Using Oracle/SDE to Teach ArcGIS in a Windows Active Directory Environment

Jerry Maedel, Michael J. Meitner, Ionut Aron
Collaborative for Advanced Landscape Planning
Department of Forest Resources Management
Forest Sciences Centre
2045-2424 Main Mall
University of British Columbia
Vancouver, BC., Canada V6T 1Z4
jerry.maedel@ubc.ca, mike.meitner@ubc.ca, ionut.aron@ubc.ca

Abstract

This paper explores the benefits and drawbacks of teaching ArcGIS using data in an Oracle/ESRI Spatial Database Engine (SDE) environment while operating within a Microsoft Windows domain. A discussion of the process required to setup class access to Oracle/SDE is followed by recommendations for making the process more effective. Special emphasis is placed on using Oracle/SDE for improving the evaluation process of ArcGIS student projects.

Keywords: education, geodatabase, GIS, laboratory instruction, Oracle SDE, working data sets

1. Introduction

Along with the conceptual challenges of teaching spatial analysis are the practical difficulties of providing students with working data and then collecting and evaluating the results of their GIS analyses. Working data sets can often be quite large, especially when combined with aerial photographs. Distributing this data over a computer network can be a slow process and typically demands a lot of hard disk space on the client side when used by a class of thirty to forty students. Not only are the technical difficulties of distribution and storage an issue but even more challenging is troubleshoot the myriad of problems resulting from student confusion. By forcing students to manage their own data sets through out the course of their analysis we have noticed several common problems that arise year after year. Students often become confused as they generate multiple copies of their data. This can lead to students overwriting current data with older versions, opening an older version and panicking when they see that all of their work is gone or simply giving up because of their frustrations. In addition, tracking down lost or corrupt data are also common practical problems faced by a GIS instructor.

Through our experiences a number of solutions to the distribution and submission of data have been undertaken with various levels of success. Perhaps the simplest was to create a personal geodatabase with relative path names and distribute it from a server as a read-only file. This works well for introductory students to view data, conserving resources and allowing for a consistent data set for all students. Any editing or creation of new data requires students to copy data to their own drives creating a great deal of redundant data and problems with submitting final data for evaluation. No matter what the distribution media, multiple copies of data have always been difficult to manage and evaluation of the assignments can be difficult due to non-standard naming conventions adopted by the student, broken links to data, etc.

Recently we have decided to use a relational database with ESRI's Spatial Database Engine (ArcSDE) installed on top, which has provided an elegant solution to many of the aforementioned problems. In this configuration, students all access the same geodatabase yet are able to create their own editable SDE version without creating any redundant data. Best of all, students submitted their final analysis as a small MXD file no matter how much data was referenced. Instructors can open the MXD file from any location that has access to the Internet and evaluate student projects without having to copy any data. Each student dataset was a separate, secure entity, yet all students used the same base data.

This paper will describe the technology required to achieve these results and then lead the reader through the process of setting up an Oracle database for students to use. Detailed scripts will be provided for the reader to create and delete student accounts in Oracle.

2. Technical Requirements

This article assumes that a computer environment has been created that has the following configuration. Windows 2003 server with Active Directory. Students logging on to the Active Directory domain with ArcGIS 9.1 installed on their workstations and a common shared drive on the server that students can access. A Windows 2003 server with Oracle 9i and ESRI's ArcSDE 9.1 installed.

3. Personal Geodatabase

For those without access to an Oracle server, we found that using a read only, personal geodatabase, copied onto a shared network drive, worked reasonably well for demonstrating introductory concepts that did not require the creation of new data. If new data was required as a part of the GIS lab, students copied a writable version of the personal geodatabase to their local drive or writable space on the network.

This approach worked well for the display and querying of data as all students used the same dataset which could be setup with all the symbology and layers required. However, it does present certain challenges for the creation and marking of projects. Students working in groups were constantly mixing up who was working on what, corrupting the MXD file and loosing data. Marking was also difficult as the data had to be transferred to CD or to a 'Dropbox' on the network which was also plagued with corrupt and missing data. A shared drive on the network, the 'Dropbox' was setup so that students could write to the drive but not see its contents. Instructors had read, write, list, and modify rights.

Three different methods were used depending on the level of access required to the data: read only; read only and create new features; read write.

Read Only
A dataset was created and all the symbology and layers arranged to minimize the amount of time a student had to spend getting started on the lab. The data files were copied by the instructor to a network shared drive available to all students and set to read only. All students accessed the same MXD file to load the same data, so data storage space on the network was minimized.

Read Only & Create New Features
The same process as above was followed except students saved a copy of the MXD file to a writeable drive. This only created a small increase in storage space requirements. Students could now save any changes they made to their map and could also create new data and save it as a part of their map document. They did however have to create their own personal geodatabase locally or create shape files in order to edit data. Students could not edit data on the read only drive.

Read Write
Students copied a read write version of the personnel geodatabase and the MXD file to their own writeable space. They could now edit existing data and create new features in the geodatabase already created. This produced a fairly significant increase in network traffic while copying the geodatabase, and in the total amount of storage space required.

4. Multi-user Access – Oracle and ArcSDE Versioning

In order to overcome the problems of having multiple student copies of GIS data we migrated our personal geodatabase to an ArcSDE / Oracle relational database. There are a number of relational databases supported by ArcSDE but we already had an Oracle license. This solution allows students access to a common repository while creating and saving their own data in a separate version of the database. Only they would be able to see and edit their new data while everyone would share one copy of the base data.

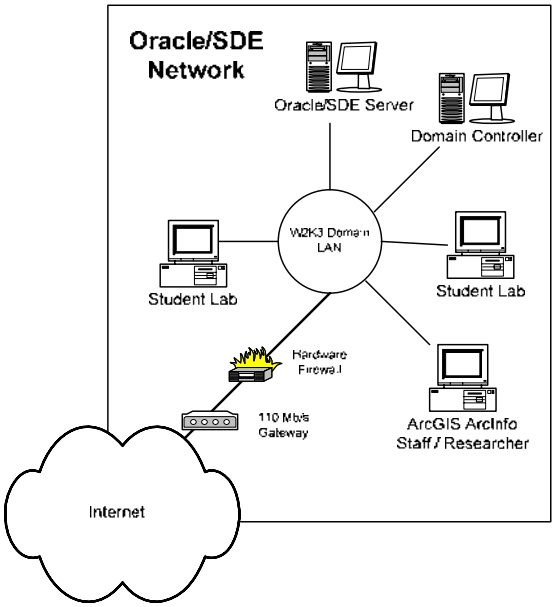


Figure 1: Oracle/SDE Network

Oracle allows multiple people access to the same data while ArcSDE allows for multiple versions of the same data to be created. ArcSDE accomplishes this feat by creating two new tables for each feature class: an A (add) table and a D (delete) table (also called Delta tables). Any additions to a feature class on the original database are kept in the A table, separate from the original data. This allow for everyone to see the original data while making changes to it that no one else can see. The D table keeps a record of any deletions to the original data. If desired, an administrator can reconcile and post any changes that need to be kept in the base database for everyone to share.

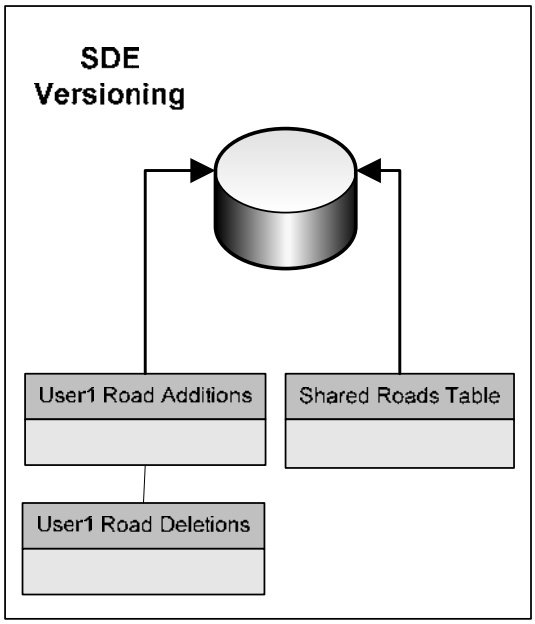


Figure 2: ArcSDE Adds/Deletes Tables

An administrator loads the GIS data on to the database previously created on an Oracle server connected to an SDE instance. There are a number of ways to do this but connecting to the SDE instance from ArcCatalog by adding a spatial database connection is recommended. Roles are created in Oracle to simplify the process of assigning users various privileges (viewer, editor, or owner) to the data. The administrator then imports feature classes from a personal geodatabase to a previously created tablespace on Oracle. Please see Appendix A for a detailed discussion of this process using data from one of our research forests, Alex Fraser Research Forest (AFRF).

After the data is loaded it is then setup for versioning by an administrator. Versioning is an ArcSDE feature that allows for various levels of access to a shared feature class. When an SDE feature class is first versioned from within ArcCatalog it appears as the default version. Feature classes are registered as versioned by an administrator who is the data owner, in this case you would have to be logged in to Oracle as the SDE user in order to register a feature class as versioned. After it is versioned, privileges can be set on the feature classes within the tablespace by Oracle role.

As new users are created they are given roles according to the level of privilege you want them to have. The roles users are given match the privileges assigned to the roles when versioning. Three steps to versioning are shown in Figure 3. For more details on setting up Versioning in ArcSDE see Appendix B.

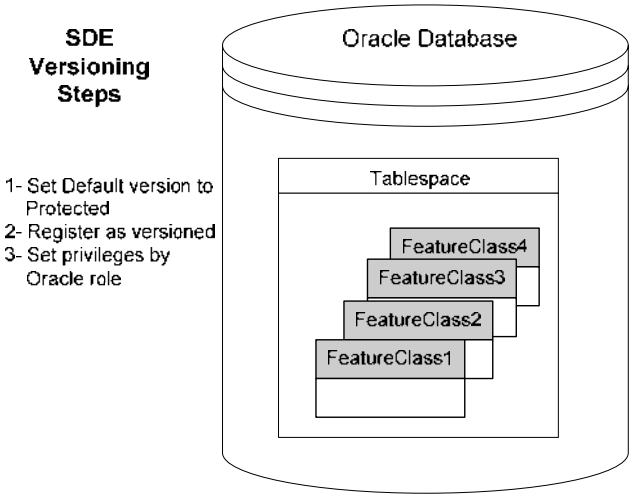


Figure 3: Versioning Steps

Summary of Oracle/SDE Setup Steps

1 – Create Oracle roles; SDE_Viewer, SDE_Editor, SDE_Owner (as SYS)
2 – Create Oracle tablespace; AFRF_GIS (as SYS)
3 – Create Oracle administrator user; CONS340ADMIN (as SYS)
4 – Import feature datasets to AFRF_GIS tablespace (as CONS340ADMIN)
5 – Set Default version to Protected (have to logon as SDE to change Default version privilege)
6 – Register feature datasets as versioned (as

CONS340ADMIN)
7 – Set Privileges; SDE_Owner role is given Select, Updata, Insert & Delete (as CONS340ADMIN)
8 – Add Oracle users giving them SDE_Owner role (as SYS - use SQL Loader & SQL Plus for multiple users)

5. Student Logon and Active Directory

The Windows 2003 Domain configuration we used is shown in Figure 1. User information is stored in Windows Active Directory. Each student has to be authenticated in some way before Oracle will allow them access to data on the server, and to the privileges associated with the versions we created in section 4. Students require unique logons/passwords so that they can each have a separate, secure version of the data to do their editing and analysis. Adding and removing user ID's for all the students cannot be done efficiently one-at-a-time so a system of adding and deleting user groups needs to be developed.

Oracle can use the Window's Active Directory to add groups of users. Windows does the authentication and Oracle accepts this authentication via the internet protocol: Lightweight Directory Access Protocol (LDAP). Unfortunately, ESRI's Spatial Database Engine (SDE) that is required for GIS access to relational databases, does not support this option in ArcSDE versions up to 9.1. If an Oracle client is installed on each student's workstation it may be possible to use Window's authentication. This option was not tested because of limited available space on student's workstations.

A work-around solution to this problem is the use of scripts for SQL Loader and SQL Plus software. The scripts were created to take a group of users from a class, and use this class list to create user ID's in Oracle. This process is outlined in Figure 4. This figure and the scripts in Appendix C refer to the Conservation 340 class (C340) that tested out this process for the first time.

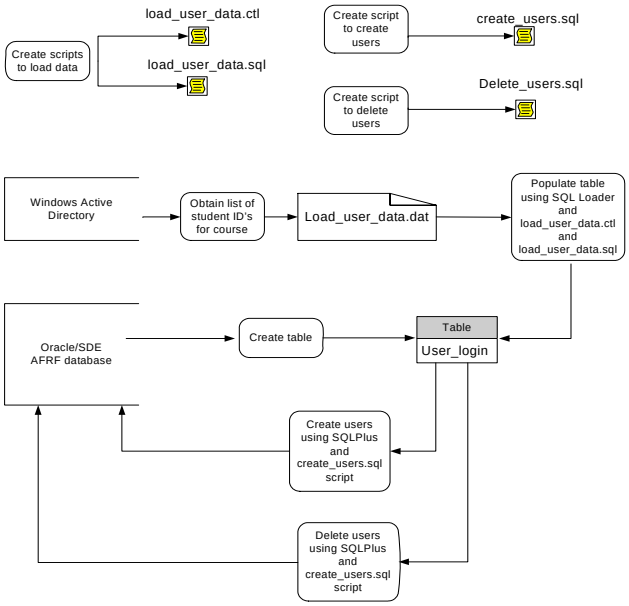


Figure 4: Adding Oracle Users Diagram

The process outlined in Figure 4 is made up of 6 basic steps:

1. Create an ASCII file of users to be added to Oracle. Although you can export a list from Active Directory, in our case students were not listed by class but by Organizational Unit so we had to create a list manually. The example in Appendix C, File 1: load_user_data.dat has the same user and password for convenience of testing. The file in this case needs to be setup with the user password first then the user login name. We used student numbers as the password as this was relatively secure. An extra step could have been added to have students change their passwords using SQL Plus but we wanted to minimize the setup time for students.

2. Next you have to create an empty table in Oracle to hold the user names. This is done in SQL Plus and detailed in Appendix C.

3. The empty table created in step 2 is now populated with the user data from step 1. This is done from the command prompt using SQL Loader and the script load_user_data.ctl as shown in Appendix C.

4. New users are now created from the table populated in step 3 in Oracle using a script run from SQL Plus. The script; create_users.sql is shown in Appendix C. This script creates users and passwords, defines which tablespace users will have access to, sets the quota on the tablespace as unlimited, and grants the role SDE_Owner (discussed in Appendix A) to each user.

5. An administrator now has to set versioning and permissions so that students can create their own versions of the data. This is done from ArcCatalog connecting to Oracle as a user with administrator privileges to the tablespace that users were given access to in step 4. The dataset is registered as versioned and the role given to users in step 4 assigned all the privileges possible; Select, Update, Insert, and Delete.

6. A final script is used from SQL Plus to delete users after the lab is complete and the marks submitted. An administrator would then have to delete all data created by the users from within Oracle. The data to be deleted are the A and D tables referred to earlier in section 4.

6. Student Creation of New Feature Classes

Users cannot add new feature classes to the CONS340 dataset (or delete feature classes). Only the owner, CONS340ADMIN can add/delete feature classes to this dataset. Users have to create their own version, with the default *Private* setting so that only they can see their new data. After creating a new feature dataset, users have to *Register as Versioned* if they wish to edit the data. Users do not have to set privileges to edit their own *Private* feature dataset.

When a user creates a feature dataset only the name is added to the Oracle database. Only when a feature class is created is a new table added to the Oracle Schema under the users login name such as UserName.Roads

Conclusion

Using a multi-user relational database to provide students with data has simplified the process of accessing data, creating new data, and marking student work. There is a significant amount of work to create an Oracle server and setup student accounts but once it is created subsequent student labs can be setup quickly and efficiently.

Appendix A – Setting up Oracle and Loading data

A1. AFRF Database Setup – Using Oracle Enterprise Manager

Roles

Database user roles were first created. SDE_Viewer, SDE_Editor, and SDE_Owner were created using Oracle Enterprise Manager to allow users to connect to the SDE database and perform various GIS operations (viewing, editing, and creating new feature classes).

Role	SDE_Viewer	SDE_Editer	SDE_Owner
System Privileges	CREATE SESSION	CREATE SESSION	CREATE SESSION CREATE PROCEDURE CREATE SEQUENCE CREATE TABLE CREATE TRIGGER

Figure A1: Database User Roles

SDE_Viewer can only view the data and export it. SDE_Editer can only edit existing data and export it based on the extra privileges assigned to it when creating the version. SDE_Owner can create new data, edit it, or export it.

This setup allows the database manager to define object privileges for datasets and assign them to these Roles once, and then as users are created assign them one of the roles. In this way, all the privileges assigned to a Role are passed to the user.

Tablespace

The AFRF_GIS tablespace was then created using Oracle Enterprise Manager logged on as SDE.

FRSCORCL.FORESTRY.UBC.CA > Storage > Tablespaces > AFRF_GIS.dbf

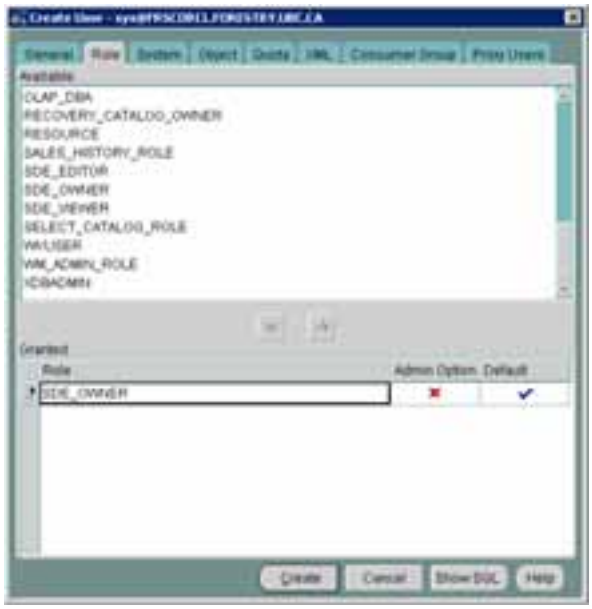
The AFRF_GIS tablespace was defined with 1GB in size, Type – Permanent, Extent Management - Locally managed, Uniform Allocation, 128Kb size, Segment Space Management - Automatic, Enable logging - Yes, with a block size of 8192 Bytes.

Figure A2: Create Tablespace Screen Shot

Then SDETEMP tablespace was created with the following parameters:
1GB in size, Type – **Temporary**, Extent Management - Locally managed, Uniform Allocation, **272**Kb size, Enable logging - **No**, with a block size of 8192 Bytes.

Users

The user AFRF_GIS was then created (CONS340ADMIN is used in place of AFRF_GIS in other parts of this article). The default tablespace was assigned to AFRF_GIS and the temporary tablespace was assigned to SDE_TEMP. It was assigned the role of SDE_Owner. The AFRF_GIS user's Quota for the newly created tablespace AFRF_GIS and SDE_TEMP were then given an unlimited quota size.



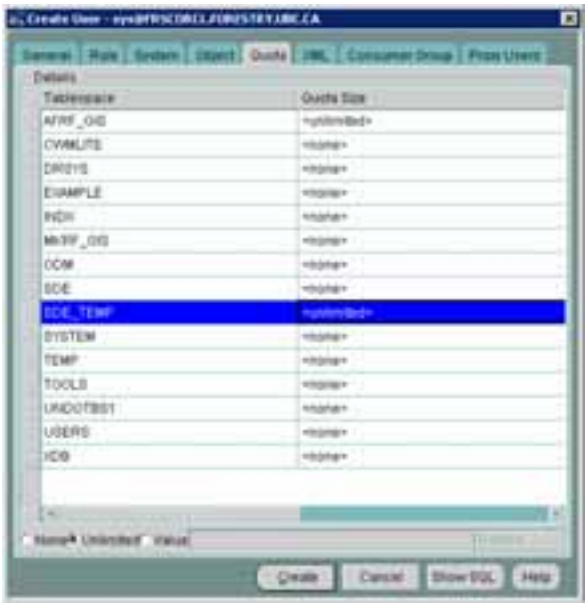


Figure A3: Create User Screen Shot

Additional users that will be editing the dataset are given accounts using Oracle Enterprise Manager or SQL Loader / SQL Plus for a large number of users. In order to be able to create log files this user has to be given the SDE_Owner role initially. After the user logs on for the first time these log files are created automatically. Afterwards, this role is revoked and replaced by SDE_Viewer or SDE_Editor depending on what activities you plan to allow. A user is given an unlimited quota size for AFRF_GIS. A user's Tablespaces are defined the same as AFRF_GIS. We only used SDE_Owner for our students so we didn't have to worry about revoking this role.

To add large groups of users, SQL Plus scripts can be used to take a list of users and passwords and assign them to the tablespaces and roles described above. A detailed description of this process can be found in Appendix C.

A2. Import Feature Datasets – Using ESRI ArcCatalog

The AFRF personal geodatabase feature datasets were then imported to Oracle using the projection and domain settings created for the personal geodatabase.

A database connection was created from ArcCatalog to Oracle using the tablespace owner AFRF_GIS user. New feature datasets were created and coordinate system information was imported from the personal geodatabase. Feature class data was then

imported from the personal geodatabase. The personal geodatabase structure versus the Oracle SDE database structure is shown in the next figure. They are same except for the addition of the tablespace owner placed before each feature dataset and feature class.

Appendix B – SDE Versioning

A version is an alternative representation of the geodatabase that has an owner, a description, and a level of access. Versioning has three basic steps;

• Create versions
• Register each dataset as versioned
• Set privileges

Each version can have only one level of access:

● Private [only owner view/modify]
● Protected [any user can view, only owner can edit] and
● Public [any user can view/modify]).

Versions allow for multi-user editing and reconciliation as well as distribution of the database as read-only among multiple users.

1 - Default Version Protected

In ArcCatalog Rt. Click on the database connection to AFRF_GIS (in this case 340_ADMIN gisorcl.forestry.ubc.ca) and choose Versions as shown in Figure B1. Edit the Default version assigning it Protected access (if it is not already) so only the owner can edit it (have to be data owner to change access, SDE is the owner of the Default version). This can also be done from ArcMap using the Versioning Toolbar after you have added a feature from the SDE database of interest.

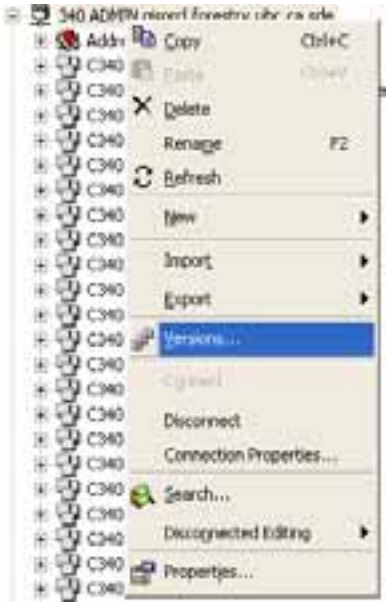


Figure B1: Default Version Protected

2 – Register As Versioned

Rt. Click on Feature Dataset (in this case CONS340ADMIN.CONS340) – Register as Versioned (users will only see versioned feature datasets) as shown in Figure B2.

Figure B2: Register as Versioned

3 – Set Privileges

A level of access is assigned to a Role (roles previously created in Oracle) using *Privileges* from ArcCatalog while connected to Oracle as the database

owner, in this case CONS340ADMIN.CONS340 as shown in Figure B3.

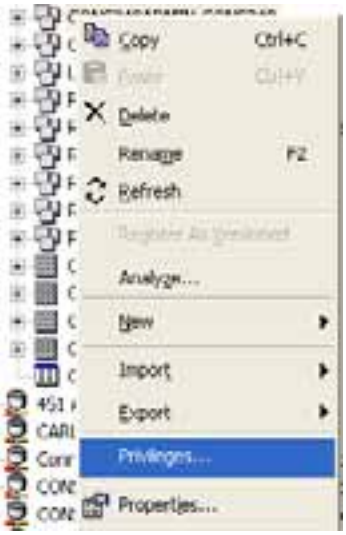


Figure B3: Set Privileges

Rt. Click on Feature Dataset – Privileges – type in Role – select privileges according to the Role as shown in Figure B4. (users will now see feature datasets and the feature classes within them)

Role	Privilege
SDE_Viewer	Select
SDE_Editor	Select, Update, Insert, Delete (can not create new feature class)
SDE_Owner	Select, Update, Insert & Delete

Figure B4: SDE_Owner Privileges

The difference between SDE_Editor and SDE_Owner is based on how Oracle has been setup to assign privileges. In this case SDE_Owner can copy/paste, create new features, create new versions, and reconcile. SDE_Editor cannot create new features within an existing dataset, only the data creator (or the person who imported the data) in this case CONS340ADMIN. All users from the Conservation 340 class were given the SDE_Owner role so they could create their own versions and edit existing CONS340 data. Only the CONS340ADMIN user could reconcile changes to the Default version as shown in Figure B5.

A user is made a member of a Group with particular privileges. The additions or deletions that a user makes are saved in A (add) or D (delete) tables while the original dataset is left unchanged until the various versions are posted. There is no duplication of tables as all users access the same original data and maintain their own A and D tables.

If you belong to a table Owner Group you can create your own version(s) and reconcile them with other table Owners. Or, an administrator can create versions and allow user access to these versions by logon. The administrator would then reconcile various versions based on whether the additions/deletions were to be saved in the original database. Reconciliation checks to see if there are any conflicts between versions and prompts you to choose what to do. Posting actually saves changes to the database.

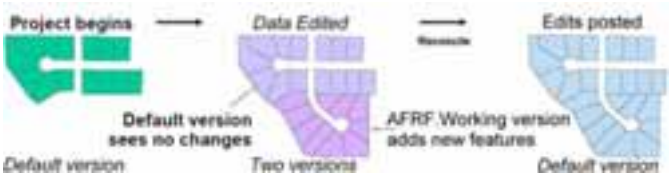


Figure B5: Reconcile Versions

If the default version is assigned private access, then users other than the owner will not be able to see it.

Appendix C – Create/Delete Oracle Users Script

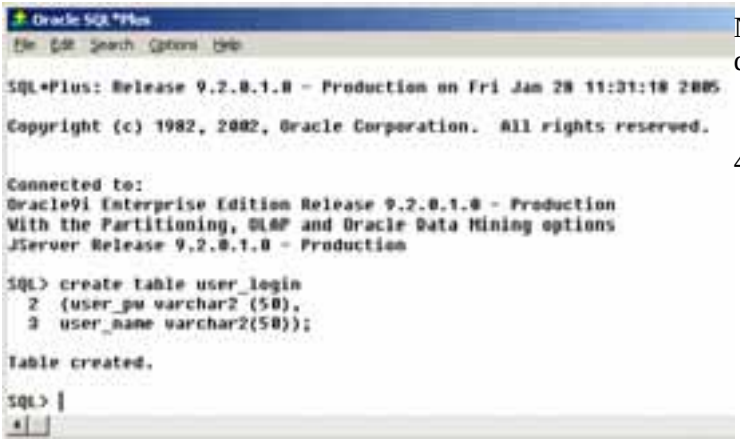
1 - Create ASCII file of users to be added to Oracle

The load_user_data.dat file (Appendix 1, File 1) was created from MicroSoft's Active Directory (AD). If users are listed by group then the DumpSec program (www.somarsoft.com) can be used to create an ASCII file of group and user name. If users are listed by Organizational Unit (OU) then the ASCII file has to be created manually. Users in a particular class in Forestry are added to the AD manually from a list obtained from class attendees, so the class list can be used to create this file. The format for the file is defined in the load_user_data.ctl (Appendix 1, File 2) file as spaces 1 to 7 for the group name and space 9 on for the user_name with a maximum number of characters of 30 for the user_name as defined in the create_users.sql (Appendix 1, File 3) script. The load_user_data.dat file with the groups and user names are outputs from the DumpSec program.

NOTE: - Illegal characters in a user name will cause an error when loading. An example of an illegal character is a "-" hyphen, whereas an "_" underline works.
- Also, columns must be separated by spaces, not tabs or data will be jumbled between comumns.

2 – <u>Create Empty Table</u>

Use SQL Plus to create the table user_login. Appendix1, Figure 1 has the SQL query as shown in the screen shot below left. (Logon as 'system' in SQL Plus.)



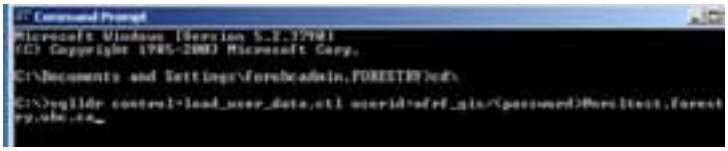

```
create table user_login
(user_pw varchar2(50),
user_name varchar(50));
```

NOTE: To create tables in the CONS340 tablespace you should be logged into SQL Plus as the tablespace owner (CONS340 in this case) otherwise you have to identify the table with a prefix CONS340.USER_LOGIN and use the CREATE TABLE command with the TABLESPACE option to identify the tablespace within which to create the table as shown to the right above.

```
create table cons340user_login
(user_pw varchar2(50),
user_name varchar(50))
tablespace afrf_gis;
```

3 - <u>Populate table using SQL Plus and SQL Loader</u>

The ASCII file of users now has to be entered into the Oracle table created in step 2. The script; load_user_data.ctl, as shown in Appendix 1, File 2 is now run using SQL Loader. This file is used in SQL Loader to load the user data from the ASCII file into the Oracle table using the following command (or batch file command.bat Appendix 1, File 3) from the Command Prompt as shown below (best to change directories to where the files are located).



NOTE: This example uses the user, password, and database name that owns this tablespace.

4 - <u>Create New Users in Oracle</u>

```
Oracle SQL*Plus
File Edit Search Options Help
SQL> @e:\dl\auth\create_users
creating user ubc_Administrator
granting permissions to user ubc_Administrator
creating user ubc_MetShowServices
granting permissions to user ubc_MetShowServices
creating user ubc_badmin
granting permissions to user ubc_badmin
creating user ubc_forubcadmin
granting permissions to user ubc_forubcadmin
creating user ubc_gerd
granting permissions to user ubc_gerd
creating user ubc_cheryl
granting permissions to user ubc_cheryl
creating user ubc_dave
granting permissions to user ubc_dave
creating user ubc_jan
granting permissions to user ubc_jan
creating user ubc_mfiaron
granting permissions to user ubc_mfiaron
creating user ubc_michael
granting permissions to user ubc_michael
creating user ubc_paul
granting permissions to user ubc_paul
creating user ubc_rick
granting permissions to user ubc_rick
creating user ubc_students
granting permissions to user ubc_students
creating user ubc_susanc
granting permissions to user ubc_susanc
creating user ubc_ASPNET
granting permissions to user ubc_ASPNET
creating user ubc_IUSR_MAINLLC
granting permissions to user ubc_IUSR_MAINLLC
creating user ubc_IUSR_MAINMKRF
granting permissions to user ubc_IUSR_MAINMKRF
creating user ubc_IWAM_MAINLLC
granting permissions to user ubc_IWAM_MAINLLC
creating user ubc_IWAM_MAINMKRF
granting permissions to user ubc_IWAM_MAINMKRF

PL/SQL procedure successfully completed.
```

A script, create_users.sql (Appendix 1, File 4) is then used to create users in Oracle from the table of users just created. This script is run from SQL Plus as shown below. To run the script, use the ampersand character "@" followed by the path and file name of the script. In order to create users you need to be logged in to SQL Plus as DBA. The results of the script commands are shown as the script executes.

5 – Set Versioning/Permissions

Using the administrator login (CONS340ADMIN in this case) from ArcCatalog, Rt click on the CONS340ADMIN.CONS340 dataset and click *Register as Versioned* then Rt click again and choose *Privileges*. Set SDE_OWNER to have all privileges so students can create their own versions and edit existing CONS340 data. This allows students to create their own versions with any changes held in the Delta tables.

6 - Delete Users in Oracle

A script, delete_users.sql (Appendix 1, File 5) can then be run from SQL Plus to remove the users if desired. The script uses the preface UBC to find and delete only the new users created.

7 - Listing of Scripts and Files

File 1: load_user_data.dat

```
test1        test1
test2        test2
```

NOTE: This users file must have its contents separated by spaces, not tabs or the data will be jumbled.

SQL Query As AFRF_GIS Or As Other User

```
CREATE TABLE user_login
        CREATE TABLE afrf_gis.user_login
(user_pw varchar2(50),
        (user_pw varchar2(50),
 user_name varchar2(50));
        user_name varchar2(50))

        TABLESPACE AFRF_GIS;
```

File 2: load_user_data.ctl

```
LOAD DATA
REPLACE
INTO TABLE
User_login
FIELDS TERMINATED BY ' ' OPTIONALLY
ENCLOSED BY ""
TRAILING NULLCOLS
(
user_pw position (1:10),
user_name position (11)
)
```

File 3: command.bat

```
sqlldr control=load_user_data.ctl
```

```
userid=cons340admin/<password>@gisorcl.forestry.
ubc.ca
```

File 4: create_users.sql

```
-- procedure to create new users if they do not already
exists
-- grant privileges to newly created users
-- Created: 27 Sept, 2004 12:57 pm
-- Edited for CONS340 18 October, 2005

set serverout on size 1000000;

declare

 create_stmt varchar2(400);
 grant_stmt  varchar2(200);

 user_var    varchar(30);
 pass_var    varchar2(30);

 tb1 varchar2(15) := 'AFRF_GIS';
 tb2 varchar2(15) := 'sdetemp';
-- tb3 varchar2(15) := 'mkrf_ver';

cursor user_cur IS
  select distinct user_pw, user_name
  from cons340admin.user_login
  where upper('340_' || user_name) not in (select
username from user_users)
;

-- the above select uses the table user_users instead of
dba_users because
-- the SQL Plus user is not a DBA but a table owner

begin
  for user_rec in user_cur
  loop

  pass_var := 'C_' || user_rec.user_pw;
  user_var := 'C340_' || user_rec.user_name;

   create_stmt := 'create user ' ||  user_var ||
                ' identified by ' || pass_var ||
                ' default tablespace cons340
                  temporary tablespace temp
                  quota unlimited on ' || tb1 ||
               ' quota unlimited on ' || tb2;
  --             ' quota unlimited on ' || tb3

  grant_stmt := 'grant sde_ubc_owner to ' || user_var ;

  dbms_output.put_line('creating user ' || user_var || '
password ' || pass_var);

  EXECUTE IMMEDIATE create_stmt;

  dbms_output.put_line('granting permissions to user '
|| user_var);

  EXECUTE IMMEDIATE grant_stmt;

 end loop;  -- user_rec


exception
   when others
   then
       dbms_output.put_line('Error in load_user_data: '||
SQLERRM);

end;
/
show errors
```

File 5: delete_users.sql

```
-- procedure to delete users if they already exists
-- Created: 27 Sept, 2004
-- Edited for CONS340 October 21, 2005
-- make sure to disconnect the users before executing
the script...
-- if not disconnected, a user cannot be dropped

set serverout on size 1000000;

declare

 delete_stmt varchar2(400);
 user_var    varchar(30);

 cursor user_cur IS
  select distinct User_name
  from cons340admin.user_login
  where upper('c340_' || User_name) in (select
username from dba_users)
```

```
;

begin
  for user_rec in user_cur
  loop

  user_var := 'c340_' || user_rec.User_name;

  delete_stmt := 'drop user ' || user_var || ' cascade';

 dbms_output.put_line('deleting user ' || user_var);

  EXECUTE IMMEDIATE delete_stmt;

 dbms_output.put_line('user ' || user_var || ' deleted');

  end loop;  -- user_rec

exception
   when others
   then
      dbms_output.put_line('Error in delete_user: '||
SQLERRM);

end;
/
show errors
```

Author Information

Mr. Maedel, a GIS and Remote Sensing Analyst, is currently a staff member in the Department of Forest Resources Management at the University of British Columbia.

Jerry Maedel
GIS/RS Analyst
Department of Forest Resources Management
Forest Sciences Centre
2045-2424 Main Mall
University of British Columbia
Vancouver, BC., Canada V6T 1Z4
E-Mail: jerry.maedel@ubc.ca
Web: www.gis.ubc.ca
Phone: (604) 822-2284
FAX: (604) 822-9106

Dr. Meitner, an environmental psychologist, is currently an Assistant Professor in the Department of Forest Resources Management at the University of British Columbia.

Michael J. Meitner
Assistant Professor
Collaborative for Advanced Landscape Planning
(www.calp.forestry.ubc.ca)
Department of Forest Resources Management
Forest Sciences Centre
2045-2424 Main Mall
University of British Columbia
Vancouver, BC., Canada V6T 1Z4
E-Mail: mike.meitner@ubc.ca
Phone: (604) 822-0029
FAX: (604) 822-9106

Mr. Aron, a Forester and GIS specialist, is currently a staff member at Malcolm Knapp Research Forest.

Ionut Aron
Research Coordinator
UBC Malcolm Knapp Research Forest
14500 Silver Valley Road
Maple Ridge, B.C. Canada
Mailing Address
P.O. Box 21120
Maple Ridge R.P.O.
B.C. Canada
V2X 1P7
E-Mail: ionut.aron@ubc.ca
Web:
http://www.mkrf.forestry.ubc.ca/general/index.htm
Phone: (604) 463-8148
Fax: (604) 463-2712