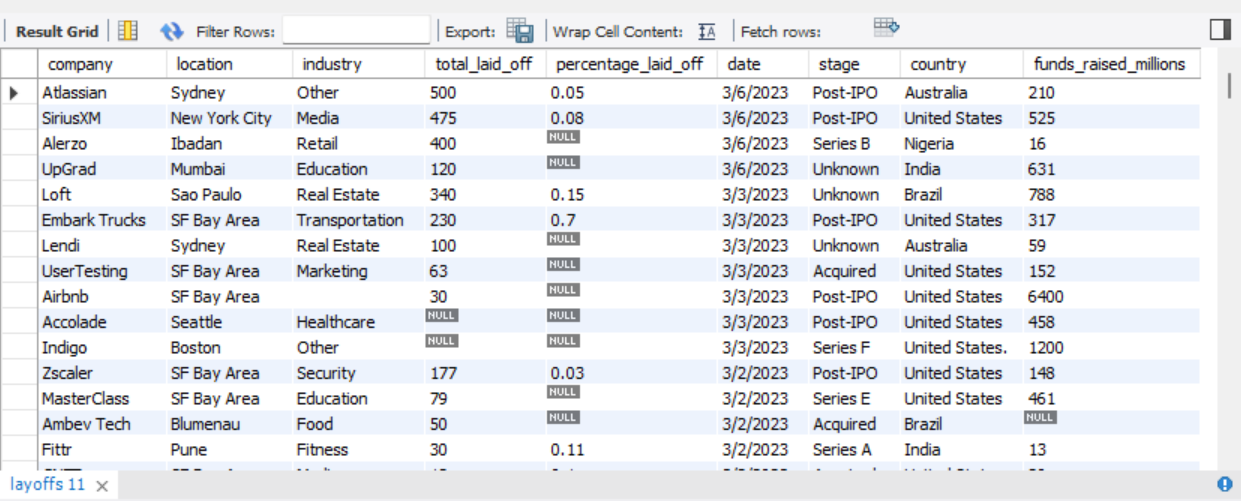
**MySQL QUERIES**

**A. EDA**

**1. Overview:**

select \* from layoffs;



**2. 100% laid-off companies by total laid off**

SELECT

\*

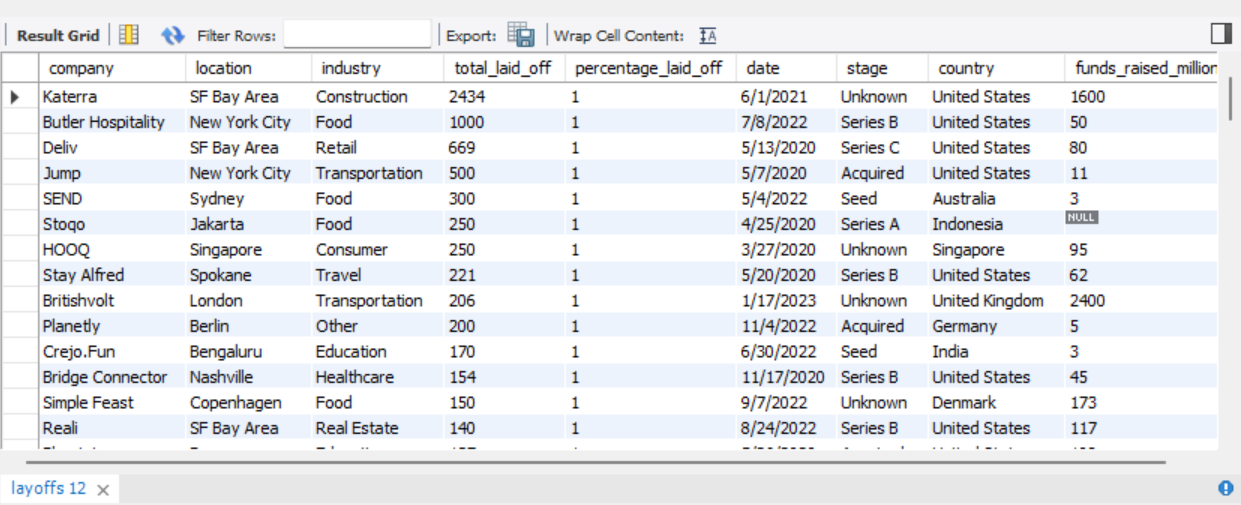
FROM

layoffs

WHERE

percentage\_laid\_off = 1

ORDER BY total\_laid\_off DESC;



**3. . 100% laid-off companies by highest funding**

SELECT

\*

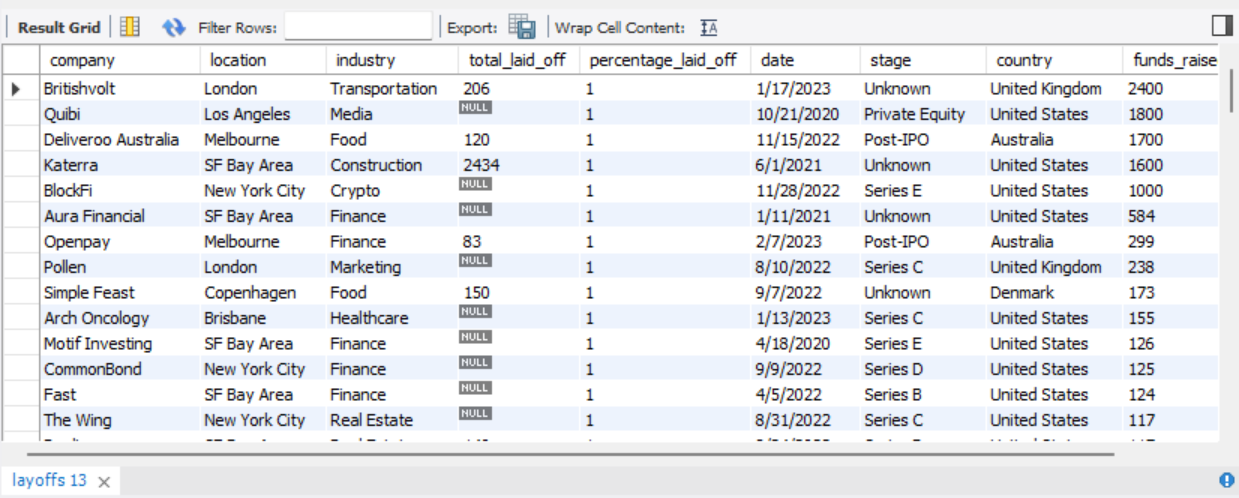
FROM

layoffs

WHERE

percentage\_laid\_off = 1

ORDER BY funds\_raised\_millions DESC**;**



**4. Total laid off by the company**

SELECT

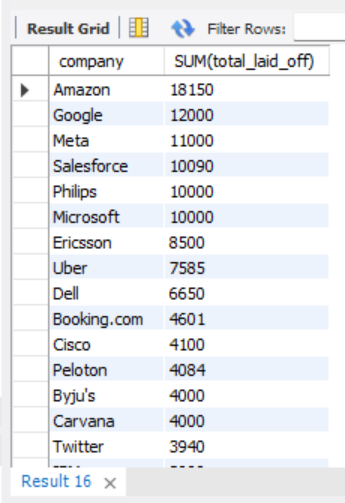
company, SUM(total\_laid\_off)

FROM

layoffs

GROUP BY company

ORDER BY 2 DESC;



**5. Find the duration of the data**

select min(new\_date), max(new\_date) from layoffs;

A screenshot of a computer

AI-generated content may be incorrect.

**6. Total laid off by the company**

SELECT

company, SUM(total\_laid\_off) as total\_laidoffs

FROM

layoffs

GROUP BY company

order by total\_laidoffs desc;

A screenshot of a computer

AI-generated content may be incorrect.

**7. Total layoffs by industry**

SELECT

industry, SUM(total\_laid\_off) as total\_laidoffs

FROM

layoffs

GROUP BY industry

order by total\_laidoffs desc;

A screenshot of a computer

AI-generated content may be incorrect.

**8. How many people lost their job by country**

SELECT

country, SUM(total\_laid\_off) as total\_laidoffs

FROM

layoffs

GROUP BY country

order by total\_laidoffs desc;

A screenshot of a computer

AI-generated content may be incorrect.

**9. Rolling total of laid off month by month**

with Rolling\_total as

(

SELECT SUBSTRING(new\_date, 1, 7) as `MONTH`, SUM(total\_laid\_off) as total\_offs

FROM layoffs

where SUBSTRING(new\_date, 1, 7) is not null

group by `MONTH`

order by 1 asc)

select`MONTH`, total\_offs, sum(total\_offs) over(order by `MONTH`) as rolling\_total

from Rolling\_total;

A screenshot of a computer

AI-generated content may be incorrect.

**5. Top 5 companies’ ranking by years with higher laid offs**

With Company\_Year (Company, Years, Total\_laid\_off) as

(

SELECT company, YEAR(`new\_date`), SUM(total\_laid\_off)

FROM layoffs

GROUP BY company, year(`new\_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;

