Exploratory Data Analysis

Select * from layoff_stage2 where percentage_laid_off=1 order by total_laid_off desc;

Select * from layoff_stage2 where percentage_laid_off=1 order by funds_raised_millions desc;

select company, SUM(total_laid_off) from layoff_stage2 group by company order by 2 desc;

---- Time Range of Data ---

select min(`date`),max(`date`) from layoff_stage2;

--- Industry having most layoffs ----

Select industry, Sum(total_laid_off) from layoff_stage2

group by industry order by 2 DESC;

--- Country having most layoffs ----

Select country, Sum(total_laid_off) from layoff_stage2

group by country order by 2 DESC;

--- Year having most layoffs ---

Select Year(`date`), Sum(total_laid_off) from layoff_stage2

group by Year(`date`) order by 2 DESC;

--- Stage of the company having most layoffs----

Select stage, Sum(total_laid_off) from layoff_stage2

group by stage order by 2 DESC;

--- Company average percentage lay off ---

```sql
select company, AVG(percentage_laid_off) from layoff_stage2 group by company order
by 2 desc;
```

```sql
Select SUBSTRING(`date`,1,7) As `month`,SUM(total_laid_off)

from  layoff_stage2 where SUBSTRING(`date`,1,7) is not null group by `month`;
```

--- Rolling Sum of Month layoffs---

```sql
with rolling_sum as (

Select SUBSTRING(`date`,1,7) As `month`,SUM(total_laid_off) as monthly_sum

from  layoff_stage2 where SUBSTRING(`date`,1,7) is not null group by `month`)
```

```sql
Select `month`,monthly_sum, Sum(monthly_sum) OVER(order by `month`) from
rolling_sum ;
```

--- Company laying off per year---

```sql
select company, year(`date`),SUM(total_laid_off) from layoff_stage2

group by company,year(`date`)

having SUM(total_laid_off) is not null

order by 1 asc;
```

--- Rank each company on basis of layoff per year ----

```sql
with rank_company_on_layoffs as (

select company, year(`date`) as year_name,SUM(total_laid_off) as total from
layoff_stage2

where year(`date`) is not null

group by company,year(`date`)

having SUM(total_laid_off) is not null
```

```sql
order by 2 desc

),

company_ranking as(

select *, DENSE_Rank()

over(partition by year_name order by total desc) as ranking

from rank_company_on_layoffs)

select * from company_ranking where ranking<=5;
```