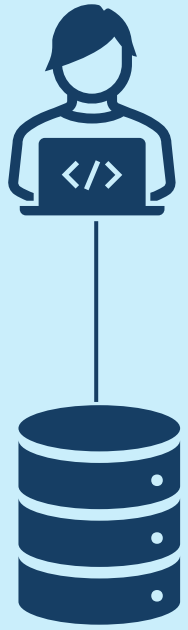
Global Layoffs
SQL Project
Part- 1



Syed Nadeem Yousuf

Introduction

In this project dataset which is used is Global Layoffs, This project is designed to research global layoff trends in a more profound way by identifying patterns of layoffs across various industries, companies, and regions for a comprehensive overview of workforce reduction worldwide..

Dataset Includes:

- Company : The companies which layoffs (String)
- Location HQ : The Location of the company's headquarter (String)
- Industry : Company belong to which industry (String)
- Laid OFF : Number of people laid off (Int)
- Date : Date of layoffs (date) **, Data is till 5th of June 2024.**
- Funds raised (In Millions or billions) : Amount of funding company has raised (string)
- Stage : Stage of the company (string)
- Country : Company headquarter is in which country (string)
- Percentage : Layoff Percentage

Start with Data Cleaning

- **Step 1 :** Making copy of actual data and working on the copy data because if any mistake happen at least original data will be there. This is a good practice to do.

```
create table copy_global_layoff
like global_layoffs;
```

Use this query to create the same table as original one.

```
insert into copy_global_layoff
select * from global_layoffs;
```

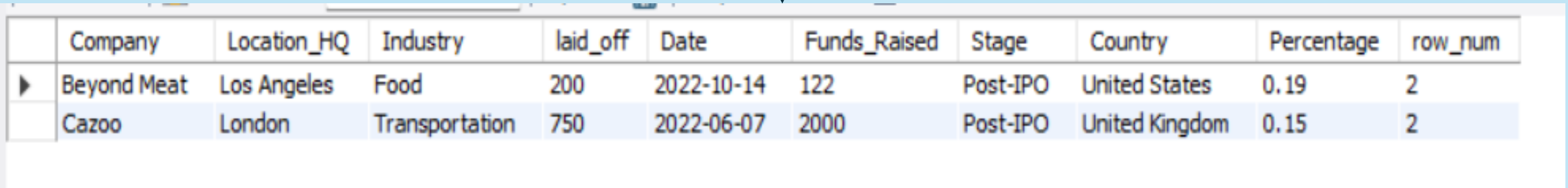
Use this query to insert same data as the original table is having.

	Company	Location_HQ	Industry	laid_off	Date	Funds_Raised	Stage	Country	Percentage
▶	Oda	Oslo	Food	150	2024-06-05	691	Unknown	Norway	
	Pagaya	Tel Aviv	Finance	100	2024-06-05	2000	Post-IPO	Israel	0.2
	Aleph Farms	Tel Aviv	Food	30	2024-06-05	119	Unknown	Israel	0.3
	MoonPay	Dover	Crypto	30	2024-06-05	651	Unknown	United States	0.1
	Yext	New York City	Marketing		2024-06-05	117	Post-IPO	United States	0.12
	Microsoft	Seattle	Other	1000	2024-06-03	1	Post-IPO	United States	
	OrCam	Jerusalem	Healthcare	100	2024-06-03	86	Unknown	Israel	0.5
	Google	SF Bay Area	Consumer	100	2024-05-31	26	Post-IPO	United States	
	Tropic	New York City	Finance	40	2024-05-31	67	Series B	United States	
	Gro Intelligence	New York City	Food		2024-05-31	118	Series B	United States	0.1

- **Step 2 :** Removing duplicates values

There was no key column in this dataset such as sr.no or something like uniquely identifier so I use window function (ROW_Number) and cte to identify duplicates and remove it.

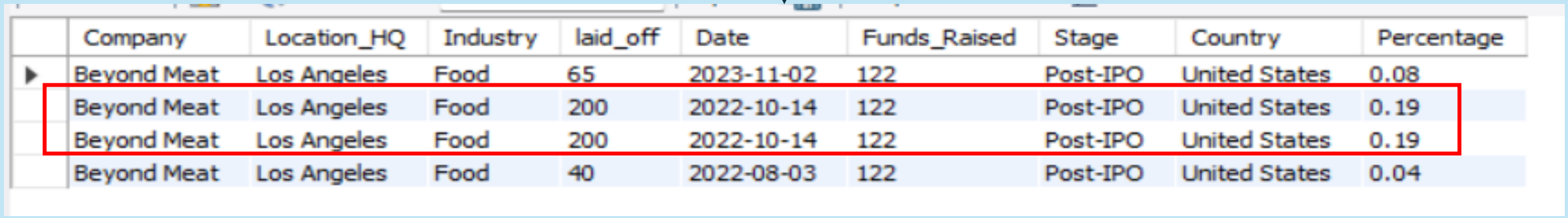
```
with cte as (select*,row_number()
over(partition by company,location_HQ,industry,laid_off,date,funds_raised,stage,country,percentage) as row_num
from copy_global_layoff)
select * from cte where row_num > 1;
```



Company	Location_HQ	Industry	laid_off	Date	Funds_Raised	Stage	Country	Percentage	row_num
Beyond Meat	Los Angeles	Food	200	2022-10-14	122	Post-IPO	United States	0.19	2
Cazoo	London	Transportation	750	2022-06-07	2000	Post-IPO	United Kingdom	0.15	2

This query shows that there are 2 duplicate records for both company.

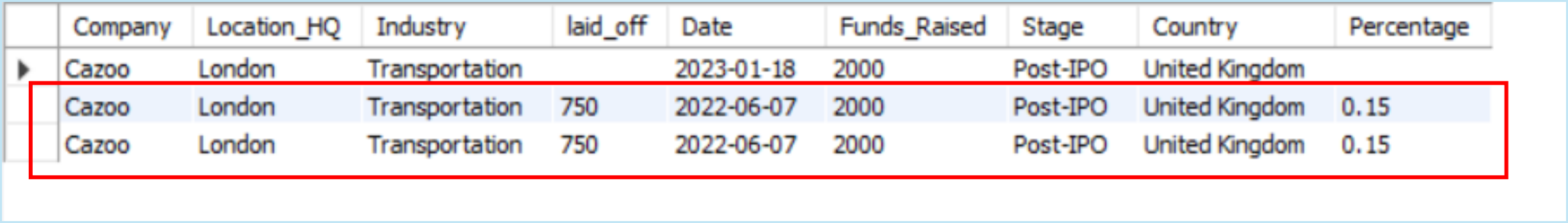
```sql
select * from copy_global_layoff
where company like '%meat'
```



Company	Location_HQ	Industry	laid_off	Date	Funds_Raised	Stage	Country	Percentage
Beyond Meat	Los Angeles	Food	65	2023-11-02	122	Post-IPO	United States	0.08
Beyond Meat	Los Angeles	Food	200	2022-10-14	122	Post-IPO	United States	0.19
Beyond Meat	Los Angeles	Food	200	2022-10-14	122	Post-IPO	United States	0.19
Beyond Meat	Los Angeles	Food	40	2022-08-03	122	Post-IPO	United States	0.04

This record was duplicate so removing one of them.

```sql
select * from copy_global_layoff
where company = 'cazoo';
```



Company	Location_HQ	Industry	laid_off	Date	Funds_Raised	Stage	Country	Percentage
Cazoo	London	Transportation		2023-01-18	2000	Post-IPO	United Kingdom	
Cazoo	London	Transportation	750	2022-06-07	2000	Post-IPO	United Kingdom	0.15
Cazoo	London	Transportation	750	2022-06-07	2000	Post-IPO	United Kingdom	0.15

This record was duplicate so removing one of them.

If there were lots of duplicate records, then creating a new table and inserting data from the window function (Row_Number) which was used to identify duplicate records. And then delete the duplicate records from the new table.

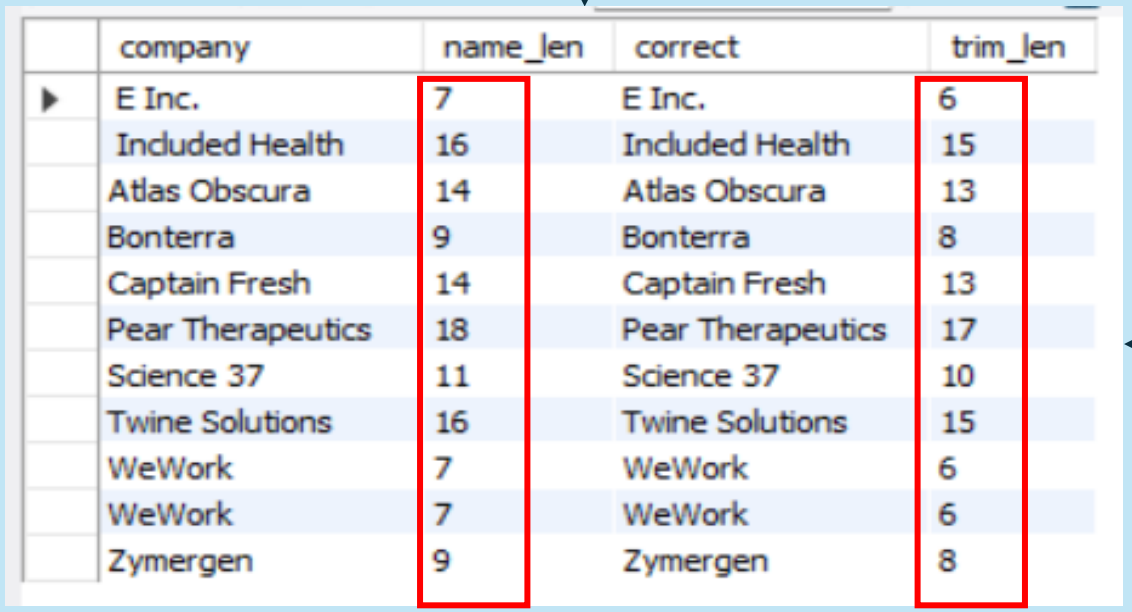
- **Step 3 :** Standardizing Formats

Check for any spelling errors or extra spaces or datatype of columns.

```
select company,length(company) name_len,
trim(company) correct,length(trim(company)) trim_len
from copy_global_layoff
having name_len != trim_len
order by company;
```

Use this query to Fix it.

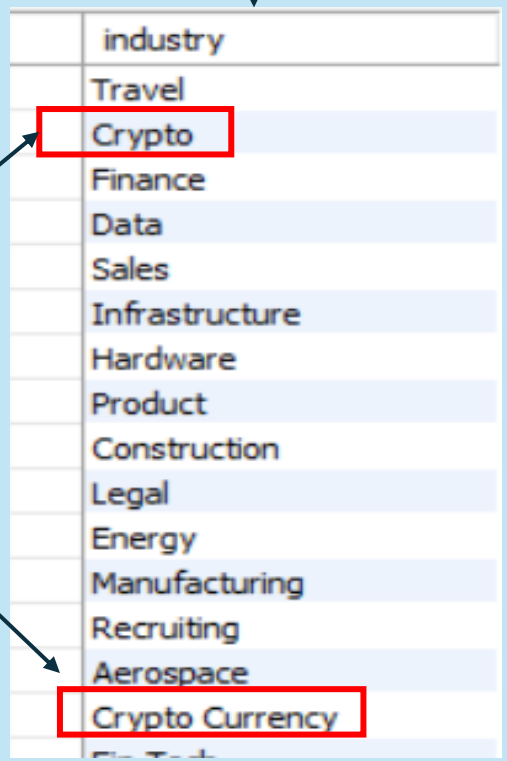
```
update copy_global_layoff
set company = trim(company);
```



company	name_len	correct	trim_len
E Inc.	7	E Inc.	6
Included Health	16	Included Health	15
Atlas Obscura	14	Atlas Obscura	13
Bonterra	9	Bonterra	8
Captain Fresh	14	Captain Fresh	13
Pear Therapeutics	18	Pear Therapeutics	17
Science 37	11	Science 37	10
Twine Solutions	16	Twine Solutions	15
WeWork	7	WeWork	6
WeWork	7	WeWork	6
Zymergen	9	Zymergen	8

In this dataset these records have extra spaces

```sql
select distinct industry
from copy_global_layoff;
```



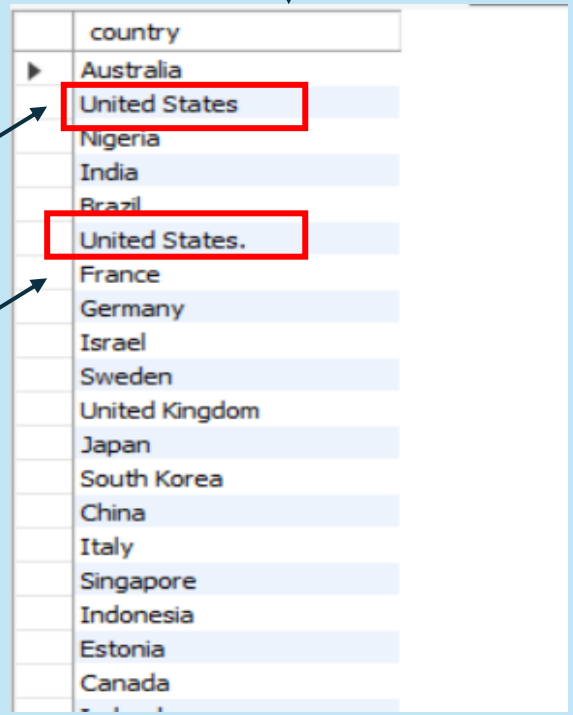
industry
Travel
Crypto
Finance
Data
Sales
Infrastructure
Hardware
Product
Construction
Legal
Energy
Manufacturing
Recruiting
Aerospace
Crypto Currency

These two are same , spelling mistake is there

Use this query to Fix it.

```sql
update copy_global_layoff
set industry = 'Crypto'
where industry like 'Crypto%'
```

```sql
select distinct country
from copy_global_layoff;
```



country
Australia
United States
Nigeria
India
Brazil
United States.
France
Germany
Israel
Sweden
United Kingdom
Japan
South Korea
China
Italy
Singapore
Indonesia
Estonia
Canada

Spelling mistake at the end of United States there is dot

Use this query to Fix it.

```sql
update copy_global_layoff
set country = 'United States'
where country = 'United States.';
```

```
desc copy_global_layoff
```

Field	Type	Null	Key	Default	Extra
Company	text	YES		NULL	
Location_HQ	text	YES		NULL	
Industry	text	YES		NULL	
laid_off	text	YES		NULL	
Date	text	YES		NULL	
Funds_Raised	double	YES		NULL	
Stage	text	YES		NULL	
Country	text	YES		NULL	
Percentage	text	YES		NULL	

Here these two-column datatype is wrong Laid_off should be integer and Date column datatype should be date.

```
alter table copy_global_layoff
modify column laid_off int;
```

Use this query to Fix it.

```
alter table copy_global_layoff
modify column `date` date;
```

Fixed

Field	Type	Null	Key	Default	Extra
Company	text	YES		NULL	
Location_HQ	text	YES		NULL	
Industry	text	YES		NULL	
Laid_Off	int	YES		NULL	
date	date	YES		NULL	
Funds_Raised	double	YES		NULL	
Stage	text	YES		NULL	
Country	text	YES		NULL	
Percentage	text	YES		NULL	

- **Step 4 :** Removing Null or Blank values

In this dataset there were no (Null) values but there are (Blank) values in Laid_off and percentage columns. But the thing was if there is a blank value in Laid_off column then there was no blank value in percentage column and vice versa. So, I decided to check whether both columns have blank value in the same row. If yes, then those records are of no use because there are no values then how can any own gather information from those records. So, I remove those records

```
select * from copy_global_layoff
where laid_off = '' and
Percentage = '';
```

Use this query to check blank values

Company	Location_HQ	Industry	laid_off	Date	Funds_Raised	Stage	Country	Percentage
Jasper Health	Boise	Healthcare		2024-05-31	32	Series A	United States	
Fisker	Los Angeles	Transportation		2024-05-29	1700	Post-IPO	United States	
Funding Circle	London	Finance		2024-05-29	746	Post-IPO	United Kingdom	
Hopin	London	Other		2024-05-07	1000	Series D	United Kingdom	
Google	SF Bay Area	Consumer		2024-04-30	26	Post-IPO	United States	
Fisker	Los Angeles	Transportation		2024-04-29	1700	Post-IPO	United States	
Grin	Sacramento	Marketing		2024-04-25	145	Series B	United States	
Expedia	Austin	Travel		2024-04-24	3300	Post-IPO	United States	
98point6	Seattle	Healthcare		2024-04-23	299	Acquired	United States	
Homie	Salt Lake City	Real Estate		2024-04-19	35	Series B	United States	
ConnectWise	Tampa Bay	Other		2024-04-17	1	Acquired	United States	
Google	SF Bay Area	Consumer		2024-04-17	26	Post-IPO	United States	
Agility Robotics	Portland	Other		2024-04-04	179	Series B	United States	
Amazon	Seattle	Retail		2024-04-03	108	Post-IPO	United States	
Kaseya	Miami	Security		2024-04-03	547	Unknown	United States	
New Relic	SF Bay Area	Infrastructure		2024-04-03	214	Acquired	United States	
Lentra	Pune	Finance		2024-04-02	104	Series B	India	
Verily	SF Bay Area	Healthcare		2024-03-22	3500	Subsidiary	United States	
Cybereason	Boston	Security		2024-03-20	750	Series F	United States	
Totango	SF Bay Area	Support		2024-03-07	147	Acquired	United States	

```sql
delete from copy_global_layoff
where laid_off = '' and
Percentage = '';
```


This delete all blank values from dataset. These records were of no use because how can we show information with blank values. So, removing blank values are the best solution I can say.

Company	Location_HQ	Industry	Laid_Off	date	Funds_Raised	Stage	Country	Percentage
&Open	Dublin	Marketing	9	2022-11-17	35	Series A	Ireland	0.09
#Paid	Toronto	Marketing	19	2023-01-27	21	Series B	Canada	0.17
10X Genomics	SF Bay Area	Healthcare	100	2022-08-04	242	Post-IPO	United States	0.08
1stdibs	New York City	Retail	70	2020-04-02	253	Series D	United States	0.17
23andMe	SF Bay Area	Healthcare	71	2023-08-08	1100	Post-IPO	United States	0.11
23andMe	SF Bay Area	Healthcare	75	2023-06-09	1100	Post-IPO	United States	0.09
2TM	Sao Paulo	Crypto	100	2022-09-01	250	Unknown	Brazil	0.15
2TM	Sao Paulo	Crypto	90	2022-06-01	250	Unknown	Brazil	0.12
54gene	Washington D.C.	Healthcare	95	2022-08-29	44	Series B	United States	0.3
6sense	SF Bay Area	Sales	150	2022-10-12	426	Series F	United States	0.1

During the cleaning process, 2 duplicate records were removed; 11 records were having extra spaces, so they were trimmed; spelling errors were found in 2 records and then corrected. Corrected datatypes for Laid_off and date columns and removed the records with blank values. Now this dataset is ready for further in-depth analysis.

This is the End of Data Cleaning; In Part 2, we will analyze the cleaned dataset regarding the trends in layoffs across industries, countries, and time to provide a broad overview of the job cuts around the world.

Thank you