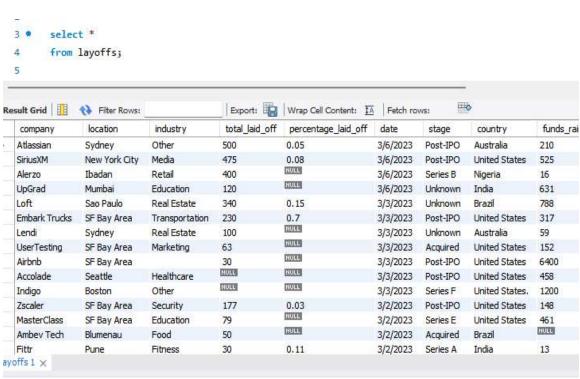
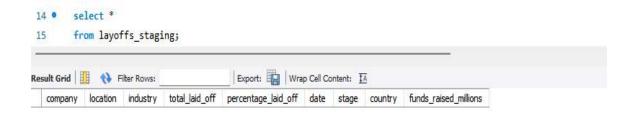
SQL Script With Result

1.



2.

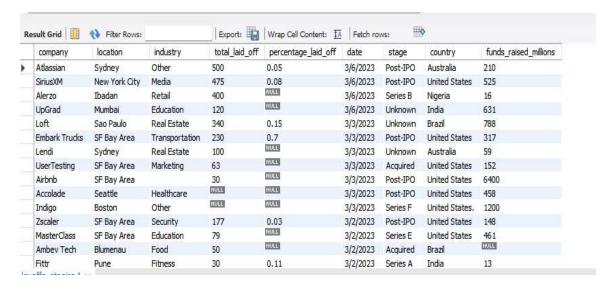
```
create table layoffs_staging
like layoffs;
```



```
ayoffs_staging3 x

4.
   insert layoffs_staging
   select *
   from layoffs;

5.
   select *
   from layoffs_staging;
```

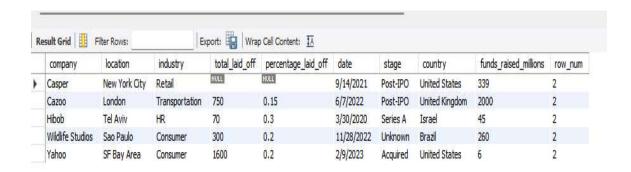


```
with duplicate_cte as

(
    select *,
    row_number()
    over(partition by company,location, industry, total_laid_off, percentage_laid_off, `date`, stage, country, funds_raised_million:
    from layoffs_staging
)
    select *
    from duplicate_cte
    where row_num>1;
```

In duplicate_CTE create a new column name row_num in which there is 1 for all unique rows and 2 for duplicate rows.

Then from duplicate_CTE display all those columns where row_num is greater than 1. (actually these are duplicate rows)



Created layoffs_staging2 to store cleaned data with row_num.

```
→ CREATE TABLE `layoffs_staging2` (
    'company' text,
    `location` text,
    'industry' text,
    `total_laid_off` int DEFAULT NULL,
    `percentage_laid_off` text,
    'date' text,
    `stage` text,
    `country` text,
    'funds_raised_millions' int DEFAULT NULL,
    'row num' int
  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

```
select *
from layoffs staging2;
```



```
insert into layoffs_staging2
select *,
row_number()
  over(partition by company,location, industry, total_laid_off, percentage_laid_off, `date`, stage, country, funds_raised_mill
from layoffs_staging;
```

Insert all data from layoffs_staging table to layoffs_staging2 along with row_num column as by using the window function.

10.

i. Removing Duplicates

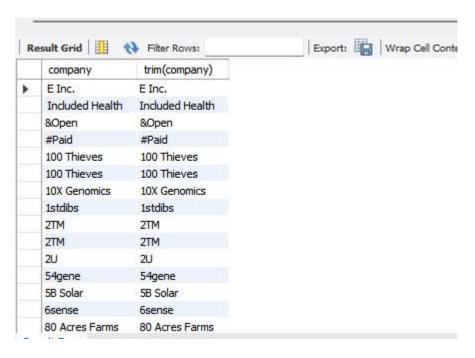
```
delete
from layoffs_staging2
where row_num>1;
```

ii. Standardizing Data

11.

```
select company, trim(company)
from layoffs_staging2;
```

As there are extra white spaces in the company column, trim function is used to remove these spaces.



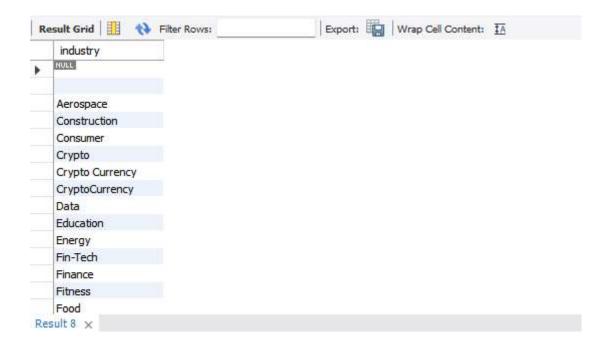
```
update layoffs_staging2
set company= trim(company);
```

This will upadate the company column by removing extra white spaces.

13.

Check industry column

```
select distinct(industry)
from layoffs_staging2
order by 1;
```



We need to standardized the Crypto or Crypto Currency to one standard name.

14.

```
update layoffs_staging2
set industry='Crypto'
where industry like 'Crypto%';
```

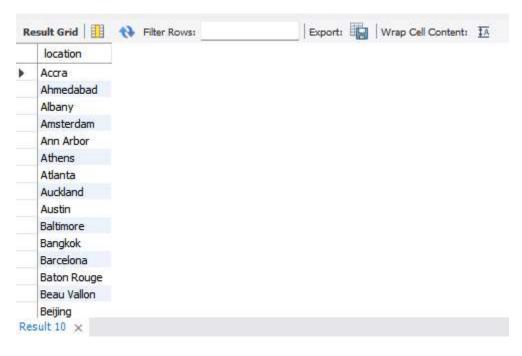
This will update industry column with single standard name Crypto for all where industry column is start with Crypto or any text.

15.

Check location column

```
select distinct(location)
from layoffs_staging2
order by 1;
```

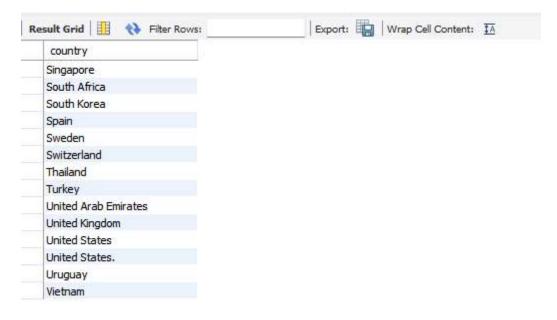
No blanks or duplicates and the data is in standard form.



```
select distinct(country)
from layoffs_staging2
order by 1;
```

check country column

United States is not in standard form.



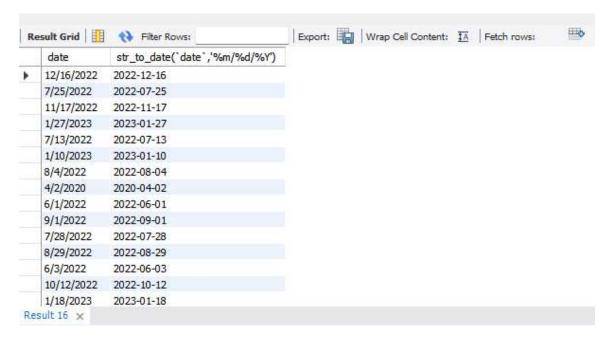
```
update layoffs_staging2
set country=trim(trailing '.' from country)
where country like 'United States%';
```

Here update the country column by triming the dot from united states and hence used a single standard name.

18.

```
select `date`,
str_to_date(`date`,'%m/%d/%Y')
from layoffs_staging2;
```

Convert date column to a standard date formate.



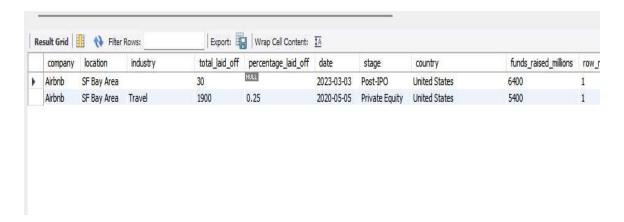
The date column have a text data type.

Convert it into date data type.

```
alter table layoffs_staging2
modify column `date` date;
```

iii.Removing NULL or Blank Values

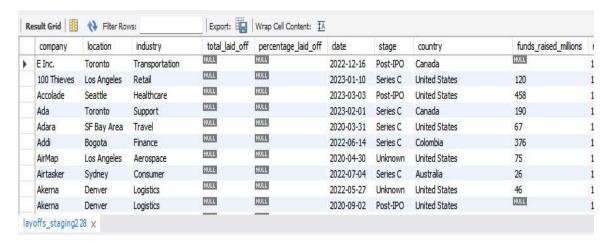
```
select *
from layoffs_staging2
where company='Airbnb';
```



```
update layoffs_staging2 as t1
join layoffs_staging2 as t2
    on t1.company=t2.company
set t1.industry=t2.industry
where (t1.industry is null or t1.industry = ' ')
and t2.industry is not null;
```

This query self join the layoffs_staging2 tables on company column and populate the null or blank industry column of t1 with the t2 industry column values but for that the company name must be the same.

```
select *
from layoffs_staging2
where total_laid_off is null
and
percentage_laid_off is null;
```



```
delete
from layoffs_staging2
where total_laid_off is null
and
percentage_laid_off is null;
```

iv. Removing Columns or Rows

24.

```
alter table layoffs_staging2
drop column row_num;
```

```
select *
from layoffs_staging2;
```

company	location	industry	total_laid_off	percentage_laid_off	date	stage	country	funds_raised_millions
BounceX	New York City	Marketing	77	0.2	2020-04-06	Series B	United States	75
Brainly	Krakow	Education	25	KULL	2022-11-06	Series D	Poland	148
Branch	New York City	Retail	3	0.27	2020-06-11	Seed	United States	2
Branch Metrics	SF Bay Area	Marketing	100	0.2	2020-04-07	Series E	United States	367
Brave Care	Portland	Healthcare	40	0.33	2022-09-06	Series B	United States	42
Breathe	New Delhi	Healthcare	50	0.33	2022-06-14	Series A	India	6
Breather	Montreal	Real Estate	120	0.8	2020-12-16	Series D	Canada	131
Brex	SF Bay Area	Finance	62	0.15	2020-05-29	Series C	United States	732
Brex	SF Bay Area	Finance	136	0.11	2022-10-11	Series D	United States	1500
Bridge Conn	Nashville	Healthcare	154	1	2020-11-17	Series B	United States	45
Bridgit	Waterloo	Construction	13	0.13	2022-12-06	Series B	Canada	36
Bright Machines	SF Bay Area	Data	30	0.08	2022-06-28	Unknown	Israel	250
Bright Money	Bengaluru	Finance	100	0.5	2022-07-15	Series A	India	31
Brighte	Sydney	Energy	58	HULL	2022-09-09	Series C	Australia	145
Brighte	Sydney	Energy	30	0.15	2022-06-20	Series C	Australia	145