Lab 3 Write Up

- Gary Miller

Commentary on Basic Query Structures

First in order to access an sqlite3 database using a python program two important statements must be executed to establish a connection, these commands are:

```
import sqlite3
conn = sqlite3.connect("DatabaseFileName.sqlite3")
```

The first of these statements imports the necessary library for sqlite3 and the second establishes a connection object named conn and it connects to the file given as a parameter. It is important to note if the sqlite3 file is not in the working directory, a direct path will have to be given as a parameter to be able to establish the connection. after successfully establishing a connection this connection object conn can be used to query the data within the database.

The first of the four basic commands used to query data is INSERT. the general structure of this command is as follows:

```
conn.execute("INSERT INTO TableName (attribute1, attribute2, ..., atributeN) VALUES (val
conn.commit()
```

By using the conn variable and its execute command that was found in the basic API for sqlite3 and python, it is as simple as typing in the command nearly as you would in a terminal with only the sqlite3 program. For general purposes this statement shows you can enter values for as many attributes as you desire, it is essential that for each attribute a value is given. Also, the database will not reflect any changes that were made unless the changes are committed to the database using the second statement.

The second of the four basic statements is DELETE, the general structure of this command is as follows:

```
conn.execute("DELETE FROM TableName WHERE attribute (meets some condition)")
conn.commit()
```

This is very similar to the INSERT statement as seen before. It is important to note that the changes must also be committed in this situation. Also, the semicolon is not needed to close each statement.

Similar to the INSERT statement, the UPDATE statement can be used to change values of data that already exists in the database. The general structure of this command is as follows:

```
conn.execute("UPDATE TableName SET attribute1 = value1 WHERE attribute2 (meets some cond
conn.commit()
```

In this statement it is possible to update the value of some attribute in all cases where a certain condition is met. As seen before changes must be committed in order to reflect the new data within the database.

The final of the four basic query commands is SELECT and is the most commonly used of all the commands. This allows a user to actually access and retrieve data form the database. The general structure of the commands looks like:

```
cursor = conn.execute("SELECT attribute FROM TableName WHERE attribute (meets some condi
```

Something that is clearly different about this statement is the fact that the values returned are saved into the variable cursor. This is done so that a for loop can easily be used to print the returned data to the console.

Output From Running NBAStats.py

Executing the program written for this lab, named NBAStats.py, gave the following output for the queries executed:

millerg2@aldenv165:~/CS380/cs380F2016-millerg2/labs/lab3$ python NBAStats.py Attempting to connect to database Opened database successfully Inserting the top four rookies drafted for this upcoming season Records created successfully Record was not successfully created Selecting rookies to show they were successfully inserted ID: 498 Name: Ben Simmons

ID: 499 Name: Brandon Ingram

ID: 500 Name: Jaylen Brown

Deleting last inserted record from the database Number of entries: 500

Show that the record was successfully deleted Number of players : 499

Updating Kevin Durant's team to be Miami Data before the update statement: Player Name: Kevin Durant Team: Okc

Data after the update statement: Player Name: Kevin Durant Team: Mia

Update was done successfully Printing players who average more than 25 points per game name: Damian Lillard PPG: 25.1

name: DeMarcus Cousins PPG: 26.9

name: James Harden PPG: 29.0

name: Kevin Durant PPG: 28.2

name: LeBron James PPG: 25.3

name: Stephen Curry PPG: 30.1

Selecting all teams from the western conference Team name: Dal Team name: Den Team name: Gol Team name: Hou Team name: Lac Team name: Lal Team name: Mem Team name: Min Team name: Nor Team name: Okc Team name: Pho Team name: Por Team name: Sac Team name: San Team name: Uta Selecting all teams that achieved more than 50 wins Team name: Cle Wins: 57

Team name: Gol Wins: 73

Team name: Lac Wins: 53

Team name: Okc Wins: 55

Team name: San Wins: 67

Team name: Tor Wins: 56

Select Players and their Team name who average more than 25 points per game and played in the Western conference Player name: Damian Lillard Player's PPG: 25.1 Player's team: Por Team's conference: West

Player name: DeMarcus Cousins Player's PPG: 26.9 Player's team: Sac Team's conference: West

Player name: James Harden Player's PPG: 29.0 Player's team: Hou Team's conference: West

Player name: Stephen Curry Player's PPG: 30.1 Player's team: Gol Team's conference: West

Selecting teams from the Eastern conference group by least amount of wins to most Team Name: Phi Wins: 10

Team Name: Bro Wins: 21

Team Name: Nyk Wins: 32

Team Name: Mil Wins: 33

Team Name: Orl Wins: 35

Team Name: Was Wins: 41

Team Name: Chi Wins: 42

Team Name: Det Wins: 44

Team Name: Ind Wins: 45

Team Name: Mia Wins: 48

Team Name: Tor Wins: 56

Team Name: Cle Wins: 57

Selecting the top 20 players in the NBA with respect to versatility index Player Name: Russell Westbrook VI Index: 14.5

Player Name: LeBron James VI Index: 12.1

Player Name: Stephen Curry VI Index: 12.0

Player Name: Kevin Durant VI Index: 11.7

Player Name: DeMarcus Cousins VI Index: 11.6

Player Name: James Harden VI Index: 11.5

Player Name: Pau Gasol VI Index: 11.4

Player Name: John Wall VI Index: 11.0

Player Name: Rajon Rondo VI Index: 10.7

Player Name: Tyreke Evans VI Index: 10.5

Player Name: Alan Williams VI Index: 10.4

Player Name: Tim Frazier VI Index: 10.3

Player Name: Nikola Jokic VI Index: 10.2

Player Name: Kris Humphries VI Index: 10.1

Player Name: Paul George VI Index: 10.0

Player Name: Jared Sullinger VI Index: 9.9

Player Name: Paul Millsap VI Index: 9.8

Player Name: Ish Smith VI Index: 9.7

Player Name: Joakim Noah VI Index: 9.6

Player Name: Shelvin Mack VI Index: 9.5

Operation done successfully

Create Table Structure

CREATE TABLE Player (ID INTEGER PRIMARY KEY NOT NULL, Name not null, Team CHAR(3), Position CHAR(2), Age INTEGER, GP INTEGER, MPG INTEGER, FTA INTEGER, FTPerc FLOAT(3), TwoPA INTEGER, TwoPPerc FLOAT(3), ThreePA INTEGER, ThreePPerc FLOAT(3), TSPerc FLOAT(3), PPG FLOAT(3), RPG FLOAT(3), TRBPerc FLOAT(3), APG FLOAT(3), ASTPerc FLOAT(3), SPG FLOAT(3), BPG FLOAT(3), VI FLOAT(1));

CREATE TABLE Team (TeamName PRIMARY KEY NOT NULL, Conference CHAR(4), Division VARCHAR(10), GP INTEGER, PPG FLOAT(1), OPPG FLOAT(1), PDIF FLOAT(1), PACE FLOAT(1), OEFF FLOAT(1), DEFF FLOAT(1), EDIF FLOAT(1), SoS FLOAT(2), SAR FLOAT(2), CONS

FLOAT(1), A4F FLOAT(3), Win INTEGER, Loss INTEGER, WinPerc FLOAT(3), PWinPerc FLOAT(3), EWinPerc FLOAT(3), ACH FLOAT(3));

Source Code

The source code for the lab exists in the python program NBAStats.py in the lab3 directory in my share repository for the course.

Show changes

Changes made to the database are shown by using SELECT statements to show the data before and after changes were made.

Challenges Faced

No major challenges were faced in completing this lab. The only difficulty was familiarizing myself with the python commands used to access sqlite3 files. Also, one difficulty was getting data form SELECT stamens to print to the console correctly, but this was eventually figured out with a trial and error approach.