GST 103: Data Acquisition & Management Lab Series

Lab 2: Basic Geodatabase Design

Document Version: 2013-07-31 (Beta)

Organization: Del Mar College
Author: Richard Smith

The development of this document is funded by the Department of Labor (DOL) Trade Adjustment Assistance Community College and Career Training (TAACCCT) Grant No. TC-22525-11-60-A-48; The National Information Security, Geospatial Technologies Consortium (NISGTC) is an entity of Collin College of Texas, Bellevue College of Washington, Bunker Hill Community College of Massachusetts, Del Mar College of Texas, Moraine Valley Community College of Illinois, Rio Salado College of Arizona, and Salt Lake Community College of Utah.

Contents

Introduction

This lab is part of a series of lab exercises designed through a grant initiative by the National Information, Security & Geospatial Technologies Consortium (NISGTC), funded by the United States Department of Labor in partnership with the Department of Education under the Trade Adjustment Assistance Community College and Career Training Grant Program (TAACCCT).

Data importing and exporting is a large component of building and managing a geodatabase. The size and number of files in a geodatabase are typically quite large and difficult to move all at the same time. We also run into problems ensuring that a recreated database will have all the same options as the previous database. Exporting the schema, or database structure, is an important method used to help resolve these issues. Exporting a schema is essential to database migration and can be used to create a backup of our geodatabase. Schemas are also often shared between GIS users for various purposes.

Another important skill to have when managing your GIS is to put together a functional and efficient database design. Part of a good design will require tables with established relationships that will maintain their integrity even when changes are made. This often requires the use of a relational database, which employs a relational structure.

Your instructor may require that you provide screen captures, exported files and database designs (as directed in the lab). Please check with your instructor for the requirements specific to your class.

This lab includes the following tasks:

1. Database Schema Import and Export
2. Designing a Basic Relational Database

Objective: Database Structure

In this lab, we will look at maintaining the database structure while it is migrated and then migrate the data, once the database schema (structure) is in place. We will also be looking at an example of how relational tables work.

Lab Settings

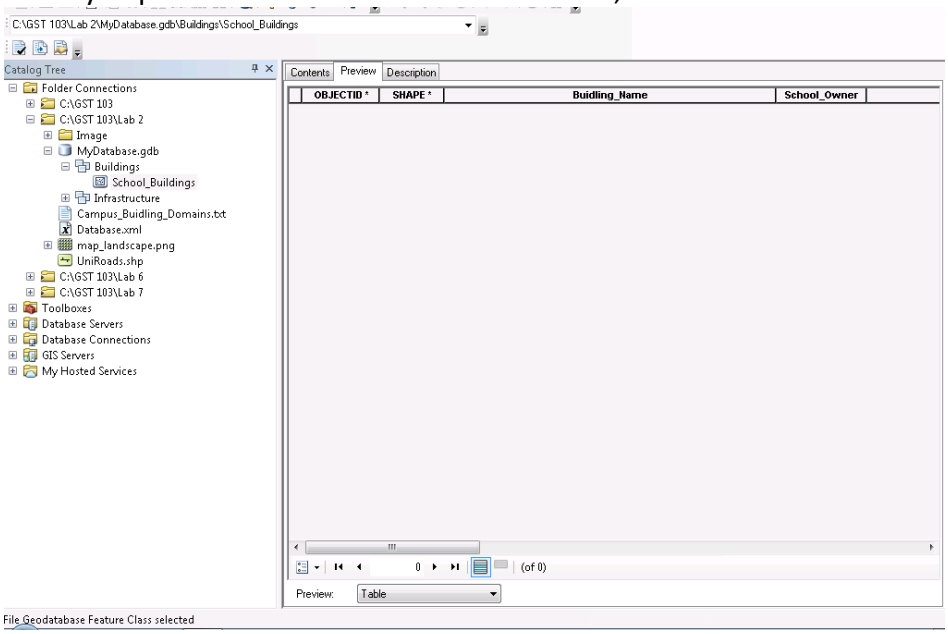
Required Virtual Machines and Applications

Windows Machine User Account	Train
Windows Machine User Password	Train1ng$

1 Database Schema Import and Export

This task will illustrate the value of being able to store and restore a database piece by piece. We need to preserve the structure of the database at all times to maintain the integrity of our data. Our product is only as good as the data we use and maintain.

1. Log into the computer using the information provided in the Lab Settings section.
2. Click **Start->all Programs->ArcGIS->ArcCatalog 10.1** this will open ArcCatalog.
3. Open up your *Lab 2* folder in the Catalog Tree.
4. Create a new file geodatabase in your *Lab 2* folder and name it **`MyDatabase`**.
5. **Right-click** on your newly formed database and select **Import->XML Workspace Document**. The wizard will pop up. Change the radio button to **Schema Only**. This will allow us to import only the tables and the structure, the skeleton of the database. Next, select your source to import by browsing to the Database.xml file in the *Lab 2* folder. This XML folder contains the structure of the database and the database can be rebuilt from this file. Click **Next**, then **Finish**. Two feature datasets should appear.
6. Expand the feature dataset titled **Buildings** and select the feature class **School_Buildings**. Next, click on the **Preview** tab on the display and you will notice a blank screen if the Preview mode is set to **Geography**. Now, change the Preview mode (below the display area) to **Table** and you will notice a table with headers but no data (see screenshot below). This tells us that we have successfully imported the structure of the database, but not the actual data.



7. Next, import the raster image called **map_landscape.png** as well as all the images in the folder called **Image** into your geodatabase. Look at the file types to figure out what type of import to do.
8. Import the **UniRoads.shp** shapefile into the feature dataset called Infrastructure. Name the output **Roads**.

9. Import the ***Campus_Building_Domains.txt*** as a Table (single) into the database. We will be using it at a later stage. We will look at domains in the next lab.
10. Export your completed geodatabase as an XML dataset by **right-clicking** on the geodatabase, clicking **Export->XML Workspace Document**. This time select the **Data** radio button. Save the xml document in your *Lab 2* folder as **`Export.xml`**. Export your database as a XML workspace again, but now select the **Schema Only** radio button call it Schema.xml. In order to see these datasets in ArcCatalog after you have created them, you will likely need to close ArcCatalog and reopen it.
11. Next, create a new file geodatabase and call it **`Data`**. Then, import the **Export.xml** file into your new database. Use the **Preview** tab in the Display window to see that both the structure and the files have imported.
12. Finally, create a third new file geodatabase and call it **`Schema`**. Then, import the ***Schema.xml*** file (be sure to select **Schema Only** when importing) and observe the differences between your Schema database which contains only the database structure and your Data database which contains both the structure and the data (remember, your School_Buildings shapefile never had data in it so preview the other files to see the differences)

2 Designing a Basic Relational Database

This task will look at relational table design and how we model relational database structures. This will allow you to see the value of relational database design.

Scenario:

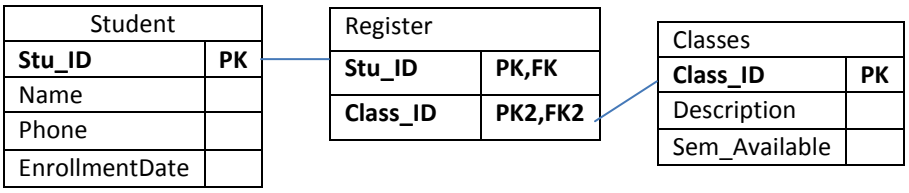
Design a relational database that models student enrollment data. This should be a simple model, indicating the classes in which a student is enrolled. A student may be enrolled in many classes and a class may have many students. Create an initial database design. You do not have to load it with data. State your assumptions as you go.

The scenario above is a typical situation requiring a database design. Even though the data being modeled is not geographic, it follows the same concepts.

Let's look at the scenario step by step. In the second sentence, we have identified our first 2 tables: *Students* and *Classes*. Next, we look for keywords describing the relationships. Students may be enrolled in many classes and classes may have many students. So, students may have many classes. This gives us a many-to-many relationship either way. We denote these relationships as (*M: N*). We will need to devise a way to link the tables and maintain the many-to-many relationship. Using a relational table, we can have the information from both tables displayed, giving us a solution and breaking down the many-to-many relationship.

Let's look at the attributes we may use and decide on our primary keys. We will need some basic information on students, such as student ID, student names, phone number and date of enrollment. Also, some information about classes, such as course code, description, and the semester it's available.

Now, we need to model the data. Using a many-to-many relationship is inefficient and will make the efficiency and integrity of a database pointless, so we will use a relational table that stores the student id and the class id. This table will act as a bridge between the tables, providing a suitable relationship. Let's call the table *Register* and build it so we can see the relationship. The relationships are built by the primary keys (PK) and foreign keys (FK).



This is the basic design for the database. Databases do get more complicated than this; however, we need to start somewhere.

Consider the next scenario. This time, you will come up with the database design.

Scenario:

You are tasked with building a database containing flight information. This database keeps track of flights, pilots and the planes to which they are assigned. Each flight has a unique flight number, departure airport and a destination airport. Each pilot has a unique employee id, name and flight hours. Pilots are assigned to flights and flights may have numerous pilots. Design a relational database model based on this information. (Hint: The primary keys are the unique values.)

Submit your database model to your instructor in an Excel spreadsheet or Word processing file.

Conclusion

In this lab, we have looked at how to store information in a database and how to ensure that the database can be transferred from one place to another without losing options available in the original database. We also looked at the relational database structure and how it works with some basic modeling.

Discussion Questions

1. Why do we use relational databases?

2. Why is the database schema so important to the database design and data storage?

3. Is it better to build a database to fit the data or build a database and manipulate the data to fit the database?

