WORLD LAYOFF ANALYSIS

BY:

Abdulrahman Hamzat

TABLE OF CONTENT

SECTION ONE

OVERVIEW

1.1 The Project

The aim of this project is to perform data wrangling and some exploratory data analysis (EDA) on the "World Layoff" dataset in MySQL Workbench. The entire project was completed using MySQL only.

This project was achieved during my participation in the Data Analyst Bootcamp created by "Alex The Analyst".

1.2 The Data

The "world layoff" dataset primarily consists of total number of employees that were laid off in various companies located in various cities across different countries. The data covered three years (2020 - 2023). It consists of over two thousand rows with nine columns namely: "company"; "location"; "industry"; "total_laid_off"; "percentage_laid_off"; "date"; "stage"; "country"; and "funds_raised_millions". It's available for download as a CSV file here.

SECTION TWO

DATA GATHERING AND ASSESSMENT

A new table ("layoffs") was created in MySQL by importing the raw CSV file. After an initial assessment of the data as shown in Fig 2.1, a copy of the table was created [Fig 2.2] and populated [Fig 2.3] with the original data before further assessment to check the data quality and structure.

```
1     -- initial assessment
2 •   SELECT *
3     FROM world_layoffs.layoffs;
4
```

Fig 2.1

```
10    -- Make a copy of the data before cleaning
11 •  CREATE TABLE layoffs_staging
12    LIKE layoffs;
13
```

Fig 2.2

```
18    -- Populate the empty table with the original data
19 •  INSERT layoffs_staging
20    SELECT *
21    FROM layoffs;
--
```

Fig 2.3

Next, the SQL table was assessed visually and programmatically to check for data quality and structural issues. Table 2.1 below gives a documentation of the data quality issues detected.

Table 2.1

S/N	Query/Mode of Assessment	Issue Detected	Quality Dimension
1	Created a row count in a Common Table Expression (CTE), and selected data where row number is greater than 1 in the main query. [Fig 2.4]	Five duplicated rows.	Non-Uniqueness

2	Visual assessment	White spaces before the names in the column `company`	Invalidity
3	Queried the distinct values in the column `industry`. [Fig 2.5]	Different word representations for the value 'Cryptocurrency'.	Inconsistency
4	Queried the distinct values in the column `industry`. [Fig 2.5]	Blank values or "Null" for missing data.	Inconsistency
5	Queried the distinct values in the column `company`. [Fig 2.6]	A period (.) after the value 'United States' in some cases.	Invalidity
6	Visual assessment	Column `date` not in standard date format	Invalidity
7	Visual assessment	Missing Information (e.g empty spaces or nulls).	Incompleteness

```
29    -- Check for duplicate rows
30    WITH duplicate_cte AS
31    (
32    SELECT *,
33    ROW_NUMBER() OVER(PARTITION BY company, location, industry,
34                        total_laid_off, percentage_laid_off, `date`,
35                        stage, country, funds_raised_millions) AS row_num
36    FROM layoffs_staging
37    )
38    SELECT *
39    FROM duplicate_cte
40    WHERE row_num > 1;
41
```

Fig 2.4

```sql
SELECT DISTINCT industry
FROM layoffs_staging
ORDER BY 1;
```

Fig 2.5

```sql
SELECT DISTINCT company
FROM layoffs_staging
ORDER BY 1;
```

Fig 2.6

SECTION THREE

DATA CLEANING

Before the cleaning process commenced, a new copy of the data was made by creating an empty table [Fig 3.1]. The empty table was populated with the original data and a new column was added to the table for row count [Fig 3.2]. All the previously documented quality issues were then treated one after the other.

```
47    -- Create an empty table `layoffs_staging2`
48    CREATE TABLE `layoffs_staging2` (
49      `company` text,
50      `location` text,
51      `industry` text,
52      `total_laid_off` int DEFAULT NULL,
53      `percentage_laid_off` text,
54      `date` text,
55      `stage` text,
56      `country` text,
57      `funds_raised_millions` int DEFAULT NULL,
58      `row_num` INT
59    ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
60
```

Fig 3.1

```
61    -- Populate the empty table with data and a new column for row count
62    INSERT layoffs_staging2
63    SELECT *,
64    ROW_NUMBER() OVER(PARTITION BY company, location, industry,
65                      total_laid_off, percentage_laid_off, `date`,
66                      stage, country, funds_raised_millions) AS row_num
67    FROM layoffs_staging;
68
```

Fig 3.2

3.1 Issue #1

Five duplicated rows.

3.1.1 Query

Delete all rows where the corresponding values in the `row_num` column is greater than 1 using the keyword “DELETE” and the “WHERE” clause [Fig 3.3].

```
74      -- delete the duplicated rows
75 •    DELETE
76      FROM layoffs_staging2
77      WHERE row_num > 1;
```

Fig 3.3

3.2 Issue #2

White spaces before the names in the column `company`.

3.2.1 Query

Update the table and trim all white spaces to the left of the names in the column `company` using the “UPDATE”, “SET”, and “TRIM” keywords [Fig 3.4].

```
86      -- trim white spaces
87 •    UPDATE layoffs_staging2
88      SET company = TRIM(company);
```

Fig 3.4

3.3 Issue #3

Different word representations for the name ‘Cryptocurrency’ in the `industry` column.

3.3.1 Query

Update the table and set the word “Crypto” where all similar representations of the word appear using keywords “UPDATE”, “SET”, “LIKE” and the “WHERE” clause [Fig 3.5].

```
103     -- standardize all representations with "Crypto"
104 •   UPDATE layoffs_staging2
105     SET industry = 'Crypto'
106     WHERE industry like "Crypto%";
107
```

Fig 3.5

3.4 Issue #4

Blank values or “Null” for missing entries in the `industry` column.

3.4.1 Query

Update and set all missing entries with blank values to “NULL” [Fig 3.6].

```
146     -- set all blank values to NULL
147 •   UPDATE layoffs_staging2
148     SET industry = NULL
149     WHERE industry = '';
150
```

Fig 3.6

3.5 Issue #5

A period (.) after the value ‘United States’ in some cases within the `company` column.

3.5.1 Query

Update and remove the period symbol using the “TRIM” function and the “TRAILING” keyword where all values like “United States” [Fig 3.7].

```
124     -- remove the period symbol
125 •   UPDATE layoffs_staging2
126     SET country = TRIM(TRAILING '.' FROM country)
127     WHERE country LIKE 'United States%';
128
```

Fig 3.7

3.6 Issue #6

Column `date` not in standard date format

3.6.1 Query

Update and convert the `date` column into MySQL’s date format [Fig 3.8], then modify the column to “date” data-type [Fig 3.9].

```
133     -- set to date format
134 •   UPDATE layoffs_staging2
135     SET `date` = STR_TO_DATE(`date`, '%m/%d/%Y');
136
```

Fig 3.8

```
137     -- set the data type to date
138 •   ALTER TABLE layoffs_staging2
139     MODIFY COLUMN `date` DATE;
140
```

Fig 3.9

3.7 Issue #7

Missing Information (e.g empty spaces or Nulls).

3.7.1 Query

Fill blank spaces and nulls in the column `industry` with a self-join and set `industry` name where it's null to where it isn't using `company` column for the "primary-foreign" key link [Fig 3.10].

```
151     -- fill blank spaces and Nulls
152 •   UPDATE layoffs_staging2 t1
153     JOIN layoffs_staging2 t2
154         ON t1.company = t2.company
155     SET t1.industry = t2.industry
156     WHERE t1.industry IS NULL
157     AND t2.industry IS NOT NULL;
158
```

Fig 3.10

3.7.2 Query

Delete rows with null entries for both `total_laid_off` and `percentage_laid_off` columns [Fig 3.11].

```
159     -- Delete rows with missing entries
160 •   DELETE
161     FROM layoffs_staging2
162     WHERE total_laid_off IS NULL
163     AND percentage_laid_off IS NULL;
164
```

Fig 3.11

Finally, the `row_num` column which was created during the assessment stage was dropped from the table [Fig 3.12]. Next is EDA.

```
165     -- Delete the row_num column
166 •   ALTER TABLE layoffs_staging2
167     DROP COLUMN row_num;
168
```

Fig 3.12

SECTION FOUR

EXPLORATORY DATA ANALYSIS (EDA)

At this stage, research questions were framed, and the data was explored to extract ten meaningful and interesting insights therefrom.

4.1 Insight #1

The highest layoff and percentage layoff recorded.

4.1.1 Query

```
180 •   SELECT MAX(total_laid_off), MAX(percentage_laid_off)
181     FROM layoffs_staging2;
182
```

Fig 4.1

4.1.2 Result

The maximum layoff record in the data is 12,000 employees, while the maximum percentage of layoff is 100% (i.e. 1).

Fig 4.2

4.2 Insight #2

The top 5 dissolved or inactive companies with respect to total layoffs.

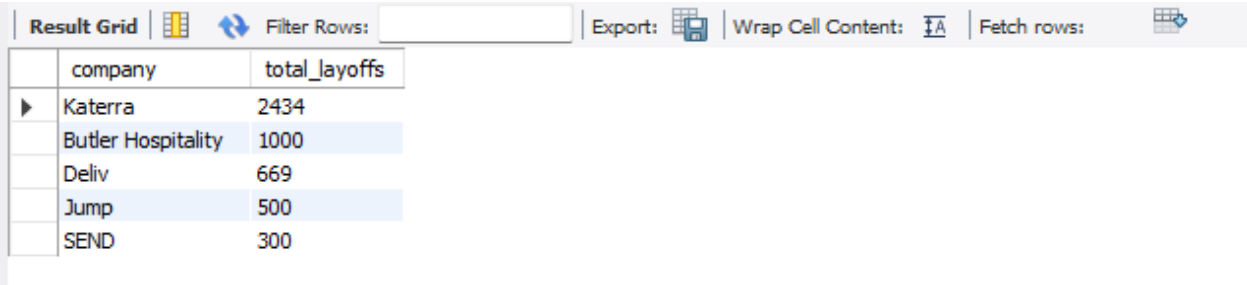
4.2.1 Query

```
191 •   SELECT company, SUM(total_laid_off) total_layoffs
192     FROM layoffs_staging2
193     WHERE percentage_laid_off = 1
194     GROUP BY company
195     ORDER BY 2 DESC
196     LIMIT 5;
```

Fig 4.3

4.2.2 Result

The first company on the list is “Katerra” laying off a total of 2,434 employees, and the fifth is “SEND” with a layoff total of 300 employees.



company	total_layoffs
Katerra	2434
Butler Hospitality	1000
Deliv	669
Jump	500
SEND	300

Fig 4.4

4.3 Insight #3

Dissolved or inactive companies with the most fundraise.

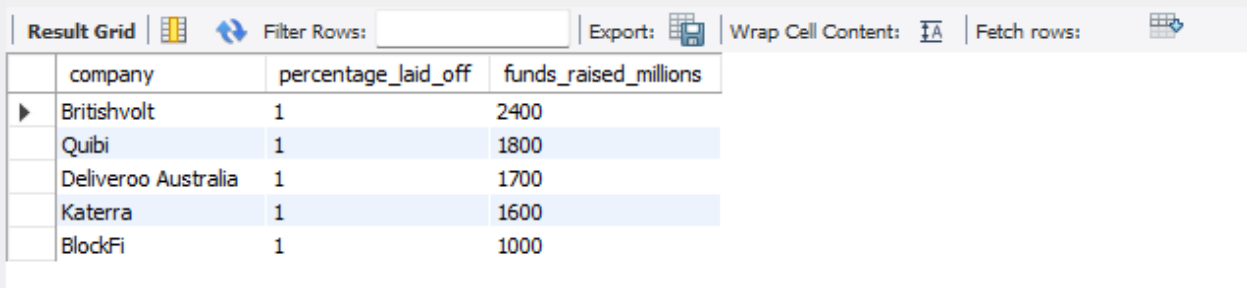
4.3.1 Query

```
198 •   SELECT company, percentage_laid_off, funds_raised_millions
199     FROM layoffs_staging2
200     WHERE percentage_laid_off = 1
201     ORDER BY 3 DESC
202     LIMIT 5;
```

Fig 4.5

4.3.2 Result

"Britishvolt" raised the most funds of about 2.4 billion USD, followed by "Quibi" which raised 1.8 billion USD, then "Deliveroo Australia" with 1.7 billion USD, "Katerra" with 1.6 billion USD, and lastly "BlockFi" with 1 billion USD.



company	percentage_laid_off	funds_raised_millions
Britishvolt	1	2400
Quibi	1	1800
Deliveroo Australia	1	1700
Katerra	1	1600
BlockFi	1	1000

Fig 4.6

4.4 Insight #4

Companies and their respective total layoffs throughout the three-year period.

4.4.1 Query

```
204 •   SELECT company, SUM(total_laid_off)
205     FROM layoffs_staging2
206     GROUP BY company
207     ORDER BY 2 DESC;
208
```

Fig 4.7

4.4.2 Result

The large total layoffs shown in Fig 4.8 compared to the relatively smaller figures obtained in Fig 4.4 for the companies which have shut down implies that those companies were mostly startups unlike the giant companies in the result below. Also, it's interesting to note that while "Amazon" topped the list below, "Google" laid off more employees (12,000) at once than Amazon between 2020 to 2023 [Fig 4.2].

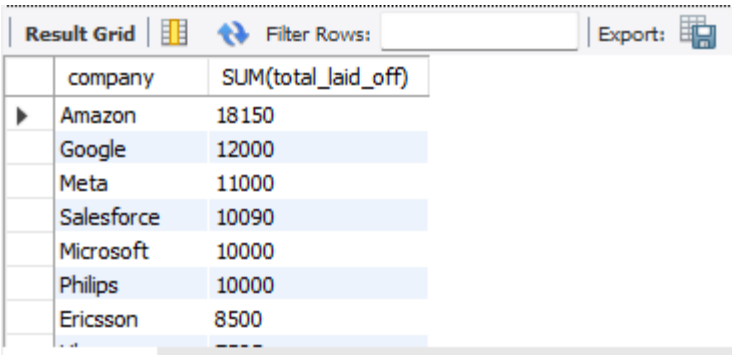


Fig 4.8

4.5 Insight #5

Start and end date of the collected data under analysis.

4.5.1 Query

```
209 •   SELECT MIN(`date`) start_date, MAX(`date`) end_date
210     FROM layoffs_staging2;
211
```

Fig 4.9

4.5.2 Result

The data covered a three-year period from March 11, 2020 till March 06, 2023.

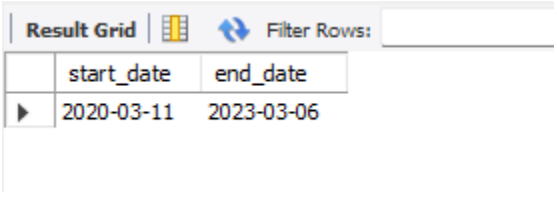


Fig 4.10

4.6 Insight #6

Total employees laid off across each of the industrial sectors.

4.6.1 Query

```
212 •   SELECT industry, SUM(total_laid_off)
213     FROM layoffs_staging2
214     GROUP BY industry
215     ORDER BY 2 DESC;
216
```

Fig 4.11

4.6.2 Result

Among the 31 industries found in the dataset, the following five emerged with the most prominent layoffs: Consumer, Retail, Other, Transportation, and Finance.

Result Grid | Filter Rows:

industry	SUM(total_laid_off)
Consumer	45182
Retail	43613
Other	36289
Transportation	33748
Finance	28344

Fig 4.12

4.7 Insight #7

Total employees laid off across each country.

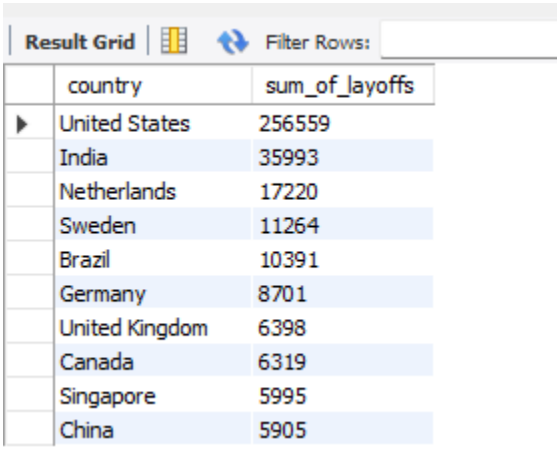
4.7.1 Query

```
218 •   SELECT country, SUM(total_laid_off) sum_of_layoffs
219     FROM layoffs_staging2
220     GROUP BY country
221     ORDER BY 2 DESC
222     LIMIT 10;
```

Fig 4.13

4.7.2 Result

The United States topped the list with over 250,000 layoffs in total within a three-year span, while China ranked the 10th among the rest of the 51 countries in the dataset.



Result Grid | Filter Rows:

country	sum_of_layoffs
United States	256559
India	35993
Netherlands	17220
Sweden	11264
Brazil	10391
Germany	8701
United Kingdom	6398
Canada	6319
Singapore	5995
China	5905

Fig 4.14

4.8 Insight #8

Companies' stages with the highest layoffs.

4.8.1 Query

```
230 •   SELECT stage, SUM(total_laid_off) total_layoffs
231     FROM layoffs_staging2
232     GROUP BY stage
233     ORDER BY 2 DESC
234     LIMIT 5;
235
```

Fig 4.15

4.8.2 Result

The bulk of the layoffs were from companies at the post-IPO stage.

Result Grid | Filter Rows:

stage	total_layoffs
Post-IPO	204132
Unknown	40716
Acquired	27576
Series C	20017
Series D	19225

Fig 4.16

4.9 Insight #9

Total layoff monthly and yearly

4.9.1 Query

```
WITH rolling_total AS (
SELECT SUBSTRING(`date`,1,4) AS `year`, SUBSTRING(`date`,1,7) AS `month`, SUM(total_laid_off) total_layoffs
FROM layoffs_staging2
WHERE SUBSTRING(`date`,1,7) IS NOT NULL
GROUP BY `year`,`month`
ORDER BY 2 ASC
)
SELECT `MONTH`, total_layoffs , SUM(total_layoffs) OVER(PARTITION BY `year` ORDER BY `MONTH`) rolled_total
FROM rolling_total;
```

Fig 4.17

4.9.2 Result

From the dataset, the year 2022 had the worst layoff record while year 2021 had the least. Interestingly, the number of employees laid off in the first 3 months of the year 2023 was about 80% of the total layoffs in the entire 12 months of year 2022.

Result Grid | Filter Rows: | Export:

MONTH	total_layoffs	rolled_total
2020-09	609	79459
2020-10	450	79909
2020-11	237	80146
2020-12	852	80998
2021-01	6813	6813
2021-02	868	7681
2021-03	47	7728
2021-04	261	7989

Fig 4.18

Result Grid | Filter Rows:

year	total_layoffs
2022	160661
2023	125677
2020	80998
2021	15823

Fig 4.19

4.10 Insight #10

The top 5 companies each year with the most layoff.

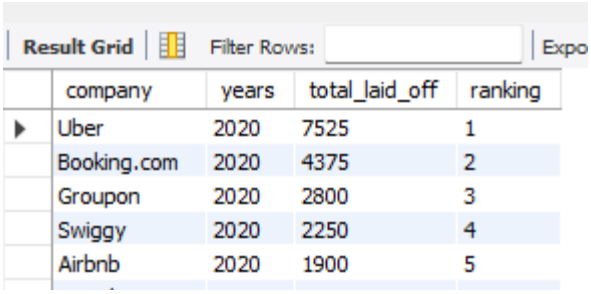
4.10.1 Query

```
265 •   WITH Company_Year (company, years, total_laid_off) AS
266     (SELECT company, YEAR(`date`), SUM(total_laid_off)
267     FROM layoffs_staging2
268     GROUP BY company, YEAR(`date`)
269     ), Company_Year_Rank AS
270     (SELECT *, DENSE_RANK() OVER (PARTITION BY years ORDER BY total_laid_off DESC) AS ranking
271     FROM Company_Year
272     WHERE years IS NOT NULL
273     )
274     SELECT *
275     FROM Company_Year_Rank
276     WHERE ranking <= 5;
```

Fig 4.20

4.10.2 Result

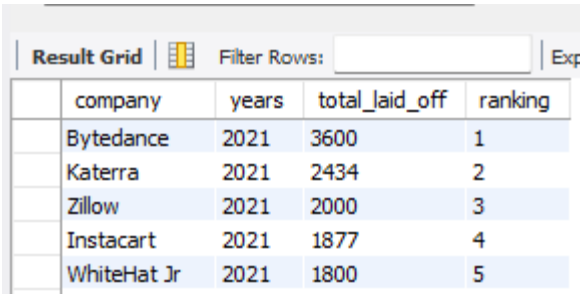
The following companies: Uber, Bytedance, Meta and Google laid off the most employees for the year 2020, 2021, 2022, and 2023 respectively.



Result Grid | Filter Rows: | Expo

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

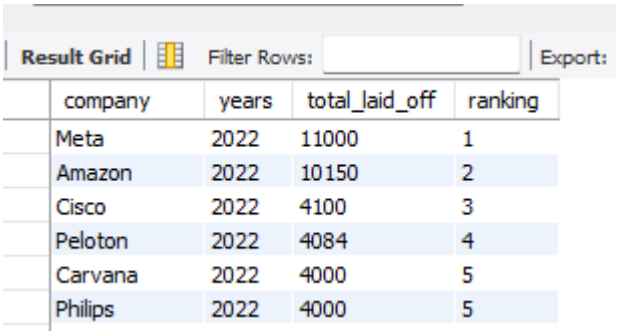
Fig 4.21



Result Grid | Filter Rows: | Exp

company	years	total_laid_off	ranking
Bytedance	2021	3600	1
Katerra	2021	2434	2
Zillow	2021	2000	3
Instacart	2021	1877	4
WhiteHat Jr	2021	1800	5

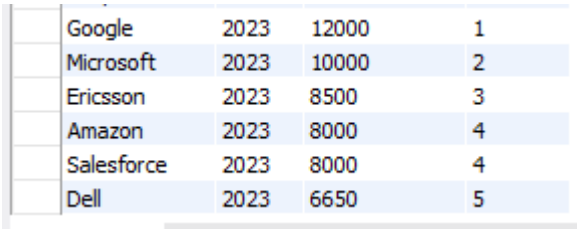
Fig 4.22



Result Grid | Filter Rows: | Export:

company	years	total_laid_off	ranking
Meta	2022	11000	1
Amazon	2022	10150	2
Cisco	2022	4100	3
Peloton	2022	4084	4
Carvana	2022	4000	5
Philips	2022	4000	5

Fig 4.23



Google	2023	12000	1
Microsoft	2023	10000	2
Ericsson	2023	8500	3
Amazon	2023	8000	4
Salesforce	2023	8000	4
Dell	2023	6650	5

Fig 4.24

SECTION FIVE

SUMMARY OF INSIGHTS

The exploratory data analysis of the "World Layoff" dataset revealed significant trends and insights regarding layoff across various companies, industries, and countries.

The largest recorded layoff in the dataset involved 12,000 employees, and a maximum layoff percentage reaching 100% which indicates the complete shutdown of some companies. A closer analysis of those companies highlighted Katerra leading with 2,434 layoffs. Interestingly, Britishvolt, which raised the most funds ($2.4 billion), and Quibi ($1.8 billion) also experienced 100% layoff. This suggests that even well-funded companies were not immune to economic downturns or some other internal challenges.

The dataset spanned from March 2020 to March 2023. Over this three-year period, large tech giants such as Amazon, Google, Meta, Salesforce and Microsoft stood out with major layoffs. In terms of industries, Consumer, Retail, Transportation and Finance were the sectors hit hardest by layoffs, while geographically, the United States had the most layoffs, exceeding 250,000.

Furthermore, post-IPO companies experienced the bulk of layoffs likely due to the pressure on public companies to manage investor expectations. Yearly trends showed that 2022 had the worst layoff record, although the first three months of 2023 accounted for a staggering amount (approx. 80%) of the total layoffs in 2022, signaling an even more severe downturn.