***-- Data cleaning in SQL***

use world\_layoffs;

-- Layoffs dataset : https://www.kaggle.com/datasets/swaptr/layoffs-2022

select \* from layoffs;

**-- now when we are data cleaning we usually follow a few steps**

**-- 1. check for duplicates and remove any**

**-- 2. standardize data and fix errors**

**-- 3. Look at null values and see what**

**-- 4. remove any columns and rows that are not necessary**

**-- 1. check for duplicates and remove any**

create table layoffs\_staging

like layoffs;

select \* from layoffs\_staging;

insert into layoffs\_staging

select \* from layoffs;

*-- lets try to find duplicates on the columns like company, industry, total\_laid & date.*

with duplicate\_cte as

(

select \*,

row\_number() over(partition by company, industry,total\_laid\_off, `date`) as row\_num

from layoffs\_staging

)

select \* from duplicate\_cte

where row\_num > 1;

*-- let’s verify whether the above results are really duplicate, lets verify for 'Oda' & others*

select \* from layoffs\_staging where company ='Oda';

select \* from layoffs\_staging where company ='Terminus';

*-- it seems that they are not duplicate, we must find out the duplicates based on the all the columns,*

*-- let’s do it with below CTE*

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;

*-- the above query giving proper duplicates, let’s reconfirm with below queries.*

select \* from layoffs\_staging where company ='Hibob';

select \* from layoffs\_staging where company ='Wildlife Studios';

*-- now we can delete these duplicates, we are trying to delete the duplicates with below CTE,*

*-- but it is not working, as CTE is not updating.*

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;

*-- for this we will create another table which is copy of existing table layoffs\_staging but with additional*

*-- row\_num column which will be indicator for our duplicate records. let’s create it with below query:* (Right click on table -> copy to clipboard -> Create statement)

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;

*-- notice the extra row\_num column added above, the table created will be empty, we need to fill the data in it*

select \* from layoffs\_staging2;

*-- add the data in it with below query*

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;

*-- now we should be good to delete the records which are duplicates*

select \* from layoffs\_staging2

where row\_num > 1;

*-- below query will delete duplicate records*

delete from layoffs\_staging2

where row\_num > 1;  
  
select \* from layoffs\_staging2;

***-- 2. standardize data and fix errors***

*-- let’s find null records in industry*

select \* from layoffs\_staging2

where industry is null or industry =’’;

*-- if you look at Airbnb record, we can see that one row in industry is blank. so we can update it considering others*

select \* from layoffs\_staging2

where company ='Airbnb';

*-- below query just updates ‘ ‘ records with null so that it is easy to filter the table.*

update layoffs\_staging2

set industry = null

where industry ='';

select \* from layoffs\_staging2

where company like 'Bally%';

select \* from layoffs\_staging2

where company ='Carvana';

update layoffs\_staging2

set industry = null

where industry ='';

*-- let’s try to select industry with self join table wherever it is null industry*

select \* from layoffs\_staging2 t1

join layoffs\_staging2 t2

on t1.company = t2.company

where t1.industry is null

and t2.industry is not null;

*-- now this is going to update the empty industry of the company by looking at other values of industry.*

*-- like for Airbnb, it will update 'Travel' in industry.*

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;

*-- verify if it is updated or not.*

select \* from layoffs\_staging2

where company ='Airbnb';

*-- let’s standardize some data*

select distinct industry from layoffs\_staging2

order by 1;

*-- in industry, there are multiple records with crypto related,*

select \* from layoffs\_staging2

where industry like 'Crypto%'

order by 1;

*-- to standardize, update to all 'Crypto'*

update layoffs\_staging2

set industry ='Crypto'

where industry like 'Crypto%';

select distinct industry from layoffs\_staging2

order by 1;

*-- now correct country column*

select distinct country from layoffs\_staging2

order by 1;

*-- this removes trailing from country name*

select distinct country, trim(trailing '.' from country)

from layoffs\_staging2

order by 1;

*-- update all country removing all trailing .*

update layoffs\_staging2

set country = trim(trailing '.' from country) ;

select distinct country from layoffs\_staging2 order by 1;

*-- now change the data type of date column*

select `date`,

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

from layoffs\_staging2;

*-- update the values of dates to date type with str\_to\_date function as below.*

update layoffs\_staging2

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

*-- update the column 'date' to date type*

alter table layoffs\_staging2

modify column `date` date ;

***-- 3. find null values and delete the records***

select \* from layoffs\_staging2

where total\_laid\_off is null

and percentage\_laid\_off is null;

delete from layoffs\_staging2

where total\_laid\_off is null

and percentage\_laid\_off is null;

select \* from layoffs\_staging2;

***-- 4. remove unnecessary columns***

***-- this will remove row\_num column which is not needed for us now.***

alter table layoffs\_staging2

drop column row\_num;

select \* from layoffs\_staging2;