Moose Monitoring Database Application

Table of contents

Overview

NPS Moose Monitoring Database Application

Overview

Moose population structure and abundance are monitored the National Park Service in Alaska. Methods follow the protocols developed by the Arctic and Central Alaska Inventory and Monitoring Networks and rely on the GeoSpatial Estimator toolset described in Kellie and Delong, 2006.

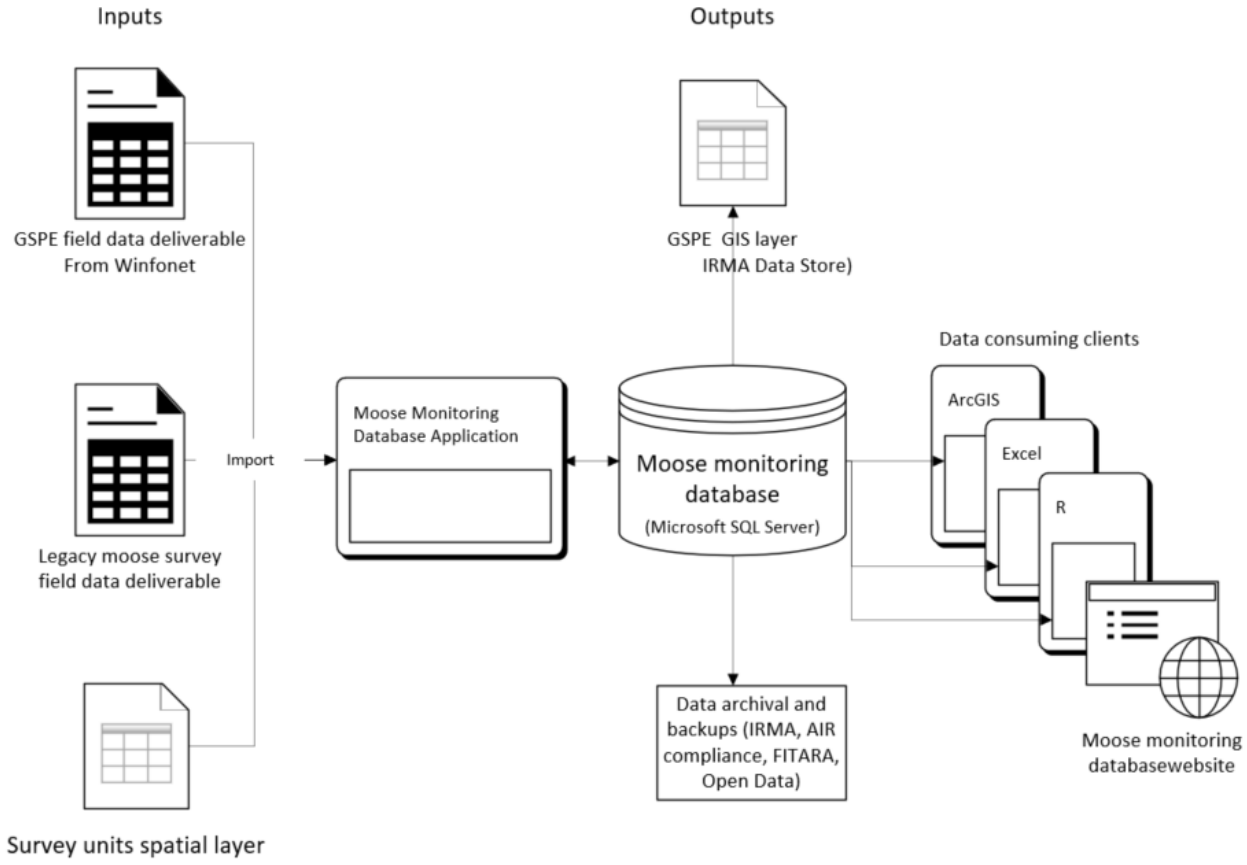
Moose monitoring deliverables exist in ADF&G's Winfonet system, an analytical and data storage system for moose monitoring survey data.NPS policy requires that moose monitoring data reside in an authorized Department of the Interior data store. GeoSpatial Estimator deliverables were copied from Winfonet into an SQL Server database in 2022 in order to align with NPS policy. A front-end application was developed in 2022 to allow for storage, retrieval, and basic analyses. This document describes the moose monitoring database, data management system, and front-end application.

Sensitivity statement

The Moose Monitoring Database Application contains information about a species of commercial interest, some or all of which may be subject to data sharing agreements with partner agencies. Please see Dataset sensitivity

Conceptual diagram

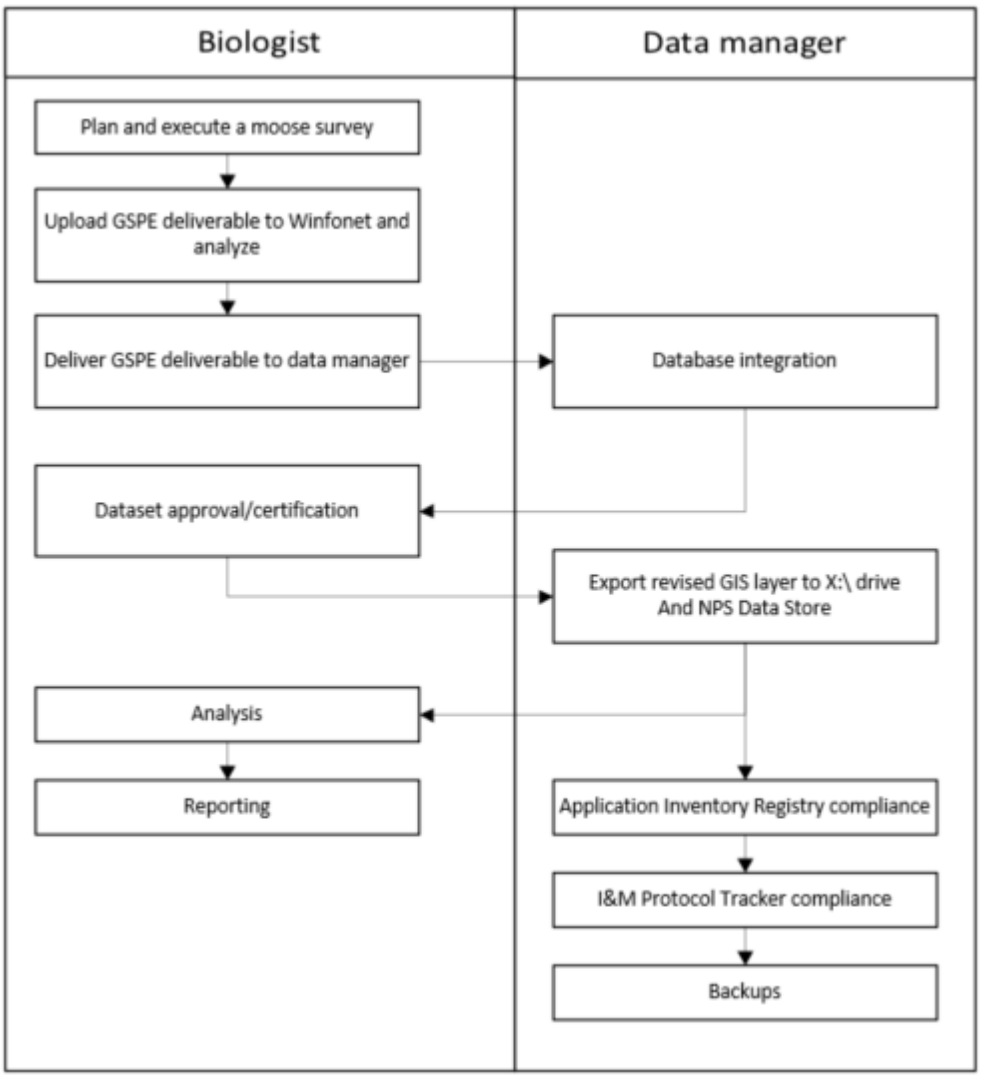
The heart of the moose monitoring system is the Moose database hosted on an SQL Server at the NPS Fairbanks Administrative Center.



Work flow

The moose monitoring work flow can be summarized simply in the steps below

1. Collect data according to ARCN/CAKN protocol
2. Process GSPE deliverable in Winfonet
3. Integrate GSPE deliverable to ARCN/CAKN moose monitoring database
4. Analyze
5. Report



Resources

Resources available to moose monitoring personnel include:\

Resource	Location
Data deliverable files	ARCN O:\Monitoring\Vital Signs\Moose\Data CAKN J:\Monitoring\Moose\Data
Moose monitoring master Project reference in the IRMA Data Store	ARCN: DataStore - Project - (Code: 2222140) (nps.gov) CAKN: DataStore - Project - (Code: 2220369) (nps.gov)
Moose monitoring database	See Database documentation
Moose monitoring database application	NPS-ARCN-CAKN/Moose3 (github.com)
Alaska Department of Fish and Game's Winfonet database	Home (alaska.gov)

System administration

The moose monitoring database is administered by the ARCN/CAKN data managers and supported by the information technology staff at the Fairbanks Administrative Center, 4175 Geist Rd, Fairbanks, AK.

Dataset sensitivity

Dataset sensitivity

Sensitivity statement
Moose monitoring data collected by the National Park Service's Arctic and Central Alaska Inventory and Monitoring Networks will be withheld from the public domain until directed otherwise in writing by the park superintendents or their delegates.

Justification
Data and products derived from data collected by federal employees during the course of their normal duties are, with few exceptions, in the public domain under the U.S. Government Works license:

U.S. Government Works Data and content created by government employees within the scope of their employment are not subject to domestic copyright protection under 17 U.S.C. § 105. Government works are by default in the U.S. Public Domain.

Wildlife data may be legally classified as sensitive and withheld by the public if they concern a species of commercial interest:

Under National Parks Omnibus Management Act, Section 207, of 16 U.S.C. § 5937; Specific location (geographic coordinates) of these harvestable (commercial) species will not be released to the general public due to their "commercial" value. Any such data collected will be redacted from the publicly released datasets. This protects collaborating agencies data as well; but does not preclude sharing between collaborating agencies. Processed data with coordinates redacted (such as summaries, processed maps, or products) remain publicly available.

Contact the Arctic or Central Alaska Inventory and Monitoring Progams to request access to moose monitoring data.

Park Superintendent's protected data memos

The moose monitoring database does not contain specific locations of harvestable species as the data is rolled up to the survey unit level and the official policy expressed by the various Park superintendents aligns with the statement above:

Network	Park	Protected data memorandum
ARCN	BELA	https://irma.nps.gov/DataStore/DownloadFile/624889
CAKN	DENA	Unpublished
ARCN	GAAR	https://irma.nps.gov/DataStore/DownloadFile/624888
SWAN	KATM	https://irma.nps.gov/DataStore/DownloadFile/624924
SWAN	KEFJ	https://irma.nps.gov/DataStore/DownloadFile/624922
SWAN	LACL	https://irma.nps.gov/DataStore/DownloadFile/624923
ARCN	WEAR (CAKR, KOVA, NOAT)	https://irma.nps.gov/DataStore/DownloadFile/624887
CAKN	WRST	Unpublished

CAKN YUCH Unpublished

Reference Sensitivity: IRMA Data Store

The Data Store help module below provides definitions for all of the reference sensitivity, quality and proprietary designations.
https://irma.nps.gov/DataStore/DownloadFile/626221

Data Sharing Agreements

Moose monitoring data may be subject to data sharing agreements between the NPS, Alaska Department of Fish and Game and possibly Yukon Fish and Game Association. As of this writing, the NPS is unaware of any such written agreement.

Current policy

Moose monitoring data will be withheld from the public domain until directed otherwise in writing by the park superintendents or their delegates.

Data sharing agreements

Data sharing agreements

None known as of 2022
[Publish here as they are discovered]

Security

Security

This section describes the moose monitoring database and dataset security model.

Database

The moose monitoring database resides in an SQL Server at the Fairbanks Administrative Center. Access rules are enforced by the database and managed by the database administrators (CAKN/ARCN data managers). Permissions are granted via Windows Authentication; the same credentials used to log into your NPS computer are passed through to the SQL Server. Permissions will be granted to Park biologists, biometricians, data managers and biotechnicians with minimum permissions granted to accomplish goals.

Summary

- No one may access the Moose database without permission.
- Permissions on the Moose database are managed by the database administrators (data managers).
- Permissions are enforced by the database, regardless of client software.
- Minimum permissions are granted to accomplish tasks (analytical tasks require only read-only permission, for example. Editing/importing data require read-write permissions. Changing the database schema requires admin privileges).
- No passwords or user accounts are needed (Windows Authentication)

- Permissions on the database are global - there is no ownership of certain surveys or data related to a certain park or network, for example. If you have permission to access data in the database, you have permission to access all the data in the database.

Dataset

For the purpose of this section 'Dataset' refers to the analytical product produced by the moose monitoring database. The primary product is a shapefile published to the IRMA Data Store (see Dataset publication). We will take a conservative approach to dataset publication. Within IRMA Data Store the certified Geospatial Dataset Reference will be tagged 'NPS Internal' and file access will be restricted to specific individuals including wildlife biologists, biometricians and data managers.

Summary

- The moose monitoring dataset will be published in IRMA Data Store using maximally restrictive security options.
- The dataset will consist of a shapefile or geodatabase of certified records.
- Only certified data will be published (see Dataset certification on how records are tagged as 'Certified').
- Records tagged 'Raw' or 'Provisional' will not be published. You may store and develop such records in the database without having to worry about them being published prematurely.
- IRMA Data Store is capable of restricting file access to specific individuals and the product will be published using that functionality.
- IRMA Data Store is capable of restricting References to NPS-only and the product will be published using that functionality.
- The certified moose monitoring data product will be published to a single, authoritative Geospatial Dataset Reference in IRMA Data Store.
- The product will be versioned by date and each version will contain an incrementally cumulating set of records.

Obtaining/installing the Moose Monitoring Database Application

Obtaining/installing the Moose Monitoring Database Application

The NPS ARCN/CAKN Moose Monitoring Database Application is available to authorized NPS wildlife biologists through the IRMA Data Store (https://irma.nps.gov/DataStore/Reference/Profile/2295016).

Database permissions

Contact either of the Arctic or Central Alaska Network data managers for database permissions.

Installation

Installation is simple: just download the zipped file archive to your computer and unzip it. Double click Moose.exe to start the application. A help file is included.

Contact either of the Arctic or Central Alaska Network data managers if you have any problems.

Adding a new moose survey and GSPE data deliverable to the moose monitoring database

Adding a new moose survey and GSPE data deliverable to the moose monitoring database

Adding a new survey to the Moose Monitoring Database Application involves two (or possibly three) steps that must be performed in order:

1. Add a Survey record to the GSPE_Surveys database table
2. Add related GSPE records to the GSPE database table
3. If you are trying to add legacy moose survey data rather than standard GSPE data then it's possible the survey units do not exist in the database. In this case add the data as above but leave the SurveyUnitSet fields blank or NULL. You will have to get your data manager to add the survey unit polygons to the database and relate them to the survey data. There is currently no within-application functionality to accomplish this task.

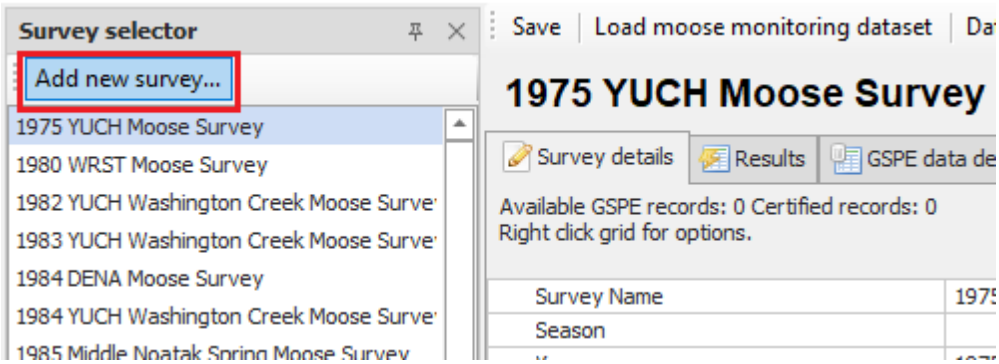
Prerequisites

- At minimum, datawriter privileges on the database. Contact your data manager.
- SQL Server Management Studio software.

Procedure

Adding a new Survey record

1. If the survey does not exist then add a new survey record to the database by clicking the Add new survey... button.



2.
3. Answer the wizard questions and save your work after the new survey appears in the surveys inventory.

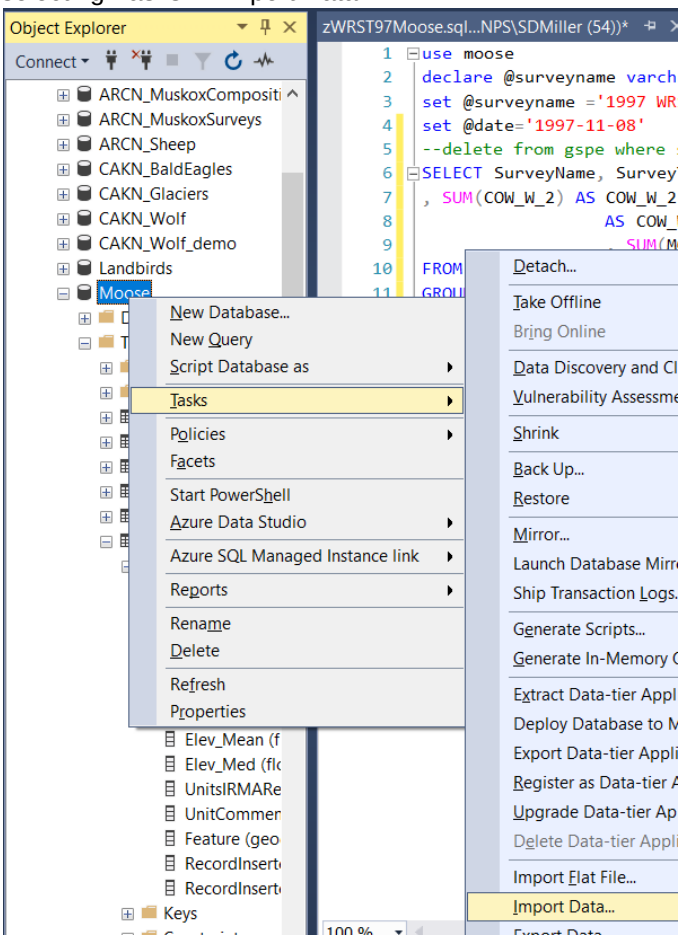
Importing GSPE records for the new Survey

As of 2022 there is no tool in the Moose Monitoring Database Application to directly import GSPE data for a new Survey. You can use Microsoft SQL Server Management Studio to import the records, or other data manipulation software, but really, the easiest way is to give the deliverable to the ARCN data manager for import. The method using SSMS is shown below.

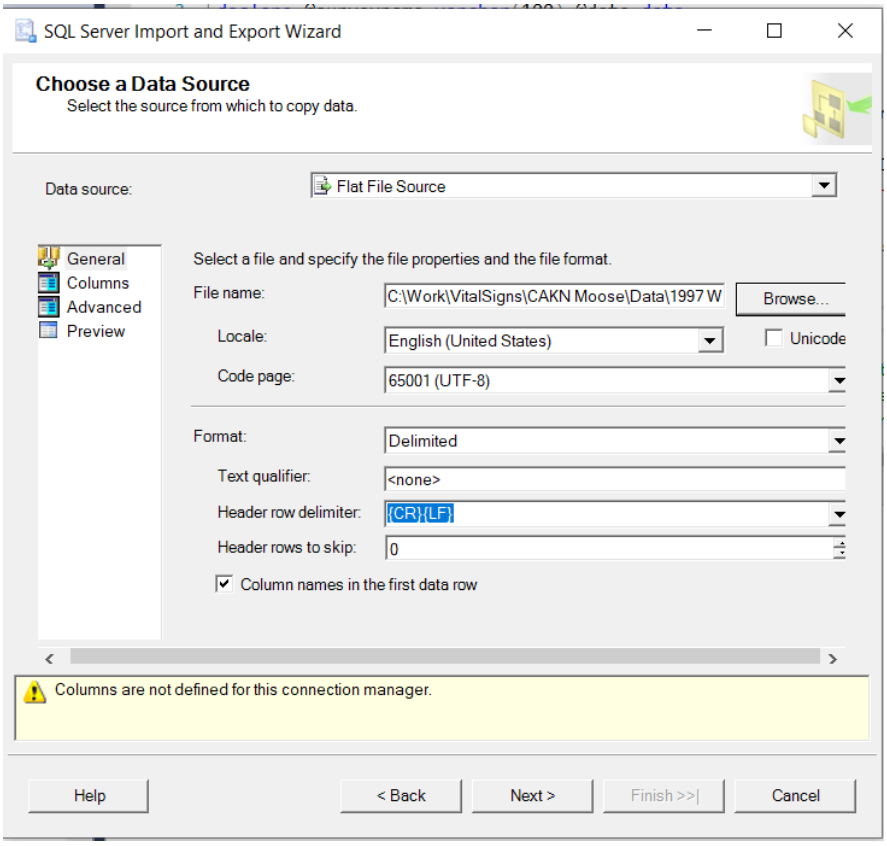
Procedure

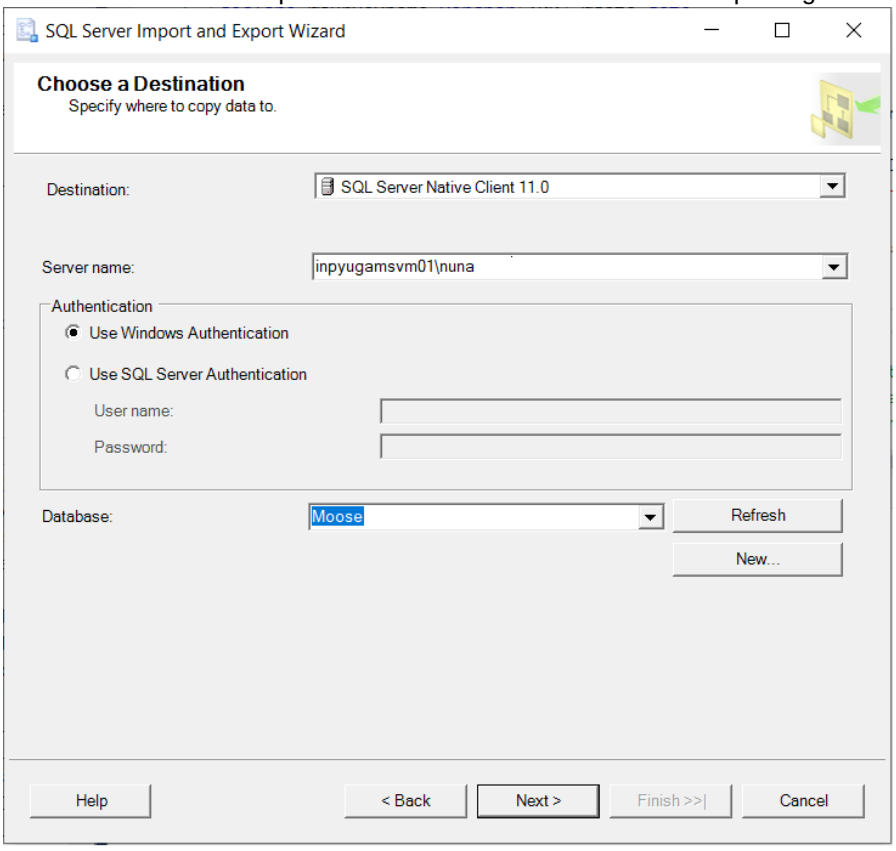
1. Start SSMS
2. Navigate to the Moose database and then locate the Sql Server Import/Export Tool: Moose -> Tasks -> Import Data...
3. *TIP: Befor you import the data in the GSPE spreadsheet, make sure the source dataset does not have*

commas in any of the fields, or, if commas exist, that the text fields are surrounded by quotes.

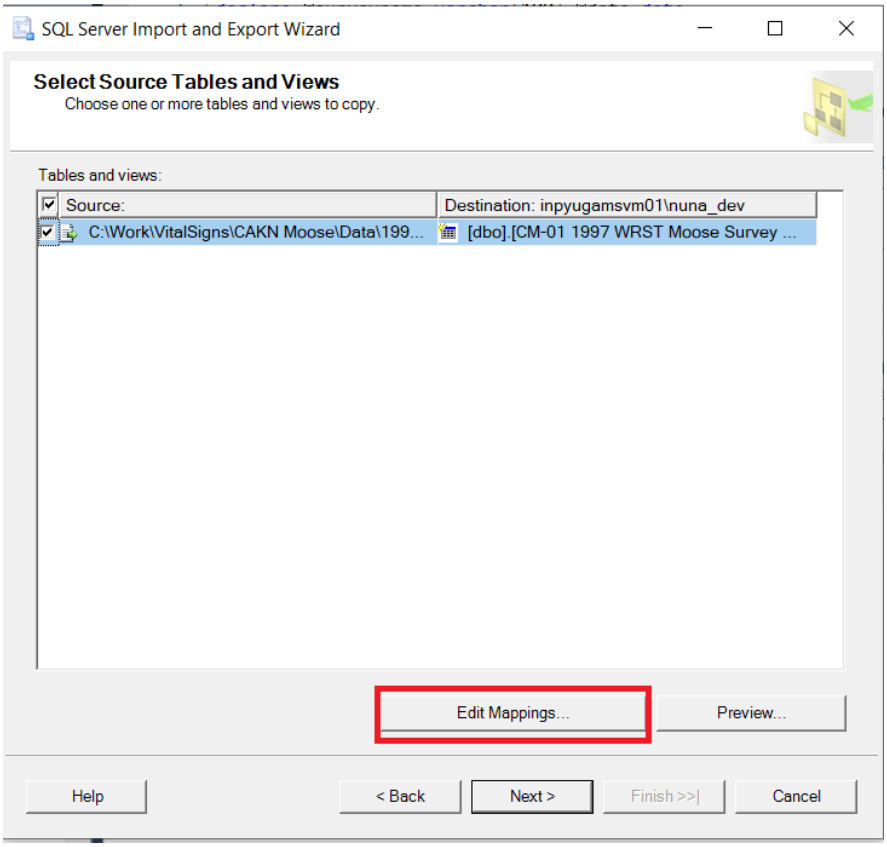
4. *TIP: To ensure that the data you import gets associated with the standard GSPE survey unit polygons, make sure your input table has a column titled **SurveyUnitSet** and each cell in this column reads **'ADFG GSPE'**.*
5. Start the Import Data... tool in Sql Server Management Studio by right clicking the database and selecting Tasks -> Import Data...
6. 

7. Choose the GSPE deliverable as a data source

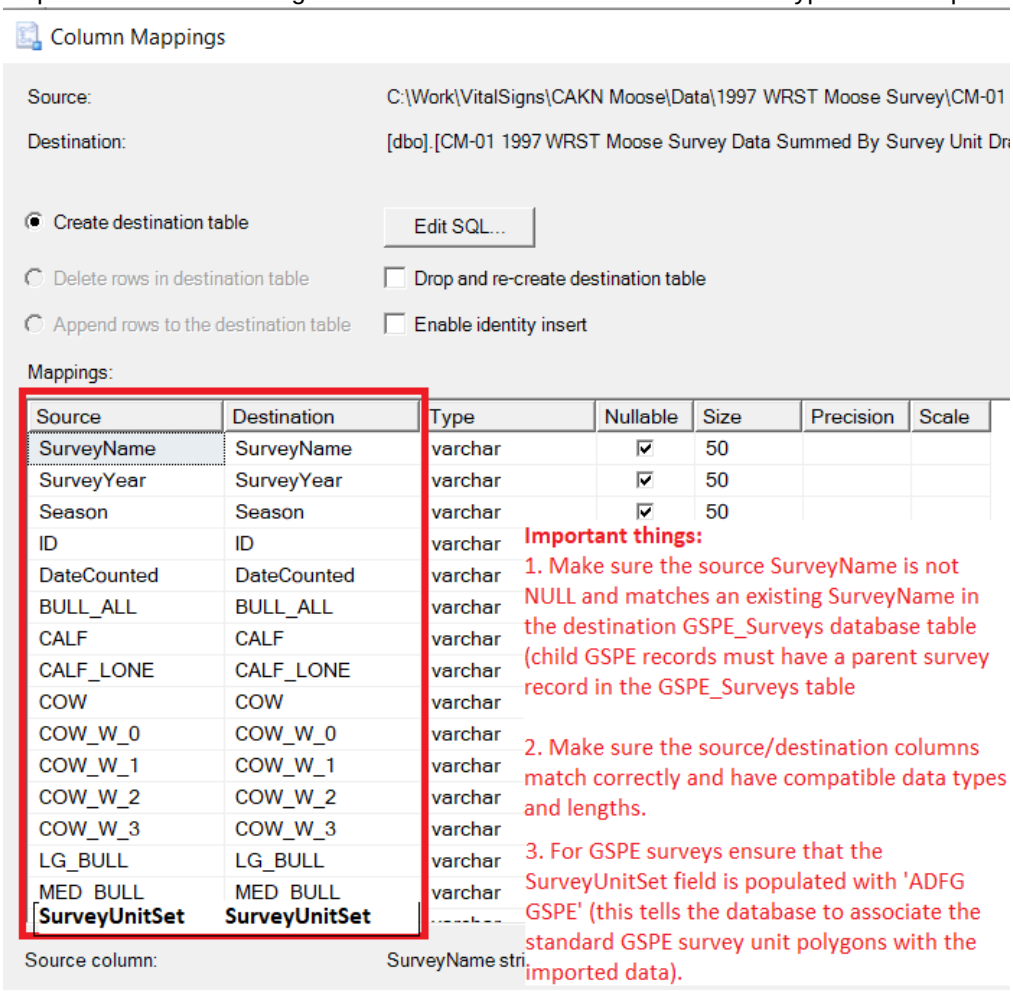
8. 

9. Choose the Moose database on the nuna SQL Server (the. Contact the ARCN/CAKN data manager for database details and/or permissions. You must have datawriter privileges at minimum).
10. 

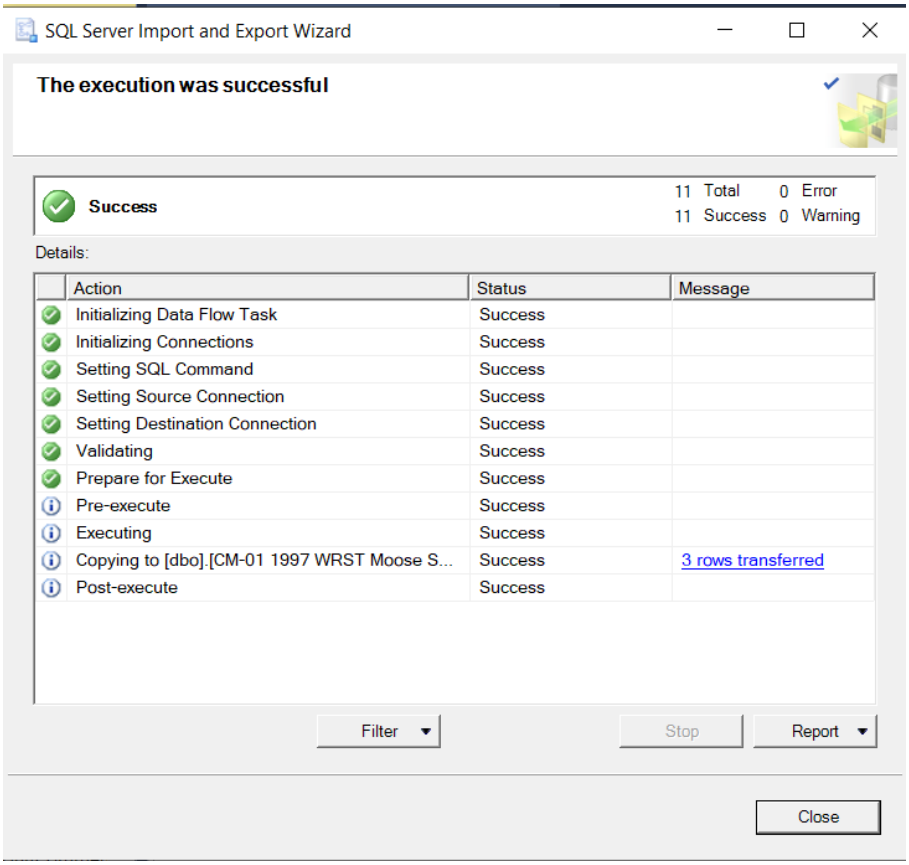
11. Click Next
12. Map the source columns to the database columns. Choose Edit Mappings...

13. 

14. Map the columns ensuring the source/destination columns and data types are compatible



15.
16. Double check all settings are correct and execute. If all goes well you should see something like this:



17.
18. Check your work to be sure the data imported correctly.
19. Validate that the destination data exactly matches the source data. This is most easily done by cross-summing the source numeric values in a pivot table in Excel and comparing them to a similar SQL query or by using the Moose Monitoring Database Application's pivot tool. Scan the comments and text fields visually.
20. Certify the dataset, if warranted. Compare the record counts, summed values, etc., against the GSPE Winfonet report. If a formal report for the dataset exists, then attempt to regenerate any summary tables. If all looks good and the biologist approves, then certify the survey dataset.

Adding a new set of survey units

Adding a new set of survey units

You shouldn't normally need to add new survey units to the moose database. Adding new survey units is a rare event exclusively associated with importing older moose surveys that did not use the standard GSPE survey units. If you are importing a standard GSPE deliverable then you should not have to add the GSPE units because they already exist in the database. Just make sure the records for your survey in the GSPE table have the field *SurveyUnitSet* set to *'ADFG GSPE'*.

If you are importing an older moose survey that did not use the standard GSPE units, then you may have to import survey units from a GIS layer into the database. This is a task requiring advanced database skills and should only be done by an experienced database administrator.

Survey units were often re-used over many years so eliminate the possibility that the survey units don't already exist in the database. The 1978 DENA moose survey re-used the 1974 units, for example.

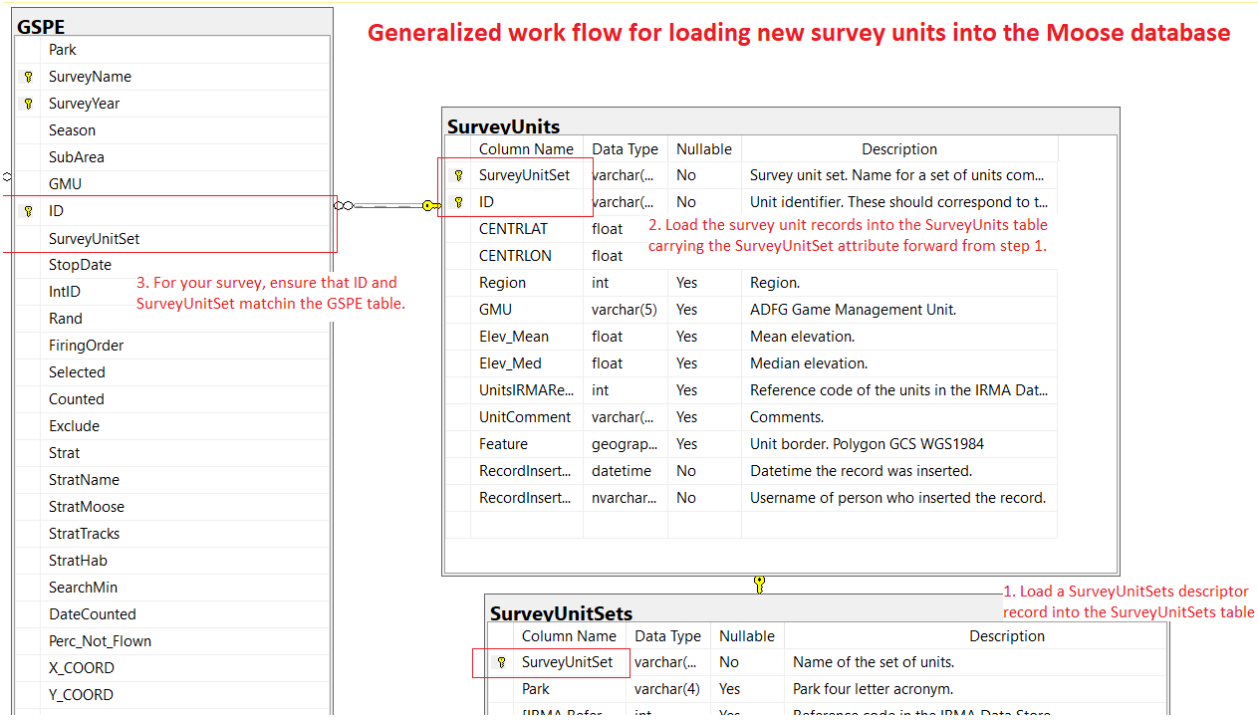
There is currently (Nov. 2022) no functionality within the Moose Monitoring Database Application to add a new set of moose survey unit polygons. You will have to do this task using GIS/Python and SQL. You can

modify the arcpy script in this section to produce a script of SQL INSERT queries that can be executed in SQL Server Management Studio to insert the units.

How survey units are related to survey records

An understanding of how survey units are related to survey data records is important for success when importing new survey records. To import survey units your objective is to:

1. Create a new SurveyUnitSet record in the SurveyUnitSets table to describe the set of units.
2. Load the new survey units into the SurveyUnits table, matching the new records to the SurveyUnitSet you chose in step 1. The survey unit polygons should be in Geographic Coordinate System, World Geodetic Survey 1984 spheroid. These polygons should be loaded into the Feature field of the SurveyUnits table.
3. Matching the SurveyUnitSet fields in the GSPE table for SurveyName and ID.
    1. If importing data from a spreadsheet simply make a column called SurveyUnitSet and fill it in with the SurveyUnitSet you chose in step 1.
    2. If retrofitting existing GSPE records then write an UPDATE query that isolates the change in SurveyUnitSet to the correct SurveyName. use extraordinary care to isolate the changes to the SurveyName in question so that you do not contaminate other records in the GSPE table. This is a job best left to an experience database admin.



Importing survey units into the Moose database from a shapefile

Normally, standard operating procedures should be terse descriptions of how to do things. I'm going to deviate from that notion because importing survey units into the Moose database has proven to be difficult so I'd like to describe the many approaches I took in the hopes that it will save someone time in the future, should the procedure need to be revised. My efforts described below mostly failed to import moose survey units into the database, but I never determined if the problems were my source file, or the software I tried to use. Someone in the future may, therefore, find the explorations below helpful.

Ogr2Ogr

I have used Ogr2Ogr in the past to convert spatial data formats successfully. I'm sure Ogr2Ogr would transfer moose unit polygons from a shapefile or other spatial data source to SQL Server. Ogr2Ogr is not currently approved for use in the NPS and I did not want to waste time trying to get it approved. I include it

as a possibility for the future.

QGIS direct import to SQL Server

QGIS will easily move data from spatial data files into spatial databases. You can load the source shapefile containing moose units into QGIS, and then create a database connection to the Moose SQL Server database and then simply drag the first over the second and it will load it into the database. From there it's a simple matter of writing an INSERT INTO FROM query to move the units into the SurveyUnits database table.

My success transferring the data was not complete because QGIS loaded the unit polygons as Geometry data types rather than the preferred Geography (geometry assumes a flat, Euclidean coordinate system). Geometry seems uncaring about valid WKT (see Remarks) where Geography is very particular. My attempts to repair/convert them to Geographies using SQL all failed (see MakeValid()).

QGIS/CSV/Python scripting

Next I tried exporting the survey units from QGIS as a CSV file with the polygons included in a column as WKT. I could then loop through the records and generate INSERT queries for the database import. This idea succeeded, but the WKTs were again all rejected by SQL Server as invalid. I then tried to do the same idea using Python scripting to generate INSERT queries. Again invalid WKTs. At this point I decided QGIS was a dead end for my purpose.

R/terra

My next step was to try to convert survey units from a shapefile to SQL INSERT queries using R and the terra package. In short, this effort also failed with invalid WKTs, similar to QGIS.

Bad spatial features?

It's possible that all the failures described above were due to a bad source shapefile. Since I accomplished my survey unit database import goal using ArcGIS as described below, I made no attempt to go back and try the above efforts again with different files to see if the problem was my source file, rather than the tools I used. I have a suspicion that QGIS and terra generated WKT with the rings in the wrong order ("left hand/foot rule"; areas lying to the left-hand side of the line drawn between the points are considered to be inside the Polygon), or were somehow unclosed polygons.

ArcGIS Pro/ArcPy (preferred method)

The ArcPy script included in this section is the preferred method for transferring moose survey units to the Moose SQL Server database. The goal of the script is to generate INSERT queries, one per feature, to insert the survey unit polygons into the SurveyUnits database table. Unlike the alternative methods described above, ArcPy converts polygons into WKT that is acceptable to SQL Server (see the script for the details of how WKT is used in INSERT queries).

Procedure
Below is an example ArcPy script that can be modified to translate survey unit polygons into SQL INSERT queries suitable for transferring data into the Moose SQL Server database. The script interrogates a shapefile of survey units and cycles through the rows one by one and converts the unit polygons into an SQL script of INSERT queries that can be executed through SQL Server Management Studio. Each polygon is converted to its Well Known Text representation and from there to an SQL Geography entity.

Prerequisites
SQL Server Management Studio
ArcMap/ArcGIS Pro
datawriter privileges on the moose monitoring database

```
#Written by Scott D. Miller
#National Park Service, Arctic and Central Alaska Inventory and Monitoring programs
#2021-08-24

# Description: This script was written to convert NPS moose monitoring survey units in
shapefile format to SQL INSERT queries suitable for executing against
# the ARCN/CAKN Moose database. The goal is to reproduce the survey polygons in the
shapefile in the database's SurveyUnitSets and SurveyUnits tables.
# For assistance, contact the Network data manager.

# How to use:
# Point the Shp variable (see 'Start here' below) to your shapefile of survey units
# In the 'for row in arcpy.da.SearchCursor(FeatureClass,['FID','UNIT','Shape@WKT']):'
line below, in the second item in the columns list,
# change the name of the column holding your survey unit identifier. It may be SU, or
UNIT or ID or some other name. This identifier must match the identifier in your data (A
gotcha to avoid: some shapefiles may contain an ID and an SUID column - confusing the two
will mismatch the survey data and survey units).
# Run the script in the Python window of ArcGIS
# The output SQL insert queries script will be in the same directory as the input
shapefile and will have the same name but with '.sql' appended.
# Open the output .sql script in SQL Server Management Studio
# Modify the .sql script as needed and execute to insert the survey units.
# IMPORTANT: The insert queries are wrapped in a transaction. The transaction must be
closed out by executing COMMIT if all the queries insert successfully, or ROLLBACK if
anything fails.
# Failure to close out the transaction will cause the database to hang, rendering it
useless until administrative action is taken to fix the problem.

# Imports
import arcpy
import getpass
import re
import os

# Function to export a featureclass as a comma separated values text file.
def ExportSQLScript(FeatureClass,SurveyUnitSetName):
    try:

        # The output SQL script file.
        OutputFile = FeatureClass + ".sql"

        # if the output file exists already then delete it.
        if os.path.exists(OutputFile):
            print("File exists: " + OutputFile + '. Deleted')
            os.remove(OutputFile)
        OutputFile = open(OutputFile,'a')

        # loop through the data rows, create Sql insert queries and add them to the
output file.
        OutputFile.write("-- Moose survey units import script.\n")
        OutputFile.write("-- Source FeatureClass: " + FeatureClass + "\n")
        OutputFile.write("-- Generated by " + getpass.getuser() + " on " +
str(datetime.datetime.now()) + "\n")
        OutputFile.write("\n")
        OutputFile.write("USE Moose\n")
        OutputFile.write("Declare @SurveyUnitSet Varchar(100),@UnitsIRMARefCode Int\n")
        OutputFile.write("Set @SurveyUnitSet = '" + SurveyUnitSetName + "'\n")
        OutputFile.write("\n-- Utility queries. Uncomment and use as needed\n")
        OutputFile.write("-- Insert the parent SurveyUnitSet record for the survey units,
if it does not exist already\n")
        OutputFile.write("-- INSERT INTO SurveyUnitSets(SurveyUnitSet)
VALUES(@SurveyUnitSet)\n")
```

```
        OutputFile.write("-- SELECT * FROM SurveyUnits WHERE SurveyUnitSet =
@SurveyUnitSet ORDER BY ID\n")
        OutputFile.write("-- DELETE FROM SurveyUnits WHERE SurveyUnitSet =
@SurveyUnitSet\n\n")
        OutputFile.write("BEGIN TRANSACTION -- ROLLBACK COMMIT -- You must rollback or
commit the insert queries in this transaction or the database will be left hanging.\n\n")


        # Loop through the feature class's rows and convert the units to WKT and create
an insert query.
        # Make sure the column names below match those in the source shapefile.
        RowCounter = 0

        # Make sure FID or OBJECTID is the zeroeth columnname, the unit identifier is the
first column and the second column is Shape@WKT.
        # Example: for row in arcpy.da.SearchCursor(FeatureClass,
['FID','UNIT','Shape@WKT']):
        for row in arcpy.da.SearchCursor(FeatureClass,['FID','UNIT','Shape@WKT']):

            # Get the row items into variables to substitute into the SQL queries below.
            ID = str(row[1]).strip()
            WKT = str(row[2])

            # Build up a Geography for the polygon using Well Known Text representation
of the Feature.
            Geog = "geography::STGeomFromText('" + WKT + "', 4326)"
            InsertQuery = "INSERT INTO SurveyUnits(SurveyUnitSet,ID,Feature)
VALUES(@SurveyUnitSet,'" + ID + "'," + Geog + ");"

            # Write the INSERT query to the output script file.
            OutputFile.write(InsertQuery +'\n')

            # Increment the row counter.
            RowCounter = RowCounter + 1
        del row

        # Output some more information for the user.
        OutputFile.write("-- " + str(RowCounter) + " rows.\n")
        OutputFile.write("-- Execute the output SQL script in SQL Server Management
Studio to insert the survey unit polygons.\n")
        OutputFile.close

    except Exception as e:
        ExMessage = '--------------------------------------------------\n-- ERROR:
Export failed for FeatureClass: ' + FeatureClass + ' ' + str(e) +
"--------------------------------------------------\n"
        print(ExMessage)
        OutputFile.write(ExMessage)

# Start here
try:
    # Shp is the input shapefile whose polygons should be converted to an SQL INSERT
query
    Shp = "C:/Work/VitalSigns/CAKN Moose/Units/1978-1996  DENA Moose Survey
Units/DENA1978.shp"
    print('Processing: ' + str(Shp))

    # This will be the unique identifier in the Moose database's SurveyUnitSets table
    SurveyUnitsetName = '1978 DENA Winter Moose Survey Units'

    # Convert the shapefile's features to a script of INSERT queries
    ExportSQLScript(Shp,SurveyUnitsetName)

except Exception as e:
```

```
    ErrorMessage = str(e)
    print(ErrorMessage)
```

If the script ran without failure you should have an SQL script file in the same directory as the source file, named the same but with the .sql suffix. Execute this script in SQL Server Management Studio to insert the polygons. NOTE: The SQL insert queries are wrapped inside a transaction - be sure to COMMIT if all the records inserted successfully, or ROLLBACK if any of them failed. Failure to do so will hang the database :(

Quality control

Quality control

Quality control (QC) procedures ensure that data in the database meet expectations. Defects should be repaired or documented.

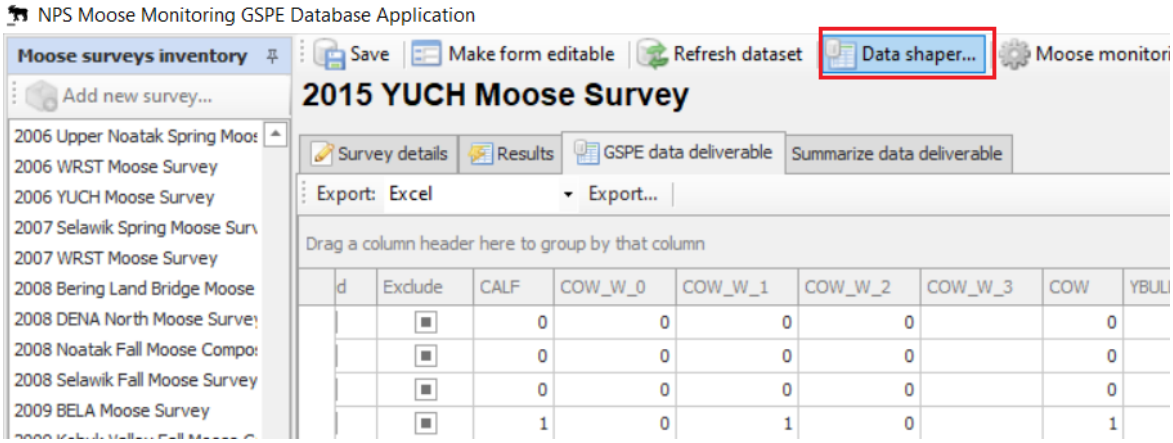
Primary QC methods involve visual record scanning, database interrogation via scripts, QC queries, and stored procedures. The primary tools for ensuring data quality are the *RunQCOnASurvey* and *DataQualityReport* stored procedures. Read about these tools in this section.

Dashboard

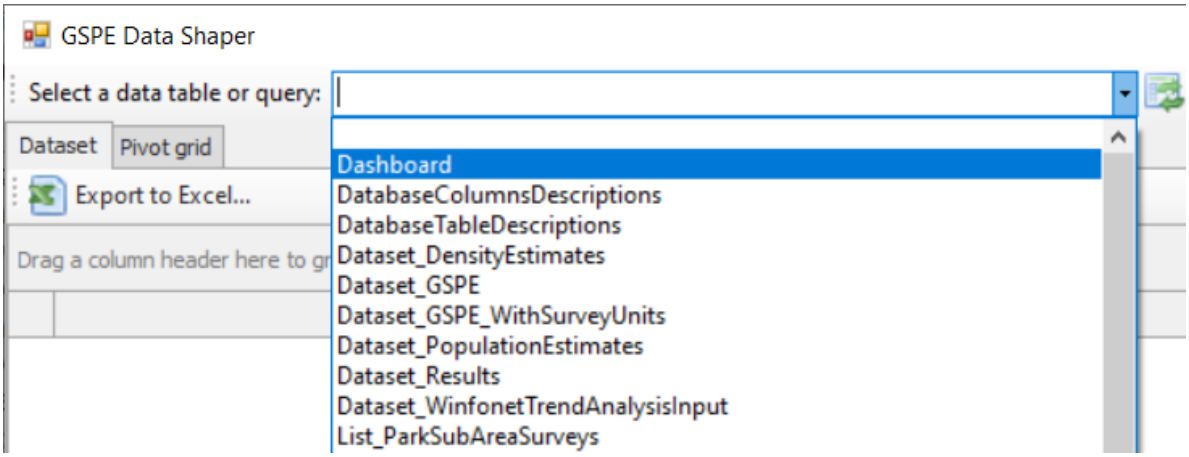
Dashboard query

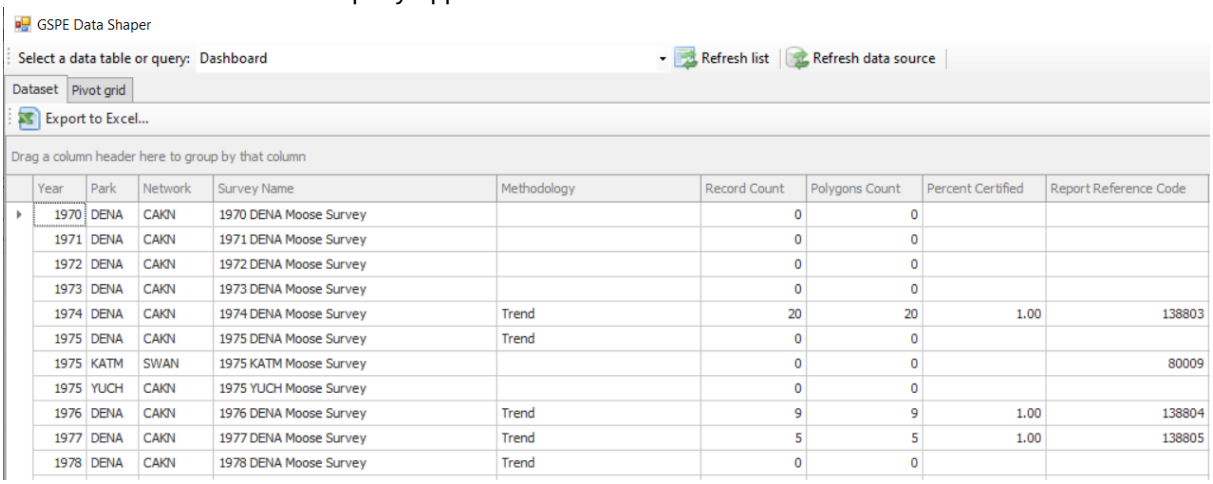
The Dashboard query provides a good summary of the database contents and quality. Run the Dashboard query through (SSMS) or the Moose Monitoring Database Application's Data Shaper tool.

Procedure

1. Open the Moose Monitoring Database Application
2. Click Data shaper in the toolbar
3. 

4. Select Dashboard from the list of available database queries:

5. 

6. The results of the Dashboard query appear:
7. 

Year	Park	Network	Survey Name	Methodology	Record Count	Polygons Count	Percent Certified	Report Reference Code
1970	DENA	CAKN	1970 DENA Moose Survey		0	0		
1971	DENA	CAKN	1971 DENA Moose Survey		0	0		
1972	DENA	CAKN	1972 DENA Moose Survey		0	0		
1973	DENA	CAKN	1973 DENA Moose Survey		0	0		
1974	DENA	CAKN	1974 DENA Moose Survey	Trend	20	20	1.00	138803
1975	DENA	CAKN	1975 DENA Moose Survey	Trend	0	0		
1975	KATM	SWAN	1975 KATM Moose Survey		0	0		80009
1975	YUCH	CAKN	1975 YUCH Moose Survey		0	0		
1976	DENA	CAKN	1976 DENA Moose Survey	Trend	9	9	1.00	138804
1977	DENA	CAKN	1977 DENA Moose Survey	Trend	5	5	1.00	138805
1978	DENA	CAKN	1978 DENA Moose Survey	Trend	0	0		

Sort, group, filter and/or summarize the information as needed.

Run quality control checks on data from a single survey

Run quality control checks on data from a single survey

It may be useful to import data into the moose monitoring database and then perform automated quality control checks. This can be done using the *RunQCOnASurvey* stored procedure. The example below shows a QC run on the 2015 YUCH Moose Survey dataset.

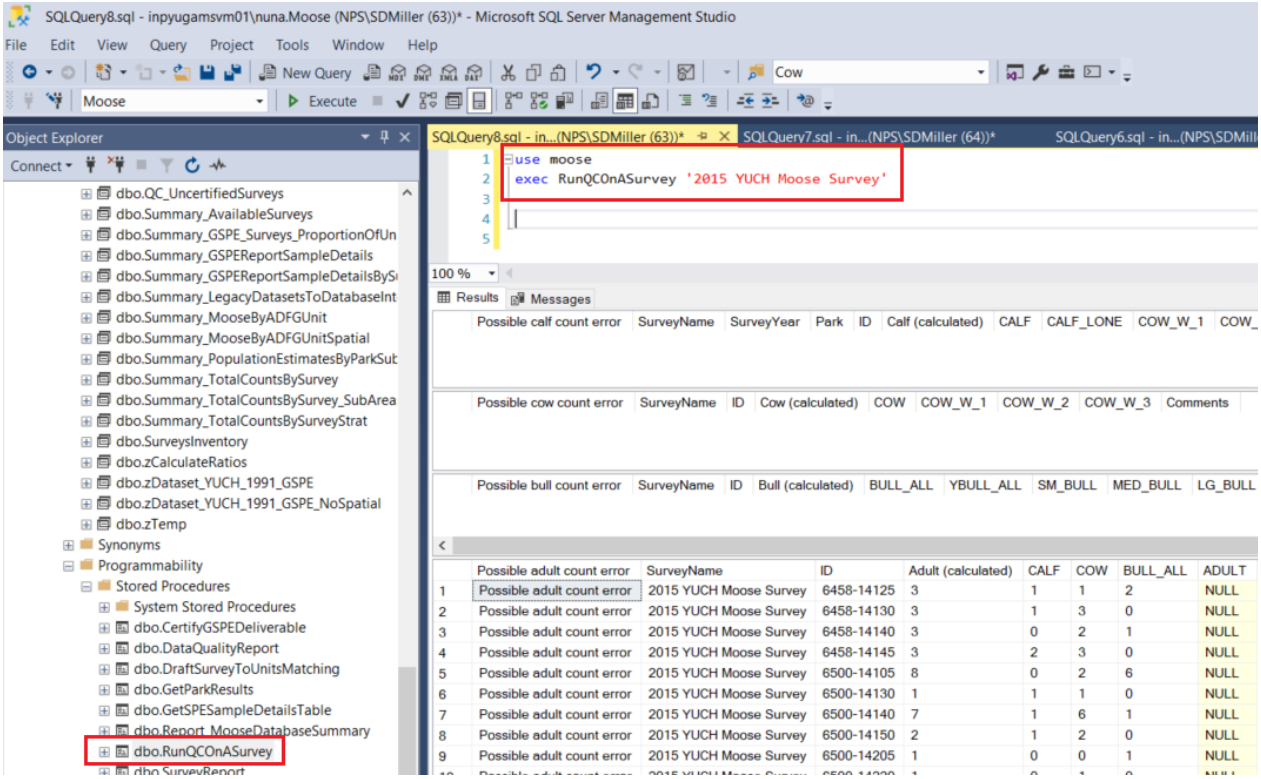
IMPORTANT NOTE: The automated QC checks described below are a *starting point* and the results should be interpreted based on knowledge of the survey before making any corrective edits. Surveys are conducted in different ways and using calculation rules that may be different from the automated database checks. Some surveys collect more or less detailed information about sex and age composition which may throw off QC checks. Project biologists or may include or exclude animals tagged UNKNOWN, for summing total ADULT or MOOSE values, for example. Use your judgement before editing the data.

Procedure

1. Open SSMS
2. Connect to the moose monitoring database
3. Click New Query
4. Type or copy the following into the new query file substituting your SurveyName attribute for 'Your survey name goes here':

5. `use moose`
6. `exec RunQCOnASurvey 'Your survey name goes here'`
7. Execute the procedure

Example QC run showing that the [Adult] attributes have not been calculated in the 2015 YUCH Moose Survey. Other QC checks returned no problems (you should still verify the data visually, by summing and comparing to the source data file, and comparing the results to any report generated on the dataset).



Global database QC checks

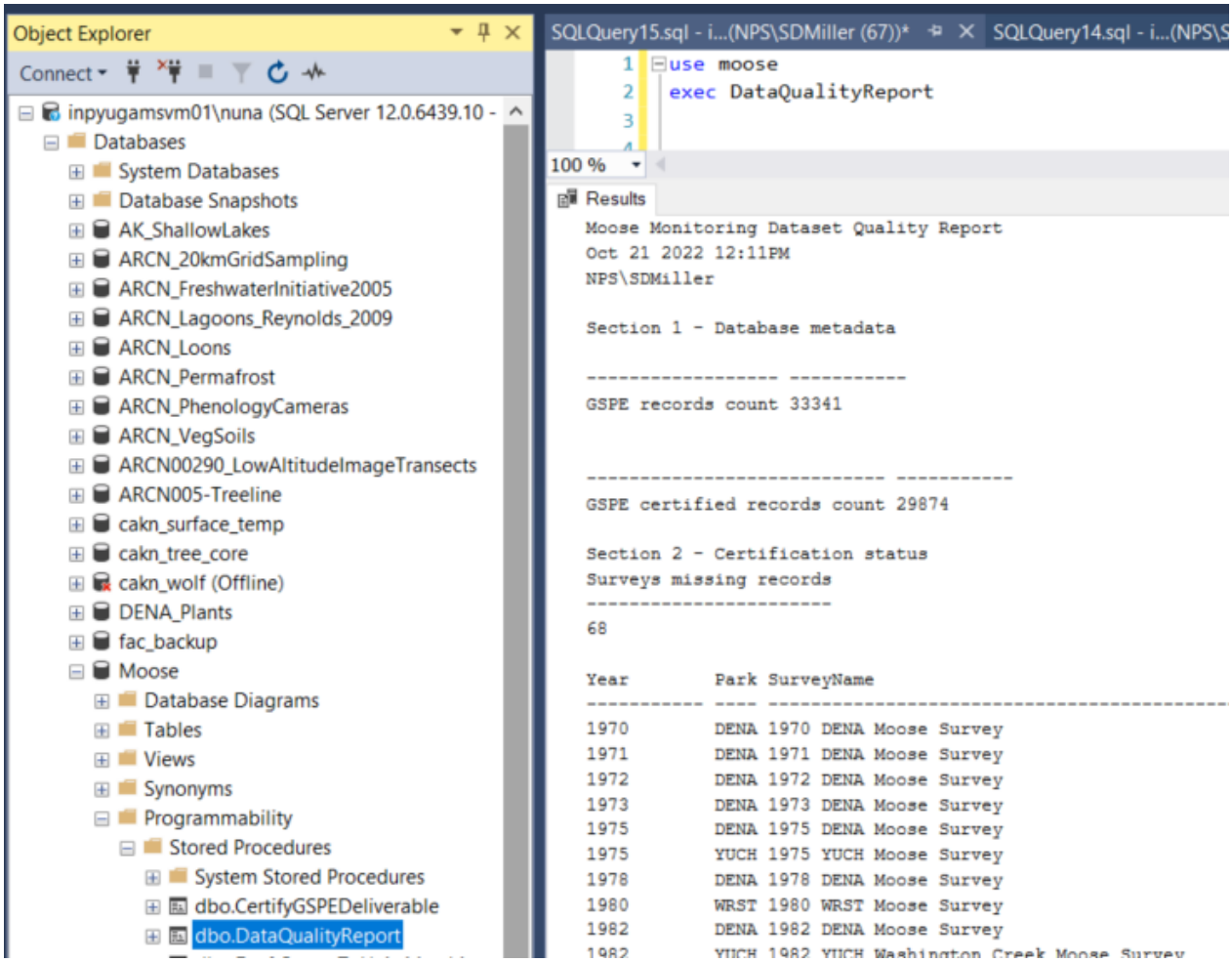
Global database QC checks

The DataQualityReport stored procedure provides a good starting point for checking data quality in the moose database.

IMPORTANT NOTE: The automated QC checks described below are a *starting point* and the results should be interpreted based on knowledge of the survey before making any corrective edits. Surveys are conducted in different ways and using calculation rules that may be different from the automated database checks. Some surveys collect more or less detailed information about sex and age composition which may throw off QC checks. Project biologists or may include or exclude animals tagged UNKNOWN, for summing total ADULT or MOOSE values, for example. Use your judgement before editing the data.

DataQualityReport stored procedure

Note: for best results, right click the query pane and select **Results to -> Results to text**; otherwise the queries will be shown as grids and some context of what QC process is being run will be lost.



Examine the output to help isolate possible errors for repair or documentation.

Some results shown in the output may not be errors.
Some records may be flagged not because there is a problem but because of the way the survey was conducted. An example is a population survey that only counts [Adult] and [Calf] data. The [Adult] column will not equal [BULL_ALL] + [COW] in such cases, and may be flagged.

Another example of confused results is where the biologist counted [UNKNOWN] in the [ADULT] tally, throwing off the QC check calculation where that sort of summing is not allowed.

Use expert judgement and please do not try to fix other people's datasets, but rather communicate such issues with the respective Park biologist.

Dataset certification

Dataset certification

Data certification is an important tool to communicate data quality to data consumers. The moose monitoring dataset consists of data from many surveys done over many parks. In many cases datasets have been recovered many decades after institutional knowledge is gone, often without the benefit of a summary report or other forms of documentation. Certification is accomplished by tagging each record with a CertificationLevel. Three levels of certification are allowed:

Raw	Records that have not undergone any quality control procedures. These records are not

	published or used analytically.
Provisional	Records that have undergone rudimentary quality control or have been received directly from a park biologist or the Winfonet system, but have not been fully processed and approved for analytical purposes. These records are not published. They may be used provisionally to develop in-house products.
Certified	Data that have been approved for analysis. These records have been processed for quality, validated against a source dataset, a report or other summary, have defects documented, and are of known quality. These records are published.

Certification process

GSPE records are certified at the record level. This means that each record in the GSPE data table has a column called CertificationLevel (and also CertifiedBy and CertificationDate).

CertificationDate	CertifiedBy	CertificationLevel
2021-10-29 13:21:45.070	NPS\SDMiller	Certified
2021-10-29 13:21:45.070	NPS\SDMiller	Certified
2021-10-29 13:21:45.070	NPS\SDMiller	Certified
2021-10-29 13:21:45.070	NPS\SDMiller	Certified
2021-10-29 13:21:45.070	NPS\SDMiller	Certified
2021-10-29 13:21:45.070	NPS\SDMiller	Certified

During data ingestion and quality control these columns should be null but CertificationLevel should be set to 'Raw' for raw data or 'Provisional' if the data are likely good enough for provisional analysis but not fully processed.

Verify the dataset ingested into the moose monitoring database exactly matches the GSPE deliverable that was submitted and analyzed in Winfonet. An alternative check for old datasets for which a report exists is to attempt to re-generate any summary tables in the report with the ingested data. Any data failing QC checks should have CertificationLevel set to 'Raw'. If the data all check out issue the following Transact-SQL command, substituting the name of your survey.

```
USE Moose
exec CertifyGSPEDeliverable '2015 YUCH Moose Survey'
```

ENSURE THE SURVEYNAME PARAMETER IS CORRECT to avoid contaminating records from other surveys. Previously certified survey datasets are immune to modification because of a trigger, but it is possible to accidentally certify other datasets. If you have any questions contact the ARCN/CAKN data manager.

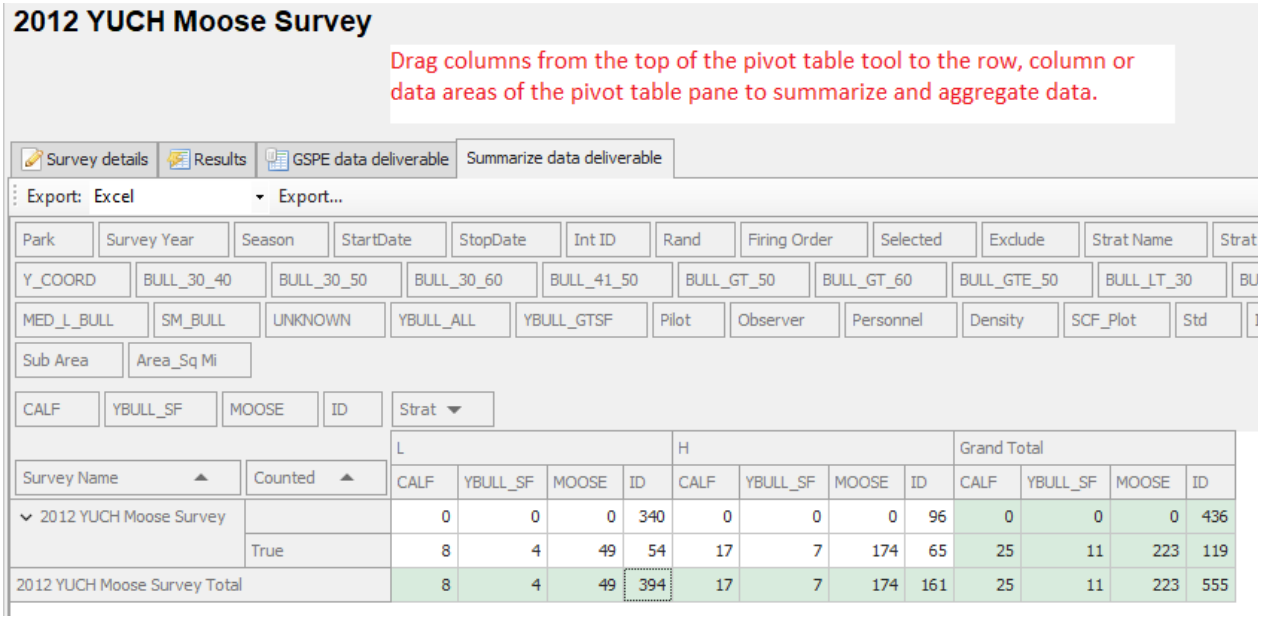
Data analysis

Data analysis

Winfonet is the primary tool for generating moose abundance estimates, calf/bull:cow ratios, etc., but the Moose Monitoring Database application has tools for accessing and summarizing survey data.

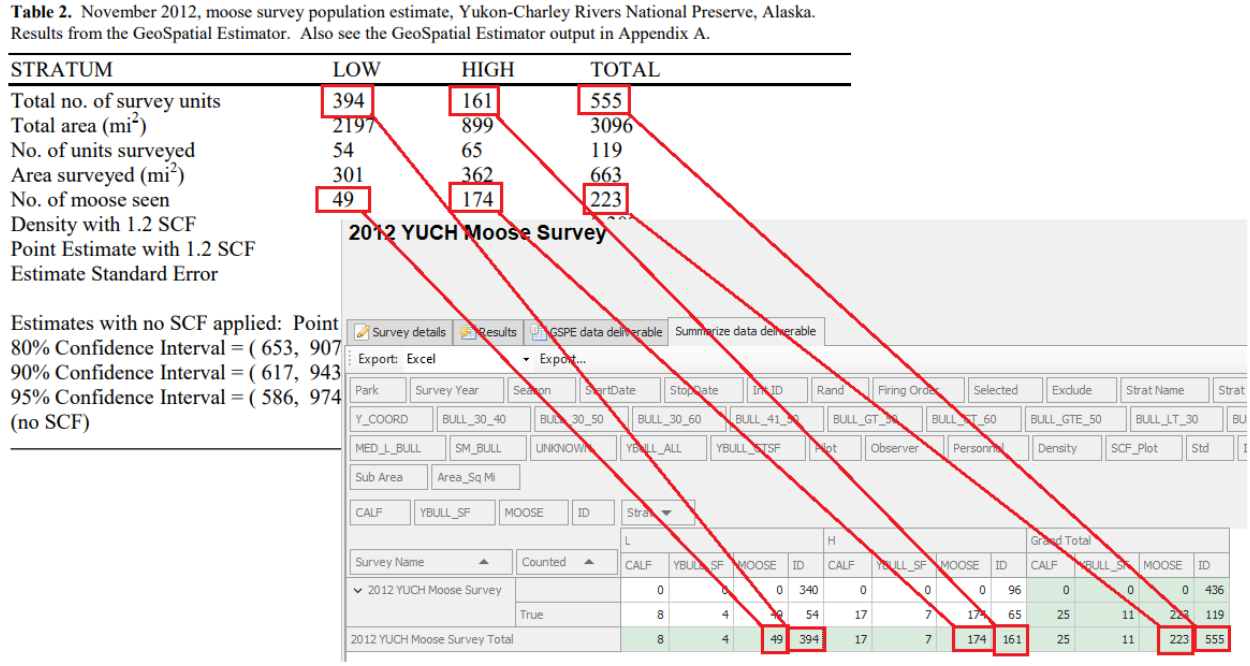
Generating pivot tables

Pivot tables are powerful tools for statistically aggregating data over groups. The Moose Monitoring Database Application offers a tool to quickly generate a pivot table for any survey containing data. Navigate to the survey you would like to summarize, click on the Summarize data deliverable tab and then drag columns from the top of the tool onto the data, rows, or columns area to quickly summarize data. Change the summary type (sum, avg, count, etc.) by right clicking on a column in the data area and selecting a different summary type.



An example of generating summary tables or validating existing summaries

The screenshot below overlays Table 2 of the 2012 YUCH moose survey report with the equivalent summary generated using the Moose Monitoring Database Application's pivot table tool.



Dataset publication

Dataset publication

The moose monitoring dataset is published to the IRMA Data Store whenever new data is added or records are edited. Incrementally cumulative dataset versions will be exported to the Moose Surveys Certified Dataset: NPS Arctic and Central Alaska Inventory and Monitoring Networks Reference at https://irma.nps.gov/DataStore/Reference/Profile/2295396. This task involves exporting the ***Dataset_GSPE_ToShapefile*** view to a shapefile and then uploading the new file to the Reference above. The Dataset_GSPE_ToShapefile view will only contain certified records (records tagged Raw or Provisional will be held back; you may store and work on such records without fear that they will be accidentally published).

Note: Dataset_GSPE is the preferred analytical view for the moose database, but because shapefiles have column length and data type restrictions I had to create the Dataset_GSPE_ToShapefile view; it contains the the exact same data as Dataset_GSPE. A shapefile was chosen as the data publication format because ArcGIS was having problems connecting to SQL Server and the product had to be exported using Ogr2Ogr which does not support proprietary ESRI formats.

Procedure

1. *Export the Dataset_GSPE_ToShapefile database view to a shapefile using one of the methods below.*
2. *Zip the shapefiles.*
3. *Append the date to the end of the zip file. Example: MooseSurveys Database Export Ver. 2022-12-20.zip*
4. *Upload the zip file to the Moose Surveys Certified Dataset: NPS Arctic and Central Alaska Inventory and Monitoring Networks Geospatial Dataset Reference in the IRMA Data Store at https://irma.nps.gov/DataStore/Reference/Profile/2295396.*
5. *Ensure the reference permissions are Internal and the file restrictions are set to only allow access to ARCN/CAKN wildlife biologists, program managers, data managers and technicians (consult the Data Store help on this topic.*

Considerations for publishing sensitive data

Consider carefully the permissions of any dataset published to Data Store. Moose monitoring data falls under the category of 'species of commercial interest', which is one justification for tagging a reference as 'Internal'. Additionally the NPS and Alaska Department of Fish and Game may have one or more data sharing agreements in place which should not be violated. Generally a moose monitoring dataset should only be available internal to the NPS or to specific individuals within the Service. Consult the Data Store help for more information on permissions. Also see Dataset sensitivity and Security.

Generating a dataset for publication

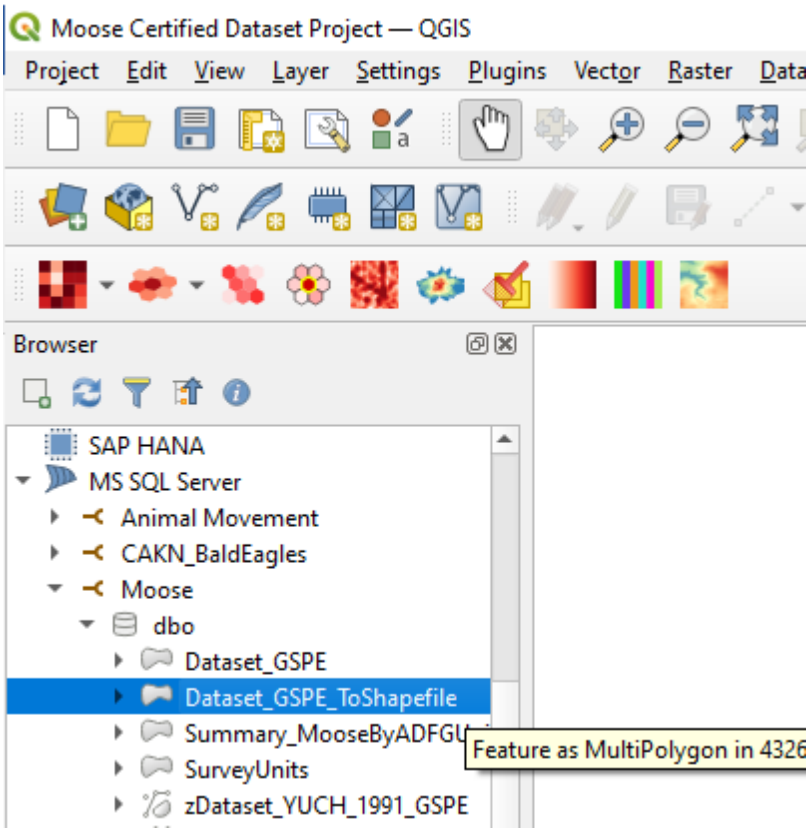
There are multiple ways to export a shapefile from the moose monitoring SQL Server database. Exporting to GIS using ArcGIS Query Layers was the preferred method until late 2022 when something happened inside ArcGIS and Query Layers started failing.

QGIS

As of 2023, this is the preferred method for generating a shapefile of certified moose records.You must have at minimum datareader permissions on the Moose database.

Procedure to generate a shapefile

1. Start QGIS
2. From the main toolbar at the top of the interface select Layer -> Add Layer -> Add MS SQL Server Layer...
3. If necessary, create a new MS SQL Server Connection. Enter the server name and database name 'Moose'. Name the connection Moose.
4. Click List Databases and select Moose.
5. Click Test Connection.
6. If the connection works (QGIS doesn't seem to provide feedback if the test succeeded) then click OK
7. In the Browser pane expand the MS SQL Server Node and look for your new Moose connection.
8. Expand the Moose -> dbo node. It should look like this:



9. Drag the Dataset_GSPE_ToShapefile node onto the map. The dataset may take a while to load since it is over 50k polygons.
10. When the dataset has loaded it will appear in the Layers pane.
11. Right click the Dataset_GSPE_ToShapefile layer in the Layers pane and select Export -> Save Features As...
12. Select ESRI Shapefile as Format
13. Enter a filename (example: Moose_2023-12-28.shp)
14. Select Automatic for Geometry type.
15. Click OK
16. QGIS will take a long time to render the shapefile.
17. Open the shapefile and compare the number of records against the same query in SQL Server Management Studio and make sure the data are the same.
18. Write metadata

ArcGIS

Worth a try, but as of 2023 QGIS seems to have better support for SQL Server.

Ogr2Ogr

In 2022 I was unable to get the Python script to work in ArcGIS Pro/ArcMap. I successfully exported the dataset to a shapefile using ogr2ogr using the following:

```
ogr2ogr -f "ESRI Shapefile" "O:\Monitoring\Vital Signs\Moose\Data\Certified dataset for
```

```
publication\MooseSurveys.shp"
"MSSQL:server=inpyugamsvm01\nuna;database=Moose;tables=GSPE;trusted_connection=yes" -sql
"SELECT SurveyName, ID, GEOMETRY::STGeomFromWKB(Feature.STAsBinary(), 4326) AS Feature,
SubArea, Season, Methodology AS Method, DateCounted AS Date, IntID, Selected, Counted,
Exclude, Strat, ADULT, BULL_30_40,                    BULL_30_50, BULL_30_60, BULL_41_50,
BULL_ALL, BULL_ADULT, BULL_GT_50, BULL_GT_60, BULL_GTE_50 AS BULL_GTE50, BULL_LT_30,
BULL_LT_50, CALF, CALF_LONE, COW, COW_W_0, COW_W_1, COW_W_2, COW_W_3, LG_BULL,
       MED_BULL, MED_L_BULL, SM_BULL, UNKNOWN, YBULL_ALL, YBULL_GTSF, YBULL_SF, MOOSE,
SearchMin, Density, SCF_Plot, Std, Int, SurveyUnitGMU AS SU_GMU, AreaSqMi AS SqMi_Calc,
Area_SqMi, GMU, Rand,                    FiringOrder AS FiringOrdr, StratName, StratMoose,
StratTracks AS StratTrack, StratHab, Perc_Not_Flown AS PctNotFlwn, X_COORD, Y_COORD,
Elev_Mean, Elev_Med, Network, SRID, CENTRLAT, CENTRLON, Pilot, Observer,
 GSPEComment AS Comment, SurveyUnitComment AS SUComment, ReportReferenceCode AS RptCode,
ReportLink, DeliverablesDatasetReferenceCode AS DlvrblsCod, CertificationLevel AS
CertLevel, CertifiedBy AS CertBy, CONVERT(Date, CertificationDate) AS CertDate, UniqueID,
 CONVERT(Date, VersionDate) AS Version FROM     Dataset_GSPE" -overwrite -a_srs EPSG:4326
-s_srs EPSG:4326
```

NOTE: For the Ogr2Ogr method to work you must submit the SQL above - The temptation is to create a view using the SQL above and then call that, but it won't work. I believe Ogr2Ogr neets to see the `GEOMETRY::STGeomFromWKB(Feature.STAsBinary(), 4326) AS Feature` to work, otherwise you get errors saying it cannot work with the binary representation of the Geography column. I wrote a view, Dataset_GSPE_ToShapefile to store the SQL in case it's needed again, but know it won't be useful for Ogr2Ogr.

ArcPy

Modify and execute the Python script below in ArcGIS to generate a shapefile of moose monitoring data. Notice that the SQL query is restricted to certified data only.
NOTE: This didn't work as of Nov, 2022. ArcGIS Pro/ArcMap query layers fail to extract all the survey units and data; unknown reason. See Ogr2Ogr solution below.

```
import arcpy

# Point to the database connection file (use Create Database Connection tool in
ArcToolbox)
database_connection_file_path = r"C:\Work\Code\ArcPy Scripts\Moose DB to
GeoDB\MooseDBLocalConnectionFile.sde"

# Get a reference to the main data frame; assumes it's the first and only one
df = arcpy.mapping.ListDataFrames(MXD)[0]

# The spatial query to submit to the QueryLayer
sql = "SELECT  * FROM Dataset_GSPE_ToShapefile"

# Layer name
layer_name = "GSPE"

# Create the QueryLayer
arcpy.MakeQueryLayer_management(database_connection_file_path, layer_name, sql,
"[UniqueID]", "POLYGON", "4326")

# Create an MXD to hold the new QueryLayer
MXD = arcpy.mapping.MapDocument(r"C:\Work\Code\ArcPy Scripts\Moose DB to
GeoDB\MooseUnits.mxd")

# Get a reference to the created QueryLayer
output_layer = arcpy.mapping.Layer(layer_name)

# Add the layer to the main data frame
arcpy.mapping.AddLayer(df, output_layer, "AUTO_ARRANGE")

# Export the layer to a shapefile
```

```
try:
    arcpy.analysis.Select(layer_name, "C:/Temp/zMooseUnits/GSPE.shp","")
except:
    print("An exception occurred exporting the dataset " + layer_name)
```

Write metadata

Use the tools in ArcCatalog to write metadata for the shapefile generated above. Validate the metadata.

Publish to Data Store

https://irma.nps.gov/DataStore/Reference/Profile/2295396

X drive

Idea abandoned. I considered publishing the data to the X drive, but the feedback I got from biologists was that they preferred a more permissions-restricted environment than the X could provide.

Data access

Data access

Most modern data consuming software can connect to the moose monitoring database. This section shows typical connection methods.

Moose Monitoring Database Application

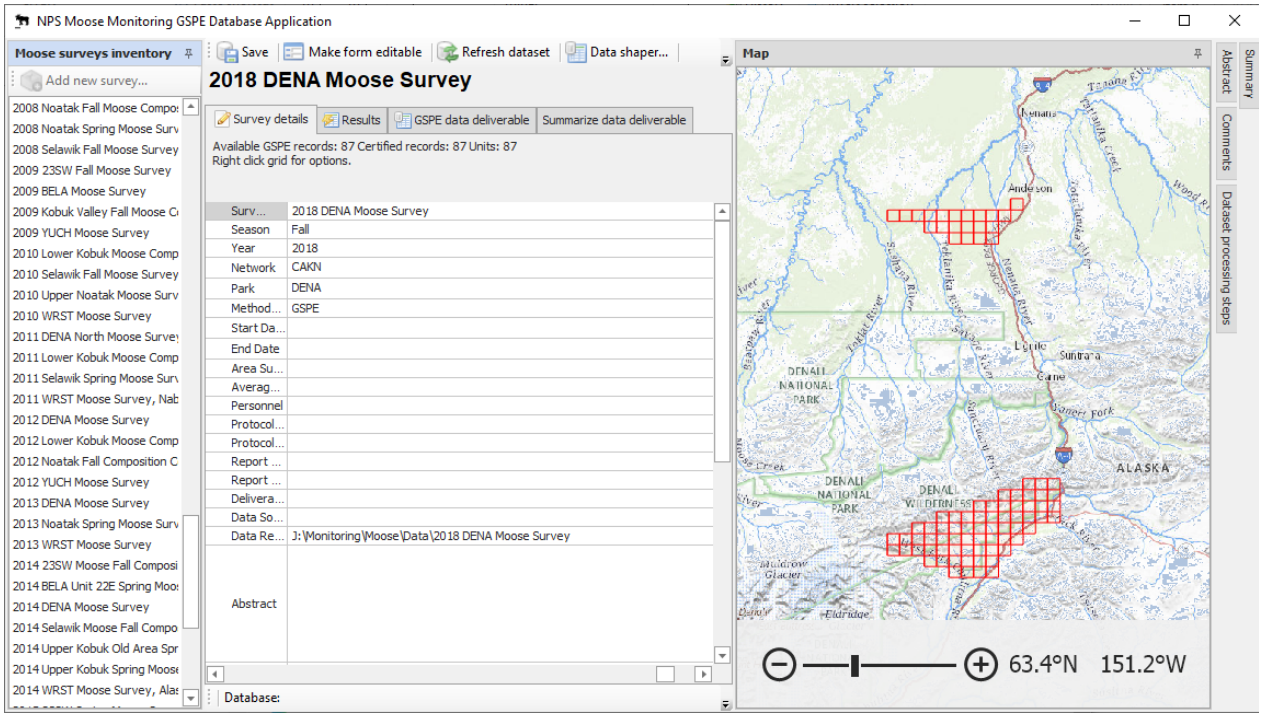
Moose Monitoring Database Application

Prerequisite
Minimally, datareader privileges on the moose monitoring database. Contact one of the ARCN or CAKN data managers for permission.

Procedure

1. Navigate to Help file (https://irma.nps.gov/DataStore/Reference/Profile/2295016).
2. Download the latest release of the software zipped archive (Release yyyy-mm-dd).zip).
3. Unzip the downloaded archive.
4. Double click Moose Monitoring Database Application.exe to start the application.

If all goes well you should see the application open. Note: Depending on your internet speed, VPN, etc., the application may take a little while to open because it pulls the entire database locally; after it is done the application should be fast.



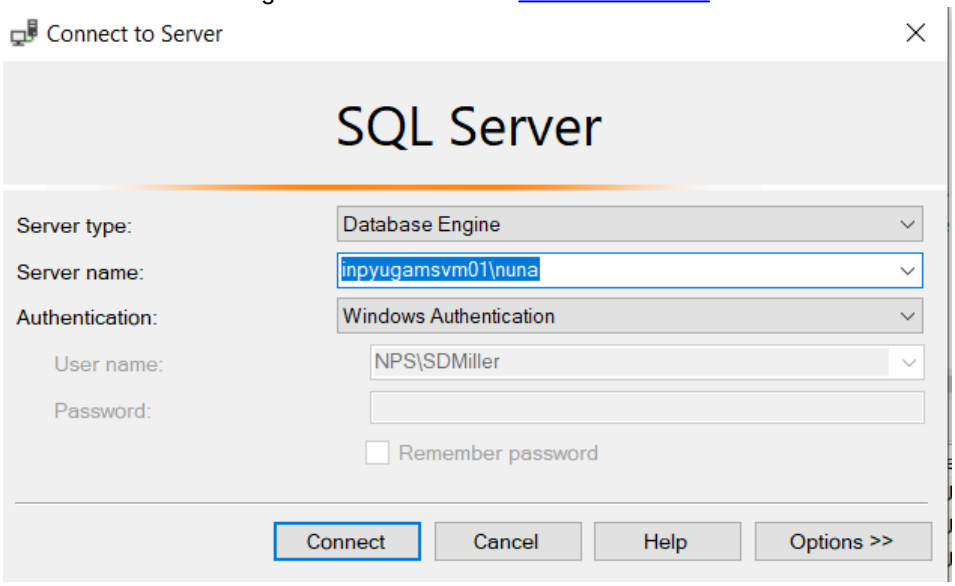
Microsoft Sql Server Management Studio

Microsoft Sql Server Management Studio

Microsoft Sql Server Management Studio (SSMS) is the best tool for administering the database and has good functionality for querying, analyzing and summarizing results. While the software is not difficult to use, it does require an advanced understanding of relational databases.

Process

1. Open SSMS.
2. In the Object Explorer click Connect
3. In the connection dialog enter the server and database details:



4.
5. Expand the server node.
6. Expand the Databases node and select Moose.

7. Consult the SSRS help and online documentation for further information on working with the Moose database.

Microsoft Excel

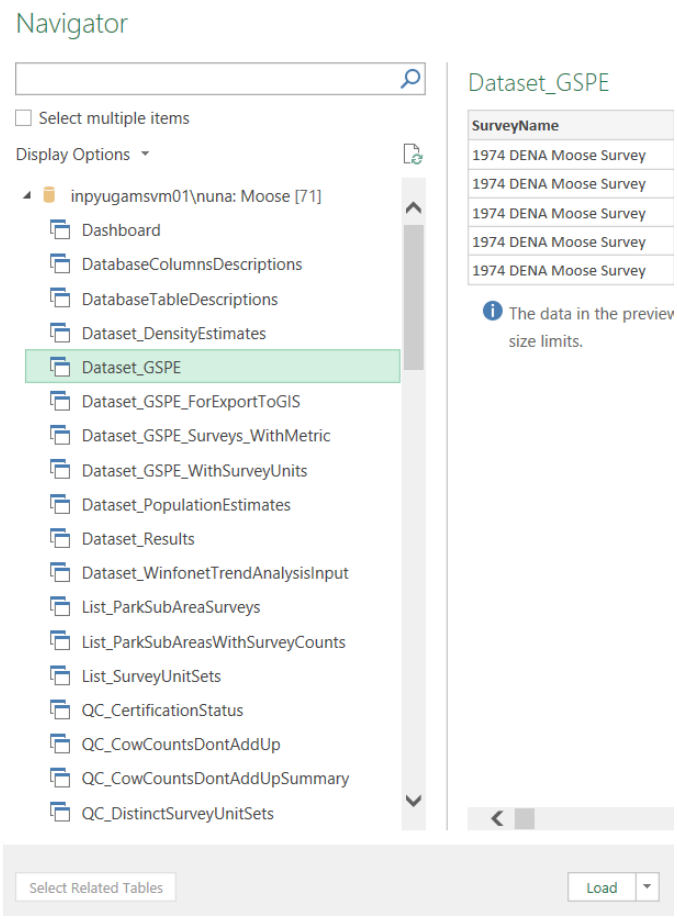
Accessing the Moose database with Microsoft Excel

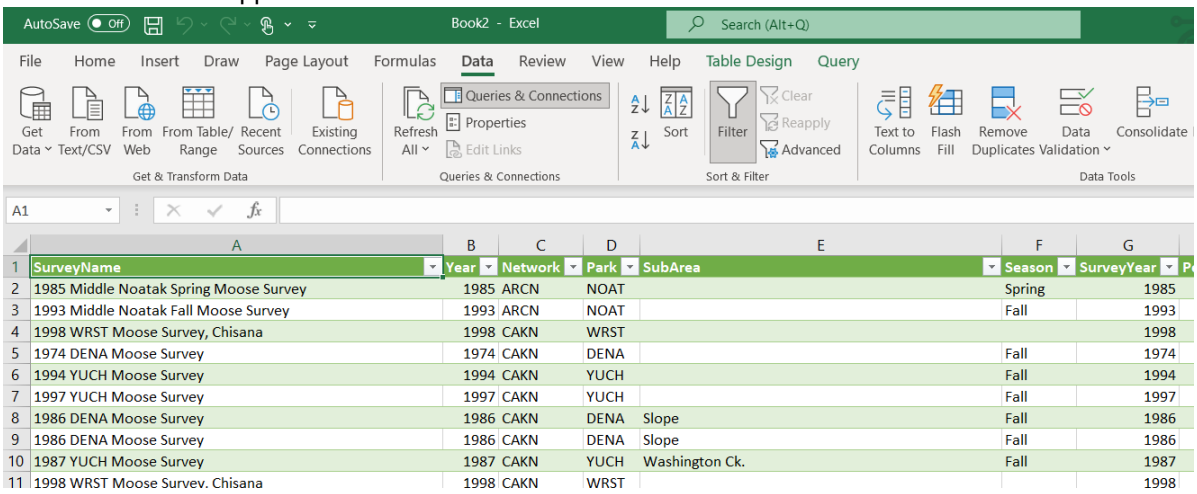
Prerequisite
At least datareader permissions. Contact your data manager.

Process

1. Open an Excel workbook
2. From the Data menu select **Get Data -> From Database -> From Sql Server Database**
3. Enter **inpyugamsvm01\nuna** for the Server and **Moose** for the database.
4. 

5. From the Navigator choose the query you would like to pull. Typically, you will request Dataset_GSPE to pull the whole dataset. Note: this dataset contains tens of thousands of records and may take a minute or so to retrieve depending on your network speed.

6. 

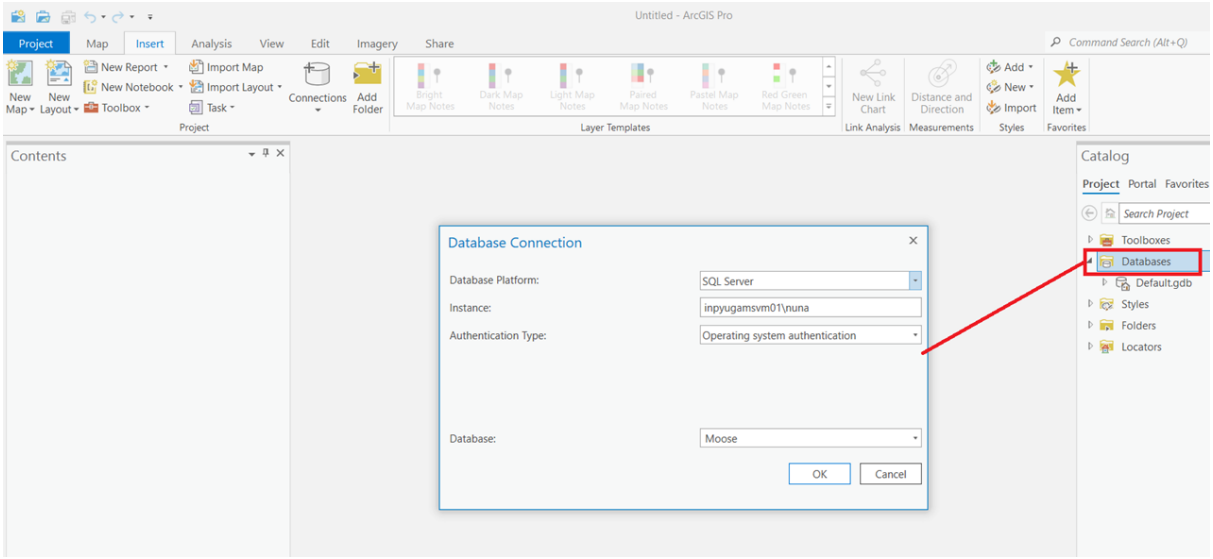
7. Click Load or Transform data.
8. The GSPE dataset appears.
9. 

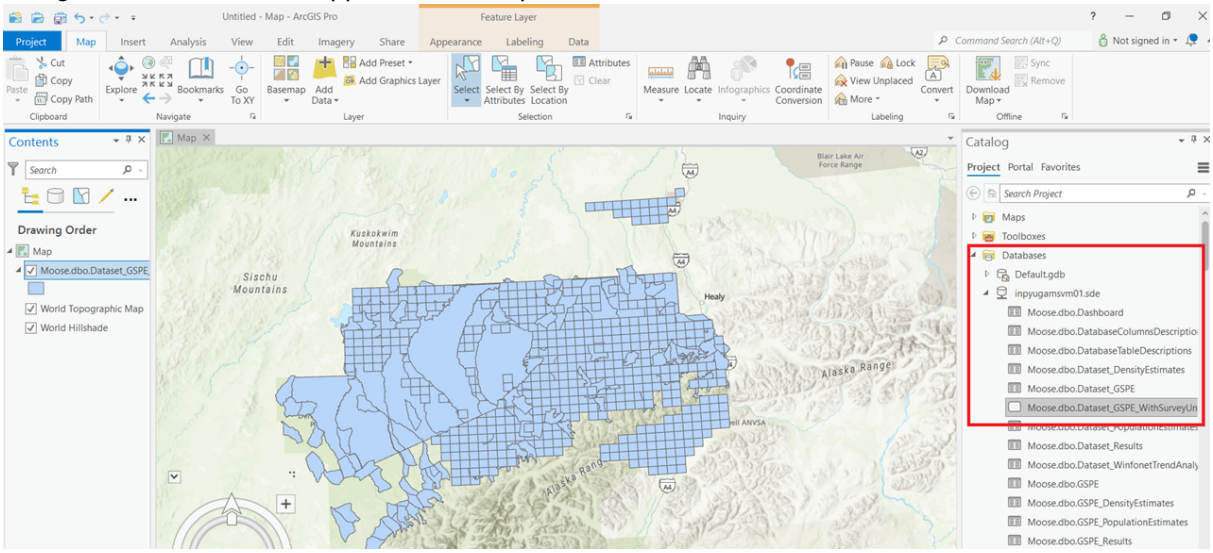
ArcGIS Pro

Accessing the Moose database with ArcGIS Pro

Prerequisite
At least datareader permissions. Contact your data manager.

Process
1. Open ArcGIS Pro

2. Locate the Catalog
3. Right click Databases -> New database connection
4. Enter the server/database information as shown in the screenshot below:
5. 

6. All the database tables and queries become available to you. Select *Dataset_GSPE* and drag it onto the map (there are tens of thousands of records which may take a while to load).
7. If all goes well the data will appear on the map
8. 

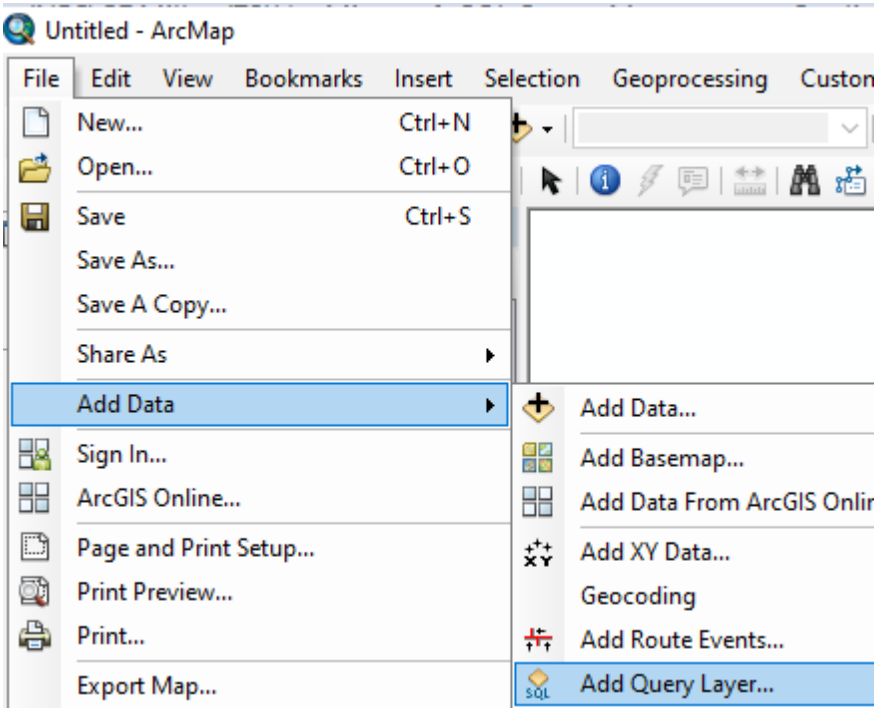
ArcMap

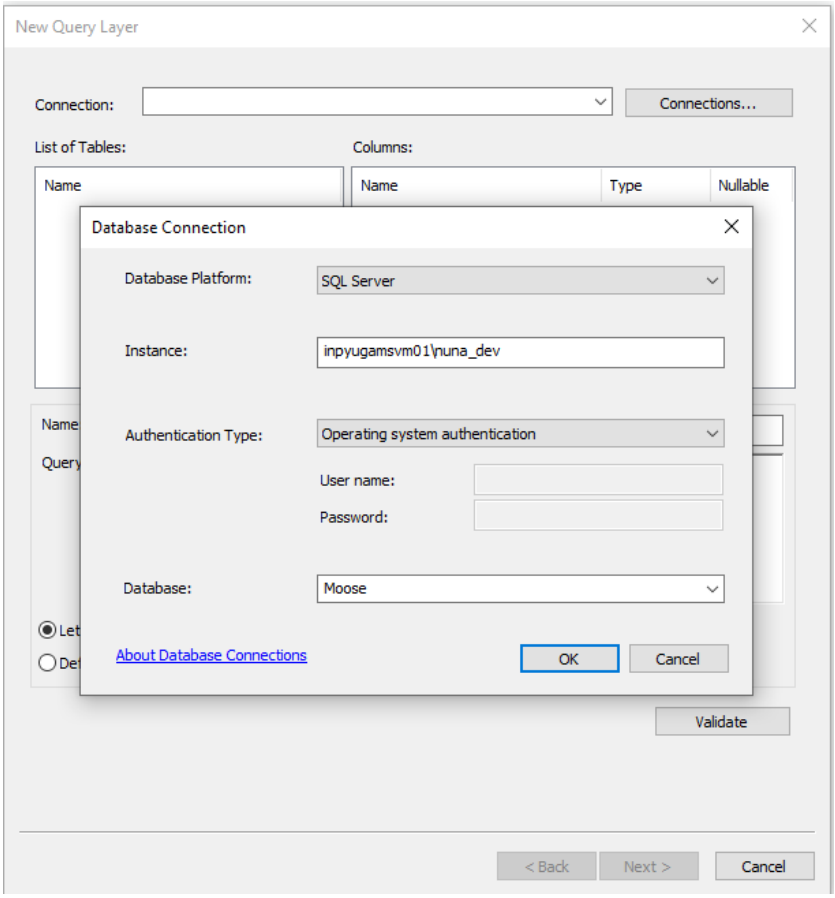
Accessing the Moose database with ArcMap

Using ArcMap to access data in the Moose Monitoring database is possible, but not recommended. It just doesn't work well. There is much better support for SQL Server database connections with ArcGIS Pro. See ArcGIS Pro.

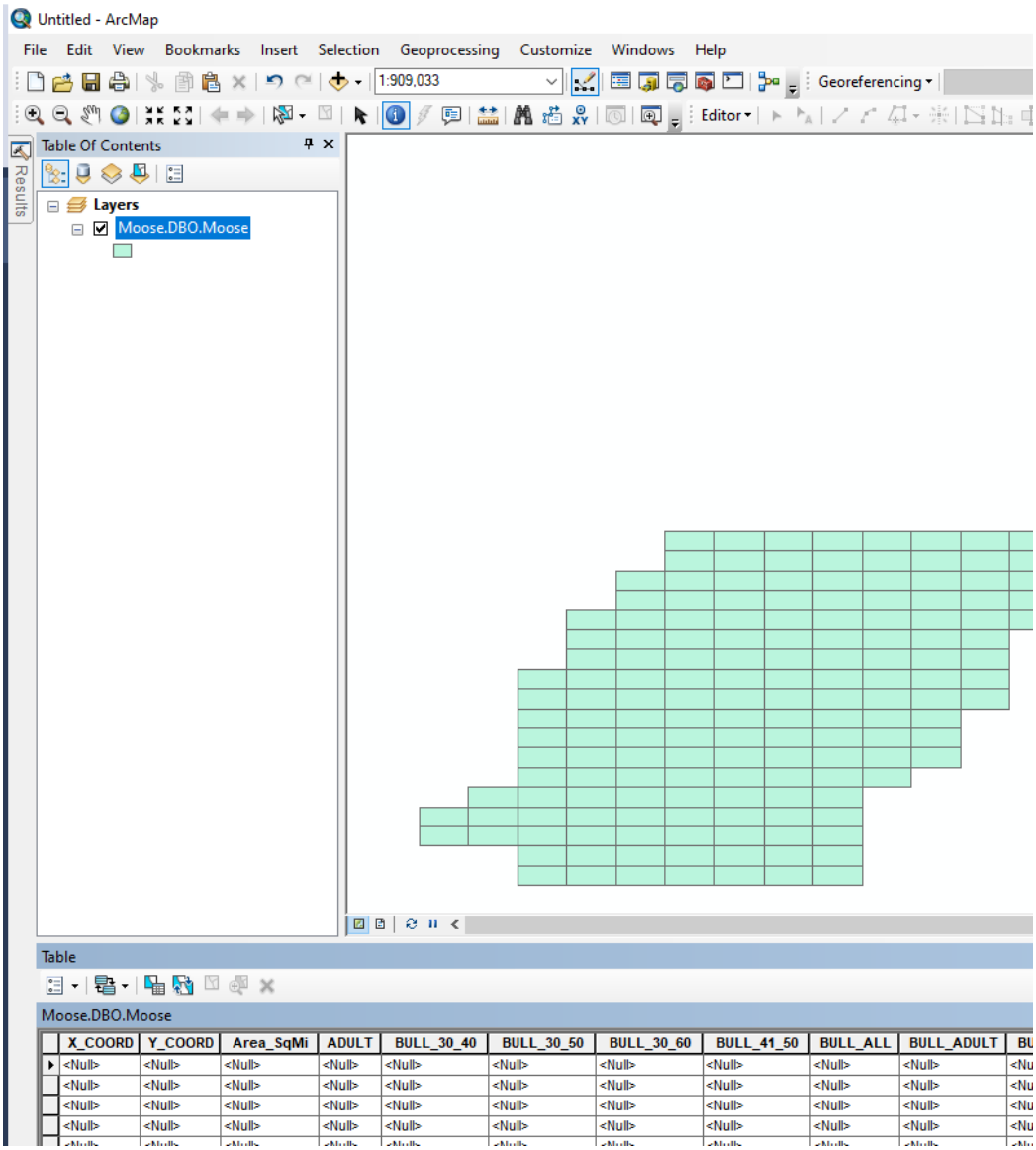
Prerequisite
At least datareader permissions. Contact your data manager.

Process
1. Open ArcMap
2. Add a Query Layer: File -> Add Data -> Add Query Layer

3. 

4. Make a new connection to the Moose database
5. 

6. Enter a query. The most likely query you will want is Dataset_GSPE_WithSurveyUnits. This is the same query as Dataset_GSPE but it includes the survey unit polygons with the GSPE data. Note: You will almost certainly want to filter your query with a WHERE clause to prevent overloading ArcMap with tens of thousands of polygons.
7. An example query: `SELECT * from Dataset_GSPE_WithSurveyUnits WHERE SurveyName='2015 DENA Moose Survey'`
8. Click Validate and move through the rest of the wizard.
9. The data should appear:

10. 

11. Consider exporting the data to a local shapefile or geodatabase to make analysis quicker by eliminating round trip passes over the network which will be slow. Otherwise analyze like any other GIS layer.
12. If you find errors please take the time to communicate them to the I&M data managers so they can be corrected for other users of the dataset!

R

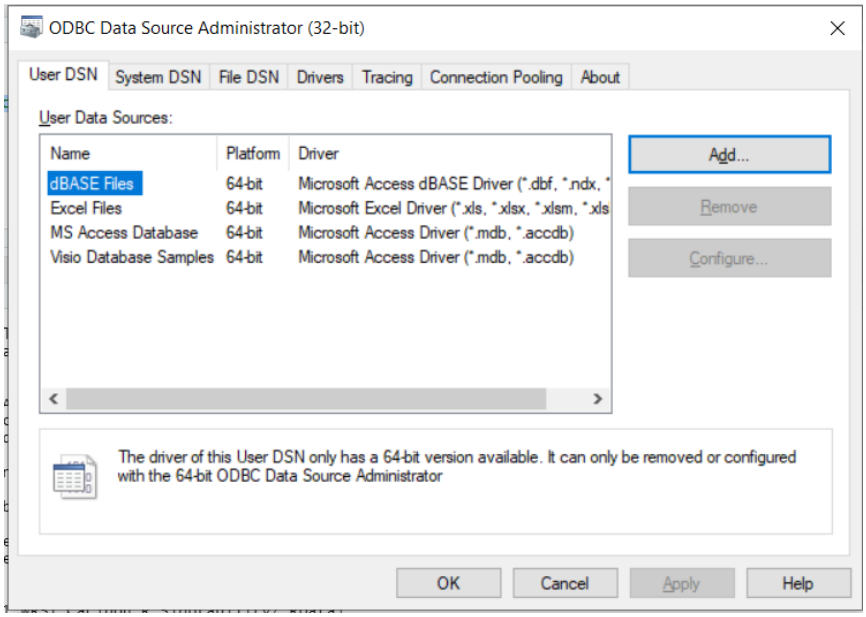
Accessing the Moose database with R

Prerequisite
At least datareader permissions. Contact your data manager.
An ODBC database connection to the Moose database.

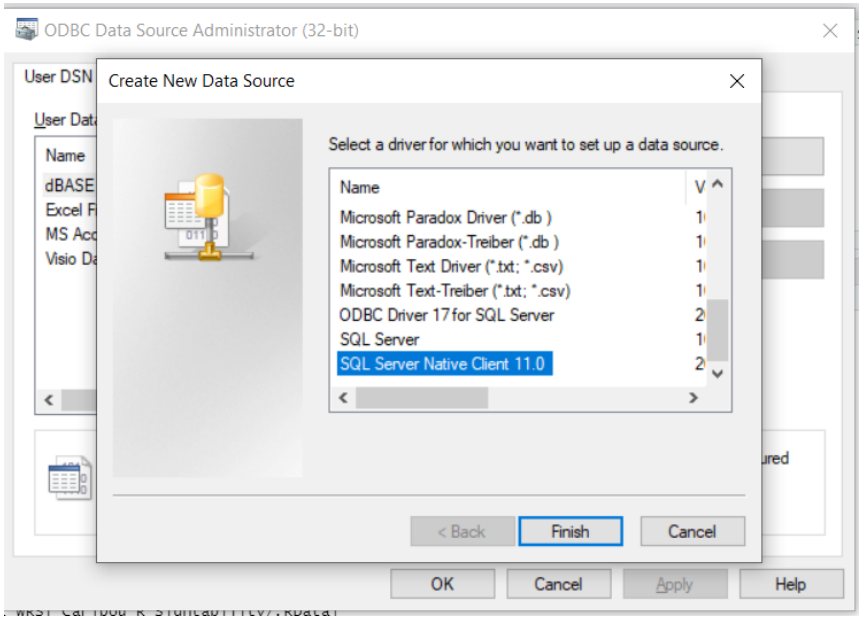
Create an ODBC connection

Process

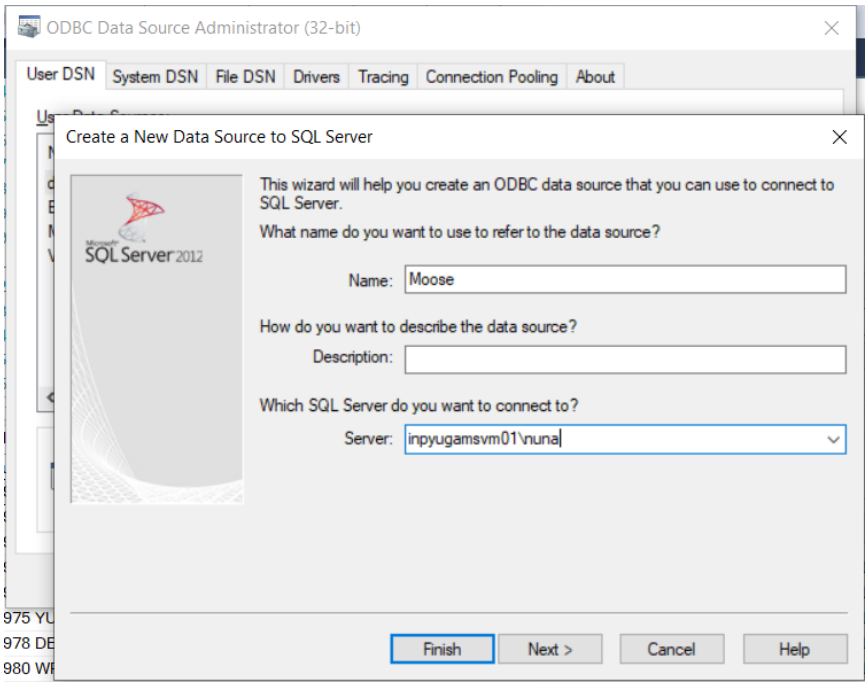
1. Click the Windows key and search 'ODBC' to start the ODBC Data Source Administrator
2. Click through the Wizard to create an ODBC connection to the Moose database
3. Click Add...

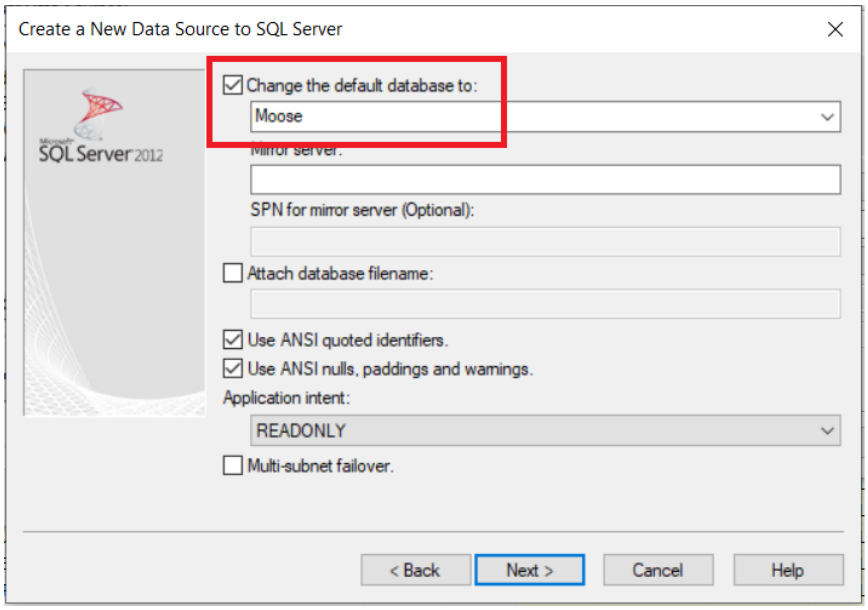


1.
2. Scroll down to Sql Server or Sql Server Native Client.If you don't have on ODBC driver for Sql Server see [Download ODBC Driver for SQL Server]



3.

4. 

5. Ensure the default database is set to Moose
6. 

4. Test the connection. If all goes well you can now connect to the database through this connection using R.

Example R script that connects to the Moose database

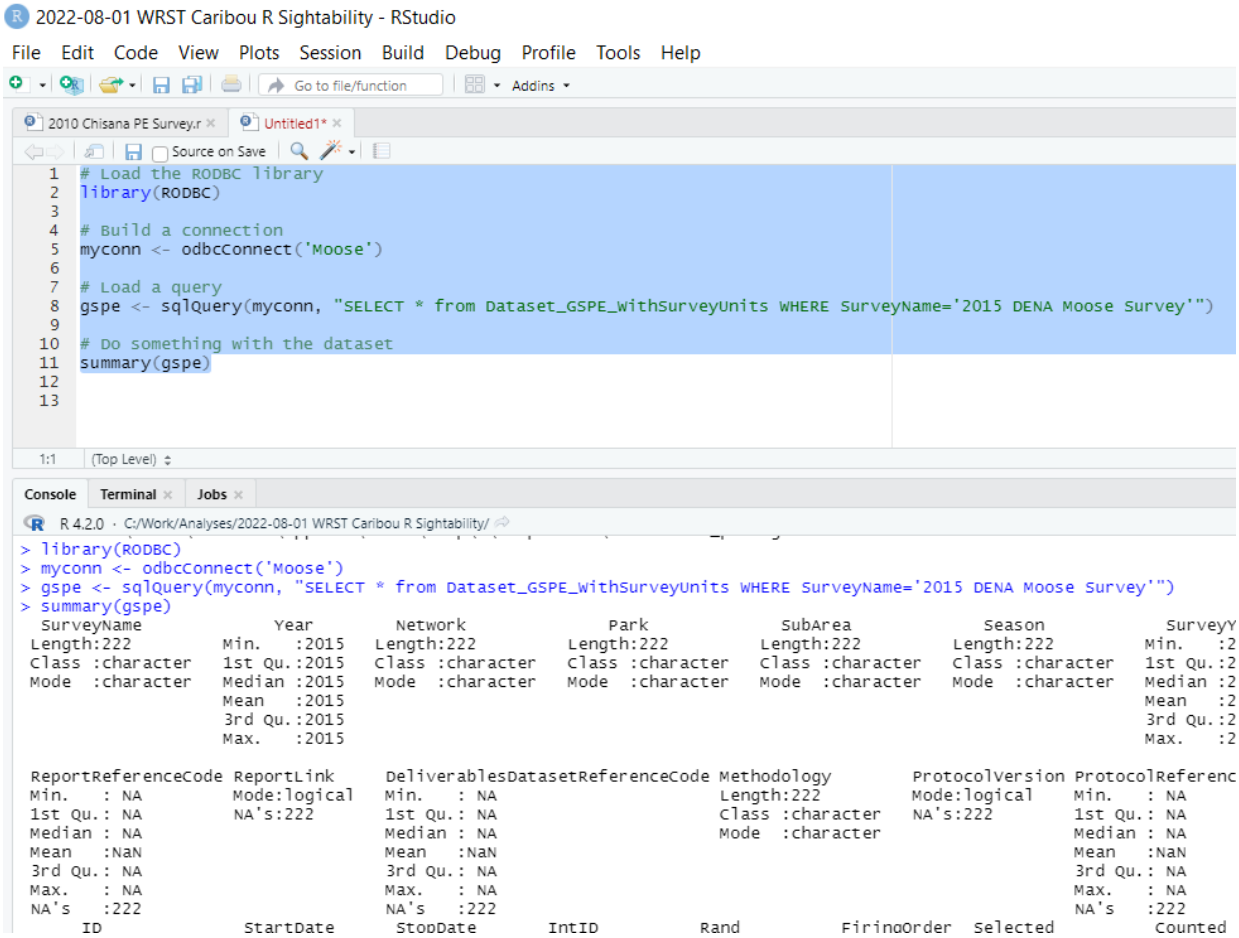
The example below shows how to connect to the Moose database and access the GSPE dataset. Modify the query as needed or omit the WHERE clause to pull the whole dataset (tens of thousands of records).

```
# Load the RODBC library
library(RODBC)

# Build a connection using the Moose ODBC connection
myconn <- odbcConnect('Moose')

# Load a query
gspe <- sqlQuery(myconn, "SELECT * from Dataset_GSPE WHERE SurveyName='2015
DENA Moose Survey'")
```

```
# Do something with the dataset
summary(gspe)
```



Python (ArcPy)

Accessing the Moose database with Python (ArcPy)

Prerequisite
At least datareader permissions. Contact your data manager.

The most likely scenario for accessing the Moose database using Python is to extract some or all of the dataset for analysis. There is a Python script at Dataset publication that loads the GSPE data into ArcMap and exports it as a shapefile. This script can be modified to suit your needs.

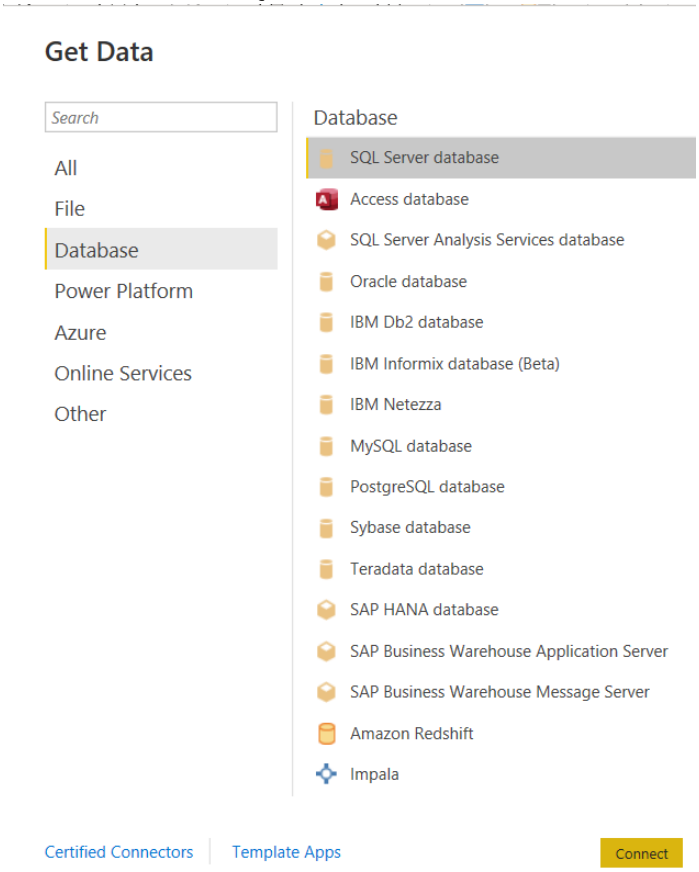
Power BI

Accessing the Moose database with Microsoft Power BI

Prerequisite
At least datareader permissions. Contact your data manager.
Power BI. Download here https://inp2300sqlsas02.nps.doi.net/PBIReports/browse/Development/CAKN

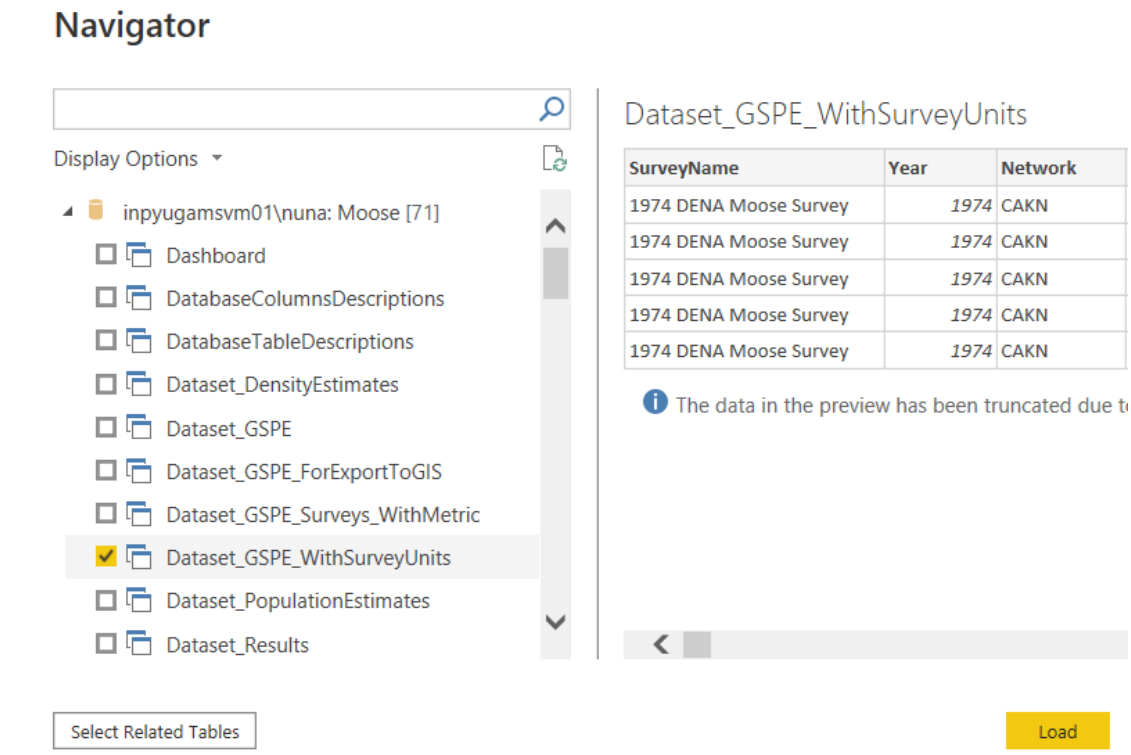
Process

1. Open Power BI
2. Click Get data
3. Select Database -> SQL Server database

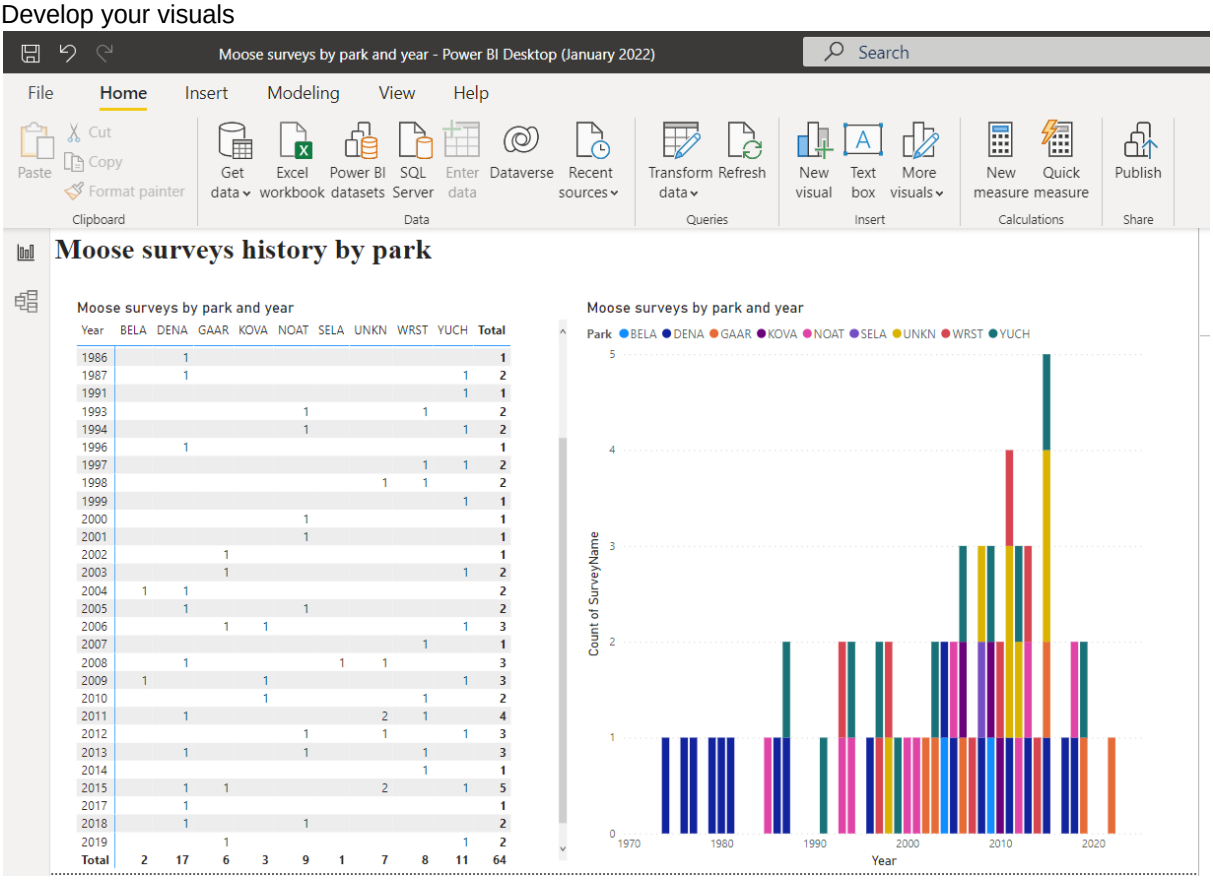


4.
5. Click Connect
6. Enter the server and database names

7.
8. The database Navigator opens:



9.
10. Select Dataset_GSPE_WithSurveyUnits
11. Develop your visuals



12.

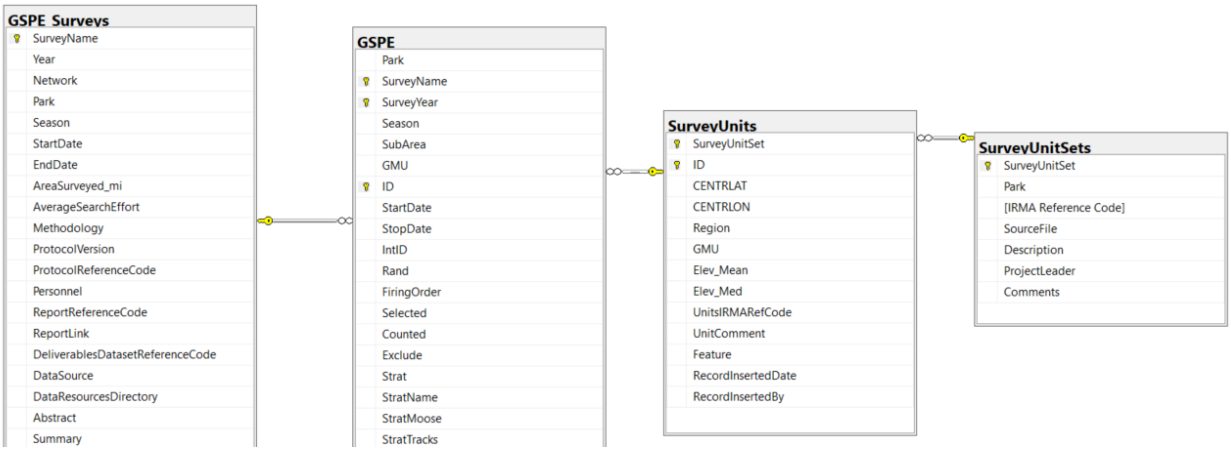
Database documentation

Database documentation

Database details

SQL Server: inpyugamsvm01\nuna
Database: Moose

Data model



Columns descriptions

Table	Column	DataType	Length	Precision	Scale	Nullab
GSPE	ADULT	int	4	10	0	1
GSPE	Area_SqMi	float	8	53	NULL	1
GSPE	BULL_30_40	int	4	10	0	1
GSPE	BULL_30_50	int	4	10	0	1
GSPE	BULL_30_60	int	4	10	0	1
GSPE	BULL_41_50	int	4	10	0	1
GSPE	BULL_ADULT	int	4	10	0	1
GSPE	BULL_ALL	int	4	10	0	1
GSPE	BULL_GT_50	int	4	10	0	1
GSPE	BULL_GT_60	int	4	10	0	1
GSPE	BULL_GTE_50	int	4	10	0	1
GSPE	BULL_LT_30	int	4	10	0	1
GSPE	BULL_LT_50	int	4	10	0	1
GSPE	CALF	int	4	10	0	1
GSPE	CALF_LONE	int	4	10	0	1
GSPE	CertificationDate	datetime	8	23	3	1

GSPE	CertificationLevel	varchar	15	15	NULL	0
GSPE	CertifiedBy	varchar	50	50	NULL	1
GSPE	Comments	varchar	255	255	NULL	1
GSPE	Counted	bit	1	1	NULL	1
GSPE	COW	int	4	10	0	1
GSPE	COW_W_0	int	4	10	0	1
GSPE	COW_W_1	int	4	10	0	1
GSPE	COW_W_2	int	4	10	0	1
GSPE	COW_W_3	int	4	10	0	1
GSPE	DateCounted	date	3	10	0	1
GSPE	Density	float	8	53	NULL	1
GSPE	Exclude	bit	1	1	NULL	1
GSPE	FiringOrder	int	4	10	0	1
GSPE	GMU	varchar	20	20	NULL	1
GSPE	ID	varchar	10	10	NULL	0
GSPE	Int	int	4	10	0	1
GSPE	IntID	int	4	10	0	1
GSPE	IsIntensivePlot	bit	1	1	NULL	1
GSPE	IsStdPlot	bit	1	1	NULL	1
GSPE	LG_BULL	int	4	10	0	1
GSPE	MED_BULL	int	4	10	0	1
GSPE	MED_L_BULL	int	4	10	0	1
GSPE	MOOSE	int	4	10	0	1
GSPE	Observer	varchar	50	50	NULL	1

GSPE	Park	varchar	4	4	NULL	0
GSPE	Perc_Not_Flown	int	4	10	0	1
GSPE	Personnel	varchar	100	100	NULL	1
GSPE	Pilot	varchar	50	50	NULL	1
GSPE	Rand	float	8	53	NULL	1
GSPE	RecordInsertedBy	nvarchar	100	50	NULL	0
GSPE	RecordInsertedBy	sysname	100	50	NULL	0
GSPE	RecordInsertedDate	datetime	8	23	3	0
GSPE	SCF_Plot	bit	1	1	NULL	1
GSPE	SearchMin	float	8	53	NULL	1
GSPE	Season	varchar	20	20	NULL	1
GSPE	Selected	bit	1	1	NULL	1
GSPE	SM_BULL	int	4	10	0	1
GSPE	SourceFilename	varchar	255	255	NULL	1
GSPE	StartDate	datetime	8	23	3	1
GSPE	Std	int	4	10	0	1
GSPE	StopDate	datetime	8	23	3	1
GSPE	Strat	varchar	1	1	NULL	1
GSPE	StratHab	varchar	100	100	NULL	1
GSPE	StratMoose	int	4	10	0	1
GSPE	StratName	varchar	10	10	NULL	1
GSPE	StratTracks	int	4	10	0	1

GSPE	SubArea	varchar	50	50	NULL	1
GSPE	SurveyID	int	4	10	0	1
GSPE	SurveyName	varchar	100	100	NULL	0
GSPE	SurveyUnitSet	varchar	100	100	NULL	1
GSPE	SurveyYear	int	4	10	0	0
GSPE	UNKNOWN	int	4	10	0	1
GSPE	X_COORD	float	8	53	NULL	1
GSPE	Y_COORD	float	8	53	NULL	1
GSPE	YBULL_ALL	int	4	10	0	1
GSPE	YBULL_GTSF	int	4	10	0	1
GSPE	YBULL_SF	int	4	10	0	1
GSPE_Surveys	Abstract	varchar	-1	-1	NULL	1
GSPE_Surveys	AreaSurveyed_mi	float	8	53	NULL	1
GSPE_Surveys	AverageSearchEffort	float	8	53	NULL	1
GSPE_Surveys	Comments	varchar	-1	-1	NULL	1
GSPE_Surveys	DataResourcesDirectory	varchar	1000	1000	NULL	1

GSPE_Surveys	DatasetProcessingSteps	varchar	-1	-1	NULL	1
GSPE_Surveys	DataSource	varchar	2000	2000	NULL	1
GSPE_Surveys	DeliverablesDatasetReferenceCode	int	4	10	0	1
GSPE_Surveys	EndDate	date	3	10	0	1
GSPE_Surveys	Methodology	varchar	50	50	NULL	1
GSPE_Surveys	Network	char	4	4	NULL	1
GSPE_Surveys	Park	varchar	4	4	NULL	1
GSPE_Surveys	Personnel	varchar	200	200	NULL	1
GSPE_Surveys	ProtocolReferenceCode	int	4	10	0	1
GSPE_Surveys	ProtocolVersion	float	8	53	NULL	1
GSPE_Surveys	RecordInsertedBy	varchar	50	50	NULL	0
GSPE_Surveys	RecordInsertedDate	datetime	8	23	3	0
GSPE_Surveys	ReportLink	varchar	2000	2000	NULL	1
GSPE_Surveys	ReportReferenceCode	int	4	10	0	1
GSPE_Surveys	Season	varchar	10	10	NULL	1
GSPE_Surveys	StartDate	date	3	10	0	1
GSPE_Surveys	Summary	varchar	-1	-1	NULL	1
GSPE_Surveys	SurveyName	varchar	100	100	NULL	0
GSPE_Surveys	ValidatedBy	varchar	50	50	NULL	1
GSPE_Surveys	ValidatedDate	date	3	10	0	1

GSPE_Surveys	Year	int	4	10	0	0
SurveyUnits	CENTRLAT	float	8	53	NULL	1
SurveyUnits	CENTRLON	float	8	53	NULL	1
SurveyUnits	Elev_Mean	float	8	53	NULL	1
SurveyUnits	Elev_Med	float	8	53	NULL	1
SurveyUnits	Feature	hierarchyid	-1	-1	NULL	1
SurveyUnits	Feature	geometry	-1	-1	NULL	1
SurveyUnits	Feature	geography	-1	-1	NULL	1
SurveyUnits	GMU	varchar	5	5	NULL	1
SurveyUnits	ID	varchar	10	10	NULL	0
SurveyUnits	RecordInsertedBy	nvarchar	100	50	NULL	0
SurveyUnits	RecordInsertedBy	sysname	100	50	NULL	0
SurveyUnits	RecordInsertedDate	datetime	8	23	3	0
SurveyUnits	Region	int	4	10	0	1
SurveyUnits	SurveyUnitSet	varchar	100	100	NULL	0
SurveyUnits	UnitComment	varchar	1000	1000	NULL	1
SurveyUnits	UnitsIRMARefCode	int	4	10	0	1
SurveyUnitSets	Comments	varchar	-1	-1	NULL	1
SurveyUnitSets	Description	varchar	4000	4000	NULL	1
SurveyUnitSets	IRMA Reference Code	int	4	10	0	1
SurveyUnitSets	Park	varchar	4	4	NULL	1
SurveyUnitSets	ProjectLeader	varchar	50	50	NULL	1
SurveyUnitSets	SourceFile	varchar	255	255	NULL	1
SurveyUnitSets	SurveyUnitSet	varchar	100	100	NULL	0

Source code repository

Source code repository

Source code for the Moose Monitoring Database Application is available through GitHub at https://github.com/NPS-ARCN-CAKN/Moose3

Data mining resources

Data mining resources

Mat Sorum put together a summary in 2016 for ARCN Moose Survey Data_3_23_2016.xlsx	N:\Natural\GAAR\Moose\GAAR Moose Surveys
DENA Moose drive	\\inpdenafiles02\teams\ResMgmt\Wildlife\Moose
CAKN Moose shared drive	J:\Monitoring\Moose\Data
ARCUS has old reports	

. He put data at

: \\inpdenafiles02\teams\ResMgmt

Hey Scott! Sounds like a big lift. You have all the YUCH surveys I have information on. I assume the red ones stem from some information you perhaps got from Shults or Lawler, but I don't have additional information on them. I tasked Mat Sorum with organizing all the ARCN survey data a few years back, which he did. So he may be in the best position to assess where we are at on those surveys. Attached is a spreadsheet that was developed. While I was on a bunch of the western surveys, I didn't end up with the data, Brad did. I assume Raime and Letty manage these datasets now.

For GAAR, we have 2 moose surveys: the Koyukuk and the upper Kobuk. So I would modify your naming conventions to reflect that. The upper Kobuk survey is really upstream of KOVA (ie it doesn't cover much of KOVA) and covers GAAR "boot". All the survey data for these 2 areas can be found here: N:\Natural\GAAR\Moose\GAAR Moose Surveys. Hopefully that will turn some of your reds green.

The Tagagawik (not Tagagawiki as spelled in first 2 lines) and the Tag are the same river, in case that helps you.

Kyle

An example data entry session using FileToSql

An example data entry session using FileToSql

There are many ways to enter data into the moose monitoring database. The example below shows the step I took to to enter the 2013 DENA Moose Survey data into the database.

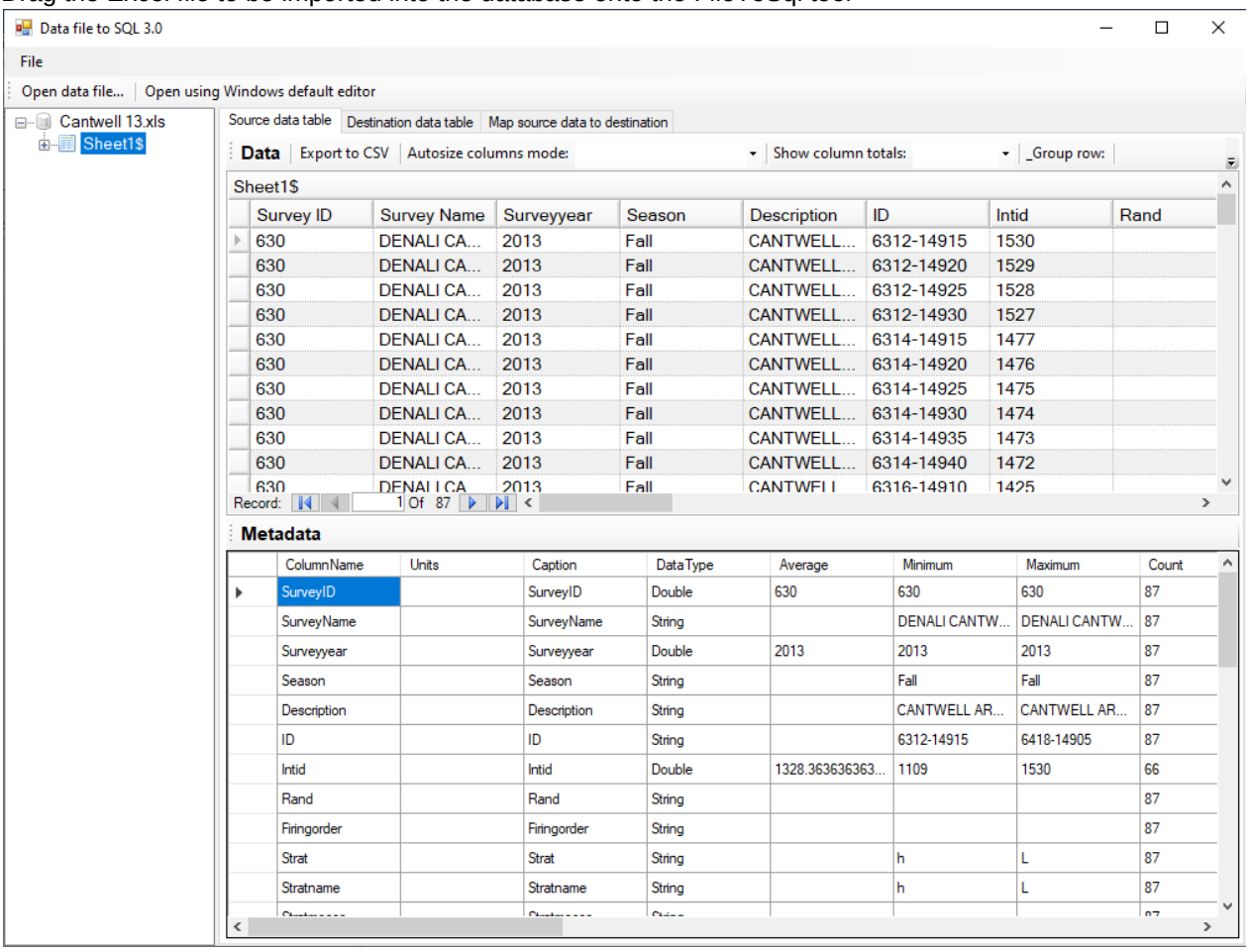
There are a couple of resources that are ideally at hand for data entry of a moose survey in addition to the ones in the Data management resources table:

Resource	Location
Survey report	Not available as of writing. Probably not written. DENA biologist did not have it.
Data deliverable file (GSPE or older Gasaway format spreadsheet).	J:\Monitoring\Moose\Data\2013 DENA Moose Survey

There are a number of ways to import data into a SQL Server database. These include:
1. SQL Server Import and Export Wizard
2. Writing INSERT queries as an Excel formula
3. Scripting with Python, R or other language
4. Using ARCN's FileToSql tool

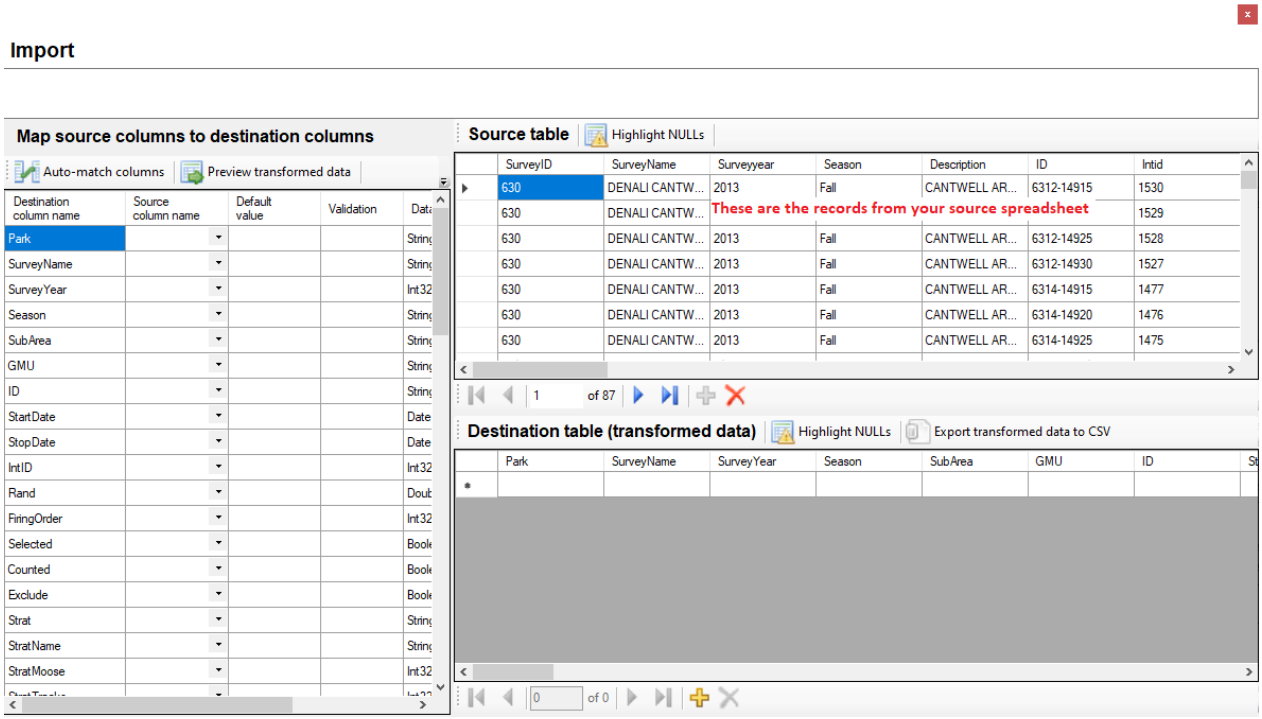
I used the FileToSql tool but the SQL Server Import and Export wizard would work just as well.

Procedure
Ensure the parent Survey record exists in the moose monitoring database for the data you wish to import. The Survey table must have a parent row before any child GSPE data records can be related to it. In this example I've ensured the parent '2013 DENA Moose Survey' records exists in the Surveys table.
Copy down or remember the unique SurveyName because you will need it later.
Start FileToSql
Drag the Excel file to be imported into the database onto the FileToSql tool

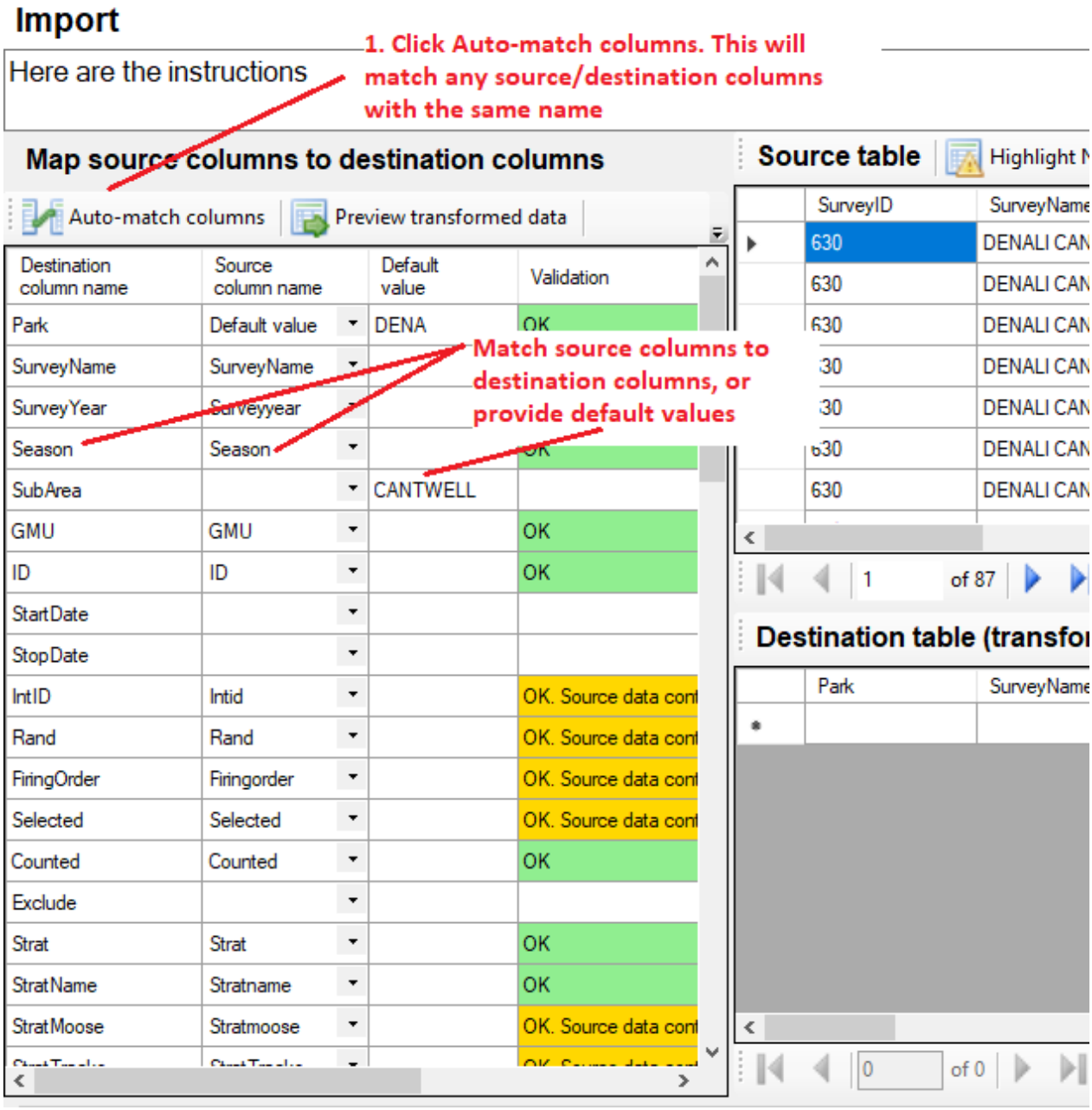


The spreadsheet and tabs show up in the tree at left. Ensure the data in the Data grid exactly represents

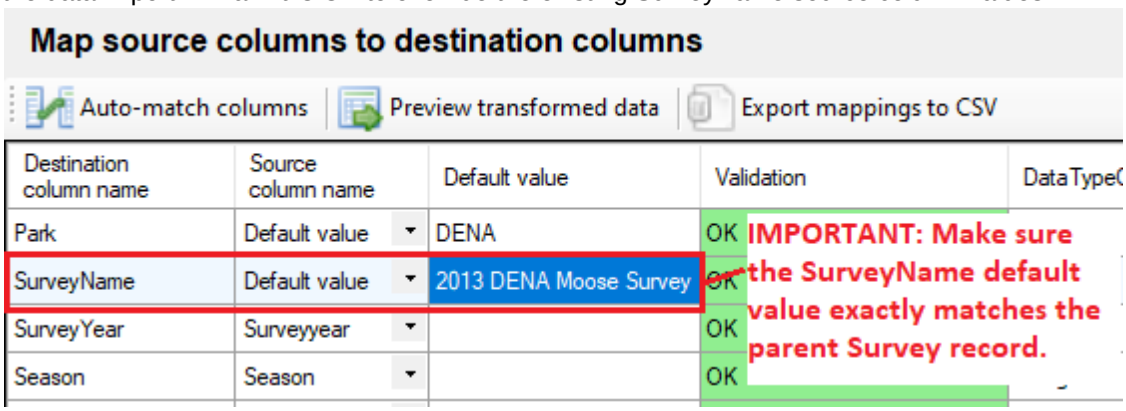
what is in the source spreadsheet.The information in the Metadata grid below may help to elucidate any problems with the data. Look at the data types, max and min values.
If the source dataset looks good click on the Destination data table tab
Enter a connection string to the moose monitoring database. As of Oct. 2021 it is
Server=inpyugamsvm01\nuna_dev;Database=moose;Trusted_Connection=True;
Enter an SQL query that duplicates the data fields in the source data file you would like to import. In most cases the following query will work: **select top 10 * from GSPE**
Click Execute
If your permissions, connection string and query are all good then you should see some database records from the GSPE table of the Moose database and a columns mapping tool should appear. The source data should appear under the Source Table grid. On the left is a tool you will use to match source data columns to destination database table columns.



Click Auto-match columns

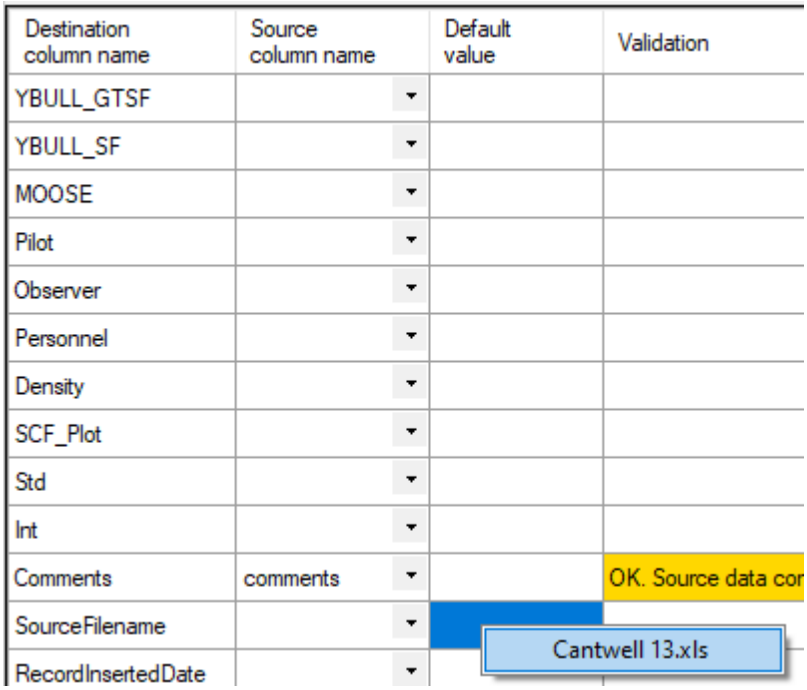


IMPORTANT: Ensure the default value for SurveyName exactly matches the SurveyName field of the parent Survey record in the database. If this is not done you will get referential integrity errors and the data import will fail. It is OK to override the existing SurveyName source column values.



Ensure the default value for SurveyName matches exactly the SurveyName of the parent Survey record; '2013 DENA Moose Survey', for example.
Match any remaining columns from the source spreadsheet to the database table columns, or provide

default values.
Click in the **SourceFilename** row
Right click the Default value column for SourceFilename
The source filename appears, select it to auto-fill the SourceFilename default value to the file name.



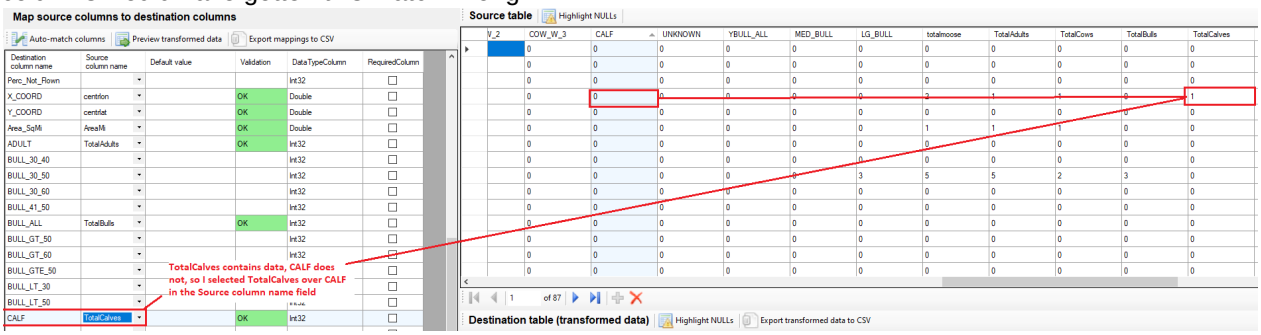
Select the **RecordInsertedDate** row
Right click the Source column name column and select Current datetime

Select the RecordInsertedBy column and set its default value to Current username using the same steps as above.
Set CertificationLevel default value to 'Provisional'

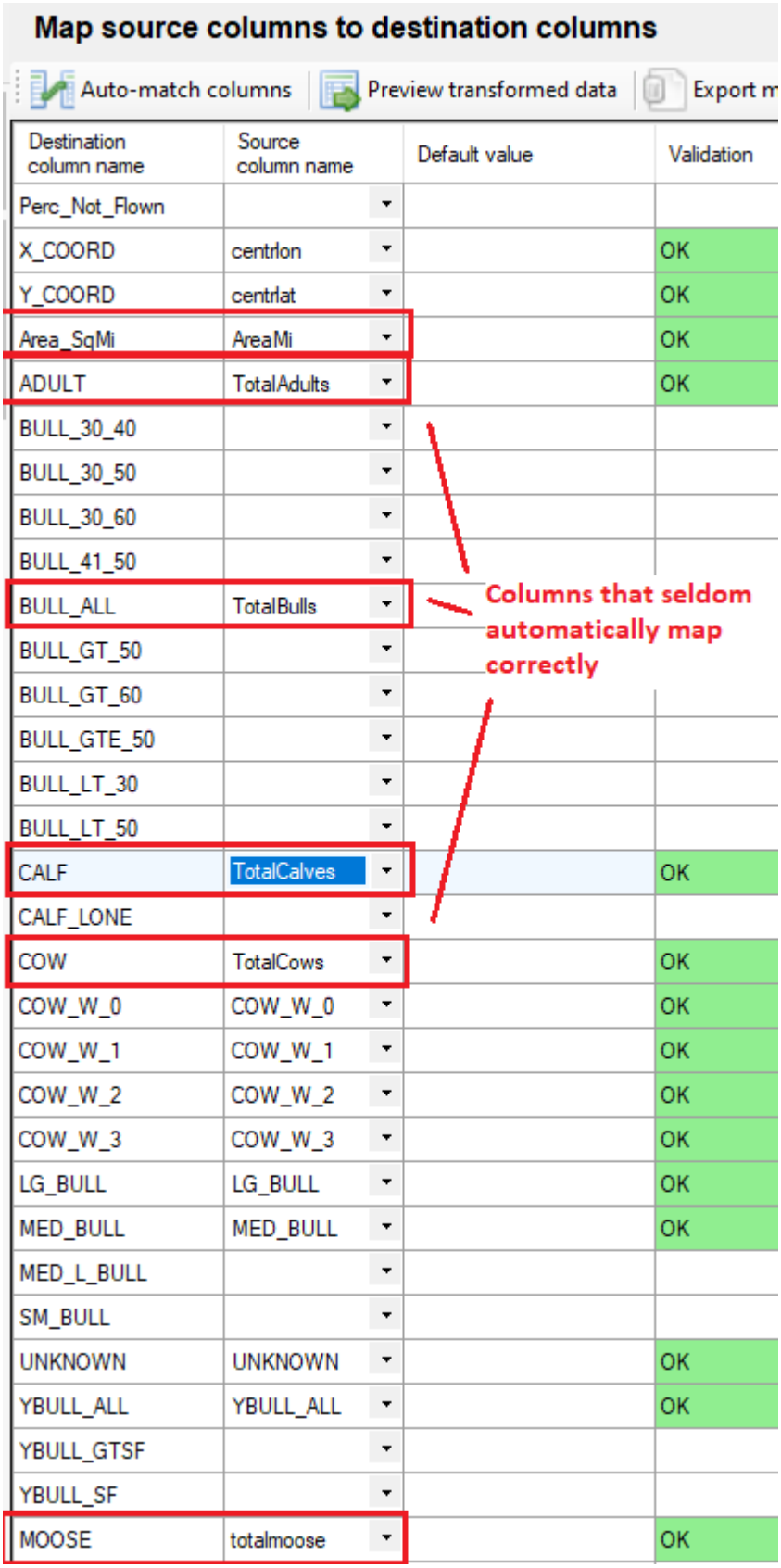
Gotchas

Notice that the source column CALF contains all zeroes where TotalCalves contains count data. Auto-match

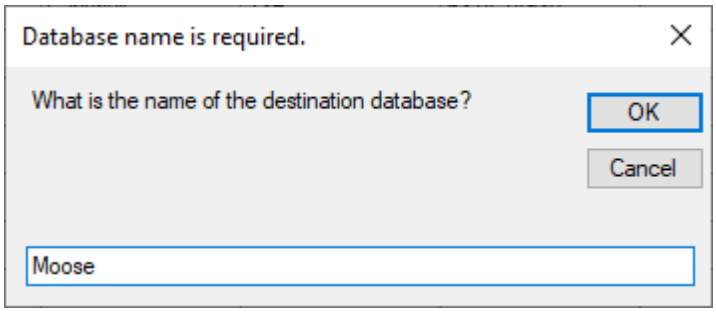
columns would have gotten this match wrong.



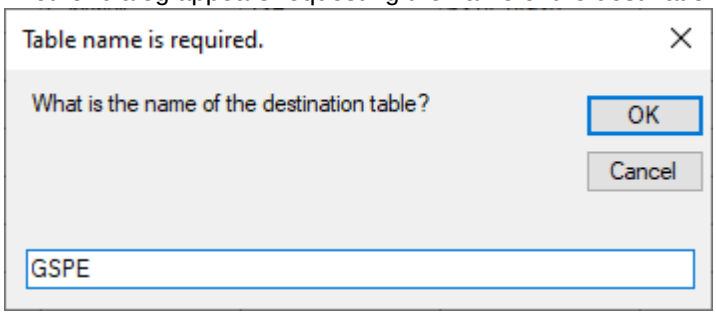
Other columns that seldom map correctly automatically appear below



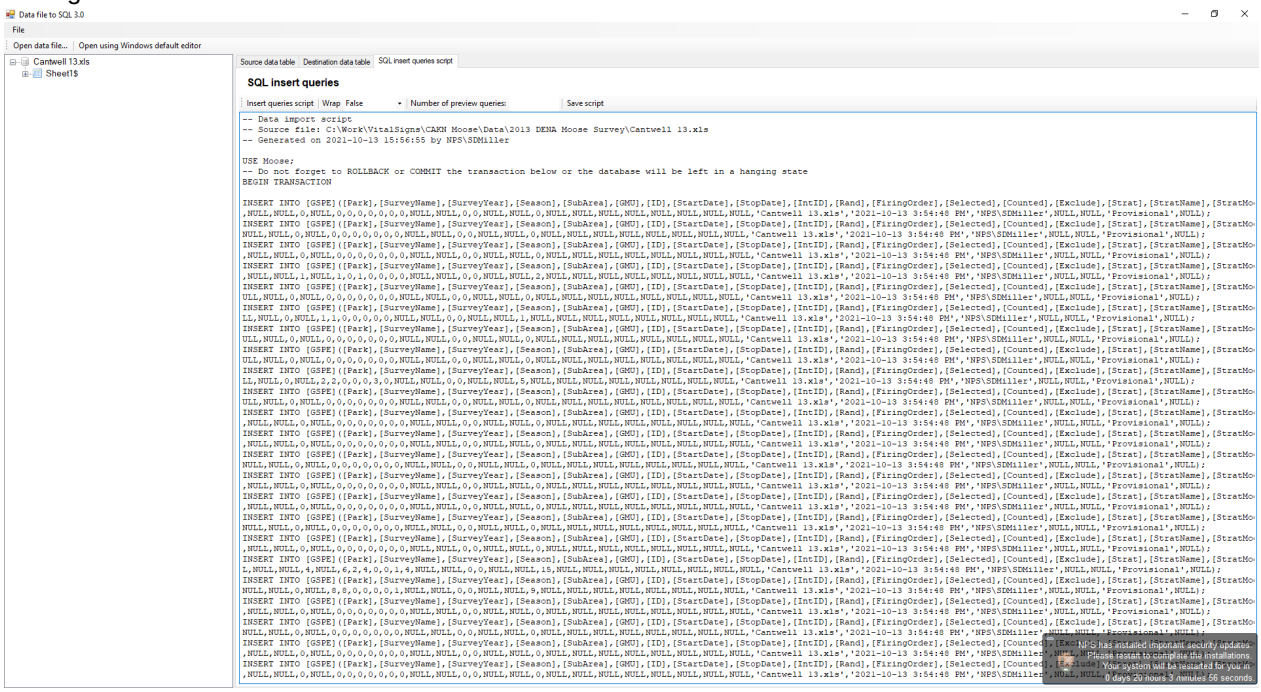
Carefully check your mappings and if all looks good click Import transformed data
A dialog appears requesting the database name. Enter 'Moose'.



Another dialog appears requesting the name of the destination database table. Enter 'GSPE'.



If all goes well the tool will generate a script of SQL insert queries that you may execute in SQL Server Management Studio to insert the records.



Copy the text of the insert queries script.
Open SSMS
Navigate to the moose monitoring database.
Open a new query window and copy the insert queries script into it.
Execute the script.
If all the queries executed with no failures then highlight the word COMMIT in the 6th line and execute the command to commit the new records to the database.
If SSMS reported any errors then highlight the work ROLLBACK to revert the insert queries. Fix any errors and try again.
YOU MUST COMMIT OR ROLLBACK or the database will be left in a hanging state with no access to others.
Issue an SQL query to ensure the records exist. Example: SELECT * FROM GSPE WHERE SurveyName='2013 DENA Moose Survey'.
Ensure the number of records inserted matches the number of rows in the source data table.
Sum the data in the source spreadsheet using a formula and compare the result to summed data in the database. Example SELECT Sum(CALF) as TotalCalves FROM GSPE WHERE SurveyName='2013 DENA

Moose Survey'.

Maintaining this help document

Maintaining this help document

This help document is written using the free version of HelpNDoc.