📘 Title: SQL Data Cleaning – Layoffs Dataset

📌 **Subtitle:**
A practice project using a real and publicly available dataset, showcasing SQL-based data cleaning & transformation skills.
By Shoeb Md Ashraf

🛠️ **Tools Used:**
SQL (MySQL), Excel

🗒️ Project Overview

This project involved cleaning a messy layoffs dataset using SQL. The data contained duplicate records, inconsistent formatting, missing values, and incorrectly typed fields. Using SQL functions like ROW_NUMBER(), STR_TO_DATE(), TRIM(), and JOIN, the dataset was cleaned and transformed into an analysis-ready format.

✅ Key Steps

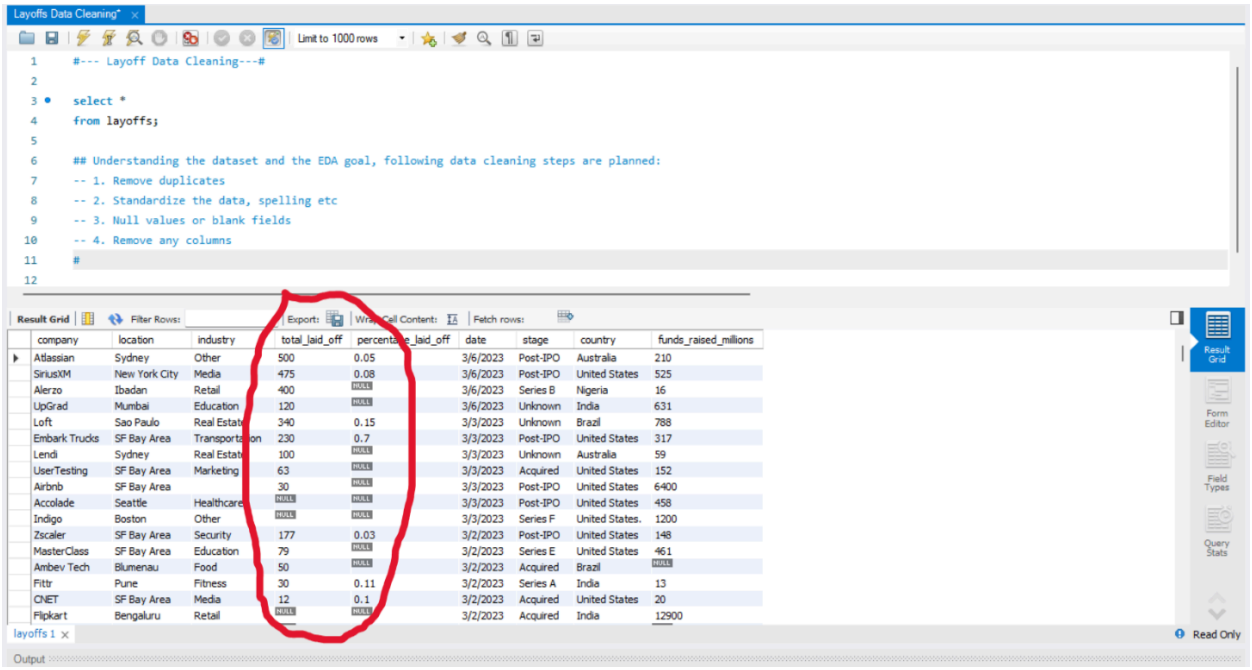
- Removed exact duplicate records using ROW_NUMBER()
- Standardized inconsistent entries (e.g., “Crypto” vs “Cryptocurrency”)
- Trimmed extra spaces and trailing characters in text fields
- Converted date fields from text to proper DATE type using STR_TO_DATE()
- Handled and filled missing values where possible
- Deleted rows with null critical values
- Removed unnecessary columns post-cleaning

🔍 SQL Data Cleaning – Step-by-Step Summary

🔹 1. Understanding and Preparing the Dataset

- Loaded the raw layoffs dataset into MySQL.
- Duplicated the dataset into a working table layoffs_duplicate for safe transformation.

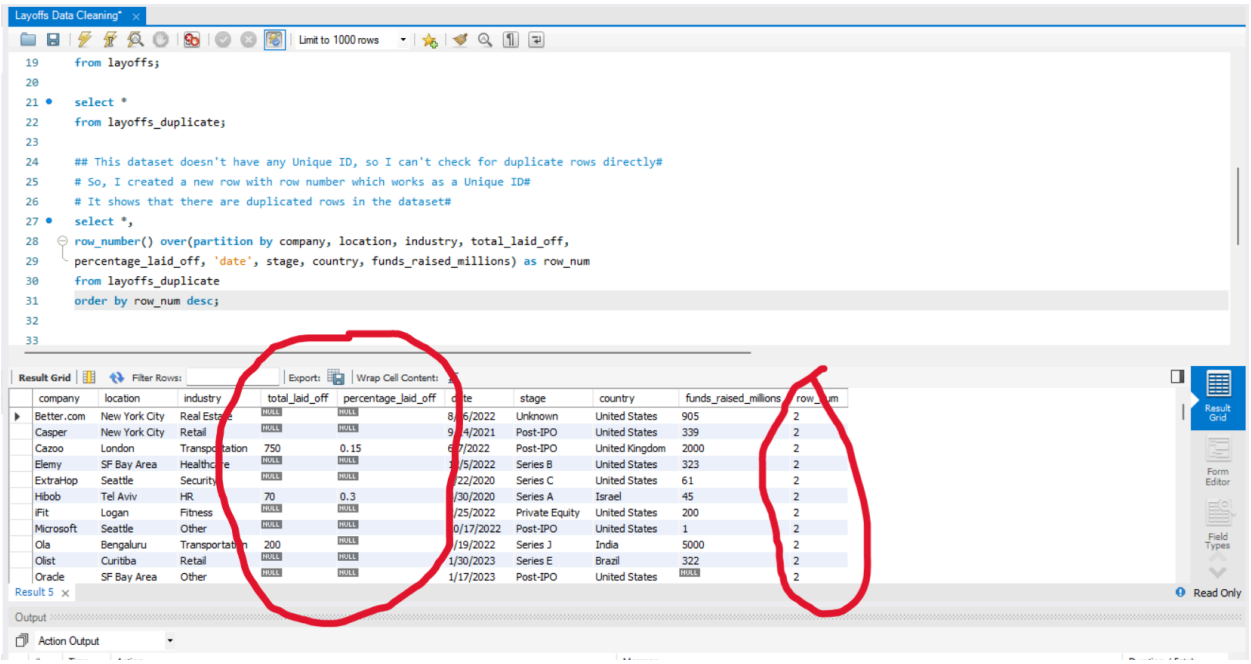
Raw Data Screen:

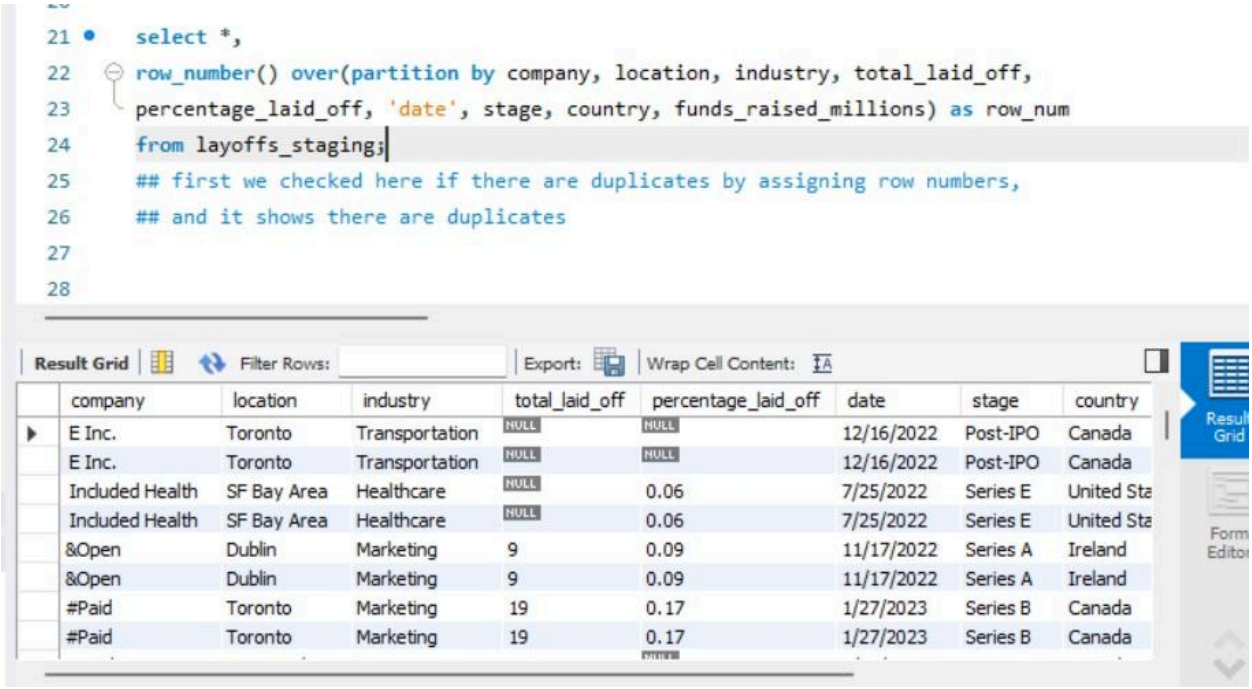


🔹 2. Removing Duplicate Rows

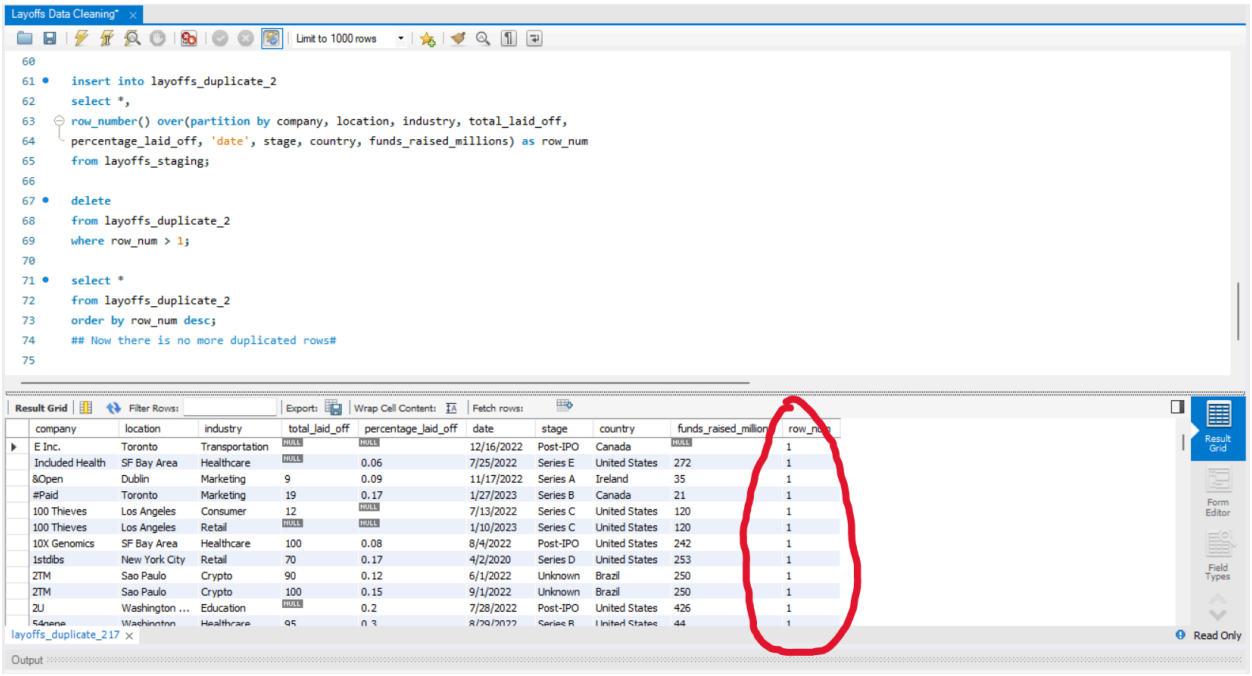
- Since there was no unique ID, used ROW_NUMBER() to identify exact duplicate rows.
- Created a new table layoffs_duplicate_2 with a row_num column.
- Deleted all rows with row_num > 1, keeping only the first occurrence of each record.

Duplicated Rows Identified:





✅ ***Duplicates cleaned successfully.***



🔹 *3. Standardizing Text Fields*

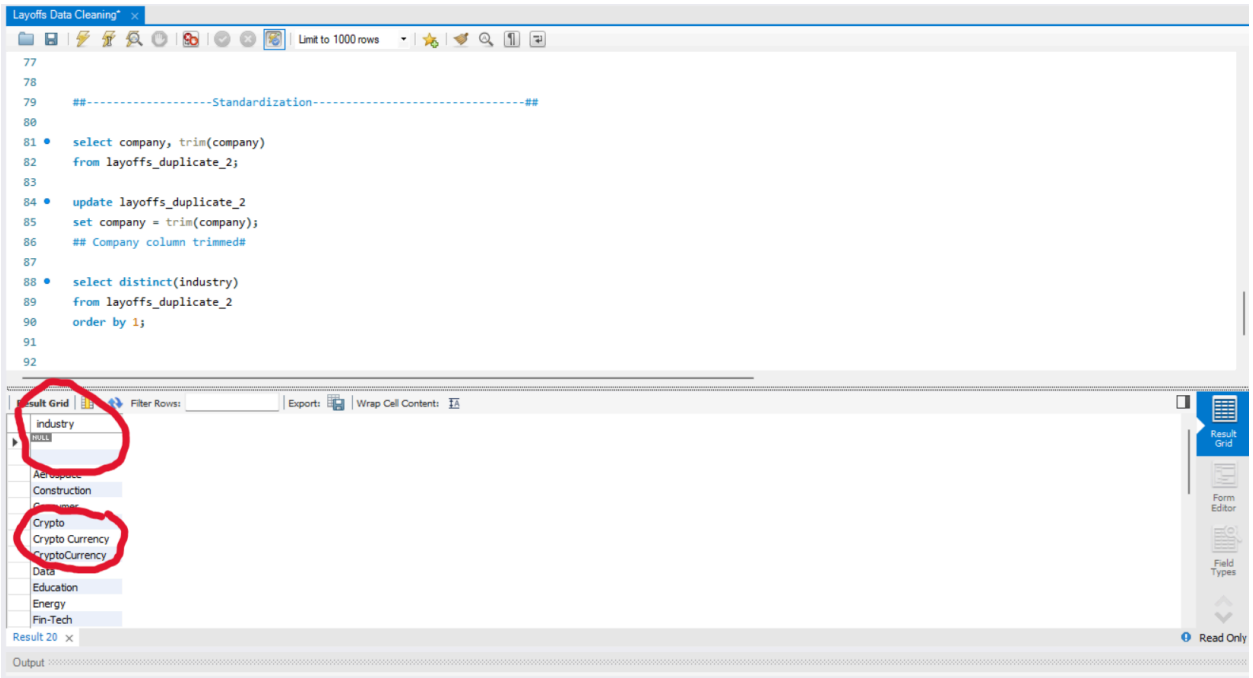
- *Trimmed leading/trailing spaces using TRIM().*
- *Fixed inconsistent naming (e.g., “Crypto” vs “Cryptocurrency”) using UPDATE ... WHERE LIKE.*
  Cleaned up country values (e.g., removed trailing dots in “United States.”)

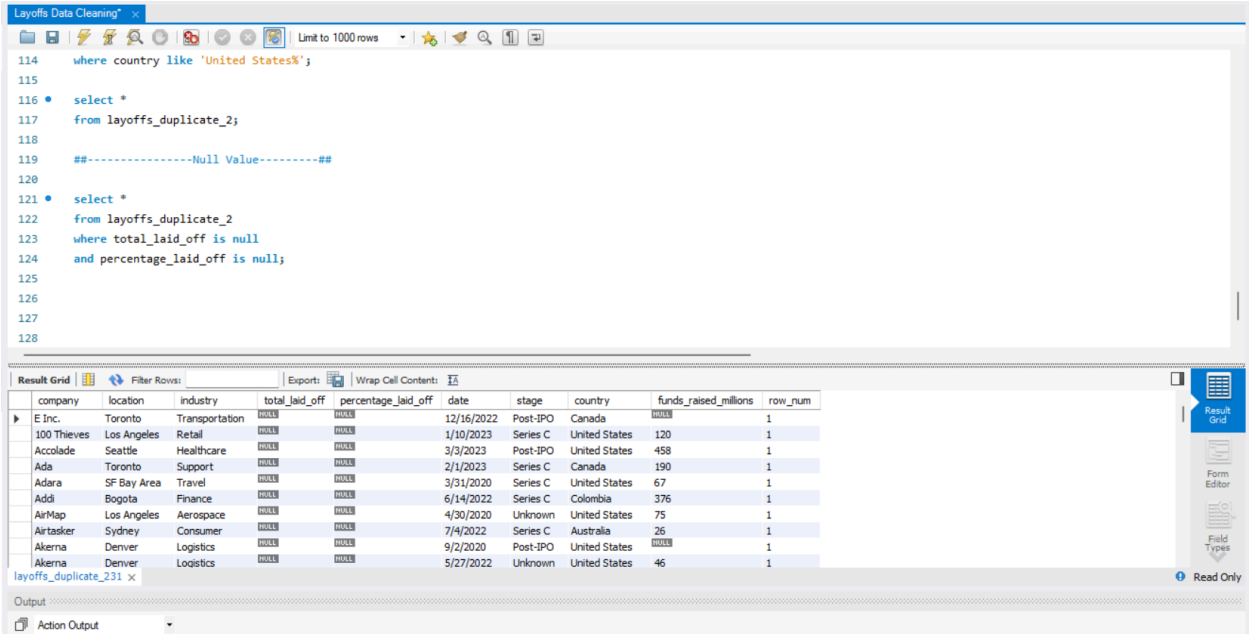
✅ *Text values standardized for consistent grouping.*

🔹 4. Handling Null and Blank Values

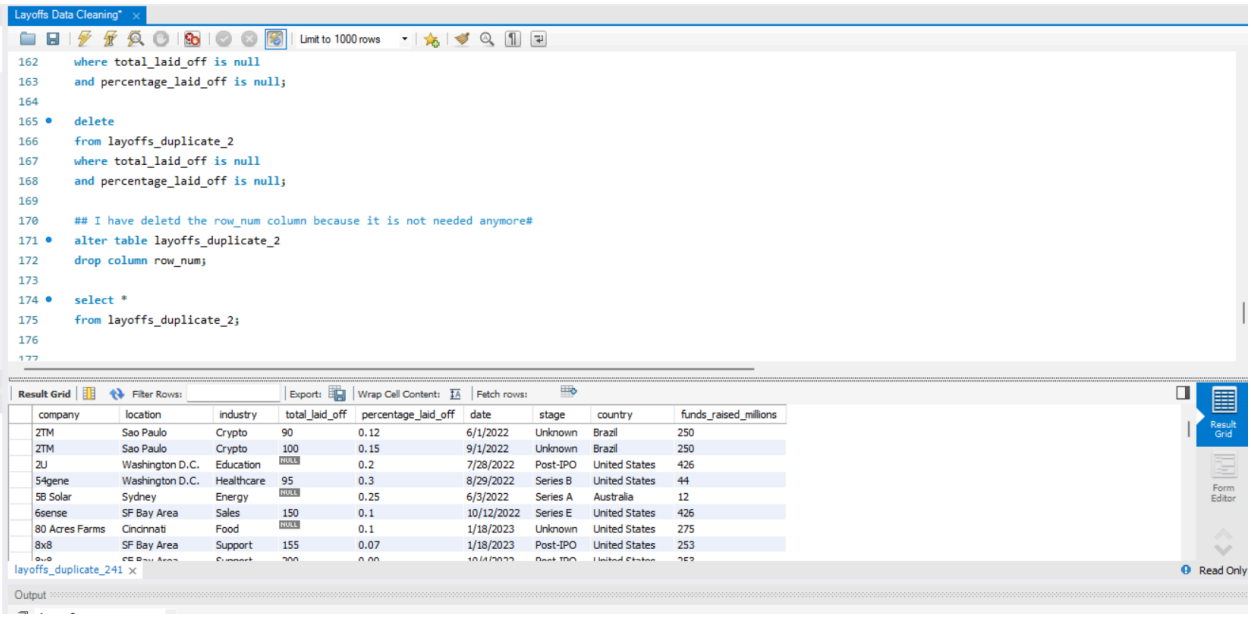
- Deleted rows where both total_laid_off and percentage_laid_off were NULL.
- Used a SELF JOIN to impute missing industry values from matching companies.
- Where no imputation was possible (e.g., “Bally’s”), left NULL or deleted.

Null and Blank Fields Screenshot: *Null values that still remain are accepted in this case.*





✅ ***Missing data handled appropriately.***



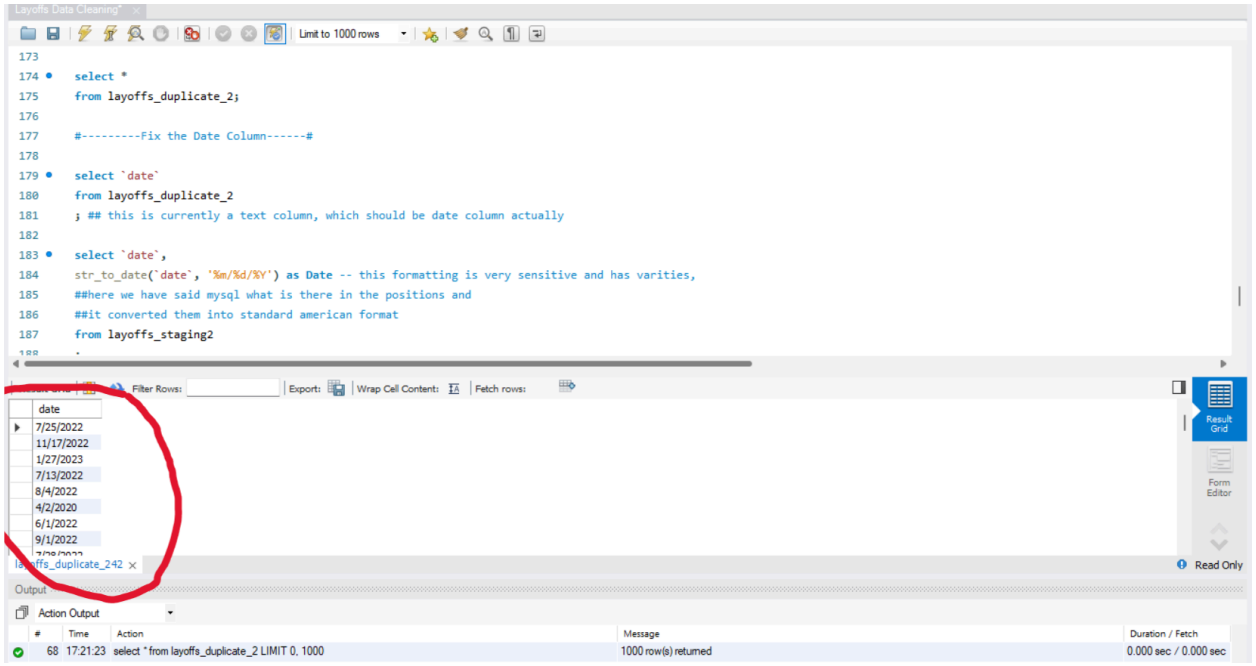
```
where total_laid_off is null
and percentage_laid_off is null;

delete
from layoffs_duplicate_2
where total_laid_off is null
and percentage_laid_off is null;

## I have deletd the row_num column because it is not needed anymore#
alter table layoffs_duplicate_2
drop column row_num;

select *
from layoffs_duplicate_2;
```

company	location	industry	total_laid_off	percentage_laid_off	date	stage	country	funds_raised_millions
ZTM	Sao Paulo	Crypto	90	0.12	6/1/2022	Unknown	Brazil	250
ZTM	Sao Paulo	Crypto	100	0.15	9/1/2022	Unknown	Brazil	250
2U	Washington D.C.	Education	NULL	0.2	7/28/2022	Post-IPO	United States	426
54gene	Washington D.C.	Healthcare	95	0.3	8/29/2022	Series B	United States	44
5B Solar	Sydney	Energy	NULL	0.25	6/3/2022	Series A	Australia	12
6sense	SF Bay Area	Sales	150	0.1	10/12/2022	Series E	United States	426
80 Acres Farms	Cincinnati	Food	NULL	0.1	1/18/2023	Unknown	United States	275
8x8	SF Bay Area	Support	155	0.07	1/18/2023	Post-IPO	United States	253

🔹 5. Fixing the Date Format

- The original date column was stored as text (MM/DD/YYYY).
- Converted it using STR_TO_DATE().
- Altered the column type to DATE.

Date Column was not appropriate: *see next page*

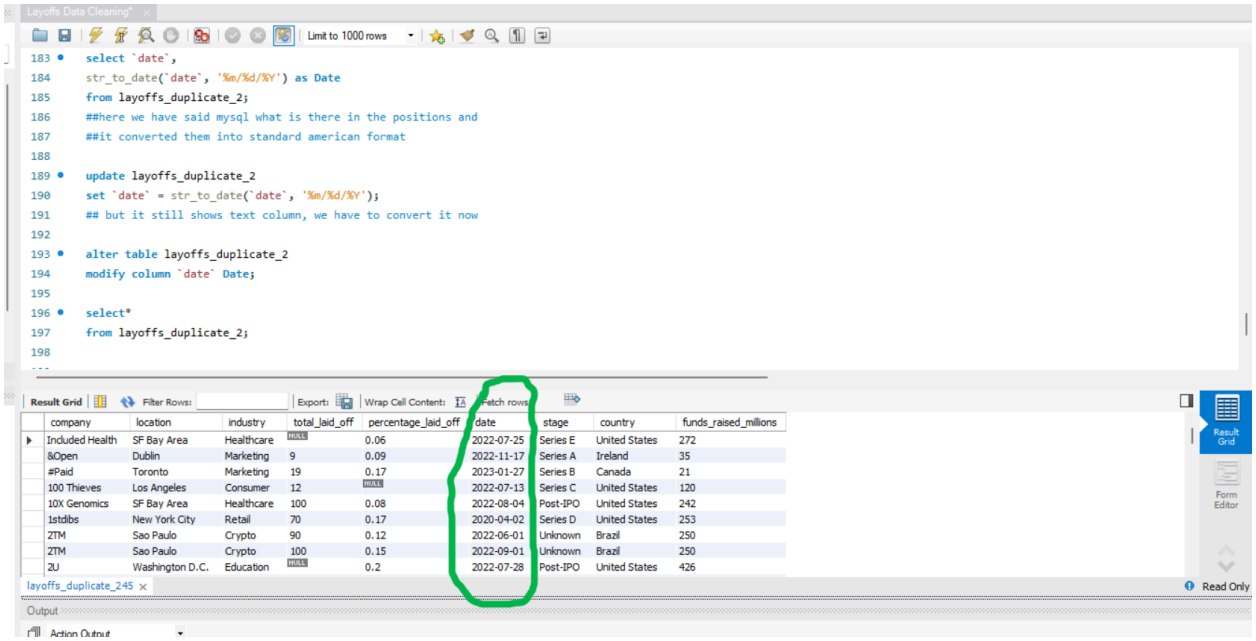


```
select *
from layoffs_duplicate_2;

#---------Fix the Date Column------#

select `date`
from layoffs_duplicate_2
; ## this is currently a text column, which should be date column actually

select `date`,
str_to_date(`date`, '%m/%d/%Y') as Date -- this formatting is very sensitive and has varities,
##here we have said mysql what is there in the positions and
##it converted them into standard american format
from layoffs_staging2
```

date
7/25/2022
11/17/2022
1/27/2023
7/13/2022
8/4/2022
4/2/2020
6/1/2022
9/1/2022

#	Time	Action	Message	Duration / Fetch
68	17:21:23	select * from layoffs_duplicate_2 LIMIT 0, 1000	1000 row(s) returned	0.000 sec / 0.000 sec

✅ ***Date field converted and ready for time-series or trend analysis.***

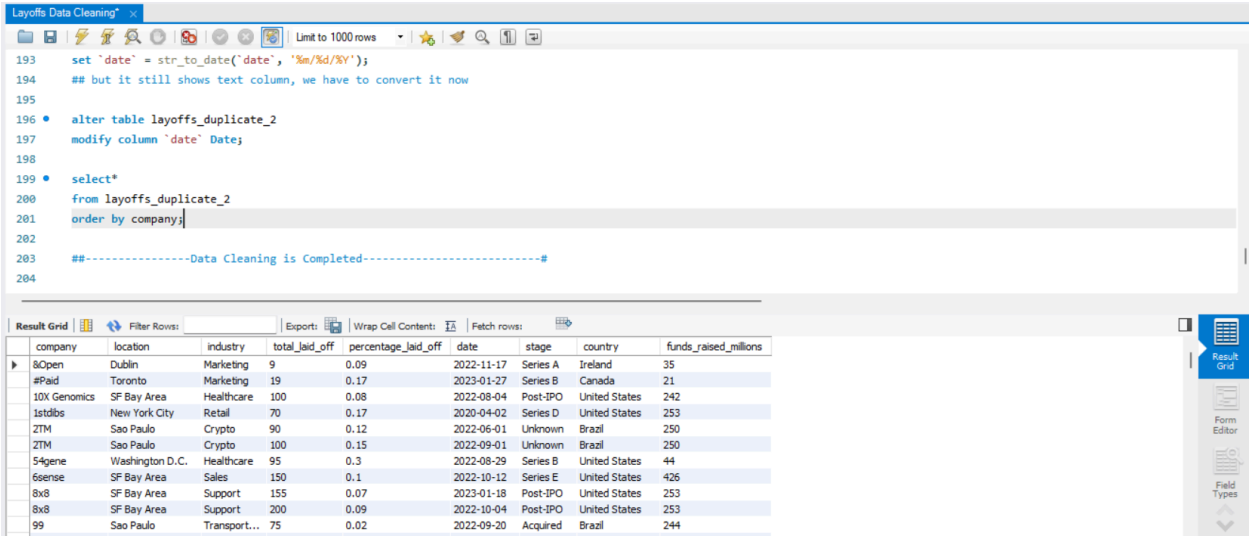


```
183 •  select `date`,
184    str_to_date(`date`, '%m/%d/%Y') as Date
185    from layoffs_duplicate_2;
186    ##here we have said mysql what is there in the positions and
187    ##it converted them into standard american format
188
189 •  update layoffs_duplicate_2
190    set `date` = str_to_date(`date`, '%m/%d/%Y');
191    ## but it still shows text column, we have to convert it now
192
193 •  alter table layoffs_duplicate_2
194    modify column `date` Date;
195
196 •  select*
197    from layoffs_duplicate_2;
198
```

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

🔹 6. Finalizing the Dataset

- Dropped unnecessary columns like row_num.
- Verified the dataset was cleaned, structured, and ready for analysis or dashboard integration.



🎯 *Outcome*

The cleaned dataset is fully prepared for use in business dashboards, trend analysis, and financial or operational reporting.

👉 Refer to my Projects section for the SQL EDA (Exploratory Data Analysis) project using this cleaned dataset.
