World Layoffs Data Cleaning & EDA: Using SQL

Abhilash Antony

Introduction…

In the modern corporate landscape, layoffs have become a critical indicator of economic health and business sustainability. The ability to analyse and understand layoff trends provides businesses and policymakers with valuable insights that can inform decision-making, guide policy interventions, and mitigate the impact of economic downturns. This report presents an analysis of layoffs data, focusing on the systematic cleaning and exploration of the dataset to uncover meaningful trends.

The primary objective of this analysis is to clean the layoffs dataset and extract actionable insights. This involves handling issues like duplicates, null values, and data inconsistencies to ensure the data is accurate and reliable. Additionally, we aim to explore key trends such as the geographic distribution of layoffs, industry-specific impacts, and temporal patterns over time.

The dataset used for this analysis was sourced from **Kaggle**, a well-known platform for sharing data and analytics. The key columns of interest in this dataset include `company`, `industry`, `country`, `total_laid_off`, and `percentage_laid_off`. These columns provide a foundation for analyzing layoffs across various sectors, regions, and periods.

The data cleaning and exploration process follows a structured approach. The stages include loading the data into a staging table to preserve the raw data, removing duplicates, standardizing column values, handling null entries, and converting data types where necessary. Once the data is cleaned, trend analysis is conducted to extract insights on layoff patterns by industry, geography, and time.

Data Loading and Staging…

The layoffs dataset was sourced from Kaggle, a platform offering a wide variety of datasets for analysis. After downloading the dataset in CSV format, the next step was to load it into SQL to facilitate the cleaning and analysis process. SQL was used as the primary tool for staging and cleaning due to its efficiency in handling large datasets.

The layoffs dataset imported into **MySQL Workbench** using the **Table Data Import Wizard**. This tool allows for quick and easy data import into SQL without manually writing import scripts. The dataset was imported into a raw table to ensure that the original data was preserved.

company	location	industry	total_laid_off	percentage_laid_off	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

The creation of a **staging table** was necessary to preserve the integrity of the original dataset before any cleaning operations were performed. The staging table allows the raw data to be maintained in its original form while we apply various cleaning techniques on a separate version of the data. This ensures data quality by allowing comparisons and rollbacks if necessary. This was achieved using the following chunk of SQL code:

```
CREATE TABLE layoffs_copy LIKE layoffs;
INSERT INTO layoffs_copy SELECT * FROM layoffs;
SELECT * FROM layoffs_copy;
```

Data Cleaning Process…

During the data cleaning process, several key steps are typically followed to ensure data quality. First, we check for and remove any duplicate records that might skew the analysis. Next, the data is standardized and errors are fixed, which may involve correcting inconsistencies in formatting, such as standardizing text fields or addressing discrepancies in numerical entries. Afterward, we address any null values by either filling them with appropriate estimates or removing records where critical information is missing. Finally, unnecessary columns and rows that do not add value to the analysis are removed to streamline the dataset and focus on relevant information.

1. **Removal of Duplicates:**

Removing duplicates is crucial for ensuring the accuracy of the analysis, as duplicate entries can distort metrics like total layoffs or company data. Identifying and removing them prevents overcounting or misrepresentation of data trends.

One approach to removing duplicates is to create a new column that assigns row numbers to each record. This row number indicates how many times a row appears in the table, helping us identify duplicates. We then delete rows where the row number is greater than 1, indicating that they occur more than once in the dataset. After successfully removing the duplicates, the newly added column can be deleted to restore the table to its original form. To ensure the integrity of the raw data, this process can be done in a new table, leaving the original data untouched.

The ROW_NUMBER() function was used to assign a unique sequential integer to rows within a partition of the dataset. This helped in flagging duplicate records based on a specified set of key columns, ensuring that we retain only the first occurrence.

```
-- look for rows that has repeated (more than once)
SELECT * FROM (
    SELECT company, location, industry,
total_laid_off,percentage_laid_off,`date`, stage, country,
funds_raised_millions,
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_copy
) duplicates WHERE row_num > 1;
```

company	location	industry	total_laid_off	percentage_laid_off	date	stage	country	funds_raised_millions	row_num
Casper	New York City	Retail	NULL	NULL	9/14/2021	Post-IPO	United States	339	2
Cazoo	London	Transportation	750	0.15	6/7/2022	Post-IPO	United Kingdom	2000	2
Hibob	Tel Aviv	HR	70	0.3	3/30/2020	Series A	Israel	45	2
Wildlife Studios	Sao Paulo	Consumer	300	0.2	11/28/2022	Unknown	Brazil	260	2
Yahoo	SF Bay Area	Consumer	1600	0.2	2/9/2023	Acquired	United States	6	2

Now that we have this information, to remove the duplicate records, we can delete rows were row_num is greater than 1, and we are free from duplicates now.

```
DELETE FROM layoffs_new WHERE row_num > 1;
```

2. Standardization of Data

- **Removing Extra Spaces**: To clean the company column, extra spaces were removed by using TRIM() to ensure consistent naming across records.

```
-- remove white spaces from the column company

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

- **Standardizing Industry Values**: Similar values in the industry column, such as "Crypto" and "Cryptocurrency," were merged to ensure consistency.

industry
Aerospace
Construction
Consumer
Crypto
Crypto Currency
CryptoCurrency

```
-- make them all 'Crypto'

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

- **Handling Country Column Discrepancies**: The country column was cleaned by removing any punctuation and standardizing country names. wee see that there are two United States where one has a period at the end.

country
United Kingdom
United States
United States.
Uruguay
Vietnam

```
-- remove the trailing period from them

UPDATE layoffs_new SET country = 'United States' WHERE country LIKE 'United States%';
```

- **Converting empty industries to NULL:** There are empty spaces in industries column, converting them to NULL, as it is better with NULL.

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

- **Converting Date Column**: The date column was originally stored as text, and it was converted to a proper DATE type for easier analysis.

```
-- change values in date column to a format of date

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

date	in proper date format
12/16/2022	2022-12-16
7/25/2022	2022-07-25
11/17/2022	2022-11-17
1/27/2023	2023-01-27
7/13/2022	2022-07-13

```
-- but though it is in date format, it is still text, so now need to convert
to date datatype
ALTER TABLE layoffs_new MODIFY COLUMN `date` DATE;
```

3. Handling Null Values

Null values were identified in key columns like industry and *total_laid_off*. We assessed the impact of these missing values and determined a strategy for handling them. For missing industry values, we used company-level data to fill in these blanks where possible, assuming the same industry for a company across records.

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

Records where both *total_laid_off* and *percentage_laid_off* were null were of no use to us as it is the primary columns of interest. These will have to be removed to ensure accuracy in the analysis.

```
-- remove records with null values for both the columns
DELETE FROM layoffs_new WHERE total_laid_off IS NULL AND percentage_laid_off
IS NULL;
```

Also, the *row_num* column created to find the duplicate records can be deleted as it is no longer required.

```
-- remove row_num column we created
ALTER TABLE layoffs_new DROP COLUMN row_num;
SELECT COUNT(*) FROM layoffs_new;
SELECT * FROM layoffs_new;
```

Exploratory Data Analysis (EDA)…

Maximum Layoffs in a Single Event: To begin the analysis, the largest layoffs recorded in the dataset was identified as 12000 in a day.

```
-- max total laid off
SELECT MAX(total_laid_off) AS Max_Layoffs FROM layoffs_new;
```

Range of Percentage Layoffs: Companies that laid off a large percentage of their workforce were explored, focusing on those that laid off 100% of their staff. 1 means 100% means that company has winded up. This highlights companies that ceased operations entirely, giving insight into extreme cases of layoffs.

```
-- Looking at Percentage to see the range of these layoffs
SELECT MAX(percentage_laid_off),  MIN(percentage_laid_off)
FROM layoffs_new WHERE  percentage_laid_off IS NOT NULL;
```

Largest Companies That Ceased Operations: In this part of the analysis, we explored companies that went out of business, measured by their percentage of workforce laid off. This metric is particularly useful to understand which large companies were most severely impacted during the layoff periods.

```
-- largest compnay to shutdown
SELECT * FROM layoffs_new WHERE  percentage_laid_off = 1 ORDER BY
total_laid_off DESC LIMIT 10; -- Katerra: 2434
```

company	location	industry	total_laid_off	percentage_laid_off	date	stage	country	funds_raised_millions
Katerra	SF Bay Area	Construction	2434	1	6/1/2021	Unknown	United States	1600
Butler Hospitality	New York City	Food	1000	1	7/8/2022	Series B	United States	50
Deliv	SF Bay Area	Retail	669	1	5/13/2020	Series C	United States	80
Jump	New York City	Transportation	500	1	5/7/2020	Acquired	United States	11
SEND	Sydney	Food	300	1	5/4/2022	Seed	Australia	3
Stoqo	Jakarta	Food	250	1	4/25/2020	Series A	Indonesia	NULL
HOOQ	Singapore	Consumer	250	1	3/27/2020	Unknown	Singapore	95
Stay Alfred	Spokane	Travel	221	1	5/20/2020	Series B	United States	62
Britishvolt	London	Transportation	206	1	1/17/2023	Unknown	United Kingdom	2400
Planetly	Berlin	Other	200	1	11/4/2022	Acquired	Germany	5

Layoffs Based on Funds Raised: An interesting aspect of the analysis was examining how well-funded companies were affected by layoffs. By correlating layoffs with the amount of funds raised, it provides insight into whether higher funding was a shield against layoffs or not.

```
SELECT * FROM layoffs_new WHERE  percentage_laid_off = 1
ORDER BY funds_raised_millions DESC LIMIT 10;
```

company	location	industry	total_laid_off	percentage_laid_off	date	stage	country	funds_raised_millions
Britishvolt	London	Transportation	206	1	1/17/2023	Unknown	United Kingdom	2400
Quibi	Los Angeles	Media	NULL	1	10/21/2020	Private Equity	United States	1800
Deliveroo Australia	Melbourne	Food	120	1	11/15/2022	Post-IPO	Australia	1700
Katerra	SF Bay Area	Construction	2434	1	6/1/2021	Unknown	United States	1600
BlockFi	New York City	Crypto	NULL	1	11/28/2022	Series E	United States	1000
Aura Financial	SF Bay Area	Finance	NULL	1	1/11/2021	Unknown	United States	584
Openpay	Melbourne	Finance	83	1	2/7/2023	Post-IPO	Australia	299
Pollen	London	Marketing	NULL	1	8/10/2022	Series C	United Kingdom	238
Simple Feast	Copenhagen	Food	150	1	9/7/2022	Unknown	Denmark	173
Arch Oncology	Brisbane	Healthcare	NULL	1	1/13/2023	Series C	United States	155

Companies with most layoffs: Companies with the biggest single Layoff on a single day and companies with most total layoffs were found.

```
-- Companies with the biggest single Layoff on a single day
SELECT company, total_laid_off FROM layoffs_new ORDER BY 2 DESC LIMIT 5;
```

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

```sql
-- Companies with the most Total Layoffs
SELECT company, SUM(total_laid_off) as total_layoffs FROM layoffs_new
GROUP BY company ORDER BY total_layoffs DESC LIMIT 5;
```

company	total_layoffs
Amazon	18150
Google	12000
Meta	11000
Salesforce	10090
Philips	10000

Layoffs by Location: A geographic analysis was conducted to explore how layoffs were distributed across different countries and locations. This helps in understanding which regions were most impacted by layoffs.

```sql
-- layoffs by location
SELECT location, SUM(total_laid_off) FROM layoffs_new
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

```sql
-- layoffs by country
SELECT country, SUM(total_laid_off)
FROM layoffs_new GROUP BY country ORDER BY 2 DESC;
```

country	SUM(total_laid_off)
United States	256559
India	35993
Netherlands	17220
Sweden	11264
Brazil	10391

Yearly Trends: To identify broader temporal patterns, the dataset was analysed for yearly layoff trends. This analysis reveals any significant spikes or drops in layoffs, helping to highlight specific periods where layoffs were more prominent.

```sql
-- layoffs throughout the years
SELECT YEAR(date), SUM(total_laid_off) FROM layoffs_new
WHERE `date` IS NOT NULL GROUP BY YEAR(date) ORDER BY 1 ASC;
```

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

Monthly Trends: Breaking down the layoffs by month allows for a more granular exploration of trends. Observing layoffs on a monthly basis may reveal patterns that are not visible on a yearly scale. However, these trends can be clearly sighted in a visualisation tool.

```
-- layoffs throughout the months
SELECT MONTH(date), SUM(total_laid_off) FROM layoffs_new
WHERE `date` IS NOT NULL GROUP BY MONTH(date) ORDER BY 1 ASC;
```

MONTH(date)	SUM(total_laid_off)
1	92037
2	41046
3	19859
4	31099
5	38689
6	27455
7	23415
8	16891
9	6651
10	17878
11	55758
12	12381

Most Affected Industries: The dataset was further explored to find which industries were hit hardest by layoffs. This helps in identifying which sectors of the economy were most vulnerable to layoffs during the periods covered in the data.

```
SELECT industry, SUM(total_laid_off)
FROM layoffs_new 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

Stage of Companies: An analysis was conducted to determine how layoffs varied depending on the stage of the companies, such as start-ups versus more established companies. This comparison provides useful insights into how a company's maturity affects its ability to navigate layoffs.

```
SELECT stage, SUM(total_laid_off)
FROM layoffs_new 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

Rolling Totals: Rolling totals were calculated to observe the cumulative progression of layoffs over time, either by month or year. This approach provides a clearer view of how layoffs accumulated during the period and highlights significant growth or reduction periods in layoff events.

```
WITH Rolling_Total AS (
SELECT SUBSTRING(`date`, 1, 7) AS `month`, SUM(total_laid_off) AS total_laid
FROM layoffs_new
WHERE SUBSTRING(`date`, 1, 7) IS NOT NULL GROUP BY `month` ORDER BY `month`
ASC
)
SELECT `month`, total_laid, SUM(total_laid)
OVER(ORDER BY `month`) AS rolling_total FROM Rolling_Total;
```

This shows a month by month progression. In 2020, it was 9600s, and by the end, we have 81000 layoffs. By 2021 end, we had 96800 layoffs, which means that the year 2021 was comparatively much better. However, Things started changing dramatically in 2022, showing a high surge up to around 2.5 lakhs. Besides this, the three months from 2023 is really devastating as is saw an increase of above 2 lakhs to get the total to the value close to 4 lakhs. It has to be kept in mind that this is only the reported layoffs, the actual layoffs might go way beyond these numbers.

Ranking Companies: Using the DENSE_RANK() function, companies were ranked based on yearly layoffs. Ranking the companies in this way provides a clear understanding of which companies were most affected by layoffs on an annual basis.

```
WITH Company_Year (Company, Years, total_laid_off) AS (
  SELECT company, YEAR(`date`) AS years, SUM(total_laid_off) AS total_laid_off
  FROM layoffs_new 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 -- max 3 ranks
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
Cisco	2022	4100	3
Google	2023	12000	1
Microsoft	2023	10000	2
Ericsson	2023	8500	3

Conclusion…

In this project, we conducted a comprehensive analysis of layoffs data, beginning with a thorough data cleaning process. The cleaning steps involved identifying and removing duplicate records, standardizing inconsistent values, and handling missing data, all of which ensured that the dataset was accurate and ready for analysis. By cleaning the data, we were able to uncover reliable insights and trends that would otherwise be skewed by errors and inconsistencies.

The findings from this analysis hold significant importance for businesses, policymakers, and stakeholders. Identifying key metrics such as maximum layoffs, geographic and industry-based trends, and the impact on well-funded companies provides valuable insights into the dynamics of layoffs. These trends help organizations better understand economic conditions, industry vulnerabilities, and the broader impact on the workforce. This information can assist businesses in strategic planning, help policymakers in formulating protective measures, and aid stakeholders in making informed decisions.

For future analysis, further exploration could involve incorporating additional data sources such as economic indicators or unemployment rates to deepen the understanding of layoff trends. Additionally, more detailed analysis of company-specific factors like leadership changes or market shifts could provide even greater insights into the causes and consequences of layoffs. By expanding the dataset and continuing to refine the analysis, we can provide an even more nuanced view of workforce trends in different sectors and regions.

This project has demonstrated the value of data cleaning and exploration in producing actionable insights, offering a solid foundation for future research and analysis in the field of workforce dynamics.