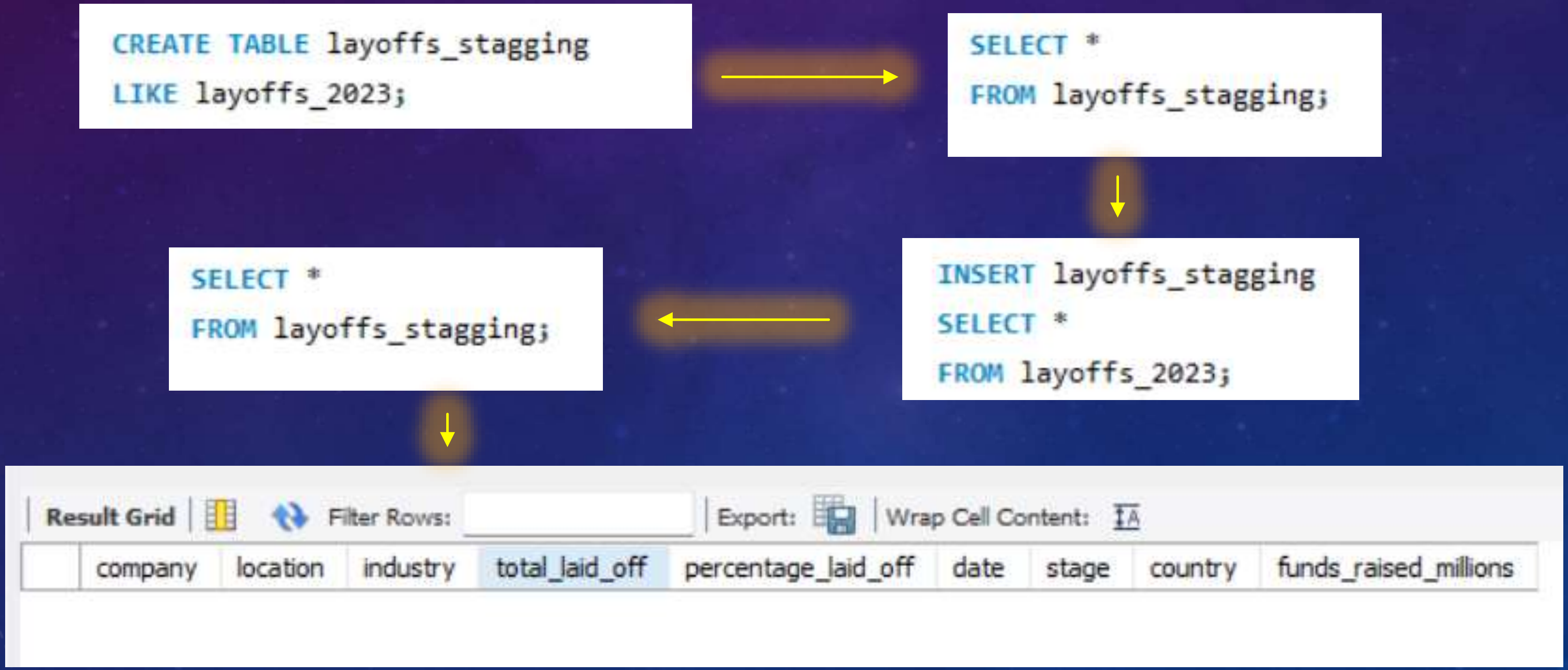
Layoffs 2020 - 2023 Cleaning & Analysis

K. NAGA PAVAN KUMAR

STEPS IN THE DATA CLEANING

- The data set consists of the layoffs information done upto 2023 by all the companies.
- The goal of the project is to clean and do data analysis in the data.
- The objectives of the cleaning data are
    1. Remove Duplicates
    2. Standardize the Data
    3. Null Values or blank values
    4. Remove Any Columns

- First to make sure the mistakes or the columns does not effect the raw data that is used elsewhere we need to secure the data by creating the duplicate of the data that can be done by creating an empty table.
- The empty table is created by name layoffs_stagging.
- Then all the data from layoffs_2023 is dumped into layoffs_stagging by using the given codes



```sql
CREATE TABLE layoffs_stagging
LIKE layoffs_2023;
```

```sql
SELECT *
FROM layoffs_stagging;
```

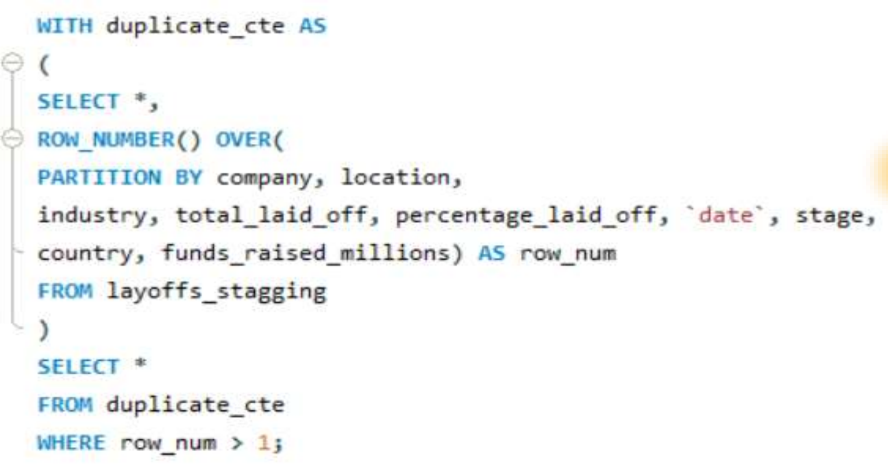
```sql
INSERT layoffs_stagging
SELECT *
FROM layoffs_2023;
```

```sql
SELECT *
FROM layoffs_stagging;
```

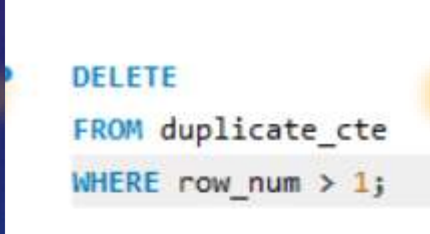
	company	location	industry	total_laid_off	percentage_laid_off	date	stage	country	funds_raised_millions

Trying to remove duplicates using CTE

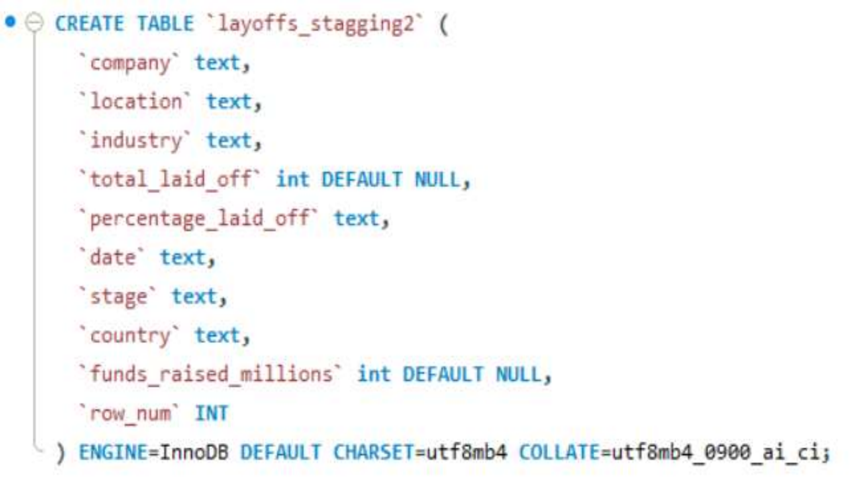
- First we created a instance of the lay offs table which can be used without effecting the raw data.
- then we need to know are there any duplicates in the data set with use of CTE.
- As we found duplicates tried to delete the duplicates using code but the error msg pooped up
- Error Code: 1288. The target table duplicate_cte of the DELETE is not updatable 0.000 sec
- So we can not delete the duplicates from the stagging in the CTE we will create our next staging table to filter and delete by using row number.
- Here we are adding the row_num as a extra row.



```sql
WITH duplicate_cte AS
(
SELECT *,
ROW_NUMBER() OVER(
PARTITION BY company, location,
industry, total_laid_off, percentage_laid_off, `date`, stage,
country, funds_raised_millions) AS row_num
FROM layoffs_stagging
)
SELECT *
FROM duplicate_cte
WHERE row_num > 1;
```

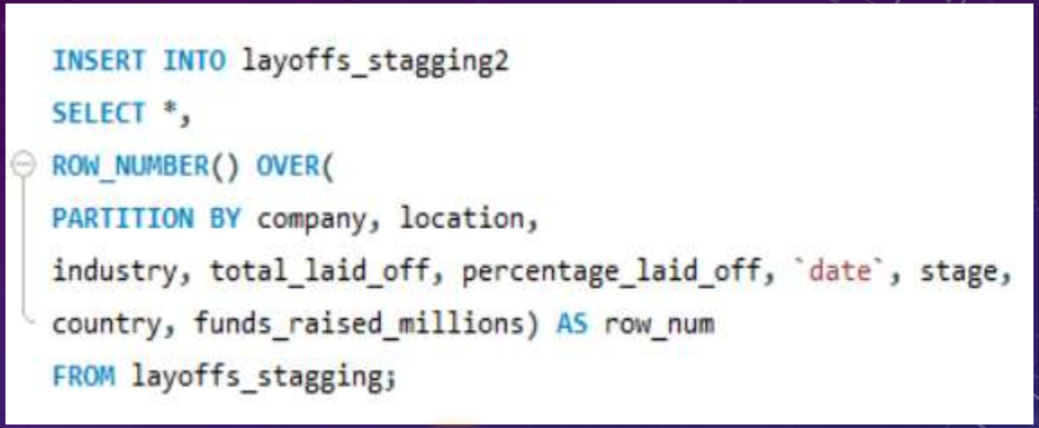


```sql
DELETE
FROM duplicate_cte
WHERE row_num > 1;
```

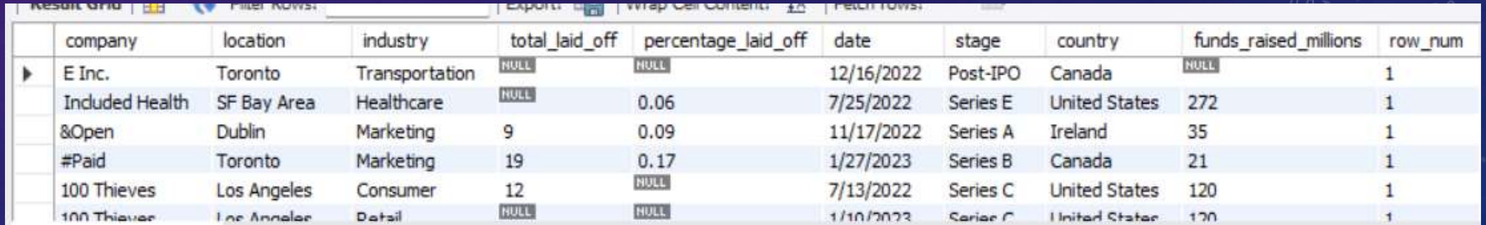


```sql
CREATE TABLE `layoffs_stagging2` (
  `company` text,
  `location` text,
  `industry` text,
  `total_laid_off` int DEFAULT NULL,
  `percentage_laid_off` text,
  `date` text,
  `stage` text,
  `country` text,
  `funds_raised_millions` int DEFAULT NULL,
  `row_num` INT
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

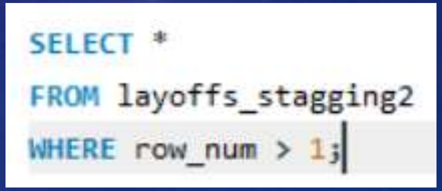
- So again a empty table is created and this time we will insert data from the layoffs_stagging.
- But as we known we have created row_num and need to assign the row number based on all the columns.
- As you can see new column has been added.
- Know we can easily delete the duplicates where row_num is greater than one using the delete code.



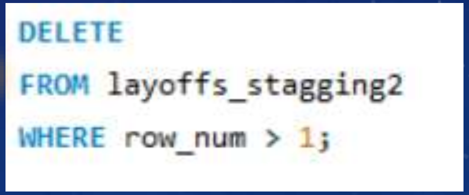
```
INSERT INTO layoffs_stagging2
SELECT *,
ROW_NUMBER() OVER(
PARTITION BY company, location,
industry, total_laid_off, percentage_laid_off, `date`, stage,
country, funds_raised_millions) AS row_num
FROM layoffs_stagging;
```



company	location	industry	total_laid_off	percentage_laid_off	date	stage	country	funds_raised_millions	row_num
E Inc.	Toronto	Transportation	NULL	NULL	12/16/2022	Post-IPO	Canada	NULL	1
Included Health	SF Bay Area	Healthcare	NULL	0.06	7/25/2022	Series E	United States	272	1
&Open	Dublin	Marketing	9	0.09	11/17/2022	Series A	Ireland	35	1
#Paid	Toronto	Marketing	19	0.17	1/27/2023	Series B	Canada	21	1
100 Thieves	Los Angeles	Consumer	12	NULL	7/13/2022	Series C	United States	120	1
100 Thieves	Los Angeles	Retail	NULL	NULL	1/10/2023	Series C	United States	120	1



```
SELECT *
FROM layoffs_stagging2
WHERE row_num > 1;
```

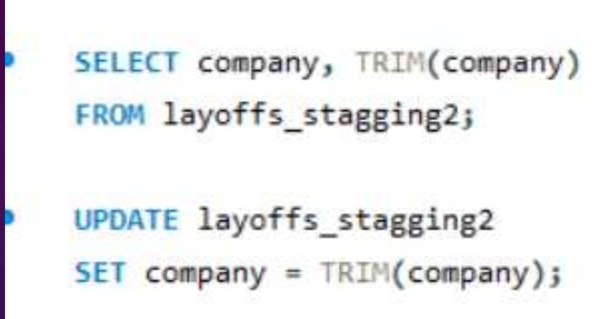


```
DELETE
FROM layoffs_stagging2
WHERE row_num > 1;
```

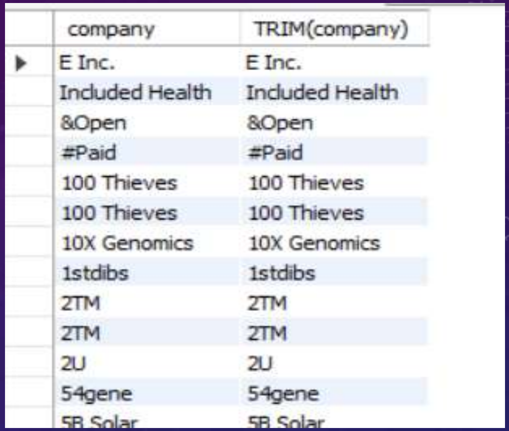
STANDARDIZING DATA

- In this part of the project the data is verified
- The spaces at the end of the data was found in company
- Multiple names were found for the same industry type (Crypto)
- And the trailing was found in the country column
- And for the time series analysis the date column was chaged to date format data type form the text data type.

Removing the extra spaces form the company column.

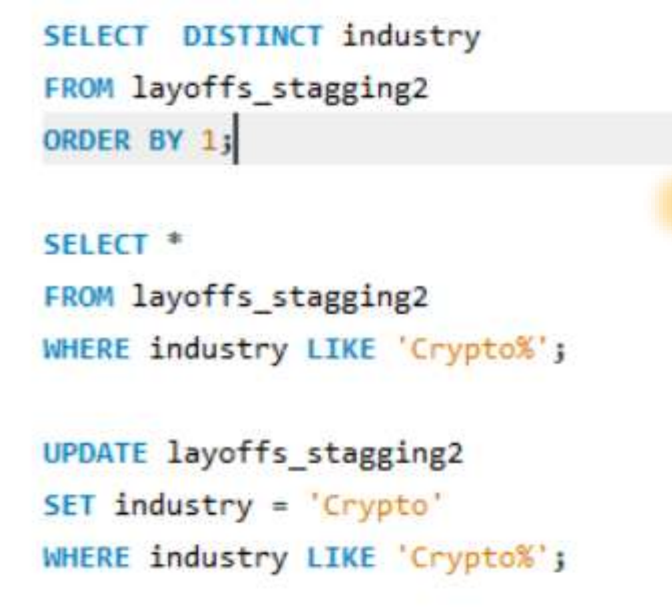


```
SELECT company, TRIM(company)
FROM layoffs_stagging2;

UPDATE layoffs_stagging2
SET company = TRIM(company);
```

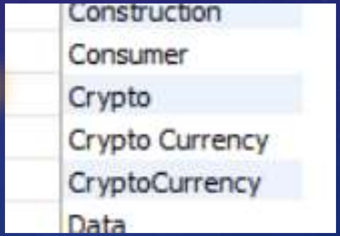


company	TRIM(company)
E Inc.	E Inc.
Included Health	Included Health
&Open	&Open
#Paid	#Paid
100 Thieves	100 Thieves
100 Thieves	100 Thieves
10X Genomics	10X Genomics
1stdibs	1stdibs
2TM	2TM
2TM	2TM
2U	2U
54gene	54gene
5B Solar	5B Solar

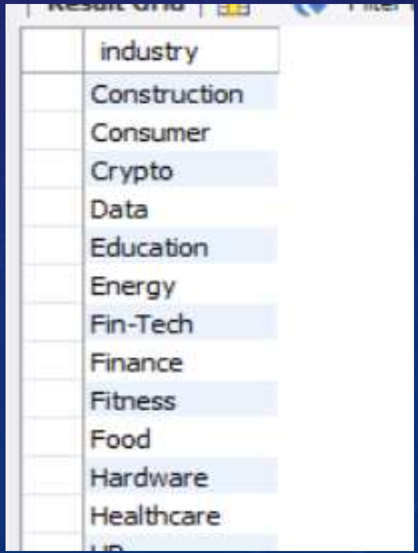
Changing all the alias names of a company into single name.



```
SELECT  DISTINCT industry
FROM layoffs_stagging2
ORDER BY 1;

SELECT *
FROM layoffs_stagging2
WHERE industry LIKE 'Crypto%';

UPDATE layoffs_stagging2
SET industry = 'Crypto'
WHERE industry LIKE 'Crypto%';
```

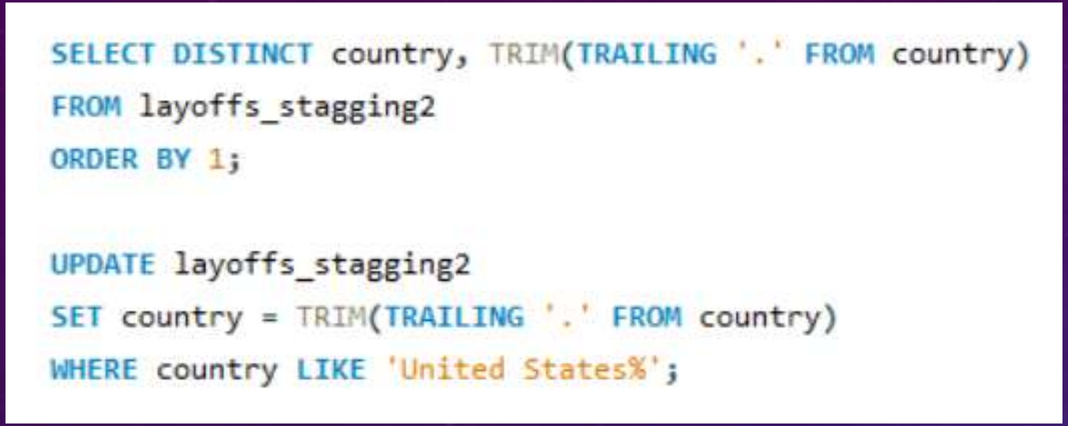


Construction
Consumer
Crypto
Crypto Currency
CryptoCurrency
Data

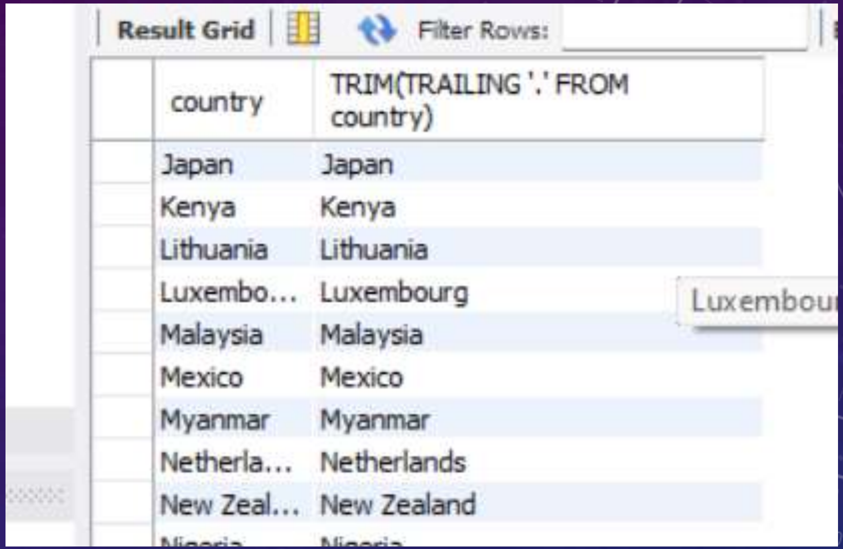


industry
Construction
Consumer
Crypto
Data
Education
Energy
Fin-Tech
Finance
Fitness
Food
Hardware
Healthcare
HR

Removing the trails such as … using the trim

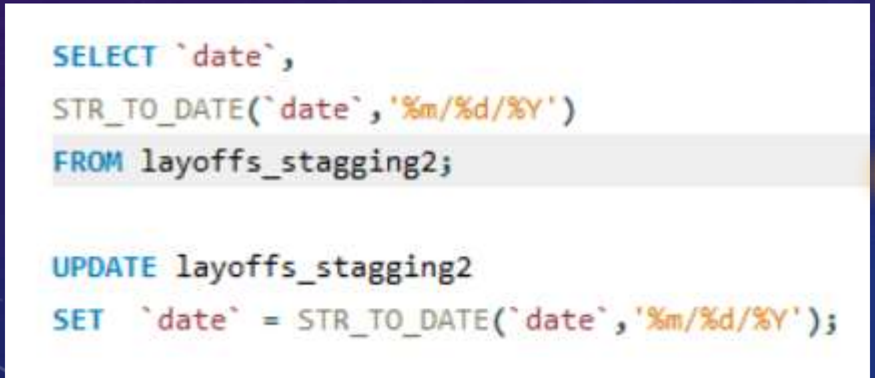


```
SELECT DISTINCT country, TRIM(TRAILING '.' FROM country)
FROM layoffs_stagging2
ORDER BY 1;

UPDATE layoffs_stagging2
SET country = TRIM(TRAILING '.' FROM country)
WHERE country LIKE 'United States%';
```

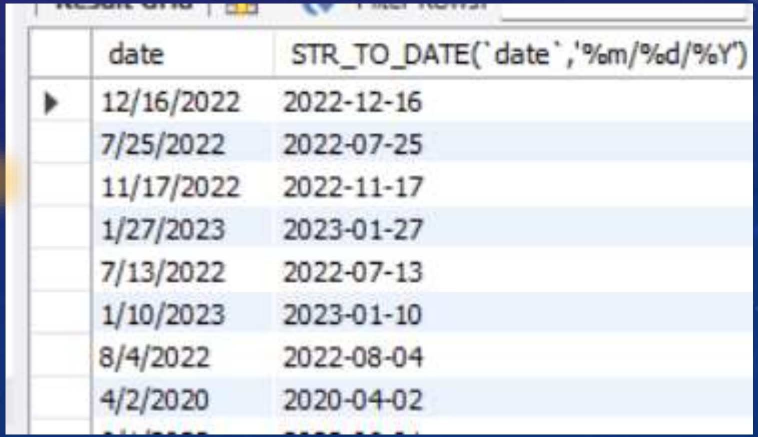


country	TRIM(TRAILING '.' FROM country)
Japan	Japan
Kenya	Kenya
Lithuania	Lithuania
Luxembo...	Luxembourg
Malaysia	Malaysia
Mexico	Mexico
Myanmar	Myanmar
Netherla...	Netherlands
New Zeal...	New Zealand
Nigeria	Nigeria

Changing the date column form string to date

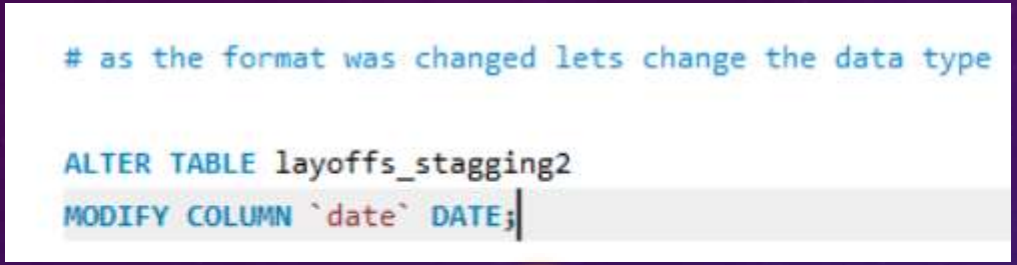


```
SELECT `date`,
STR_TO_DATE(`date`,'%m/%d/%Y')
FROM layoffs_stagging2;

UPDATE layoffs_stagging2
SET  `date` = STR_TO_DATE(`date`,'%m/%d/%Y');
```

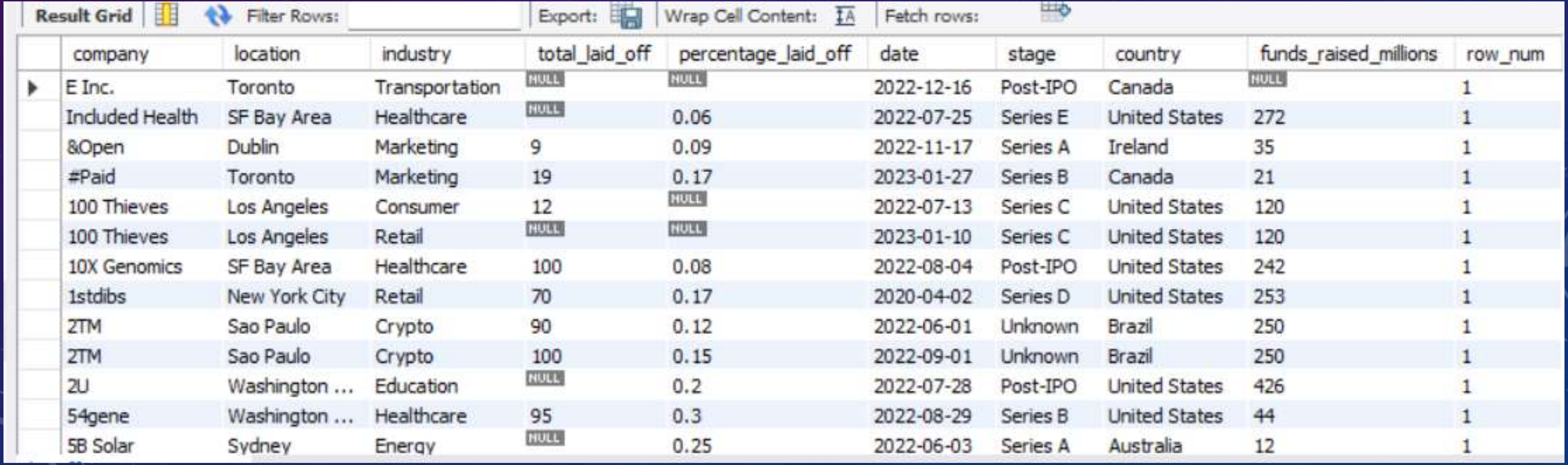


date	STR_TO_DATE(`date`,'%m/%d/%Y')
12/16/2022	2022-12-16
7/25/2022	2022-07-25
11/17/2022	2022-11-17
1/27/2023	2023-01-27
7/13/2022	2022-07-13
1/10/2023	2023-01-10
8/4/2022	2022-08-04
4/2/2020	2020-04-02

Updating the date data type to date format for time series analysis



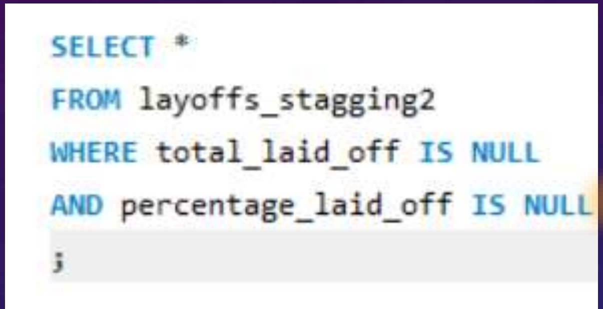
```
# as the format was changed lets change the data type

ALTER TABLE layoffs_stagging2
MODIFY COLUMN `date` DATE;
```



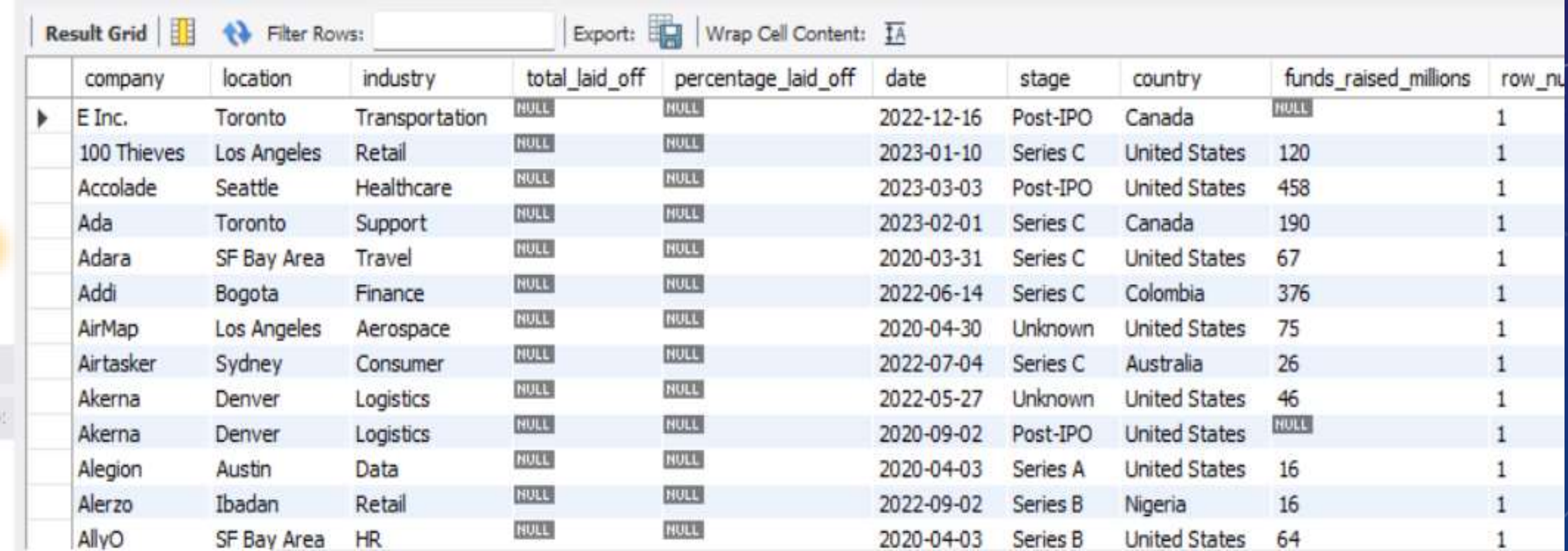
company	location	industry	total_laid_off	percentage_laid_off	date	stage	country	funds_raised_millions	row_num
E Inc.	Toronto	Transportation	NULL	NULL	2022-12-16	Post-IPO	Canada	NULL	1
Included Health	SF Bay Area	Healthcare	NULL	0.06	2022-07-25	Series E	United States	272	1
&Open	Dublin	Marketing	9	0.09	2022-11-17	Series A	Ireland	35	1
#Paid	Toronto	Marketing	19	0.17	2023-01-27	Series B	Canada	21	1
100 Thieves	Los Angeles	Consumer	12	NULL	2022-07-13	Series C	United States	120	1
100 Thieves	Los Angeles	Retail	NULL	NULL	2023-01-10	Series C	United States	120	1
10X Genomics	SF Bay Area	Healthcare	100	0.08	2022-08-04	Post-IPO	United States	242	1
1stdibs	New York City	Retail	70	0.17	2020-04-02	Series D	United States	253	1
2TM	Sao Paulo	Crypto	90	0.12	2022-06-01	Unknown	Brazil	250	1
2TM	Sao Paulo	Crypto	100	0.15	2022-09-01	Unknown	Brazil	250	1
2U	Washington ...	Education	NULL	0.2	2022-07-28	Post-IPO	United States	426	1
54gene	Washington ...	Healthcare	95	0.3	2022-08-29	Series B	United States	44	1
5B Solar	Sydney	Energy	NULL	0.25	2022-06-03	Series A	Australia	12	1

NULL AND BLANK VALUES

- As in the data standardized we can see the total_laid_off, percentage_laid_of show the null values.
- This can be identified by the code below.

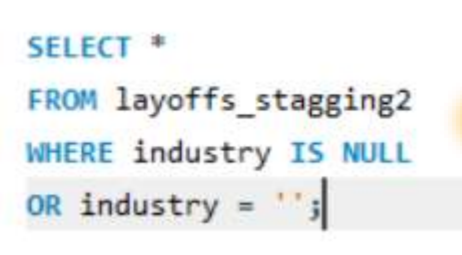


```
SELECT *
FROM layoffs_stagging2
WHERE total_laid_off IS NULL
AND percentage_laid_off IS NULL
;
```

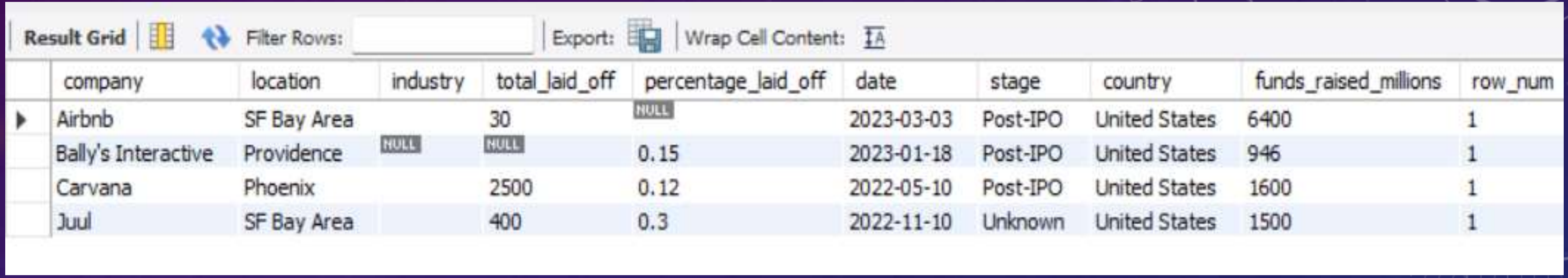


company	location	industry	total_laid_off	percentage_laid_off	date	stage	country	funds_raised_millions	row_nu
E Inc.	Toronto	Transportation	NULL	NULL	2022-12-16	Post-IPO	Canada	NULL	1
100 Thieves	Los Angeles	Retail	NULL	NULL	2023-01-10	Series C	United States	120	1
Accolade	Seattle	Healthcare	NULL	NULL	2023-03-03	Post-IPO	United States	458	1
Ada	Toronto	Support	NULL	NULL	2023-02-01	Series C	Canada	190	1
Adara	SF Bay Area	Travel	NULL	NULL	2020-03-31	Series C	United States	67	1
Addi	Bogota	Finance	NULL	NULL	2022-06-14	Series C	Colombia	376	1
AirMap	Los Angeles	Aerospace	NULL	NULL	2020-04-30	Unknown	United States	75	1
Airtasker	Sydney	Consumer	NULL	NULL	2022-07-04	Series C	Australia	26	1
Akerna	Denver	Logistics	NULL	NULL	2022-05-27	Unknown	United States	46	1
Akerna	Denver	Logistics	NULL	NULL	2020-09-02	Post-IPO	United States	NULL	1
Alegion	Austin	Data	NULL	NULL	2020-04-03	Series A	United States	16	1
Alerzo	Ibadan	Retail	NULL	NULL	2022-09-02	Series B	Nigeria	16	1
AllyO	SF Bay Area	HR	NULL	NULL	2020-04-03	Series B	United States	64	1

Industry column is also verified for the null and empty values.

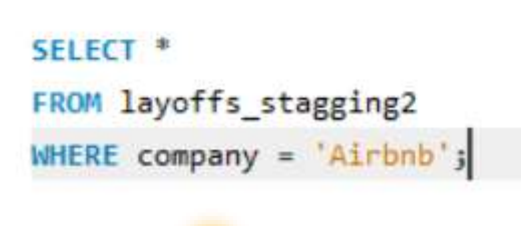


```
SELECT *
FROM layoffs_stagging2
WHERE industry IS NULL
OR industry = '';
```

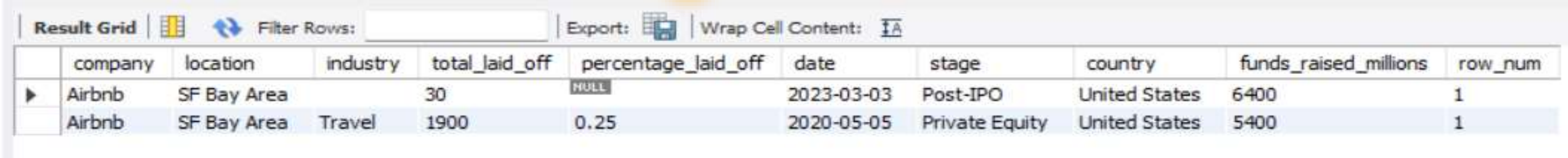


	company	location	industry	total_laid_off	percentage_laid_off	date	stage	country	funds_raised_millions	row_num
▶	Airbnb	SF Bay Area		30	NULL	2023-03-03	Post-IPO	United States	6400	1
	Bally's Interactive	Providence	NULL	NULL	0.15	2023-01-18	Post-IPO	United States	946	1
	Carvana	Phoenix		2500	0.12	2022-05-10	Post-IPO	United States	1600	1
	Juul	SF Bay Area		400	0.3	2022-11-10	Unknown	United States	1500	1

To verify that the industry is updated in any other column for the particular company the below code was run and found the industry type.

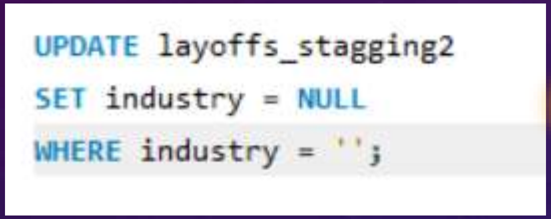


```
SELECT *
FROM layoffs_stagging2
WHERE company = 'Airbnb';
```

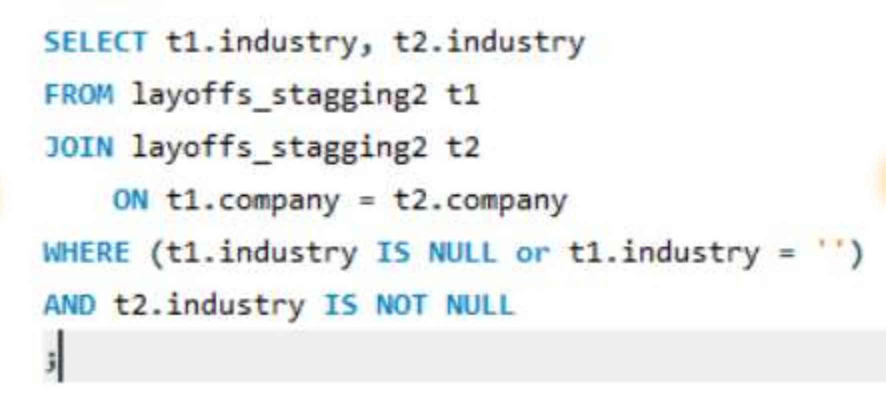


	company	location	industry	total_laid_off	percentage_laid_off	date	stage	country	funds_raised_millions	row_num
▶	Airbnb	SF Bay Area		30	NULL	2023-03-03	Post-IPO	United States	6400	1
	Airbnb	SF Bay Area	Travel	1900	0.25	2020-05-05	Private Equity	United States	5400	1

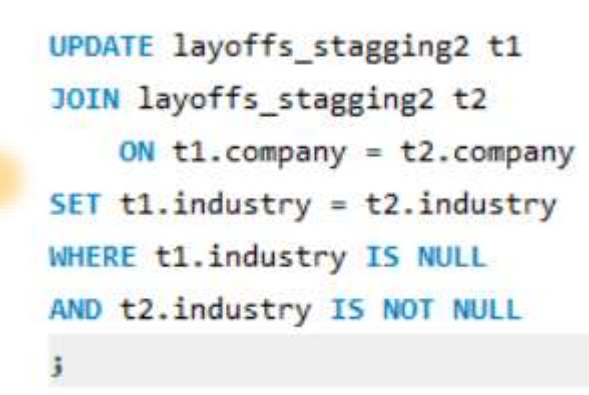
- To update the blank and null values first change the blank values to null values
- Then in the industry table to fill the blank values with appropriate industry type by using the Self join as the results were good the same is updated in the table.



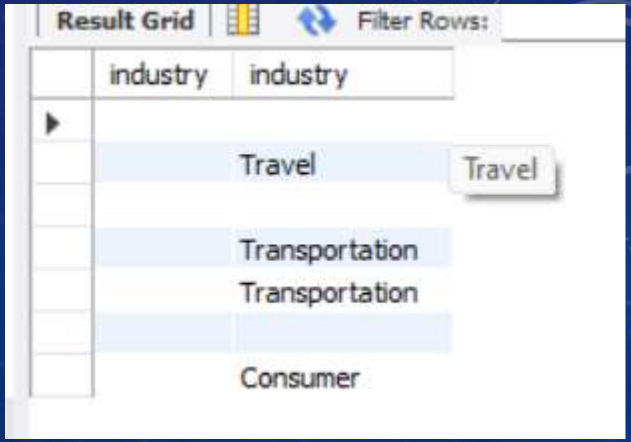
```
UPDATE layoffs_stagging2
SET industry = NULL
WHERE industry = '';
```



```
SELECT t1.industry, t2.industry
FROM layoffs_stagging2 t1
JOIN layoffs_stagging2 t2
    ON t1.company = t2.company
WHERE (t1.industry IS NULL or t1.industry = '')
AND t2.industry IS NOT NULL
;
```



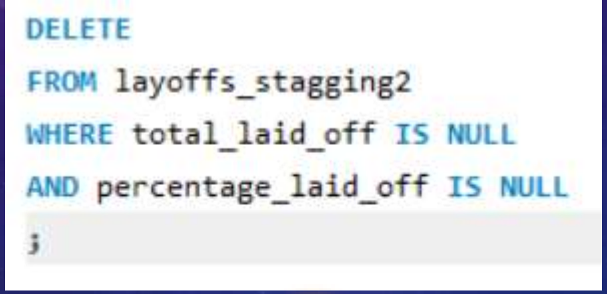
```
UPDATE layoffs_stagging2 t1
JOIN layoffs_stagging2 t2
    ON t1.company = t2.company
SET t1.industry = t2.industry
WHERE t1.industry IS NULL
AND t2.industry IS NOT NULL
;
```



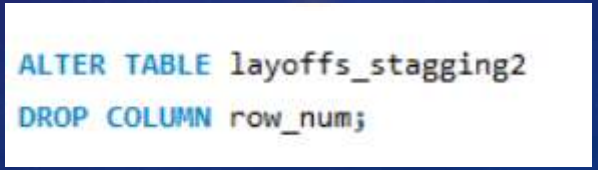
industry	industry
	Travel
	Travel
	Transportation
	Transportation
	Consumer

REMOVING THE COLUMNS AND ROWS

All the null values from the columns total_laid_off, percentage_laid_of were deleted and finally the row_num column which was created to remove duplicates was dropped form the table.

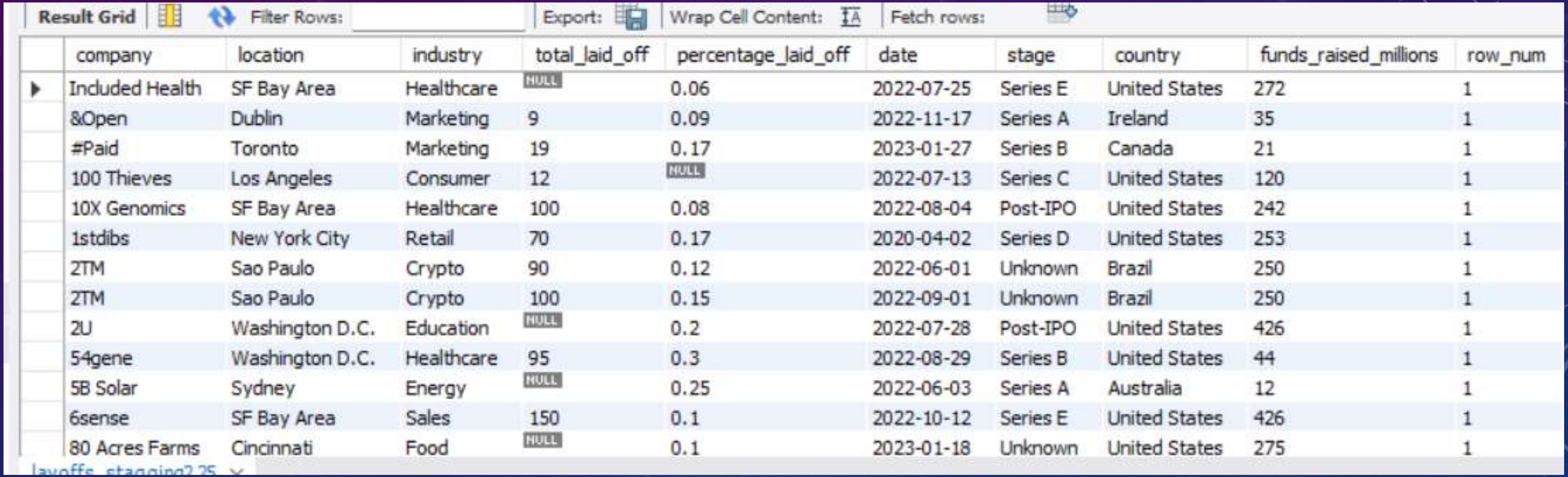


```
DELETE
FROM layoffs_stagging2
WHERE total_laid_off IS NULL
AND percentage_laid_off IS NULL
;
```



```
ALTER TABLE layoffs_stagging2
DROP COLUMN row_num;
```

THE FINAL RESULT OF THE DATA CLEANING IS AS FOLLOWS



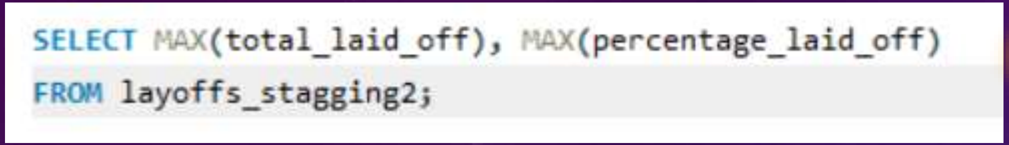
company	location	industry	total_laid_off	percentage_laid_off	date	stage	country	funds_raised_millions	row_num
Included Health	SF Bay Area	Healthcare	NULL	0.06	2022-07-25	Series E	United States	272	1
&Open	Dublin	Marketing	9	0.09	2022-11-17	Series A	Ireland	35	1
#Paid	Toronto	Marketing	19	0.17	2023-01-27	Series B	Canada	21	1
100 Thieves	Los Angeles	Consumer	12	NULL	2022-07-13	Series C	United States	120	1
10X Genomics	SF Bay Area	Healthcare	100	0.08	2022-08-04	Post-IPO	United States	242	1
1stdibs	New York City	Retail	70	0.17	2020-04-02	Series D	United States	253	1
2TM	Sao Paulo	Crypto	90	0.12	2022-06-01	Unknown	Brazil	250	1
2TM	Sao Paulo	Crypto	100	0.15	2022-09-01	Unknown	Brazil	250	1
2U	Washington D.C.	Education	NULL	0.2	2022-07-28	Post-IPO	United States	426	1
54gene	Washington D.C.	Healthcare	95	0.3	2022-08-29	Series B	United States	44	1
5B Solar	Sydney	Energy	NULL	0.25	2022-06-03	Series A	Australia	12	1
6sense	SF Bay Area	Sales	150	0.1	2022-10-12	Series E	United States	426	1
80 Acres Farms	Cincinnati	Food	NULL	0.1	2023-01-18	Unknown	United States	275	1

EDA FOR THE CLEANED DATA

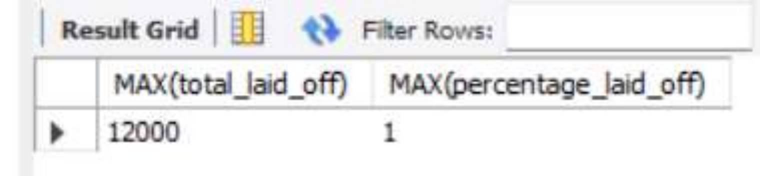
Questions Answered

- Maximum number of the persons laid off and the percent of persons laid of by a company?
- Percentage laid off is 1 and the total laid off in descending fashion?
- The maximum funds raised by the laid offs by the company?
- The maximum laid offs done by which company?
- The laid off duration mentioned in the data set?
- Which industry has maximum laid offs?
- Which country has highest laid offs?
- Sort the laid offs by the date?
- What are the number of laid offs each year?
- Maximum laid off by the company at which stage of their growth?
- What is the progression of layoffs and rolling total of the total laid off by every month?
- Top five companies in lay offs every year?

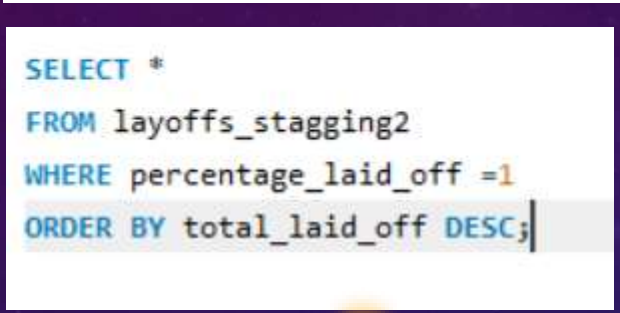
Maximum number of the persons laid off and the percent of persons laid of by a company?
This is done by applying the MAX aggregator



```
SELECT MAX(total_laid_off), MAX(percentage_laid_off)
FROM layoffs_stagging2;
```

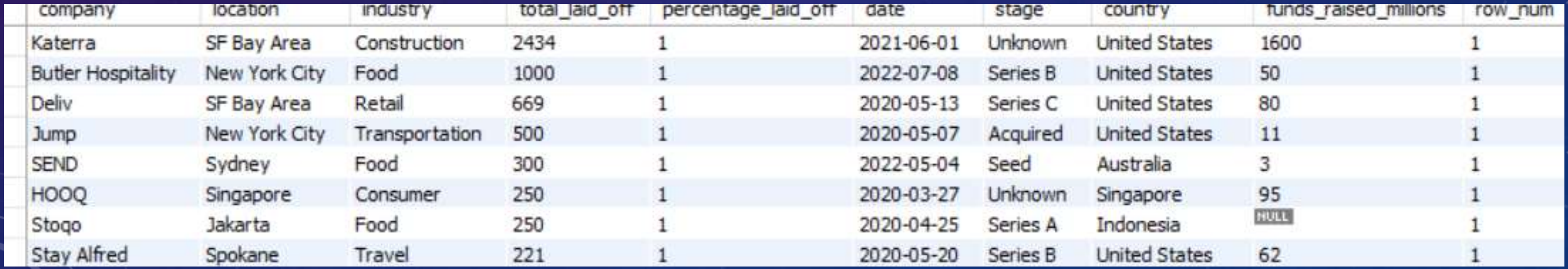


MAX(total_laid_off)	MAX(percentage_laid_off)
12000	1

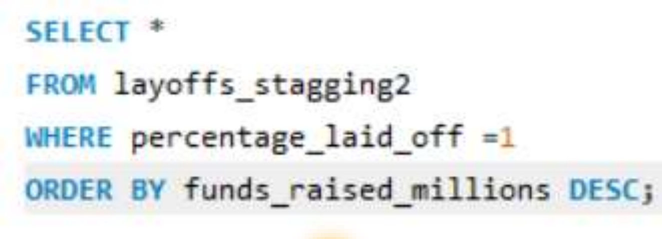


```
SELECT *
FROM layoffs_stagging2
WHERE percentage_laid_off =1
ORDER BY total_laid_off DESC;
```

Percentage laid off is 1 and the total laid off in descending fashion?
This is done by using the WHERE clause with percentage paid off =1 and ORDER BY total laid off

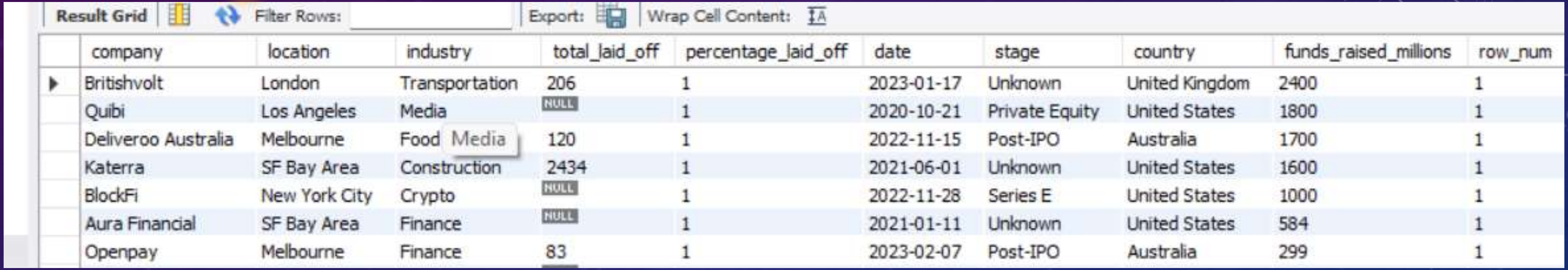


company	location	industry	total_laid_off	percentage_laid_off	date	stage	country	funds_raised_millions	row_num
Katerra	SF Bay Area	Construction	2434	1	2021-06-01	Unknown	United States	1600	1
Butler Hospitality	New York City	Food	1000	1	2022-07-08	Series B	United States	50	1
Deliv	SF Bay Area	Retail	669	1	2020-05-13	Series C	United States	80	1
Jump	New York City	Transportation	500	1	2020-05-07	Acquired	United States	11	1
SEND	Sydney	Food	300	1	2022-05-04	Seed	Australia	3	1
HOOQ	Singapore	Consumer	250	1	2020-03-27	Unknown	Singapore	95	1
Stoqo	Jakarta	Food	250	1	2020-04-25	Series A	Indonesia	NULL	1
Stay Alfred	Spokane	Travel	221	1	2020-05-20	Series B	United States	62	1



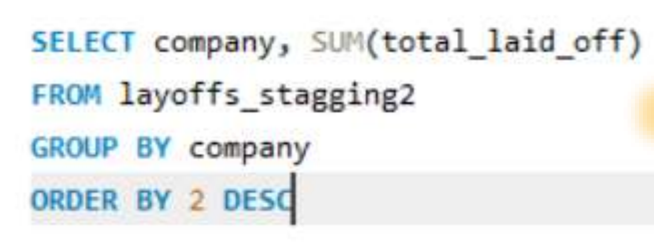
```sql
SELECT *
FROM layoffs_stagging2
WHERE percentage_laid_off =1
ORDER BY funds_raised_millions DESC;
```

The maximum funds raised by the laid offs by the company?

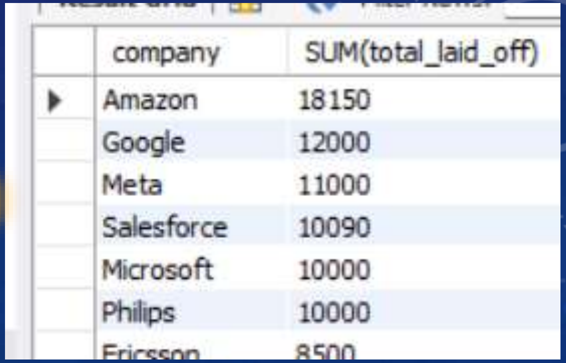


company	location	industry	total_laid_off	percentage_laid_off	date	stage	country	funds_raised_millions	row_num
Britishvolt	London	Transportation	206	1	2023-01-17	Unknown	United Kingdom	2400	1
Quibi	Los Angeles	Media	NULL	1	2020-10-21	Private Equity	United States	1800	1
Deliveroo Australia	Melbourne	Food	120	1	2022-11-15	Post-IPO	Australia	1700	1
Katerra	SF Bay Area	Construction	2434	1	2021-06-01	Unknown	United States	1600	1
BlockFi	New York City	Crypto	NULL	1	2022-11-28	Series E	United States	1000	1
Aura Financial	SF Bay Area	Finance	NULL	1	2021-01-11	Unknown	United States	584	1
Openpay	Melbourne	Finance	83	1	2023-02-07	Post-IPO	Australia	299	1

The maximum laid offs done by which company? This is obtained by GROUP BY company and ORDER BY sum of total laid off.

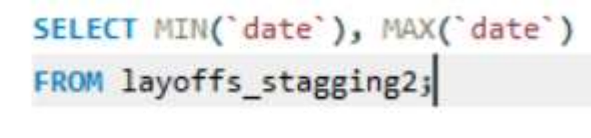


```sql
SELECT company, SUM(total_laid_off)
FROM layoffs_stagging2
GROUP BY company
ORDER BY 2 DESC
```

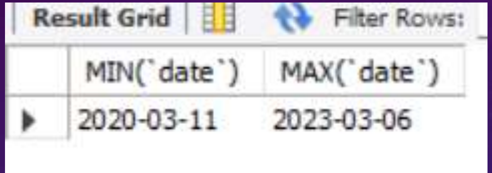


company	SUM(total_laid_off)
Amazon	18150
Google	12000
Meta	11000
Salesforce	10090
Microsoft	10000
Philips	10000
Ericsson	8500

The laid off duration mentioned in the data set?
This is by using the MIN and MAX

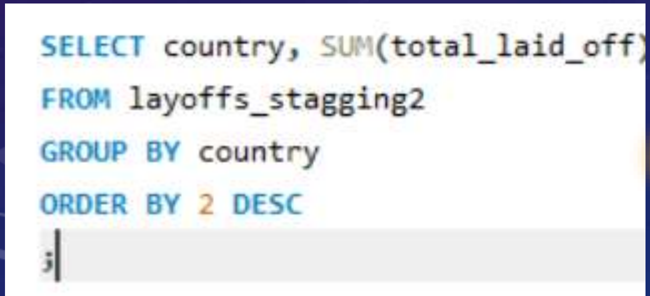


```
SELECT MIN(`date`), MAX(`date`)
FROM layoffs_stagging2;
```

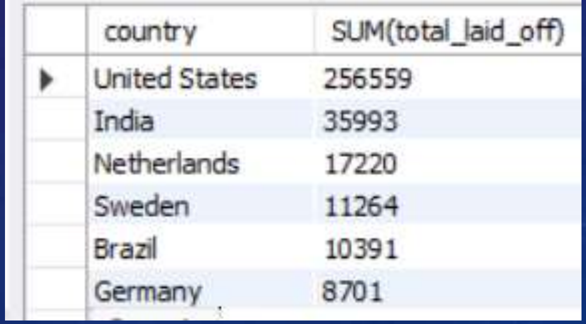


MIN(`date`)	MAX(`date`)
2020-03-11	2023-03-06

Which country has highest laid offs?
This is by GROUP BY country and ORDER BY total laid off.

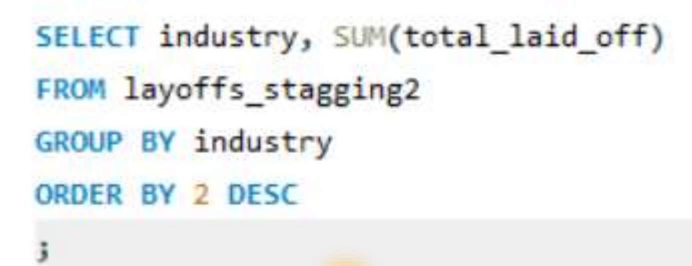


```
SELECT country, SUM(total_laid_off)
FROM layoffs_stagging2
GROUP BY country
ORDER BY 2 DESC
;
```

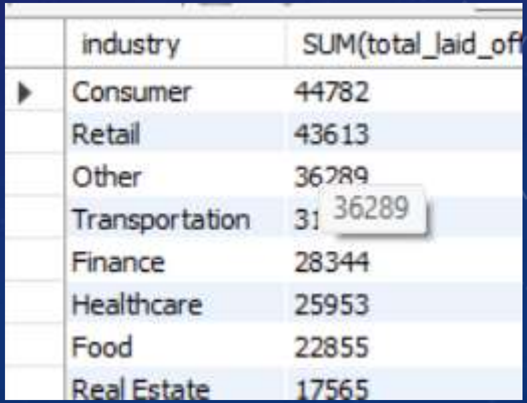


country	SUM(total_laid_off)
United States	256559
India	35993
Netherlands	17220
Sweden	11264
Brazil	10391
Germany	8701

Which industry has maximum laid offs?
This is obtained by GROUP BY industry and ORDER BY sum of total laid off.

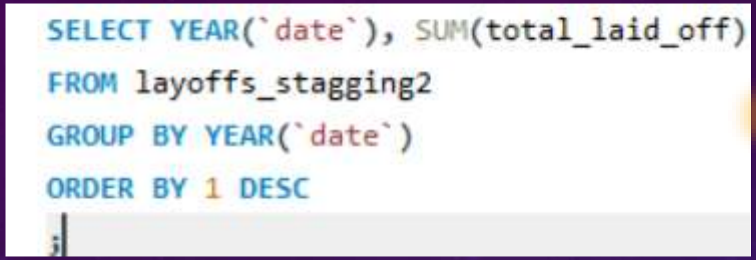


```
SELECT industry, SUM(total_laid_off)
FROM layoffs_stagging2
GROUP BY industry
ORDER BY 2 DESC
;
```

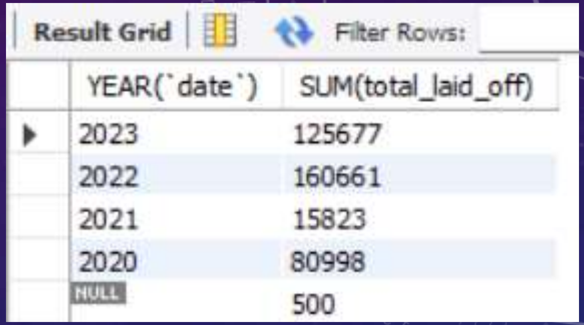


industry	SUM(total_laid_of
Consumer	44782
Retail	43613
Other	36289
Transportation	31 36289
Finance	28344
Healthcare	25953
Food	22855
Real Estate	17565

What are the number of laid offs each year? This is by GROUP BY Year and ORDER BY total laid off.

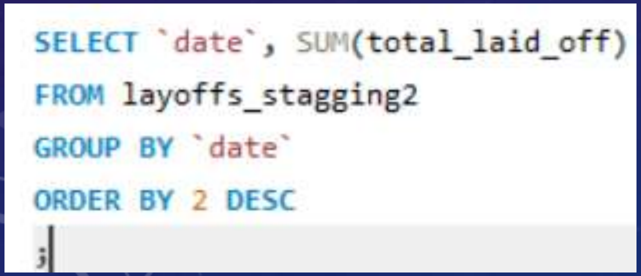


```
SELECT YEAR(`date`), SUM(total_laid_off)
FROM layoffs_stagging2
GROUP BY YEAR(`date`)
ORDER BY 1 DESC
;
```

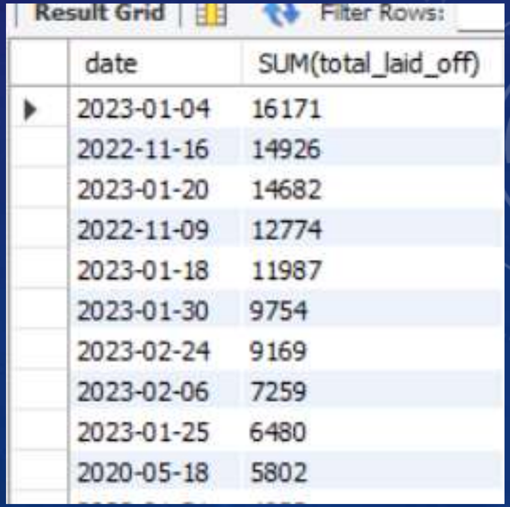


YEAR(`date`)	SUM(total_laid_off)
2023	125677
2022	160661
2021	15823
2020	80998
NULL	500

Sort the laid offs by the date? This is by GROUP BY date and ORDER BY total laid off.

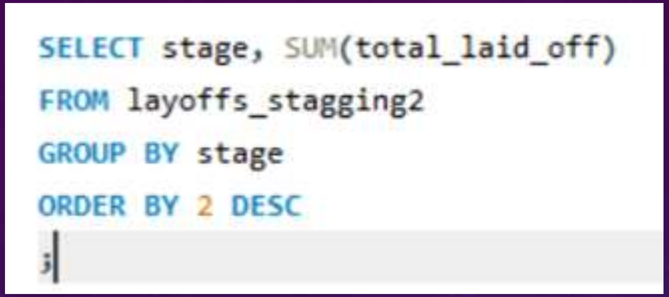


```
SELECT `date`, SUM(total_laid_off)
FROM layoffs_stagging2
GROUP BY `date`
ORDER BY 2 DESC
;
```

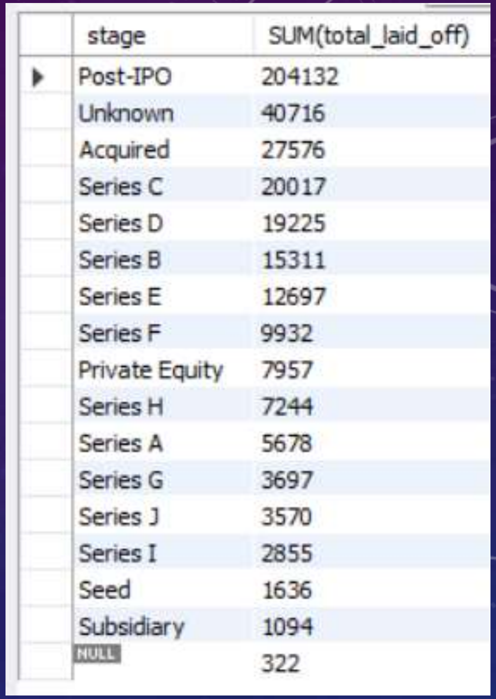


date	SUM(total_laid_off)
2023-01-04	16171
2022-11-16	14926
2023-01-20	14682
2022-11-09	12774
2023-01-18	11987
2023-01-30	9754
2023-02-24	9169
2023-02-06	7259
2023-01-25	6480
2020-05-18	5802

Maximum laid off by the company at which stage of their growth?
This is by GROUP BY stage and ORDER BY total laid off

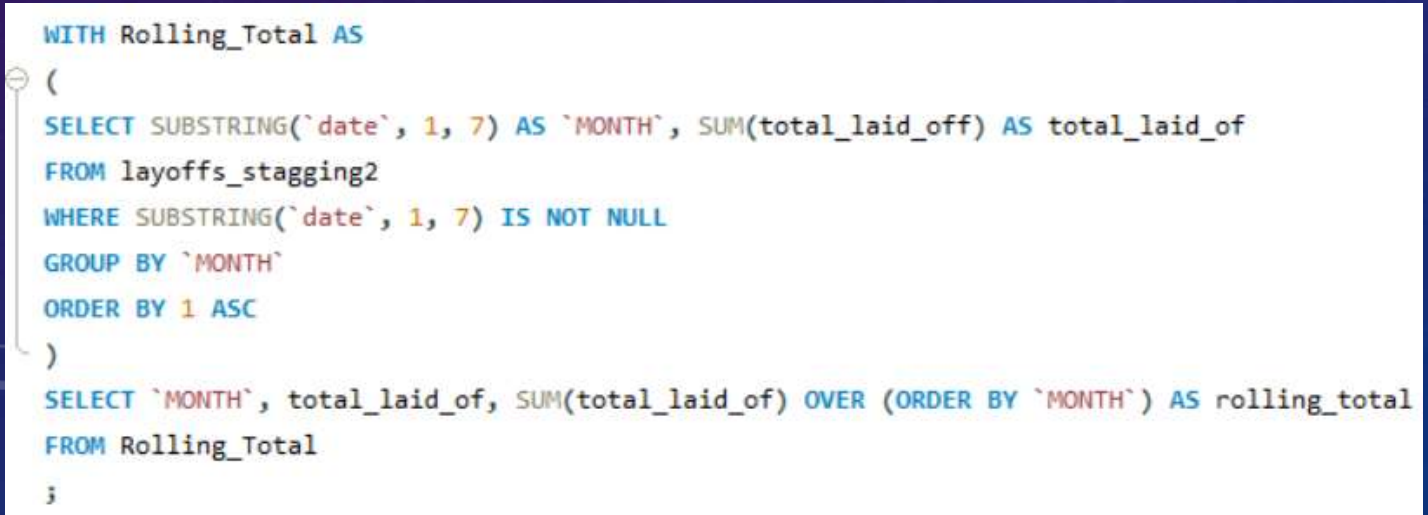


```
SELECT stage, SUM(total_laid_off)
FROM layoffs_stagging2
GROUP BY stage
ORDER BY 2 DESC
;
```

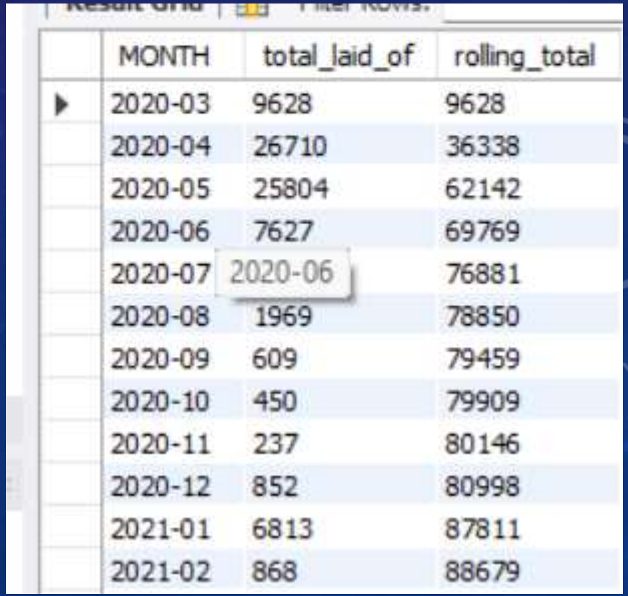


stage	SUM(total_laid_off)
Post-IPO	204132
Unknown	40716
Acquired	27576
Series C	20017
Series D	19225
Series B	15311
Series E	12697
Series F	9932
Private Equity	7957
Series H	7244
Series A	5678
Series G	3697
Series J	3570
Series I	2855
Seed	1636
Subsidiary	1094
NULL	322

What is the progression of layoffs and rolling total of the total laid off by every month? The progression can be obtained by the SUB STRING created as CTE then the Rolling Total is obtained by OVER and ORDER BY.

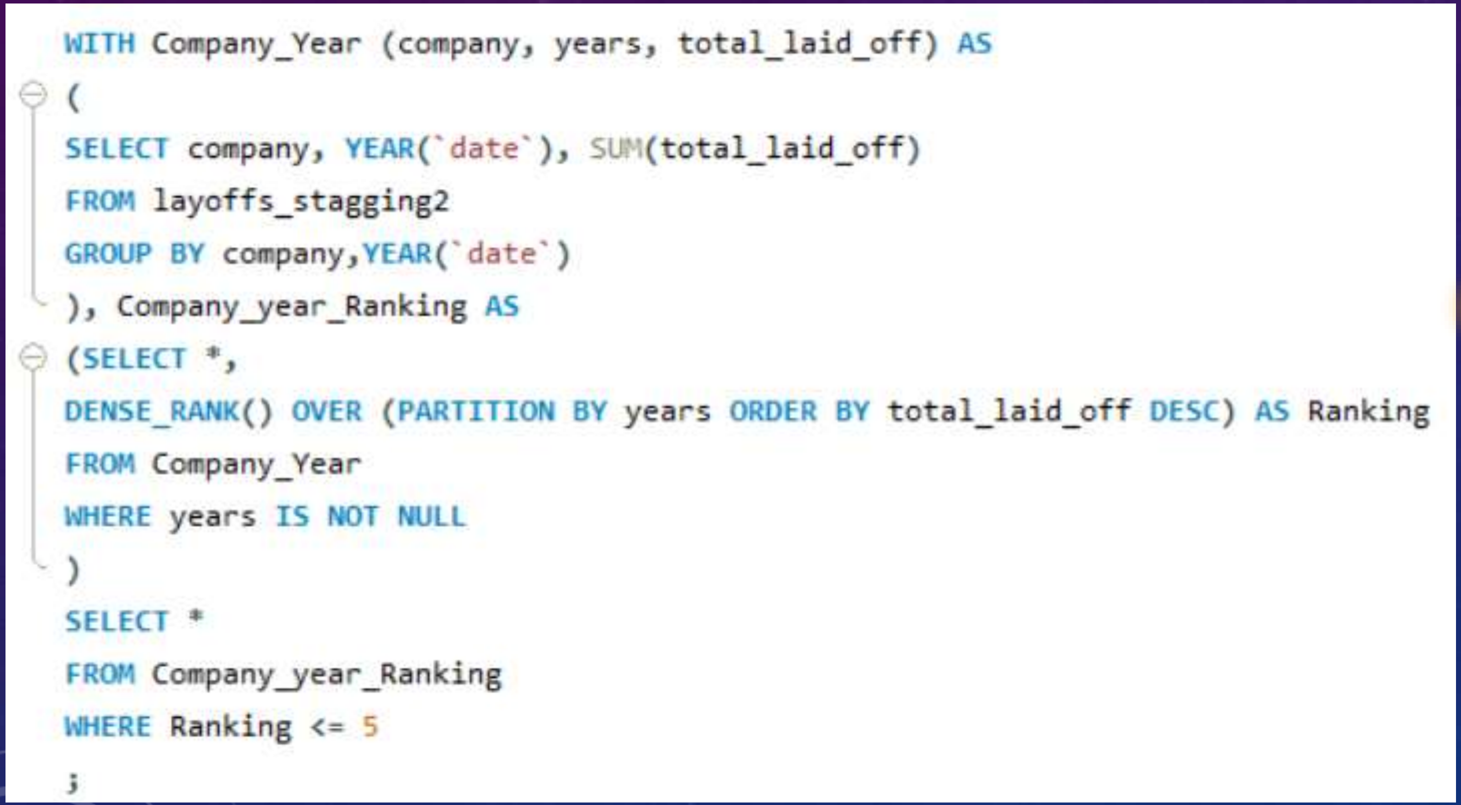


```
WITH Rolling_Total AS
(
SELECT SUBSTRING(`date`, 1, 7) AS `MONTH`, SUM(total_laid_off) AS total_laid_of
FROM layoffs_stagging2
WHERE SUBSTRING(`date`, 1, 7) IS NOT NULL
GROUP BY `MONTH`
ORDER BY 1 ASC
)
SELECT `MONTH`, total_laid_of, SUM(total_laid_of) OVER (ORDER BY `MONTH`) AS rolling_total
FROM Rolling_Total
;
```

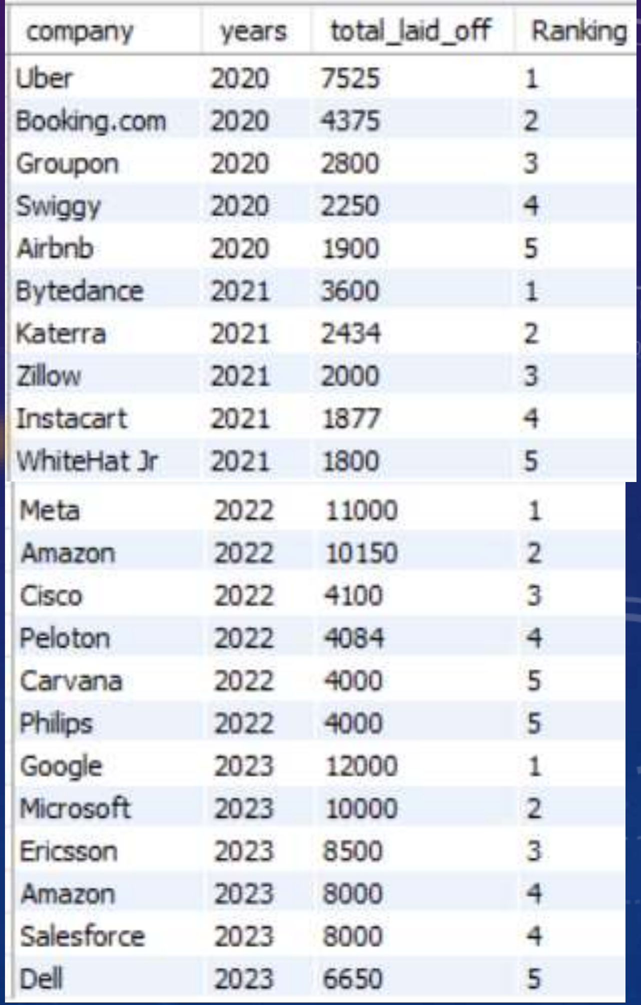


MONTH	total_laid_of	rolling_total
2020-03	9628	9628
2020-04	26710	36338
2020-05	25804	62142
2020-06	7627	69769
2020-07	2020-06	76881
2020-08	1969	78850
2020-09	609	79459
2020-10	450	79909
2020-11	237	80146
2020-12	852	80998
2021-01	6813	87811
2021-02	868	88679

Top five companies in lay offs every year?
Answer is obtained by using the CTE Sub Query, OVER and Partition By, Order by And Ranking.



```
WITH Company_Year (company, years, total_laid_off) AS
(
SELECT company, YEAR(`date`), SUM(total_laid_off)
FROM layoffs_stagging2
GROUP BY company,YEAR(`date`)
), Company_year_Ranking 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_Ranking
WHERE Ranking <= 5
;
```



company	years	total_laid_off	Ranking
Uber	2020	7525	1
Booking.com	2020	4375	2
Groupon	2020	2800	3
Swiggy	2020	2250	4
Airbnb	2020	1900	5
Bytedance	2021	3600	1
Katerra	2021	2434	2
Zillow	2021	2000	3
Instacart	2021	1877	4
WhiteHat Jr	2021	1800	5
Meta	2022	11000	1
Amazon	2022	10150	2
Cisco	2022	4100	3
Peloton	2022	4084	4
Carvana	2022	4000	5
Philips	2022	4000	5
Google	2023	12000	1
Microsoft	2023	10000	2
Ericsson	2023	8500	3
Amazon	2023	8000	4
Salesforce	2023	8000	4
Dell	2023	6650	5

INSIGHTS

- This data is from 11-03-2020 to 06-03-2023.
- Maximum number of persons laid off were 12000 and the percent are 1% in the range 0-1.
- The highest laid offs were 2434, 1000 ,669 by kateera, butler hospitality, Deliv date wise.
- The highest funds raised in millions by the laid offs were 2400, 1800, 1700 by Britishvolt, qubi, Deliveroo Australia.
- The highest laid offs were done by the companies which are in Post IPO stage (204132) showing that top companies has laid off maximum employees.
- The total laid offs done by each company has the highest as 18150, 12000, 11000 by Amazon, Google and Meta and followed by others.
- The industry which has highest laid offs were 44782, 43613, 36289 in Consumer, Retail, other (were the industry is not specified) and follows.
- The laid off country wise as follows 256559, 35993, 17220 in United States, India, Netherlands.
- The Year wise laid offs were like 125677, 160661,15823, 80998 in 2023,2022, 2021, 2020 were the highest was observed in 2022 followed by 2023,2020 and 2021.
- The date wise total laid offs has the highest as 16171, 14926, 14682 in 04-01-2023, 16-11-202 and 20-01-2023 as week can see the top 1st and 3rd spots were taken by 2023 we can say lot off the employees loose their job in a single day in 2023.
- The year wise top spot in laid off was taken by uber, bytedance, Meta and google from 2020-2023.

Thank You