



# WORLD LAYOFFS SQL DATA CLEANING PROJECT

ALAN GAUGLER  
MAY 2024

# INTRODUCTION

This is a quick introductory project to demonstrate an understanding of SQL coding and data wrangling.

In this project, I will demonstrate the important process of data cleaning using SQL, working on a dataset related to global layoffs.

Data cleaning is a fundamental step in data analysis and involves preparing raw data to ensure it is accurate, consistent, clean and usable. -- This project will involve removing duplicates, standardizing data, handling null or blank values, and removing unnecessary columns.

This project is a data loading and cleaning exercise in SQL.

The following steps will be implemented:

- 1. Remove Duplicate Rows:** Identify and eliminate duplicate rows to ensure each record is unique.
- 2. Standardize the Data:** Ensure consistency in data formats and values across the dataset.
- 3. Handle Null or Blank Values:** Address missing values appropriately, whether through imputation or exclusion.
- 4. Remove Unnecessary Columns:** Eliminate columns that do not contribute to the analysis to streamline the dataset.

The cleaning stage will be followed by some exploratory data analysis to gain meaningful insights from the cleaned dataset.

# INTRODUCTION

The following SQL script demonstrates these steps in detail, including the cautious approach of not disabling SAFE UPDATES to prevent accidental data loss.

Where there are updates to the table's values, SET SQL\_SAFE\_UPDATES = 0; -- Disable Safe updates and SELECT \* FROM world\_layoffs.layoffs; commands are entered before and after, to ensure that all alterations are carefully managed.

# 1. Remove Duplicate Rows

## Code

```
26    -- Drop the layoffs_staging table if it exists
27 •  DROP TABLE IF EXISTS layoffs_staging;
28    -- Create a staging table (a new temporary table)
29 •  CREATE TABLE layoffs_staging
30    LIKE layoffs;
31
32
33    -- Observe that layoffs_staging is empty
34 •  SELECT *
35    FROM layoffs_staging;
36
37
38    -- Fill layoffs_staging (a new temporary table) with all the data from layoffs
39 •  INSERT layoffs_staging
40    SELECT *
41    FROM layoffs;
42
43    -- How many rows?
44 •  SELECT COUNT(*)
45    FROM layoffs_staging;
```

## Output

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	COUNT(*)			
▶	2361			

-- There are 2361 rows or items.

## Code

```
49      ----- 1. Remove Duplicate Rows -----
50
51      -- First identify which rows are duplicates.alter
52      -- Company should be the unique identifier
53
54 •   SELECT company, count(*) AS NumEntries
55     FROM layoffs_staging
56     GROUP BY company
57     HAVING count(*) > 1
58     ORDER BY count(*) DESC;
```

## Output

company	NumEntries
Loft	6
Swiggy	5
OYO	5
Uber	5
Twitter	4
Convoy	4
ClearCo	4
Ola	4
Crypto.com	4
Blend	4
Bounce	4
Compass	4
Salesforce	4
ByteDance	4
Vedantu	4
Argo AI	4
Gopuff	4
Truepill	4
Netflix	4

```
60      -- Several companies have more than one entry. Loft has the most with 6 entries.
61      -- However we want to find true duplicates where all columns have the same value.
62      -- A smarter way of doing this is as follows:
```

## Code

```
63 • WITH duplicates_table AS
64  (
65      SELECT *,
66          COUNT(*) OVER(PARTITION BY company, location, industry, total_laid_off, percentage_laid_off, `date`, stage, country, funds_raised_millions) AS NumRows,
67          ROW_NUMBER() OVER(PARTITION BY company, location, industry, total_laid_off, percentage_laid_off, `date`, stage, country, funds_raised_millions ORDER BY `date` DESC) AS RowNum
68  FROM layoffs_staging
69 )
70  SELECT *
71  FROM duplicates_table
72  WHERE NumRows > 1
73  ORDER BY `date`;
```

## Output

	company	location	industry	total_laid_off	percentage_laid_off	date	stage	country	funds_raised_millions	NumRows	RowNum
▶	Wildlife Studios	Sao Paulo	Consumer	300	0.2	11/28/2022	Unknown	Brazil	260	2	1
	Wildlife Studios	Sao Paulo	Consumer	300	0.2	11/28/2022	Unknown	Brazil	260	2	2
	Yahoo	SF Bay Area	Consumer	1600	0.2	2/9/2023	Acquired	United States	6	2	1
	Yahoo	SF Bay Area	Consumer	1600	0.2	2/9/2023	Acquired	United States	6	2	2
	Hibob	Tel Aviv	HR	70	0.3	3/30/2020	Series A	Israel	45	2	1
	Hibob	Tel Aviv	HR	70	0.3	3/30/2020	Series A	Israel	45	2	2
	Cazoo	London	Transportation	750	0.15	6/7/2022	Post-IPO	United Kingdom	2000	2	1
	Cazoo	London	Transportation	750	0.15	6/7/2022	Post-IPO	United Kingdom	2000	2	2
	Casper	New York City	Retail	NULL	NULL	9/14/2021	Post-IPO	United States	339	2	1
	Casper	New York City	Retail	NULL	NULL	9/14/2021	Post-IPO	United States	339	2	2

```
75    -- Observing these rows, the layoffs by the same company happen in different locations or different dates.
76    -- They are likely all valid layoffs and these rows should not be deleted.
77
78    -- However there are 5 pairs where all rows are exactly the same. These with RowNum >= 2 are duplicates and need to be removed.
79
```

```

80 •  DROP TABLE IF EXISTS layoffs_staging2;
81
82    -- Create a new table
83 •  CREATE TABLE `layoffs_staging2` (
84        `company` text,
85        `location` text,
86        `industry` text,
87        `total_laid_off` int DEFAULT NULL,
88        `percentage_laid_off` text,
89        `date` text,
90        `stage` text,
91        `country` text,
92        `funds_raised_millions` int DEFAULT NULL,
93        `row_num` INT
94    ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
95
96    -- And fill it
97 •  INSERT INTO layoffs_staging2
98    SELECT *,
99        ROW_NUMBER() OVER(PARTITION BY company, location, industry, total_laid_off, percentage_laid_off, `date`, stage, country, funds_raised_millions) AS row_num
100   FROM layoffs_staging;
101
102    -- Observe the duplicate rows
103 •  SELECT *
104    FROM layoffs_staging2
105   WHERE row_num > 1;

```

## Code

### Output

	company	location	industry	total_laid_off	percentage_laid_off	date	stage	country	funds_raised_millions	row_num
▶	Casper	New York City	Retail	NULL	NULL	9/14/2021	Post-IPO	United States	339	2
	Cazoo	London	Transportation	750	0.15	6/7/2022	Post-IPO	United Kingdom	2000	2
	Hibob	Tel Aviv	HR	70	0.3	3/30/2020	Series A	Israel	45	2
	Wildlife Studios	Sao Paulo	Consumer	300	0.2	11/28/2022	Unknown	Brazil	260	2
	Yahoo	SF Bay Area	Consumer	1600	0.2	2/9/2023	Acquired	United States	6	2

## Code

```
106  
107      -- Let's delete where RowNum is > 1 or all duplicates  
108  
109 •  SET SQL_SAFE_UPDATES = 0; -- Disable Safe updates  
110  
111 •  DELETE FROM layoffs_staging2 WHERE row_num > 1;  
112  
113 •  SET SQL_SAFE_UPDATES = 1; -- Re-enable safe updates  
114  
115 •  SELECT COUNT(*)  
116     FROM layoffs_staging2;
```

## Code

```
118      -- Observe the table  
119 •  SELECT *  
120     FROM layoffs_staging2;
```

## Output

## Output

	Count
	2356

	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
	Included Health	SF Bay Area	Healthcare	NULL	0.06	7/25/2022	Series E	United States	272	1
	&Open	Dublin	Marketing	9	0.09	11/17/2022	Series A	Ireland	35	1
	#Paid	Toronto	Marketing	19	0.17	1/27/2023	Series B	Canada	21	1
	100 Thieves	Los Angeles	Consumer	12	NULL	7/13/2022	Series C	United States	120	1
	100 Thieves	Los Angeles	Retail	NULL	NULL	1/10/2023	Series C	United States	120	1
	10X Genomics	SF Bay Area	Healthcare	100	0.08	8/4/2022	Post-IPO	United States	242	1
	1stdibs	New York City	Retail	70	0.17	4/2/2020	Series D	United States	253	1
	2TM	Sao Paulo	Crypto	90	0.12	6/1/2022	Unknown	Brazil	250	1
	2TM	Sao Paulo	Crypto	100	0.15	9/1/2022	Unknown	Brazil	250	1
	2U	Washington D.C.	Education	NULL	0.2	7/28/2022	Post-IPO	United States	426	1
	54gene	Washington D.C.	Healthcare	95	0.3	8/29/2022	Series B	United States	44	1
	5B Solar	Sydney	Energy	NULL	0.25	6/3/2022	Series A	Australia	12	1
	6sense	SF Bay Area	Sales	150	0.1	10/12/2022	Series E	United States	426	1
	80 Acres Farms	Cincinnati	Food	NULL	0.1	1/18/2023	Unknown	United States	275	1
	8x8	SF Bay Area	Support	155	0.07	1/18/2023	Post-IPO	United States	253	1
	8x8	SF Bay Area	Support	200	0.09	10/4/2022	Post-IPO	United States	253	1
	98point6	Seattle	Healthcare	NULL	0.1	7/21/2022	Series E	United States	247	1
	99	Sao Paulo	Transportation	75	0.02	9/20/2022	Acquired	Brazil	244	1
	Abra	SF Bay Area	Crypto	12	0.05	6/30/2022	Series C	United States	106	1
	Absci	Vancouver	Healthcare	40	NULL	8/9/2022	Post-IPO	United States	237	1
	Acast	Stockholm	Media	70	0.15	9/15/2022	Post-IPO	Sweden	126	1
	Accolade	Seattle	Healthcare	NULL	NULL	3/3/2023	Post-IPO	United States	458	1
	Acko	Mumbai	Finance	45	0.09	4/1/2020	Unknown	India	143	1
	Acorns	Portland	Finance	50	NULL	5/26/2020	Unknown	United States	207	1
	Actifio	Boston	Data	54	NULL	12/16/2020	Acquired	United States	352	1
	ActiveCampaign	Chicago	Marketing	NULL	0.15	10/3/2022	Series C	United States	360	1
	Ada	Toronto	Support	NULL	NULL	2/1/2023	Series C	Canada	190	1
	Ada	Toronto	Support	78	0.16	9/20/2022	Series C	Canada	190	1

```
122      -- It can now be observed that there are now no duplicate rows as they have been removed.
```

## 2. Standardization (Consistency) of the Data

Code

```
125      ----- 2. Standardization (Consistency) of the Data -----
126
127      -- First it can be observed that there are some unnecessary spaces in the company names as can be seen below:
128 •  SELECT company, TRIM(company)
129      FROM layoffs_staging2;
```

Output

	company	TRIM(company)
▶	E Inc.	E Inc.
	Included Health	Included Health
	&Open	&Open
	#Paid	#Paid
	100 Thieves	100 Thieves
	100 Thieves	100 Thieves
	10X Genomics	10X Genomics
	1stdibs	1stdibs
	2TM	2TM
	2TM	2TM
	2U	2U
	54gene	54gene
	5B Solar	5B Solar
	6sense	6sense
	80 Acres Farms	80 Acres Farms
	8x8	8x8
	8x8	8x8
	98point6	98point6
	99	99
	Abra	Abra
	Absci	Absci
	Acast	Acast
	Accolade	Accolade
	Acko	Acko
	Acorns	Acorns
	Actifio	Actifio
	ActiveCampaign	ActiveCampaign
	Ada	Ada
	Ada	Ada

Code

```
131      -- Observing carefully above, E Inc and Included Health have a space at the start. These need to be removed.
132      -- To be consistent. I will do the same for all text based columns
133 •  SET SQL_SAFE_UPDATES = 0; -- Disable Safe updates
134
135 •  UPDATE layoffs_staging2
136      SET company = TRIM(company),
137          location = TRIM(location),
138          industry = TRIM(industry),
139          stage = TRIM(stage),
140          country = TRIM(country);
141
142      -- The company column is now trimmed as well as others.
```

## Code

```
144    -- Check for similar values in the text columns that may be typos.  
145  
146 •  SELECT DISTINCT company  
147    FROM layoffs_staging2  
148    ORDER BY company;  
149    -- Many names here, this will be a long check
```

company
&Open
#Paid
100 Thieves
10X Genomics
1stdibs
2TM
2U
54gene
5B Solar
6sense
80 Acres Farms
8x8
98point6
99
Abra
Absci
Acast
Accolade
Acko
Acorns
Actifio
ActiveCampa...
Ada
Ada Health
Ada Support
Adaptive Bio...
Adara
Addepar

## Output

## Code

```
151 •  SELECT DISTINCT location  
152    FROM layoffs_staging2  
153    ORDER BY location;
```

location
Coimbra
Columbus
Copenhagen
Cork
Curitiba
DÃ¼sseldorf
Dakar
Dallas
Davenport
Denver
Detroit
Dover
Dubai
Dublin
Durham
Dusseldorf
Edinburgh
Eindhoven
Fayetteville
Ferdericton
FlorianÃ³polis
Frankfurt
Gothenburg
Grand Rapids
Guadalajara
Gurugram

## Output

155 -- Dusseldorf is incorrectly entered due to the umlaut. Florianopolis is also incorrect

## Code

```
157  -- Change the bad characters
158 • SET SQL_SAFE_UPDATES = 0; -- Disable Safe updates
159
160 • UPDATE layoffs_staging2
161   SET location = 'Düsseldorf'
162   WHERE location = 'DÃ¶sseldorf'
163     OR location = 'Dusseldorf';
164 • UPDATE layoffs_staging2
165   SET location = 'Florianópolis'
166   WHERE location = 'FlorianÃ³polis';
167 • UPDATE layoffs_staging2
168   SET location = 'Malmö'
169   WHERE location = 'MalmÃ¶'
170     OR location = 'Malmo';
171
172 • SET SQL_SAFE_UPDATES = 1; -- Re-enable safe updates
173
174 -- Display the Industry column and examine for issues.
175 • SELECT DISTINCT industry
176   FROM layoffs_staging2
177   ORDER BY industry;
178
```

## Output

industry
NULL
Aerospace
Construction
Consumer
Crypto
Crypto Currency
CryptoCurrency
Data
Education
Energy
Fin-Tech
Finance
Fitness
Food
Hardware
Healthcare
HR
Infrastructure
Legal
Logistics
Manufacturing
Marketing

178

179 -- Observing above, there are three similar industries for Crypto currency.

## Code

```
181    -- Display all distinct names containing 'crypto'  
182 •  SELECT DISTINCT *  
183    FROM layoffs_staging2  
184    WHERE industry LIKE '%crypto%';
```

## Output

company	location	industry	total_laid_off	percentage_laid_off	date	stage	country	funds_raised_millions	row_nu
ZTM	Sao Paulo	Crypto	90	0.12	6/1/2022	Unknown	Brazil	250	1
ZTM	Sao Paulo	Crypto	100	0.15	9/1/2022	Unknown	Brazil	250	1
Abra	SF Bay Area	Crypto	12	0.05	6/30/2022	Series C	United States	106	1
Amber Group	Hong Kong	Crypto	NULL	0.1	9/9/2022	Series B	Hong Kong	328	1
Autograph	Los Angeles	Crypto	NULL	NULL	12/16/2022	Series B	United States	205	1
Bakkt	Atlanta	Crypto	NULL	0.15	12/8/2022	Post-IPO	United States	932	1
Banxa	Melbourne	Crypto	70	0.3	6/27/2022	Post-IPO	Australia	13	1
Bitfarms	Quebec	Crypto	NULL	NULL	4/6/2020	Post-IPO	Canada	25	1
Bitfront	SF Bay Area	Crypto	NULL	1	11/29/2022	Unknown	United States	NULL	1
Gemini	New York City	CryptoCurrency	68	0.07	7/18/2022	Unknown	United States	423	1
Genesis	New York City	Crypto	52	0.2	8/17/2022	Series A	United States	NULL	1
Genesis	New York City	Crypto	60	0.3	1/5/2023	Series A	United States	NULL	1
GSR	Hong Kong	Crypto Currency	NULL	NULL	10/11/2022	Unknown	Hong Kong	NULL	1

## Code

```
186    -- The name 'Crypto' is by far the most common. They will all be set to 'Crypto'  
187 •  SET SQL_SAFE_UPDATES = 0; -- Disable Safe updates  
188  
189 •  UPDATE layoffs_staging2  
190    SET industry = 'Crypto'  
191    WHERE industry LIKE '%Crypto%';  
192  
193 •  SET SQL_SAFE_UPDATES = 1; -- Re-enable safe updates
```

## Code

```
195 -- View Distinct value of stage
196 • SELECT DISTINCT stage
197 FROM layoffs_staging2
198 ORDER BY stage;
199 -- 'Stage' looks good apart from the unknown values
```

stage
HULL
Acquired
Post-IPO
Private Equity
Seed
Series A
Series B
Series C
Series D
Series E
Series F
Series G
Series H
Series I
Series J
Subsidiary
Unknown

## Output

## Code

```
201 • SELECT DISTINCT country
202 FROM layoffs_staging2
203 ORDER BY country;
```

## Output

country
Norway
Pakistan
Peru
Poland
Portugal
Romania
Russia
Senegal
Seychelles
Singapore
South Africa
South Korea
Spain
Sweden
Switzerland
Thailand
Turkey
United Arab Emirates
United Kingdom
United States
United States.
Uruguay
Vietnam

```
205 -- There is 'United States' and 'United States'.
206 -- The 2nd entry will be renamed.
```

## Code

```
206      -- The 2nd entry will be renamed.  
207  
208 •  SET SQL_SAFE_UPDATES = 0; -- Disable Safe updates  
209  
210      -- The types in the text columns have been corrected.  
211  
212 •  UPDATE layoffs_staging2  
213      SET country = 'United States'  
214      WHERE country = 'United States.' ;  
215  
216 •  SET SQL_SAFE_UPDATES = 1;   -- Re-enable safe updates  
217      -- 4 rows were changed  
218  
219      -- The 'date' column was deliberately loaded as a text column. It will be changed into a DATE format  
220  
221 •  SET SQL_SAFE_UPDATES = 0; -- Disable Safe updates  
222  
223 •  UPDATE layoffs_staging2  
224      SET `date` = STR_TO_DATE(`date` , '%m/%d/%Y')  
225      WHERE `date` IS NOT NULL;  
226  
227 •  ALTER TABLE layoffs_staging2  
228      MODIFY COLUMN `date` DATE;  
229  
230 •  SET SQL_SAFE_UPDATES = 1;   -- Re-enable safe updates  
231  
232  
233      -- The 'percentage_laid_off' column is not expressed in percent.  
234      -- To make it into a percentage, it will be multiplied by 100, making the values more interpretable.  
235 •  SET SQL_SAFE_UPDATES = 0; -- Disable Safe updates  
236 •  UPDATE layoffs_staging2  
237      SET percentage_laid_off = ROUND(100 * percentage_laid_off, 2) ;  
238 •  SET SQL_SAFE_UPDATES = 1;   -- Re-enable safe updates
```

## Sample Output

### Code

```

240    -- Display the updated table
241 •  SELECT *
242    FROM layoffs_staging2;
243
244    -- Observing the table, the changes have been successfully updated.

```

	company	location	industry	total_laid_off	percentage_laid_off	date	stage	country	funds_raised_millions	row_num
▶	E Inc.	Toronto	Transportation	NULL	NULL	2022-12-16	Post-IPO	Canada	NULL	1
	Included Health	SF Bay Area	Healthcare	NULL	0.06	2022-07-25	Series E	United States	272	1
	&Open	Dublin	Marketing	9	0.09	2022-11-17	Series A	Ireland	35	1
	#Paid	Toronto	Marketing	19	0.17	2023-01-27	Series B	Canada	21	1
	100 Thieves	Los Angeles	Consumer	12	NULL	2022-07-13	Series C	United States	120	1
	100 Thieves	Los Angeles	Retail	NULL	NULL	2023-01-10	Series C	United States	120	1
	10X Genomics	SF Bay Area	Healthcare	100	0.08	2022-08-04	Post-IPO	United States	242	1
	1stdibs	New York City	Retail	70	0.17	2020-04-02	Series D	United States	253	1
	2TM	Sao Paulo	Crypto	90	0.12	2022-06-01	Unknown	Brazil	250	1
	2TM	Sao Paulo	Crypto	100	0.15	2022-09-01	Unknown	Brazil	250	1
	2U	Washington D.C.	Education	NULL	0.2	2022-07-28	Post-IPO	United States	426	1
	54gene	Washington D.C.	Healthcare	95	0.3	2022-08-29	Series B	United States	44	1
	5B Solar	Sydney	Energy	NULL	0.25	2022-06-03	Series A	Australia	12	1
	6sense	SF Bay Area	Sales	150	0.1	2022-10-12	Series E	United States	426	1
	80 Acres Farms	Cincinnati	Food	NULL	0.1	2023-01-18	Unknown	United States	275	1
	8x8	SF Bay Area	Support	155	0.07	2023-01-18	Post-IPO	United States	253	1
	8x8	SF Bay Area	Support	200	0.09	2022-10-04	Post-IPO	United States	253	1
	98point6	Seattle	Healthcare	NULL	0.1	2022-07-21	Series E	United States	247	1
	99	Sao Paulo	Transportation	75	0.02	2022-09-20	Acquired	Brazil	244	1
	Abra	SF Bay Area	Crypto	12	0.05	2022-06-30	Series C	United States	106	1
	Absci	Vancouver	Healthcare	40	NULL	2022-08-09	Post-IPO	United States	237	1
	Acast	Stockholm	Media	70	0.15	2022-09-15	Post-IPO	Sweden	126	1
	Accolade	Seattle	Healthcare	NULL	NULL	2023-03-03	Post-IPO	United States	458	1
	Acko	Mumbai	Finance	45	0.09	2020-04-01	Unknown	India	143	1
	Acorns	Portland	Finance	50	NULL	2020-05-26	Unknown	United States	207	1
	Actifio	Boston	Data	54	NULL	2020-12-16	Acquired	United States	352	1
	ActiveCampaign	Chicago	Marketing	NULL	0.15	2022-10-03	Series C	United States	360	1

### 3. Handling Null or Blank Values

Code

```
246 ----- 3. Handling Null or Blank Values -----
247
248 -- Observe the table
249 • SELECT *
250 FROM layoffs_staging2;
```

Output

	company	location	industry	total_laid_off	percentage_laid_off	date	stage	country	funds_raised_millions	row_num
▶	E Inc.	Toronto	Transportation	NULL	NULL	2022-12-16	Post-IPO	Canada	NULL	1
	Included Health	SF Bay Area	Healthcare	NULL	0.06	2022-07-25	Series E	United States	272	1
	&Open	Dublin	Marketing	9	0.09	2022-11-17	Series A	Ireland	35	1
	#Paid	Toronto	Marketing	19	0.17	2023-01-27	Series B	Canada	21	1
	100 Thieves	Los Angeles	Consumer	12	NULL	2022-07-13	Series C	United States	120	1
	100 Thieves	Los Angeles	Retail	NULL	NULL	2023-01-10	Series C	United States	120	1
	10X Genomics	SF Bay Area	Healthcare	100	0.08	2022-08-04	Post-IPO	United States	242	1
	1stdibs	New York City	Retail	70	0.17	2020-04-02	Series D	United States	253	1
	2TM	Sao Paulo	Crypto	90	0.12	2022-06-01	Unknown	Brazil	250	1
	2TM	Sao Paulo	Crypto	100	0.15	2022-09-01	Unknown	Brazil	250	1
	2U	Washington D.C.	Education	NULL	0.2	2022-07-28	Post-IPO	United States	426	1
	54gene	Washington D.C.	Healthcare	95	0.3	2022-08-29	Series B	United States	44	1
	5B Solar	Sydney	Energy	NULL	0.25	2022-06-03	Series A	Australia	12	1
	6sense	SF Bay Area	Sales	150	0.1	2022-10-12	Series E	United States	426	1
	80 Acres Farms	Cincinnati	Food	NULL	0.1	2023-01-18	Unknown	United States	275	1
	8x8	SF Bay Area	Support	155	0.07	2023-01-18	Post-IPO	United States	253	1
	8x8	SF Bay Area	Support	200	0.09	2022-10-04	Post-IPO	United States	253	1
	98point6	Seattle	Healthcare	NULL	0.1	2022-07-21	Series E	United States	247	1
	99	Sao Paulo	Transportation	75	0.02	2022-09-20	Acquired	Brazil	244	1
	Abra	SF Bay Area	Crypto	12	0.05	2022-06-30	Series C	United States	106	1
	Absci	Vancouver	Healthcare	40	NULL	2022-08-09	Post-IPO	United States	237	1
	Acast	Stockholm	Media	70	0.15	2022-09-15	Post-IPO	Sweden	126	1
	Accolade	Seattle	Healthcare	NULL	NULL	2023-03-03	Post-IPO	United States	458	1
	Acko	Mumbai	Finance	45	0.09	2020-04-01	Unknown	India	143	1
	Acorns	Portland	Finance	50	NULL	2020-05-26	Unknown	United States	207	1
	Actifio	Boston	Data	54	NULL	2020-12-16	Acquired	United States	352	1
	ActiveCampaign	Chicago	Marketing	NULL	0.15	2022-10-03	Series C	United States	360	1

252 -- There are several fields that are either NULL or missing values (blanks)

Code

```
254    -- Let's start with 'company'  
255 •  SELECT *  
256    FROM layoffs_staging2  
257    WHERE company IS NULL  
258    OR company = '';  
259    -- No nulls or blanks
```

Output

	company	location	industry	total_laid_off	percentage_laid_off	date	stage	country	funds_raised_millions	row_num
--	---------	----------	----------	----------------	---------------------	------	-------	---------	-----------------------	---------

Code

```
260  
261    -- location  
262 •  SELECT *  
263    FROM layoffs_staging2  
264    WHERE location IS NULL  
265    OR location = '';  
266    -- No nulls or blanks
```

Code

```
272    -- industry  
273 •  SELECT *  
274    FROM layoffs_staging2  
275    WHERE industry IS NULL  
276    OR industry = '';  
277    -- There is 1 null and 3 blanks.
```

Output

	company	location	industry	total_laid_off	percentage_laid_off	date	stage	country	funds_raised_millions	row_num
--	---------	----------	----------	----------------	---------------------	------	-------	---------	-----------------------	---------

Output

	company	location	industry	total_laid_off	percentage_laid_off	date	stage	country	funds_raised_millions	row_num
▶	Airbnb	SF Bay Area		30	NULL	2023-03-03	Post-IPO	United States	6400	1
	Bally's Interactive	Providence	NULL	NULL	1500	2023-01-18	Post-IPO	United States	946	1
	Carvana	Phoenix		2500	1200	2022-05-10	Post-IPO	United States	1600	1
	Juul	SF Bay Area		400	3000	2022-11-10	Unknown	United States	1500	1

```
279    -- These missing values should be imputed if possible  
280    -- using CTE (Common Table Expressions) to impute missing values
```

## Code

```
282 •   SET SQL_SAFE_UPDATES = 0; -- Disable Safe updates
283
284 •   WITH ImputedIndustries AS (
285       SELECT
286           company,
287           industry,
288           MAX(industry) OVER (PARTITION BY company) AS imputed_industry
289       FROM
290           layoffs_staging2
291   )
292
293     UPDATE layoffs_staging2 ls2
294     JOIN ImputedIndustries i
295         ON ls2.company = i.company
296         AND ls2.industry IS NULL
297         OR ls2.industry = ''
298     SET
299         ls2.industry = i.imputed_industry
300     WHERE
301         ls2.industry IS NULL
302         OR ls2.industry = '';
303
304 •   SET SQL_SAFE_UPDATES = 1; -- Re-enable safe updates
```

## Code

```
306      -- check industry for nulls or blanks
307      -- industry
308 •   SELECT *
309      FROM layoffs_staging2
310      WHERE industry IS NULL
311      OR industry = '';
312
313      -- Bally's Interactive still has a null for industry
```

## Output

	company	location	industry	total_laid_off	percentage_laid_off	date	stage	country	funds_raised_millions	row_num
▶	Bally's Interactive	Providence	NULL	NULL	1500	2023-01-18	Post-IPO	United States	946	1

```
314
315 •   SELECT *
316      FROM layoffs_staging2
317      WHERE company LIKE '%Bally%';
318
319      -- This is the only value for Bally's Interactive.
320      -- A search in Google finds that Bally's is an online casino.
```

	company	location	industry	total_laid_off	percentage_laid_off	date	stage	country	funds_raised_millions	row_num
▶	Bally's Interactive	Providence	NULL	NULL	1500	2023-01-18	Post-IPO	United States	946	1

## Code

```
322 •  SELECT DISTINCT industry  
323    FROM layoffs_staging2  
324    ORDER BY industry;
```

## Output

industry
Consumer
Crypto
Data
Education
Energy
Fin-Tech
Finance
Fitness
Food
Hardware
Healthcare
HR
Infrastructure
Legal
Logistics
Manufacturing

## Code

```
336    -- Recheck to see if the update was successful.  
337 •  SELECT *  
338    FROM layoffs_staging2  
339    WHERE company LIKE '%Bally%';  
340    -- The update was successful
```

## Output

	company	location	industry	total_laid_off	percentage_laid_off
▶	Bally's Interactive	Providence	Gaming	NULL	1500

## Code

```
326    -- None of the existing listed industries appear to have a category similar to gaming or entertainment.  
327    -- I will impute it as 'Gaming'  
328 •  SET SQL_SAFE_UPDATES = 0; -- Disable Safe updates  
329  
330 •  UPDATE layoffs_staging2  
331    SET industry = 'Gaming'  
332    WHERE company LIKE 'Bally%';  
333  
334 •  SET SQL_SAFE_UPDATES = 1; -- Re-enable safe updates
```

## Code

```
342    -- check 'stage' for nulls or blanks
343 •  SELECT *
344    FROM layoffs_staging2
345   WHERE stage IS NULL
346  OR stage = '';
347    -- There are 6 NULLs in stage, however it is not possible to determine what stage the layoffs were at.
348    -- They will have to remain as NULL values
```

## Output

company	location	industry	total_laid_off	percentage_laid_off	date	stage	country	funds_raised_millions	row_num
Advata	Seattle	Healthcare	32	2100	2022-10-28	NULL	United States	NULL	1
Gatherly	Atlanta	Marketing	NULL	5000	2022-07-31	NULL	United States	NULL	1
Relevel	Bengaluru	HR	40	2000	2023-01-10	NULL	India	NULL	1
Spreetail	Austin	Retail	NULL	NULL	2022-10-27	NULL	United States	NULL	1
Verily	SF Bay Area	Healthcare	250	1500	2023-01-11	NULL	United States	3500	1
Zapp	London	Food	NULL	1000	2022-05-25	NULL	United Kingdom	300	1

## Code

```
350    -- check 'country' for nulls or blanks
351 •  SELECT *
352    FROM layoffs_staging2
353   WHERE country IS NULL
354  OR country = '';
355    -- There are no NULLs or missing values in 'country'.
356
357    -- NULLs and missing values are also observed in the integer fields.
358    -- These values cannot be imputed based on other information available in this table.
```

## Output

	company	location	industry	total_laid_off	percentage_laid_off	date	stage	country	funds_raised_millions	row_num

## Code

```
360    -- Let's observe where total_laid_off and percentage_laid_off are both null or blank
361 •  SELECT COUNT(*) AS Count_Nulls
362  FROM layoffs_staging2
363 WHERE total_laid_off IS NULL
364 OR percentage_laid_off IS NULL;
365 -- There are 1162 rows where either total_laid_off or percentage_laid_off is NULL.
366 -- We still have some useful information if one of those two columns is not blank.alter
```

## Output

	Count_Nulls
▶	1162

## Code

```
367
368    -- Let's observe where they are both blank
369 •  SELECT COUNT(*) AS CountNulls
370  FROM layoffs_staging2
371 WHERE total_laid_off IS NULL
372 AND percentage_laid_off IS NULL;
373 -- There are 361 rows where they are both blank. This is a lot.
```

## Output

	CountNulls
▶	361

## Code

```
375    -- Let's observe these
376 •  SELECT *
377  FROM layoffs_staging2
378 WHERE total_laid_off IS NULL
379 AND percentage_laid_off IS NULL;
```

## Output

	company	location	industry	total_laid_off	percentage_laid_off	date	stage	country	funds_raised_millions	row_num
▶	E Inc.	Toronto	Transportation	NULL	NULL	2022-12-16	Post-IPO	Canada	NULL	1
	100 Thieves	Los Angeles	Retail	NULL	NULL	2023-01-10	Series C	United States	120	1
	Accolade	Seattle	Healthcare	NULL	NULL	2023-03-03	Post-IPO	United States	458	1
	Ada	Toronto	Support	NULL	NULL	2023-02-01	Series C	Canada	190	1
	Adara	SF Bay Area	Travel	NULL	NULL	2020-03-31	Series C	United States	67	1
	Addi	Bogota	Finance	NULL	NULL	2022-06-14	Series C	Colombia	376	1
	AirMap	Los Angeles	Aerospace	NULL	NULL	2020-04-30	Unknown	United States	75	1
	Airtasker	Sydney	Consumer	NULL	NULL	2022-07-04	Series C	Australia	26	1
	Akerna	Denver	Logistics	NULL	NULL	2022-05-27	Unknown	United States	46	1
	Akerna	Denver	Logistics	NULL	NULL	2020-09-02	Post-IPO	United States	NULL	1
	Alegion	Austin	Data	NULL	NULL	2020-04-03	Series A	United States	16	1
	Alerzo	Ibadan	Retail	NULL	NULL	2022-09-02	Series B	Nigeria	16	1
	AllyO	SF Bay Area	HR	NULL	NULL	2020-04-03	Series B	United States	64	1
	Almanac	SF Bay Area	Other	NULL	NULL	2022-08-13	Series A	United States	45	1
	Alto Pharm...	SF Bay Area	Healthcare	NULL	NULL	2022-07-14	Series E	United States	560	1
	Amobee	SF Bay Area	Marketing	NULL	NULL	2022-11-09	Acquired	United States	72	1

## Code

```
381 -- Where both are NULL, these rows will be removed as no information of the layoff numbers is present.  
382 • SET SQL_SAFE_UPDATES = 0; -- Disable Safe updates  
383  
384 • DELETE  
385   FROM layoffs_staging2  
386   WHERE total_laid_off IS NULL  
387   AND percentage_laid_off IS NULL;  
388  
389 • SET SQL_SAFE_UPDATES = 1; -- Re-enable safe updates  
390  
391 -- Observe the table after the updates  
392 • SELECT *  
393   FROM layoffs_staging2;
```

Many null or blank values have been removed and the dataset is looking a lot tidier.  
It was not possible to remove all as the information in the rest of the rows are still important.  
Imputing is not advised for this project as these numbers are very difficult to estimate and could range significantly.

## Output

company	location	industry	total_laid_off	percentage_laid_off	date	stage	country	funds_raised_millions	row_num
54gene	Washington D.C.	Healthcare	95	3000	2022-08-29	Series B	United States	44	1
5B Solar	Sydney	Energy	NULL	2500	2022-06-03	Series A	Australia	12	1
6sense	SF Bay Area	Sales	150	1000	2022-10-12	Series E	United States	426	1
80 Acres Farms	Cincinnati	Food	NULL	1000	2023-01-18	Unknown	United States	275	1
8x8	SF Bay Area	Support	155	700	2023-01-18	Post-IPO	United States	253	1
8x8	SF Bay Area	Support	200	900	2022-10-04	Post-IPO	United States	253	1
98point6	Seattle	Healthcare	NULL	1000	2022-07-21	Series E	United States	247	1
99	Sao Paulo	Transportation	75	200	2022-09-20	Acquired	Brazil	244	1
Abra	SF Bay Area	Crypto	12	500	2022-06-30	Series C	United States	106	1
Absci	Vancouver	Healthcare	40	NULL	2022-08-09	Post-IPO	United States	237	1
Acast	Stockholm	Media	70	1500	2022-09-15	Post-IPO	Sweden	126	1
Acko	Mumbai	Finance	45	900	2020-04-01	Unknown	India	143	1
Acorns	Portland	Finance	50	NULL	2020-05-26	Unknown	United States	207	1
Actifio	Boston	Data	54	NULL	2020-12-16	Acquired	United States	352	1
ActiveCampaign	Chicago	Marketing	NULL	1500	2022-10-03	Series C	United States	360	1
Ada	Toronto	Support	78	1600	2022-09-20	Series C	Canada	190	1
Ada Health	Berlin	Healthcare	50	NULL	2022-10-17	Series B	Germany	189	1
Ada Support	Toronto	Support	36	2300	2020-04-24	Series B	Canada	60	1
Adaptive Biote...	Seattle	Healthcare	100	1200	2022-03-08	Post-IPO	United States	406	1
Addepar	SF Bay Area	Finance	20	300	2023-01-18	Series F	United States	491	1
Adobe	SF Bay Area	Marketing	100	NULL	2022-12-06	Post-IPO	United States	2	1
AdRoll	Salt Lake City	Marketing	210	3000	2020-03-31	Series C	United States	89	1
Advata	Seattle	Healthcare	32	2100	2022-10-28	NULL	United States	NULL	1

## 4. Remove Unnecessary Columns

Code

```
400 ----- 4. Remove Unnecessary Columns: -----
401
402 -- Column 'row_num' is no longer needed. It will be dropped.
403 -- All of the original columns will be kept
404 • ALTER TABLE layoffs_staging2
405   DROP COLUMN row_num;
406
407 • SELECT *
408   FROM layoffs_staging2;
409
410 -- The data has been cleaned as best as possible for this dataset.
411
412 ----- Conclusion -----
413 /*
414 Data cleaning is a very important process that ensures data accuracy and reliability.
415 By removing duplicates, standardizing the data, handling missing values, and eliminating unnecessary columns,
416 we can clean and improve the quality of the dataset and thereby improve the outcomes of our data analysis.
417 This project demonstrates the detailed steps involved in data cleaning through SQL and highlights the importance
418 in maintaining the integrity of data for insightful and actionable analysis.
419 */
```

Output

company	location	industry	total_laid_off	percentage_laid_off	date	stage	country	funds_raised_millions
Included Health	SF Bay Area	Healthcare	NULL	600	2022-07-25	Series E	United States	272
&Open	Dublin	Marketing	9	900	2022-11-17	Series A	Ireland	35
#Paid	Toronto	Marketing	19	1700	2023-01-27	Series B	Canada	21
100 Thieves	Los Angeles	Consumer	12	NULL	2022-07-13	Series C	United States	120
10X Genomics	SF Bay Area	Healthcare	100	800	2022-08-04	Post-IPO	United States	242
1stdibs	New York City	Retail	70	1700	2020-04-02	Series D	United States	253
2TM	Sao Paulo	Crypto	90	1200	2022-06-01	Unknown	Brazil	250
2TM	Sao Paulo	Crypto	100	1500	2022-09-01	Unknown	Brazil	250
2U	Washington D.C.	Education	NULL	2000	2022-07-28	Post-IPO	United States	426

# Exploratory Data Analysis

Code

```
422 ----- Exploratory Data Analysis -----
423 -- After cleaning, let's continue to have a look and reveal some of the insights in this dataset.
424
425 -- First, let's see what time period this dataset covers.
426 • SELECT MIN(date) AS `Earliest Date`
427     ,MAX(date) AS `Latest Date`
428 FROM layoffs_staging2;
429 -- This dataset covers layoffs during and after the Covid 19 Pandemic
430 -- spanning three years from the 11th of March 2020 to the 6th of March 2023.
```

Code

```
432
433 -- Total Number of Layoffs in the data set and the total sum of funds raised.
434 • SELECT SUM(total_laid_off) AS Total_Layoffs
435     ,SUM(funds_raised_millions) AS `Funds Raised ($Millions)`
436 FROM layoffs_staging2;
437
438 -- For values where there are no NULLs or blanks, there were a total of:
439 -- 383,659 layoffs
440 -- and $1.601450 billion raised in funds.
441
```

Code

```
442 -- Where percentage laid off is 100%, this entails that the company folded and went bankrupt.
443 -- Let's see how many companies folded.
444 • SELECT COUNT(total_laid_off) AS NumberOfInsolventCompanies
445 FROM layoffs_staging2
446 WHERE percentage_laid_off = 100;
447
448 -- A total of 43 companies went into insolvency.
```

Output

Earliest Date	Latest Date
2020-03-11	2023-03-06

Output

	Total_Layoffs	Funds Raised (\$Millions)
▶	383659	1601450

	NumberOfInsolventCompanies
▶	43

## Code

```
449
450      -- Many companies had layoffs in different stages or regions.
451      -- Let's see the total number of layoffs by each company, sorted in descending order.
452 •   SELECT company, SUM(total_laid_off) AS `Total Laid Off By Company`
453     FROM layoffs_staging2
454     GROUP BY company
455     ORDER BY 2 DESC;
```

## Output

company	Total Laid Off By Company
Amazon	18150
Google	12000
Meta	11000
Salesforce	10090
Microsoft	10000
Philips	10000
Ericsson	8500
Uber	7585
Dell	6650
Booking.com	4601
Flockjay	37
Springbig	37
Ada Support	36
Culture Amp	36
Hireology	36
Labelbox	36
Plum	36
Anodot	35
Copper	35
Dynamic Sig...	35
Frontdesk	35
Jetty	35
Kabam	35
Linkfire	35
Monese	35

```
457      -- Amazon, Google and Meta had the largest number of layoffs
458      -- The least amount of total layoffs in this dataset were only 35.
```

## Code

```
460    -- Let's observe which industries had the largest number of layoffs.  
461 • SELECT industry, SUM(total_laid_off) AS `Total Laid Off By Company`  
462     FROM layoffs_staging2  
463     GROUP BY industry  
464     ORDER BY 2 DESC;  
465  
466    -- The Consumer and Retail industries had the largest number of layoffs with  
467    -- 44,782 and 43,613 layoffs respectfully.  
468    -- Manufacturing was least affect with only 20 layoffs  
469    -- Gaming is the new industry we identified earlier.  
470    -- As ementioned, we did not have figures for Ballys.
```

## Output

	industry	Total Laid Off By Company
▶	Consumer	44782
	Retail	43613
	Other	36289
	Transportation	31248
	Finance	28344
	Healthcare	25953
	Food	22855
	Real Estate	17565

industry	Total Laid Off By Company
Travel	17129
Hardware	13828
Education	13338
Sales	13216
Marketing	13188
Crypto	10693
Fitness	8748
Security	5979
Infrastructure	5785
Media	5234
Data	5135
Logistics	4026
Construction	3863
Support	3523
HR	2783
Recruiting	2775
Product	1233
Legal	836
Energy	802
Aerospace	661
Fin-Tech	215
Manufacturing	20
Gaming	NULL

## Code

```
472    -- Let's observe which countries were had the largest number of layoffs.  
473 • SELECT country, SUM(total_laid_off) AS `Total Laid Off By Country`  
474     FROM layoffs_staging2  
475     GROUP BY country  
476     ORDER BY 2 DESC;  
477  
478    -- The USA and India had the largest number of layoffs with  
479    -- 256,559 and 35,993 layoffs respectfully. These countries also have the largest populations.
```

## Output

	country	Total Laid Off By Country
▶	United States	256559
	India	35993
	Netherlands	17220
	Sweden	11264
	Brazil	10391
	Germany	8701
	United Kingdom	6398
	Canada	6319
	Singapore	5995
	China	5905
	Israel	3638
	Indonesia	3521
	Australia	2324
	Nigeria	1882
	United Arab Emirates	995
	France	915
	Hong Kong	730
	Austria	570
	Russia	400
	Kenya	349
	Estonia	333
	Argentina	323
	Senegal	300
	Mexico	270

	country	Total Laid Off By Country
	Myanmar	200
	Norway	140
	Colombia	130
	Bulgaria	120
	Portugal	115
	Malaysia	100
	Japan	85
	Romania	80
	Seychelles	75
	Switzerland	62
	Lithuania	60
	Thailand	55
	Luxembourg	45
	New Zealand	45
	Chile	30
	Poland	25
	Pakistan	NULL
	Hungary	NULL
	Italy	NULL
	Turkey	NULL
	South Korea	NULL
	Vietnam	NULL
	Egypt	NULL

## Code

```

482 -- Let's see how layoffs were affected over time, say per month.
483 -- Ensure the date column is in DATE format (This was already done)
484 -- ALTER TABLE your_table_name
485 -- MODIFY date DATE;
486
487 -- Extract year and month, and aggregate the data.
488 -- A Common Table Expression will be used to help create a rolling total.
489 • WITH monthly_layoffs AS (
490   SELECT
491     DATE_FORMAT(date, '%Y-%m') AS `Year and Month`,
492     SUM(total_laid_off) AS `Monthly Layoffs`
493   FROM layoffs_staging2
494   WHERE DATE_FORMAT(date, '%Y-%m') IS NOT NULL
495   GROUP BY DATE_FORMAT(date, '%Y-%m')
496 )
497   SELECT
498     `Year and Month`,
499     `Monthly Layoffs`,
500     SUM(`Monthly Layoffs`) OVER (ORDER BY `Year and Month`) AS `Accumulated Layoffs`
501   FROM
502     monthly_layoffs
503   ORDER BY
504     `Year and Month`;
505
506 -- The code above produces a table which is a time-series of total layoffs per month and the accumulation of layoffs by month
507 -- from the companies in this dataset.
508 -- April and May of 2020 had heavy layoffs of 26,710 and 25,804 respectfully. This is just after lockdowns started appearing worldwide.
509 -- Layoffs have also increased considerably towards the end of the timeframe in this dataset.
510 -- There were 53,451 layoffs in November 2022 and the largest number per month was January 2023 with 84,714.
511 -- February 2023 was also large with 36493 layoffs.
512 -- This table could be exported to perform time-series analysis and plots.

```

## Output

Year and Month	Monthly Layoffs	Accumulated Layoffs	Year and Month	Monthly Layoffs	Accumulated Layoffs
2020-03	9628	9628	2021-04	261	88987
2020-04	26710	36338	2021-06	2434	91421
2020-05	25804	62142	2021-07	80	91501
2020-06	7627	69769	2021-08	1867	93368
2020-07	7112	76881	2021-09	161	93529
2020-08	1969	78850	2021-10	22	93551
2020-09	609	79459	2021-11	2070	95621
2020-10	450	79909	2021-12	1200	96821
2020-11	237	80146	2022-01	510	97331
2020-12	852	80998	2022-02	3685	101016
2021-01	6813	87811	2022-03	5714	106730
2021-02	868	88679	2022-04	4128	110858
2021-03	47	88726	2022-05	12885	123743
			2022-06	17394	141137
			2022-07	16223	157360
			2022-08	13055	170415
			2022-09	5881	176296
			2022-10	17406	193702
			2022-11	53451	247153
			2022-12	10329	257482
			2023-01	84714	342196
			2023-02	36493	378689
			2023-03	4470	383159

## Code

```
516 -- Let's see layoffs by year and industry using the rank() function.  
517 -- The RANK() function ranks companies within each industry and year based on total layoffs in descending order.  
518 -- Equal values receive the same rank, and the next rank is skipped to account for the tie.  
519 • SELECT  
520     company,  
521     YEAR(`date`) AS `Year`,  
522     industry,  
523     location,  
524     total_laid_off,  
525     -- funds_raised_millions,  
526     SUM(total_laid_off) OVER(PARTITION BY YEAR(`date`), industry) AS `Total Laid Off by Year And Industry`,  
527     RANK() OVER(PARTITION BY industry, YEAR(`date`) ORDER BY total_laid_off DESC) AS `Rank By Year And Industry`  
528 FROM layoffs_staging2  
529 WHERE  
530     YEAR(`date`) IS NOT NULL  
531     AND industry IS NOT NULL  
532 ORDER BY `Year`, industry, `Rank By Year And Industry`;
```

## Output

	company	Year	industry	location	total_laid_off	Total Laid Off by Year And Industry	Rank By Year And Industry
▶	OneWeb	2020	Aerospace	London	451	561	1
	Kitty Hawk	2020	Aerospace	SF Bay Area	70	561	2
	Astra	2020	Aerospace	SF Bay Area	40	561	3
	Katerra	2020	Construction	SF Bay Area	400	896	1
	Katerra	2020	Construction	SF Bay Area	240	896	2
	Procure	2020	Construction	Los Angeles	180	896	3
	Raken	2020	Construction	San Diego	60	896	4
	Rhumbix	2020	Construction	SF Bay Area	16	896	5
	Nearmap	2020	Construction	Sydney	NULL	896	6
	Magic Leap	2020	Consumer	Miami	1000	6063	1
	Yelp	2020	Consumer	SF Bay Area	1000	6063	1
	Juul	2020	Consumer	SF Bay Area	900	6063	3
	Eventbrite	2020	Consumer	SF Bay Area	500	6063	4
	BookMySh...	2020	Consumer	Mumbai	270	6063	5
	Thumbtack	2020	Consumer	SF Bay Area	250	6063	6
	HOOQ	2020	Consumer	Singapore	250	6063	6
	Mozilla	2020	Consumer	SF Bay Area	250	6063	6
	GoPro	2020	Consumer	SF Bay Area	200	6063	9
	Booksby	2020	Consumer	SF Bay Area	200	6063	9
	StubHub	2020	Consumer	SF Bay Area	200	6063	9
	Rover	2020	Consumer	Seattle	194	6063	12

## Code

```
535 -- This query identifies the top 5 companies with the highest layoffs for each year using Common Table Expressions (CTEs).
536 --
537 -- Step 1: The first CTE (CTE1) groups the data by company and year, summing up the total number of layoffs per company per year.
538 -- Step 2: The second CTE (CTE2) ranks companies within each year based on the total layoffs in descending order.
539 -- Step 3: The final query filters the ranked data to include only the top 5 companies per year and sorts it by year and rank.
540 -- This approach demonstrates a common method for producing rankings using CTEs.
541
542 • Ⓜ WITH CTE1 AS (
543     SELECT
544         company,
545         YEAR(date) AS `Year`,
546         SUM(total_laid_off) AS `Total Laid Off`
547     FROM
548         layoffs_staging2
549     WHERE
550         YEAR(date) IS NOT NULL
551     GROUP BY
552         company,
553         YEAR(date)
554 ),
555 Ⓜ CTE2 AS (
556     SELECT
557         company,
558         `Year`,
559         `Total Laid Off`,
560         RANK() OVER(PARTITION BY `Year` ORDER BY `Total Laid Off` DESC) AS `Ranking`
561     FROM
562         CTE1
563 )
564     SELECT *
565     FROM
566         CTE2
567     WHERE
568         `Ranking` <= 5 -- Filters for the top 5 companies with the highest layoffs per year.
569     ORDER BY
570         `Year`,
571         `Ranking`; -- Orders the results by year and ranking.
572
```

# Output

	company	Year	Total Laid Off	Ranking
▶	Uber	2020	7525	1
	Booking.com	2020	4375	2
	Groupon	2020	2800	3
	Swiggy	2020	2250	4
	Airbnb	2020	1900	5
	Bytedance	2021	3600	1
	Katerra	2021	2434	2
	Zillow	2021	2000	3
	Instacart	2021	1877	4
	WhiteHat Jr	2021	1800	5
	Meta	2022	11000	1
	Amazon	2022	10150	2
	Cisco	2022	4100	3
	Peloton	2022	4084	4
	Carvana	2022	4000	5
	Philips	2022	4000	5
	Google	2023	12000	1
	Microsoft	2023	10000	2
	Ericsson	2023	8500	3
	Amazon	2023	8000	4
	Salesforce	2023	8000	4

```
572
573    -- Output Explanation:
574    -- The output lists the top 5 companies with the highest layoffs for each year, sorted by year and rank.
575    -- For example:
576    -- In 2020, Uber ranks #1 with 7,525 layoffs, followed by Booking.com with 4,375 layoffs.
577    -- In 2022, Meta ranks #1 with 11,000 layoffs, while Philips and Carvana tie at rank #5 with 4,000 layoffs each.
578    -- Tied rankings (e.g., rank #5 in 2022) indicate multiple companies had the same number of layoffs, and the next rank is skipped.
```