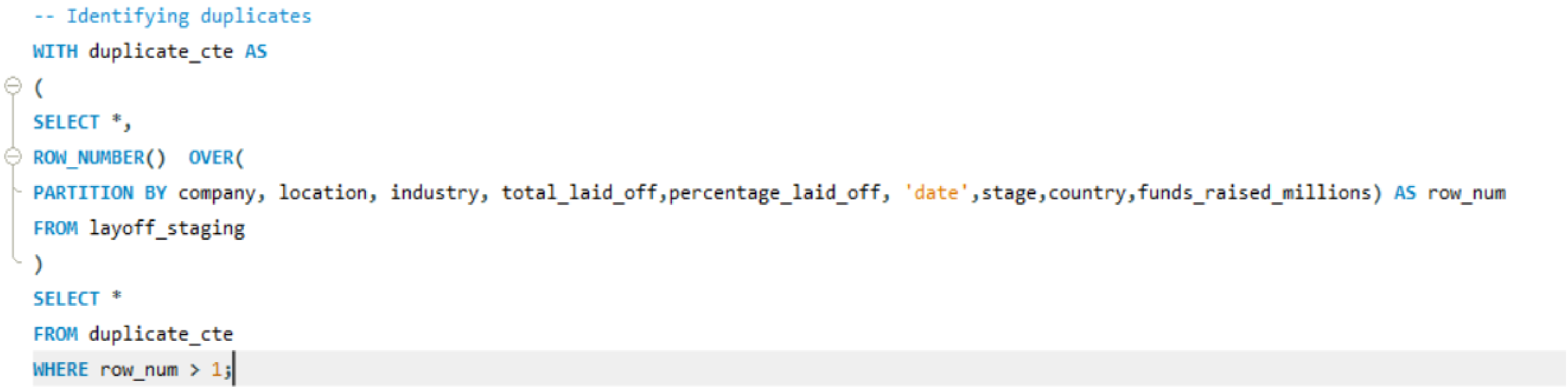
PROJECT TITLE : DATA WRANGLING USING SQL

CONTENT

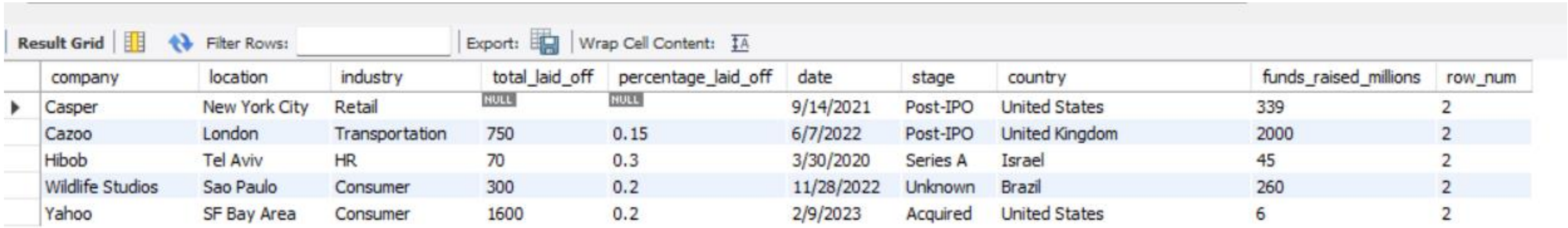
1. Remove Duplicates
2. Standardizing the data
3. Dealing with Null Values or Blank values
4.Remove Unnecessary Columns

1. Remove Duplicates

1.Identifying duplicate rows using window functions

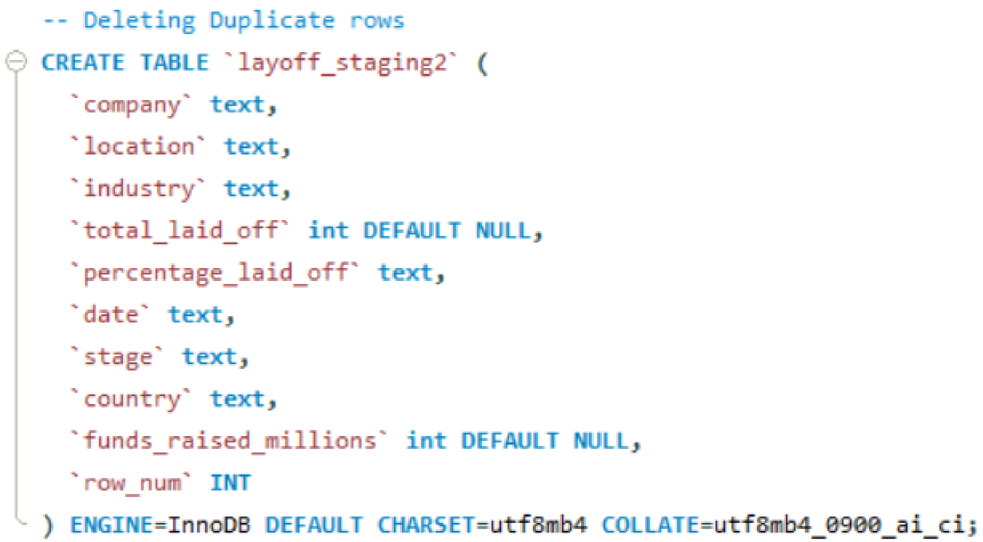


```
-- Identifying duplicates
WITH duplicate_cte AS
(
SELECT *,
ROW_NUMBER()  OVER(
PARTITION BY company, location, industry, total_laid_off,percentage_laid_off, 'date',stage,country,funds_raised_millions) AS row_num
FROM layoff_staging
)
SELECT *
FROM duplicate_cte
WHERE row_num > 1;
```



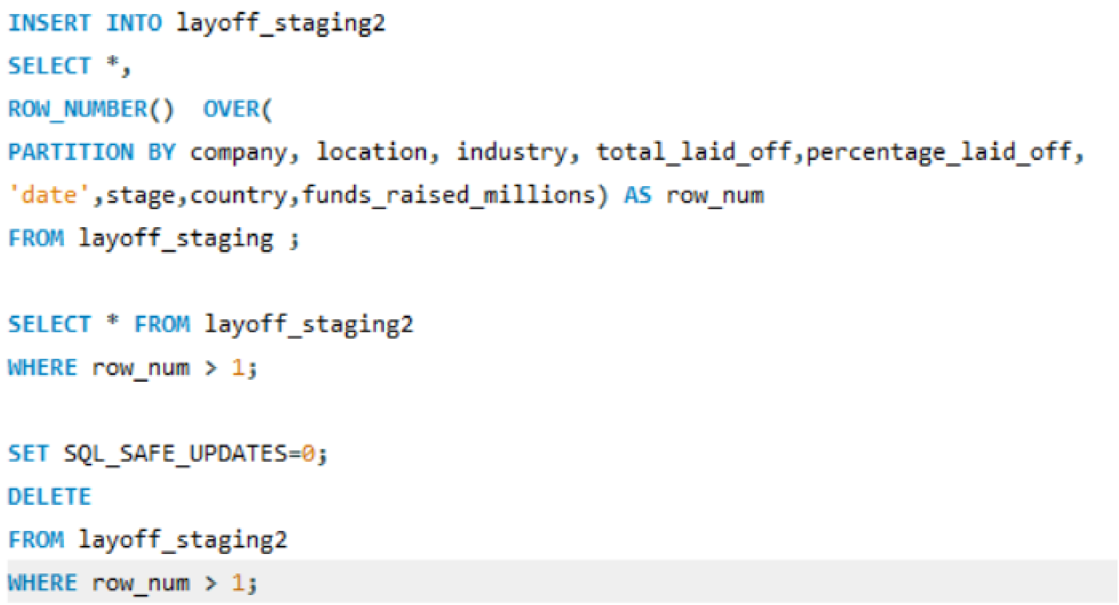
company	location	industry	total_laid_off	percentage_laid_off	date	stage	country	funds_raised_millions	row_num
Casper	New York City	Retail	NULL	NULL	9/14/2021	Post-IPO	United States	339	2
Cazoo	London	Transportation	750	0.15	6/7/2022	Post-IPO	United Kingdom	2000	2
Hibob	Tel Aviv	HR	70	0.3	3/30/2020	Series 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

2.Creating a new table with the same schema



```
-- Deleting Duplicate rows
CREATE TABLE `layoff_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;
```

3.Insert raw data and remove duplicates

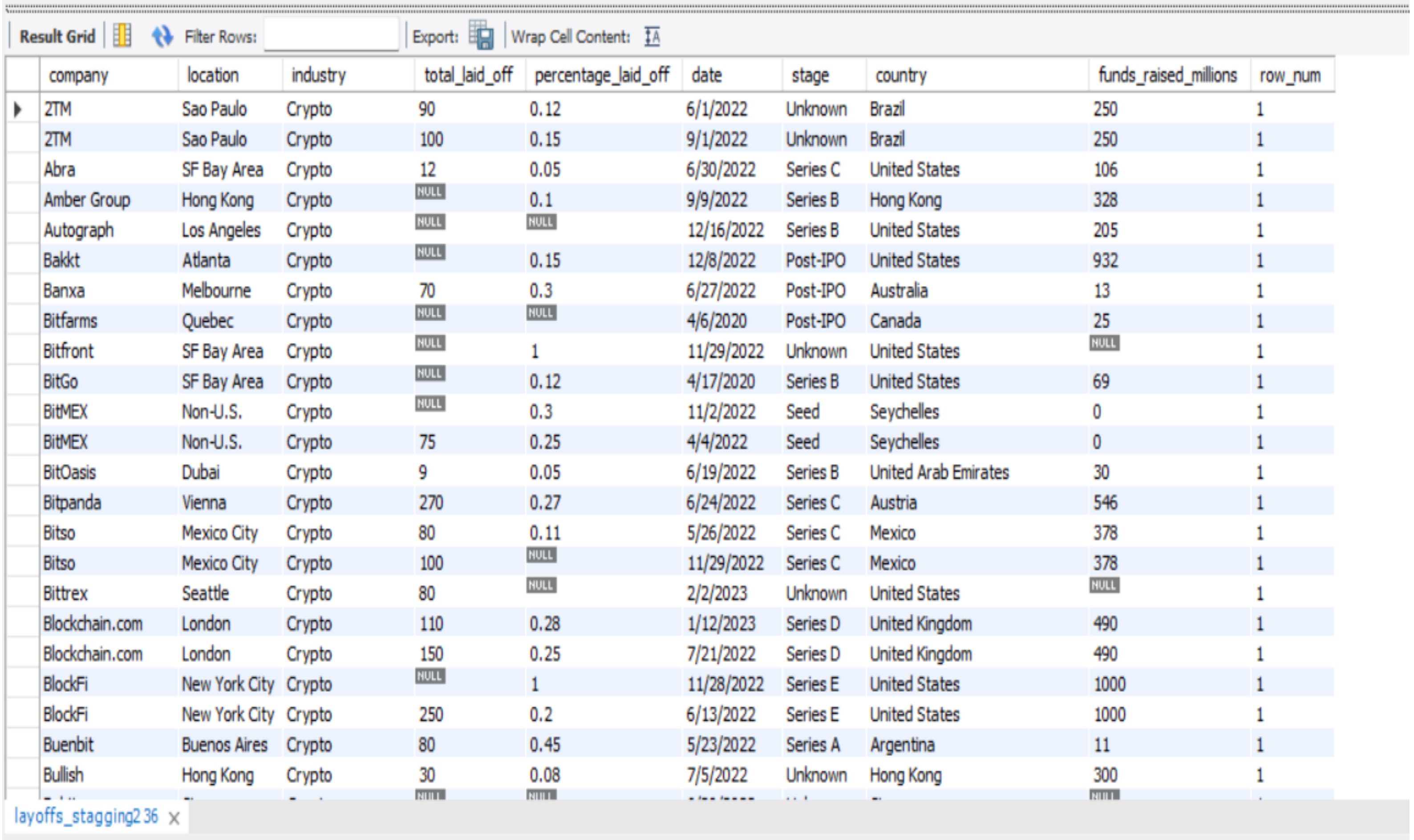


```
INSERT INTO layoff_staging2
SELECT *,
ROW_NUMBER()  OVER(
PARTITION BY company, location, industry, total_laid_off,percentage_laid_off,
'date',stage,country,funds_raised_millions) AS row_num
FROM layoff_staging ;

SELECT * FROM layoff_staging2
WHERE row_num > 1;

SET SQL_SAFE_UPDATES=0;
DELETE
FROM layoff_staging2
WHERE row_num > 1;
```

2. Standardizing the data

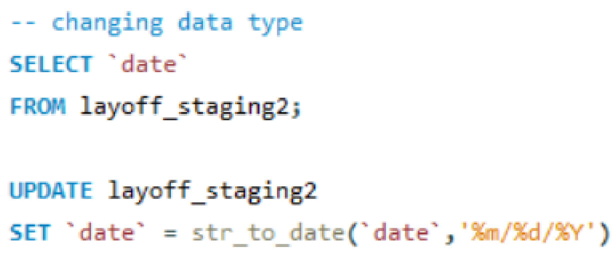
1.Standardising string

```
-- 2. Standardize the data
SELECT company, TRIM(company)
FROM layoff_staging2;

UPDATE layoff_staging2
SET company = TRIM(company);

SELECT *
FROM layoff_staging2
WHERE industry LIKE 'Crypto%';

UPDATE layoff_staging2
SET industry = 'Crypto'
WHERE industry LIKE 'Crypro';

SELECT DISTINCT industry
FROM layoff_staging2;

SELECT DISTINCT country, TRIM( TRAILING '.' FROM  country)
FROM layoff_staging2
ORDER BY 1;

UPDATE layoff_staging2
SET country = TRIM( TRAILING '.' FROM  country)
WHERE country LIKE 'United States%';

SELECT DISTINCT country
FROM layoff_staging2;
```

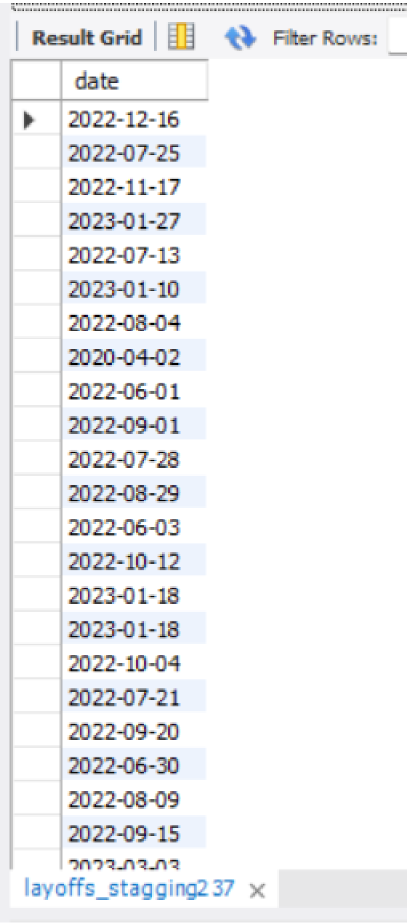


company	location	industry	total_laid_off	percentage_laid_off	date	stage	country	funds_raised_millions	row_num
2TM	Sao Paulo	Crypto	90	0.12	6/1/2022	Unknown	Brazil	250	1
2TM	Sao Paulo	Crypto	100	0.15	9/1/2022	Unknown	Brazil	250	1
Abra	SF Bay Area	Crypto	12	0.05	6/30/2022	Series C	United States	106	1
Amber 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
BitGo	SF Bay Area	Crypto	NULL	0.12	4/17/2020	Series B	United States	69	1
BitMEX	Non-U.S.	Crypto	NULL	0.3	11/2/2022	Seed	Seychelles	0	1
BitMEX	Non-U.S.	Crypto	75	0.25	4/4/2022	Seed	Seychelles	0	1
BitOasis	Dubai	Crypto	9	0.05	6/19/2022	Series B	United Arab Emirates	30	1
Bitpanda	Vienna	Crypto	270	0.27	6/24/2022	Series C	Austria	546	1
Bitso	Mexico City	Crypto	80	0.11	5/26/2022	Series C	Mexico	378	1
Bitso	Mexico City	Crypto	100	NULL	11/29/2022	Series C	Mexico	378	1
Bittrex	Seattle	Crypto	80	NULL	2/2/2023	Unknown	United States	NULL	1
Blockchain.com	London	Crypto	110	0.28	1/12/2023	Series D	United Kingdom	490	1
Blockchain.com	London	Crypto	150	0.25	7/21/2022	Series D	United Kingdom	490	1
BlockFi	New York City	Crypto	NULL	1	11/28/2022	Series E	United States	1000	1
BlockFi	New York City	Crypto	250	0.2	6/13/2022	Series E	United States	1000	1
Buenbit	Buenos Aires	Crypto	80	0.45	5/23/2022	Series A	Argentina	11	1
Bullish	Hong Kong	Crypto	30	0.08	7/5/2022	Unknown	Hong Kong	300	1

2.Changing data type



```
-- changing data type
SELECT `date`
FROM layoff_staging2;

UPDATE layoff_staging2
SET `date` = str_to_date(`date`,'%m/%d/%Y')
```

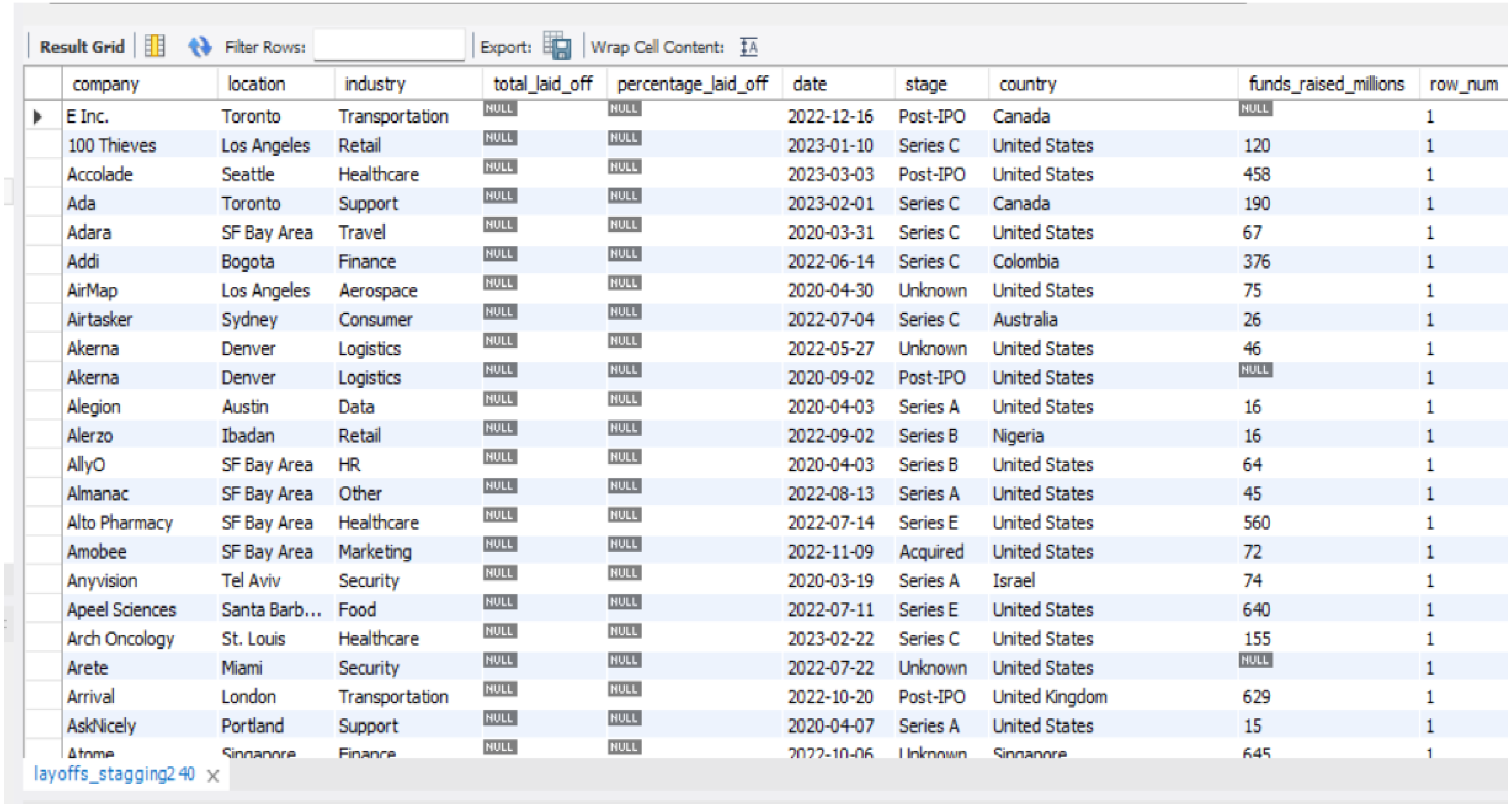


```
date
2022-12-16
2022-07-25
2022-11-17
2023-01-27
2022-07-13
2023-01-10
2022-08-04
2020-04-02
2022-06-01
2022-09-01
2022-07-28
2022-08-29
2022-06-03
2022-10-12
2023-01-18
2023-01-18
2022-10-04
2022-07-21
2022-09-20
2022-06-30
2022-08-09
2022-09-15
2023-03-03
```

3. Dealing with Null Values or Blank values

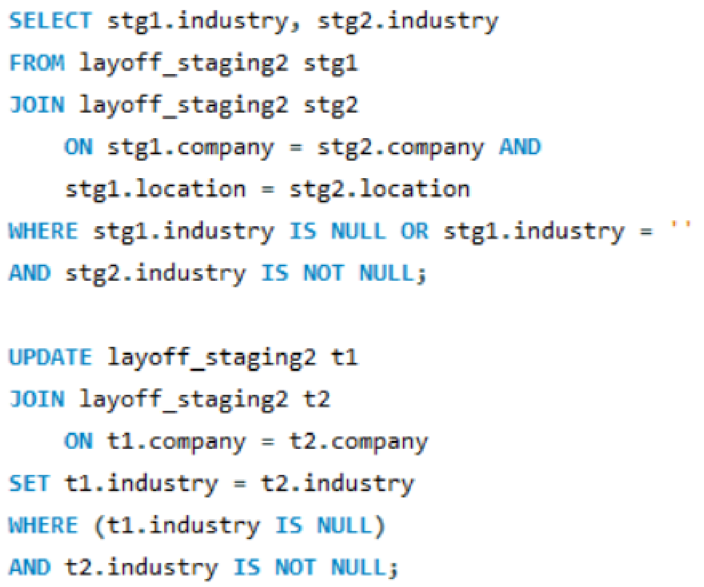
1.Identifying null values

```
-- 3. Null Values or Blank values

SELECT *
FROM layoff_staging2
WHERE total_laid_off IS NULL
AND percentage_laid_off IS NULL;


SELECT *
FROM layoff_staging2
WHERE industry IS NULL OR industry = '';

SELECT *
FROM layoff_staging2
WHERE company LIKE  "Bally's Interactive";

UPDATE layoff_staging2
SET industry = NULL
WHERE industry = '';
```



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 Pharmacy	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
Anyvision	Tel Aviv	Security	NULL	NULL	2020-03-19	Series A	Israel	74	1
Apeel Sciences	Santa Barb...	Food	NULL	NULL	2022-07-11	Series E	United States	640	1
Arch Oncology	St. Louis	Healthcare	NULL	NULL	2023-02-22	Series C	United States	155	1
Arete	Miami	Security	NULL	NULL	2022-07-22	Unknown	United States	NULL	1
Arrival	London	Transportation	NULL	NULL	2022-10-20	Post-IPO	United Kingdom	629	1
AskNicely	Portland	Support	NULL	NULL	2020-04-07	Series A	United States	15	1
Atome	Singapore	Finance	NULL	NULL	2022-10-06	Unknown	Singapore	645	1

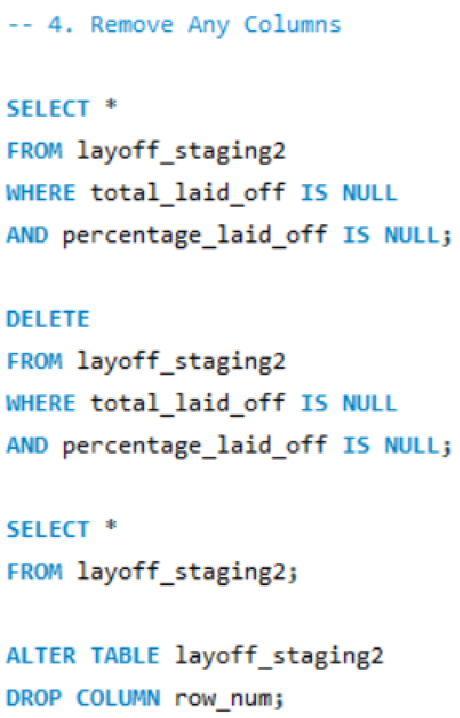
2.Standardising null values



```sql
SELECT stg1.industry, stg2.industry
FROM layoff_staging2 stg1
JOIN layoff_staging2 stg2
    ON stg1.company = stg2.company AND
    stg1.location = stg2.location
WHERE stg1.industry IS NULL OR stg1.industry = ''
AND stg2.industry IS NOT NULL;

UPDATE layoff_staging2 t1
JOIN layoff_staging2 t2
    ON t1.company = t2.company
SET t1.industry = t2.industry
WHERE (t1.industry IS NULL)
AND t2.industry IS NOT NULL;
```

4. Remove Unnecessary Columns

1.Remove unwanted rows and columns



```
-- 4. Remove Any Columns

SELECT *
FROM layoff_staging2
WHERE total_laid_off IS NULL
AND percentage_laid_off IS NULL;

DELETE
FROM layoff_staging2
WHERE total_laid_off IS NULL
AND percentage_laid_off IS NULL;

SELECT *
FROM layoff_staging2;

ALTER TABLE layoff_staging2
DROP COLUMN row_num;
```