Create table layoff

```
CREATE TABLE layoff (
    company TEXT,
    location TEXT,
    industry TEXT,
    total_laid_off INT,
    percentage_laid_off DECIMAL,
    date DATE,
    stage TEXT,
    country TEXT,
    funds_raised DECIMAL
);
```

Insert data using import/export tool of pgAdmin

```
SELECT * FROM layoff
        LIMIT 10;
```

Creating a working version of table

```
CREATE TABLE layoff_cl AS TABLE layoff;
```

```
SELECT * FROM layoff_cl
        LIMIT 10;
```

Finding duplicates by:

```
SELECT * FROM (
        SELECT *, ROW_NUMBER() OVER(
                PARTITION BY company, location, industry, total_laid_off,
percentage_laid_off,
                date, stage, country, funds_raised
                                ) AS dup_id
                FROM layoff_cl) AS dup_sub
                        WHERE dup_id > 1;
```

or by:

```
WITH dup_cte as (
        SELECT *, ROW_NUMBER() OVER(
PARTITION BY company, location, industry, total_laid_off,
percentage_laid_off, date, stage, country, funds_raised
               ) AS dup_id
        FROM layoff_cl)
SELECT * FROM dup_cte
WHERE dup_id > 1;
```

Two duplicates were found and need to be checked:

```
SELECT * FROM layoff_cl
WHERE company IN ('Beyond Meat', 'Cazoo')
ORDER BY company
```

There is one duplicate record for each of them.

Create a new table with an additional column for row_num

```
CREATE TABLE layoff_cl2(
    company TEXT,
    location TEXT,
    industry TEXT,
    total_laid_off INT,
    percentage_laid_off TEXT,
    date TEXT,
    stage TEXT,
    country TEXT,
    funds_raised DECIMAL,
        row_num INT);
```

Inser data from previous table plus row_num

```
INSERT INTO layoff_cl2
        (company, location, industry, total_laid_off,
        percentage_laid_off, date, stage,
        country, funds_raised, row_num)
        SELECT
                company, location, industry, total_laid_off,
                percentage_laid_off, date, stage,
                country, funds_raised,
                ROW_NUMBER() OVER(
                        PARTITION BY company, location, total_laid_off,
                        percentage_laid_off, date, stage, country,
                        funds_raised) AS row_num
        FROM layoff_cl;
```

Remove duplicates

```
DELETE FROM layoff_cl2
WHERE row_num > 1
```

Check again for duplicates

```
SELECT * FROM(
        SELECT *, ROW_NUMBER() OVER(
                PARTITION BY company, location, industry, total_laid_off,
                percentage_laid_off, date, stage, country, funds_raised
                               ) AS dup_id
                FROM layoff_cl2) AS dup_sub
                        WHERE dup_id > 1;
```

Null values in industry column

```
SELECT DISTINCT(industry)
      FROM layoff_cl2;
```

Returned 31 rows including null. Check records with null industry:

```
SELECT * FROM layoff_cl2
      WHERE industry IS NULL;
```

There is only one company, "Appsmith," in the dataset with null industry. Since it is a software company, I replaced its industry value with "Other," consistent with how most software companies are categorized in the dataset

```
UPDATE layoff_cl2
      SET industry = 'Other'
      WHERE company = 'Appsmith';
```

double check

```
SELECT * FROM layoff_cl2
      WHERE industry = '';
```

Null values in location column

```
SELECT DISTINCT (location)
      FROM layoff_cl2;
```

```
SELECT * FROM layoff_cl2
      WHERE location IS NULL;

SELECT * FROM layoff_cl2
      WHERE location LIKE '%U.S%';
```

I noticed several issues with the location data while extracting distinct values. For instance, there were multiple spellings for cities such as Düsseldorf (listed as both "Düsseldorf" and "Dusseldort") and Malmö (listed as both "Malmö" and "Malmo").Additionally, I encountered a null value and a "Non.U.S." value. To address these issues, I replaced "Dusseldort" with "Düsseldorf" and "Malmo" with "Malmö." For the company with a null location, "Product Hunt," the correct location is San Francisco, which is indicated as "SF Bay Area" in the dataset.The "Non.U.S." value corresponded to two companies, BitMex and WeDoctor. BitMex is located in the Republic of Seychelles, and WeDoctor is located in Beijing.

I have corrected these locations accordingly:

```
UPDATE layoff_cl2
SET location = 'Seychelles'
      WHERE company = 'BitMEX';

UPDATE layoff_cl2
SET location = 'Beijing'
      WHERE company = 'WeDoctor';

UPDATE layoff_cl2
SET location = 'SF Bay Area'
      WHERE company = 'Product Hunt';
```

Checking date column

change the data format to "date"

```
ALTER TABLE layoff_cl2
ALTER COLUMN date TYPE DATE USING date::date;
```

I chose to create separate year and month columns for convenience

```
ALTER TABLE layoff_cl2
ADD COLUMN year INT,
ADD COLUMN month TEXT;
```

Populate the year and month columns based on date values

```
UPDATE layoff_cl2
SET year = EXTRACT(YEAR FROM date);

UPDATE layoff_cl2
SET month = TO_CHAR(date, 'Month');
```

Checking stage column

```
SELECT DISTINCT(stage) FROM layoff_cl2;
```

```
SELECT * FROM layoff_cl2
      WHERE stage IS NULL;
```

There are seven records with null values in the stage column. I replaced them with the value "Unknown"

```
UPDATE layoff_cl2
SET stage = 'Unknown'
WHERE stage IS NULL;
```

Columns such as total_laid_off, percentage_laid_off and funds_raised are the most useful for EDA here. Let's identify records with null values in these three columns:

```
SELECT * FROM layoff_cl2
      WHERE total_laid_off IS NULL
      AND percentage_laid_off IS NULL
      AND funds_raised IS NULL;
```

There were 91 records, all of which were removed. This accounts for 2.5% of the dataset

```
DELETE FROM layoff_cl2
WHERE total_laid_off IS NULL
      AND percentage_laid_off IS NULL
      AND funds_raised IS NULL;
```

Removing and renaming columns

```
ALTER TABLE layoff_cl2
DROP COLUMN row_num;
```

```
ALTER TABLE layoff_cl2
RENAME COLUMN total_laid_off TO layoff_num;
```

```
ALTER TABLE layoff_cl2
RENAME COLUMN percentage_laid_off TO layoff_perc;
```

```
ALTER TABLE layoff_cl2
ALTER COLUMN layoff_perc TYPE DECIMAL USING layoff_perc::DECIMAL;
```

```
SELECT COUNT(*) FROM layoff_cl2;
```

3849 records

```
SELECT * FROM layoff_cl2;
```