratory data analysis to extract meaningful insights.

This document provides an overview of the dataset, the challenges encountered, the approach followed, the SQL queries implemented, and the insights discovered.

Dataset Overview

The dataset, named layoffs, contains the following fields:

Column	Description
company	Name of the company
location	City or area where the layoffs occurred
industry	Sector of the company
total_laid_off	Number of employees laid off
percentage_laid_off	Percentage of employees laid off
date	Date when layoffs were announced
stage	Funding stage of the company
country	Country of the company
funds_raised_millions	Funding raised in millions USD

Issues identified in the dataset:

- Duplicate rows across multiple columns
- Inconsistent naming (e.g., industries, countries)
- Missing values in important columns
- Date fields in unreadable formats

Problem Statement

The raw dataset was difficult to work with due to:

- Repeated entries causing inaccuracies
- Blank or null values affecting aggregation
- Unstructured data making analysis unreliable
- Formatting inconsistencies in dates and text fields

The objective was to clean the dataset and structure it for proper analysis.

company	location	industry	total_laid_off	percentage_laid	date	stage	country	funds_raised_millions
Atlassian	Sydney	Other	500	0.05	3/6/2023	Post-IPO	Australia	210
SiriusXM	New York City	Media	475	0.08	3/6/2023	Post-IPO	United States	525
Alerzo	Ibadan	Retail	400	NULL	3/6/2023	Series B	Nigeria	16
UpGrad	Mumbai	Education	120	NULL	3/6/2023	Unknown	India	631
Loft	Sao Paulo	Real Estate	340	0.15	3/3/2023	Unknown	Brazil	788
Embark Trucks	SF Bay Area	Transportation	230	0.7	3/3/2023	Post-IPO	United States	317
Lendi	Sydney	Real Estate	100	NULL	3/3/2023	Unknown	Australia	59
UserTesting	SF Bay Area	Marketing	63	NULL	3/3/2023	Acquired	United States	152
Airbnb	SF Bay Area		30	NULL	3/3/2023	Post-IPO	United States	6400
Accolade	Seattle	Healthcare	NULL	NULL	3/3/2023	Post-IPO	United States	458
Indigo	Boston	Other	NULL	NULL	3/3/2023	Series F	United States.	1200
Zscaler	SF Bay Area	Security	177	0.03	3/2/2023	Post-IPO	United States	148
MasterClass	SF Bay Area	Education	79	NULL	3/2/2023	Series E	United States	461
Ambev Tech	Blumenau	Food	50	NULL	3/2/2023	Acquired	Brazil	NULL
Fittr	Pune	Fitness	30	0.11	3/2/2023	Series A	India	13
CNET	SF Bay Area	Media	12	0.1	3/2/2023	Acquired	United States	20
Flipkart	Bengaluru	Retail	NULL	NULL	3/2/2023	Acquired	India	12900
Kandela	Los Angeles	Consumer	NULL	1	3/2/2023	Acquired	United States	NULL
Truckstop.com	Boise	Logistics	NULL	NULL	3/2/2023	Acquired	United States	NULL
Thoughtworks	Chicago	Other	500	0.04	3/1/2023	Post-IPO	United States	748
iFood	Sao Paulo	Food	355	0.06	3/1/2023	Subsidiary	Brazil	2100
Color Health	SF Bay Area	Healthcare	300	NULL	3/1/2023	Series E	United States	482
Waymo	SF Bay Area	Transportation	209	0.08	3/1/2023	Subsidiary	United States	5500
PayFit	Paris	HR	200	0.2	3/1/2023	Series E	France	495

Figure 1: Snapshot of the raw dataset before cleaning.

4.Data Cleaning Approach

The cleaning process was divided into four main steps using SQL:

4.1 Removing Duplicate Data

Duplicates were identified using the ROW_NUMBER() window function partitioned by columns such as company, location, industry, etc. Duplicates were removed by keeping only the first record.

Key Queries:

```
WITH duplicate_cte AS (
 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
)
DELETE FROM layoffs_staging
WHERE row_num > 1;
```

4.2 Standardizing Text Fields

Company Names:

Trimmed extra spaces using:

```
UPDATE layoffs_staging
SET company = TRIM(company);
```

Industry Names:

Standardized entries such as “Crypto-based” to “Crypto”:

```
UPDATE layoffs_staging
SET industry = 'Crypto'
WHERE industry LIKE 'Crypto%';
```

Country Names:

Removed trailing periods from “United States.” and similar entries:

```
UPDATE layoffs_staging
SET country = TRIM(TRAILING '.' FROM country)
WHERE country LIKE 'United States%';
```

4.3 Handling Null and Blank Values

Industry:

Blank entries were set to NULL:

```
UPDATE layoffs_staging
SET industry = NULL
WHERE industry = '';
```

Values were filled using other available records:

```
UPDATE layoffs_staging t1
JOIN layoffs_staging t2
ON t1.company = t2.company
SET t1.industry = t2.industry
WHERE t1.industry IS NULL
AND t2.industry IS NOT NULL;
```

Total Laid Off and Percentage:
Rows where both were missing were removed:

```
DELETE FROM layoffs_staging
WHERE total_laid_off IS NULL
AND percentage_laid_off IS NULL;
```

layoffs_cleaned

company	location	industry	total_laid_off	percentage_laid	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
8x8	SF Bay Area	Support	155	0.07	2023-01-18	Post-IPO	United States	253
8x8	SF Bay Area	Support	200	0.09	2022-10-04	Post-IPO	United States	253
98point6	Seattle	Healthcare	NULL	0.1	2022-07-21	Series E	United States	247
99	Sao Paulo	Transportation	75	0.02	2022-09-20	Acquired	Brazil	244
Abra	SF Bay Area	Crypto	12	0.05	2022-06-30	Series C	United States	106
Absci	Vancouver	Healthcare	40	NULL	2022-08-09	Post-IPO	United States	237
Acast	Stockholm	Media	70	0.15	2022-09-15	Post-IPO	Sweden	126
Acko	Mumbai	Finance	45	0.09	2020-04-01	Unknown	India	143
Acorns	Portland	Finance	50	NULL	2020-05-26	Unknown	United States	207
Actifio	Boston	Data	54	NULL	2020-12-16	Acquired	United States	352
ActiveCampaign	Chicago	Marketing	NULL	0.15	2022-10-03	Series C	United States	360

layoffs_cleaned.csv

Figure 2: Cleaned dataset with standardized and properly formatted entries.

4.4 Formatting Dates:
Converted date strings into DATE format using:

```
UPDATE layoffs_staging
SET `date` = STR_TO_DATE(`date`, '%m/%d/%Y');

ALTER TABLE layoffs_staging
MODIFY COLUMN `date` DATE;
```

5. Exploratory Data Analysis (EDA)

After cleaning, multiple analytical queries were run to extract patterns.

5.1 Aggregation by Company

```
SELECT company, SUM(total_laid_off)
FROM layoffs_staging
GROUP BY company
ORDER BY 2 DESC;
```

This query revealed which companies had the highest layoffs.

5.2 Aggregation by Industry

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

his showed the most impacted sectors.

5.3 Yearly Trends

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

This displayed how layoffs fluctuated over the years.

5.4 Rolling Total Over Months

```
WITH rolling_total AS (
  SELECT SUBSTRING(`date`,1,7) AS month,
      SUM(total_laid_off) AS total_off
  FROM layoffs_staging
  GROUP BY month
)
SELECT month, total_off, SUM(total_off) OVER (ORDER BY month) AS
cumulative_off
FROM rolling_total;
```

Used to track cumulative layoffs over time.

5.5 Ranking by Year

```
WITH company_year AS (
  SELECT company, YEAR(`date`) AS year, SUM(total_laid_off) AS total_off
  FROM layoffs_staging
  GROUP BY company, YEAR(`date`)
),
ranking AS (
  SELECT *,
    DENSE_RANK() OVER (
      PARTITION BY year
      ORDER BY total_off DESC
    ) AS rank
  FROM company_year
)
SELECT *
FROM ranking
WHERE rank <= 5;
```

This query identified the top 5 companies affected by layoffs each year.

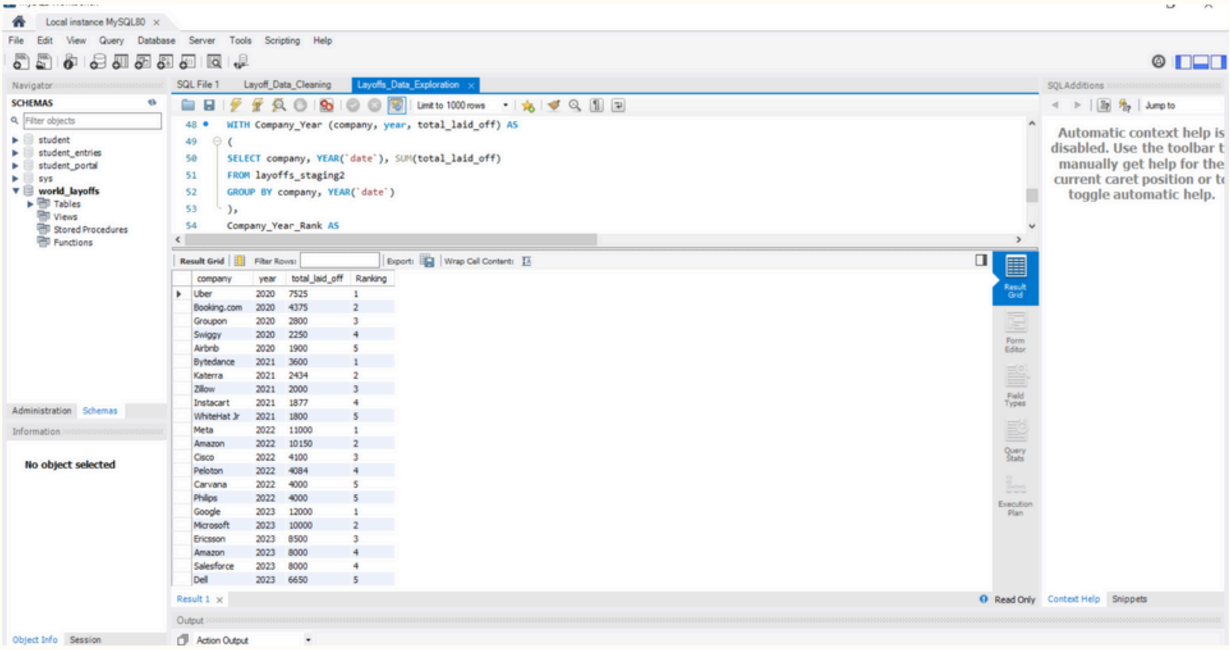

Figure 3: Top 5 Companies by Layoffs for Each Year (2020–2023)

This table displays the top 5 companies with the highest layoffs for each year, using the DENSE_RANK() function. It helps identify which companies faced the most workforce reductions over time.

6. Insights and Observations

- Industry Trends: Industries like technology, crypto, and healthcare showed higher layoffs.
- Time Trends: Layoffs peaked at certain months and years, indicating seasonal or economic patterns.
- Key Companies: Some companies consistently had large layoffs, which could be a sign of internal restructuring or market shifts.

7. Conclusion

This project demonstrates how structured SQL techniques can transform messy data into actionable insights. The cleaned dataset is now ready for deeper analysis or visualization using dashboards. The methodologies applied here can be adapted to other datasets with similar challenges.

8. Contact

Mohammad Junaid
📧 mdjunaid34343@gmail.com
🔗 github.com/junaid876/layoffs-data-analysis

LARANA
STUDIO

THANK YOU

+91 9030816488 mdjunaid34343@gmail.com