SHOW DATABASES

select \*

from layoffs;

-- 1. Remove Duplicates

-- 2. Standardize the Data

-- 3. Null Values or Blank Values

-- 4. Remove Unecassary Columns

create table layoffs\_staging

like layoffs;

select \*

from layoffs\_staging

insert layoffs\_staging

select \*

from layoffs

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;

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;

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

)

DELETE

FROM duplicate\_cte

where row\_num >1;

CREATE TABLE `layoffs\_staging2` (

`company` text,

`location` text,

`industry` text,

`total\_laid\_off` bigint 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;

SELECT \*

FROM layoffs\_staging2

WHERE row\_num > 1;

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

SELECT \*

FROM layoffs\_staging2

WHERE row\_num > 1;

-- 2 STANDARDIZING DATA, FINDING ISSUES INT OUR DATA AND FIXING IT.

-- REMOVE WHITESPACES BEFORE AND AFTER TEXT IF EXISTS

select company, trim(company)

from layoffs\_staging2;

update layoffs\_staging2

set company = trim(company);

-- REMOVE DUPLICATES OR REDUNDENCY IN THE "INDUSTRY" COLUMN (SAME INDUSRTY WITH TWO DIFFERENT NAMES)

SELECT distinct industry

FROM layoffs\_staging2

UPDATE layoffs\_staging2

SET industry = 'Crypto'

where industry LIKE 'Crypto%';

-- BRIEF CHECK THROUGH THE "location" COLUMN AND EVERYTHING LOOKS GOOD.

SELECT DISTINCT location

from layoffs\_staging2

order by 1

-- SYNCRONYZE SPELLING OF "country" NAMES TO AVOID REDUNDENCY OR DUPLICATES

SELECT DISTINCT country

from layoffs\_staging2

order by 1;

SELECT DISTINCT country

from layoffs\_staging2

where country like 'United States%'

order by 1;

SELECT DISTINCT country, trim(trailing '.' from country)

from layoffs\_staging2

order by 1;

update layoffs\_staging2

set country = trim(TRAILING '.' from country)

where country LIKE 'United States%';

SELECT \*

from layoffs\_staging2

-- CHANGE fORMAT OF 'date' COLUMN

SELECT `date`,

str\_to\_date(`date`, '%m/%d/%Y')

from layoffs\_staging2;

update layoffs\_staging2

set `date` = str\_to\_date(`date`, '%m/%d/%Y')

SELECT `date`

from layoffs\_staging2;

Alter table layoffs\_staging2

modify column `date` DATE;

-- WORKING WITH NULL AND BLANK VALUES

SELECT \*

FROM layoffs\_staging2

WHERE total\_laid\_off IS null

AND percentage\_laid\_off is null;

SELECT \*

FROM layoffs\_staging2

WHERE industry is null

or industry = ''

SELECT \*

FROM layoffs\_staging2

where company = 'Airbnb';

select \*

from layoffs\_staging2 t1

join layoffs\_staging2 t2

on t1.company = t2.company

where (t1.industry is null or t1.industry = '')

and t2.industry is not null

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;

update layoffs\_staging2

set industry = null

where industry = '';

select \*

from layoffs\_staging2

delete

from layoffs\_staging2

where total\_laid\_off IS NULL

and percentage\_laid\_off IS NULL;

alter table layoffs\_staging2

drop column row\_num

-- THERE ARE PLENTY OF REMAINING NULL VALUES IN THE TABLE BUT WE CANNOT POPULATE THEM BASED ON THE DATA THAT WE HAVE

-- MEANING WE CANNOT RUN AGGREGATE QUERIES TO POPULATE THESE NULL CELLS, NIETHER CAN WE POPULATE THEM BASED ON PREDICTIONS

-- FOUND USING OTHER RELATED COLUMNS,SO OUR BEST OPTION IS TO LEAVE THE REMAINING NULL VALUES AS IS.

-- THANK YOU AND PLEASE SEE THE NEXT DATA EXPLORATORY PROJECT BASED ON OUR NEWLY CLEANED DATASHEET