**Shelter Dogs**

For this project, we play the role of a volunteer data analyst for a shelter dog organization where the management have few questions.

Data in CSV format is from Kaggle.com and is quite messy. The scope of this project is not to clean the data but rather to answer the questions using SQL.

The database contains one table:

**shelter\_dogs**

|  |  |  |
| --- | --- | --- |
| column | data\_type | description |
| ID | bigint | Input ID |
| name | text | Given name |
| age | double | Estimate age |
| sex | text | Sex |
| breed | text | Breed |
| date found | text | Date dog was found |
| adoptable\_from | text | Date is ready for adoption |
| posted | text | Date adoption ad is posted |
| color | text | Fur colour |
| coat | text | Coat length |
| size | text | Size when found |
| neutered | text | Neutered when found |
| housebroken | text | Housebroken |
| likes\_people | text | Likes people |
| likes\_children | text | Likes children |
| get\_along\_males | text | Get along male dogs |
| get\_along females | text | Get along female dogs |
| get along\_cats | text | Get along cats |
| keep\_in | text | Where recommended to keep |

**1. Upload data to SQL server**

To work with SQL requires a server. I choose an online server rather to work locally. Will use python’s module panda to read the CSV file and then push the data to the server.

|  |
| --- |
| # Import python modules  from sqlalchemy import create\_engine  import pandas as pd  # Create connection to server  host="sql.freedb.tech"  database="freedb\_mysql8.0database"  user="freedb\_Albert"  password="\*\*\*\*\*\*\*\*\*\*\*\*\*"  port = 3306  def get\_connection():  return create\_engine(  url="mysql+pymysql://{0}:{1}@{2}:{3}/{4}".format(  user, password, host, port, database  )  )  # Send data from CSV file to server  df = pd.read\_csv('/Users/albert/Data Analytics/1. Projects & Databases/6.(SQL) Shelter-Dogs/ShelterDogs.csv')  df.to\_sql('shelter\_dogs',get\_connection(), index=False)  print ("Data\_Exported")  # Server to df (if needed)  # sql =  # '''  # SELECT \*  # FROM shelter\_dogs  # LIMIT 3;  # '''  # df = pd.read\_sql(sql, get\_connection())  # print (df) |

**2. Testing the connection and see the data**

Created a new connection with SQLTOOLS in VS Code.

|  |
| --- |
| /\* DESCRIBE shelter\_dogs; \*/  SELECT \* FROM shelter\_dogs LIMIT 2; |
| ID name age sex breed date\_found adoptable\_from posted color coat size neutered housebroken likes\_people likes\_children get\_along\_males get\_along\_females get\_along\_cats keep\_in  23807 Gida 0.25 female Unknown Mix 12/10/2019 12/11/2019 12/11/2019 red short small no NULL NULL NULL NULL NULL NULL NULL  533 Frida √âs Ricsi 0.17 female Unknown Mix 12/01/2019 12/01/2019 12/09/2019 black and white short small no NULL yes yes yes yes yes NULL |

Connection successfully. From the first look can see bad data formatting, misspelling, missing data.

**3. Any record for 'Border Terrier'?**

|  |
| --- |
| SELECT name  FROM shelter\_dogs  WHERE name LIKE '\_order%'; |
| name  No data |

No Border terrier in the name list.

**4. Check for missing data**

Perform a quick check for missing data.

|  |
| --- |
| SELECT COUNT(\*) AS total\_rows,  COUNT(ID),  COUNT(name),  COUNT(age),  COUNT(sex),  COUNT(breed),  COUNT(date\_found),  COUNT(neutered),  COUNT(housebroken),  COUNT(likes\_people)  FROM shelter\_dogs; |
| total\_rows COUNT(ID) COUNT(name) COUNT(age) COUNT(sex) COUNT(breed) COUNT(date\_found) COUNT(neutered) COUNT(housebroken) COUNT(likes\_people)  2937 2937 2845 2937 2937 2937 2937 1852 460 1999 |

**5. Is ID the primary key / unique?**

|  |
| --- |
| SELECT ID, COUNT(ID)  FROM shelter\_dogs  GROUP BY ID  ORDER BY COUNT(ID) DESC; |
| ID COUNT(ID)  1 12  9 12  7 11  10 11  8 10  … |

ID not unique. Not great.

**6. Are there duplicate rows?**

|  |
| --- |
| SELECT ID, name, age, sex, breed, date\_found, adoptable\_from, posted, color, coat, size, neutered, housebroken, likes\_people, likes\_children, get\_along\_males, get\_along\_females, get\_along\_cats, keep\_in  FROM shelter\_dogs  GROUP BY ID, name, age, sex, breed, date\_found, adoptable\_from, posted, color, coat, size, neutered, housebroken, likes\_people, likes\_children, get\_along\_males, get\_along\_females, get\_along\_cats, keep\_in  HAVING COUNT(\*) > 1; |
| ID name age sex breed date\_found adoptable\_from posted color coat size neutered housebroken likes\_people likes\_children get\_along\_males get\_along\_females get\_along\_cats keep\_in  No data |

No duplicate rows.

**7. Are more puppies or older dogs abandoned?**

I am thinking to create 3 groups(bins) of age, up to 10 months for puppy, up to 12 years for adult and over 12 years old category.

|  |
| --- |
| SELECT category,  COUNT(category) as 'count',  CONCAT(ROUND( COUNT(category)/(SELECT COUNT(age) FROM shelter\_dogs), 2),' %') AS percent  FROM(  SELECT CASE WHEN age < 0.84 THEN 'puppy'  WHEN age > 12.0 THEN 'old'  ELSE 'adult'  END AS category,  age  FROM shelter\_dogs  ) AS tab  GROUP BY category  ORDER BY count DESC |
| |  |  |  | | --- | --- | --- | | category | count | percent | | adult | 2319 | 0.79 % | | old | 531 | 0.18 % | | puppy | 87 | 0.03 % | |

**8. How many days pass from when a dog is found up to is ready for adoption.**

One way would be to extract day, month and year. Then create new columns to easily extract date difference in days.

|  |
| --- |
| -- extract number of days and count, where number of days more than 0  SELECT  DATEDIFF(adoptable\_from\_2, date\_found\_2) as days,  COUNT(\*) as count  FROM(  -- created new columns for date\_found and adaptable\_from in date format  WITH tab AS (  -- extract day, month, year from date\_found and adaptable\_from columns  SELECT SUBSTRING\_INDEX(date\_found, '/', 1) AS date\_found\_month,  SUBSTRING\_INDEX( SUBSTRING\_INDEX(date\_found, '/', 2),'/',-1 ) AS date\_found\_day,  SUBSTRING\_INDEX(date\_found, '/', -1) AS date\_found\_year,  SUBSTRING\_INDEX(adoptable\_from, '/', 1) AS adoptable\_from\_month,  SUBSTRING\_INDEX( SUBSTRING\_INDEX(adoptable\_from, '/', 2),'/',-1 ) AS adoptable\_from\_day,  SUBSTRING\_INDEX(adoptable\_from, '/', -1) AS adoptable\_from\_year  FROM shelter\_dogs  )    SELECT DATE(CONCAT(date\_found\_year, '-',date\_found\_month, '-', date\_found\_day)) as date\_found\_2,  DATE(CONCAT(adoptable\_from\_year, '-',adoptable\_from\_month, '-', adoptable\_from\_day)) as adoptable\_from\_2 FROM tab  ) AS sub  WHERE DATEDIFF(adoptable\_from\_2, date\_found\_2) >= 0 -- as they are dates with adoption dates before date found  GROUP BY days  ORDER BY count DESC |
| days count  0 2404  15 75  1 43  2 13  38 12  31 11  … |

**9. Create a new table with dog breeds.**

|  |
| --- |
| SELECT ROW\_NUMBER() OVER (ORDER BY breed) as ID,  breed  FROM shelter\_dogs  WHERE breed NOT LIKE 'Adoptable From:'  GROUP BY breed  HAVING breed NOT LIKE '%Mix%' |
| ID breed  1 Afghan Hound  2 Akita  3 American Bulldog  4 Beagle  5 Belgian Malinois  … |

**10. Top 3 ranked dog names for both male and female**

|  |
| --- |
| SELECT name, sex, \_count, \_rank  FROM (  SELECT name,  sex,  COUNT(name) AS \_count,  RANK() OVER(ORDER BY COUNT(name) DESC) AS \_rank  FROM shelter\_dogs  WHERE sex = 'male'  GROUP BY name ) AS male\_table  WHERE \_rank < 4  UNION  SELECT name, sex, \_count, \_rank  FROM (  SELECT name,  sex,  COUNT(name) AS \_count,  RANK() OVER(ORDER BY COUNT(name) DESC) AS \_rank  FROM shelter\_dogs  WHERE sex = 'female'  GROUP BY name ) AS female\_table  WHERE \_rank < 4 |
| name sex \_count \_rank  Maci male 9 1  Kormi male 8 2  Bobi male 8 2  Buksi male 8 2  Lili female 18 1  Man√≥ female 7 2  Kira female 6 3  Bella female 6 3  L√©di female 6 3  Pic√∫r female 6 3 |