Worksheet 20 - SQL joins

Wednesday, November 20, 2024

DS 002R - Jo Hardin

Name: _____

Names of people you worked with: _____

Name one tradition around next week's time off that you grew up with.

Task:[1]

Consider the following two tables. The first table, `dogs`, lists different dogs as observational units. The owners table, `users`, contains information on each dog owner. The `users` own dogs. The `ownerid` column in the `dogs` table corresponds to the `userid` column of the `users` table (`ownerid` is a foreign key that references the `users` table).

```
CREATE TABLE dogs (
    dogid integer,
    ownerid integer,
    name varchar,
    breed varchar,
    age integer,
    PRIMARY KEY (dogid),
    FOREIGN KEY (ownerid) REFERENCES users(userid)  );
```

```
CREATE TABLE users (
    userid integer,
    name varchar,
    age integer,
    PRIMARY KEY (userid)  );
```

1. Write a query that lists the names of all the dogs that "Josh Hug" owns.
2. Write a query that finds the name of the user and the number of dogs that user owns for the user that owns the most dogs in the database. The query should only return 1 user.

[1] Questions come from CS 186 at UC Berkeley, https://cs186berkeley.net/notes/note2/#practice-questions.

Solution

1. Write a query that lists the names of all the dogs that "Josh Hug" owns.

```
SELECT dogs.name
FROM dogs INNER JOIN users ON dogs.ownerid = users.userid
WHERE users.name="Josh Hug";
```

We now need information from both tables (the dog name is only in the `dogs` table and the owner name is only in the `users` table). The join condition is `dogs.ownerid=users.userid` because we only want to get rows with the dog and its owner in it. Finally we add the predicate to the `WHERE` clause to only get Josh's dogs.

2. Write a query that finds the name of the user and the number of dogs that user owns for the user that owns the most dogs in the database. The query should only return 1 user.

```
SELECT users.name, COUNT(*) AS num_dogs
FROM users INNER JOIN dogs on users.userid = dogs.ownerid
GROUP BY users.userid
ORDER BY num_dogs DESC
LIMIT 1;
```

We can use an `ORDER BY` combined with a `LIMIT` to select the first n most rows (with n being 1 in this case). We `GROUP BY` the name because we want our groups to be all about one user. We have to include `userid` in the `GROUP BY`, because users may share the same name.