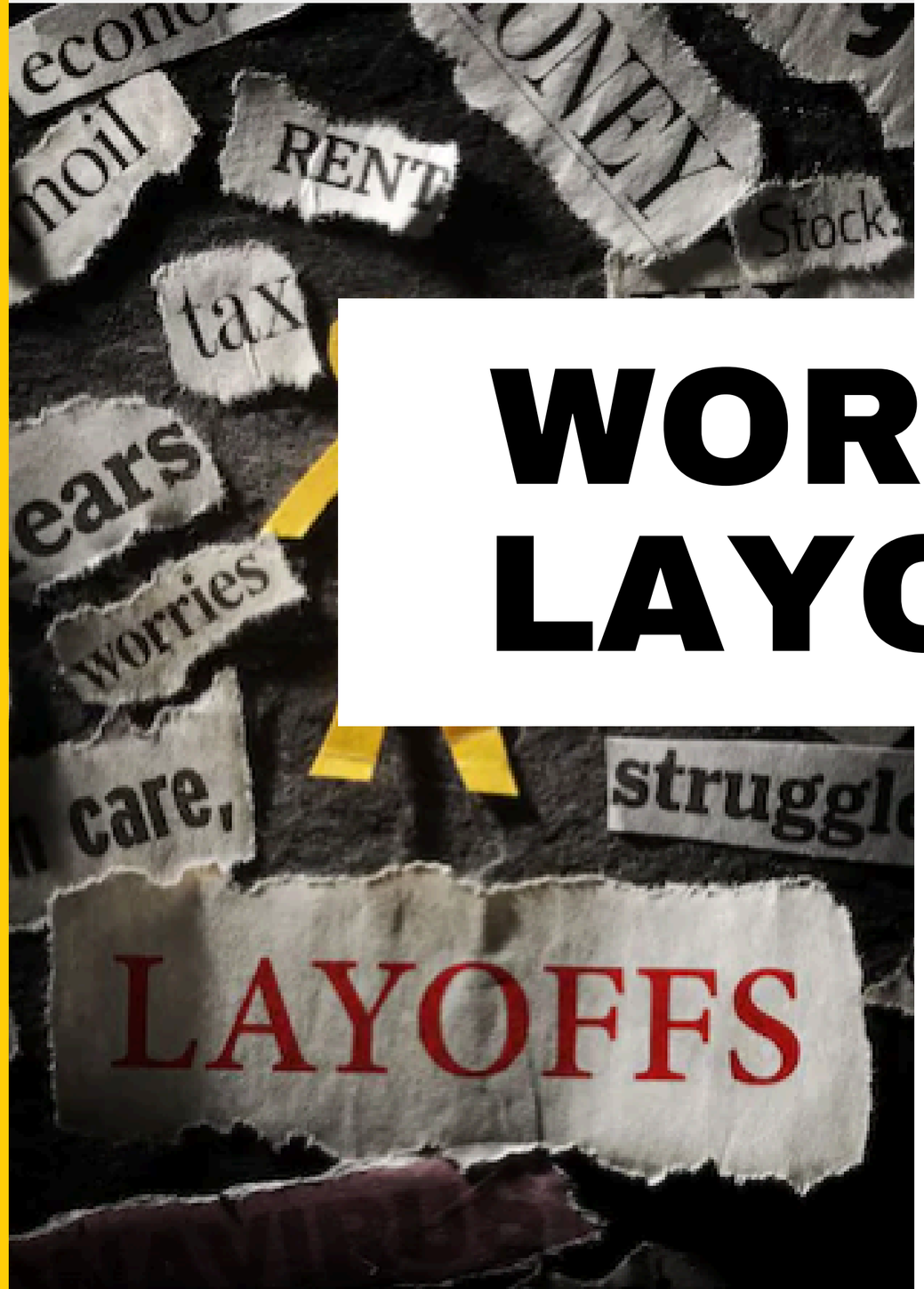
ABHISHEK VERMA



WORLD'S LAYOFFS

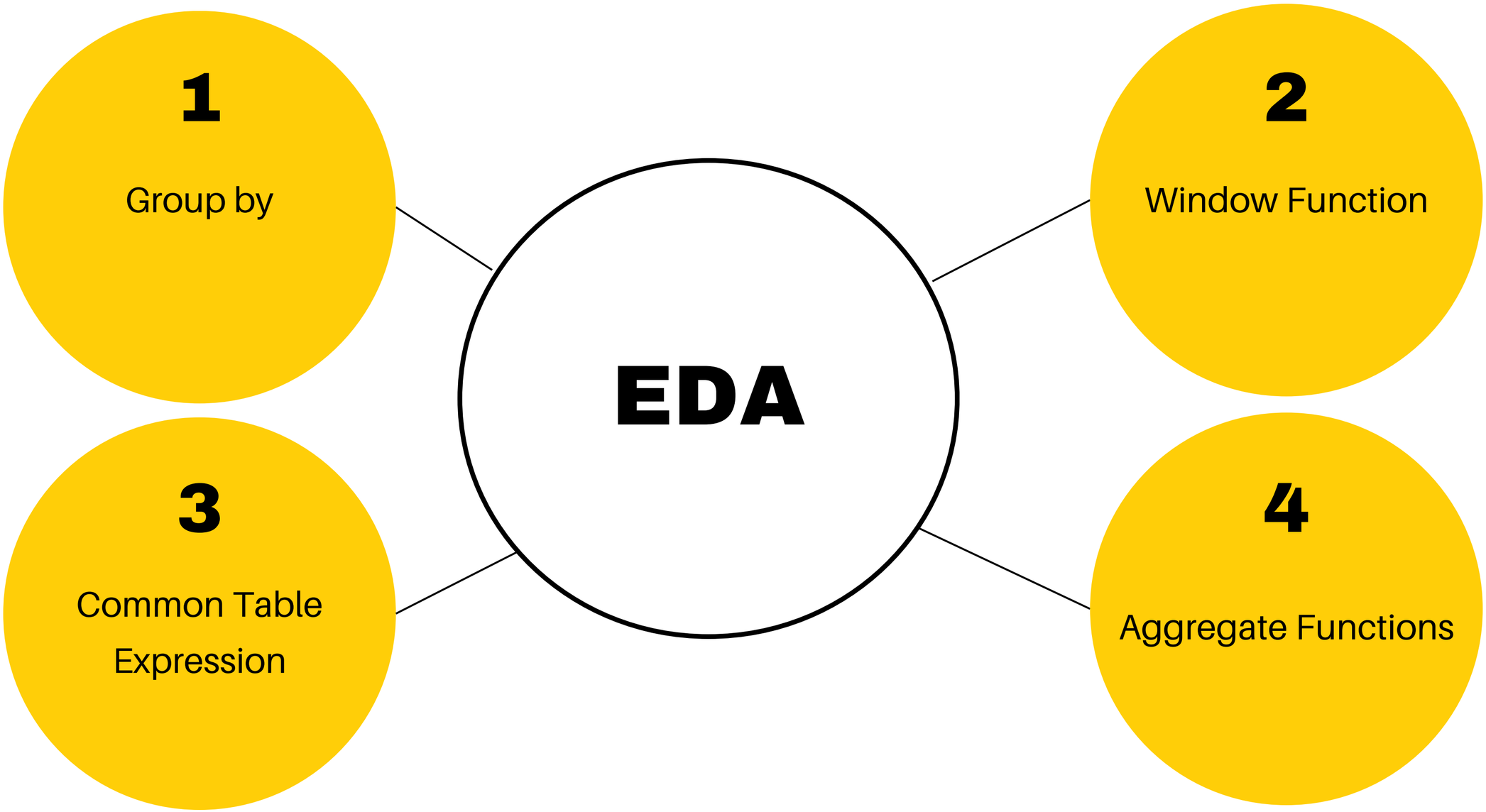
Github : https://github.com/abhishekvermacu20/World-s-Layoffs

Portfolio website

Table of Content

ABOUT PROJECT

In this project, we will transition from data cleaning to exploratory data analysis (EDA) using MySQL. EDA is the process of analyzing datasets to summarize their main characteristics, often employing visual methods. This phase involves identifying patterns, spotting anomalies, testing hypotheses, and checking assumptions with the aid of summary statistics and graphical representations. By systematically exploring the dataset, we can uncover insights that inform subsequent stages of data processing and analysis. This project will enable us to generate actionable insights and ensure that our data is well-understood before making data-driven decisions.


1
Group by
2
Window Function
EDA
3
Common Table Expression
4
Aggregate Functions

Our Dataset

Layoffs Dataset 2024

This dataset was scraped from Layoffs.fyi with the hope to enable Kaggle community to look into analyzing recent mass layoffs and discover useful insights and patterns.

Link : https://www.kaggle.com/datasets/theakhilb/layoffs-data-2022/datas.

Queries

1. **Maximum Layoffs and Layoff Percentage,**

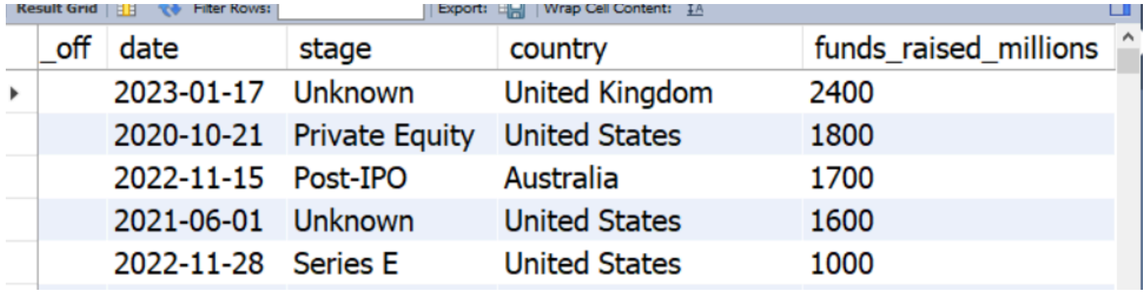
```
6    -- Maximum laid off and percentage laid off
7 •  select max(total_laid_off), max(percentage_laid_off)
8    from layoffs_staging2;
9
```

max(total_laid_off)	max(percentage_laid_off)
12000	1

Queries

```
11 •  select *
12    from layoffs_staging2
13    where percentage_laid_off = 1
14    order by total_laid_off desc;
15    -- order by funds_raised_millions
16 •  select *
17    from layoffs_staging2
18    where percentage_laid_off = 1
19    order by funds_raised_millions desc;
```

2. Maximum Layoffs and 100% Layoff Percentage, Ordered by Funds Raised (in Millions)



_off	date	stage	country	funds_raised_millions
	2023-01-17	Unknown	United Kingdom	2400
	2020-10-21	Private Equity	United States	1800
	2022-11-15	Post-IPO	Australia	1700
	2021-06-01	Unknown	United States	1600
	2022-11-28	Series E	United States	1000

Queries

3. Top 10 Companies with the Highest Number of Layoffs Over

```
22 •  select company, sum(total_laid_off) 'Number of employees
23     laid off'
24    from layoffs_staging2
25    group by company
26    order by sum(total_laid_off) desc
27    limit 10;
```

company	Number of employees laid off
Amazon	18150
Google	12000
Meta	11000
Salesforce	10090
Microsoft	10000
Philips	10000
Ericsson	8500
Uber	7585
Dell	6650
Booking.com	4601

Queries

4. Starting and ending date.

```
29    -- Total number of years.
30 •  select max(`date`), min(`date`)
31    from layoffs_staging2;
32
```

Result Grid | Filter Rows: | Export:

	max(`date`)	min(`date`)
▸	2023-03-06	2020-03-11

Queries

5. Top 10 Industries with the Highest Number of Layoffs Over

```
32    -- Top 10 industries laid off maximum number of employees
33 •  select industry, sum(total_laid_off)
34    'Number of employees laid off'
35    from layoffs_staging2
36    group by 1
37    order by 2 desc
38    limit 10;
39
```

industry	Number of employees laid off
Consumer	45182
Retail	43613
Other	36289
Transportation	33748
Finance	28344
Healthcare	25953
Food	22855
Real Estate	17565
Travel	17159
Hardware	13828

Queries

6. Years with the Highest Number of Layoffs

```
48 •  select year(`date`), sum(total_laid_off) AS
49    'Number of employees laid off'
50    from layoffs_staging2
51    group by 1
52    order by 2 desc
53    limit 10;
```

year(`date`)	Number of employees laid off
2022	160661
2023	125677
2020	80998
2021	15823
NULL	500

Queries

7. Stage with the Highest Number of Layoffs

```
select stage, sum(total_laid_off) AS
'Number of employees laid off'
from layoffs_staging2
group by 1
order by 2 desc;
```

Post-IPO	204132
Unknown	40716
Acquired	27576
Series C	20017
Series D	19225
Series B	15311
Series E	12697
Series F	9932

Queries

```
58    -- now use it in a CTE so we can query off of it
59 •  WITH DATE_CTE AS
60    (
71    SELECT SUBSTRING(date,1,7) as dates,
72    SUM(total_laid_off) AS total_laid_off
73    FROM layoffs_staging2
74    GROUP BY dates
75    ORDER BY dates ASC
76    )
77    SELECT dates, SUM(total_laid_off)
78    OVER (ORDER BY dates ASC) as rolling_total_layoffs
79    FROM DATE_CTE
80    ORDER BY dates ASC;
```

8. Rolling Total of Layoffs Per Month

dates	rolling_total_layoffs
NULL	500
2020-03	10128
2020-04	36838
2020-05	62642
2020-06	70269
2020-07	77381
2020-08	79350
2020-09	79959

Queries

```
85 •  WITH Company_Year AS
86    (
87      SELECT company, YEAR(date) AS years,
88      SUM(total_laid_off) AS total_laid_off
89      FROM layoffs_staging2
90      GROUP BY company, YEAR(date)
91    )
92    , Company_Year_Rank AS (
93      SELECT company, years, total_laid_off,
94      DENSE_RANK() OVER (PARTITION BY years ORDER BY total_laid_off DESC)
95      AS ranking
96      FROM Company_Year
97    )
98    SELECT company, years, total_laid_off, ranking
99    FROM Company_Year_Rank
100   WHERE ranking <= 3
101   AND years IS NOT NULL
102   ORDER BY years ASC, total_laid_off DESC;
```

9. Earlier we looked at Companies with the most Layoffs. Now let's look at that per year.

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

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

ETL - EXTRACT TRANSFORMATION LOAD

Let's Connect With Us!

abhishekvermacu20@gmail.com

www.linkedin.com