-- Importing the data that is related to layoffs that started after CORONA. The task is

-- 1. Remove Duplicates

-- 2. Standardize the data

-- 3. Removing any extra columns

-- 4. Handelling NULL values

-- 1. Removing duplicates

-- uaing row\_number giving 1 to each unique row. if there are duplicate rows then the duplicate row will be populated as 2

select l.\*,

row\_number() over( partition by company, location, industry,total\_laid\_off, percentage\_laid\_off, dates, stage, country order by company) as row\_num

from layoff l;

-- Duplicate rows

with dups as

(

select l.\*,

row\_number() over( partition by company, location, industry,total\_laid\_off, percentage\_laid\_off, dates, stage, country order by company) as row\_num

from layoff l

)

select \* from dups where row\_num >1

order by company;

-- Table with unique rows

with dups as

(

select l.\*,

row\_number() over( partition by company, location, industry,total\_laid\_off, percentage\_laid\_off, dates, stage, country order by company) as row\_num

from layoff l

)

select \* from dups where row\_num = 1

order by company;

-- Now assign the unique rows to a diffrent table and that will be a table with no duplicate rows.

create table layoffs as

with dups as

(

select l.\*,

row\_number() over( partition by company, location, industry,total\_laid\_off, percentage\_laid\_off, dates, stage, country order by company) as row\_num

from layoff l

)

select \* from dups where row\_num = 1

order by company;

-- New Table Layoffs. 2356 Rows

select \* from layoffs;

select count(\*) from layoffs;

-- 2. Standardize the data

-- There are starting and training spaces in the company column. Removing that with trim function

select company from layoffs;

update layoffs

set company = trim(company);

-- Checking distinct values in the locationa and industry column and removing any inaccuracies

select

distinct location

from layoffs

order by 1;

select

distinct country

from layoffs

order by 1;

-- Replacing United States. with United States

update layoffs

set country = 'United States'

where country = 'United States.';

-- Dates column includes the dates values but the current data type is text.

Describe layoffs;

-- To correct5 this, need to update a new colum then convert the date value to date type and input the value in the new colum then deleting the old column

ALTER TABLE layoffs

ADD Date1 DATE;

UPDATE layoffs

SET Date1 = TO\_DATE(Dates, 'MM/DD/YYYY');

alter table layoffs

drop column dates;

alter table layoffs

rename column date1 to dates;

select \* from layoffs;

--adding two new colum laid\_off and laid\_off\_perc and deleting two old colums total\_laid\_off and percentage\_laid\_off.

describe layoffs;

alter table layoffs

modify total\_laid\_off number;

alter table layoffs

add laid\_off number;

update layoffs

set laid\_off = total\_laid\_off;

describe layoffs

alter table layoffs

drop column total\_laid\_off;

select \* from layoffs;

alter table layoffs

add laid\_off\_perc number(4,2);

update layoffs

set laid\_off\_perc = percentage\_laid\_off;

describe layoffs;

alter table layoffs

drop column percentage\_laid\_off;

select \* from layoffs;

-- 3. Removing any extra columns

alter table layoffs

drop column row\_num;

-- 4. Handelling NULL values

-- There are various rows in the data set where total\_laid\_off and percentage\_laid\_off rows are blank. These are of no use. SO deleting those rows.

select \* from layoffs

where total\_laid\_off is null and

percentage\_laid\_off is null;

delete from layoffs

where total\_laid\_off is null and

percentage\_laid\_off is null;

select \* from layoffs;

select count(\*) from layoffs;

Commit;

-- 1995 rows left. The data is now cleaned and ready for analysis.

select \* from layoffs;

-- Importing the data that is related to layoffs that started after CORONA. The task is to do exploratory data analysys. The data is already cleaned in the previous project.

select \* from layoffs

where laid\_off\_perc = 1

order by funds desc;

-- COmpany wise layoff

select

company,

sum(laid\_off)

from layoffs

group by company

order by sum(laid\_off) desc;

-- Date range

select

min(dates),

max(dates)

from layoffs;

-- Industry wise layoff

select

industry,

sum(laid\_off) as total\_laid\_off

from layoffs

group by industry

order by 2 desc;

-- Country wise layoff

select

Country,

sum(laid\_off) as total\_laid\_off

from layoffs

group by Country

order by 2 desc;

-- Year wise layoff

select

extract(Year from Dates)as year,

sum(laid\_off) as total\_laid\_off

from layoffs

group by extract(Year from Dates)

order by 2 desc;

-- Month wise layoff

select

extract(Month from Dates)as Month,

sum(laid\_off) as total\_laid\_off

from layoffs

group by extract(Month from Dates)

order by 2 desc;

-- Stage wise layoff

select

Stage,

sum(laid\_off) as total\_laid\_off

from layoffs

group by Stage

order by 2 desc;

-- Rolling Total Layoff

select

extract(month from dates) as month,

sum(laid\_off)

from layoffs

group by extract(month from dates);

SELECT

TO\_CHAR(Dates, 'YYYY-MM') AS month\_year,

SUM(laid\_off) AS total\_laid\_off

FROM

layoffs

GROUP BY

TO\_CHAR(Dates, 'YYYY-MM')

ORDER BY

month\_year;

with running\_ttl as

(

SELECT

TO\_CHAR(Dates, 'YYYY-MM') AS month\_year,

SUM(laid\_off) AS total\_laid\_off

FROM

layoffs

GROUP BY

TO\_CHAR(Dates, 'YYYY-MM')

ORDER BY

month\_year

)

select

month\_year,

total\_laid\_off,

sum(total\_laid\_off) over(order by month\_year) as running\_ttl

from running\_ttl;