**-- 02\_data\_cleaning.sql**

**-- Purpose: Data cleaning and preparation**

**USE layoffs\_project;**

**-- 1. Create staging table (safe copy)**

CREATE TABLE IF NOT EXISTS layoffs\_staging LIKE layoffs;

INSERT INTO layoffs\_staging

SELECT \*

FROM layoffs;

**-- 2. Remove duplicates using ROW\_NUMBER**

CREATE TABLE layoffs\_staging2 AS

SELECT \*

FROM (

SELECT \*,

ROW\_NUMBER() OVER (

PARTITION BY company, location, industry, total\_laid\_off,

percentage\_laid\_off, date, stage, country,

funds\_raised\_millions

ORDER BY (SELECT NULL)

) AS row\_num

FROM layoffs\_staging

) t;

DELETE FROM layoffs\_staging2

WHERE row\_num > 1;

ALTER TABLE layoffs\_staging2

DROP COLUMN row\_num;

**-- 3. Trim spaces**

UPDATE layoffs\_staging2

SET company = TRIM(company);

UPDATE layoffs\_staging2

SET industry = TRIM(industry);

UPDATE layoffs\_staging2

SET country = TRIM(country);

**-- 4. Standardize industry & country**

UPDATE layoffs\_staging2

SET industry = 'Crypto'

WHERE industry LIKE 'Crypto%';

UPDATE layoffs\_staging2

SET country = 'United States'

WHERE country LIKE 'United States%';

**-- 5. Convert date column**

UPDATE layoffs\_staging2

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

ALTER TABLE layoffs\_staging2

MODIFY COLUMN date DATE;

**-- 6. Handle nulls and blanks**

UPDATE layoffs\_staging2

SET industry = NULL

WHERE industry = '';

UPDATE layoffs\_staging2

SET total\_laid\_off = NULL

WHERE total\_laid\_off < 0;

**-- 7. Fill missing industry using self-join**

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;