# SQL PROJECT

# CONTENTS

This data contains large laid_off dataset of different companies at different location .

## DATA CLEANING

1. check for duplicates and remove any.
2. standardize data and fix errors.
3. Look at null values .
4. remove any columns and rows that are not necessary.

## Exploratory data analysis (EDA)

1.Going to explore the data and find trends or patterns or anything interesting like outliers
2.some idea of what you're looking for with this info we are just going to look around and see what we find!

```
SELECT COUNT(`DATE`)
FROM layoffs_staging2
group by 'date';
```

```
COUNT(`DATE`)
'1994'
```

```
-- first thing we want to do is create a staging table. This is the one we will work in and clean the data. We want a table with the raw data in case something happens

CREATE TABLE world_layoffs.layoffs_staging
LIKE world_layoffs.layoffs;

INSERT layoffs_staging
SELECT * FROM world_layoffs.layoffs;
```

| company | location | industry | total_laid_off | percentage_laid_off | date | stage | country | funds_raised_millions |
|---|---|---|---|---|---|---|---|---|
| Atlassian | Sydney | Other | 500 | 0.05 | 3/6/2023 | Post-IPO | Australia | 210 |
| SiriusXM | New York City | Media | 475 | 0.08 | 3/6/2023 | Post-IPO | United States | 525 |
| Alerzo | Ibadan | Retail | 400 | NULL | 3/6/2023 | Series B | Nigeria | 16 |
| UpGrad | Mumbai | Education | 120 | NULL | 3/6/2023 | Unknown | India | 631 |
| Loft | Sao Paulo | Real Estate | 340 | 0.15 | 3/3/2023 | Unknown | Brazil | 788 |
| Embark Trucks | SF Bay Area | Transportation | 230 | 0.7 | 3/3/2023 | Post-IPO | United States | 317 |
| Lendi | Sydney | Real Estate | 100 | NULL | 3/3/2023 | Unknown | Australia | 59 |
| UserTesting | SF Bay Area | Marketing | 63 | NULL | 3/3/2023 | Acquired | United States | 152 |
| Airbnb | SF Bay Area | | 30 | NULL | 3/3/2023 | Post-IPO | United States | 6400 |
| Accolade | Seattle | Healthcare | NULL | NULL | 3/3/2023 | Post-IPO | United States | 458 |
| Indigo | Boston | Other | NULL | NULL | 3/3/2023 | Series F | United States. | 1200 |
| Zscaler | SF Bay Area | Security | 177 | 0.03 | 3/2/2023 | Post-IPO | United States | 148 |
| MasterClass | SF Bay Area | Education | 79 | NULL | 3/2/2023 | Series E | United States | 461 |
| Ambev Tech | Blumenau | Food | 50 | NULL | 3/2/2023 | Acquired | Brazil | NULL |

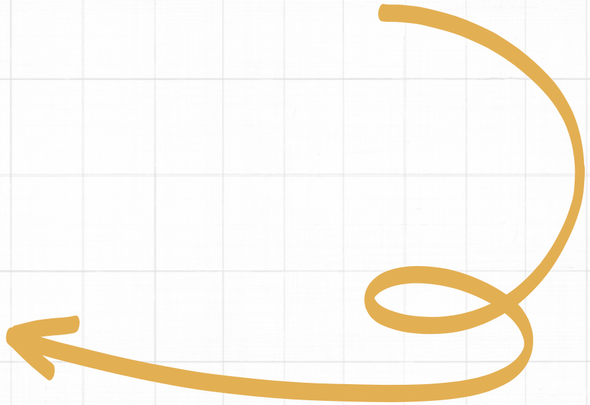
```
ALTER TABLE world_layoffs.layoffs_staging ADD row_num INT;


SELECT *
FROM world_layoffs.layoffs_staging
;

CREATE TABLE `world_layoffs`.`layoffs_staging2` (
`company` text,
`location`text,
`industry`text,
`total_laid_off` INT,
`percentage_laid_off` text,
`date` text,
`stage`text,
`country` text,
`funds_raised_millions` int,
row_num INT
);

INSERT INTO `world_layoffs`.`layoffs_staging2`
(`company`,
`location`,
`industry`,
`total_laid_off`,
`percentage_laid_off`,
`date`,
`stage`,
`country`,
`funds_raised_millions`,
`row_num`)
SELECT `company`,
`location`,
`industry`,
`total_laid_off`,
`percentage_laid_off`,
`date`,
`stage`,
`country`,
`funds_raised_millions`,
ROW_NUMBER() OVER (
PARTITION BY company, location, industry, total_laid_off,percentage_laid_off,`date`, stage, country, funds_raised_millions
) AS row_num
FROM
world_layoffs.layoffs_staging;
```

| company | location | industry | total_laid_off | percentage_laid_off | date | stage | country | funds_raised_millions | row_num |
|---|---|---|---|---|---|---|---|---|---|
| E Inc. | Toronto | Transportation | NULL | NULL | 12/16/2022 | Post-IPO | Canada | NULL | 1 |
| E Inc. | Toronto | Transportation | NULL | NULL | 12/16/2022 | Post-IPO | Canada | NULL | 2 |
| Included Health | SF Bay Area | Healthcare | NULL | 0.06 | 7/25/2022 | Series E | United States | 272 | 1 |
| Included Health | SF Bay Area | Healthcare | NULL | 0.06 | 7/25/2022 | Series E | United States | 272 | 2 |
| &Open | Dublin | Marketing | 9 | 0.09 | 11/17/2022 | Series A | Ireland | 35 | 1 |
| &Open | Dublin | Marketing | 9 | 0.09 | 11/17/2022 | Series A | Ireland | 35 | 2 |
| #Paid | Toronto | Marketing | 19 | 0.17 | 1/27/2023 | Series B | Canada | 21 | 1 |
| #Paid | Toronto | Marketing | 19 | 0.17 | 1/27/2023 | Series B | Canada | 21 | 2 |
| 100 Thieves | Los Angeles | Consumer | 12 | NULL | 7/13/2022 | Series C | United States | 120 | 1 |
| 100 Thieves | Los Angeles | Consumer | 12 | NULL | 7/13/2022 | Series C | United States | 120 | 2 |
| 100 Thieves | Los Angeles | Retail | NULL | NULL | 1/10/2023 | Series C | United States | 120 | 1 |
| 100 Thieves | Los Angeles | Retail | NULL | NULL | 1/10/2023 | Series C | United States | 120 | 2 |
| 10X Genomics | SF Bay Area | Healthcare | 100 | 0.08 | 8/4/2022 | Post-IPO | United States | 242 | 1 |



.Standardize Data

```
SELECT *
FROM world_layoffs.layoffs_staging2;

-- if we look at industry it looks like we have some null and empty rows, let's take a look at these
SELECT DISTINCT industry
FROM world_layoffs.layoffs_staging2
ORDER BY industry;

SELECT *
FROM world_layoffs.layoffs_staging2
WHERE industry IS NULL
OR industry = ''
ORDER BY industry;
```

```
select company ,trim(company)
from  layoffs_staging2;
update layoffs_staging2
set company = trim(company);

---   we convert croptocurrency or crypto to crypto

select *
from layoffs_staging2
where industry like 'Crypto%';

update layoffs_staging2
set industry = 'crypto'
where industry like 'Crypto%';
-- we standardize some error
select * from layoffs_staging2;
select distinct country, trim(country)
from layoffs_staging2
where country like 'united states%'
order by 1;
UPDATE layoffs_staging2
SET country = TRIM(TRAILING '.' FROM country)
where country like 'united states%'
;
```

--- find the null values and blank values

```
select * from layoffs_staging2
where total_laid_off is null
and percentage_laid_off is 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 industry from layoffs_staging2
where industry is null ;

select * from layoffs_staging2
where total_laid_off is null
and percentage_laid_off is null;
```

-- remove columns and row (row_num) which is not required

```
delete
from layoffs_staging2
where total_laid_off is null
and percentage_laid_off is null;

alter table layoffs_staging2
drop row_num ;
```

```sql
-- Looking at Percentage to see how big these layoffs were
SELECT MAX(percentage_laid_off),  MIN(percentage_laid_off)
FROM layoffs_staging2
WHERE  percentage_laid_off IS NOT NULL;
-- Which companies had 1 which is basically 100 percent of they company laid off
SELECT *
FROM layoffs_staging2
WHERE  percentage_laid_off = 1;
-- these are mostly startups it looks like who all went out of business during this time
```

| MAX(PERCENTAGE_LAID_OFF) | MIN(PERCENTAGE_LAID_OFF) |
|---|---|
| 1 | 0 |

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| company | location | industry | total_laid_off | percentage_laid_off | date | stage | country | funds_raised_millions |
|---|---|---|---|---|---|---|---|---|
| Ahead | SF Bay Area | Healthcare | 44 | 1 | 2022-04-14 | Unknown | United States | 9 |
| Airlift | Lahore | Logistics | NULL | 1 | 2022-07-12 | Series B | Pakistan | 109 |
| Airy Rooms | Jakarta | Travel | NULL | 1 | 2020-05-07 | Unknown | Indonesia | NULL |
| Amplero | Seattle | Marketing | 17 | 1 | 2020-03-29 | Series B | United States | 25 |
| Arch Oncology | Brisbane | Healthcare | NULL | 1 | 2023-01-13 | Series C | United States | 155 |
| Assure | Salt Lake City | Finance | NULL | 1 | 2022-11-23 | Seed | United States | 2 |
| Atsu | Seattle | Infrastructure | 6 | 1 | 2020-04-10 | Unknown | United States | 1 |
| Aura Financial | SF Bay Area | Finance | NULL | 1 | 2021-01-11 | Unknown | United States | 584 |

```sql
---Using group by
-- Companies with the biggest single Layoff

SELECT company, total_laid_off
FROM layoffs_staging2
ORDER BY 2 DESC
LIMIT 5;
-- now that's just on a single day
```

Result Grid | Filter Rows: | Export:

| company | total_laid_off |
|---|---|
| Google | 12000 |
| Meta | 11000 |
| Amazon | 10000 |
| Microsoft | 10000 |
| Ericsson | 8500 |

```sql
-- Companies with the most Total Layoffs
SELECT company, SUM(total_laid_off)
FROM layoffs_staging2
GROUP BY company
ORDER BY 2 DESC
LIMIT 10;
```

Result Grid | Filter Rows: | Export:

| company | SUM(total_laid_off) |
|---|---|
| Amazon | 18150 |
| Google | 12000 |
| Meta | 11000 |
| Salesforce | 10090 |
| Microsoft | 10000 |
| Philips | 10000 |
| Ericsson | 8500 |

```sql
-- by location
SELECT location, SUM(total_laid_off)
FROM layoffs_staging2
GROUP BY location
ORDER BY 2 DESC
LIMIT 10;
```

Result Grid | Filter Rows: | Export:

| location | SUM(total_laid_off) |
|---|---|
| SF Bay Area | 125631 |
| Seattle | 34743 |
| New York City | 29364 |
| Bengaluru | 21787 |
| Amsterdam | 17140 |
| Stockholm | 11217 |
| Boston | 10785 |

```sql
-- this it total in the past 3 years or in the dataset

SELECT country, SUM(total_laid_off)
FROM layoffs_staging2
GROUP BY country
ORDER BY 2 DESC;

SELECT YEAR(date), SUM(total_laid_off)
FROM layoffs_staging2
GROUP BY YEAR(date)
ORDER BY 1 deSC
limit 3;
```

Result Grid | Filter Rows:

| YEAR(date) | SUM(total_laid_off) |
|---|---|
| 2023 | 125677 |
| 2022 | 160661 |
| 2021 | 15823 |

```sql
SELECT industry, SUM(total_laid_off)
FROM layoffs_staging2
GROUP BY industry
ORDER BY 2 DESC;
```

Result Grid | Filter Rows: | Export:

| industry | SUM(total_laid_off) |
|---|---|
| Consumer | 45182 |
| Retail | 43613 |
| Other | 36289 |
| Transportation | 33748 |
| Finance | 28344 |
| Healthcare | 25953 |
| Food | 22855 |
| Real Estate | 17565 |
| Travel | 17159 |
| Hardware | 13828 |
| Education | 13338 |

```sql
-- by location
SELECT location, SUM(total_laid_off)
FROM layoffs_staging2
GROUP BY location
ORDER BY 2 DESC
LIMIT 10;
```

Result Grid | Filter Rows: | Export:

| location | SUM(total_laid_off) |
|---|---|
| SF Bay Area | 125631 |
| Seattle | 34743 |
| New York City | 29364 |
| Bengaluru | 21787 |
| Amsterdam | 17140 |
| Stockholm | 11217 |
| Boston | 10785 |
| Sao Paulo | 9081 |
| Austin | 8980 |
| Chicago | 6419 |