SQL Data Cleaning Project – Step-by-Step Process

Step 1: Download Dataset

- Downloaded the Layoffs dataset (CSV) from Kaggle.
- The dataset contained real-world company layoff data, including columns like:
    - company, location, industry, total_laid_off, percentage_laid_off, date, stage, country, funds_raised_millions.

Step 2: Create Database and Import Dataset

- Created a new database in MySQL:
- CREATE DATABASE layoffs_project;
- USE layoffs_project;
- Imported layoffs.csv into MySQL using Table Data Import Wizard in MySQL Workbench.
- Verified data:
- SELECT * FROM layoffs LIMIT 10;

Step 3: Create Staging Table (Safe Copy)

- Created a staging table to avoid modifying raw data:
- CREATE TABLE layoffs_staging LIKE layoffs;
- INSERT INTO layoffs_staging SELECT * FROM layoffs; • Confirmed data count matched the original.

Step 4: Remove Duplicates

- Identified duplicates using CTE and ROW_NUMBER:

```
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_staging

)

SELECT * FROM duplicate_cte WHERE row_num > 1;

- Created a second staging table layoffs_staging2 with row_num column.
- Deleted rows where row_num > 1:
- DELETE FROM layoffs_staging2 WHERE row_num > 1;

Step 5: Standardize Data

- Trimmed spaces from text columns:
- UPDATE layoffs_staging2 SET company = TRIM(company);
- UPDATE layoffs_staging2 SET industry = TRIM(industry);
- Standardized industry names:
- UPDATE layoffs_staging2 SET industry = 'Crypto'
- WHERE industry LIKE 'Crypto%';
- Fixed country names:
- UPDATE layoffs_staging2
- SET country = TRIM(TRAILING '.' FROM country)
- WHERE country LIKE 'United States%';

Step 6: Fix Date Column

- Converted date column from text to DATE:
- UPDATE layoffs_staging2

- SET date = STR_TO_DATE(date, '%m/%d/%Y');
- ALTER TABLE layoffs_staging2
- MODIFY COLUMN date DATE;

Step 7: Handle Null and Blank Values

- Converted blanks to NULL:
- UPDATE layoffs_staging SET industry = NULL WHERE industry = '';
- Populated missing industry values using self-join:
- UPDATE layoffs_staging t1
- JOIN layoffs_staging 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;
- Deleted rows with null layoffs and percentage:
- DELETE FROM layoffs_staging2
- WHERE total_laid_off IS NULL AND percentage_laid_off IS NULL;

Step 8: Drop Temporary Columns

- Removed row_num column after cleaning:
- ALTER TABLE layoffs_staging2 DROP COLUMN row_num;

Step 9: Validate Cleaned Data

- Checked for duplicates:
- SELECT company, COUNT(*) FROM layoffs_staging2

- GROUP BY company HAVING COUNT(*) > 1;
- Verified unique industry and country values:
- SELECT DISTINCT industry FROM layoffs_staging2;
- SELECT DISTINCT country FROM layoffs_staging2;

Step 10: Final Clean Table

- Delivered a fully cleaned and optimized table layoffs_staging2:
  No duplicates
  Standardized industry/country names
  Converted date column
  Nulls handled and irrelevant rows removed

Conclusion:

- This project demonstrated a complete end-to-end data cleaning workflow using SQL on a real-world layoffs dataset. By systematically removing duplicates, standardizing inconsistent fields, converting data types, handling null values, and optimizing the table structure, I transformed a raw, unstructured dataset into a clean, analysis-ready dataset.
- The process showcased practical skills in SQL (CTEs, window functions, joins, string and date operations) and reflected how data cleaning is a critical step in preparing data for meaningful business insights and reporting.
- With this cleaned dataset, businesses and analysts can now confidently perform trend analysis (e.g., layoffs by industry or country), build visual dashboards, and derive actionable insights without concerns about data accuracy or integrity.
- This project highlights my ability to handle real-world data quality issues, automate cleaning workflows in SQL, and deliver structured datasets suitable for advanced analytics and decision-making.