ETL Process in Python: Data Extraction to Database Design

By: Sarah Brittle, Harriet Orleans, Rebecca Skinner

What is ETL?

Definition: Extract, Transform, Load

Importance in Data Management and Analytics

Theme of the Project

Databases

National Neighborhood Data Archive (NaNDA): Crimes by County, United States, 2002 - 2014

National Center for Education Statistics

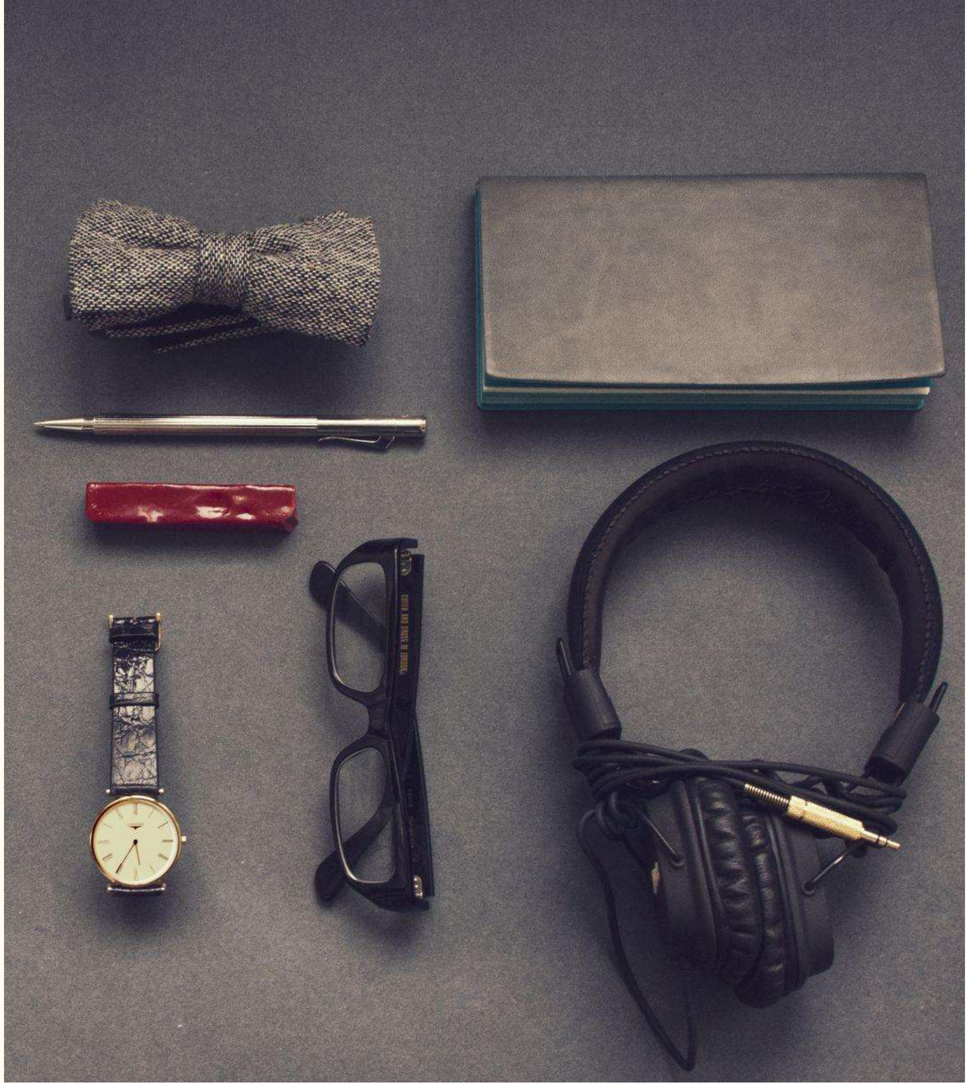
Institute of Museum and Library Services

These databases were selected as they can be helpful for analyzing the social, educational, and safety dynamics of different counties within the United States.

Data Ethics

All sources from government agencies

All allow open use for statistical purposes



Coding Approach

Tools Used

Python Libraries: Pandas, Glob

Transformation: Jupyter Notebook

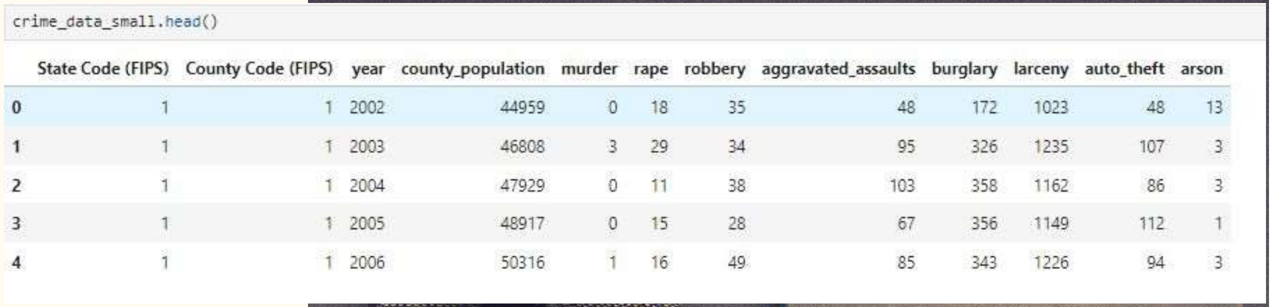
Database: PostgreS

Data Extraction

Overview of Process

From FBI

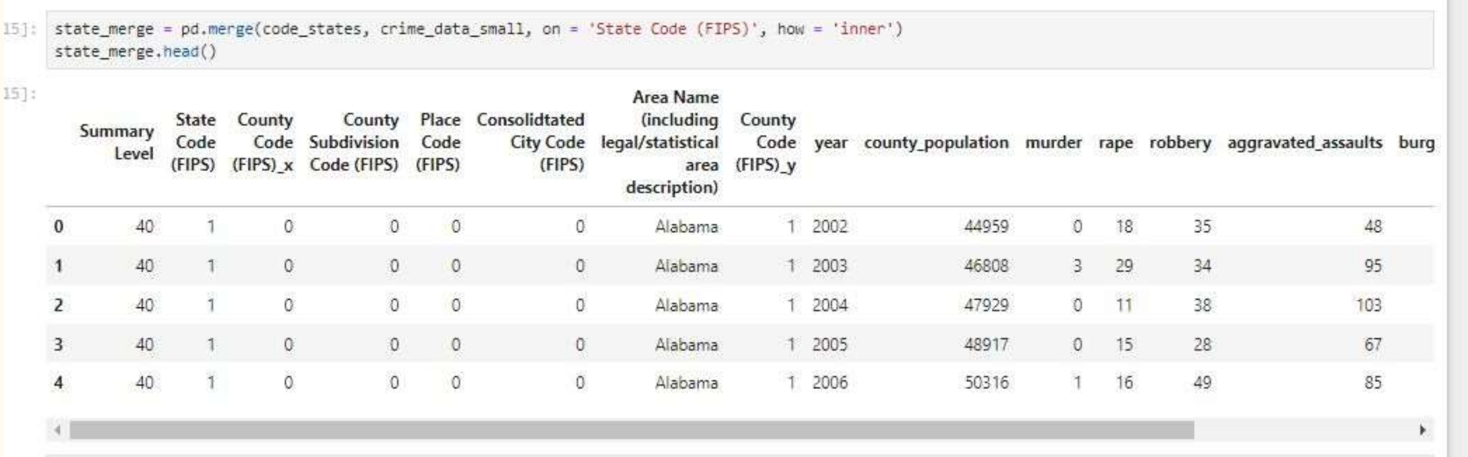
Already turned into a csv



```
crime_data_small.head()
```

	State Code (FIPS)	County Code (FIPS)	year	county_population	murder	rape	robbery	aggravated_assaults	burglary	larceny	auto_theft	arson
0	1	1	2002	44959	0	18	35	48	172	1023	48	13
1	1	1	2003	46808	3	29	34	95	326	1235	107	3
2	1	1	2004	47929	0	11	38	103	358	1162	86	3
3	1	1	2005	48917	0	15	28	67	356	1149	112	1
4	1	1	2006	50316	1	16	49	85	343	1226	94	3

Had to merge with code sheet to get locations



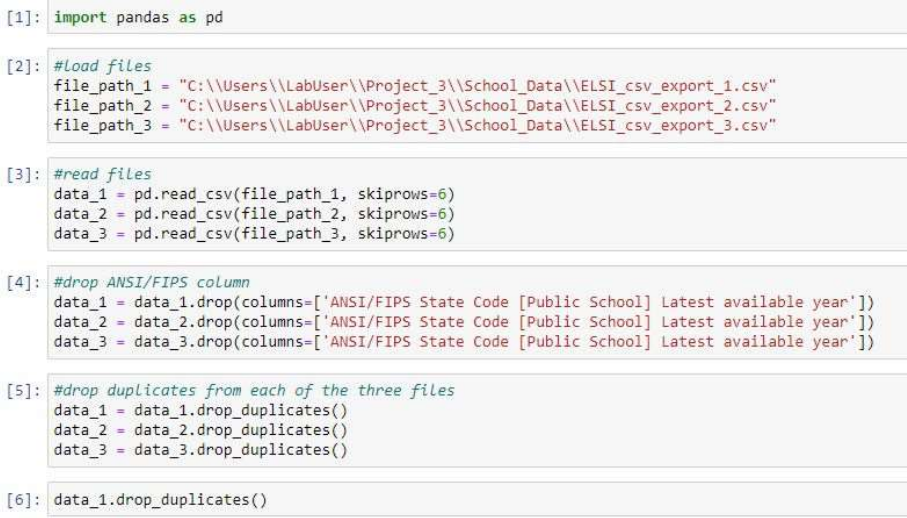
```
15]: state_merge = pd.merge(code_states, crime_data_small, on = 'State Code (FIPS)', how = 'inner')
     state_merge.head()
```

15]:

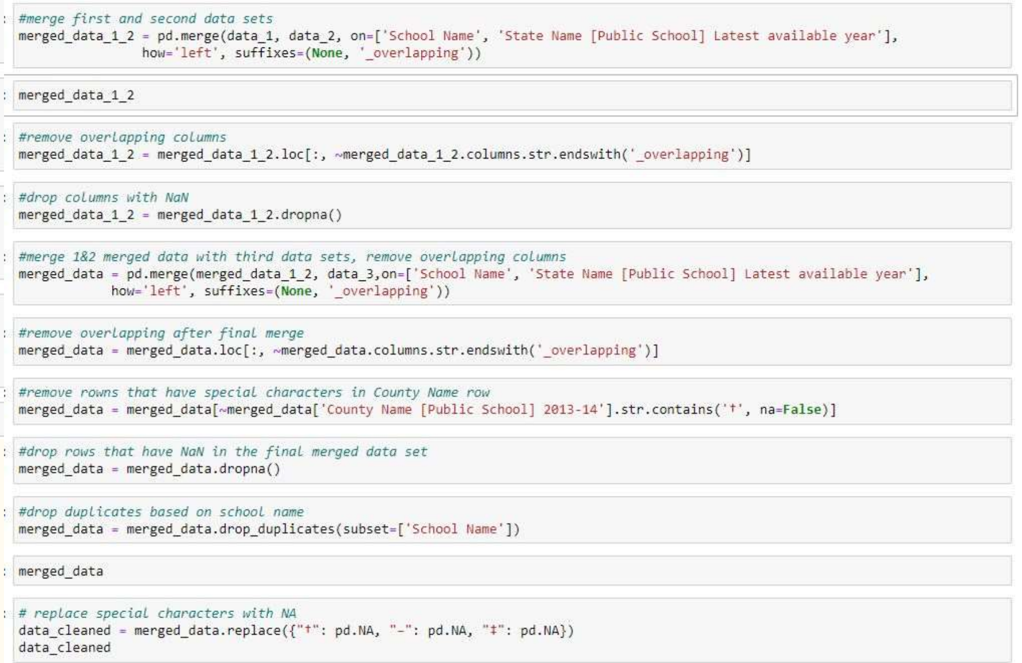
	Summary Level	State Code (FIPS)	County Code (FIPS)_x	County Subdivision Code (FIPS)	Place Code (FIPS)	Consolidtated City Code (FIPS)	Area Name (including legal/statistical area description)	County Code (FIPS)_y	year	county_population	murder	rape	robbery	aggravated_assaults	burg
0	40	1	0	0	0	0	Alabama	1	2002	44959	0	18	35	48	
1	40	1	0	0	0	0	Alabama	1	2003	46808	3	29	34	95	
2	40	1	0	0	0	0	Alabama	1	2004	47929	0	11	38	103	
3	40	1	0	0	0	0	Alabama	1	2005	48917	0	15	28	67	
4	40	1	0	0	0	0	Alabama	1	2006	50316	1	16	49	85	

Data Extraction: National Neighborhood Data Archive (NaNDA): Crimes by County, United States, 2002 - 2014

Overview of Process



```
[1]: import pandas as pd

[2]: #load files
     file_path_1 = "C:\\Users\\LabUser\\Project_3\\School_Data\\ELSI_csv_export_1.csv"
     file_path_2 = "C:\\Users\\LabUser\\Project_3\\School_Data\\ELSI_csv_export_2.csv"
     file_path_3 = "C:\\Users\\LabUser\\Project_3\\School_Data\\ELSI_csv_export_3.csv"

[3]: #read files
     data_1 = pd.read_csv(file_path_1, skiprows=6)
     data_2 = pd.read_csv(file_path_2, skiprows=6)
     data_3 = pd.read_csv(file_path_3, skiprows=6)

[4]: #drop ANSI/FIPS column
     data_1 = data_1.drop(columns=['ANSI/FIPS State Code [Public School] Latest available year'])
     data_2 = data_2.drop(columns=['ANSI/FIPS State Code [Public School] Latest available year'])
     data_3 = data_3.drop(columns=['ANSI/FIPS State Code [Public School] Latest available year'])

[5]: #drop duplicates from each of the three files
     data_1 = data_1.drop_duplicates()
     data_2 = data_2.drop_duplicates()
     data_3 = data_3.drop_duplicates()

[6]: data_1.drop_duplicates()
```

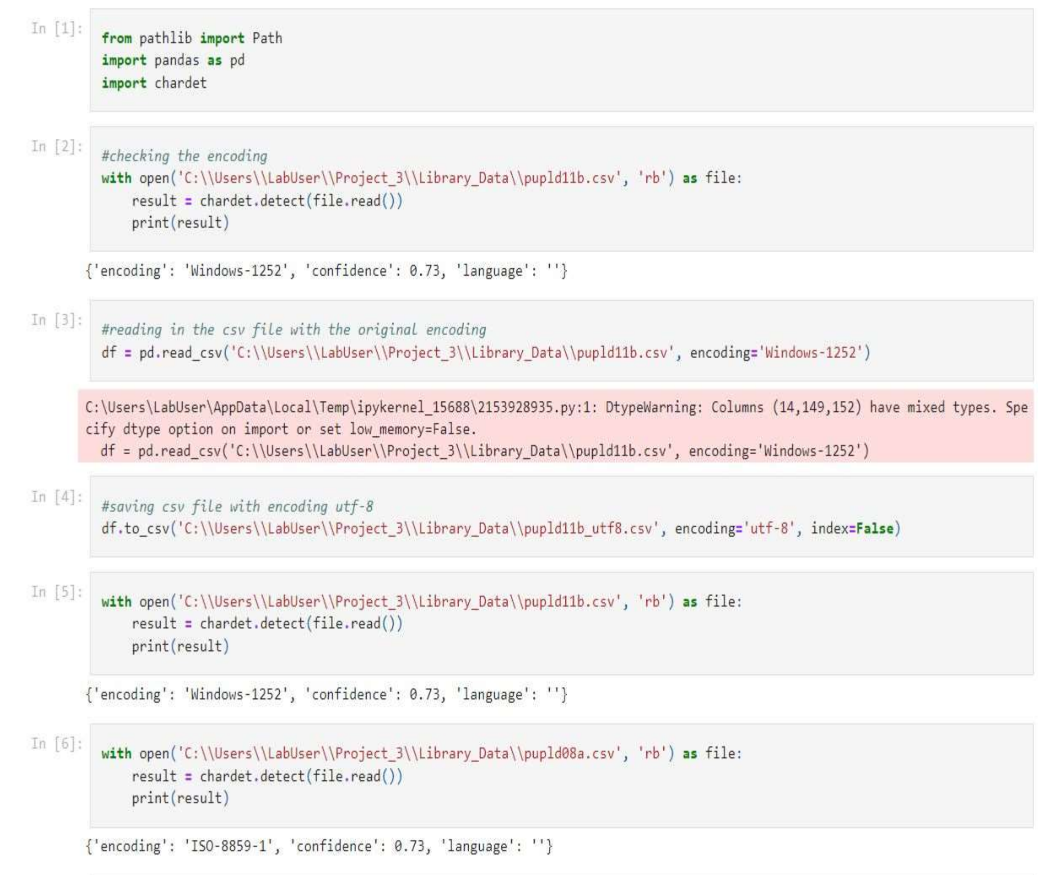


```
#merge first and second data sets
merged_data_1_2 = pd.merge(data_1, data_2, on=['School Name', 'State Name [Public School] Latest available year'],
                how='left', suffixes=(None, '_overlapping'))

merged_data_1_2

#remove overlapping columns
merged_data_1_2 = merged_data_1_2.loc[:, ~merged_data_1_2.columns.str.endswith('_overlapping')]

#drop columns with NaN
merged_data_1_2 = merged_data_1_2.dropna()

#merge 1&2 merged data with third data sets, remove overlapping columns
merged_data = pd.merge(merged_data_1_2, data_3,on=['School Name', 'State Name [Public School] Latest available year'],
            how='left', suffixes=(None, '_overlapping'))

#remove overlapping after final merge
merged_data = merged_data.loc[:, ~merged_data.columns.str.endswith('_overlapping')]

#remove rowns that have special characters in County Name row
merged_data = merged_data[~merged_data['County Name [Public School] 2013-14'].str.contains('†', na=False)]

#drop rows that have NaN in the final merged data set
merged_data = merged_data.dropna()

#drop duplicates based on school name
merged_data = merged_data.drop_duplicates(subset=['School Name'])

merged_data

# replace special characters with NA
data_cleaned = merged_data.replace({"†": pd.NA, "–": pd.NA, "‡": pd.NA})
data_cleaned
```

Data Extraction: National Center for Education Statistics

Overview of Process

I used chardet to encode the csv files to utf-8.
I then read in the csv files using glob.
Then I concatenated the csv files.

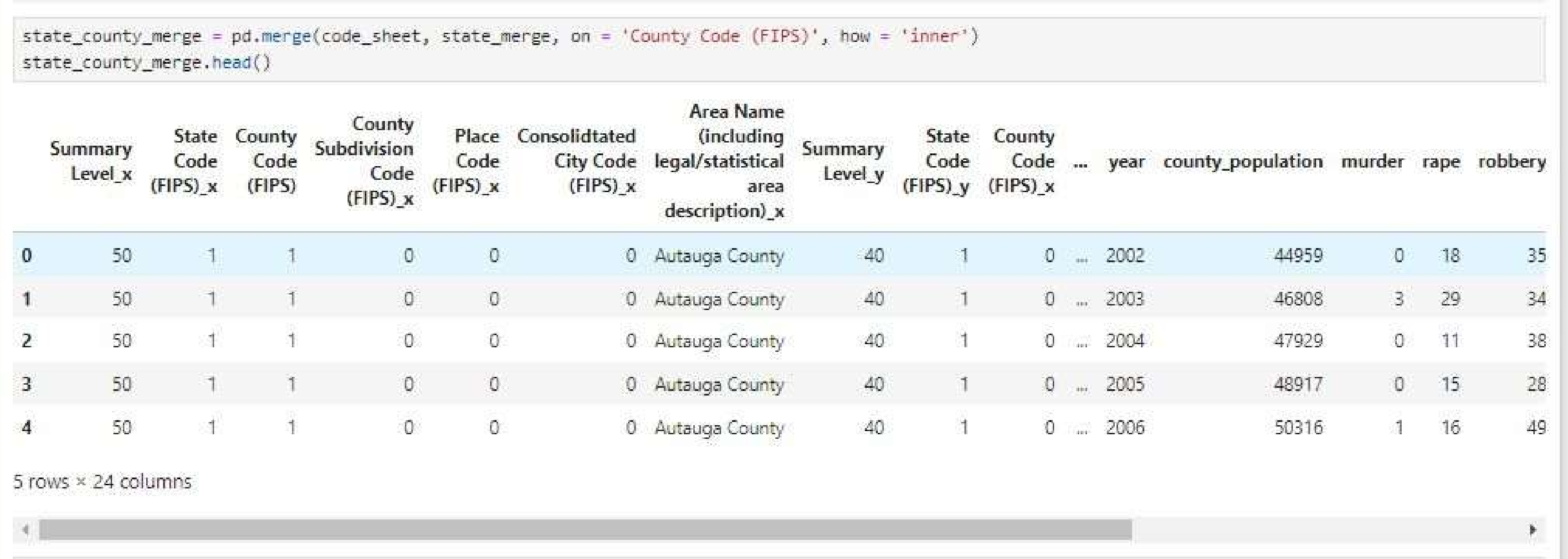


```
In [1]:
from pathlib import Path
import pandas as pd
import chardet

In [2]:
#checking the encoding
with open('C:\\Users\\LabUser\\Project_3\\Library_Data\\pupld11b.csv', 'rb') as file:
    result = chardet.detect(file.read())
    print(result)

{'encoding': 'Windows-1252', 'confidence': 0.73, 'language': ''}

In [3]:
#reading in the csv file with the original encoding
df = pd.read_csv('C:\\Users\\LabUser\\Project_3\\Library_Data\\pupld11b.csv', encoding='Windows-1252')

C:\Users\LabUser\AppData\Local\Temp\ipykernel_15688\2153928935.py:1: DtypeWarning: Columns (14,149,152) have mixed types. Spe
cify dtype option on import or set low_memory=False.
  df = pd.read_csv('C:\\Users\\LabUser\\Project_3\\Library_Data\\pupld11b.csv', encoding='Windows-1252')

In [4]:
#saving csv file with encoding utf-8
df.to_csv('C:\\Users\\LabUser\\Project_3\\Library_Data\\pupld11b_utf8.csv', encoding='utf-8', index=False)

In [5]:
with open('C:\\Users\\LabUser\\Project_3\\Library_Data\\pupld11b.csv', 'rb') as file:
    result = chardet.detect(file.read())
    print(result)

{'encoding': 'Windows-1252', 'confidence': 0.73, 'language': ''}

In [6]:
with open('C:\\Users\\LabUser\\Project_3\\Library_Data\\pupld08a.csv', 'rb') as file:
    result = chardet.detect(file.read())
    print(result)

{'encoding': 'ISO-8859-1', 'confidence': 0.73, 'language': ''}
```

Data Extraction: Institute of Museum and Library Services

Data Transformation

Overview of Process



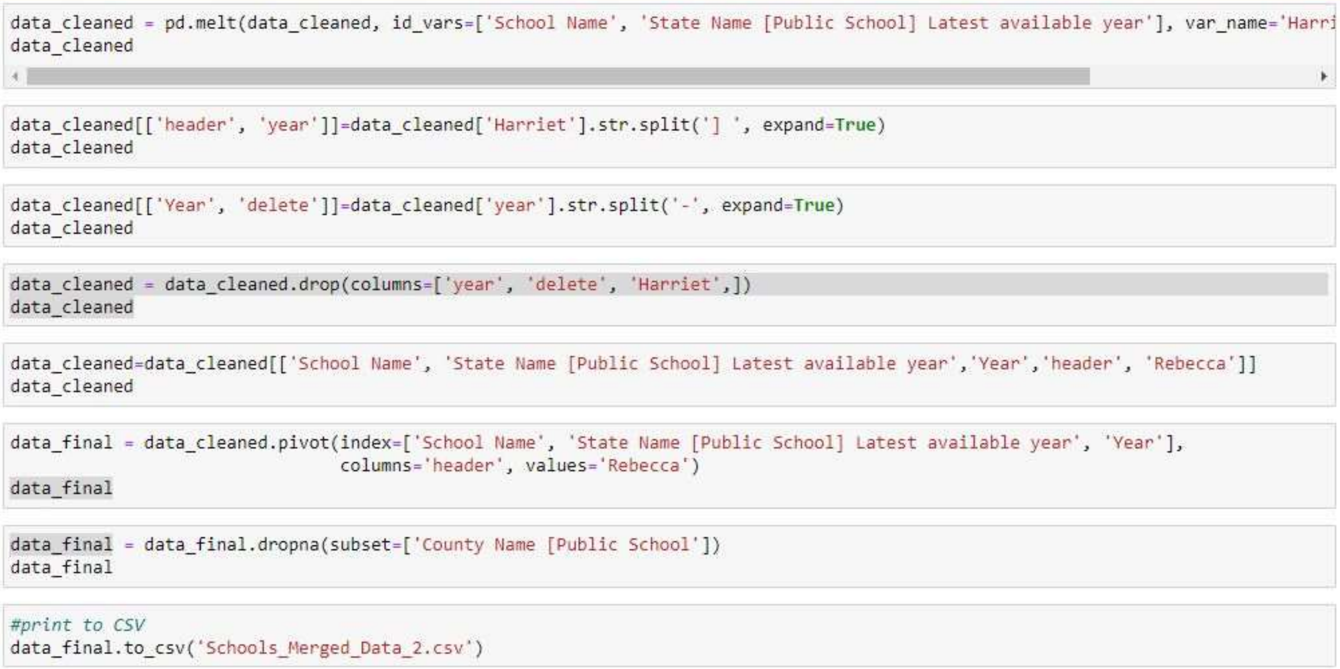
```
state_county_merge = pd.merge(code_sheet, state_merge, on = 'County Code (FIPS)', how = 'inner')
state_county_merge.head()
```

	Summary Level_x	State Code (FIPS)_x	County Code (FIPS)	County Subdivision Code (FIPS)_x	Place Code (FIPS)_x	Consolidtated City Code (FIPS)_x	Area Name (including legal/statistical area description)_x	Summary Level_y	State Code (FIPS)_y	County Code (FIPS)_x	...	year	county_population	murder	rape	robbery
0	50	1	1	0	0	0	Autauga County	40	1	0	...	2002	44959	0	18	35
1	50	1	1	0	0	0	Autauga County	40	1	0	...	2003	46808	3	29	34
2	50	1	1	0	0	0	Autauga County	40	1	0	...	2004	47929	0	11	38
3	50	1	1	0	0	0	Autauga County	40	1	0	...	2005	48917	0	15	28
4	50	1	1	0	0	0	Autauga County	40	1	0	...	2006	50316	1	16	49

5 rows × 24 columns

Data Transformation: National Neighborhood Data Archive (NaNDA): Crimes by County, United States, 2002 - 2014

Data Transformation: National Center for Education Statistics



```
data_cleaned = pd.melt(data_cleaned, id_vars=['School Name', 'State Name [Public School] Latest available year'], var_name='Harri
data_cleaned
```

```
data_cleaned[['header', 'year']]=data_cleaned['Harriet'].str.split('] ', expand=True)
data_cleaned
```

```
data_cleaned[['Year', 'delete']]=data_cleaned['year'].str.split('-', expand=True)
data_cleaned
```

```
data_cleaned = data_cleaned.drop(columns=['year', 'delete', 'Harriet',])
data_cleaned
```

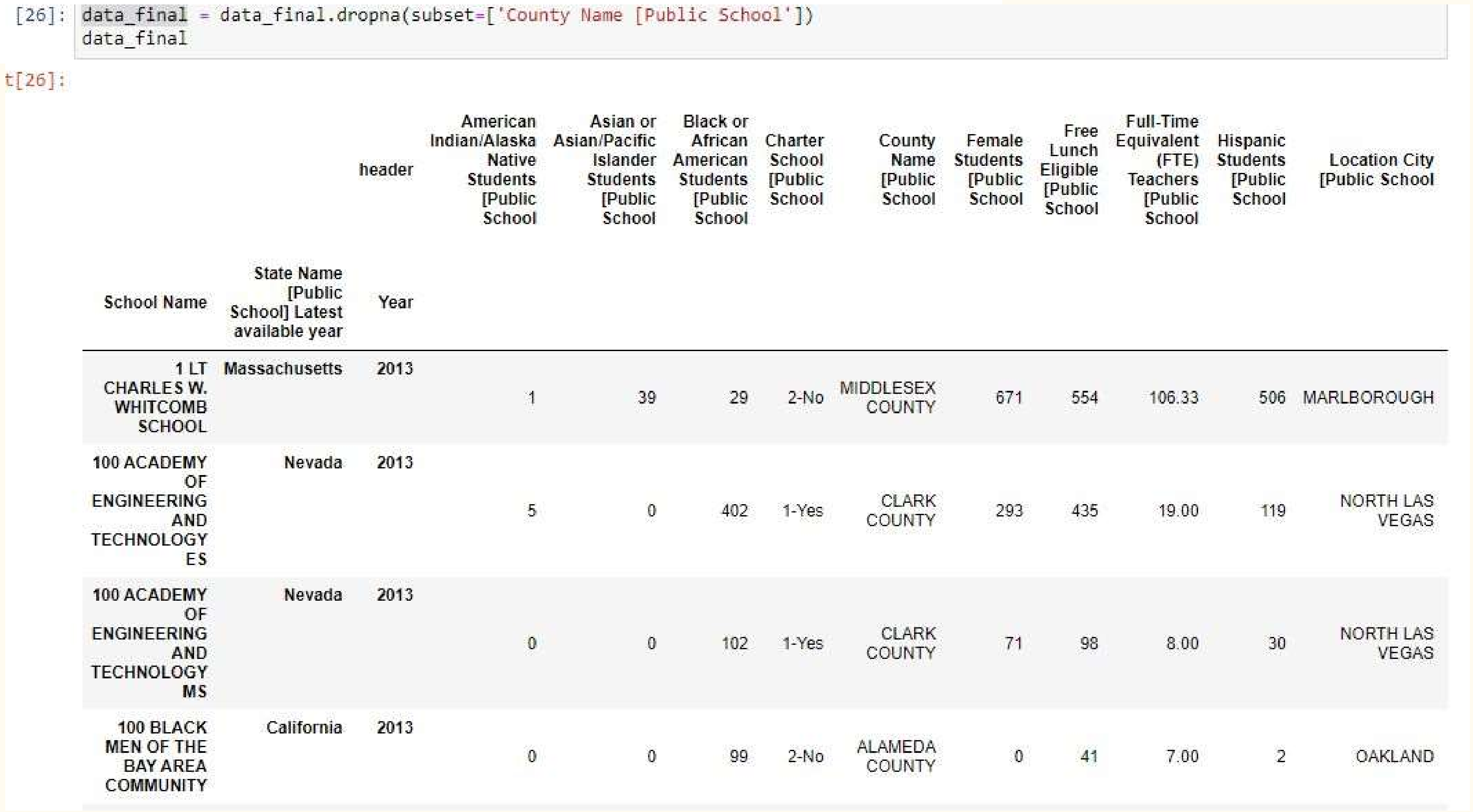
```
data_cleaned=data_cleaned[['School Name', 'State Name [Public School] Latest available year','Year','header', 'Rebecca']]
data_cleaned
```

```
data_final = data_cleaned.pivot(index=['School Name', 'State Name [Public School] Latest available year', 'Year'],
                                columns='header', values='Rebecca')
data_final
```

```
data_final = data_final.dropna(subset=['County Name [Public School']])
data_final
```

```
#print to CSV
data_final.to_csv('Schools_Merged_Data_2.csv')
```

Data Transformation: National Center for Education Statistics

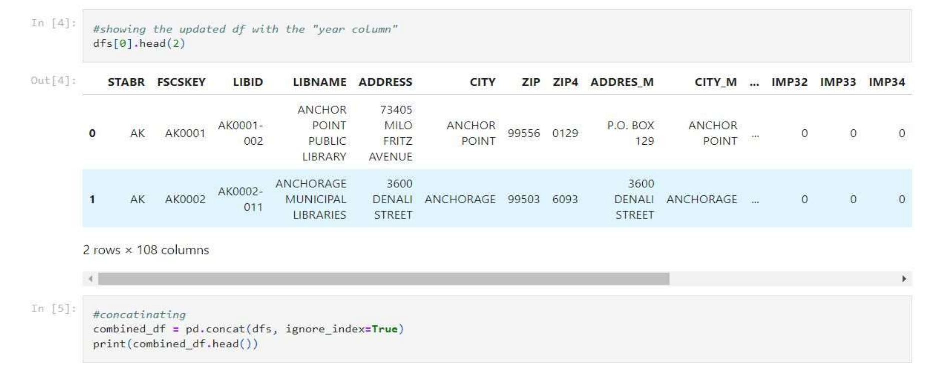


```
[26]: data_final = data_final.dropna(subset=['County Name [Public School'])
      data_final
```

t[26]:

		header	American Indian/Alaska Native Students [Public School	Asian or Asian/Pacific Islander Students [Public School	Black or African American Students [Public School	Charter School [Public School	County Name [Public School	Female Students [Public School	Free Lunch Eligible [Public School	Full-Time Equivalent (FTE) Teachers [Public School	Hispanic Students [Public School	Location City [Public School
School Name	State Name [Public School] Latest available year	Year										
1 LT CHARLES W. WHITCOMB SCHOOL	Massachusetts	2013	1	39	29	2-No	MIDDLESEX COUNTY	671	554	106.33	506	MARLBOROUGH
100 ACADEMY OF ENGINEERING AND TECHNOLOGY ES	Nevada	2013	5	0	402	1-Yes	CLARK COUNTY	293	435	19.00	119	NORTH LAS VEGAS
100 ACADEMY OF ENGINEERING AND TECHNOLOGY MS	Nevada	2013	0	0	102	1-Yes	CLARK COUNTY	71	98	8.00	30	NORTH LAS VEGAS
100 BLACK MEN OF THE BAY AREA COMMUNITY	California	2013	0	0	99	2-No	ALAMEDA COUNTY	0	41	7.00	2	OAKLAND

Overview of Process



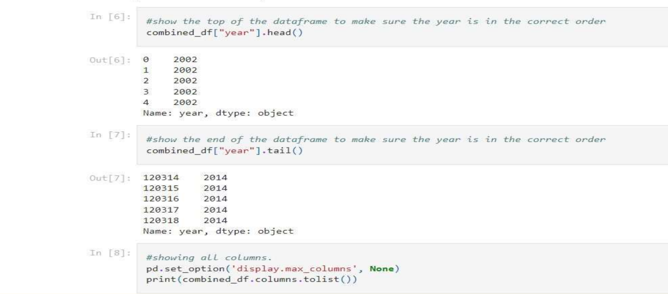
```
In [4]:  #showing the updated df with the "year column"
         dfs[0].head(2)
```

Out[4]:

	STABR	FSCSKEY	LIBID	LIBNAME	ADDRESS	CITY	ZIP	ZIP4	ADDRES_M	CITY_M	...	IMP32	IMP33	IMP34
0	AK	AK0001	AK0001-002	ANCHOR POINT PUBLIC LIBRARY	73405 MILO FRITZ AVENUE	ANCHOR POINT	99556	0129	P.O. BOX 129	ANCHOR POINT	...	0	0	0
1	AK	AK0002	AK0002-011	ANCHORAGE MUNICIPAL LIBRARIES	3600 DENALI STREET	ANCHORAGE	99503	6093	3600 DENALI STREET	ANCHORAGE	...	0	0	0

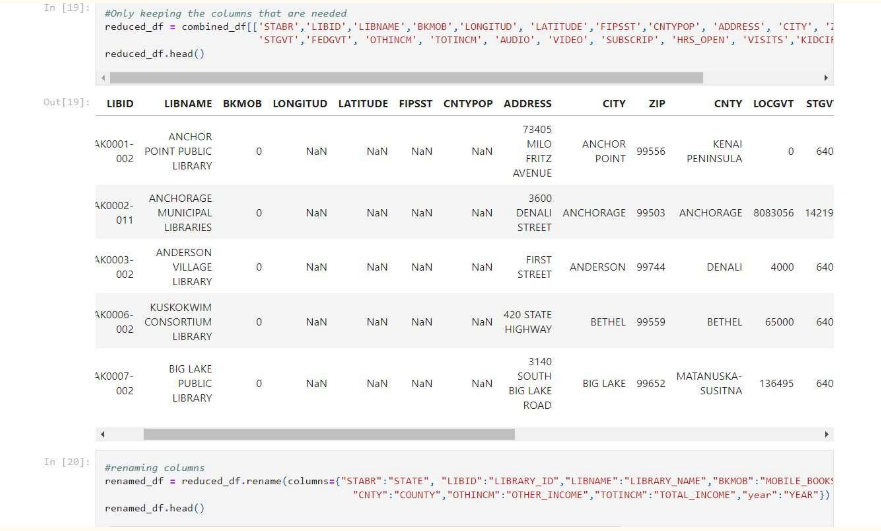
2 rows × 108 columns

```
In [5]:  #concatinating
         combined_df = pd.concat(dfs, ignore_index=True)
         print(combined_df.head())
```



```
In [6]:  #show the top of the dataframe to make sure the year is in the correct order
         combined_df["year"].head()

Out[6]:  0    2002
         1    2002
         2    2002
         3    2002
         4    2002
         Name: year, dtype: object

In [7]:  #show the end of the dataframe to make sure the year is in the correct order
         combined_df["year"].tail()

Out[7]:  120314    2014
         120315    2014
         120316    2014
         120317    2014
         120318    2014
         Name: year, dtype: object

In [8]:  #showing all columns.
         pd.set_option('display.max_columns', None)
         print(combined_df.columns.tolist())
```

Data Transformation: Institute of Museum and Library Services

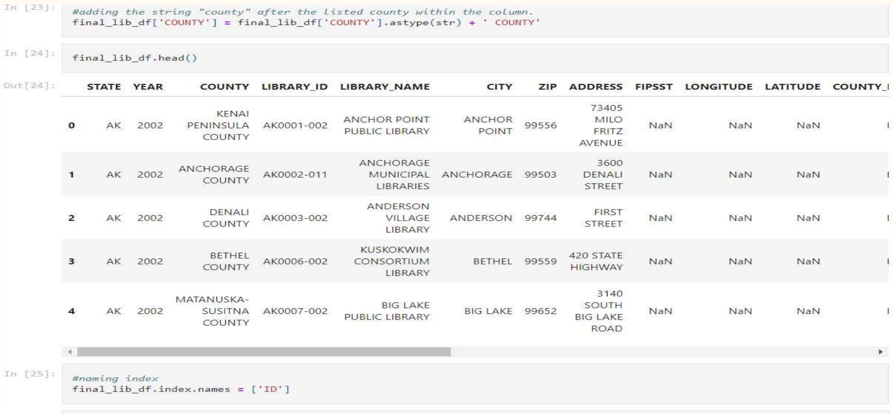


```
In [19]: #Only keeping the columns that are needed
         reduced_df = combined_df[['STABR','LIBID','LIBNAME','BKMOB','LONGITUD', 'LATITUDE','FIPSST','CNTYPOP', 'ADDRESS', 'CITY', 'Z
                                   'STGVT','FEDGVT', 'OTHINCM', 'TOTINCM', 'AUDIO', 'VIDEO', 'SUBSCRIP', 'HRS_OPEN', 'VISITS','KIDCIF
         reduced_df.head()
```

Out[19]:

LIBID	LIBNAME	BKMOB	LONGITUD	LATITUDE	FIPSST	CNTYPOP	ADDRESS	CITY	ZIP	CNTY	LOCGVT	STGV
AK0001-002	ANCHOR POINT PUBLIC LIBRARY	0	NaN	NaN	NaN	NaN	73405 MILO FRITZ AVENUE	ANCHOR POINT	99556	KENAI PENINSULA	0	640
AK0002-011	ANCHORAGE MUNICIPAL LIBRARIES	0	NaN	NaN	NaN	NaN	3600 DENALI STREET	ANCHORAGE	99503	ANCHORAGE	8083056	14219
AK0003-002	ANDERSON VILLAGE LIBRARY	0	NaN	NaN	NaN	NaN	FIRST STREET	ANDERSON	99744	DENALI	4000	640
AK0006-002	KUSKOKWIM CONSORTIUM LIBRARY	0	NaN	NaN	NaN	NaN	420 STATE HIGHWAY	BETHEL	99559	BETHEL	65000	640
AK0007-002	BIG LAKE PUBLIC LIBRARY	0	NaN	NaN	NaN	NaN	3140 SOUTH BIG LAKE ROAD	BIG LAKE	99652	MATANUSKA-SUSITNA	136495	640

```
In [20]: #renaming columns
         renamed_df = reduced_df.rename(columns={"STABR":"STATE", "LIBID":"LIBRARY_ID","LIBNAME":"LIBRARY_NAME","BKMOB":"MOBILE_BOOKS
                                                 "CNTY":"COUNTY","OTHINCM":"OTHER_INCOME","TOTINCM":"TOTAL_INCOME","year":"YEAR"})
         renamed_df.head()
```

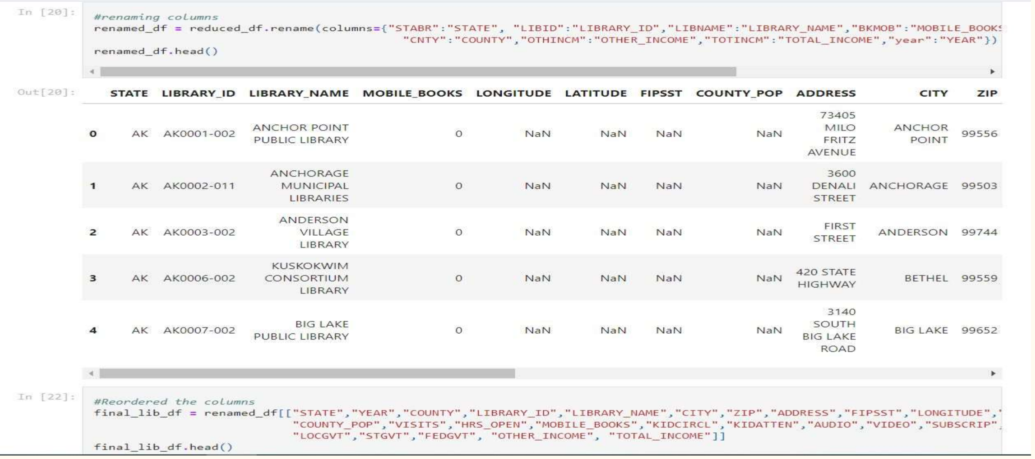


```
In [23]: #adding the string "county" after the listed county within the column.
         final_lib_df['COUNTY'] = final_lib_df['COUNTY'].astype(str) + ' COUNTY'

In [24]: final_lib_df.head()
```

Out[24]:

	STATE	YEAR	COUNTY	LIBRARY_ID	LIBRARY_NAME	CITY	ZIP	ADDRESS	FIPSST	LONGITUDE	LATITUDE	COUNTY_I
0	AK	2002	KENAI PENINSULA COUNTY	AK0001-002	ANCHOR POINT PUBLIC LIBRARY	ANCHOR POINT	99556	73405 MILO FRITZ AVENUE	NaN	NaN	NaN	I
1	AK	2002	ANCHORAGE COUNTY	AK0002-011	ANCHORAGE MUNICIPAL LIBRARIES	ANCHORAGE	99503	3600 DENALI STREET	NaN	NaN	NaN	I
2	AK	2002	DENALI COUNTY	AK0003-002	ANDERSON VILLAGE LIBRARY	ANDERSON	99744	FIRST STREET	NaN	NaN	NaN	I
3	AK	2002	BETHEL COUNTY	AK0006-002	KUSKOKWIM CONSORTIUM LIBRARY	BETHEL	99559	420 STATE HIGHWAY	NaN	NaN	NaN	I
4	AK	2002	MATANUSKA-SUSITNA COUNTY	AK0007-002	BIG LAKE PUBLIC LIBRARY	BIG LAKE	99652	3140 SOUTH BIG LAKE ROAD	NaN	NaN	NaN	I

```
In [25]: #naming index
         final_lib_df.index.names = ['ID']
```

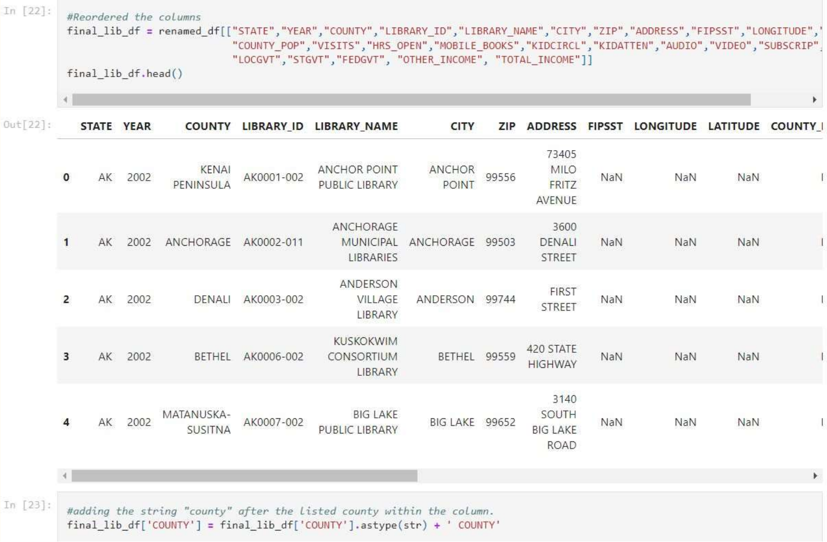


```
In [20]: #renaming columns
         renamed_df = reduced_df.rename(columns={"STABR":"STATE", "LIBID":"LIBRARY_ID","LIBNAME":"LIBRARY_NAME","BKMOB":"MOBILE_BOOKS
                                                 "CNTY":"COUNTY","OTHINCM":"OTHER_INCOME","TOTINCM":"TOTAL_INCOME","year":"YEAR"})
         renamed_df.head()
```

Out[20]:

	STATE	LIBRARY_ID	LIBRARY_NAME	MOBILE_BOOKS	LONGITUDE	LATITUDE	FIPSST	COUNTY_POP	ADDRESS	CITY	ZIP
0	AK	AK0001-002	ANCHOR POINT PUBLIC LIBRARY	0	NaN	NaN	NaN	NaN	73405 MILO FRITZ AVENUE	ANCHOR POINT	99556
1	AK	AK0002-011	ANCHORAGE MUNICIPAL LIBRARIES	0	NaN	NaN	NaN	NaN	3600 DENALI STREET	ANCHORAGE	99503
2	AK	AK0003-002	ANDERSON VILLAGE LIBRARY	0	NaN	NaN	NaN	NaN	FIRST STREET	ANDERSON	99744
3	AK	AK0006-002	KUSKOKWIM CONSORTIUM LIBRARY	0	NaN	NaN	NaN	NaN	420 STATE HIGHWAY	BETHEL	99559
4	AK	AK0007-002	BIG LAKE PUBLIC LIBRARY	0	NaN	NaN	NaN	NaN	3140 SOUTH BIG LAKE ROAD	BIG LAKE	99652

```
In [22]: #Reordered the columns
         final_lib_df = renamed_df[["STATE","YEAR","COUNTY","LIBRARY_ID","LIBRARY_NAME","CITY","ZIP","ADDRESS","FIPSST","LONGITUDE","
                                    "COUNTY_POP","VISITS","HRS_OPEN","MOBILE_BOOKS","KIDCIRCL","KIDATTEN","AUDIO","VIDEO","SUBSCRIP",
                                    "LOCGVT","STGVT","FEDGVT", "OTHER_INCOME", "TOTAL_INCOME"]]
         final_lib_df.head()
```

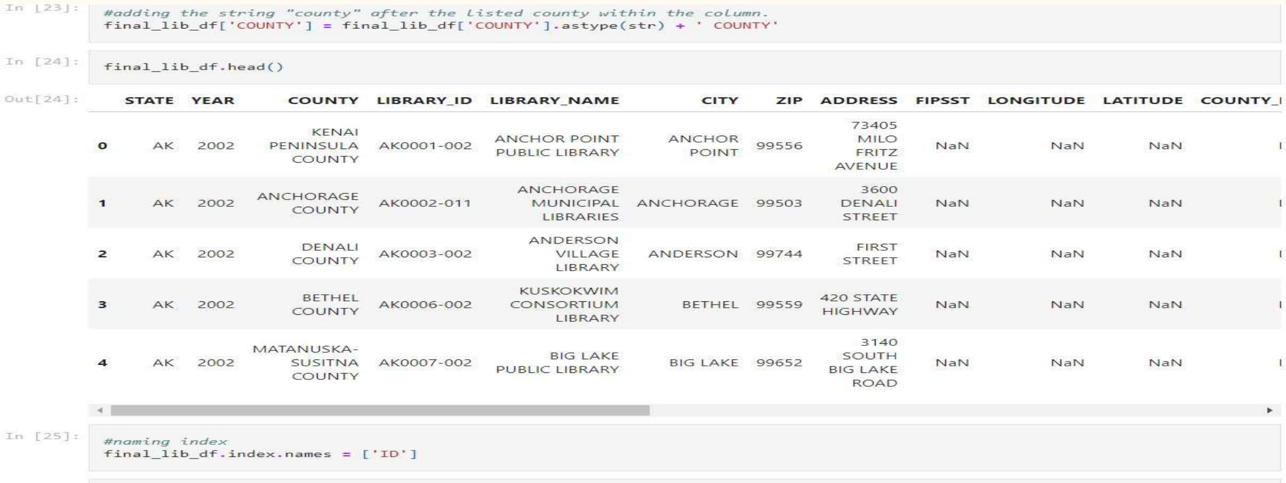


```
In [22]: #Reordered the columns
         final_lib_df = renamed_df[["STATE","YEAR","COUNTY","LIBRARY_ID","LIBRARY_NAME","CITY","ZIP","ADDRESS","FIPSST","LONGITUDE","
                                    "COUNTY_POP","VISITS","HRS_OPEN","MOBILE_BOOKS","KIDCIRCL","KIDATTEN","AUDIO","VIDEO","SUBSCRIP",
                                    "LOCGVT","STGVT","FEDGVT", "OTHER_INCOME", "TOTAL_INCOME"]]
         final_lib_df.head()
```

Out[22]:

	STATE	YEAR	COUNTY	LIBRARY_ID	LIBRARY_NAME	CITY	ZIP	ADDRESS	FIPSST	LONGITUDE	LATITUDE	COUNTY_I
0	AK	2002	KENAI PENINSULA	AK0001-002	ANCHOR POINT PUBLIC LIBRARY	ANCHOR POINT	99556	73405 MILO FRITZ AVENUE	NaN	NaN	NaN	I
1	AK	2002	ANCHORAGE	AK0002-011	ANCHORAGE MUNICIPAL LIBRARIES	ANCHORAGE	99503	3600 DENALI STREET	NaN	NaN	NaN	I
2	AK	2002	DENALI	AK0003-002	ANDERSON VILLAGE LIBRARY	ANDERSON	99744	FIRST STREET	NaN	NaN	NaN	I
3	AK	2002	BETHEL	AK0006-002	KUSKOKWIM CONSORTIUM LIBRARY	BETHEL	99559	420 STATE HIGHWAY	NaN	NaN	NaN	I
4	AK	2002	MATANUSKA-SUSITNA	AK0007-002	BIG LAKE PUBLIC LIBRARY	BIG LAKE	99652	3140 SOUTH BIG LAKE ROAD	NaN	NaN	NaN	I

```
In [23]: #adding the string "county" after the listed county within the column.
         final_lib_df['COUNTY'] = final_lib_df['COUNTY'].astype(str) + ' COUNTY'
```



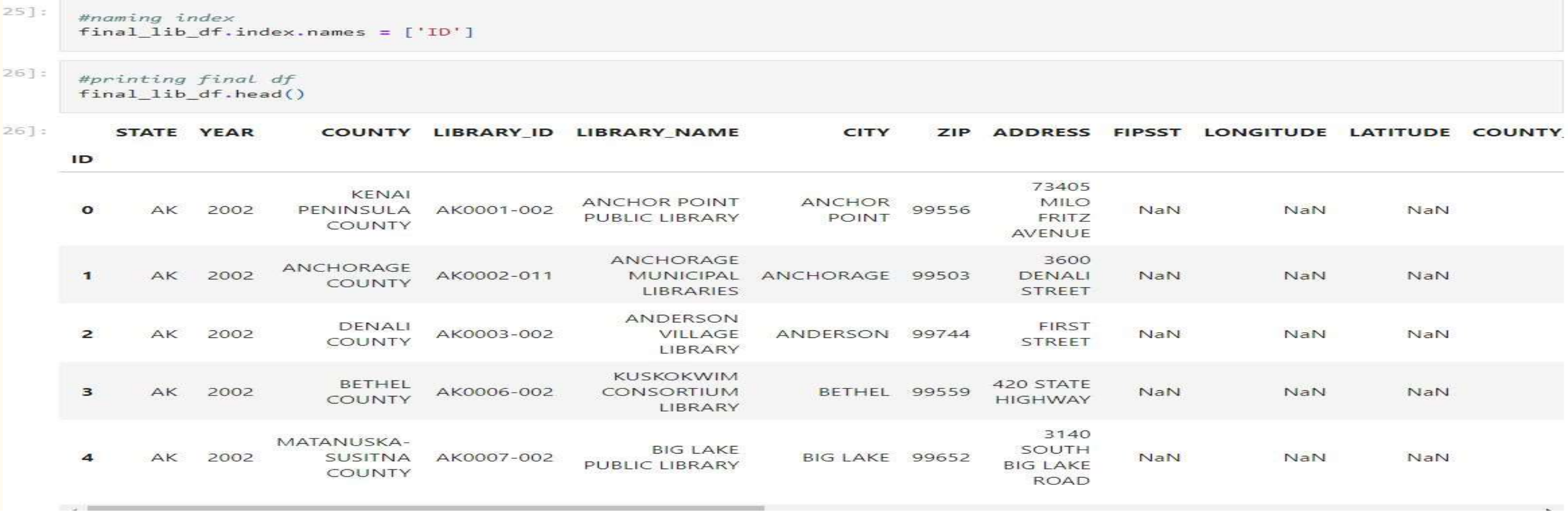
In [23]:
```
#adding the string "county" after the listed county within the column.
final_lib_df['COUNTY'] = final_lib_df['COUNTY'].astype(str) + ' COUNTY'
```

In [24]:
```
final_lib_df.head()
```

Out[24]:

	STATE	YEAR	COUNTY	LIBRARY_ID	LIBRARY_NAME	CITY	ZIP	ADDRESS	FIPSST	LONGITUDE	LATITUDE	COUNTY_I
0	AK	2002	KENAI PENINSULA COUNTY	AK0001-002	ANCHOR POINT PUBLIC LIBRARY	ANCHOR POINT	99556	73405 MILO FRITZ AVENUE	NaN	NaN	NaN	I
1	AK	2002	ANCHORAGE COUNTY	AK0002-011	ANCHORAGE MUNICIPAL LIBRARIES	ANCHORAGE	99503	3600 DENALI STREET	NaN	NaN	NaN	I
2	AK	2002	DENALI COUNTY	AK0003-002	ANDERSON VILLAGE LIBRARY	ANDERSON	99744	FIRST STREET	NaN	NaN	NaN	I
3	AK	2002	BETHEL COUNTY	AK0006-002	KUSKOKWIM CONSORTIUM LIBRARY	BETHEL	99559	420 STATE HIGHWAY	NaN	NaN	NaN	I
4	AK	2002	MATANUSKA-SUSITNA COUNTY	AK0007-002	BIG LAKE PUBLIC LIBRARY	BIG LAKE	99652	3140 SOUTH BIG LAKE ROAD	NaN	NaN	NaN	I

In [25]:
```
#naming index
final_lib_df.index.names = ['ID']
```

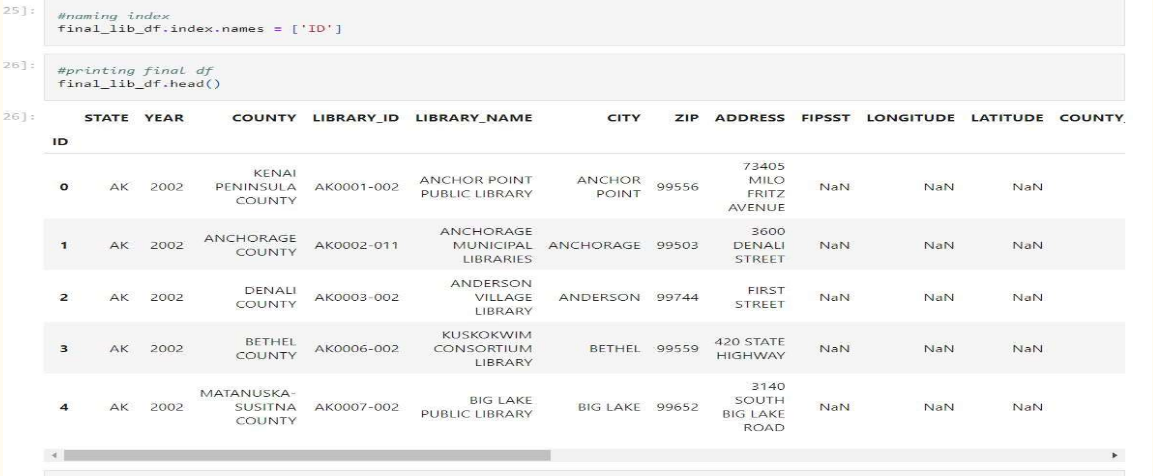


25]:
```
#naming index
final_lib_df.index.names = ['ID']
```

26]:
```
#printing final df
final_lib_df.head()
```

26]:

	STATE	YEAR	COUNTY	LIBRARY_ID	LIBRARY_NAME	CITY	ZIP	ADDRESS	FIPSST	LONGITUDE	LATITUDE	COUNTY.
ID												
0	AK	2002	KENAI PENINSULA COUNTY	AK0001-002	ANCHOR POINT PUBLIC LIBRARY	ANCHOR POINT	99556	73405 MILO FRITZ AVENUE	NaN	NaN	NaN	
1	AK	2002	ANCHORAGE COUNTY	AK0002-011	ANCHORAGE MUNICIPAL LIBRARIES	ANCHORAGE	99503	3600 DENALI STREET	NaN	NaN	NaN	
2	AK	2002	DENALI COUNTY	AK0003-002	ANDERSON VILLAGE LIBRARY	ANDERSON	99744	FIRST STREET	NaN	NaN	NaN	
3	AK	2002	BETHEL COUNTY	AK0006-002	KUSKOKWIM CONSORTIUM LIBRARY	BETHEL	99559	420 STATE HIGHWAY	NaN	NaN	NaN	
4	AK	2002	MATANUSKA-SUSITNA COUNTY	AK0007-002	BIG LAKE PUBLIC LIBRARY	BIG LAKE	99652	3140 SOUTH BIG LAKE ROAD	NaN	NaN	NaN	

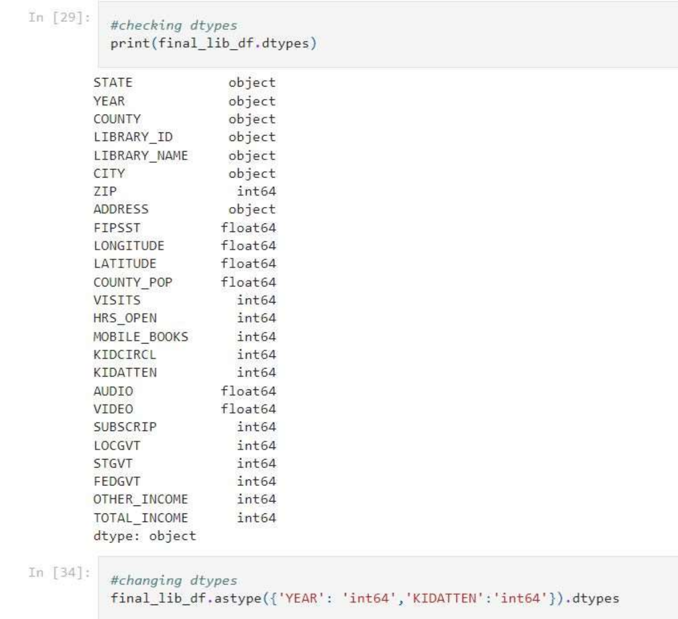


```
25]:  #naming index
      final_lib_df.index.names = ['ID']
```

```
26]:  #printing final df
      final_lib_df.head()
```

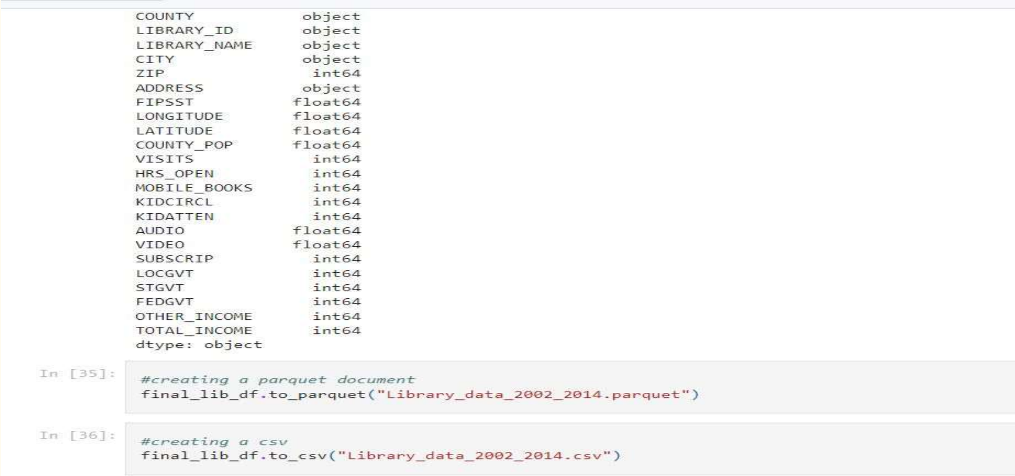
26]:

ID	STATE	YEAR	COUNTY	LIBRARY_ID	LIBRARY_NAME	CITY	ZIP	ADDRESS	FIPSST	LONGITUDE	LATITUDE	COUNTY_
0	AK	2002	KENAI PENINSULA COUNTY	AK0001-002	ANCHOR POINT PUBLIC LIBRARY	ANCHOR POINT	99556	73405 MILO FRITZ AVENUE	NaN	NaN	NaN	
1	AK	2002	ANCHORAGE COUNTY	AK0002-011	ANCHORAGE MUNICIPAL LIBRARIES	ANCHORAGE	99503	3600 DENALI STREET	NaN	NaN	NaN	
2	AK	2002	DENALI COUNTY	AK0003-002	ANDERSON VILLAGE LIBRARY	ANDERSON	99744	FIRST STREET	NaN	NaN	NaN	
3	AK	2002	BETHEL COUNTY	AK0006-002	KUSKOKWIM CONSORTIUM LIBRARY	BETHEL	99559	420 STATE HIGHWAY	NaN	NaN	NaN	
4	AK	2002	MATANUSKA-SUSITNA COUNTY	AK0007-002	BIG LAKE PUBLIC LIBRARY	BIG LAKE	99652	3140 SOUTH BIG LAKE ROAD	NaN	NaN	NaN	



```
In [29]:  #checking dtypes
          print(final_lib_df.dtypes)

STATE             object
YEAR              object
COUNTY            object
LIBRARY_ID        object
LIBRARY_NAME      object
CITY              object
ZIP                int64
ADDRESS           object
FIPSST           float64
LONGITUDE        float64
LATITUDE         float64
COUNTY_POP       float64
VISITS             int64
HRS_OPEN           int64
MOBILE_BOOKS       int64
KIDCIRCL           int64
KIDATTEN           int64
AUDIO            float64
VIDEO            float64
SUBSCRIP           int64
LOCGVT             int64
STGVT              int64
FEDGVT             int64
OTHER_INCOME       int64
TOTAL_INCOME       int64
dtype: object
```

```
In [34]:  #changing dtypes
          final_lib_df.astype({'YEAR': 'int64','KIDATTEN':'int64'}).dtypes
```



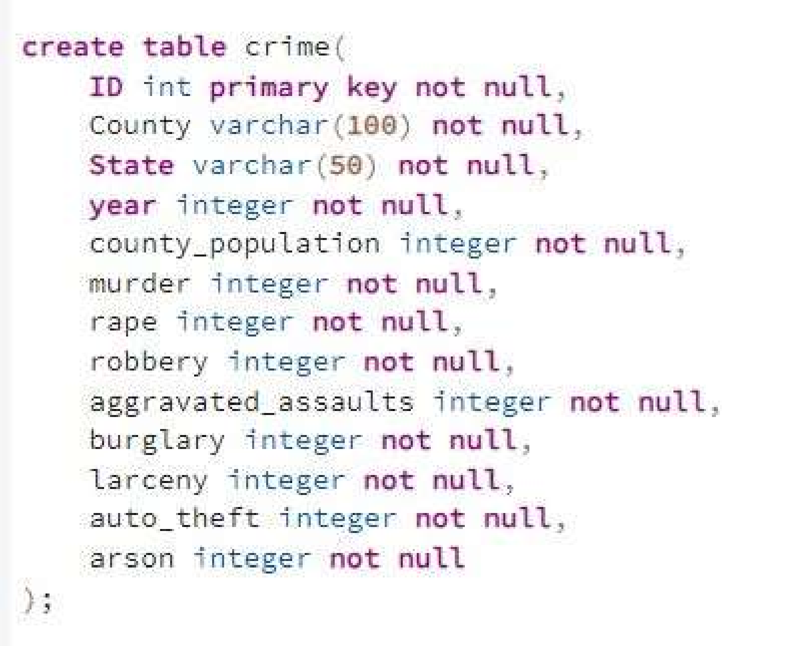
```
COUNTY            object
LIBRARY_ID        object
LIBRARY_NAME      object
CITY              object
ZIP                int64
ADDRESS           object
FIPSST           float64
LONGITUDE        float64
LATITUDE         float64
COUNTY_POP       float64
VISITS             int64
HRS_OPEN           int64
MOBILE_BOOKS       int64
KIDCIRCL           int64
KIDATTEN           int64
AUDIO            float64
VIDEO            float64
SUBSCRIP           int64
LOCGVT             int64
STGVT              int64
FEDGVT             int64
OTHER_INCOME       int64
TOTAL_INCOME       int64
dtype: object
```

```
In [35]:  #creating a parquet document
          final_lib_df.to_parquet("Library_data_2002_2014.parquet")
```

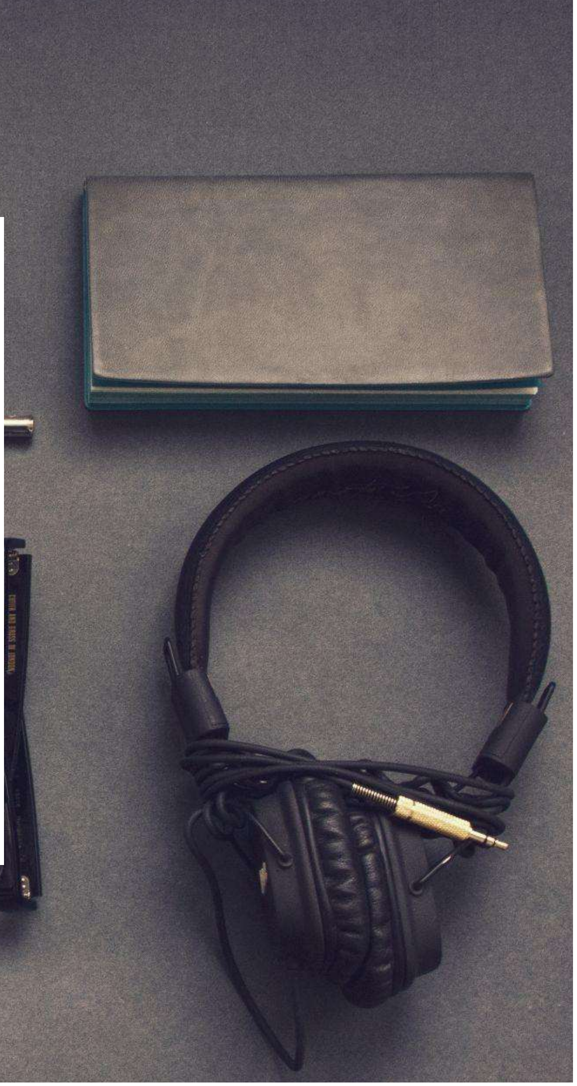
```
In [36]:  #creating a csv
          final_lib_df.to_csv("Library_data_2002_2014.csv")
```

Loading Data into PostgreSQL

Overview of Process

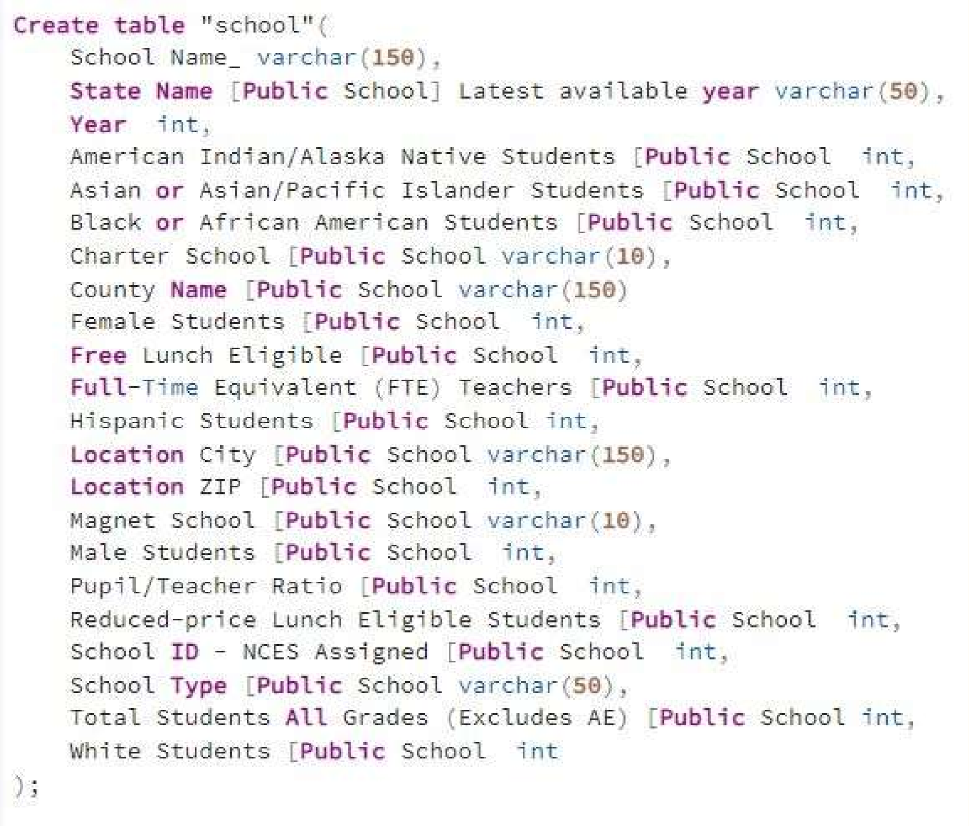


```
create table crime(
    ID int primary key not null,
    County varchar(100) not null,
    State varchar(50) not null,
    year integer not null,
    county_population integer not null,
    murder integer not null,
    rape integer not null,
    robbery integer not null,
    aggravated_assaults integer not null,
    burglary integer not null,
    larceny integer not null,
    auto_theft integer not null,
    arson integer not null
);
```



Postgres Table Schema Creation: National Neighborhood Data Archive (NaNDA): Crimes by County, United States, 2002 - 2014

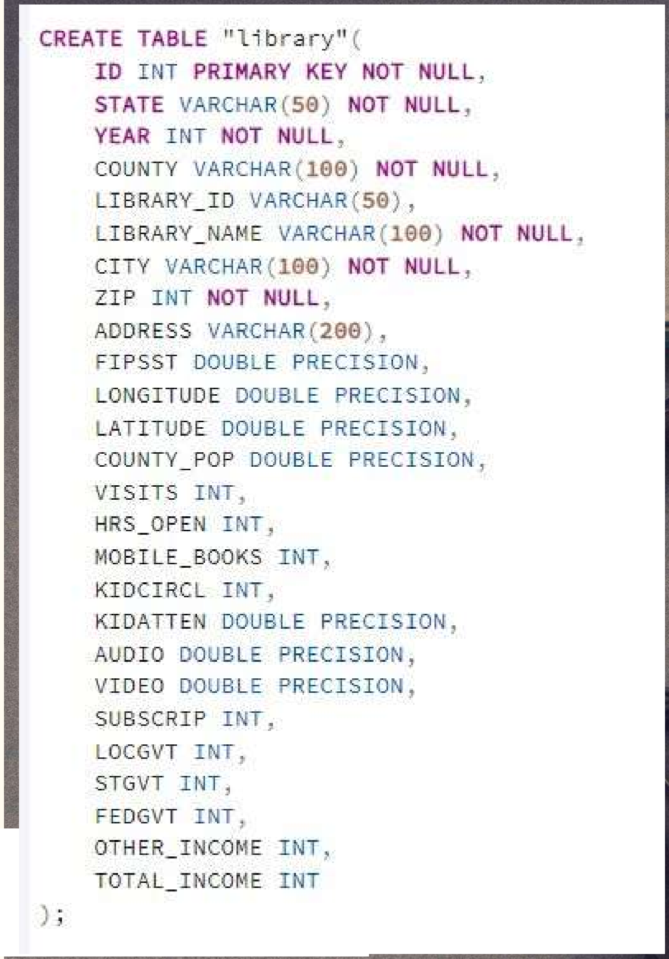
Overview of Process



```
Create table "school"(
    School Name_ varchar(150),
    State Name [Public School] Latest available year varchar(50),
    Year  int,
    American Indian/Alaska Native Students [Public School  int,
    Asian or Asian/Pacific Islander Students [Public School  int,
    Black or African American Students [Public School  int,
    Charter School [Public School varchar(10),
    County Name [Public School varchar(150)
    Female Students [Public School  int,
    Free Lunch Eligible [Public School  int,
    Full-Time Equivalent (FTE) Teachers [Public School  int,
    Hispanic Students [Public School int,
    Location City [Public School varchar(150),
    Location ZIP [Public School  int,
    Magnet School [Public School varchar(10),
    Male Students [Public School  int,
    Pupil/Teacher Ratio [Public School  int,
    Reduced-price Lunch Eligible Students [Public School  int,
    School ID - NCES Assigned [Public School  int,
    School Type [Public School varchar(50),
    Total Students All Grades (Excludes AE) [Public School int,
    White Students [Public School  int
);
```

Postgres Table Schema Creation: National Center for Education Statistics

Overview of Process

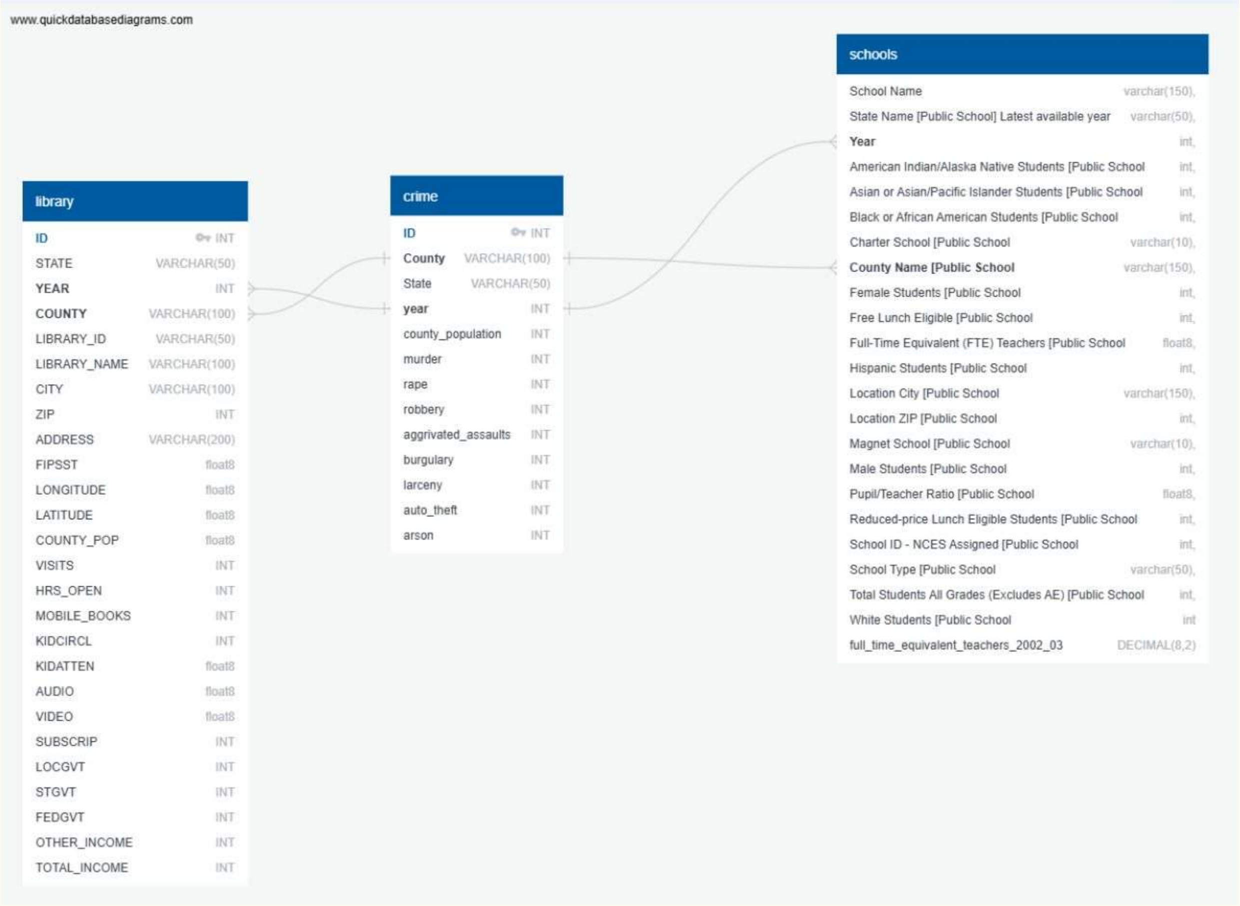


```
CREATE TABLE "library"(
    ID INT PRIMARY KEY NOT NULL,
    STATE VARCHAR(50) NOT NULL,
    YEAR INT NOT NULL,
    COUNTY VARCHAR(100) NOT NULL,
    LIBRARY_ID VARCHAR(50),
    LIBRARY_NAME VARCHAR(100) NOT NULL,
    CITY VARCHAR(100) NOT NULL,
    ZIP INT NOT NULL,
    ADDRESS VARCHAR(200),
    FIPSST DOUBLE PRECISION,
    LONGITUDE DOUBLE PRECISION,
    LATITUDE DOUBLE PRECISION,
    COUNTY_POP DOUBLE PRECISION,
    VISITS INT,
    HRS_OPEN INT,
    MOBILE_BOOKS INT,
    KIDCIRCL INT,
    KIDATTEN DOUBLE PRECISION,
    AUDIO DOUBLE PRECISION,
    VIDEO DOUBLE PRECISION,
    SUBSCRIP INT,
    LOCGVT INT,
    STGVT INT,
    FEDGVT INT,
    OTHER_INCOME INT,
    TOTAL_INCOME INT
);
```

Postgres Table Schema Creation: Institute of Museum and Library Services

Final Database Design

Entity Relationship Diagram



www.quickdatabasediagrams.com

library	
ID	INT
STATE	VARCHAR(50)
YEAR	INT
COUNTY	VARCHAR(100)
LIBRARY_ID	VARCHAR(50)
LIBRARY_NAME	VARCHAR(100)
CITY	VARCHAR(100)
ZIP	INT
ADDRESS	VARCHAR(200)
FIPSST	float8
LONGITUDE	float8
LATITUDE	float8
COUNTY_POP	float8
VISITS	INT
HRS_OPEN	INT
MOBILE_BOOKS	INT
KIDCIRCL	INT
KIDATTEN	float8
AUDIO	float8
VIDEO	float8
SUBSCRIP	INT
LOCGVT	INT
STGVT	INT
FEDGVT	INT
OTHER_INCOME	INT
TOTAL_INCOME	INT

crime	
ID	INT
County	VARCHAR(100)
State	VARCHAR(50)
year	INT
county_population	INT
murder	INT
rape	INT
robbery	INT
aggrivated_assaults	INT
burgulary	INT
larceny	INT
auto_theft	INT
arson	INT

schools	
School Name	varchar(150),
State Name [Public School] Latest available year	varchar(50),
Year	int,
American Indian/Alaska Native Students [Public School	int,
Asian or Asian/Pacific Islander Students [Public School	int,
Black or African American Students [Public School	int,
Charter School [Public School	varchar(10),
County Name [Public School	varchar(150),
Female Students [Public School	int,
Free Lunch Eligible [Public School	int,
Full-Time Equivalent (FTE) Teachers [Public School	float8,
Hispanic Students [Public School	int,
Location City [Public School	varchar(150),
Location ZIP [Public School	int,
Magnet School [Public School	varchar(10),
Male Students [Public School	int,
Pupil/Teacher Ratio [Public School	float8,
Reduced-price Lunch Eligible Students [Public School	int,
School ID - NCES Assigned [Public School	int,
School Type [Public School	varchar(50),
Total Students All Grades (Excludes AE) [Public School	int,
White Students [Public School	int
full_time_equivalent_teachers_2002_03	DECIMAL(8,2)

Key Takeaways