```sql
-- Exploratory Data Analysis --

select * from layoffs_copy2;

# looking at maximum total_laid_off and percentage_laid_off
select max(total_laid_off), max(percentage_laid_off)
from layoffs_copy2;

# taking a look at companies where percentage_laid_off = 1
select * from layoffs_copy2
where percentage_laid_off = 1
order by total_laid_off desc;

select * from layoffs_copy2
where percentage_laid_off = 1
order by funds_raised_millions desc;

# looking at the companies and their sum total_laid_off
select company ,sum(total_laid_off)
from layoffs_copy2
group by company
order by 2 desc;

--
select max(`date`) ,min(`date`)
from layoffs_copy2;

# looking at the industry and their sum total_laid_off
select industry ,sum(total_laid_off)
from layoffs_copy2
group by industry
order by 2 desc;

# looking at the countries and their sum total_laid_off
select country ,sum(total_laid_off)
from layoffs_copy2
group by country
order by 2 desc;

# looking at the year and their sum total_laid_off

select year(`date`) ,sum(total_laid_off)
from layoffs_copy2
group by year(`date`)
order by 1 desc;

#  looking at the stage and their sum total_laid_off
select stage ,sum(total_laid_off)
```

```sql
from layoffs_copy2
group by stage
order by 2 desc;

# Rolling total of total_laid_off
select substring(`date`,1,7) as `month`,
sum(total_laid_off)
from layoffs_copy2
where substring(`date`,1,7) is not null
group by `month`
order by 1 asc;

with rolling_total as
(
select substring(`date`,1,7) as `month`,
sum(total_laid_off) as total_off
from layoffs_copy2
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;

# Ranking companies on the basis of total_laid_off
select company ,year(`date`),sum(total_laid_off)
from layoffs_copy2
group by company, year(`date`)
order by 3 desc;

--

with company_year (company , years ,total_laid_off) as
(
select company ,year(`date`),sum(total_laid_off)
from layoffs_copy2
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;
```