

SQL code for Exploratory Data Analysis (EDA) on the Cleaned Dataset:-

1. View All Data

```
Select *  
FROM layoffs_stagging2;
```

Step: Retrieve all records from `layoffs_stagging2`.

Purpose: Perform a general inspection of the entire dataset.

2. Find Maximum Values

```
Select max(total_laid_off), max(percentage_laid_off)  
FROM layoffs_stagging2;
```

Step: Find the maximum `total_laid_off` and `percentage_laid_off`.

Purpose: Identify the largest layoff event and the highest percentage laid off in the dataset.

3. Identify Companies with 100% Layoffs

```
Select *  
FROM layoffs_stagging2  
WHERE percentage_laid_off = 1  
ORDER BY funds_raised_millions DESC;
```

Step: Select records where the entire workforce was laid off (`percentage_laid_off = 1`), sorted by the funds raised.

Purpose: Analyze companies with 100% layoffs, ordered by their fundraising capabilities.

4. Total Layoffs by Company

```
Select company, sum(total_laid_off)
FROM layoffs_stagging2
GROUP BY company
ORDER BY 2 DESC;
```

Step: Group the data by company and calculate the total layoffs for each company.

Purpose: Find which companies laid off the most employees, ordered by the sum of layoffs.

5. Range of Dates

```
Select min(`date`), max(`date`)
FROM layoffs_stagging2;
```

Step: Find the minimum and maximum dates in the dataset.

Purpose: Determine the date range covered by the layoffs data.

6. Total Layoffs by Industry

```
Select industry, sum(total_laid_off)
FROM layoffs_stagging2
GROUP BY industry
ORDER BY 2 DESC;
```

Step: Group by `industry` and sum the `total_laid_off`.

Purpose: Analyze layoffs by industry to see which sectors were hit the hardest.

7. Total Layoffs by Country

```
Select country, sum(total_laid_off)
```

```
FROM layoffs_stagging2
```

```
GROUP BY country
```

```
ORDER BY 2 DESC;
```

Step: Group by `country` and sum the `total_laid_off`.

Purpose: Identify which countries had the most layoffs.

8. Yearly Layoffs

```
Select year(`date`), sum(total_laid_off)
```

```
FROM layoffs_stagging2
```

```
GROUP BY year(`date`)
```

```
ORDER BY 1 DESC;
```

Step: Group by year (extracted from the `date` column) and sum the layoffs.

Purpose: Observe layoff trends by year.

9. Monthly Layoffs

```
Select substring(`date`,1,7) as `month`, sum(total_laid_off)
```

```
FROM layoffs_stagging2
```

```
WHERE substring(`date`,1,7) IS NOT NULL
```

```
GROUP BY `month`
```

```
ORDER BY 1 ASC;
```

Step: Group by the first 7 characters of the `date` (year and month) and sum the `total_laid_off`.

Purpose: Analyze the total layoffs for each month.

10. Rolling Total Layoffs by Month

```
WITH Rolling_Total AS
(
  SELECT substring(`date`,1,7) as `month`, sum(total_laid_off) as total_off
  FROM layoffs_stagging2
  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;
```

Step: Compute the rolling total of layoffs for each month.

Purpose: Track the cumulative total layoffs month over month.

11. Layoffs by Company and Year

```
Select COMPANY, year(`date`), sum(total_laid_off)
FROM layoffs_stagging2
GROUP BY COMPANY, `date`
ORDER BY COMPANY ASC;
```

Step: Group by `COMPANY` and the `year` from `date`, and sum the layoffs.

Purpose: Analyze the total layoffs for each company year by year.

12. Top 5 Companies by Layoffs Each Year

```
WITH company_year AS (  
    SELECT COMPANY, year(`date`), sum(total_laid_off)  
    FROM layoffs_stagging2  
    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;
```

Step: First, calculate the total layoffs for each company by year, then rank them within each year using `dense_rank()`. Finally, select the top 5 companies per year.

Purpose: Identify the top 5 companies with the most layoffs each year.

Summary of Steps:

1. General Data Overview: Viewing all records for initial inspection.
2. Basic Aggregation: Maximum values, totals by company, country, industry.
3. Filter and Sort: Identifying specific cases (e.g., companies with 100% layoffs), sorting by relevant metrics.
4. Time-based Analysis: Aggregating by year, month, and calculating rolling totals.
5. Ranking and Advanced Analysis: Ranking companies based on layoffs and identifying top performers each year.

This set of SQL commands is used to explore and analyze layoffs data, with a focus on trends over time, company-specific impacts, and country/industry-level breakdowns.