



Administering your Enterprise Geodatabase using Python

Annie Sasidar & Jillian Penney

2019 Esri DEVSummit Conference | Palm Springs, CA

Assumptions

- Basic knowledge of python
- Basic knowledge enterprise geodatabases and workflows
- You want code

- Please turn off or silence cell phones



Roadmap

Session is divided into three parts

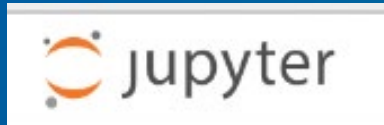
- **Part 1: Types of administrators**
- **Part 2: Geodatabase Creation**
- **Part 3: Version Management**

- **Demos throughout**



Python

- Free
- Simple and easy to learn
- Easy to maintain
- Wide-acceptance
- Modular
- Cross platform
- Scheduling

A screenshot of a PythonWin IDE window titled "PythonWin - [WeightedAttributeOverlay.py]". The window has a menu bar (File, Edit, View, Tools, Window, Help) and a toolbar. The main editor area contains a Python script with comments and code for creating a geoprocessor object, defining a traceback object, and setting tool inputs. The console at the bottom shows the execution of the script, including the import of arcpy and the usage of the "buffer" and "buffer_analysis" tools. The console output is in Russian, showing the help text for the "buffer" tool and the parameters for the "buffer_analysis" tool.

```
#import system modules
import arcpy, math, os, sys, traceback

#main function, all functions run in WeightedAttributeOverlay
def WeightedAttributeOverlay():
    #create the geoprocessor object
    gp = arcpy.Describe.create(93)
    #set overwrite output property
    gp.overwriteoutput = True

    #define traceback object
    def AddPrintMessage(msg, severity):
        print msg
        if severity == 0: gp.AddMessage(msg)
        elif severity == 1: gp.AddWarning(msg)
        elif severity == 2: gp.AddError(msg)

    #set tool inputs
    input = gp.getparameterastext(0)
    uniqueid = gp.getparameterastext(1)
    fieldstrs = gp.getparameterastext(2)
    outfc = gp.getparameterastext(3)
    outfield = gp.getparameterastext(4)

    #load multiple overlay fields into list

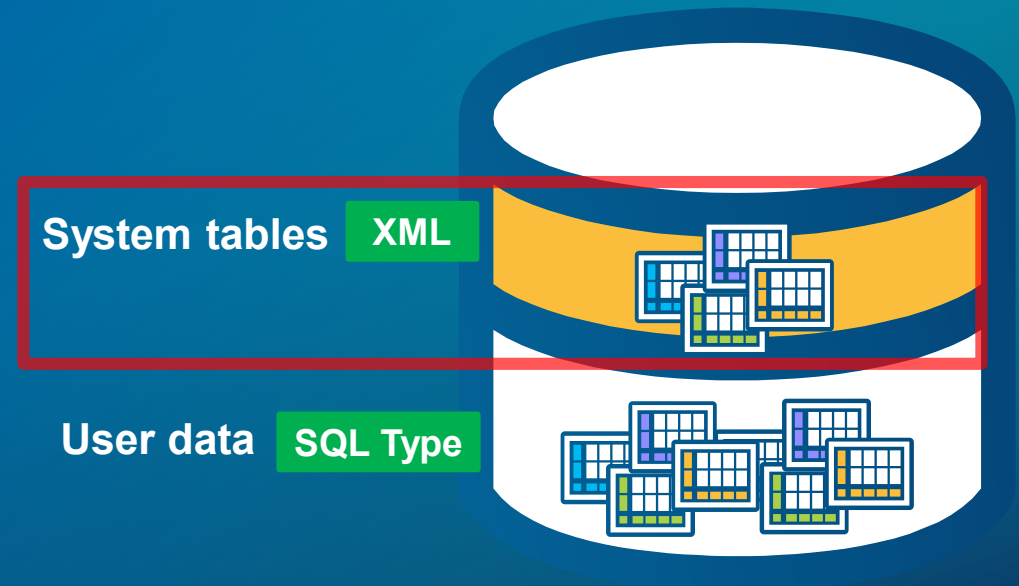
    #input features
    #unique ID field
    #field weight string, composed of field | # classes | classif
    #output feature class containing weighted and output fields
    #name of the output field

    #load multiple overlay fields into list

>>> import arcpy
>>> arcpy.Usage("buffer")
u'buffer(object [, offset[, size]])\n\nCreate a new buffer object which references the given object.\n\nThe
buffer will reference a slice of the target object from the\nstart of the object (or at the specified offset).
The slice will\nnextend to the end of the target object (or with the specified size).'
>>> arcpy.Usage("buffer_analysis")
'Buffer_analysis(in_features, out_feature_class, buffer_distance_or_field, (FULL | LEFT | RIGHT |
OUTSIDE_ONLY), (ROUND | FLAT), (NONE | ALL | LIST), (dissolve_field;dissolve_field...))\nBuffer Features'
>>>
```

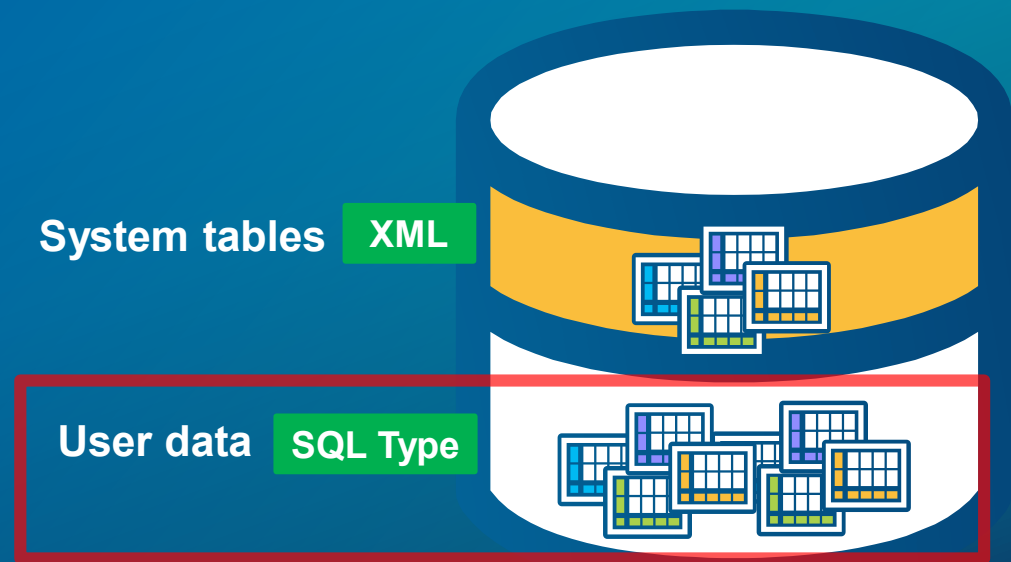
Geodatabase system tables

- System tables store definitions, rules, and behavior for datasets
- Tracks contents within a geodatabase
- Stores some database level metadata
 - Versions, domains, etc.
- Admin operations:
 - Geodatabase upgrade
 - Connection management
 - Version management



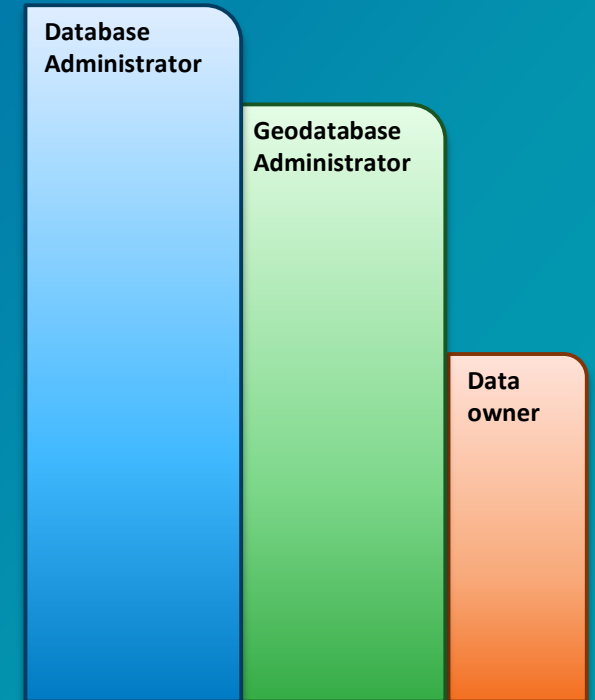
User-defined tables

- Stores the content of each dataset in the geodatabase
 - Datasets are stored in one or more tables
- Administrative Operations:
 - Grant/revoke privileges
 - Update statistics/indexes
 - Register as versioned
 - Add global id's
 - Enable editor tracking
 - Enable Archiving



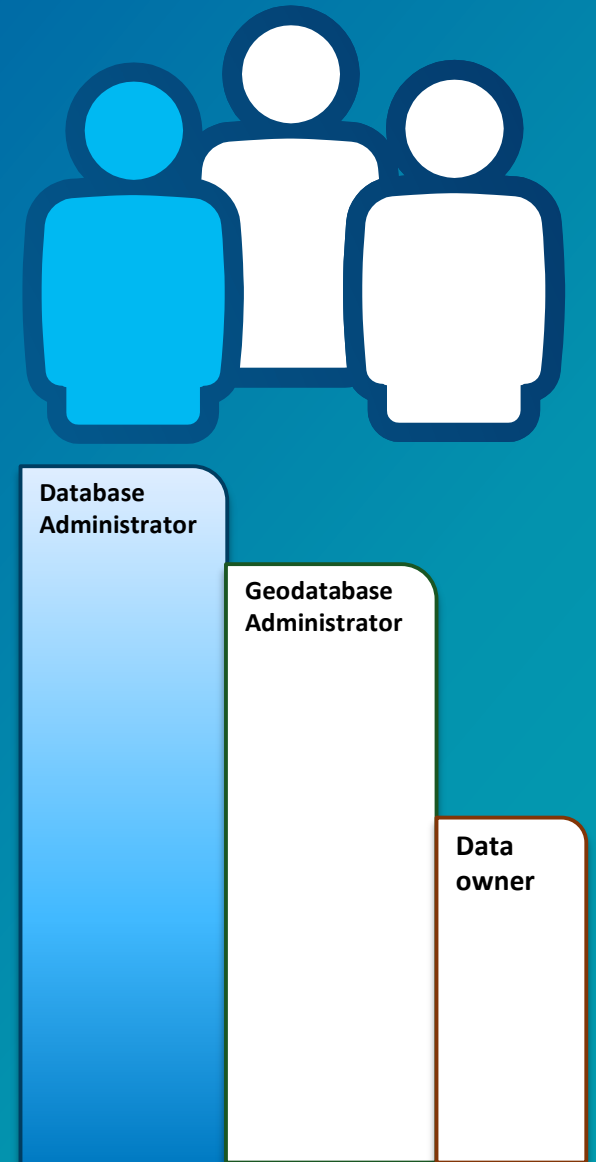
Types of administrators

- Database administrator (DBA)
- Geodatabase administrator
- Data owner (aka dataset administrator)
- May or may not be the same person.



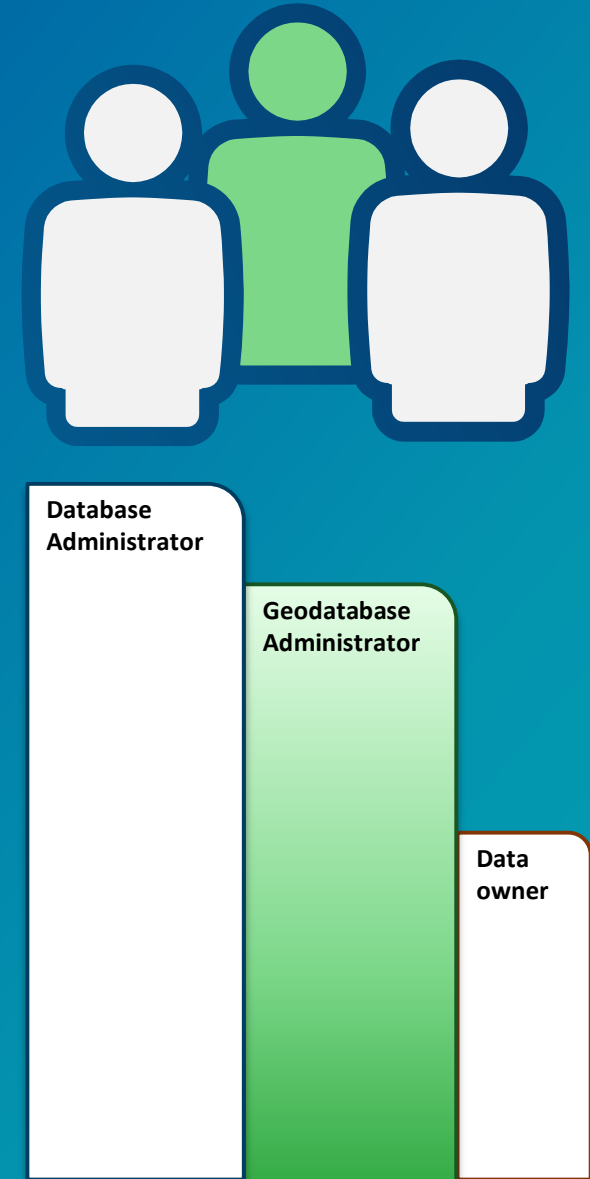
Database Administrator

- Instance level admin
- User management
- Database backup
- System security
- Performance monitoring
- DBMS client software



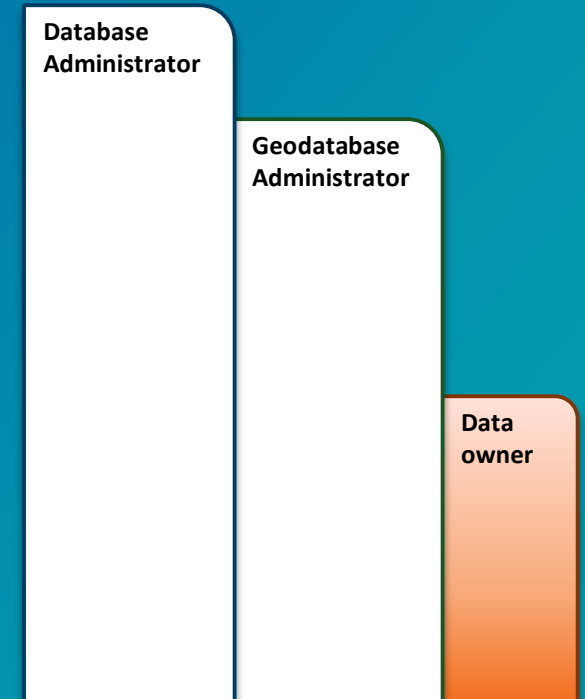
Geodatabase Administrator

- Owns the geodatabase repository
- Can edit any versioned data
- Perform compress



Data Owner

- **Dataset administrator**
 - Granting privileges to data
 - Modifying schema of data
 - Enabling Geodatabase behavior on the data tables
 - Statistics and index maintenance of the data tables





Database Admin Workflows

Creating Enterprise Geodatabases

- **Create Enterprise Geodatabase tool**

When you have a need for:

- Creating testing or development environments
- Database does not already exist

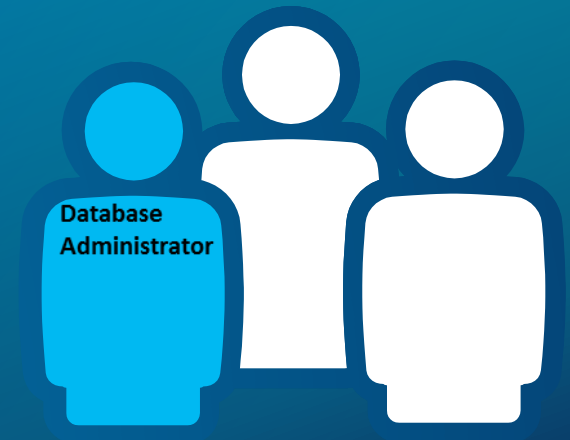
- **Enable Enterprise Geodatabase tool**

- When you have an existing database



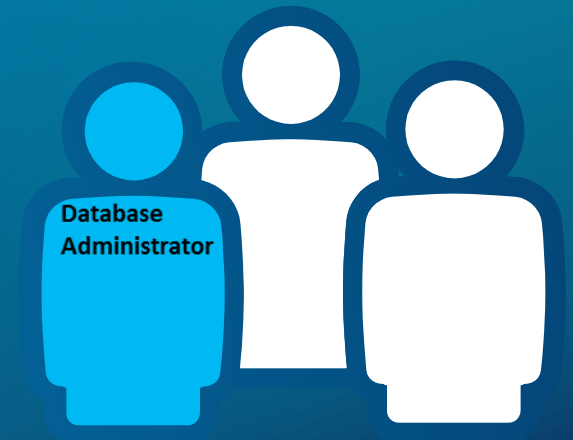
Creating Users

- Create database user tool
- Creates a user in enterprise geodatabase or database
- Cannot create geodatabase administrative user
- <https://pro.arcgis.com/en/pro-app/tool-reference/data-management/create-database-user.htm>

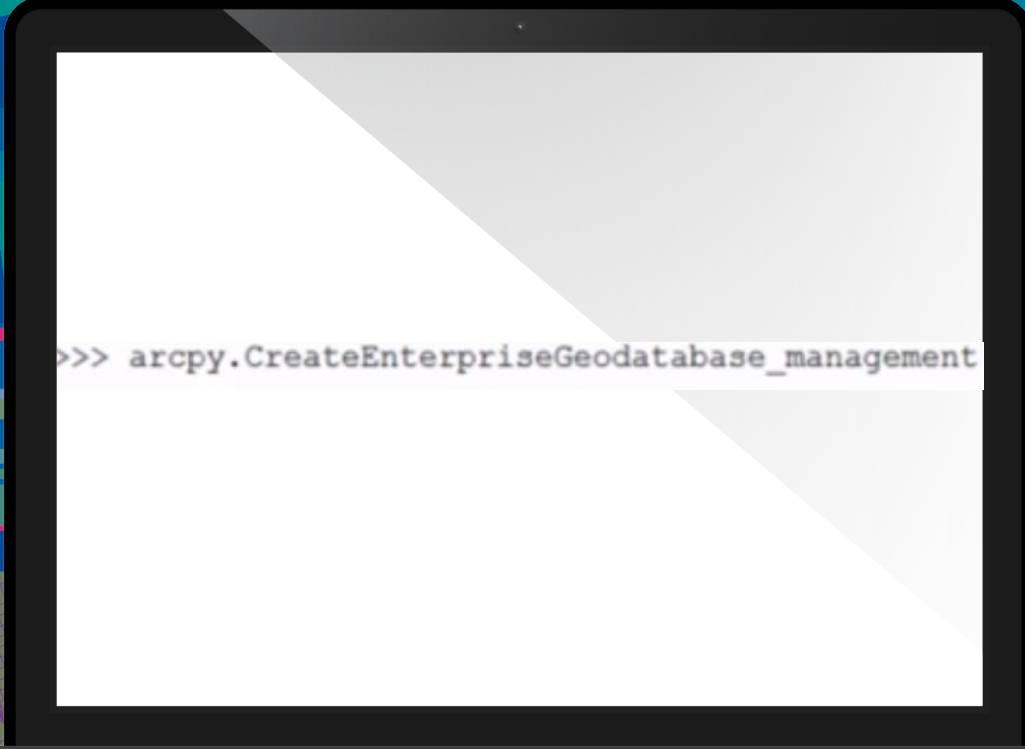


Creating Database Roles

- Makes it easier to assign privileges to a group
- Prior to creating users in the geodatabase
- When creating users you can assign them to a role



Demo 1

A laptop screen with a black frame and a white background. A diagonal line from the top-left corner to the bottom-right corner separates the screen into two triangles. The top-right triangle is light gray, and the bottom-left triangle is white. The code snippet is displayed in the white triangle.

```
>>> arcpy.CreateEnterpriseGeodatabase_management
```

Database Admin workflow

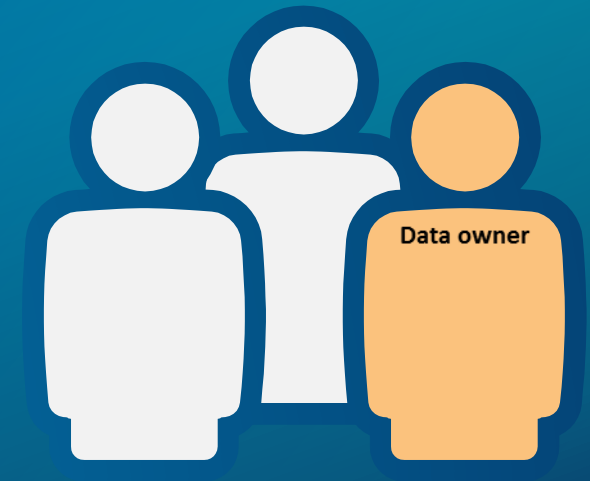
Jillian Penney



Dataset Admin Workflows

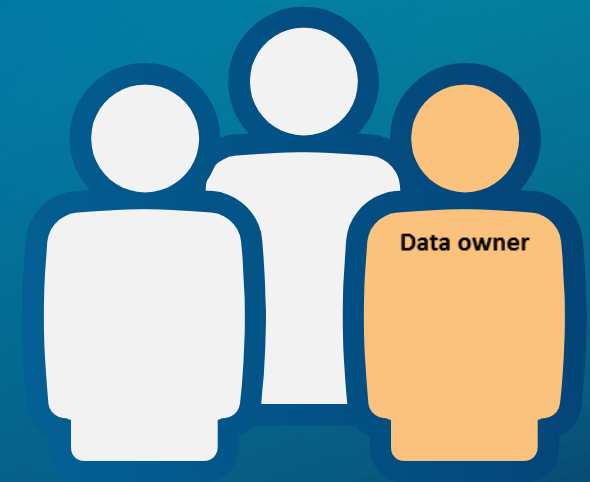
Creating and Loading Data

- Numerous tools for creating any type of data:
 - Create table, Create feature class, Create Raster Dataset, etc.
 - Create Geometric Network, Create Topology, Create Domain, etc.
- Also tools for loading data:
 - Feature class to feature class (single)
 - Feature class to geodatabase (multiple)
 - Import XML workspace
 - Append



Managing privileges

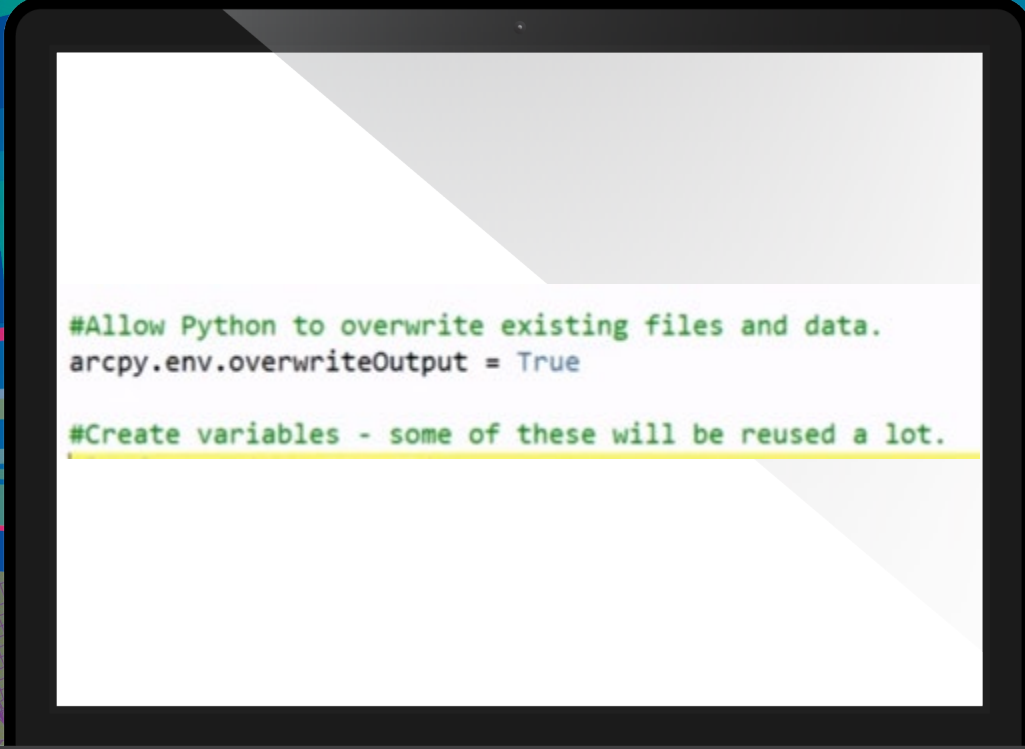
- Allow other users of the geodatabase to view or edit data that you own
- Change privileges tool
- Allows multiple input datasets to be passed in
- Grant view only or view and edit
 - View = select
 - Edit = insert, update, delete



Demo 2

Putting it together via Python

Jillian Penney

A laptop screen is shown, displaying a code editor with Python code. The code is as follows:

```
#Allow Python to overwrite existing files and data.  
arcpy.env.overwriteOutput = True  
  
#Create variables - some of these will be reused a lot.
```

The second line of code is highlighted in yellow. The laptop is black and is positioned in the lower-left quadrant of the slide. The background of the slide is a dark blue gradient with abstract, colorful geometric shapes and patterns on the left side.

```
#Allow Python to overwrite existing files and data.  
arcpy.env.overwriteOutput = True  
  
#Create variables - some of these will be reused a lot.
```


Disconnecting user connections

- A user who is connected but has gone home
- Create a cold backup of the database
- Running large queries that are using up resources

Managing user connections with arcpy functions






- **Block/allow connections**
 - `arcpy.AcceptConnections`
 - Provide boolean
- **Finding connected users**
 - `arcpy.ListUsers`
 - Returns a tuple of properties for each connected user
 - ID, name, machine name, connection time, connection type
- **Disconnecting users**
 - `arcpy.DisconnectUser`
 - Use ids provided from listusers function or use 'ALL' keyword

Disconnecting services

- Services connect as regular users
- Disconnecting all users will disconnect services
- ArcGIS Server logic will try to reconnect
- Workflow:
 - Block connections
 - Stop service
 - Perform administrative tasks
 - Allow connections
 - Start service




Update Geodatabase license

-  Repair Version Metadata
-  Repair Version Tables
-  Update Enterprise Geodatabase License
-  Upgrade Dataset
-  Upgrade Geodatabase

Connection Management

Jillian Penney

A stylized illustration of a laptop with a black frame and a white screen. The screen displays a single line of Python code. The background of the slide is a solid teal color. On the left side, there are several overlapping, semi-transparent rectangular shapes in various colors (blue, yellow, pink, green) and a faint map pattern.

```
>>> arcpy.AcceptConnections()
```



Geodatabase Admin Workflows

Parts of the version administration workflow

- Reconciling, posting, compressing
- Updating statistics and indexes on system tables
- Updating statistics and indexes on user data tables

Reconciling and Posting Versions

- Reconcile = pulling changes from a parent to child version
- Post = pushing reconciled changes to parent version from child
- Reconcile versions tool
- Recommended to run as geodatabase administrator



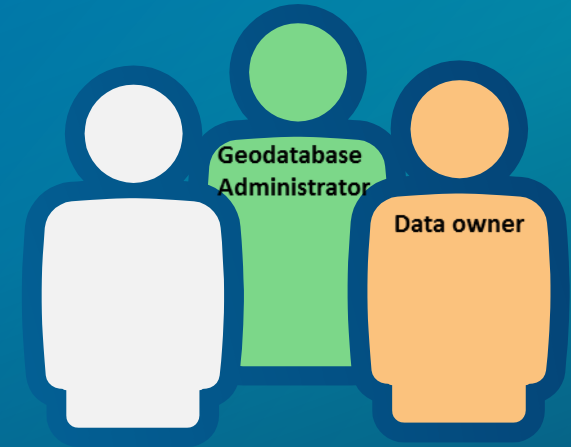
Compressing your geodatabase

- **Compress geoprocessing tool**
- **Run regularly**
 - Trims states
 - Moves edits from delta tables



Indexes and Statistics

- Update after major 'data change' events
 - Reconcile
 - Compress
 - Appending data
 - Typically not necessary after loading new data
- Can be done by both:
 - Geodatabase Admin (system tables)
 - Data owner (data tables)
- We suggest to run these tools as required.



Demo 4

```
# Name: Maintenance.py
# Description: This script will reconcile, post and compress an
#              enterprise geodatabase. It will then rebuild indexes
#              and gather statistics on the data in the geodatabase and
#              send an email report.

# Author: Esri

# Import the required modules
import arcpy, smtplib, sys
```

Version Maintenance

Jillian Penney

An abstract graphic on the left side of the slide. It features a blue gradient background. Overlaid on this are several overlapping, semi-transparent rectangles in various colors including yellow, green, cyan, and magenta. Some of these rectangles are tilted at angles. In the lower-left corner, there is a small, semi-transparent map fragment showing a street grid and several white circular markers of varying sizes.

Summary

Other sessions...

ArcGIS API for Python: Advanced Scripting - 9948 -> Primrose B

Thursday 3/7/2019 -> 9.00 AM- 10.00AM

ArcGIS API for Python for Analysts and Data Scientists – 9973 -> Primrose C- D

Thursday 3/7/2019 -> 10.30 AM – 11.30 AM

**ArcGIS API for Python: Getting to know Pandas and the Spatial Enabled DataFrame-
10299-> Smoketree A- E**

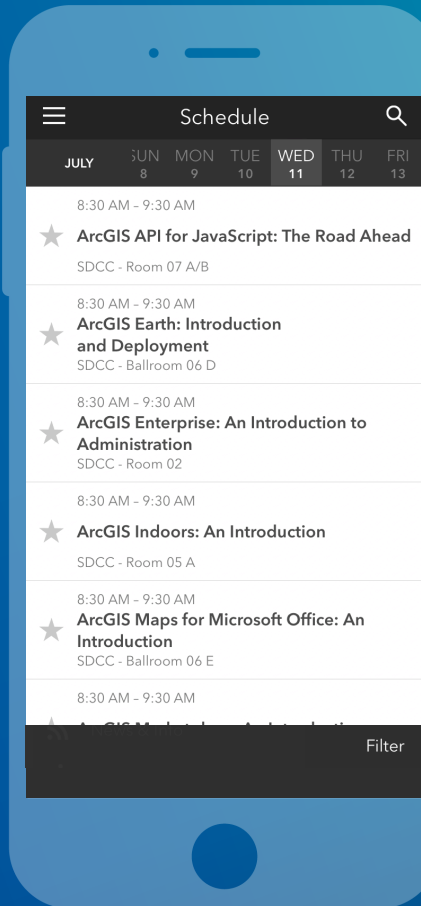
Thursday 3/7/2019 -> 1.00 – 2.00 PM

Please Take Our Survey on the App

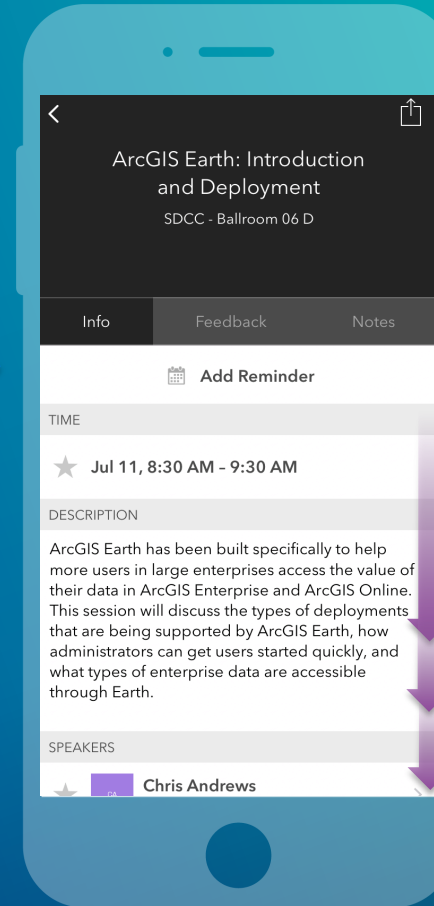
Download the Esri Events app and find your event



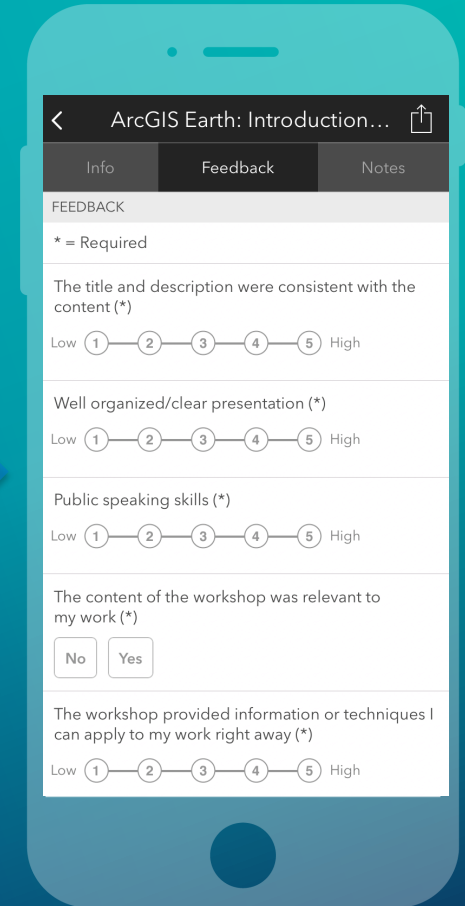
Select the session you attended



Scroll down to find the feedback section



Complete answers and select "Submit"



Slides and Code

<http://esriurl.com/PyGDB2019>





esri

THE
SCIENCE
OF
WHERE