

Data sources:

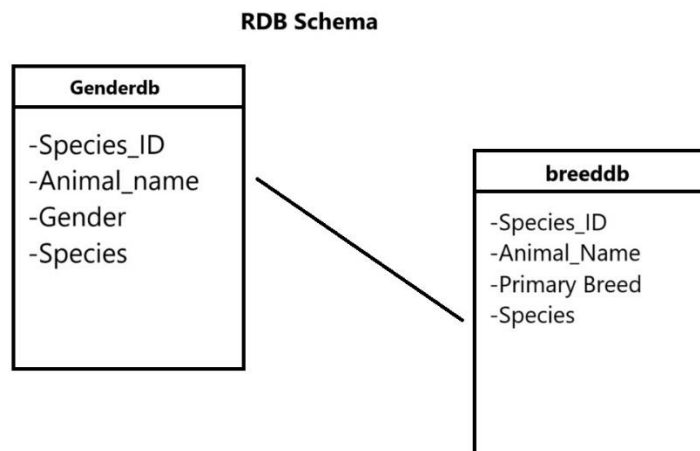
[pet-names/seattle_pet_licenses.csv at main · jnolis/pet-names](#) – used for animals data frame, to see pet names and breed

<https://www.api-ninjas.com/api/dogs> - API data source that has dog breed and temperament for our top 5 breeds.

https://gist.github.com/arturschaefer/abf8f94bcff14ace1b88c7977d651a74#file-breed_list-json – used to create generated name based on breed group and temperament

Cleaning challenges:

There were numerous cleaning challenges such as figuring out what data were not needed, what columns to drop, and how to deal with Nan values. When merging files, it became harder to do some when some parts of the data didn't match up.



Plots and Queries:

Copy of ISTA_322_HW5_Marvel-N | What's new in Microsoft Edge | Marvel Njunge ISTA 322 Final Pro | Njunge ISTA 322 Project.pdf

File | C:/Users/uai-laptop/Downloads/Njunge%20ISTA%20322%20Project.pdf

8 of 21

```
36934 Lary Huse M Dog
36935 Cricket M Dog
36936 Caledonia F Dog
36937 Ziggle M Dog
36938 Nala F Dog

[36826 rows x 3 columns]

tq """ CREATE TABLE Genderdb (
Species_ID INT AUTO_INCREMENT PRIMARY KEY,
Animal_Name VARCHAR(255) NOT NULL,
Gender VARCHAR(255) NOT NULL,
Species VARCHAR(255) NOT NULL
);"""

conn, cur = get_conn_cur()
cur.execute(tq)
conn.commit()
```

Traceback (most recent call last)
File <ipython-input-129-62da186cae2c> in <cell line: 9>():
7
8 conn, cur = get_conn_cur()
----> 9 cur.execute(tq)
10 conn.commit()

3 frames
/usr/local/lib/python3.10/dist-packages/mysql/connector/connection.py in _handle_result(self, packet)
646 return self._handle_eof(packet)
647 if packet[4] == 255:
--> 648 raise get_exception(packet)
649
650 # We have a text result set

ProgrammingError: 1058 (42501): Table 'Genderdb' already exists

Next steps: [Explain error](#)

49° Search 3:05 AM 12/11/2024

Copy of ISTA_322_HW5_Marvel-N | What's new in Microsoft Edge | Marvel Njunge ISTA 322 Final Pro | Njunge ISTA 322 Project.pdf

File | C:/Users/uai-laptop/Downloads/Njunge%20ISTA%20322%20Project.pdf

9 of 21

https://colab.research.google.com/drive/1XLhqn_UjzK_UAaPaagUuQyYb4tRUK#scrollto=IbUuWCNEBku&printMode=true 8/21

12/11/24, 2:37 AM Copy of ista_322_final_project.ipynb - Colab

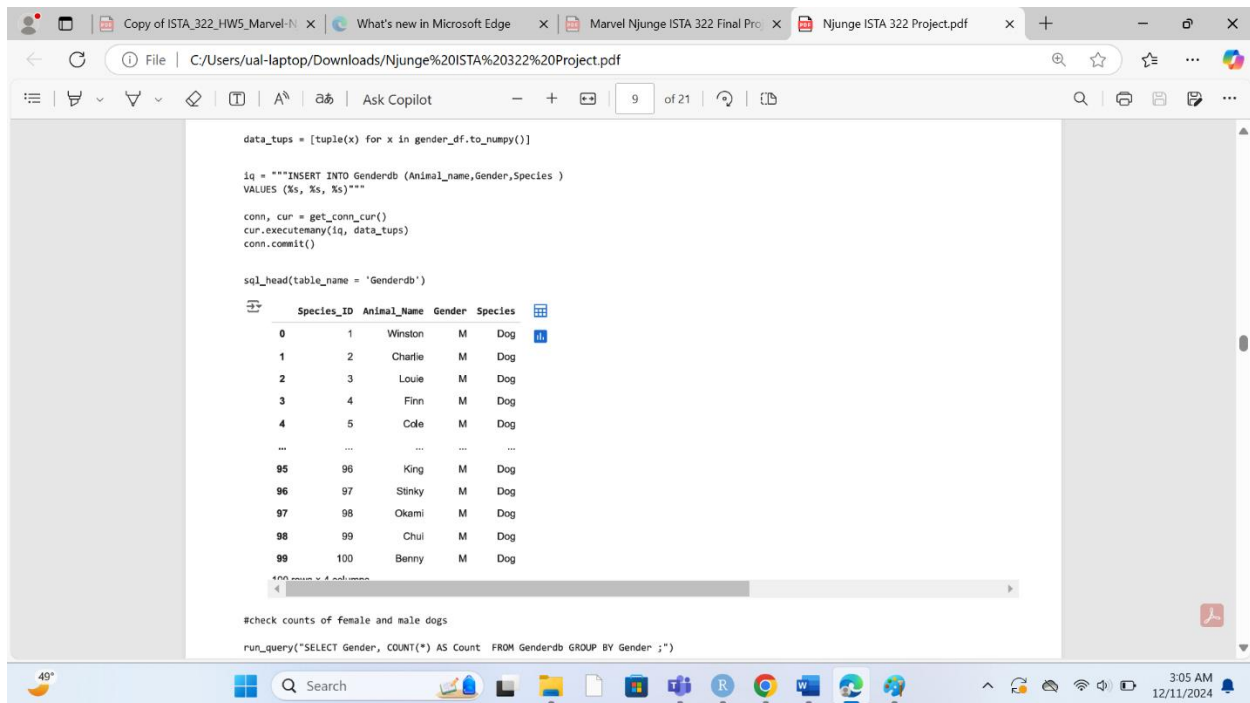
| | Species_ID | Animal_Name | Gender | Species |
|-----|------------|-------------|--------|---------|
| 0 | 1 | Winston | M | Dog |
| 1 | 2 | Charlie | M | Dog |
| 2 | 3 | Louie | M | Dog |
| 3 | 4 | Finn | M | Dog |
| 4 | 5 | Cole | M | Dog |
| ... | ... | ... | ... | ... |
| 95 | 96 | King | M | Dog |
| 96 | 97 | Stinky | M | Dog |
| 97 | 98 | Okami | M | Dog |
| 98 | 99 | Chui | M | Dog |
| 99 | 100 | Benny | M | Dog |

```
data_tups = [tuple(x) for x in gender_df.to_numpy()]

lq = """INSERT INTO Genderdb (Animal_name,Gender,Species )
VALUES (%s, %s, %s)"""

conn, cur = get_conn_cur()
cur.executemany(lq, data_tups)
conn.commit()
```

49° Search 3:05 AM 12/11/2024



#creating SQL

```
mysql_address = 'myista322db.cb00mkm2tlt.us-east-2.rds.amazonaws.com'
mysql_username='admin'
mysql_password='Rdg_Rugg43!'
mysql_database = 'myista322db'

def get_conn_cur():
    cnx = mysql.connector.connect(user=mysql_username, password=mysql_password,
                                  host=mysql_address,
                                  database=mysql_database, port='3306');
    return (cnx, cnx.cursor())

def run_query(query_string):
    conn, cur = get_conn_cur() # get connection and cursor
    cur.execute(query_string) # executing string as before
    my_data = cur.fetchall() # fetch query data as before
    result_df = pd.DataFrame(my_data, columns=cur.column_names)
    cur.close() # close
    conn.close() # close
    return result_df

def sql_head(table_name):
    conn, cur = get_conn_cur() #get connection and cursor
```

https://colab.research.google.com/drive/1XLhqsh_JfzR_UaApdabg0OdPy64trOUK#scrollTo=ifbU0wCNEBk0&printMode=

12/11/24, 2:37 AM

Copy of ista_322_final_project.ipynb - Colab

```
#get head information
table rows query = """ SELECT * FROM %s LIMIT 100; """ % table name
```

```

# get row information
table_rows_query = """ SELECT * FROM %s LIMIT 100; """ % table_name
cur.execute(table_rows_query)
my_data = cur.fetchall() # fetch results

# Create a dataframe that combines sql table with column names and return
df = pd.DataFrame(my_data, columns=cur.column_names)

cur.close()
conn.close()
return df

```

Gender Table

```
gender_df = dogs[["Animal_Name", 'Gender', 'Species']]
```

```
print(gender_df)
```

```

➡ Animal_Name Gender Species
0      Winston      M      Dog
1      Charlie      M      Dog
2       Louie      M      Dog
3        Finn      M      Dog
4         Cole      M      Dog
...      ...      ...      ...
36934  Carly Rose      M      Dog
36935   Cricket      M      Dog
36936  Caledonia      F      Dog
36937    Ziggie      M      Dog
36938     Nala      F      Dog

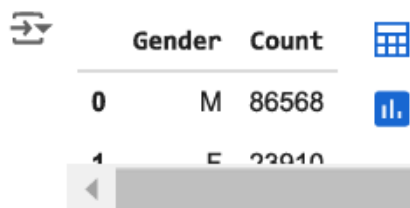
```


```
[36826 rows x 3 columns]
```

```

tq = """ CREATE TABLE Genderdb (
    Species_ID INT AUTO_INCREMENT PRIMARY KEY,
    Animal_Name VARCHAR(255) NOT NULL,

```



| | Gender | Count | |
|---|--------|-------|---|
| 0 | M | 86568 |  |
| 1 | F | 23910 | |

Results show that there are 86568 male genders and 23910 female genders present

Breed Table

Start coding or [generate](#) with AI.

https://colab.research.google.com/drive/1XLhqsh_JfzR_UaApdabg0OdPy64trOUK#scrollTo:

12/11/24, 2:37 AM


Copy of ista_322_fin

```
breed_df = dogs[["Animal_Name", 'Primary Breed', 'Species']]
```

```
print(breed_df)
```



| Animal Name | Primary Breed Species |
|-------------|-----------------------|
|-------------|-----------------------|



| | Animal_Name | Primary Breed | Species |
|-------|-------------|----------------------|---------|
| 0 | Winston | Pug | Dog |
| 1 | Charlie | Pug | Dog |
| 2 | Louie | Boxer | Dog |
| 3 | Finn | Hound | Dog |
| 4 | Cole | Pug | Dog |
| ... | ... | ... | ... |
| 36934 | Carly Rose | Retriever, Labrador | Dog |
| 36935 | Cricket | Poodle, Miniature | Dog |
| 36936 | Caledonia | Bernese Mountain Dog | Dog |
| 36937 | Ziggie | Miniature Pinscher | Dog |
| 36938 | Nala | Kai Ken | Dog |

[36826 rows x 3 columns]

```
tq = """ CREATE TABLE breddb (
    Species_ID INT AUTO_INCREMENT PRIMARY KEY,
    Animal_Name VARCHAR(255) NOT NULL,
    `Primary Breed` VARCHAR(255) NOT NULL,
    Species VARCHAR(255) NOT NULL
);"""
```

```
conn, cur = get_conn_cur()
cur.execute(tq)
conn.commit()
```

```
sql_head(table_name = 'breddb')
```



| Species ID | Animal Name | Primary Breed | Species |
|------------|-------------|---------------|---------|
|------------|-------------|---------------|---------|





Species ID Animal Name Primary Breed Species



```
data_tups = [tuple(x) for x in breed_df.to_numpy()]
```

```
iq = """INSERT INTO breddb (Animal_name,`Primary Breed`,Species )
VALUES (%s, %s, %s)"""
```

```
conn, cur = get_conn_cur()
cur.executemany(iq, data_tups)
conn.commit()
```

```
sql_head(table_name = 'breddb')
```



Species_ID Animal_Name Primary Breed Species





| | | | | |
|-----|-----|---------|----------------------------|-----|
| 0 | 1 | Winston | Pug | Dog |
| 1 | 2 | Charlie | Pug | Dog |
| 2 | 3 | Louie | Boxer | Dog |
| 3 | 4 | Finn | Hound | Dog |
| 4 | 5 | Cole | Pug | Dog |
| ... | ... | ... | ... | ... |
| 95 | 96 | King | Terrier, American Pit Bull | Dog |
| 96 | 97 | Stinky | Chihuahua, Short Coat | Dog |
| 97 | 98 | Okami | Retriever, Labrador | Dog |
| 98 | 99 | Chui | Chihuahua, Short Coat | Dog |
| 99 | 100 | Benny | Retriever, Golden | Dog |



```
#most popular breed
popular_breeds = """
SELECT `Primary Breed`, COUNT(*) AS Top_Breed
FROM breeddb
GROUP BY `Primary Breed`
ORDER BY Top_Breed DESC
LIMIT 5;
"""
```

```
run_query(popular_breeds)
```



| | Primary Breed | Top_Breed |  |
|---|-----------------------|-----------|---|
| 0 | Retriever, Labrador | 5281 |  |
| 1 | Retriever, Golden | 1967 | |
| 2 | Chihuahua, Short Coat | 1952 | |
| 3 | German Shepherd | 1060 | |
| 4 | Terrier | 1003 | |

Results show that top breeds are the Labrador Retriever, Golden Retriever, C

```
#which genders are in the top 5 breeds
top_breed_gender = """
SELECT b.`Primary Breed`, g.Gender, COUNT(*) AS top_gender_breeds
FROM breeddb b
JOIN Genderdb g ON b.Animal_Name = g.Animal_Name
WHERE b.`Primary Breed` IN (SELECT `Primary Breed` FROM breeddb GROUP
GROUP BY b.`Primary Breed`, g.Gender
ORDER BY top_gender_breeds DESC;
"""
```



```

top_breed_gender = run_query("""
SELECT b.`Primary Breed`, g.Gender, COUNT(*)
FROM breeddb b
JOIN Genderdb g ON b.Animal_Name = g.Animal_
WHERE b.`Primary Breed` IN (SELECT `Primary
GROUP BY b.`Primary Breed`, g.Gender
ORDER BY top_gender_breeds DESC
limit 50;
""")

```

```

# Reshape the data for plotting
plot_data = top_breed_gender.pivot(index='Pr

```

```

# Create the grouped bar chart
plot_data.plot(kind='bar', figsize=(10, 6),
plt.title('Gender Distribution in Top 5 Dog
plt.xlabel('Breed')
plt.ylabel('Number of Dogs')
plt.xticks(rotation=45, ha='right')
plt.legend(title='Gender')
plt.tight_layout()
plt.show()

```

```

top_breed_gender = run_query("""
SELECT b.`Primary Breed`, g.Gender, COUNT(*) AS top_gender_breeds
FROM breeddb b
JOIN Genderdb g ON b.Animal_Name = g.Animal_Name
WHERE b.`Primary Breed` IN (SELECT `Primary Breed` FROM breeddb GROUP BY `Primary Breed` ORDER BY COUNT(*) DESC)
GROUP BY b.`Primary Breed`, g.Gender
ORDER BY top_gender_breeds DESC
limit 50;
""")

```

```

# Reshape the data for plotting
plot_data = top_breed_gender.pivot(index='Primary Breed', columns='Gender', values='top_gender_breeds')

```

```

# Create the grouped bar chart
plot_data.plot(kind='bar', figsize=(10, 6), color=['pink', 'skyblue'])
plt.title('Gender Distribution in Top 5 Dog Breeds')
plt.xlabel('Breed')
plt.ylabel('Number of Dogs')
plt.xticks(rotation=45, ha='right')
plt.legend(title='Gender')
plt.tight_layout()
plt.show()

```

