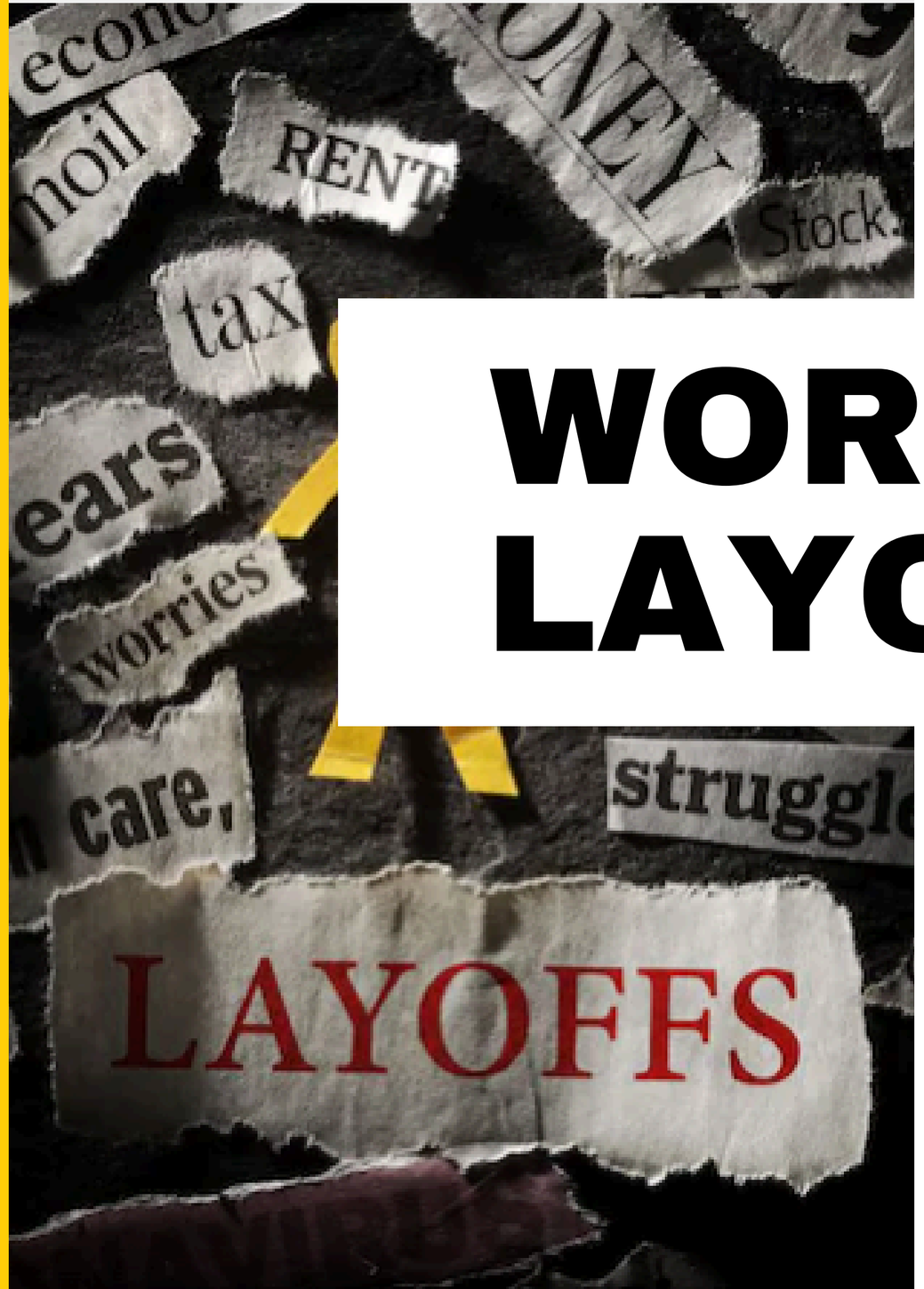

WORLD'S LAYOFFS

ABHISHEK VERMA

Github : https://github.com/abhishekvermacu20/World-s-Layoffs

Portfolio website

Table of Content

ABOUT PROJECT

DATASET

ETL - DATA TRANSFORMATION

QUERIES

CONTACT

ETL - EXTRACT TRANSFORMATION LOAD

ABOUT PROJECT

In this project, we will undertake a comprehensive data cleaning initiative using MySQL. Data cleaning, also referred to as data scrubbing, involves the identification and correction or removal of inaccurate, incomplete, or duplicate entries within a dataset. This process is essential to maintaining data integrity and is a critical component of data science. Ensuring high data quality is paramount for organizations that rely on data-driven decisions, making this project fundamental to our objectives.

Data Cleaning

1

Remove Duplicates.

2

Standardize the Data

3

Null Values or Blank Values

4

Remove Any Columns

Our Dataset

ETL - EXTRACT TRANSFORMATION LOAD

Layoffs Dataset 2024

This dataset was scraped from Layoffs.fyi with the hope to enable Kaggle community to look into analyzing recent mass layoffs and discover useful insights and patterns.

Link : https://www.kaggle.com/datasets/theakhilb/layoffs-data-2022/datas.

Queries

```
create table layoffs_staging
like layoffs;

select *
from layoffs_staging;

insert layoffs_staging
select *
from layoffs;
```

1. **Make raw data as a backup and create same new table where we do changes.**

As part of the World's Layoff project, we will begin by creating a backup of the raw data to ensure data integrity and preserve the original dataset. A new, identical table will be created where all subsequent data cleaning and transformation activities will be conducted. This approach allows us to maintain the original data in its untouched state while systematically improving data quality in the newly created table

Queries

```sql
select *,
row_number() over(partition by
company, industry, total_laid_off, percentage_laid_off,
`date`, stage, country, funds_raised_millions) as row_num
from layoffs_staging;

with duplicate_ctc 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_ctc
where row_num > 1;
```

2. Remove Duplicates using row_number

To eliminate duplicate records from our dataset, we will utilize the ROW_NUMBER() function within MySQL. This function allows us to assign a unique identifier to each row within groups of potential duplicates, based on specified columns. By systematically retaining only the first occurrence of each duplicate, we can ensure that our dataset remains clean and free of redundant entries, which is critical for accurate analysis.

Queries

```sql
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;

select *
from layoffs_staging2;
```

3. Create new table where row_num column is present

-- row_num is not updatable because it is not present in raw data.
-- click layoffs_staging table -> copy to clickboard -> create statement -> crtl + v (paste it)
-- change table's name layoffs_staging to layoffs_staging2
-- add row_num in table column with int datatype

Queries

4. Insert rows in layoffs_staging2

```sql
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;
```

Queries

5. Remove duplicates

```sql
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;
```

Queries

```
-- Trim company column
select company, trim(company)
from layoffs_staging2;

update layoffs_staging2
set company = trim(company);
```

6. Standardizing data

Standardizing data is a crucial step in ensuring consistency and accuracy across a dataset. This process involves converting data into a common format, which allows for seamless comparison, analysis, and integration. By standardizing variables such as dates, names, and categorical values, we eliminate discrepancies that can arise from variations in data entry or formatting.

Effective data standardization enhances the quality of the dataset, making it more reliable for downstream processes, including reporting and predictive modeling. This step is essential for maintaining data integrity and ensuring that the insights drawn from the data are based on uniform and comparable information.

Queries

```sql
select distinct industry
from layoffs_staging2
order by 1;

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

update layoffs_staging2
set industry = 'Crypto'
where industry like 'Crypto%';

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%';
```

Queries

```sql
select *
from layoffs_staging2
where location like 'DÃ¼sseldorf'
order by location;

update layoffs_staging2
set location = 'Dusseldorf'
where location like 'DÃ¼sseldorf%';

select *
from layoffs_staging2
where location like 'FlorianÃ³polis'
order by location;

update layoffs_staging2
set location = 'Florianopolis'
where location like 'FlorianÃ³polis%';

select *
from layoffs_staging2
where location like 'MalmÃ¶'
order by location;
```

ETL - EXTRACT TRANSFORMATION LOAD

Queries

```sql
update layoffs_staging2
set location = 'Malmo'
where location like 'MalmÃ¶%';

select distinct location
from layoffs_staging2;

select `date`, str_to_date(`date`,'%m/%d/%Y')
from layoffs_staging2;

update layoffs_staging2
set `date` = str_to_date(`date`, '%m/%d/%Y');

alter table layoffs_staging2
modify column `date` date;

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

update layoffs_staging2
set industry = NULL
where industry = '';
```

Queries

```sql

select *
from layoffs_staging2
where industry is null
or industry = '';

select *
from layoffs_staging2
where company = 'Airbnb';

select t1.industry, t2.industry
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;

```

Queries

```sql
select distinct industry
from layoffs_staging;

select *
from layoffs_staging2
where company = 'Bally\'s Interactive';

-- Droping Data

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;

Alter table layoffs_staging2
drop column row_num;

select *
from layoffs_staging2;
```

Output

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

company	location	industry	total_laid_off	percentage_laid_off	date	stage	country
Included Health	SF Bay Area	Healthcare	NULL	0.06	2022-07-25	Series E	United States
&Open	Dublin	Marketing	9	0.09	2022-11-17	Series A	Ireland
#Paid	Toronto	Marketing	19	0.17	2023-01-27	Series B	Canada
100 Thieves	Los Angeles	Consumer	12	NULL	2022-07-13	Series C	United States
10X Genomics	SF Bay Area	Healthcare	100	0.08	2022-08-04	Post-IPO	United States
1stdibs	New York City	Retail	70	0.17	2020-04-02	Series D	United States
2TM	Sao Paulo	Crypto	90	0.12	2022-06-01	Unknown	Brazil
2TM	Sao Paulo	Crypto	100	0.15	2022-09-01	Unknown	Brazil
2U	Washington D.C.	Education	NULL	0.2	2022-07-28	Post-IPO	United States
54gene	Washington D.C.	Healthcare	95	0.3	2022-08-29	Series B	United States
5B Solar	Sydney	Energy	NULL	0.25	2022-06-03	Series A	Australia
6sense	SF Bay Area	Sales	150	0.1	2022-10-12	Series E	United States
80 Acres Farms	Cincinnati	Food	NULL	0.1	2023-01-18	Unknown	United States
8x8	SF Bay Area	Support	155	0.07	2023-01-18	Post-IPO	United States
8x8	SF Bay Area	Support	200	0.09	2022-10-04	Post-IPO	United States

ETL - EXTRACT TRANSFORMATION LOAD

Let's Connect With Us!

abhishekvermacu20@gmail.com

www.linkedin.com