Here is a data cleaning project I undertook with a sample dataset on meat, poultry, and egg producers. The table name is “meat_poultry_egg_establishments”. The “mpi” in front of the table name refers to the dataset I had to create to clean the data in GCP.

To start off this data cleaning project, I counted the number of rows in the table.

```
SELECT count(*) FROM mpi.meat_poultry_egg_establishments;
```

6,000 results were returned. I then counted how many times an address appeared in the table.

```
SELECT company,
    street,
    city,
    st,
    count(*) AS address_count
FROM mpi.meat_poultry_egg_establishments
GROUP BY company, street, city, st
HAVING count(*) > 1
ORDER BY company, street, city, st;
```

Row	company	street	city	st	address_count
1	Acre Station Meat Farm	17076 Hwy 32 N	Pinetown	NC	2
2	Beltex Corporation	3801 North Grove Street	Fort Worth	TX	2
3	Cloverleaf Cold Storage	111 Imperial Drive	Sanford	NC	2
4	Crete Core Ingredients, LLC	2220 County Road I	Crete	NE	2
5	Crider, Inc.	1 Plant Avenue	Stillmore	GA	3
6	Dimension Marketing & Sales, Inc.	386 West 9400 South	Sandy	UT	2

I then counted how many states appeared in the addresses. I wanted to see if there were any null values here, which there were.

```
SELECT st,
    count(*) AS st_count
FROM mpi.meat_poultry_egg_establishments
GROUP BY st
ORDER BY st;
```

Row	st	st_count
1	*null*	3
2	AK	17
3	AL	93
4	AR	87
5	AS	1
6	AZ	37

I then wanted to see information about facilities where the value for state is null. The following query returned the three rows with null state values.

```
SELECT est_number,
    company,
    city,
    st,
    zip
FROM mpi.meat_poultry_egg_establishments
WHERE st IS NULL;
```

Row	est_number	company	city	st	zip
1	M263A+P263A+V263A	Jones Dairy Farm	*null*	*null*	53538
2	M45319+P45319	Hall-Namie Packing Company, Inc	*null*	*null*	36671
3	V18677A	Atlas Inspection, Inc.	Blaine	*null*	55449

Next I looked for inconsistencies with the names of the companies. Note the four instances of the company Armour-Eckrich Meats, LLC.

```
SELECT company,
    count(*) AS company_count
FROM mpi.meat_poultry_egg_establishments
GROUP BY company
ORDER BY company ASC;
```

Row	company	company_count
305	Armour - Eckrich Meats, LLC	1
306	Armour-Eckrich Meats LLC	3
307	Armour-Eckrich Meats, Inc.	1
308	Armour-Eckrich Meats, LLC	2
309	Arnold's & Eddies Foods Inc.	1
310	Arsho Meat Products	1

Another check performed was consistency with values in the ZIP code field.

```
SELECT length(zip),
    count(*) AS length_count
FROM mpi.meat_poultry_egg_establishments
GROUP BY length(zip)
ORDER BY length(zip) ASC;
```

The following query shows the values in the ZIP column where the ZIP code has less than 5 digits.

```
SELECT st,
    count(*) AS st_count
FROM mpi.meat_poultry_egg_establishments
WHERE length(zip) < 5
GROUP BY st
ORDER BY st ASC;
```

Now that unclean data has been observed, it's time to clean the data. I first backed up the table to preserve data in case of any missteps.

```
CREATE TABLE mpi.meat_poultry_egg_establishments_backup AS
SELECT * FROM mpi.meat_poultry_egg_establishments;
```

This query confirmed that all rows were duplicated. They both returned the same 6000+ value.

```
SELECT
  (SELECT count(*) FROM mpi.meat_poultry_egg_establishments) AS original,
  (SELECT count(*) FROM mpi.meat_poultry_egg_establishments_backup) AS backup;
```

```
UPDATE mpi.meat_poultry_egg_establishments
SET st_copy = st
WHERE st_copy = st;
```

I then started cleaning the data. I started with missing state values as mentioned earlier. Here's an example of one of the queries:

```
UPDATE mpi.meat_poultry_egg_establishments
SET st = 'MN'
WHERE est_number = 'V18677A';
```

Another fix to make is making sure establishment names are consistent. I updated the previous variations of “Armour-Ekrich Meats, LLC” using a wildcard syntax.

```
UPDATE mpi.meat_poultry_egg_establishments
SET company = 'Armour-Eckrich Meats'
WHERE company LIKE 'Armour%';
```

The return following the query stated “This statement modified 7 rows in meat_poultry_egg_establishments”.

Row	company	city	st
1	Armour-Eckrich Meats	Omaha	NE
2	Armour-Eckrich Meats	Springfield	MA
3	Armour-Eckrich Meats	St James	MN
4	Armour-Eckrich Meats	Peru	IN
5	Armour-Eckrich Meats	Junction City	KS
6	Armour-Eckrich Meats	St Charles	IL
7	Armour-Eckrich Meats	Mason City	IA

The following query is an example of altering ZIP codes to fill in missing zeros.

```
UPDATE mpi.meat_poultry_egg_establishments
SET zip = '00' || zip
WHERE st IN('PR','VI') AND length(zip) = 3;
```