CSC 355 Database Systems 502
Assignment 2 (1/16)

Due 11:59:00pm, Thursday 1/24.

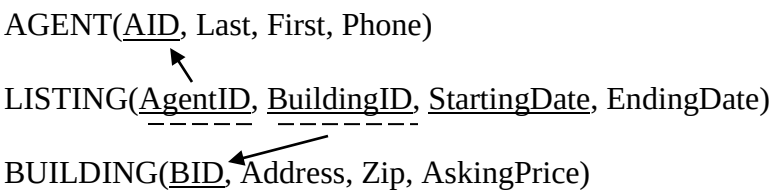
Reading: The posted Lecture 3 and Lecture 4 Slides, and Ullman/Widom Sections 2.3, 6.5, 7.1-7.3, and 6.1. [Next week: Ullman/Widom Sections 6.2-6.4.]

Your task in this assignment is to write a script to create a small database (three linked tables).

Steps:

Write a script to do the following. (Do this one part at a time, testing the partial results by inspecting the tables in SQLDeveloper. Do not go on the next part until you have the previous part working.)

1. Define the following database, representing real estate agents, buildings for sale, and the listings that associate agents with the houses they are trying to sell.



IDs of agents should be exactly three characters long, IDs of buildings should be exactly four characters long, zip codes should be exactly five characters long, and phone numbers should be exactly ten characters long. You may decide on appropriate maximum lengths for names and addresses. Starting and ending dates of listings should be stored as DATE objects, not strings. The asking price for each building must be a positive number (use a CHECK constraint to insure it is positive) that contains up to eight digits to the left of the decimal point and two to the right.

Define the needed primary and foreign keys in each relation. (Note that you will have to create AGENT and BUILDING before LISTING.)

In order to avoid conflicts, start your script file with DROP TABLE commands for all three tables (since LISTING contains foreign keys, you will have to drop it before AGENT and BUILDING). Run your script and look at the columns and constraints of each table to verify that they have been created correctly before going on. Now you can set up yourself and a couple of friends in the real estate business.

2. Populate the BUILDING table with addresses, zip codes and asking prices for four buildings for sale in the city of your choice. (Go to https://www.realtor.com/realestateforsale and type in a zip code or city and state of your choice to find some buildings.) You can make up the IDs for the buildings in the table. Then populate the AGENT table with information on you and two of your friends. (The phone numbers, and the friends, may be real or imaginary.) Again, you can make up the IDs. Look at the data in each table to verify that they have been populated correctly.

3. Next, insert at least five rows into the LISTING table. There should be at least one row for each building, and at least one building should have two or more rows in the LISTING table. Be sure that

each building is listed with only one agent at a time – that is, a building's listing with a second agent cannot begin until after its previous listing with another agent ends. (This particular constraint – insuring that listings do not overlap – is too complicated for us to write code to enforce right now.) Every record in LISTING must have a starting date and an ending date, and the ending date should be later than the starting date. (Again, we have not seen how to implement this constraint yet.) Look at the data in the table to verify that it has been populated correctly.

4. Display the contents of each table by adding three commands of the form SELECT * FROM *TABLENAME* ; to the end of your script file. Run the complete script and save the complete contents of the output window, including the displayed tables, to a text file. (Clear the window before running the script for the last time so that only the output of the last run of the script is displayed and saved.)

5. Include a comment at the top of your script file giving your name, the course number and your section number, the assignment number, and the date of submission, e.g.:

```
/*
YourName
CSC 355 Section 502
Assignment 2
SubmissionDate, 2019
*/
```

6. There are two files to submit for this assignment. Submit both (1) your .sql script file and (2) the text file containing the output generated by running your script to the Assignment 2 dropbox.

Remarks:

1. There are in fact tools within SQLDeveloper that allow you to construct tables without writing your own CREATE statements; they can also automatically generate SQL code that would build the tables you have constructed using the tools. Do <u>not</u> use these table creation and code generation tools to complete the assignment; you should write the CREATE statements to create the tables yourself and include the CREATE statements that you have written in your script.

2. It is your responsibility to make sure that the files you have uploaded are readable and in the correct locations. You should check that you can successfully download them back to yourself from the submissions dropbox after submitting them to be sure that they have been uploaded correctly.

3. Please remember that all work must be completed individually and without copying, either entirely or in part, from any examples I have posted or anyone else's work.

Eric J. Schwabe – 01/16/19