**-- Switch to the database**

USE world\_Layoffs;

**-- Initial Data Inspection**

SELECT \* FROM layoffs;

**-- Create a new staging table with the same structure as the original table**

CREATE TABLE layoffs\_staging LIKE layoffs;

**-- Verify staging table structure**

SELECT \* FROM layoffs\_staging;

**-- Insert data from original table into staging table**

INSERT INTO layoffs\_staging

SELECT \* FROM layoffs;

**-- Check data in the staging table**

SELECT \* FROM layoffs\_staging;

**-- Identify duplicate entries with row numbers for duplicate removal**

SELECT \*,

ROW\_NUMBER() OVER(PARTITION BY Company, Industry, Total\_Laid\_off, percentage\_laid\_Off, `Date`) AS Row\_Num

FROM layoffs\_staging;

**-- CTE for identifying duplicates**

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;

**-- Create a second staging table to store cleaned data**

CREATE TABLE layoffs\_staging2 (

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;

**-- Verify table structure**

SELECT \* FROM layoffs\_staging2;

**-- Insert data into the second staging table, with row numbers for duplicates**

INSERT INTO layoffs\_staging2

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;

**-- Check for duplicates by row number**

SELECT \* FROM layoffs\_staging2

WHERE row\_num > 1;

**-- Remove duplicate rows based on row number**

DELETE FROM layoffs\_staging2

WHERE row\_num > 1;

**-- Verify clean data after duplicate removal**

SELECT \* FROM layoffs\_staging2;

**-- Standardize company names by trimming whitespace**

UPDATE layoffs\_staging2

SET company = TRIM(company);

**-- Check standardized company names**

SELECT DISTINCT industry

FROM layoffs\_staging2

ORDER BY 1;

**-- Standardize industry names (e.g., unify 'Crypto%' variations to 'Crypto')**

UPDATE layoffs\_staging2

SET industry = 'Crypto'

WHERE industry LIKE 'Crypto%';

**-- Verify industry standardization**

SELECT DISTINCT industry

FROM layoffs\_staging2

ORDER BY 1;

**-- Standardize country names by trimming trailing periods**

UPDATE layoffs\_staging2

SET country = TRIM(TRAILING '.' FROM country)

WHERE country LIKE 'United States%';

**-- Verify country standardization**

SELECT DISTINCT country

FROM layoffs\_staging2

ORDER BY 1;

**-- Convert `date` column format from text to SQL DATE format**

UPDATE layoffs\_staging2

SET `date` = STR\_TO\_DATE(`date`, '%m/%d/%Y');

**-- Modify `date` column type to DATE**

ALTER TABLE layoffs\_staging2

MODIFY COLUMN `date` DATE;

**-- Verify date format conversion**

SELECT \* FROM layoffs\_staging2;

**-- Handle null or blank industry values by updating from other rows**

UPDATE layoffs\_staging2 T1

JOIN layoffs\_staging2 T2

ON T1.company = T2.company

SET T1.industry = T2.industry

WHERE T1.industry IS NULL

AND T2.industry IS NOT NULL;

**-- Delete rows where both `Total\_Laid\_off` and `percentage\_laid\_off` are NULL**

DELETE FROM layoffs\_staging2

WHERE total\_laid\_off IS NULL

AND percentage\_laid\_off IS NULL;

**-- Final cleanup: remove the `row\_num` column**

ALTER TABLE layoffs\_staging2

DROP COLUMN row\_num;

**-- Final check on cleaned data**

SELECT \* FROM layoffs\_staging2;