## Question 4

*Q: Identify all patient addresses that do not have a valid country or state code.*

***1. Isolate rows where `'Address 1'` is populated.***

In [19]:
```
has_address = charges[['Address 1', 'Address 2', 'City', 'State', 'Zip', 'Country']][charges
has_address.head()
```

Out[19]:

| | Address 1 | Address 2 | City | State | Zip | Country |
|---|---|---|---|---|---|---|
| 0 | 107 Main St. | NaN | West Jordan | UT | 84081 | USA |
| 1 | 863 Main St. | NaN | Salt Lake City | UT | 84106 | USA |
| 2 | 216 Main St. | NaN | Moroni | UT | 84646 | USA |
| 3 | 932 Main St. | NaN | Bluffdale | UT | 84065 | USA |
| 4 | 4 Main St. | NaN | Eagle Mountain | UT | 84005 | USA |

***2. Compare country codes in `charges` table with those in the `countries` table.***

In [20]:
```
# Identify all Country codes that occur in original charges table
set(charges['Country'])
```

Out[20]:
```
{'CA', 'ESP', 'USA', nan}
```

In [21]:
```
# Set of all Country codes that exist in the has_address table
country_charges = set(has_address['Country'])
country_charges
```

Out[21]:
```
{'CA', 'USA', nan}
```

In [22]:
```
# Set of all Country codes that exist in the countries table
country_countries = set(list(countries['Country Abbreviation']))
```

In [23]:
```
# Find invalid Country codes
invalid_country = list(country_charges - country_countries)
invalid_country
```

Out[23]:
```
['USA']
```