

Houses

Implement a program to import student data into a database, and then produce class rosters.

```
$ python import.py characters.csv
$ python roster.py Gryffindor
```

```
Lavender Brown, born 1979
Colin Creevey, born 1981
Seamus Finnigan, born 1979
Hermione Jean Granger, born 1979
Neville Longbottom, born 1980
Parvati Patil, born 1979
Harry James Potter, born 1980
Dean Thomas, born 1980
Romilda Vane, born 1981
Ginevra Molly Weasley, born 1981
Ronald Bilius Weasley, born 1980
```

Getting Started

Here's how to download this problem into your own CS50 IDE. Log into [CS50 IDE](https://cs50.net/2019/fall/psets/7/houses/houses.zip) and then, in a terminal window, execute each of the below.

Execute `cd` to ensure that you're in `~/` (i.e., your home directory, aka `~`).

If you haven't already, execute `mkdir pset7` to make (i.e., create) a directory called `pset7` in your home directory.

Execute `cd pset7` to change into (i.e., open) that directory.

Execute `wget https://cdn.cs50.net/2019/fall/psets/7/houses/houses.zip` to download a (compressed) ZIP file with this problem's distribution.

Execute `unzip houses.zip` to uncompress that file.

Execute `rm houses.zip` followed by `yes` or `y` to delete that ZIP file.

Execute `ls`. You should see a directory called `houses`, which was inside of that ZIP file.

Execute `cd houses` to change into that directory.

Execute `ls`. You should see a `characters.csv` file and a `students.db` database.

Background

Hogwarts is in need of a student database. For years, the professors have been maintaining a CSV file containing all of the students' names and houses and years. But that file didn't make it particularly easy to get access to certain data, such as a roster of all the Ravenclaw students, or an alphabetical listing of the students enrolled at the school.

The challenge ahead of you is to import all of the school's data into a SQLite database, and write a Python program to query that database to get house rosters for each of the houses of Hogwarts.

Specification

In `import.py`, write a program that imports data from a CSV spreadsheet.

Your program should accept the name of a CSV file as a command-line argument.

If the incorrect number of command-line arguments are provided, your program should print an error and exit.

You may assume that the CSV file will exist, and will have columns `name`, `house`, and `birth`.

For each student in the CSV file, insert the student into the `students` table in the `students.db` database.

While the CSV file provided to you has just a `name` column, the database has separate columns for `first`, `middle`, and `last` names. You'll thus want to first parse each name and separate it into first, middle, and last names. You may assume that each person's name field will contain either two space-separated names (a first and last name) or three space-separated names (a first, middle, and last name). For students without a middle name, you should leave their `middle` name field as `NULL` in the table.

In `roster.py`, write a program that prints a list of students for a given house in alphabetical order.

Your program should accept the name of a house as a command-line argument.

If the incorrect number of command-line arguments are provided, your program should print an error and exit.

Your program should query the `students` table in the `students.db` database for all of the students in the specified house.

Your program should then print out each student's full name and birth year (formatted as, e.g., `Harry James Potter, born 1980` or `Luna Lovegood, born 1981`).

Each student should be printed on their own line.

Students should be ordered by last name. For students with the same last name, they should be ordered by first name.

Walkthrough

Usage

Your program should behave per the example below:

```
$ python import.py characters.csv
```

```
$ python roster.py Gryffindor
Hermione Jean Granger, born 1979
Harry James Potter, born 1980
Ginevra Molly Weasley, born 1981
Ronald Bilius Weasley, born 1980
...
```

Hints

Recall that after you've imported `SQL` from `cs50`, you can set up a database connection using syntax like

```
db = SQL("sqlite:///students.db")
```

Then, you can use `db.execute` to execute SQL queries from inside of your Python script.

Recall that when you call `db.execute` and perform a `SELECT` query, the return value will be a `list` of rows that are returned, where each row is represented by a Python `dict`.

Testing

No `check50` for this problem, but be sure to test your code for each of the following.

```
$ python import.py characters.csv
$ python roster.py Gryffindor
Lavender Brown, born 1979
Colin Creevey, born 1981
Seamus Finnigan, born 1979
Hermione Jean Granger, born 1979
Neville Longbottom, born 1980
Parvati Patil, born 1979
Harry James Potter, born 1980
Dean Thomas, born 1980
Romilda Vane, born 1981
Ginevra Molly Weasley, born 1981
Ronald Bilius Weasley, born 1980
```

```
$ python roster.py Hufflepuff
Hannah Abbott, born 1980
Susan Bones, born 1979
Cedric Diggory, born 1977
Justin Finch-Fletchley, born 1979
Ernest Macmillan, born 1980
```

```
$ python roster.py Ravenclaw
Terry Boot, born 1980
Mandy Brocklehurst, born 1979
Cho Chang, born 1979
Penelope Clearwater, born 1976
Michael Corner, born 1979
Roger Davies, born 1978
Marietta Edgecombe, born 1978
Anthony Goldstein, born 1980
Robert Hilliard, born 1974
Luna Lovegood, born 1981
Isobel MacDougal, born 1980
Padma Patil, born 1979
Lisa Turpin, born 1979
```

```
$ python roster.py Slytherin
Millicent Bulstrode, born 1979
Vincent Crabbe, born 1979
Tracey Davis, born 1980
Marcus Flint, born 1975
Gregory Goyle, born 1980
Terence Higgs, born 1979
Draco Lucius Malfoy, born 1980
Adelaide Murton, born 1982
Pansy Parkinson, born 1979
Adrian Pucey, born 1977
Blaise Zabini, born 1979
```