ECO 5445: Introduction to Business Analytics
Department of Economics
College of Business Administration
University of Central Florida
Fall 2019

Assignment 2

Due Tuesday, October 1, 2019 at 2:59 PM
in *your* fork of the ECO5445F19 GitHub repo.

Instructions:

Complete this assignment within the space on your private mirror of the ECO5445F19 GitHub repo in the folder `assignment_02`. Create a folder called `my_answers` that will contain all of your work for this assignment. When you are finished, use `git` to `add`, `commit` and `push` your code to your private mirror of the ECO5445F19 repo. You are free to discuss your approach to each question with your classmates but you must `git push` in your own work.

Question 1:

Create a shell script that copies the required `.csv` files from the `pgbookData` folder to populate your workspace with the required datasets. Use the database `AuctionsDataBase.db` to pull new datasets from those tables as described below. The outputs are requested to provide information for the accounting department of the auctioneers. Their accounting system is down and they have called you to calculate the information direct from the database, using SQL queries. Save your queries and resulting tables for this question in a folder called `question_01` under your main folder `my_answers`.

a) Write a SQL query that calculates the highest bid for each auction and stores it with the associated AuctionID.

b) Write another query that collects the AuctionID, Date and highest bid for each auction, along with the BidID and BidderID from the winning Bid. You can build on your answer to part (a) by embedding that query into this one.

c) Make a similar table that joins these fields with the volumes from the auctions to calculate the total revenue for each auction. Then output the AuctionID, Date, TotalRevenue and BidderID for each auction. This list will help accounting reconcile the bank statements.

d) Extend this by tabulating the TotalRevenue by fiscal quarter. For financial statements, only the Quarter (YYYYQX form, X from 1-4) and TotalRevenue are required.

Question 2:

This exercise continues the activity in Question 2 using `AuctionsDataBase.db` by adding a new table to pull new datasets from the tables. Save your queries and resulting tables for this question in a folder called `question_02` under your main folder `my_answers`.

The bidders are busy executives and they each have representatives, which are listed in a spreadsheet `BidderReps.csv`. The auctioneers would like to contact the representatives of the winning bidders to arrange delivery and payment. An efficient way to achieve this is to create some form letters and invoices populated with the appropriate fields. You can help them by creating some text files with the required fields.

Begin by copying the spreadsheet `BidderReps.csv` to your own folder.

a) Create a schema for the new table `BiddersReps` and save it in a script that you will call `CreateBidderRepsTable.sql`. Then you will need to import the data from `BidderReps.csv` into the `BiddersReps` table.

b) The next step is to create a text file of the fields for an invoice to be presented on delivery. For each auction, this will require the BidderID, the representative's name and full address, the auction date and auction ID, along with the volume, price and balance owing. Embed your query from Question 2, part (c) into another query that joins the names and addresses of the bidders' representatives with the information from each winning bid. Save this query as `BiddersRepsDeliveryList.sql` and the resulting output as `BiddersRepsDeliveryList.out`.

c) Finally, the transactions are completed by collecting the payments from the representatives. Modify the above query to calculate the total balance owing from each representative, so that payments can be settled over a single transaction with each. There is no need to specify the details of individual auctions, only the contact information and balances owing. Save this query as `BiddersRepsBalanceOwing.sql` and the resulting output as `BiddersRepsBalanceOwing.out`.

Question 3:

In this exercise, you will design your own database to organize information contained in a list of spreadsheets. The database consists of information about students and employees at a major university you might be familiar with. Save your queries and resulting tables for this question in a folder called `question_03` under your main folder `my_answers`.

a) Begin by getting familiar with the contents of the spreadsheets in the folder `question_04_data` (after copying them to your own folder, of course). Form an entity-relationship diagram as in page 65 of the textbook and save an image of it in the folder for future analysts to understand the structure of the database. A photo of the whiteboard or notepad will do fine.

b) Create a schema for each of the tables, each named `<TableName>`, to match the file names `<TableName>.csv` and save them in a files called `Create<TableName>Table.sql`. Then import the data from the `.csv` files into each of the tables.

c) Some administrators at the Institute for the Prevention of Student Overtiredness are concerned that students' academic performance may suffer if they work while studying. To help these students, you can provide a list of all full-time students who also work on campus, along with their name, NID and GPA.

d) The Centre for the Investigation of Student Connectivity is interested in documenting student and employee involvement in coursework, employment and extracurriculars. Create a list with NID, first name and last name for each member of the university community, along with their employer and job title, extracurricular group and position, and course codes.