Practical Assignment 4.1-4.4 – Creating a Database, Data Conversion, and Import

In this course, we have learned how to analyze data and the processes supported by data, and develop logical and physical designs. In this exercise, we will use real data from a public data source, attempt to analyze that data, and devise an appropriate schema for this data. In addition, DBA's are typically responsible for converting and importing raw data to databases as part of the overall process to on-board new databases.

As part of this exercise, you will find four tutorial videos which will help you through each of the steps in the process, and explains the deliverables to be submitted as part of this assignment. In addition, there are instructions below which outline the steps demonstrated in the video.

There are also several attachments with this assignment, which contain files you may need for this exercise. In addition, the source for each of these files is fully explained in the videos.

Video One: (INFO-365 Practical Assignment 4: Creating and Importing a Database Part 1)

1. Follow the instructions to learn how to use the PowerShell to connect to your Linux server using SSH (PowerShell has a native SSH client, so you will not need PuTTY. Note: Windows 7 users will need to use PuTTY, as only Windows 10 includes SSH. Apple users should be able to use the SSH command in the same way demonstrated using the console/command line).
2. Google "Open Government Data", or use the URL http://data.gov
    a. Find the College Scorecard data page: https://catalog.data.gov/dataset/college-scorecard
    b. Download the "Raw Data": https://ed-public-download.app.cloud.gov/downloads/CollegeScorecard_Raw_Data.zip
    c. Download the "Data Dictionary", which describes the data in the raw CSV files:
        i. https://collegescorecard.ed.gov/data/documentation/
        ii. https://collegescorecard.ed.gov/assets/CollegeScorecardDataDictionary.xlsx
    d. Open the CSV File, and the supporting documentation to understand the data provided, and begin thinking about the schema to support this, and a strategy to develop the schema (more on this later).
3. For this scenario, we will assume that we are building an application which needs some supporting tables for authentication. This information is provided in the Users_Table.xlsx; we will analyze this file, and create a schema and import this data using utilities for MariaDB and MySQL Workbench:
    a. Open the Users_Table.xlsx
    b. To create the user table, and populate with data:
        i. Create a tab delimited text file called APIUsers.txt from the Users tab, and include the following columns:
            1. ID, UserID, UserName, UserKey, UserPassphrase
            2. (You will need to add the ID column, and populate with a value).
        ii. Create a CSV called APIUsersDetail.csv, and include the following columns:
            1. ID, UserID, Phone, Email

iii. Use the “`scp`” utility to copy the file to your Linux server hosting MariaDB.

iv. Login to your MariaDB server using the `mysql` client utility, and create a database called “CollegeScorecard”:

v. Using the `mysql` client utility, and create a table with the following specifications:

vi.

1. Table Name: APIUsers
2. Columns:
   a. ID: not null, primary key, auto increments.
   b. UserID
   c. UserName
   d. UserKey
   e. UserPassphrase
3. Use the mysqlimport utility to import the table contents from the tab delimited file.

vii. Use MySQL Workbench to both create and import data using the APIUsersDetail.csv

viii. Submit a screen capture image including both tables, one in MySQL Workbench, and the other with the `mysql` client on the server. Be sure to include your name in each screen capture image.

4. Scripting to create database:

a. OPTIONAL: Install Notepad++ (in Windows) with the NppFTP plugin, to more easily work the text and scripting files on your server (alternatively, you can use the VI text editor in your Linux server, or work with the files on your own computer, and copy to them to/from the server with the “scp” command).

b. Install PHP and PHP-MySQL on your Linux server using the yum command.

i. We will be using PHP as a scripting language to work with the CSV files, to script the create table and insert SQL scripts form the files we download/prepare from Data.gov.

ii. Test your PHP installation by creating a php file with the contents “`<?php echo phpinfo(); ?>`”. Use the command “`php <filename>.php`” to test to ensure PHP is working.

1. We will execute our PHP scripts from the command line; note that we are not using PHP as a web scripting language, just to manipulate and work with the text files in the console. This task could also be done using many other scripting utilities, such as Python, Perl, BASH, etc.

c. Prepare the CSV files that contain the data dictionary, which we will convert to a database schema:

i. Create a CSV called “CollegeScorecard_OtherTables.csv” with all the columns in the Data_Dictionary tab. You will also need to add a column called ID, with a value for each row.

ii. Follow the instructions in the video to remove unneeded rows… this CSV will only include the values we need for the supporting tables.

5. Continue to Video Two: (INFO-365 Practical Assignment 4: Creating and Importing a Database Part 2)
    a. Copy the “CollegeScorecard_OtherTables.csv” to your Linux server hosting MariaDB
    b. Follow the instructions to create a PHP script, which will read the contents of the CSV, and generate SQL code to create the tables needed and the insert statements to populate these tables with data.
        i. Note: the source for all the PHP code presented is included in the assignment files. You will need to make any changes for your file names/paths that may be different from my example.
    c. Follow the instructions to run the create statements, and the insert statements.
    d. Submit a screen capture showing the output of the “SHOW TABLES” command, which should show the supporting tables. Also include a screen capture showing the table that contains religious affiliation information. Be sure to include your name in all screen captures.
6. Continue to Video Three: (INFO-365 Practical Assignment 4: Creating and Importing a Database Part 3)
    a. Create a CSV called “CollegeScorecard_DataDictionary.csv” with all the columns in the Data_Dictionary tab. You will also need to add a column called ID, with a value for each row.
    b. Use the “scp” utility to copy the CSV to your MariaDB server.
    c. Follow the instructions to create a PHP script, which will read the contents of the CSV, and generate SQL code to create the tables needed and the insert statements to populate these tables with data.
        i. Note: the source for all the PHP code presented is included in the assignment files. You will need to make any changes for your file names/paths that may be different from my example.
    d. Follow the instructions to run the create statements, and verify the tables have been created.
    e. Combine your “create_other_tables.sql” and “create_tables.sql” scripts to a single file, then follow the instructions for using MySQL Workbench to create an EER diagram from the SQL script.
        i. Submit a PNG file of your completed EER diagram.
7. Continue to Video Four: (INFO-365 Practical Assignment 4: Creating and Importing a Database Part 4)
    a. Copy the CSV containing the raw data for the College Scorecard in 2016 to your MariaDB server.
    b. Follow the instructions to use PHP to create and run the insert statements to populate the various tables in the CollegeScorecard database.
    c. Submit a screen capture showing the results of the following two queries:
        i. All institution names with their religious affiliation.
        ii. Information about Drexel University.