-- Data Cleaning

CREATE DATABASE layoff\_db;

USE layoff\_db;

-- 0. Load initial data

SELECT \* FROM layoffs;

-- 1. Remove Duplicates

-- Create staging table

CREATE TABLE layoff\_staging AS

SELECT \* FROM layoffs;

SELECT \* FROM layoff\_staging;

-- Identify duplicates using ROW\_NUMBER

WITH cte\_layoff AS (

SELECT \*,

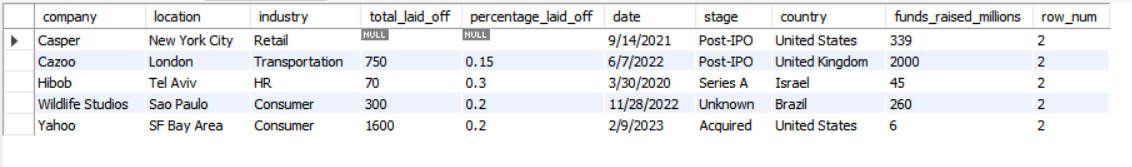
ROW\_NUMBER() OVER (PARTITION BY company, location, industry, total\_laid\_off, percentage\_laid\_off,

`date`, stage, country, funds\_raised\_millions) AS row\_num

FROM layoff\_staging

)

SELECT \* FROM cte\_layoff WHERE row\_num > 1;



-- Delete duplicates

WITH cte\_layoff 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 layoff\_staging

)

DELETE FROM cte\_layoff WHERE row\_num > 1;

-- Create layoff\_staging2 table with row\_num

CREATE TABLE layoff\_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

);

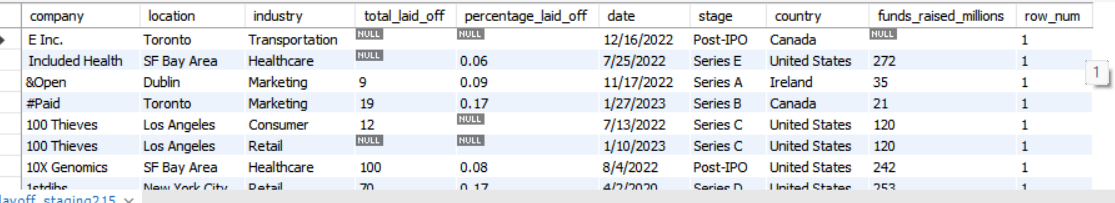
-- Insert cleaned data into layoff\_staging2

INSERT INTO layoff\_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 layoff\_staging;



-- Disable safe updates

SET SQL\_SAFE\_UPDATES = 0;

-- Remove duplicates from layoff\_staging2

DELETE FROM layoff\_staging2 WHERE row\_num > 1;

-- 2. Standardize Data

-- Trim company name whitespace

UPDATE layoff\_staging2 SET company = TRIM(company);

-- Standardize 'Crypto' industry name

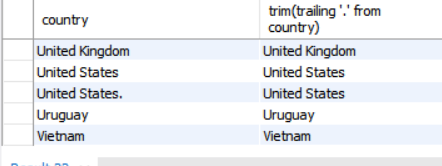
SELECT industry FROM layoff\_staging2 WHERE industry LIKE 'Crypto%';

UPDATE layoff\_staging2 SET industry = 'Crypto' WHERE industry LIKE 'Crypto%';

-- Standardize country names by removing trailing '.'

SELECT DISTINCT country, TRIM(TRAILING '.' FROM country) FROM layoff\_staging2 ORDER BY 1;

UPDATE layoff\_staging2 SET country = TRIM(TRAILING '.' FROM country) WHERE country LIKE '%.';



-- Convert date from text to proper DATE format

SELECT `date`, STR\_TO\_DATE(`date`, '%m/%d/%Y') FROM layoff\_staging2;

UPDATE layoff\_staging2 SET `date` = STR\_TO\_DATE(`date`, '%m/%d/%Y');

ALTER TABLE layoff\_staging2 MODIFY `date` DATE;

-- 3. Handle Null or Blank Values

-- Find rows with both total\_laid\_off and percentage\_laid\_off NULL

SELECT COUNT(\*) FROM layoff\_staging2 WHERE total\_laid\_off IS NULL AND percentage\_laid\_off IS NULL;

-- Find and replace blank industry with NULL

SELECT \* FROM layoff\_staging2 WHERE industry IS NULL OR industry = '';

UPDATE layoff\_staging2 SET industry = NULL WHERE industry = '';

-- Use self join to impute missing industry

SELECT \* FROM layoff\_staging2 t1

JOIN layoff\_staging2 t2 ON t1.company = t2.company

WHERE (t1.industry IS NULL OR t1.industry = '') AND t2.industry IS NOT NULL;

UPDATE layoff\_staging2 t1

JOIN layoff\_staging2 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;

-- Delete fully NULL layoff rows

DELETE FROM layoff\_staging2 WHERE total\_laid\_off IS NULL AND percentage\_laid\_off IS NULL;

-- 4. Remove row\_num column after cleaning

ALTER TABLE layoff\_staging2 DROP COLUMN row\_num;