291 Mini project 2 Report

General Overview:

load-json.py:

This program first prompts the user to enter the name of a json file in the same directory as the program and the port number which a mongodb server is assumed to be running on. The program then will create a database name ‘291db’ and a collection named ‘dblp’ assuming they don’t exist. If the collection does exist then the program will drop it and create a new one. After all this the program will add the data from the json file into the collection and end.

main.py:

This is the main program for this assignment. It includes a menu that allows the user to 1. Search for articles, 2. Search for authors, 3. List the venues, and 4. Add an article.

1. Searching for articles will prompt the user to enter 1 or more keywords then for all matching articles displays the id, the title, the year and the venue fields. The user can then choose to select an article which will display more information on that article and information on articles that reference the article.
2. Searching for authors will prompt the user to enter 1 or more keywords then for all matching authors the authors name and their number of publications is displayed. The user can then select an author to display more information on them and their articles.
3. Listing the venues will prompt the user to enter the number(n) of venues they want to list. Then for the top ‘n’ venues their number of articles, and the number of papers that reference an article from the venue. The venues are sorted by the number of articles that reference an article from that venue.
4. Adding an article will first prompt the user to provide a unique id, a title, a list of authors, and a year. The program will then add that article to the database and set its abstract and venue to NULL. References is set to an empty array and n_citations is set to 0.

After any action the user will be returned to the menu. There will also be an option to quit the program in the menu.

User Guide:

Both load-json.py and main.py will run in the terminal and will provide prompts to the user so the user knows what to type into the terminal for their intended purpose. At any stage where user input is required the user will be told by the prompt what they can enter and what each option will do.

Software Design:

load-json.py:

valid_port():

This function prompts the user to enter a port number, then tries to convert the input to an int, if invalid input is received then the program will prompt the user to enter a port number again until one is received. Returns the port number int.

create_index():

This function indexes the collection

init_collection():

This function adds the data from the json file into the database and calls create_index to index the collection.

main():

Main function of the program

Main.py:

valid_port():

Same as valid_port in load-json.py

list_venues(db):

Handles listing the venue's functionality. This function prompts the user to enter a number ‘n’ then displays the top ‘n’ venues along with the number of articles from that venue and the number of references to an article from that venue made by other articles. The function takes the database object as an argument and returns nothing.

add_article(db):

Allows the user to add a new article to the database. The user will be prompted to enter a unique id, a title, a list of authors and a year. All other fields will be set to 0 or NULL depending on the datatype of that field. Once valid input is received this function adds the articles data to the database. Takes a database object ‘db’ as an argument and returns nothing.

searchArticles(db):

This function allows the user to search for articles by providing keywords. An article matches if the keyword appears in any of title, authors, abstract, venue and year fields. For each matching article its id, title, the year and the venue fields are shown. The user can then select an article and all information on that article will be displayed along with information about articles that reference the selected article.Takes a database object ‘db’ as an argument and returns nothing.

searchAuthor(db):

This function allows the user to search for authors by providing keywords. For each matching author their name and number of publications are shown. Matching authors are determined by how if the authors name contains the keyword provided. The user is then able to select an author and display the title, year and venue for all of their articles. Takes a database object ‘db’ as an argument and returns nothing.

main():

This is the main function of the program, it starts calling valid_port() to get the port number from the user then connects to the database and makes a database object. The main function also runs the menu screen and calls other functions for the different program functionalities depending on input from the user.

Testing Strategy:

Individual functions/functionalities were tested as they were being created by using print statements to view the data being collected from the database and manual checks on the database to make sure that all queries were functioning correctly. A large dataset was used to test more rare edge cases and a smaller database was used for checking the correctness of a query manually.

Group Work Breakdown:

mawilso1:

Made the report.pdf and readme.txt files
Made load-json.py more efficient for loading large files

dek:

Made load-json.py
Made the ‘add an article’ functionality
Worked on the ‘List venues’ functionality

vinayan:

Made the ‘search for articles’ and ‘search for authors’ functionality
Worked on the ‘List Venues’ functionality

A github repository was created for all partners to access for updating the code they worked on. Everyone in the group met in person to work on the project simultaneously for easy coordination. All group members met to work on the project together and spent approximately 10 hours on the project each over the course of 3 days.