DEPARTMENT of COMPUTING
College of Business & Technology
EAST TENNESSEE STATE UNIVERSITY

CSCI 5000 – Data Management

Project 2 – Relational Database Creation and Import

Overview

During project 1, you worked with Excel data and did a few formulas to provide information about farmer's markets. When working with Excel, some formulas, like VLOOKUP or SUMIF, can get quite complex with a large data set. Not only that, your personal computer becomes more of a limitation as Excel file sizes increase due to memory constraints. In this project, you will take data from Excel (really, CSV files) and convert it to the relational format by (1) creating relational tables in SQL, and (2) importing the existing data into those tables.

Project Goals

1. Analyze the structure of existing data to determine a basic database design.
2. Interface with an Oracle database using Oracle SQL Developer.
3. Write SQL to create relational tables within Oracle.
4. Import existing data into newly-created tables.

Project Preparation

To complete this project, you should install Oracle SQL Developer. You may download the software for free from https://www.oracle.com/tools/downloads/sqldev-downloads.html. (Note that you may need to create an Oracle account, but the software is free.)

1. From D2L, download the following CSV files.
    a. Person.csv - A file containing information about students and professors.
    b. Section.csv - A file containing all course section information for a particular semester.
    c. Enrollment.csv - A file containing student enrollments in course sections for a semester.

2. Save these files in a well-known location so that you can refer to them during the project.

Part 1: Exploring and Diagramming the Data

In the first part of this project, you will explore the data and prepare the data for import into a database. As a result of your exploration, you will create a basic Entity-Relationship Diagram that allows you to show how the data are related.

1. Visit the Enrollment CSV file and compare it to the Section CSV file. Notice that the Enrollment CSV uses a "Section_ID" field, but the Section CSV has three separate fields: Subject, Number, and Section Number.
    a. Use Excel's Text-To-Columns feature to separate the text into three separate fields. You should add two blank columns after the Section_ID field.
    b. Be sure to change the column headings to indicate appropriate titles.
    c. Save the Enrollment CSV file (do not convert it to Excel format; it should remain in CSV format).
2. Visit diagrams.net (www.diagrams.net), which is a free online drawing tool. You may need to sign up for an account so you can save diagrams you create. You can also download versions of the software for Windows, Mac, Linux, and Chromebook.
3. Draw a basic Entity-Relationship Diagram that represents the CSV files as entities.
    a. Draw each entity as a box that lists the Entity name followed by the list of attributes describing it.
    b. Draw lines between entities that are related. Label the lines with the name of the relationship.
        i. Speculate on the cardinality. If a maximum of one instance can participate in the relationship, draw a straight line. If many instances can participate in the relationship, draw a crow's foot.

ii. There is no need to consider the modality.
iii. **See the supplemental videos on D2L in the Discovery Project 2 folder for assistance in completing this part.**

4. Export your diagram as a PNG file (which is an available option in the File→Export As menu). Be sure to select the Diagram option with Size). Name the file **Project2.png**.

Part 2: Creating the Database Tables

Now that you have an idea of the data's structure, you can create scripts that will build the database tables.

Create a file named Project2.sql that contains the SQL statements that build the database tables you drew in your ERD. For each table you create, your script must do the following:

1. Give the tables an appropriate name,
2. Give each attribute an appropriate name and an appropriate data type,
3. Formally set the primary key for each table, and
4. Formally set up foreign keys if needed.

Be sure you are connected to the Pythia server using SQL Developer. Run the creation statements. Resolve any errors. Note that, after the tables are created, you cannot simply recreate them. You must first drop the existing tables from the database before running the create statements a second time. If you need to drop the tables and re-create them, you may run the following command:

DROP TABLE <table_name> CASCADE CONSTRAINTS;

For example: DROP TABLE enrollment CASCADE CONSTRAINTS;

Part 3: Importing the Data

Using the Import tool in SQL Developer, import the data into your tables. You may need to do this in a specific order to ensure the data is imports are successful. To import data:

1. Right-click the table where you want to import data. Select Import Data…
2. On the Data Preview screen:
    a. Browse to the Import Data File and select the CSV file that corresponds to this table.
    b. Review the area under File Format and choose the appropriate options.
    c. Click Next.
3. On the Import Method screen:
    a. Select Insert as the Import Method.
    b. Be sure the Table Name matches what you expect.
    c. Do not select an Import Row Limit.
    d. Click Next.
4. On the Choose Columns screen:
    a. Choose the columns that you want to import.
    b. Click Next.
5. On the Column Definition Screen:
    a. Select each Source Data Column (the columns selected out of the CSV).
    b. For each source column, set the appropriate Target Table Columns information.
    c. Complete this for all Source Columns and click Next.
6. On the Finish Screen:
    a. Review the selected options in the Import Summary.
    b. Click Finish.

An import might fail for a couple of reasons, so here are some basic things to watch:

1. Importing a file in the wrong order. Foreign keys require you to import a file that is "pointed to" before a file that points to it.
2. Column mismatches. Be sure your columns are correctly matched in your import statement. If not, the import could fail due to data type mismatches.
3. Incorrect data types or precisions in your database tables. If you chose a data type that is not compatible with a column you're importing—OR if your data field is too small, the data might not import. In this case,

you should revisit Part 2 and adjust your create script for the particular table, drop your tables, and recreate them.

If the import ran successfully, you might still have data problems. It is best to browse the data at this point to see whether things imported correctly. To browse the data, left-click the table's name in your Tables list. Select the Data tab and scroll through the data to look for any inconsistencies.

Part 4: Reflection

This portion counts as Journal 3-2. Include your reflection in a **Journal3.2.docx** document. Be sure to include:

1. A reflection of your experience learning through this activity.
2. A reflection of the importance of learning this material.

Project Submission

Submit the following work to the **Project 2 Dropbox**.

1. **Enrollment.csv** – your updated Enrollment file after completing the text-to-columns operation.
2. **Project2.png** – the entity relationship diagram you created, based on the three provided CSV files
3. **Project2.sql** – the SQL file that contains the CREATE script for each of the database tables developed from your ERD.

Submit the following work to the **Lab Journal 3.2 Dropbox**.

1. **Journal3.2.docx** – the reflection questions from Part 4.

The suggested deadline to maintain pacing is Monday, October 4, 2021.

Grading Rubric

Project Activity Grading Rubric

Criterion	A-Level 10 Points	B-Level 8 Points	C-Level 6 Points	D-Level 4 Points	F-Level 2 Points	No Attempt 0 Points
Part 1 – Data Exploration						
1.1 – Text-to-columns in Enrollment.csv						
1.2 – Entity Relationship Diagram						
1.2.1 – Correct Use of entities						
1.2.2 – Correct attributes						
1.2.3 – Correct relationships						
1.2.4 – Correct Cardinalities						
Part 2 – Database Creation						
2.1 – CREATE scripts run correctly.						
2.2 – CREATE scripts include appropriate data types.						
2.3 – CREATE scripts define primary keys						
2.4 – CREATE scripts define foreign keys as appropriate						
Part 3 – Data Imports						
3.1 – Each table imported successfully						

Reflection Grading Rubric

Criterion	Excellent	Good	Average	Below Average	Poor	No Attempt
Activity	10 points Completes all aspects of the activity correctly	8 points Completes most aspects of the activity correctly	6 points Completes aspects of the activity correctly, and some incorrectly	4 points Completes most aspects of the activity incorrectly, or does not attempt many aspects	2 points Minimal effort or completes few aspects of the activity or very few correctly	0 points Did not complete the activity
Reflection	10 points Reflection clearly ties to the module content; experience and importance	8 points Reflection mostly ties to the module content; experience & importance	6 points Reflection ties minimally to the module content; experience & importance are discussed	4 points Reflection does not tie to the module content; experience & importance are minimally discussed	2 points Minimal effort to tie to content; minimal effort to describe experience/ importance	0 points Did not complete the reflection

	clearly laid out	are discussed	but not thoroughly			