Housing Project Specifications

Summary **1**

Files **1**

housing-info.csv 2

Fields 2

income-info.csv 2

Fields 2

zip-city-county-state.csv 3

Fields 3

Sample Data 3

NOTE 3

Database **3**

Fields 4

Requirements **4**

Program Location (5 points) 4

Required File (5 points) 4

NOTE 4

Program Execution (5 points) 5

Output (10 points) 5

NOTES 5

Correct Data (25 points) 6

Errors (-3 points each) 6

Cleaning Data **6**

Consultation **6**

Summary

You have been given a set of sample data, in three files, that contain housing data. You have been instructed to clean the data, and push that cleaned data into a database.

Files

You will be given four files. There are three CSV files and one SQL file. Each CSV file has a header row. The CSV files are:

housing-info.csv

This file contains information about housing in the western region of the United States.

Fields

- guid
    - Global Unique Identifier, this is an ID for each record
- zip_code
    - ZIP code
- housing_median_age
    - Median age for houses in that ZIP code
- total_rooms
    - Total number of rooms in that ZIP code
- total_bedrooms
    - Total number of bedrooms in that ZIP code
- population
    - Total population in that ZIP code
- households
    - Total number of households in that ZIP code
- median_house_value
    - Median value of houses in that ZIP code

income-info.csv

This file contains information about median household incomes for ZIP codes in the western region of the United States.

Fields

- guid
    - Global Unique Identifier, this is an ID for each record
- zip_code
    - ZIP code
- median_income
    - Median household income in that ZIP code

zip-city-county-state.csv

This file contains information about

Fields

- guid
    - Global Unique Identifier, this is an ID for each record
- zip_code
    - ZIP code
- city
    - City within that ZIP code
- state
    - State containing that ZIP code
- county
    - County containing that ZIP code

databaseCreationScript.sql

This SQL file will create the database you need to complete this task. It will create a database called **housing_project**. Inside that database, there will be a table called **housing**.

Sample Data

Sample data will be provided via github: https://github.com/kholm-umc/HousingProject.git

NOTE

Some of the data has been corrupted. That corrupted data will show up as a random four-character code. Corrupt data may have occurred in any field, in any record.

Database

Follow the directions in the SQL file to create the database.

Fields

```
+--------------------+-------------+------+-----+---------+----------------+
| Field              | Type        | Null | Key | Default | Extra          |
+--------------------+-------------+------+-----+---------+----------------+
| id                 | int(11)     | NO   | PRI | NULL    | auto_increment |
| guid               | varchar(32) | NO   |     | NULL    |                |
| zip_code           | int(5)      | NO   |     | NULL    |                |
| city               | varchar(32) | NO   |     | NULL    |                |
| state              | varchar(2)  | NO   |     | NULL    |                |
| county             | varchar(32) | NO   |     | NULL    |                |
| median_age         | int(3)      | NO   |     | NULL    |                |
| total_rooms        | int(5)      | NO   |     | NULL    |                |
| total_bedrooms     | int(5)      | NO   |     | NULL    |                |
| population         | int(5)      | NO   |     | NULL    |                |
| households         | int(5)      | NO   |     | NULL    |                |
| median_income      | int(8)      | NO   |     | NULL    |                |
| median_house_value | int(8)      | NO   |     | NULL    |                |
+--------------------+-------------+------+-----+---------+----------------+
13 rows in set (0.00 sec)
```

Requirements

Program Location (5 points)

Create a PyCharm project called Housing project. The main program should be called **main.py**

Required File (5 points)

Within your PyCharm project, create a file called **files.py**. This file will be imported from **main.py** by:

```
from files import *
```

Within the **files.py** file, there will be three assignments:

```
housingFile = "</path/to/file>"
incomeFile = "</path/to/file>"
zipFile = "</path/to/file>"
```

NOTE

`</path/to/file>` should point to your sample files. Prior to running your program, I will edit the **files.py** file to point at the files with the complete set of data.

Program Execution (5 points)

I will run your program within PyCharm, executing only the **main.py** file.

Output (10 points)

When I run your program I expect to see only the following:

```
Beginning import
Cleaning Housing File data
100 records imported into the database
Cleaning Income File data
100 records imported into the database
Cleaning ZIP File data
100 records imported into the database
Import completed

Beginning validation

Total Rooms: 111
For locations with more than 111 rooms, there are a total of
222 bedrooms.

ZIP Code: 33333
The median household income for ZIP code 33333 is 444,444.

Program exiting.
```

NOTES

Replace the `100` number with the actual number of records imported into the database.

Total rooms `111` should be an input. I will provide that number during runtime.

The result (`222`) should come from the database. This should be an integer. The SQL will be provided for that calculation.

ZIP Code `33333` should be an input. I will provide that number during runtime.

The result (`444,444`) should come from the database. This should be an integer rounded to the nearest dollar. Be sure to place a comma after the thousands place. The SQL will be provided for that calculation.

Spelling, grammar, and punctuation are important. Copy and paste from above.

Correct Data (25 points)

I will look at the numbers from **NOTES** section above. You will lose points for each of these numbers where the actual answer is incorrect.

Errors (-3 points each)

For every uncaught exception, spelling, grammar, and punctuation error you make will accrue a 3 point deduction. This only applies to program output.

Cleaning Data

When your program runs across corrupt data, you should attempt to correct it. For each field, use the following rules:

- guid
  - Drop the entire record completely
- zip_code
  - Assign a new ZIP code to that record. The ZIP code should begin with the same digit as a close by city with the last four digits being zeros. i.e., If you correct a ZIP code for San Francisco, it should be 90000.
- housing_median_age
  - Generate a random number between 10 and 50.
- total_rooms
  - Generate a random number between 1,000 and 2,000
- total_bedrooms
  - Generate a random number between 1,000 and 2,000
- population
  - Generate a random number between 5,000 and 10,000
- households
  - Generate a random number between 500 and 2,500
- median_house_value
  - Generate a random number between 100,000 and 250,000
- median_income
  - Generate a random number between 100,000 and 750,000

Consultation

I will be available for consultation in class, via Slack, email, or phone.