Project: National Park Database

Team Zombie Tigers: Sami Coalson, Renee White, Steven Zych, Melissa Schafer, Raj Rajaraman

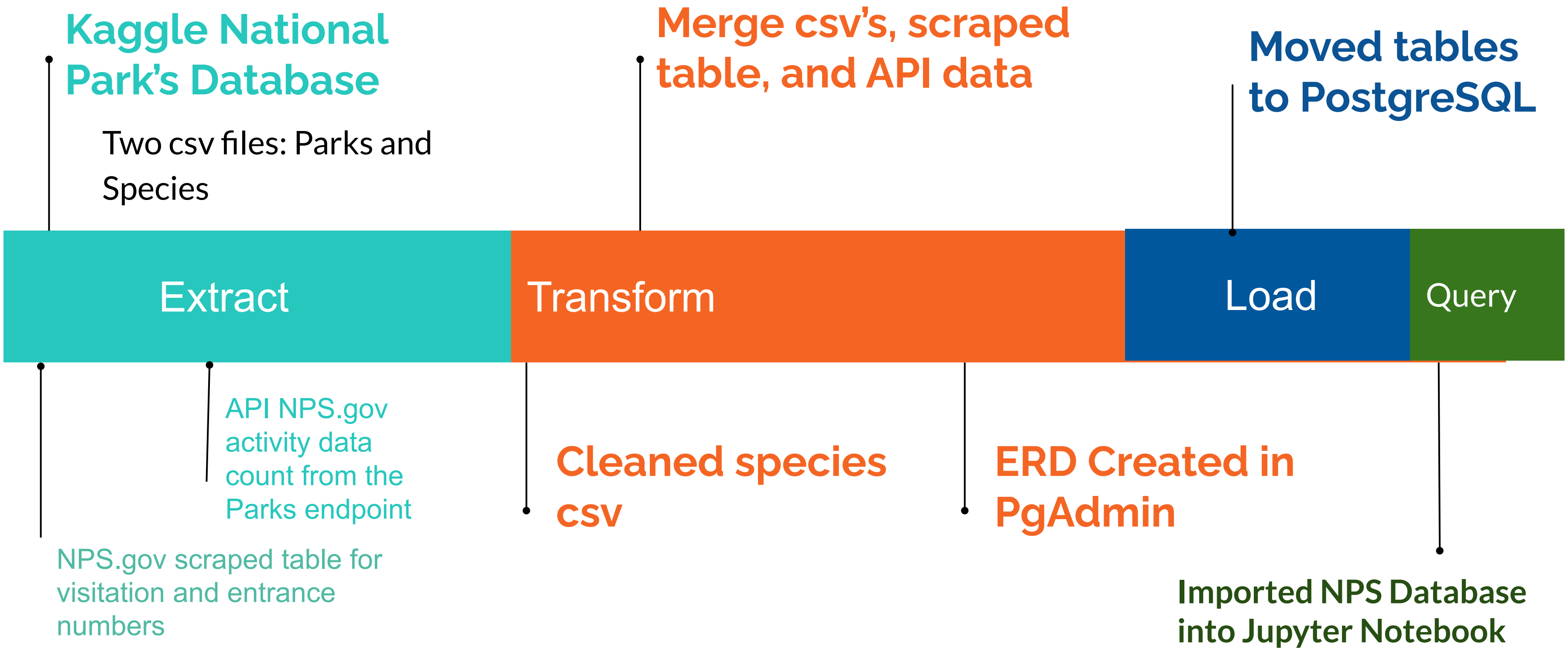
Background

Our group combined datasets to contain information on the biodiversity, visitation, activities, and acreage of the United States' National Park system.

Our data sources come from:

- https://www.kaggle.com/nationalparkservice/park-biodiversity
- https://www.nps.gov/aboutus/visitation-numbers.htm
- https://www.nps.gov/subjects/developer/api-documentation.htm

ETL Timeline: May 2021



Kaggle National Park's Database

Two csv files: Parks and Species

Merge csv's, scraped table, and API data

Moved tables to PostgreSQL

Extract

Transform

Load

Query

NPS.gov scraped table for visitation and entrance numbers

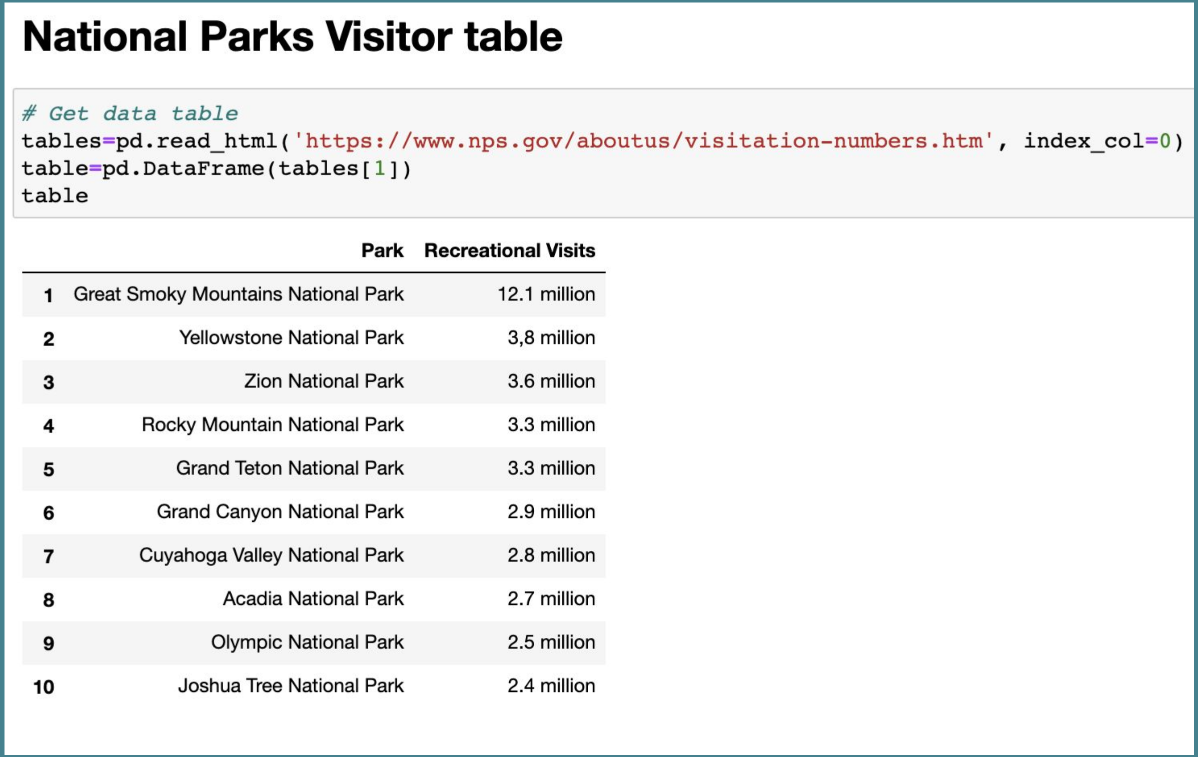
API NPS.gov activity data count from the Parks endpoint

Cleaned species csv

ERD Created in PgAdmin

Imported NPS Database into Jupyter Notebook

Extract: scrape method

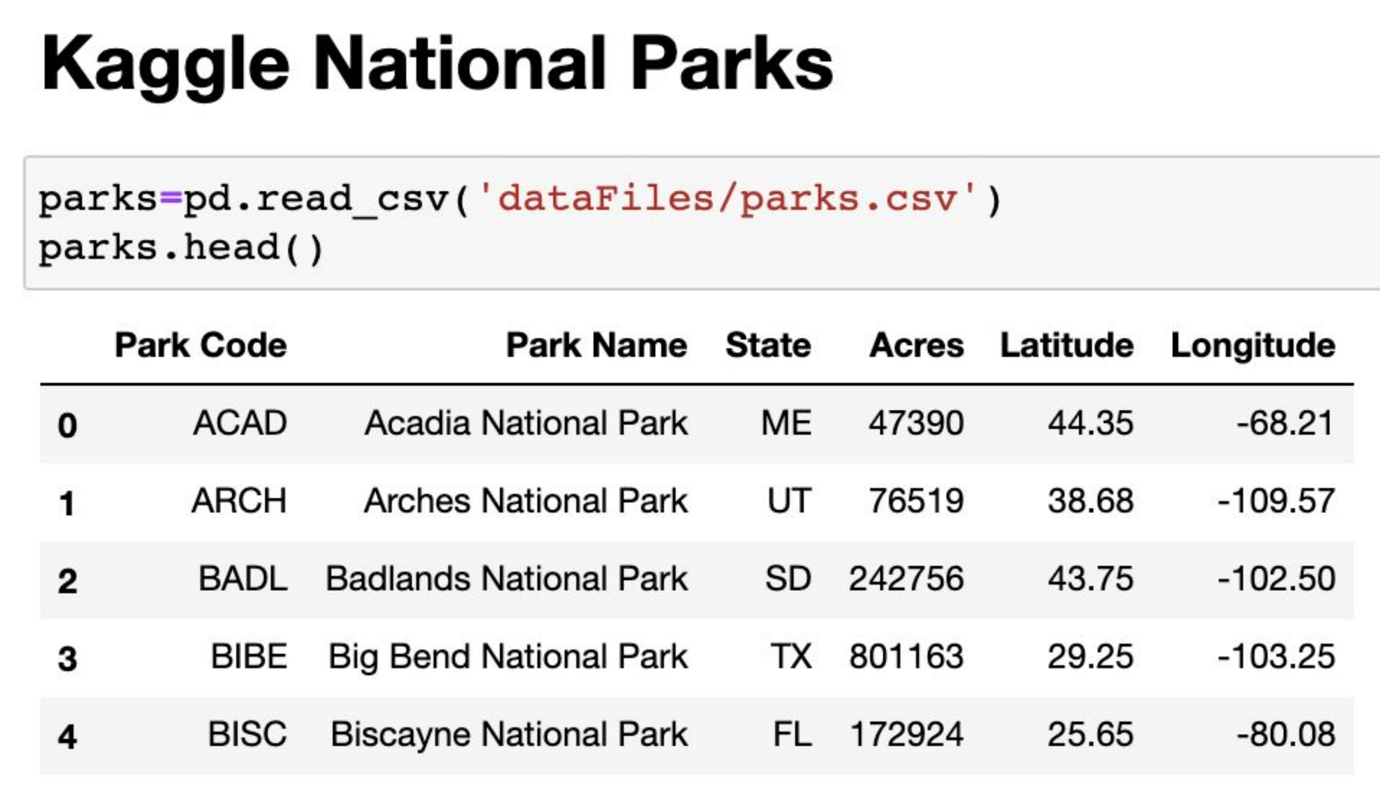


National Parks Visitor table

```
# Get data table
tables=pd.read_html('https://www.nps.gov/aboutus/visitation-numbers.htm', index_col=0)
table=pd.DataFrame(tables[1])
table
```

	Park	Recreational Visits
1	Great Smoky Mountains National Park	12.1 million
2	Yellowstone National Park	3,8 million
3	Zion National Park	3.6 million
4	Rocky Mountain National Park	3.3 million
5	Grand Teton National Park	3.3 million
6	Grand Canyon National Park	2.9 million
7	Cuyahoga Valley National Park	2.8 million
8	Acadia National Park	2.7 million
9	Olympic National Park	2.5 million
10	Joshua Tree National Park	2.4 million

Extract: csv pandas import

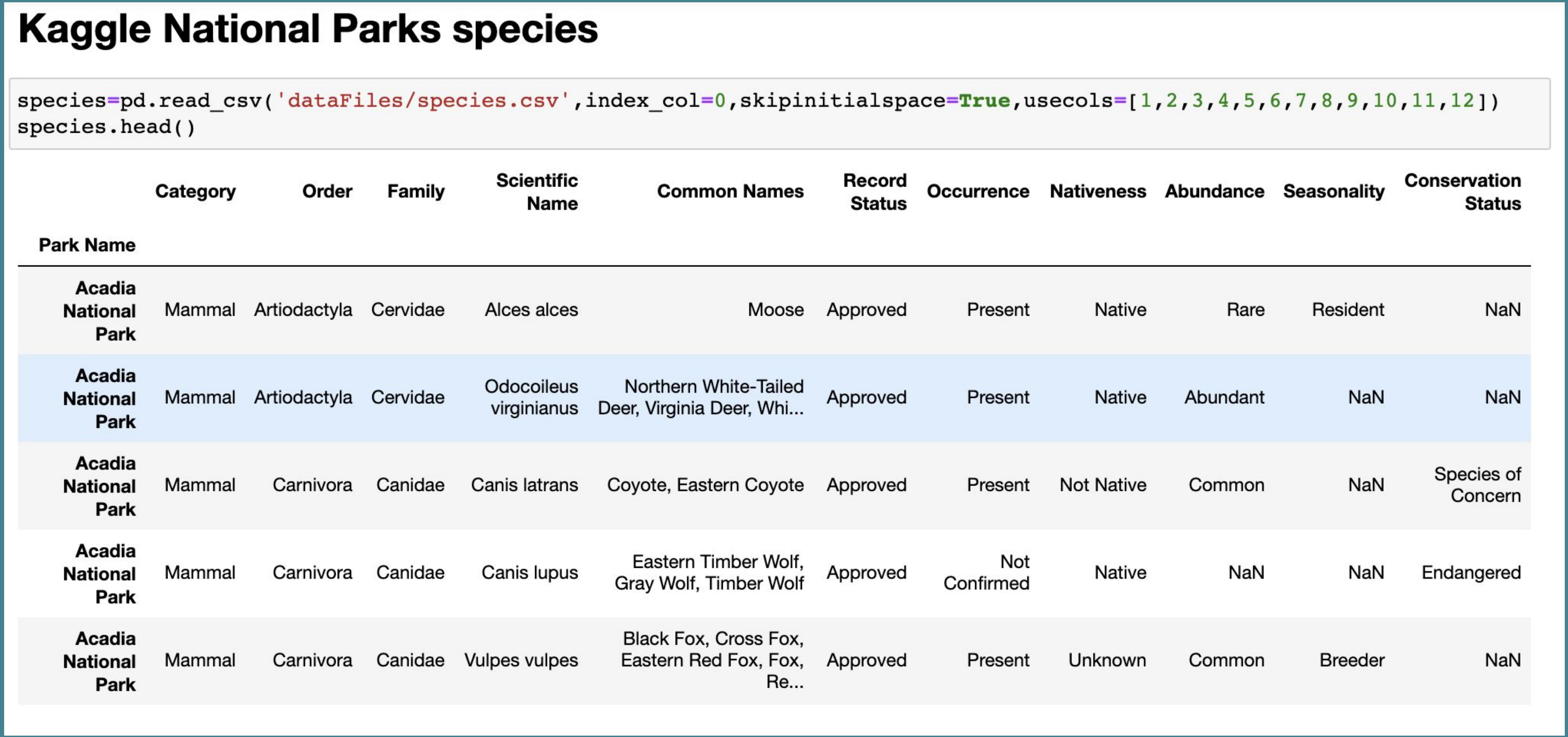


Kaggle National Parks

```
parks=pd.read_csv('dataFiles/parks.csv')
parks.head()
```

	Park Code	Park Name	State	Acres	Latitude	Longitude
0	ACAD	Acadia National Park	ME	47390	44.35	-68.21
1	ARCH	Arches National Park	UT	76519	38.68	-109.57
2	BADL	Badlands National Park	SD	242756	43.75	-102.50
3	BIBE	Big Bend National Park	TX	801163	29.25	-103.25
4	BISC	Biscayne National Park	FL	172924	25.65	-80.08

Extract: csv pandas import

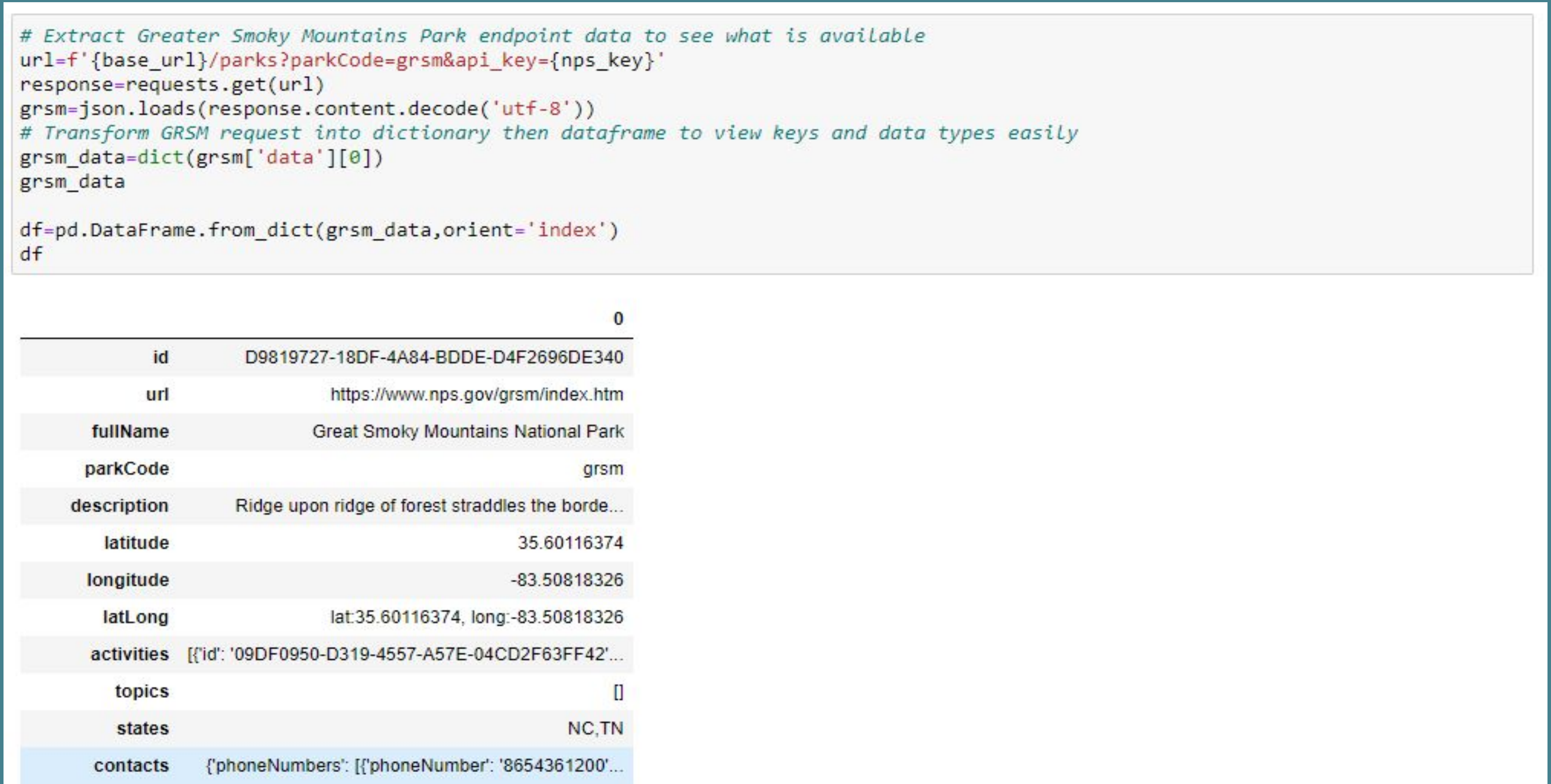


Kaggle National Parks species

```
species=pd.read_csv('dataFiles/species.csv',index_col=0,skipinitialspace=True,usecols=[1,2,3,4,5,6,7,8,9,10,11,12])
species.head()
```

	Category	Order	Family	Scientific Name	Common Names	Record Status	Occurrence	Nativeness	Abundance	Seasonality	Conservation Status
Park Name											
Acadia National Park	Mammal	Artiodactyla	Cervidae	Alces alces	Moose	Approved	Present	Native	Rare	Resident	NaN
Acadia National Park	Mammal	Artiodactyla	Cervidae	Odocoileus virginianus	Northern White-Tailed Deer, Virginia Deer, Whi...	Approved	Present	Native	Abundant	NaN	NaN
Acadia National Park	Mammal	Carnivora	Canidae	Canis latrans	Coyote, Eastern Coyote	Approved	Present	Not Native	Common	NaN	Species of Concern
Acadia National Park	Mammal	Carnivora	Canidae	Canis lupus	Eastern Timber Wolf, Gray Wolf, Timber Wolf	Approved	Not Confirmed	Native	NaN	NaN	Endangered
Acadia National Park	Mammal	Carnivora	Canidae	Vulpes vulpes	Black Fox, Cross Fox, Eastern Red Fox, Fox, Re...	Approved	Present	Unknown	Common	Breeder	NaN

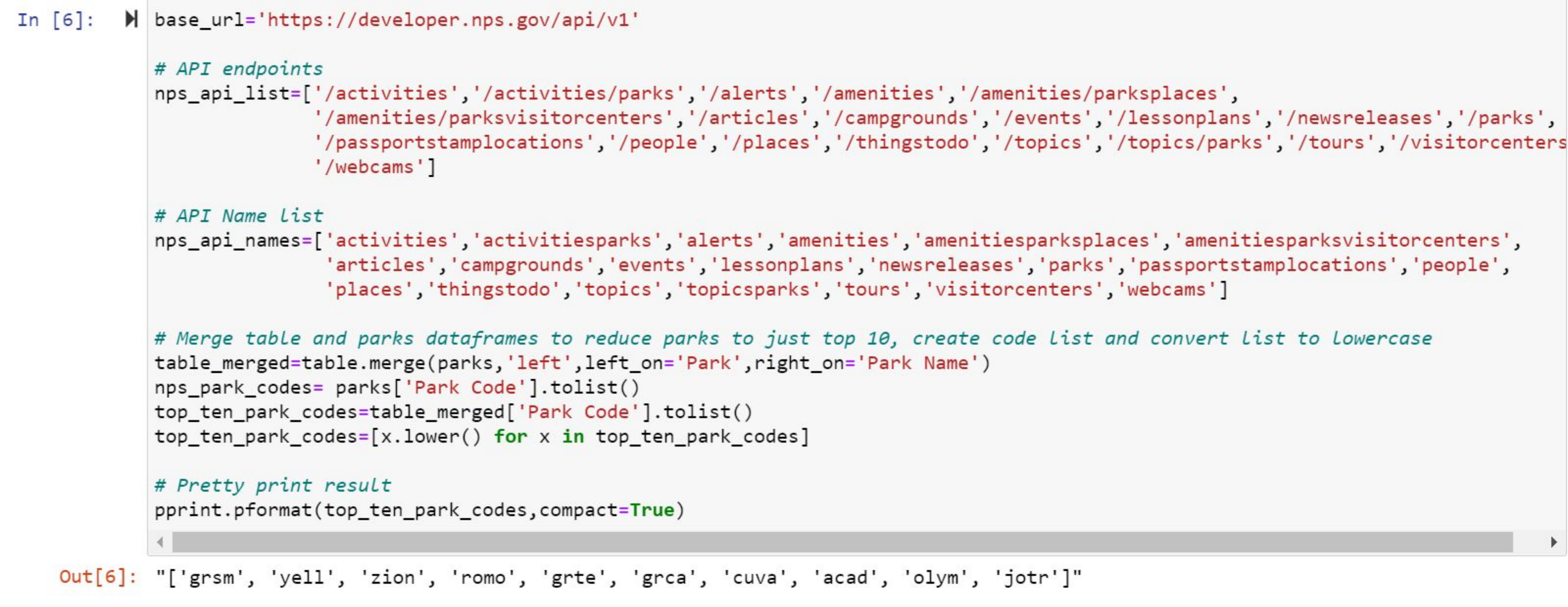
Extract/Transform: JSON to DF



```python
# Extract Greater Smoky Mountains Park endpoint data to see what is available
url=f'{base_url}/parks?parkCode=grsm&api_key={nps_key}'
response=requests.get(url)
grsm=json.loads(response.content.decode('utf-8'))
# Transform GRSM request into dictionary then dataframe to view keys and data types easily
grsm_data=dict(grsm['data'][0])
grsm_data

df=pd.DataFrame.from_dict(grsm_data,orient='index')
df
```

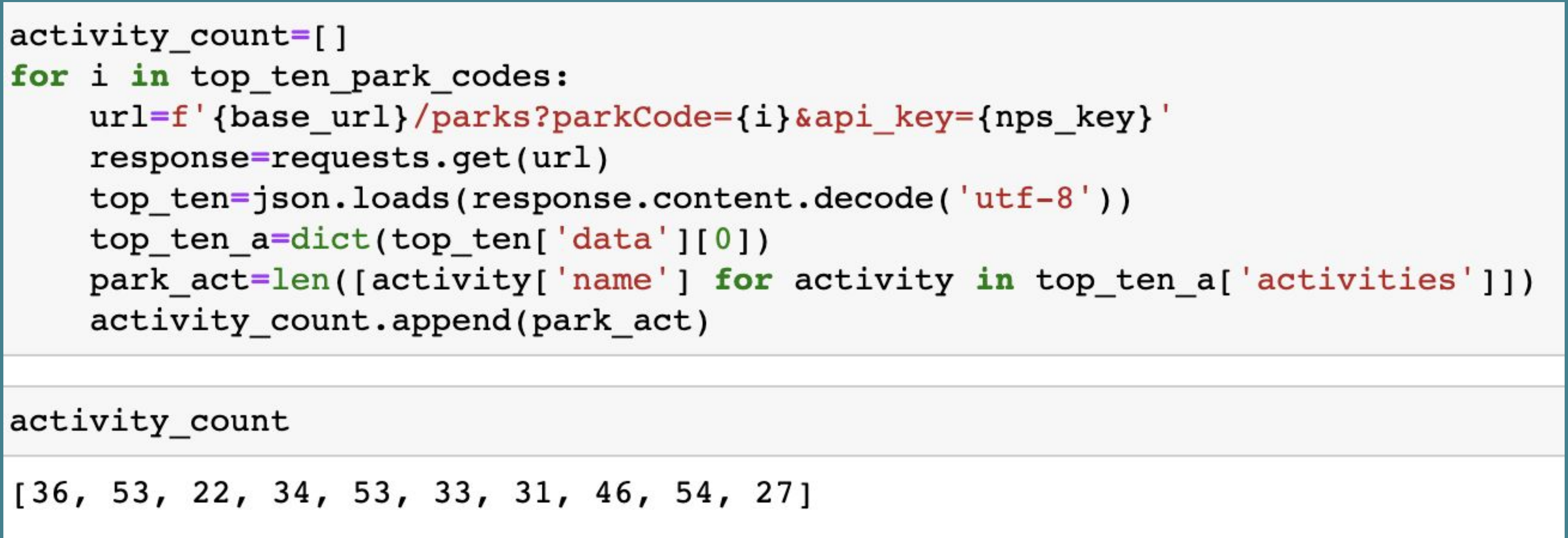
	0
id	D9819727-18DF-4A84-BDDE-D4F2696DE340
url	https://www.nps.gov/grsm/index.htm
fullName	Great Smoky Mountains National Park
parkCode	grsm
description	Ridge upon ridge of forest straddles the borde...
latitude	35.60116374
longitude	-83.50818326
latLong	lat:35.60116374, long:-83.50818326
activities	[{'id': '09DF0950-D319-4557-A57E-04CD2F63FF42'...
topics	[]
states	NC,TN
contacts	{'phoneNumbers': [{'phoneNumber': '8654361200'...

Transform: prep. for possible API Iterations



```
In [6]:  base_url='https://developer.nps.gov/api/v1'

         # API endpoints
         nps_api_list=['/activities','/activities/parks','/alerts','/amenities','/amenities/parksplaces',
                       '/amenities/parksvisitorcenters','/articles','/campgrounds','/events','/lessonplans','/newsreleases','/parks',
                       '/passportstamplocations','/people','/places','/thingstodo','/topics','/topics/parks','/tours','/visitorcenters
                       '/webcams']

         # API Name list
         nps_api_names=['activities','activitiesparks','alerts','amenities','amenitiesparksplaces','amenitiesparksvisitorcenters',
                        'articles','campgrounds','events','lessonplans','newsreleases','parks','passportstamplocations','people',
                        'places','thingstodo','topics','topicsparks','tours','visitorcenters','webcams']

         # Merge table and parks dataframes to reduce parks to just top 10, create code list and convert list to lowercase
         table_merged=table.merge(parks,'left',left_on='Park',right_on='Park Name')
         nps_park_codes= parks['Park Code'].tolist()
         top_ten_park_codes=table_merged['Park Code'].tolist()
         top_ten_park_codes=[x.lower() for x in top_ten_park_codes]

         # Pretty print result
         pprint.pformat(top_ten_park_codes,compact=True)

Out[6]:  "['grsm', 'yell', 'zion', 'romo', 'grte', 'grca', 'cuva', 'acad', 'olym', 'jotr']"
```

Extract/Transform: API loop

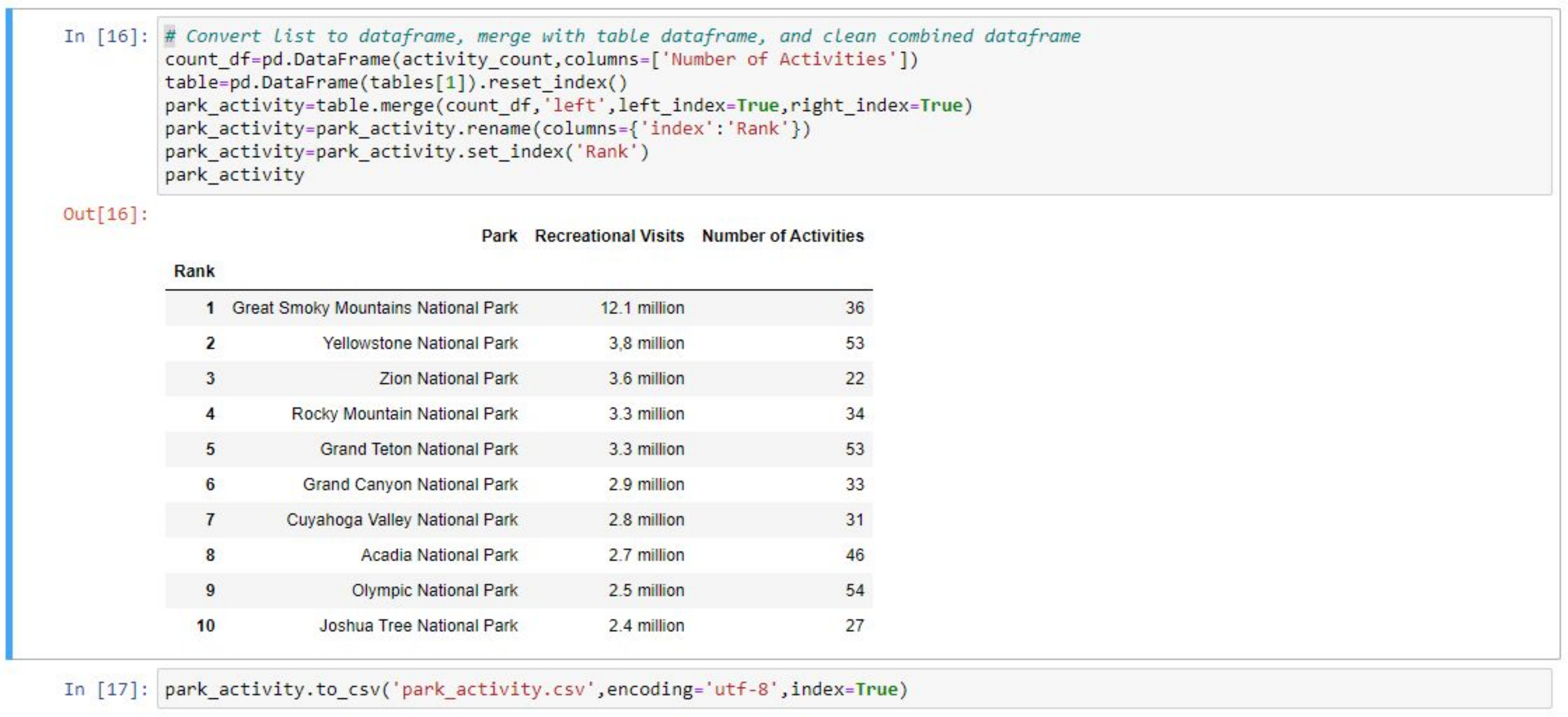


```
activity_count=[]
for i in top_ten_park_codes:
    url=f'{base_url}/parks?parkCode={i}&api_key={nps_key}'
    response=requests.get(url)
    top_ten=json.loads(response.content.decode('utf-8'))
    top_ten_a=dict(top_ten['data'][0])
    park_act=len([activity['name'] for activity in top_ten_a['activities']])
    activity_count.append(park_act)
```

```
activity_count
```

```
[36, 53, 22, 34, 53, 33, 31, 46, 54, 27]
```

Extract/Transform: API loop cont.



```
In [16]: # Convert list to dataframe, merge with table dataframe, and clean combined dataframe
         count_df=pd.DataFrame(activity_count,columns=['Number of Activities'])
         table=pd.DataFrame(tables[1]).reset_index()
         park_activity=table.merge(count_df,'left',left_index=True,right_index=True)
         park_activity=park_activity.rename(columns={'index':'Rank'})
         park_activity=park_activity.set_index('Rank')
         park_activity
```

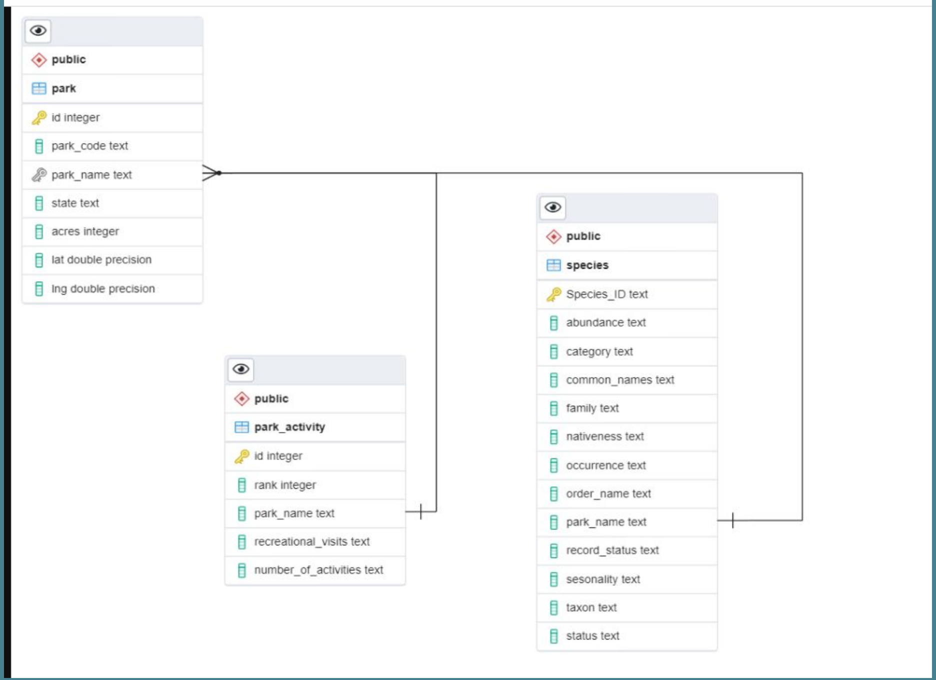
Out[16]:

	Park	Recreational Visits	Number of Activities
Rank			
1	Great Smoky Mountains National Park	12.1 million	36
2	Yellowstone National Park	3,8 million	53
3	Zion National Park	3.6 million	22
4	Rocky Mountain National Park	3.3 million	34
5	Grand Teton National Park	3.3 million	53
6	Grand Canyon National Park	2.9 million	33
7	Cuyahoga Valley National Park	2.8 million	31
8	Acadia National Park	2.7 million	46
9	Olympic National Park	2.5 million	54
10	Joshua Tree National Park	2.4 million	27

```
In [17]: park_activity.to_csv('park_activity.csv',encoding='utf-8',index=True)
```

Load Set-up: Created Tables in ERD tool

Entity Relationship Diagram

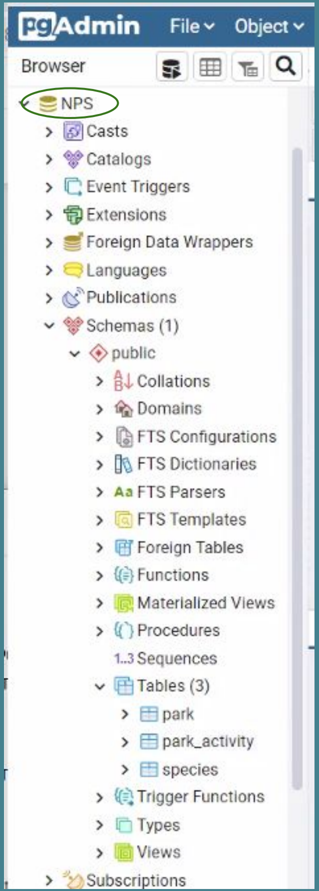


public
park
id integer
park_code text
park_name text
state text
acres integer
lat double precision
lng double precision

public
park_activity
id integer
rank integer
park_name text
recreational_visits text
number_of_activities text

public
species
Species_ID text
abundance text
category text
common_names text
family text
nativeness text
occurrence text
order_name text
park_name text
record_status text
sesonality text
taxon text
status text

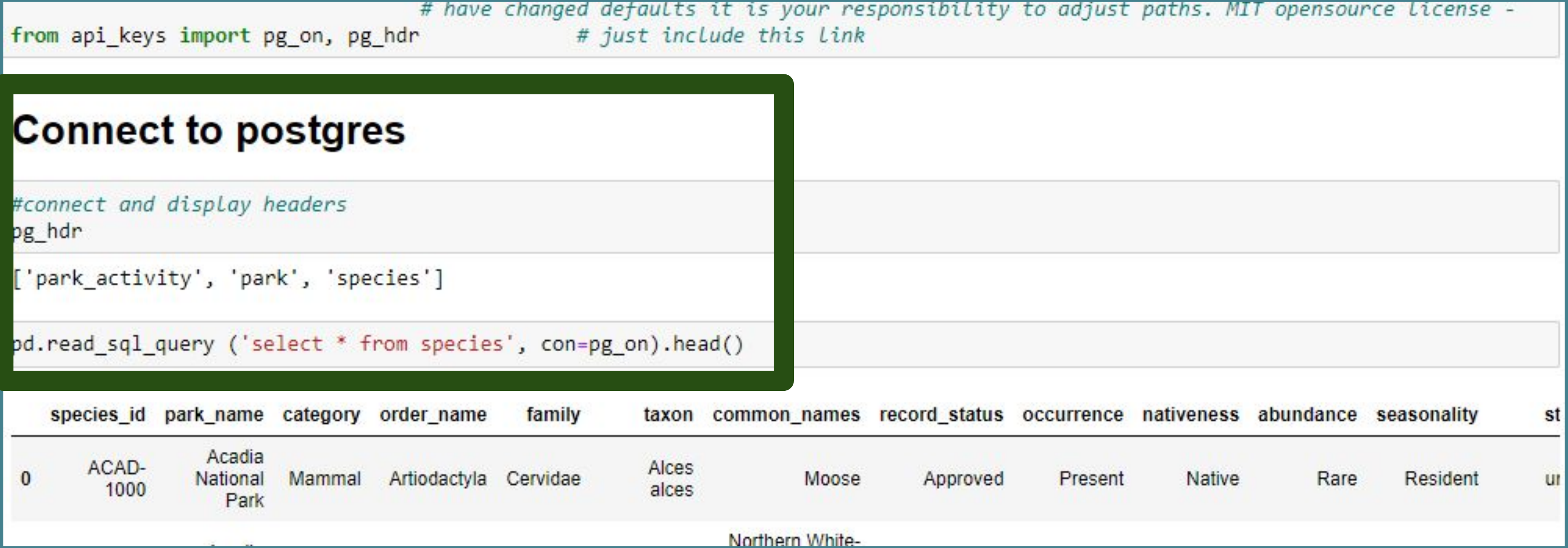
Load



```
-- This script was generated by a beta version of the ERD tool in pgAdmin 4.
BEGIN;

CREATE TABLE public.species
(
    "Species_ID" text NOT NULL,
    abundance text,
    category text,
    common_names text,
    family text,
    nativeness text,
    occurrence text,
    order_name text,
    park_name text,
    record_status text,
    sesonality text,
    taxon text,
    status text,
    PRIMARY KEY ("Species_ID")
);
CREATE TABLE public.park
(
    id integer NOT NULL,
    park_code text,
    park_name text,
    state text,
    acres integer,
    lat double precision,
    lng double precision,
    PRIMARY KEY (id)
);
```

```
CREATE TABLE public.park_activity
(
    id integer NOT NULL,
    rank integer,
    park_name text,
    recreational_visits text,
    number_of_activities text,
    PRIMARY KEY (id)
);

ALTER TABLE public.park
    ADD FOREIGN KEY (park_name)
    REFERENCES public.species (park_name)
    NOT VALID;


ALTER TABLE public.park
    ADD FOREIGN KEY (park_name)
    REFERENCES public.park_activity (park_name)
    NOT VALID;

END;
```

Load: PostSQL to Jupyter Notebook



```
                                              # have changed defaults it is your responsibility to adjust paths. MIT opensource license -
from api_keys import pg_on, pg_hdr                  # just include this link
```

Connect to postgres

```
#connect and display headers
pg_hdr
```

```
['park_activity', 'park', 'species']
```

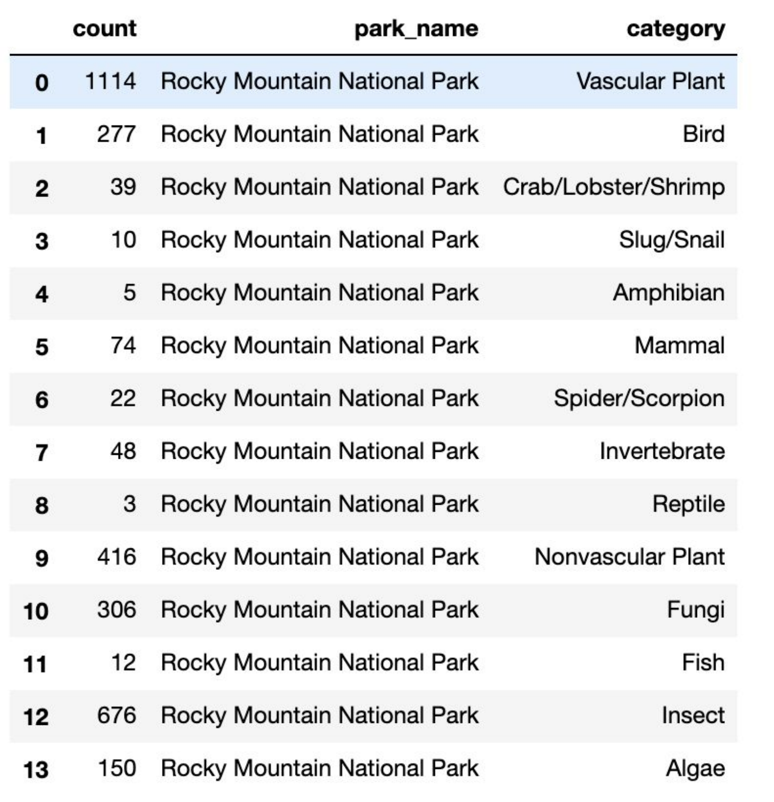
```
pd.read_sql_query ('select * from species', con=pg_on).head()
```

	species_id	park_name	category	order_name	family	taxon	common_names	record_status	occurrence	nativeness	abundance	seasonality	st
0	ACAD-1000	Acadia National Park	Mammal	Artiodactyla	Cervidae	Alces alces	Moose	Approved	Present	Native	Rare	Resident	ur
							Northern White-						

Connected by ezmode module, which was made possible by Sami.

Query Example

```
pd.read_sql_query("select count(*) as count, park_name, category
from species  WHERE park_name='Rocky Mountain National
Park'  group by park_name, category",  con=engine)
```



	count	park_name	category
0	1114	Rocky Mountain National Park	Vascular Plant
1	277	Rocky Mountain National Park	Bird
2	39	Rocky Mountain National Park	Crab/Lobster/Shrimp
3	10	Rocky Mountain National Park	Slug/Snail
4	5	Rocky Mountain National Park	Amphibian
5	74	Rocky Mountain National Park	Mammal
6	22	Rocky Mountain National Park	Spider/Scorpion
7	48	Rocky Mountain National Park	Invertebrate
8	3	Rocky Mountain National Park	Reptile
9	416	Rocky Mountain National Park	Nonvascular Plant
10	306	Rocky Mountain National Park	Fungi
11	12	Rocky Mountain National Park	Fish
12	676	Rocky Mountain National Park	Insect
13	150	Rocky Mountain National Park	Algae

Possible Query/ETL

Possible Queries:

- Abundance stats: Max, min, describe
- Price/park: activities, hours open, pets allowed, etc
- Possibilities are endless

Summary:

- Extract:
  - Kaggle dataset named Biodiversity in National Parks
    - Two csv’s: Park and Species
  - National Park Service ‘Visitation Numbers’ table scrapped
  - National Park Service API for top ten parks and activity count

- Transform:
  - Species dataset cleaning: removed commas and made into DataFrame
  - Table DF was used to filter Parks DataFrame by merging
  - Table DF was transformed into a list of park abbreviations for API queries
  - NPS API Parks activities data was merged into the Table DF and then cleaned

- Load:
  - ERD created for three tables: park, park_activity, species
  - The tables were then loaded into PostgreSQL: labeled NPS Database
  - NPS Queries in Jupyter Notebook