

DATA CLEANING

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator: dc1* x dc2* SQL File 4* SQL File 5* SQL File 6* SQL File 8* SQL File 7* SQL File 9* SQL File 10* SQL File 11*

SCHEMAS

Filter objects

- ecommerce
- sys
- world_layouts**
 - Tables
 - layoffs
 - layoffs_staging
 - layoffs_staging2**
 - Views
 - Stored Procedures
 - Functions

Administration Schemas Information

Table:
layoffs_staging2

Columns:

company	text
location	text
industry	text
total_laid_off	int
percentage_laid_off	text
date	date
stage	text
country	text
funds_raised_millions	int

```
1  -- REMOVE DUPLICATES
2
3  • with duplicate_cte as
4  (
5      select * ,
6      row_number() over(partition by company ,location , industry , total_laid_off , percentage_laid_off , date ,stage , country,
7      funds_raised_millions) as row_num
8      from layoffs_staging
9  )
10
11  select * from duplicate_cte
12  where row_num > 1;
13
14
15  • CREATE TABLE `layoffs_staging2` (
16      `company` text,
17      `location` text,
18      `industry` text,
19      `total_laid_off` int DEFAULT NULL,
20      `percentage_laid_off` text,
21      `date` text,
22      `stage` text,
23      `country` text,
24      `funds_raised_millions` int DEFAULT NULL,
25      `row_num` int
26  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
27
28
29  • insert into layoffs_staging2
```

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator: dc1* x dc2* SQL File 4* SQL File 5* SQL File 6* SQL File 8* SQL File 7* SQL File 9* SQL File 10* SQL File 11*

SCHEMAS

Filter objects

- ecommerce
- sys
- world_layouts
 - Tables
 - layoffs
 - layoffs_staging
 - layoffs_staging2
 - Views
 - Stored Procedures
 - Functions

Table: layoffs_staging2

Columns:

company	text
location	text
industry	text
total_laid_off	int
percentage_laid_off	text
date	date
stage	text
country	text
funds_raised_millions	int

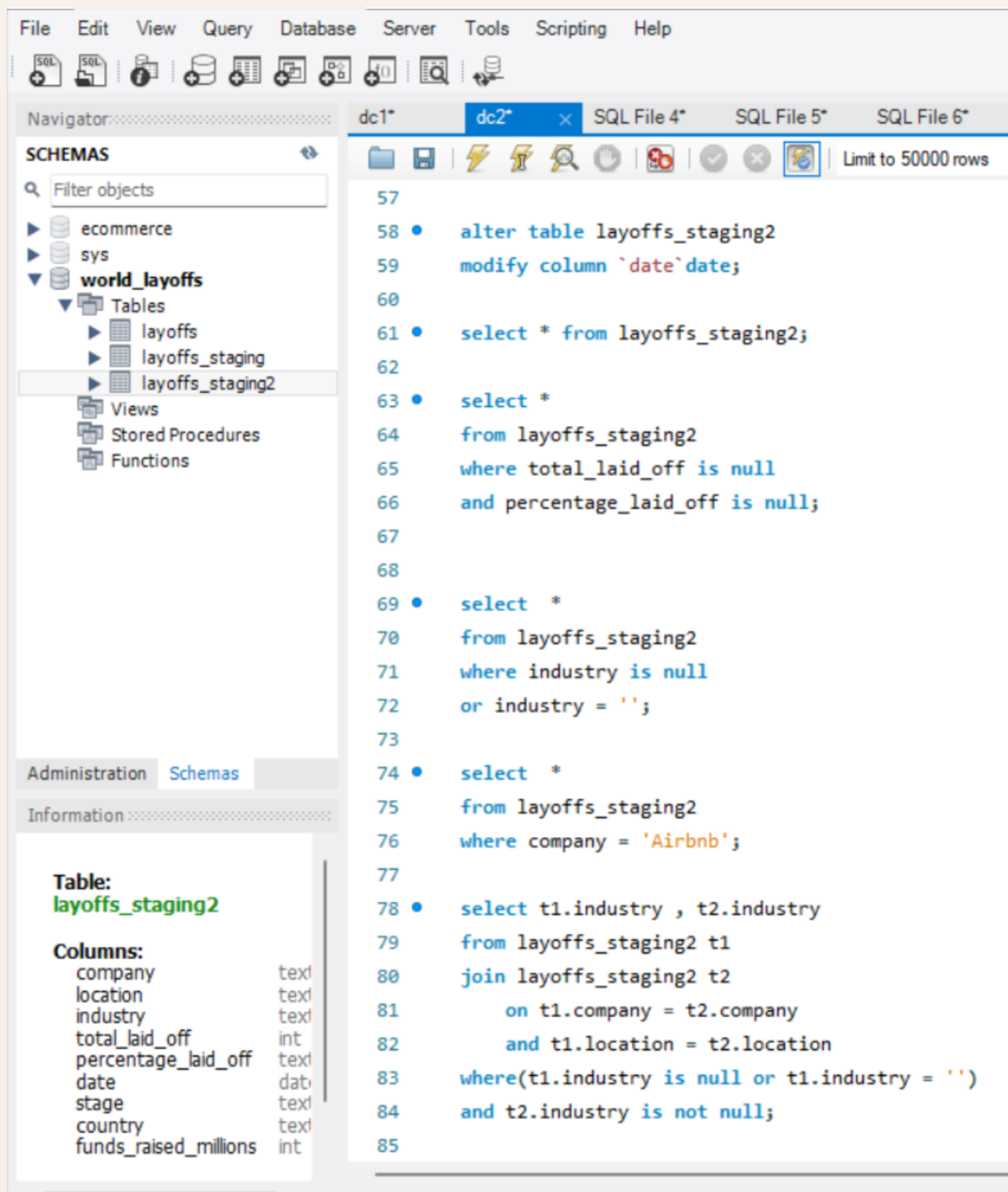
```
20 `percentage_laid_off` text,
21 `date` text,
22 `stage` text,
23 `country` text,
24 `funds_raised_millions` int DEFAULT NULL,
25 `row_num` int
26 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
27
28
29 • insert into layoffs_staging2
30
31 select * ,
32 row_number() over(partition by company ,location , industry , total_laid_off , percentage_laid_off , date ,stage , country,
33 funds_raised_millions) as row_num
34 from layoffs_staging;
35
36
37 • SET SQL_SAFE_UPDATES = 0;
38 • delete from layoffs_staging2
39 where row_num>1;
40 • SET SQL_SAFE_UPDATES = 1;
41
42
43 • select * from layoffs_staging2;
44
45
46
47
```


The screenshot displays the SQL Server Enterprise Manager interface. On the left, the 'Schemas' pane shows the 'world_layoffs' database with tables 'layoffs', 'layoffs_staging', and 'layoffs_staging2'. The 'Table: layoffs_staging2' is selected, showing columns: company, location, industry, total_laid_off, percentage_laid_off, date, stage, country, and funds_raised_millions. The main window shows a SQL query script with 54 lines of code. The script includes several SELECT and UPDATE statements, using functions like TRIM, STR_TO_DATE, and DATE_FORMAT. The 'Output' pane at the bottom is empty.

```

26 • select distinct country
27   from layoffs_staging2
28  order by 1;
29
30 • select *
31   from layoffs_staging2
32  where country like 'United states%'
33  order by 1;
34
35 • select distinct country , TRIM(trailing '.' from country)
36   from layoffs_staging2
37  order by 1;
38
39 • update layoffs_staging2
40    set country = TRIM(trailing '.' from country)
41  where country like 'United states%';
42
43
44 • select distinct country , TRIM(trailing '.' from country)
45   from layoffs_staging2
46  order by 1;
47
48 • select date ,
49        STR_TO_DATE (date, '%m/%d/%Y')
50   from layoffs_staging2;
51
52 • update layoffs_staging2
53    set date = STR_TO_DATE (date, '%m/%d/%Y')
54

```

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator: dc1* dc2* x SQL File 4* SQL File 5* SQL File 6* SQL File

SCHEMAS

Filter objects

- ecommerce
- sys
- world_layoffs**
 - Tables
 - layoffs
 - layoffs_staging
 - layoffs_staging2
 - Views
 - Stored Procedures
 - Functions

Administration Schemas Information

Table: **layoffs_staging2**

Columns:

company	text
location	text
industry	text
total_laid_off	int
percentage_laid_off	text
date	date
stage	text
country	text
funds_raised_millions	int

```

85
86 • update layoffs_staging t1
87   join layoffs_staging t2
88     on t1.company = t2.company
89   set t1.industry = t2.industry
90   where(t1.industry is null or t1.industry = '')
91   and t2.industry is not null;
92
93 • select *
94   from layoffs_staging2
95   where industry is null
96   or industry = '';
97
98 • select * from layoffs_staging2;
99
100 • select *
101   from layoffs_staging2
102   where total_laid_off is null
103   and percentage_laid_off is null;
104
105
106 • delete
107   from layoffs_staging2
108   where total_laid_off is null
109   and percentage_laid_off is null;
110
111 • select *
112   from layoffs_staging2;
113

```


MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator: dc1* dc2* x SQL File 4* SQL File 5* SQL File 6* SQL File 8*

SCHEMAS

Filter objects

- ecommerce
- sys
- world_layoffs**
 - Tables
 - layoffs
 - layoffs_staging
 - layoffs_staging2
 - Views
 - Stored Procedures
 - Functions

Administration Schemas Information

Table: **layoffs_staging2**

Columns:

company	text
location	text
industry	text
total_laid_off	int
percentage_laid_off	text
date	date
stage	text
country	text
funds_raised_millions	int

```

95     where industry is null
96     or industry = '';
97
98 •   select * from layoffs_staging2;
99
100 •  select *
101     from layoffs_staging2
102     where total_laid_off is null
103     and percentage_laid_off is null;
104
105
106 •  delete
107     from layoffs_staging2
108     where total_laid_off is null
109     and percentage_laid_off is null;
110
111 •  select *
112     from layoffs_staging2;
113
114 •  alter table layoffs_staging2
115     drop column row_num;
116
117 •  select *
118     from layoffs_staging2;
119
120
121
122

```

Limit to 50000 rows