Data Cleaning Procedure

# Overview:

* Analyses a dataset of **2,500 records** capturing layoffs from **2,000 companies** across **60 countries**.
* Key columns: *company, location, industry, total\_laid\_off, percentage\_laid\_off, date, stage, country, and funds\_raised\_millions*.
* Transforms raw, messy data into a polished and analysis-ready dataset, ensuring it’s clean, consistent, and reliable for future insights.
* Ensures data accuracy and readiness for analysis through **4-part cleaning process**:
  1. **Remove Duplicates**
  2. **Standardize the Data** (spell-checking)
  3. **Handle NULL/Blank Values**
  4. **Delete Unnecessary Data**

# *Remove Duplicates:*

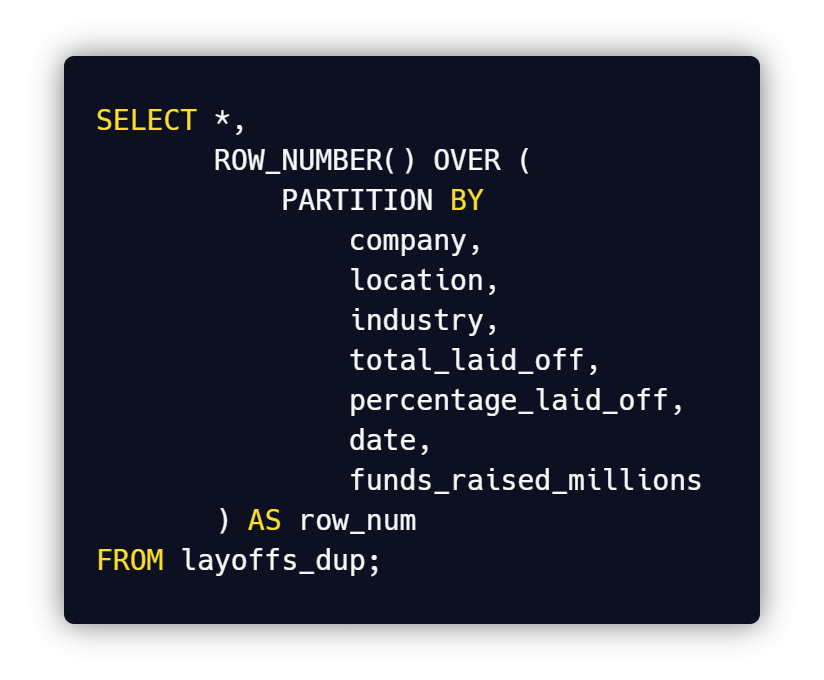
# Create a duplicate table:

* Preserves Data Integrity: Keeps the original data safe for reference or rollback.
* Enables Traceability: Facilitates comparison between cleaned and raw data.
* Supports Experimentation: Allows testing cleaning methods without risking the source data.
* Optimizes Performance: Refined data can be indexed for faster queries.

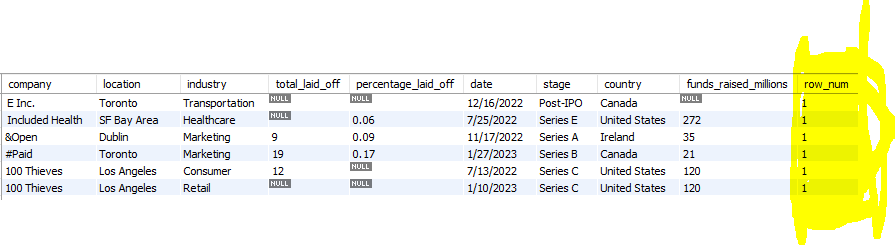


# Deleting Duplicate Records:

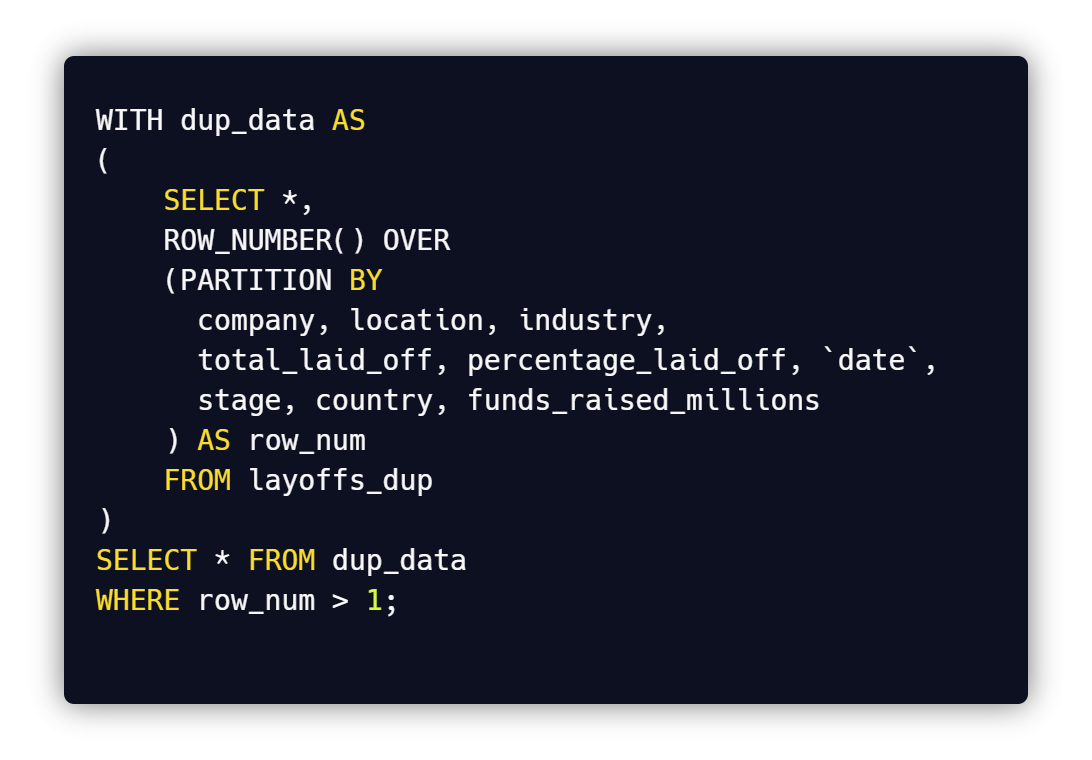
* Creating a ‘duplicate indicator’ column:
* To delete duplicate records, we first have to create a new column that acts as an indicator if a record is duplicate or not.
* We can do that with the window function **ROW\_NUMBER ()**



* This creates a separate column which identifies unique rows by assigning them with “1” and duplicates with values greater than “1”



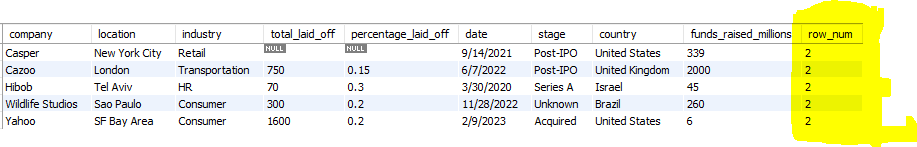
iii. Use ROW\_NUMBER () to identify duplicates:  
Modify the query to find records whose “row\_num” is greater 1 (Use either Subqueries or Common Table Expressions (CTEs)):



## Create another duplicate table for safe deletion of duplicate records:

* Since this step involves deletion of data, create another table to ensure safe deletion of files
* Add the ‘*layoffs\_dup’* data into the new table and rename it ‘***layoffs\_dup2’***





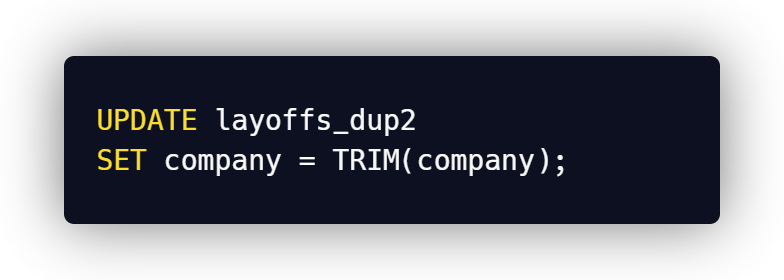
* Execute the same queries to find the duplicate records and delete those records from *‘layoffs\_dup2’*



# *Standardizing the data:*

## Use TRIM () Function to remove the gaps before the names:

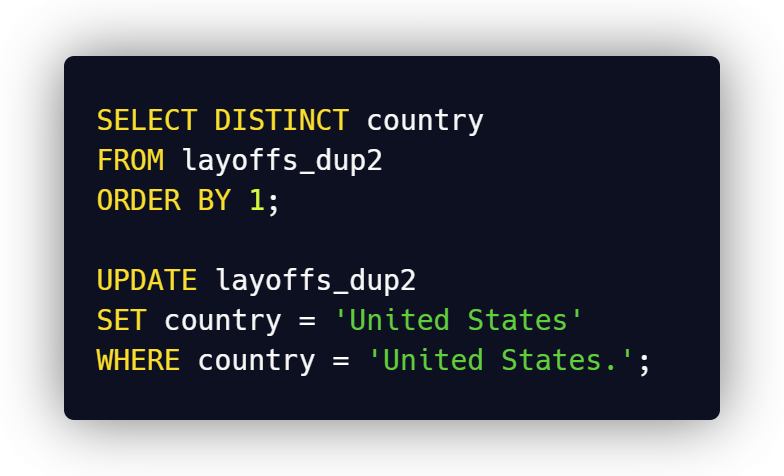
* TRIM () function removes spaces present before and after a company name



## Merge similarly named industries like ' crypto', 'crypto currency', 'crypto-currency':



## Look for more unclean data in ‘countries’ columns with DISTINCT ():



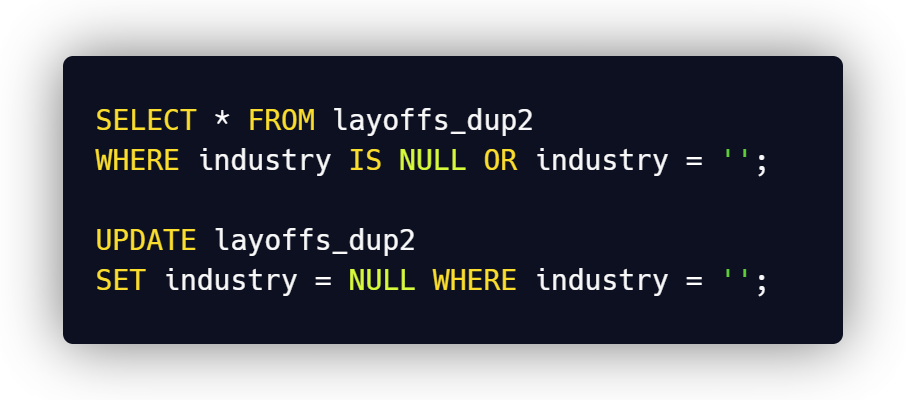
## Change formats of some columns like ‘date’:

* First convert the text in the ‘date’ column into data datatype with the help of **‘STR\_TO\_DATE ()’** function
* Then update the column values with the same **‘STR\_TO\_DATE ()’**
* Finally, change the datatype of the ‘date’ column from text to date datatype:



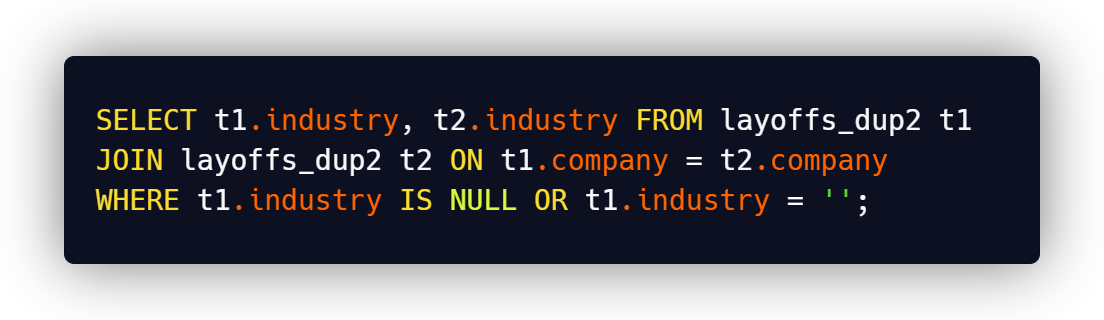
# *Dealing with NULL values*

## Converting blanks into null values for consistency:



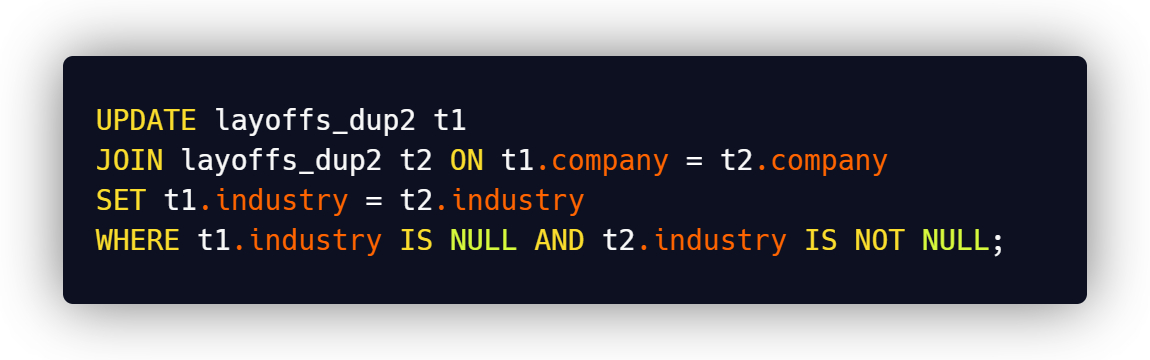
## Check for Possible Replacements:

* Find existing values for the same company to use as replacements



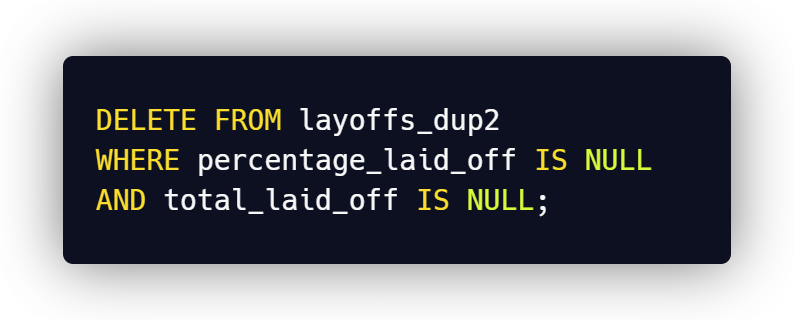
## Fill missing values:

* Update NULL industries using available values from the same company



# *Deleting unnecessary data:*

## Deleting records that don’t play a big role in analysis:



## Dropping the ‘row\_num’ column:

* Row\_num has been used just to identify the duplicate records, which isn’t of a big use

