Working with Geodatabases Using SQL and Python

Jochen Manegold

Gerhard Trichtl

ESRI EUROPEAN DEVELOPER SUMMIT

esri

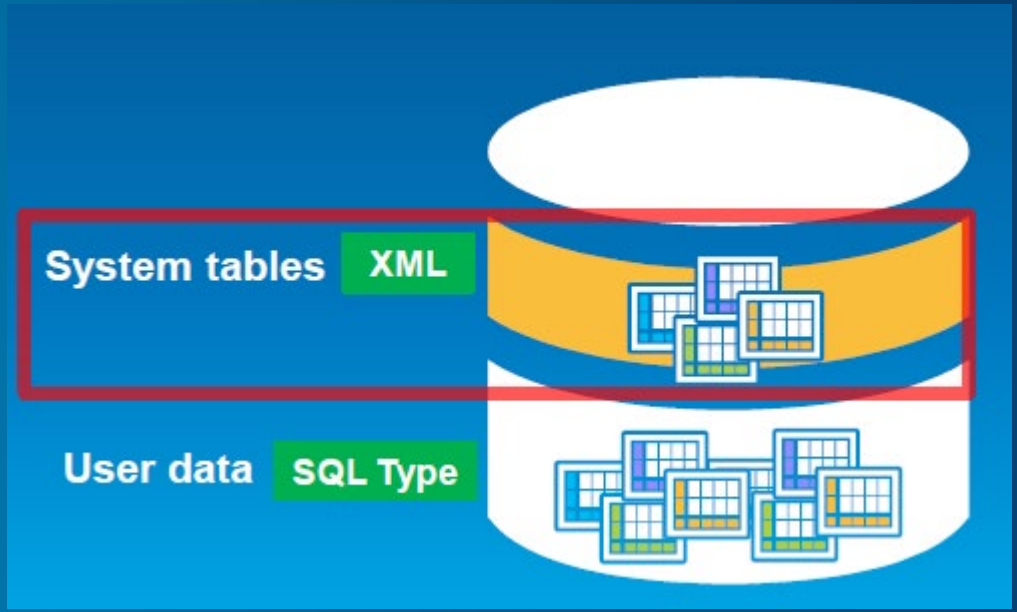
Geodatabase

Geodatabase – What is this?

- A physical store of geographic data
  - Scalable storage model supported on different platforms
- Core ArcGIS information model
  - A comprehensive model for representing and managing GIS data
  - Implemented as a series of simple tables
- A transactional model for managing GIS workflows
- Set of components for accessing data

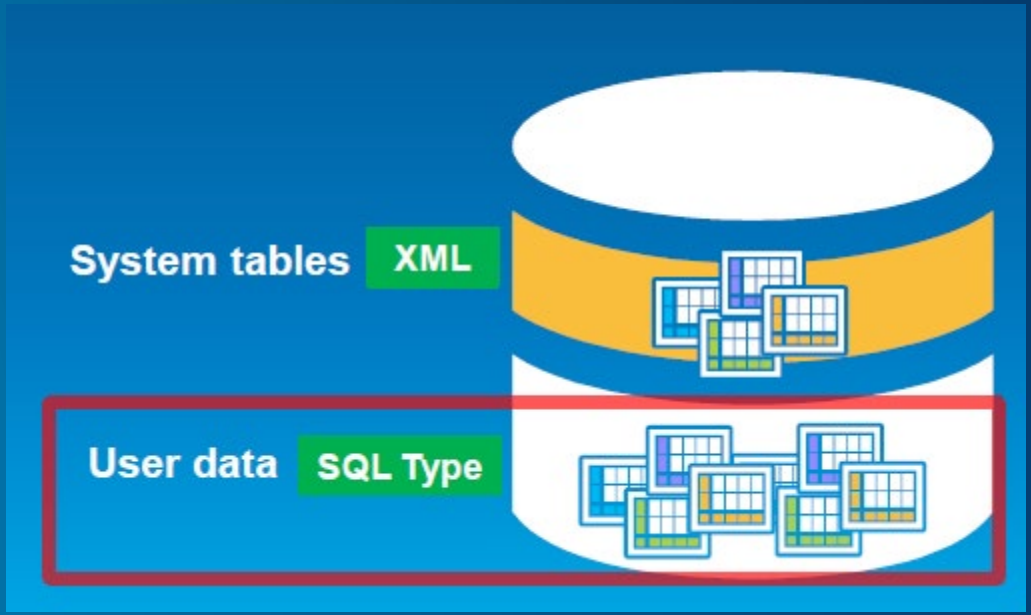
Tables in Geodatabase - Geodatabase system tables

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



Tables in Geodatabase - User defined Tables

- Stores the content of each dataset in the geodatabase
  - Datasets are stored in one or more tables
- Administrative Operations:
  - Granting/revoking privileges
  - Updating statistics/indexes
  - Registering as versioned
  - Adding global id's
  - Enabling editor tracking



Accessing data in a DBMS

- **You can access spatial or non-spatial data in a DBMS to use in ArcGIS**

Geodatabase	Database – Simple Feature Access
DB2	ALTIBASE (deprecated)
Informix	Dameng
ORACLE	Teradata
PostgreSQL	Netezza (deprecated)
Microsoft SQLServer	DB2, Informix
SAP HANA	ORACLE, PostgreSQL
	Microsoft SQLServer, SAP HANA

Working with Geodatabases using Python

or how you create your own Geodatabase

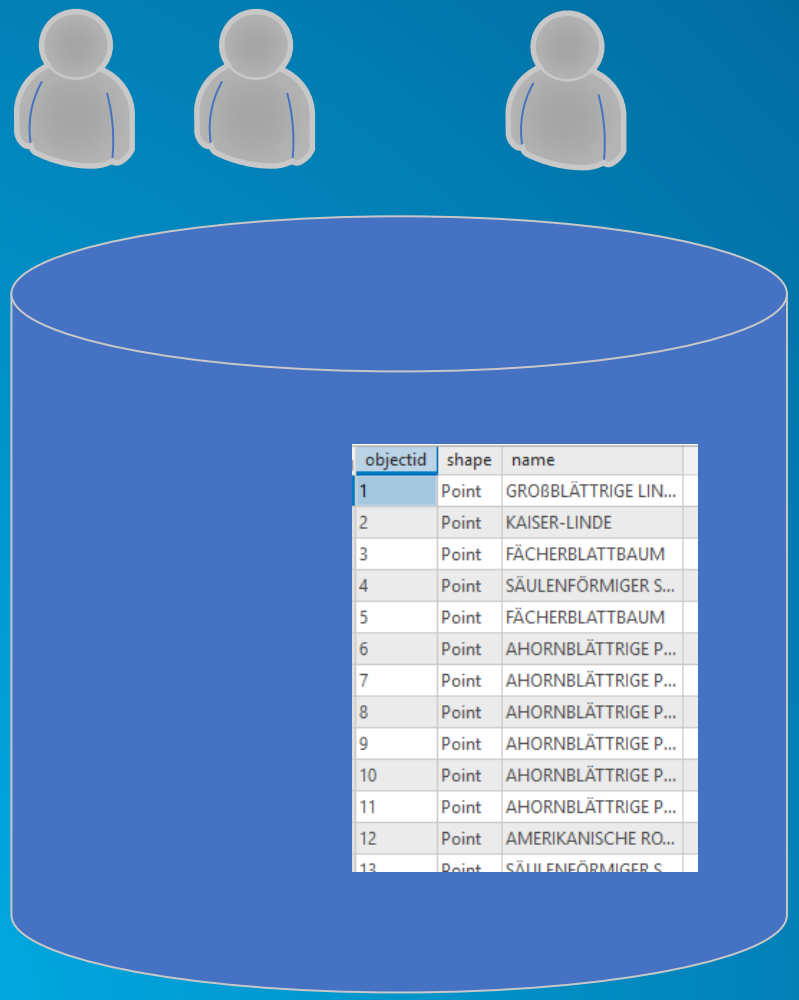
by Jochen Manegold

The Scenario

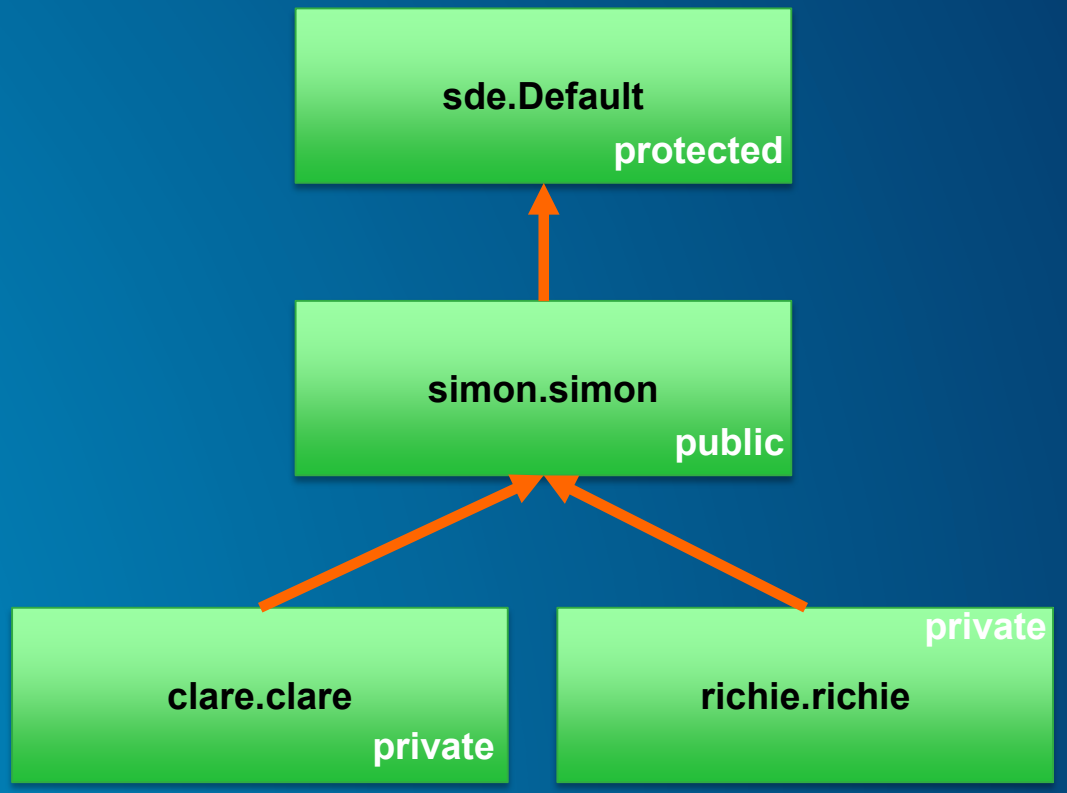
- I am the database administrator in our office
- I want to create a Geodatabase with featureclass and data
- Simon should be the data-owner – responsable for maintenance and data quality
- Clare and Richie are responsible to capture the data
- They want to edit the data in an isolated editor environment (Versioning)

- As this happens in many different ways, I want to automate the setup of this scenario

The Scenario – a database, users, a table, data and a version tree

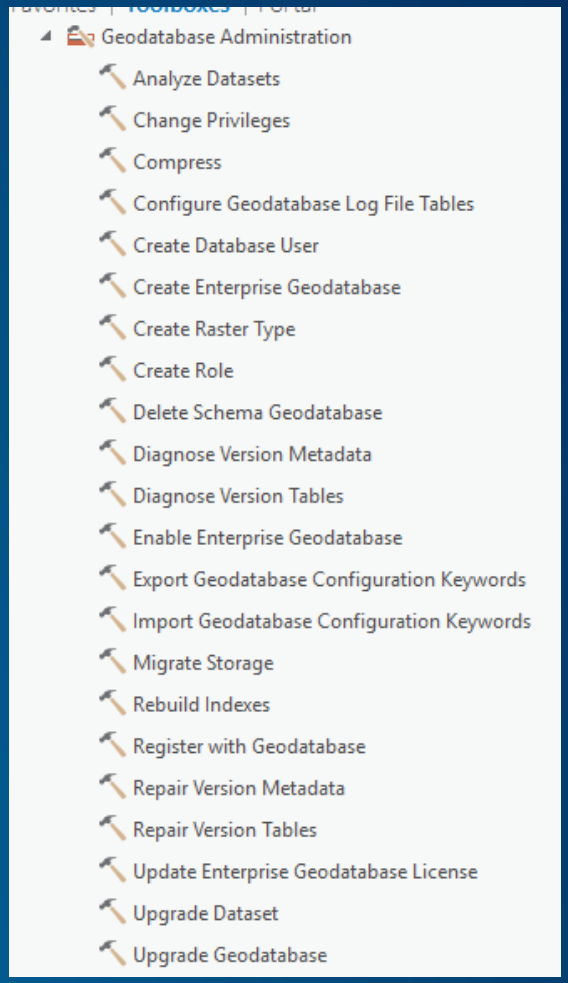


objectid	shape	name
1	Point	GROßBLÄTTRIGE LIN...
2	Point	KAISER-LINDE
3	Point	FÄCHERBLATTBAUM
4	Point	SÄULENFÖRMIGER S...
5	Point	FÄCHERBLATTBAUM
6	Point	AHORNBLÄTTRIGE P...
7	Point	AHORNBLÄTTRIGE P...
8	Point	AHORNBLÄTTRIGE P...
9	Point	AHORNBLÄTTRIGE P...
10	Point	AHORNBLÄTTRIGE P...
11	Point	AHORNBLÄTTRIGE P...
12	Point	AMERIKANISCHE RO...
13	Point	SÄULENFÖRMIGER S...



The Tools

- Database Server (f.e. PostgreSQL)
- ArcGIS Pro Geoprocessing Framework – Standard or Advanced
- Scripting Environment for Python (f.e. PyScripter)



The Steps

1. Create an Enterprise Geodatabase
2. Create an Administrator Connection to the Geodatabase
3. Create the Database Users (Simon, Clare, Richie)
4. Create a Database Role (pg_giseditor) and add the Users to that Role
5. Create User Connections to the Geodatabase for each User
6. Create a Featureclass for the data, add fields and indexes
7. Load Data to the Featureclass
8. Register Featureclass ‚as versioned‘
9. Create a QA-Version for Simon
10. Create an Edit-Version for Clare and an Edit-Version for Richi
11. Create a User Connection to the Version for each User
12. Grant Read and Write Access to the Featureclass for the pg_giseditor Role

The Script

- **Import ArcPy**

```python
import arcpy

# server administrator creates the geodatabase and the sde-user - the
# geodatabase administrator
arcpy.AddMessage("CreateEnterpriseGeodatabase")
arcpy.management.CreateEnterpriseGeodatabase("PostgreSQL",
                                             "halsey",
                                             "devsummit",
                                             "DATABASE_AUTH",
                                             "postgres",
                                             "dbadmin",
                                             "SDE_SCHEMA",
                                             "sde",
                                             "gis12345",
                                             '',
                                             r"C:\data\portal\keycodes.ecp")

# server administrator creates a database connection for the
# geodatabase administrator
arcpy.AddMessage("CreateDatabaseConnection")
arcpy.management.CreateDatabaseConnection(r"C:\Temp\GeodatabasePython",
                                          "devsummit_sde",
```

ArcPy is a Python site package that provides a useful and productive way to perform geographic data analysis, data conversion, data management, and map automation with Python.

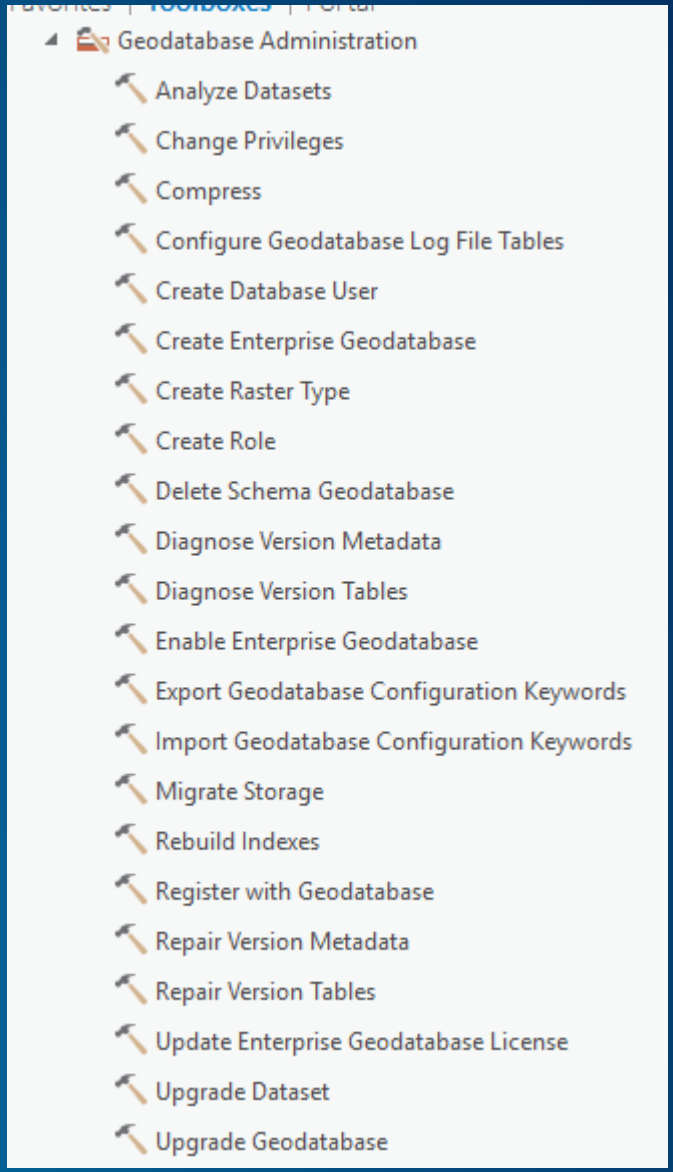
```python
users_list = ["simon", "richie", "clare"]
for user in users_list:
    arcpy.AddMessage("CreateDatabaseUser " + user)
    arcpy.management.CreateDatabaseUser(r"C:\Temp\GeodatabasePython\devsummit_sde.sde",
                                        "DATABASE_USER",
                                        user, "gis12345", '', '')
    arcpy.AddMessage("CreateDatabaseConnection for " + user)
    arcpy.management.CreateDatabaseConnection(r"C:\Temp\GeodatabasePython",
                                              "devsummit_" + user,
                                              "POSTGRESQL",
                                              "halsey",
                                              "DATABASE_AUTH",
                                              user,
                                              "gis12345",
                                              "SAVE_USERNAME",
                                              "devsummit",
                                              '',
                                              "TRANSACTIONAL",
                                              "sde.DEFAULT",
                                              None)
```

Let‘s rock...

Demo

Ressources

- Geoprocessing Tools (Geodatabase Administration Toolset)
- ArcPy Functions for Enterprise Geodatabase
- ArcPy Class for Enterprise Geodatabase



ArcPy Functions for Enterprise Geodatabase

AcceptConnections

Zusammenfassung

Allows an administrator to enable or disable the ability of nonadministrative users to make connections to an enterprise geodatabase.

Auswertung

The **AcceptConnections** function is used by an administrative user to temporarily block connections to an Enterprise geodatabase. This function is used to complement the Connections tab on an Enterprise geodatabase properties page found in ArcGIS Desktop.

- The **AcceptConnections** function must utilize an administrative connection to the database.
- If this function is attempted to be run by a nonadministrative user the function will fail.

Syntax

```
AcceptConnections (sde_workspace, accept_connections)
```

ArcPy Class for Enterprise Geodatabase

- **arcpy.ArcSDESQLExecute**

Methode	Erklärung
commitTransaction ()	No DML statements will be committed until the CommitTransaction method is called. Hinweis: A commit may also occur when the connection to ArcSDE it terminated (check specific DBMS documentation to see how each DBMS deals with a disconnect while in a transaction).
execute (sql_statement)	Sends the SQL statement to the database via an ArcSDE connection. If execute is run outside of a transaction, a commit will automatically take place once the SQL DML (INSERT, UPDATE, DELETE . . .) statement has been executed.
rollbackTransaction ()	Rollback any DML operations to the previous commit.
startTransaction ()	To control when your changes are committed to the database, call the startTransaction method before calling execute. This starts a transaction and no DML statements will be committed until the commitTransaction method is called.

https://pro.arcgis.com/de/pro-app/arcpy/classes/arcsdesqlexecute.htm

Let‘s rock again...

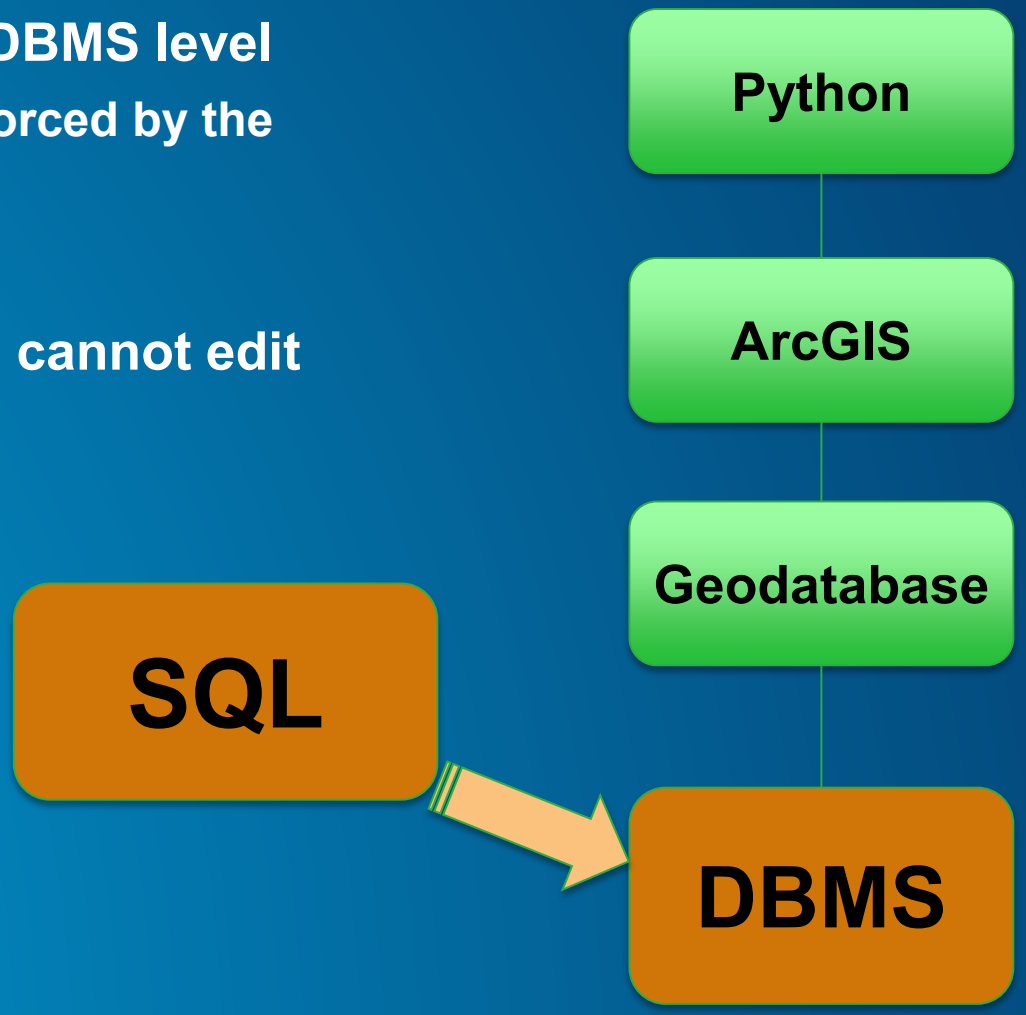
Demo

Access with SQL

Gerhard Trichtl

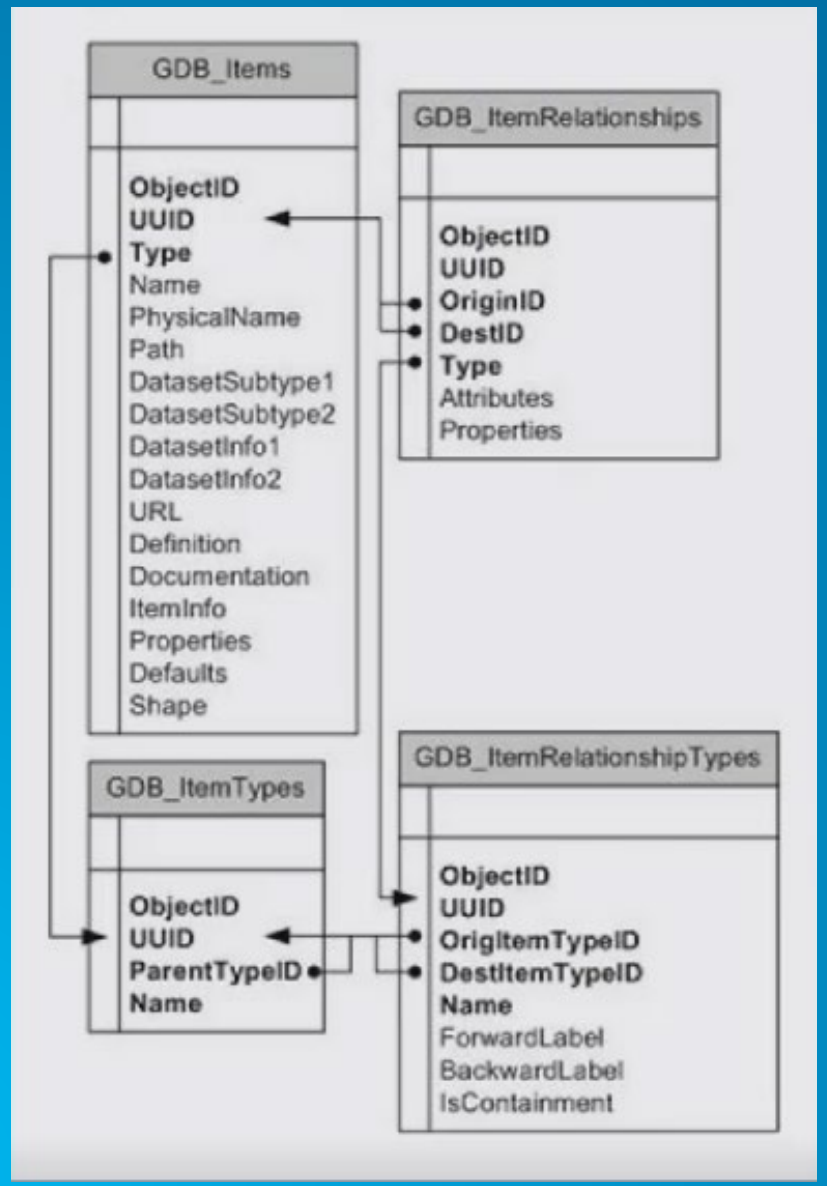
Accessing your geodatabase using SQL

- With SQL, you access the data at the DBMS level
  - Bypass behaviors and functionality enforced by the
  - Geodatabase or ArcGIS clients
- Need to be aware of what you can and cannot edit
  - Know your data
  - Use discovery functions



Information from Geodatabase System Tables

Geodatabase schema – four main system tables



- GDB_Items
  - List all geodatabase items
- GDB_ItemTypes
  - Fixed list of items
- GDB_ItemRelationships
  - List all relationships
- GDB_ItemRelationshipTypes
  - Fixed list of relationships

- XML document for each item
- Native XML
  - SQLServer
  - PostgreSQL
  - DB2
- ArcSDE XML
  - ORACLE
  - Informix

List of Domains in Geodatabase

- **See Example:**
  - **http://desktop.arcgis.com/en/arcmap/latest/manage-data/using-sql-with-gdbs/example-finding-domain-owners.htm**

```sql
--Get List of Domains and Owners
SELECT items.Name AS "Domain Name",
 items.Definition.value('(/*/Owner)[1]','nvarchar(max)') AS "Owner"
FROM sde.GDB_ITEMS AS items INNER JOIN sde.GDB_ITEMTYPES AS itemtypes
ON items.Type = itemtypes.UUID
WHERE itemtypes.Name IN ('Coded Value Domain', 'Range Domain')
```

Get List of Featureclasses with a specific Domain

- See Example:
  - http://desktop.arcgis.com/en/arcmap/latest/manage-data/using-sql-with-gdbs/example-discovering-domain-usage.htm

```sql
--Queries an sde-schema geodatabase in SQL Server for Domain - 'DKM_Nutzungssymbole
DECLARE @DOMAIN_NAME NVARCHAR(MAX);
SET @DOMAIN_NAME = 'DKM_Nutzungssymbole';

DECLARE @CLASS_DEFS TABLE
(
     Name nvarchar(max),
     Definition XML
)
--Insert records to temporary record set
INSERT INTO @CLASS_DEFS
SELECT
  sde.gdb_items.Name,
  sde.gdb_items.Definition
FROM
-- Get the domain item's UUID.
((SELECT GDB_ITEMS.UUID AS UUID
  FROM sde.gdb_items INNER JOIN sde.gdb_itemtypes
  ON sde.gdb_items.Type = sde.gdb_itemtypes.UUID
  WHERE
    sde.gdb_items.Name = @DOMAIN_NAME AND
    sde.gdb_itemtypes.Name IN ('Coded Value Domain','Range Domain')) AS Domain
-- Find the relationships with the domain as the DestinationID.
INNER JOIN sde.gdb_itemrelationships
ON Domain.UUID = sde.gdb_itemrelationships.DestID)
-- Find the names of the origin items in the relationships.
INNER JOIN sde.gdb_items
ON Domain.UUID = sde.gdb_itemrelationships.DestID

-- Extract the field definitions.
SELECT
  ClassDefs.Name AS "Class Name",
  fieldDef.value('Name[1]', 'nvarchar(max)') AS "Field Name",
  NULL AS "Subtype Name"
FROM
  @CLASS_DEFS AS ClassDefs
CROSS APPLY
  Definition.nodes('/*/GPFieldInfoExs/GPFieldInfoEx') AS FieldDefs(fieldDef)
WHERE
  fieldDef.value('DomainName[1]', 'nvarchar(max)') = @DOMAIN_NAME
UNION
SELECT
  ClassDefs.Name AS "Class Name",
  fieldDef.value('FieldName[1]', 'nvarchar(max)') AS "Field Name",
  fieldDef.value('(../../SubtypeName)[1]', 'nvarchar(max)') AS "Subtype Name"
FROM
  @CLASS_DEFS AS ClassDefs
CROSS APPLY
   Definition.nodes('/*/Subtypes/Subtype/FieldInfos/SubtypeFieldInfo') AS FieldDefs(fieldDef)
WHERE
  fieldDef.value('DomainName[1]', 'nvarchar(max)') = @DOMAIN_NAME
```

Function would be also availible within ArcGIS Pro 2.5 - DomainUsage

Domains: BEVx

Domain Name	Description	Field Typ	Domain Type	Split Policy	Merge F
AnnotationStatus	Valid annotation state values.	Short	Coded Value Domair	Duplicate	Default
BooleanSymbolValue	Valid values are Yes and No.	Short	Coded Value Domair	Duplicate	Default
DKM_Anlegungsmaßstab	Beschreibung des Anlegemaßstabes	Short	Coded Value Domair	Default	Default
DKM_Festpunkte	Beschreibung der DKM-Festpunkttypen	Text	Coded Value Domair	Default	Default
DKM_Grenzpunkte	Beschreibung der DKM-Grenzpunkttypen	Short	Coded Value Domair	Default	Default
DKM_Grundstückart	Beschreibung der DKM-Grundstücksnummerndarstellung	Short	Coded Value Domair	Default	Default
DKM_KennzNatur	Beschreibung der Kennzeichnung in der Natur	Text	Coded Value Domair	Default	Default
DKM_Nutzungsgrenzen	Beschreibung der DKM-Nutzungsgrenztypen	Short	Coded Value Domair	Default	Default
DKM_Nutzungslinientyp	Beschreibung des Typs der DKM-Nutzungslinien	Short	Coded Value Domair	Default	Default

Code	Description
0	Placed
1	Unplaced

Domain Usage

Workspace

BEVx

Domain

DKM_Sichtbarkeit

Search

Dataset	Subtype	Field
FPT		SICH_PNR
SSB		SICH_NR
SGG		SICH_PNR
SGG		SICH_SGP

Catalog | Domain Usage

Geodatabase-Version/Versioned Featureclasses

- **Geodatabase Version:**
  - **http://desktop.arcgis.com/en/arcmap/latest/manage-data/using-sql-with-gdbs/example-finding-the-geodatabase-release.htm**

```sql
-- Gets the geodatabase release from the workspace catalog item.

SELECT
  Definition.value('(/DEWorkspace/MajorVersion)[1]', 'smallint') AS "Major version",
  Definition.value('(/DEWorkspace/MinorVersion)[1]', 'smallint') AS "Minor version",
  Definition.value('(/DEWorkspace/BugfixVersion)[1]', 'smallint') AS "Bug fix version"
FROM
 sde.gdb_items AS items INNER JOIN
  (SELECT UUID
   FROM sde.gdb_itemtypes
   WHERE Name = 'Workspace') AS itemtypes
  ON items.Type = itemtypes.UUID
```

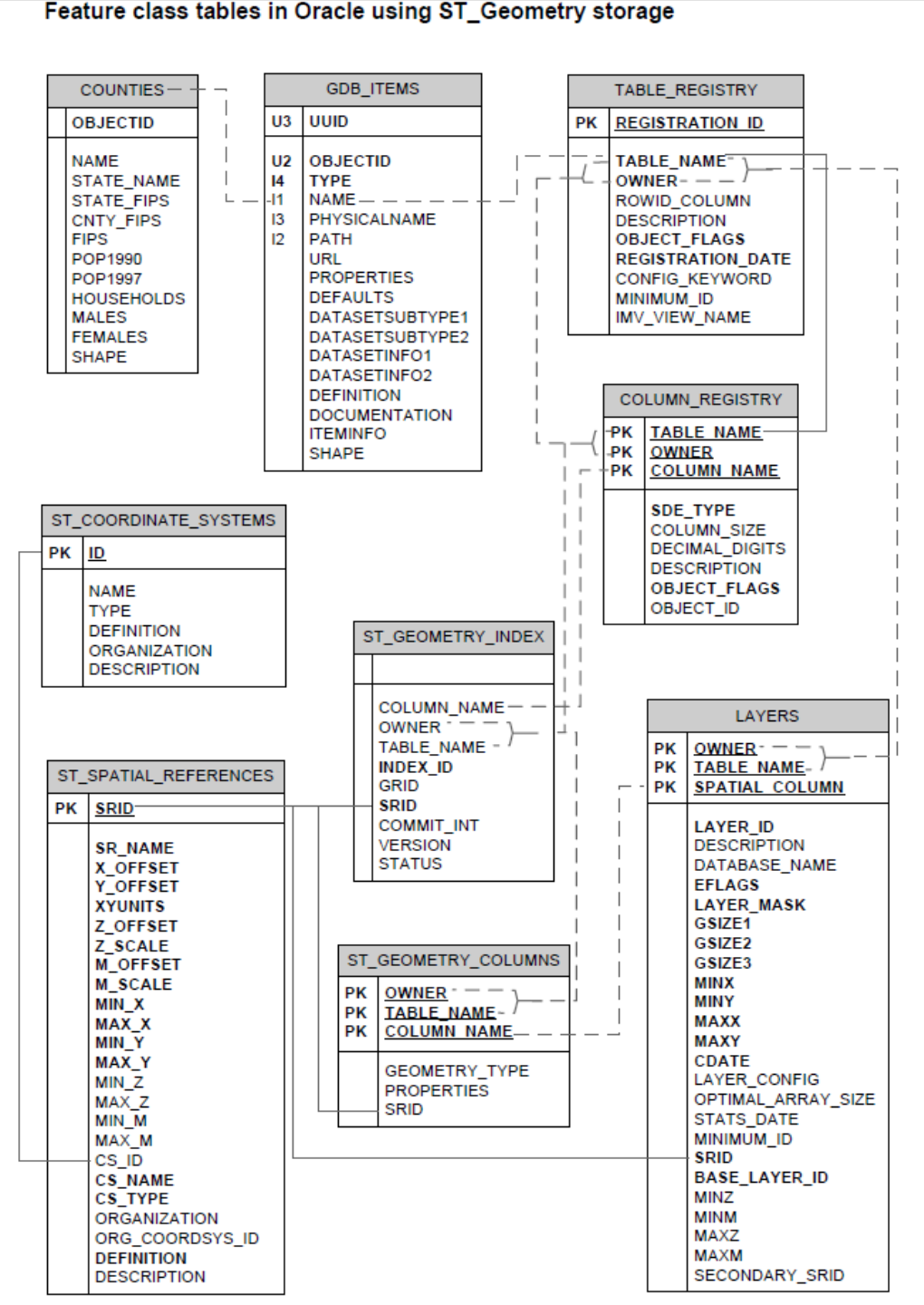
- **Versioned Featureclasses:**
  - **http://desktop.arcgis.com/en/arcmap/latest/manage-data/using-sql-with-gdbs/determining-which-data-is-versioned.htm**

```sql
--Returns a list of versioned datasets in the specified geodatabase

SELECT NAME AS "Versioned feature class"
FROM sde.GDB_ITEMS
WHERE Definition.exist('(/*/Versioned)[1]') = 1
AND Definition.value('(/*/Versioned)[1]', 'nvarchar(4)') = 'true'
```

Additional RepostitoryTables

- Beside GDB_xxx-Tables there exists other Tables to Maintain Enterprise Geodatabase
- Overview of the modells in the Desktop-Installation-Folder\Documentation



Information/Analysis from User-Tables

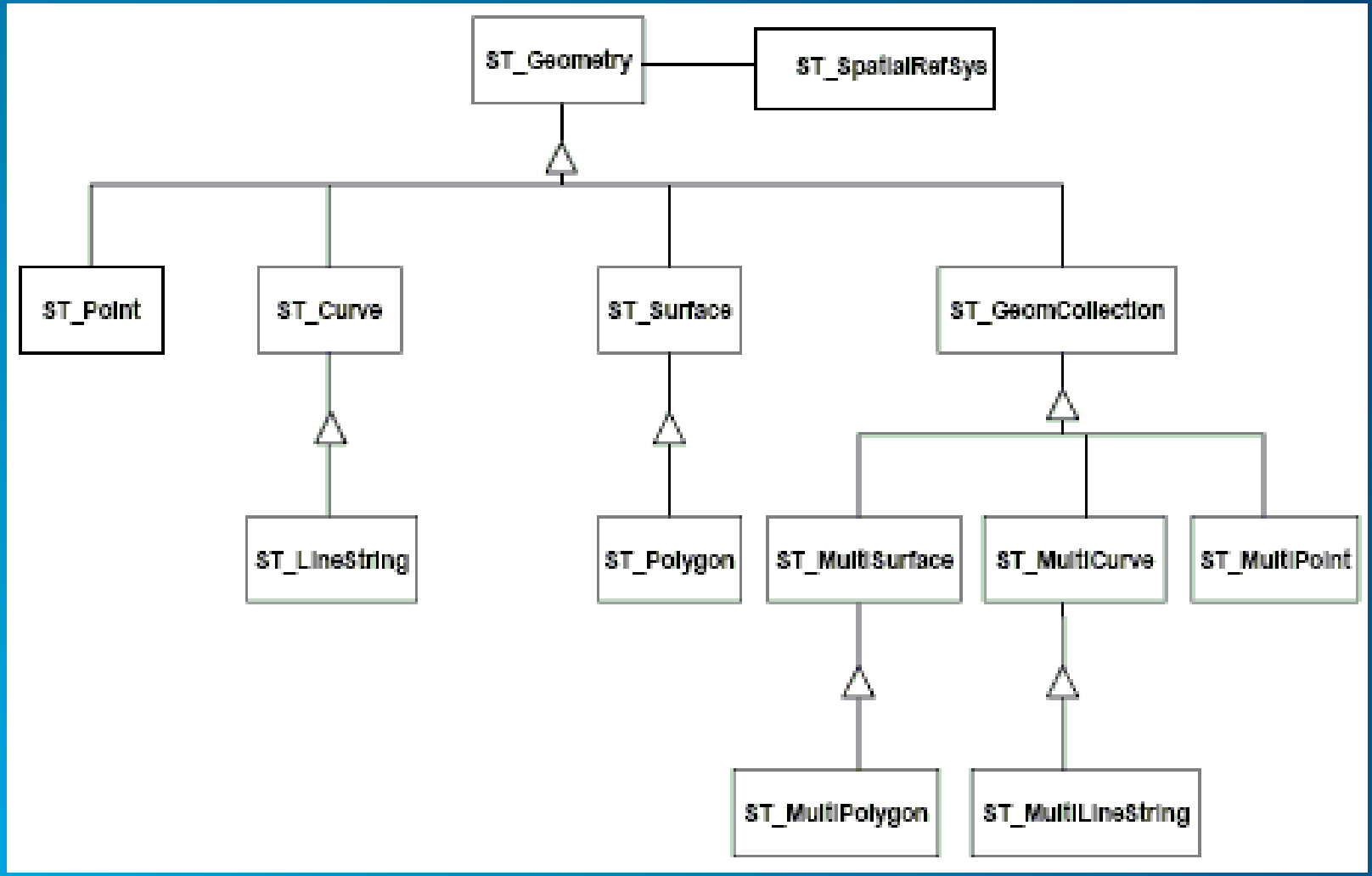
Quering geodatabase user-data

- Why use SQL when I have a GIS?
  - Use power of DBMS enginge to query and analyze your data
  - DBMS spatial methods for performing spatial analysis
  - Bridge between GIS and Business Intelligence / Insights
  - Sometimes you want a single result and not a map

What is a Spatial Type

- A Type that stores geometry data in a single spatial attribute
  - Geometry type, coordinates, dimension, spatial reference
- Spatial index
  - Improves spatial serach
- Relational and geometry operations and functions
  - Constructors – creates new geometry
  - Accessors – return property of a geometry
  - Relational – perform spatial operations
  - Geometry – transform from one geometry to another

Spatial Type

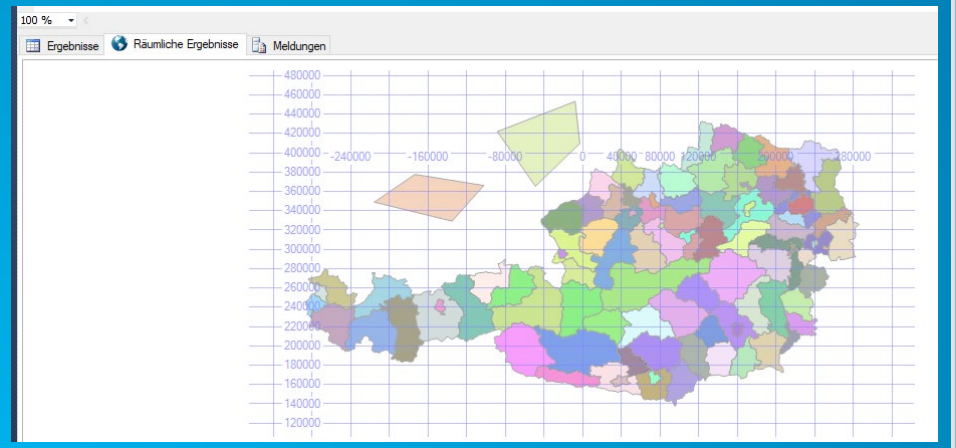


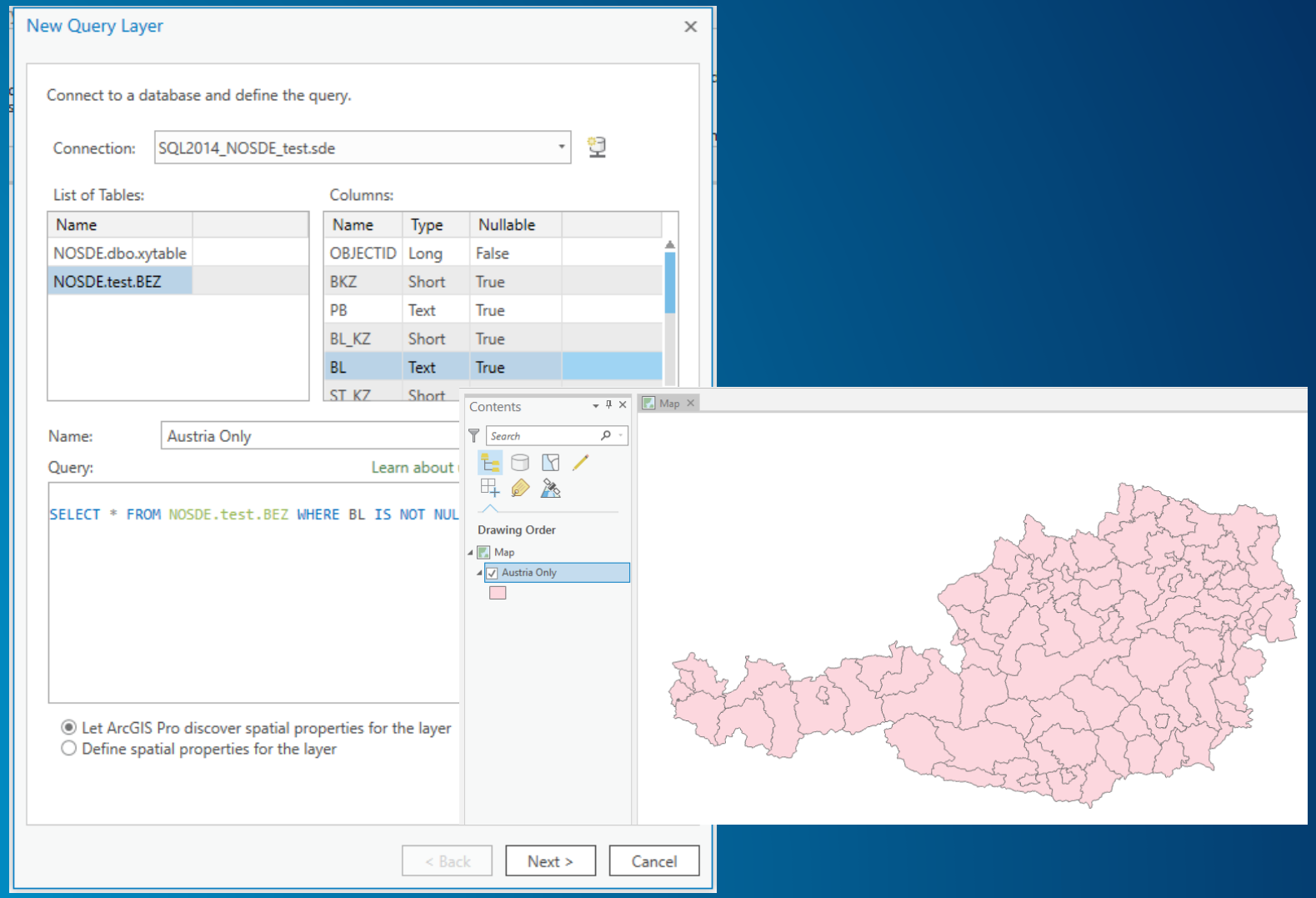
Benefits for a Spatial Type

- With SQL and a Spatial Type you can
  - Create Tables with a spatial attribute
  - Read and analyze spatial data
  - Insert, update and delete simple features
- Enhacnes efficiency
  - Data and methods are stored in the database
  - Applications access native dbms type
- Access using common API‘s and SQL
  - Standard functions
  - Well-known interchange formats

Viewing database data in ArcGIS

- **SQL Query with QueryLayer**



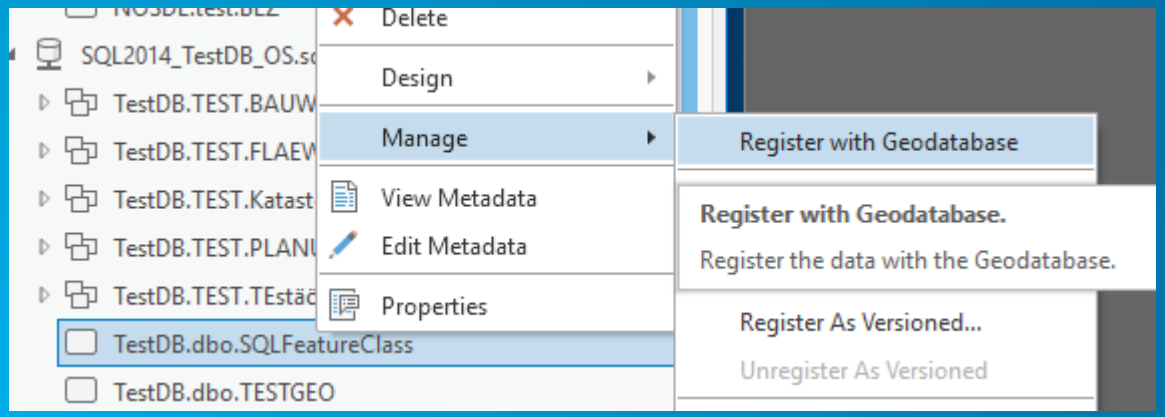


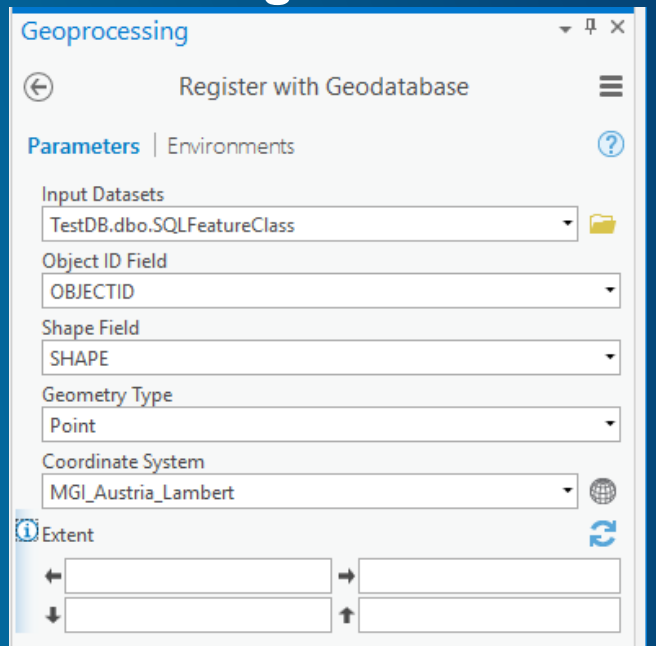
Create geodatabase feature classes using SQL

- Use SQL to create and populate tables

```sql
USE TESTDB
GO
create table SQLFeatureClass (OBJECTID INT,
                              NAME VARCHAR(20),
                              SHAPE GEOMETRY);
GO
```

- Need to register the table with the geodatabase to participate in the geodatabase functionality





Editing geodatabase feature classes using SQL

- What can you edit?
  - Simple features (points, lines, polygons)
  - Without geodatabase behavior
  - Use the Is_Simple function to determine whethever your data ca be edit
- Editing non-versioned tables
  - Edit tables directly
- Editing versioned tables
  - Edit special versioned views instead of tables

Populate fields with Spatial functions

Scenario for Showcase

- We want to add features via SQL by coordinates (eg. Addresses) from a 3rd Party Application
- We want to have information from a polygon featureclass in the created point (e.g. Name of Postal District)

Code for Trigger

```
prompt Check SRID for TRIGGER:
SELECT SRID from SDE.ST_GEOMETRY_COLUMNS where TABLE_NAME = 'POINT_BSP3';

prompt Create TRIGGERS:
CREATE OR REPLACE TRIGGER TRIGGER_POINT_BSP3 BEFORE INSERT ON POINT_BSP3 FOR EACH ROW
declare
amt varchar(100);

BEGIN
  -- Got Gemetrie - important is to use the correct SRID (see statment on top)
  :NEW.shape := sde.ST_PointFromText('POINT('||:NEW.RECHTSWERT||' '||:NEW.HOCHWERT||')',300014);
  -- Now Select based on Point Geometry the PostOffice
  SELECT POSTAMT
  into amt
  from plz_bsp3 where sde.st_intersects(sde.ST_PointFromText('POINT('||:NEW.RECHTSWERT||' '||:NEW.HOCHWERT||')',300014),shape) = 1;
  -- And Use value for the column on points
  :NEW.POSTAMT := amt;
END;
/
quit;
```

Code for Insert

```
-- Prepare - get REGISTRATION IDENTIFIED
SELECT registration_id FROM sde.table_registry WHERE table_name = 'POINT_BSP3' and owner = 'TEST2';

prompt Insert values in Database:
INSERT INTO POINT_BSP3 (OBJECTID,ADRESSE,RECHTSWERT,HOCHWERT) values (sde.version_user_ddl.next_row_id('TEST2', 83900),'Strandgasse 1'      ,435275,331335);
INSERT INTO POINT_BSP3 (OBJECTID,ADRESSE,RECHTSWERT,HOCHWERT) values (sde.version_user_ddl.next_row_id('TEST2', 83900),'Pruggerner Platz 4' ,441783,335079);
INSERT INTO POINT_BSP3 (OBJECTID,ADRESSE,RECHTSWERT,HOCHWERT) values (sde.version_user_ddl.next_row_id('TEST2', 83900),'Schladminger Ried 5',430494,325517);

quit
```

Important how to maintain „OBJECTID“

- „OBJECTID“ is Maintained by Geodatabase, so „OBJECTID“ couldn‘t be directly inserted with a vlaue
- Value nneds to be inserted with the function sde.version_user_dll.next_row_id, which get the next availible „OBJECTID“

- Wrong Insert

```
INSERT INTO POINT_BSP3 (OBJECTID,ADRESSE,RECHTSWERT,HOCHWERT) values (1,'Strandgasse 1',435275,331335);
INSERT INTO POINT_BSP3 (OBJECTID,ADRESSE,RECHTSWERT,HOCHWERT) values (2,'Pruggerner Platz 4',441783,335079);
INSERT INTO POINT_BSP3 (OBJECTID,ADRESSE,RECHTSWERT,HOCHWERT) values (3,'Schladminger Ried 5',430494,325517);
```

- Obtain RegistrationID from TABLE_REGISTRY-Table

```
-- Prepare - get REGISTRATION IDENTIFIED
SELECT registration_id FROM sde.table_registry WHERE table_name = 'POINT_BSP3' and owner = 'TEST2';
```

- Correct Insert with function

```
prompt Insert values in Database:
INSERT INTO POINT_BSP3 (OBJECTID,ADRESSE,RECHTSWERT,HOCHWERT) values (sde.version_user_ddl.next_row_id('TEST2', 83900),'Strandgasse 1'      ,435275,331335);
INSERT INTO POINT_BSP3 (OBJECTID,ADRESSE,RECHTSWERT,HOCHWERT) values (sde.version_user_ddl.next_row_id('TEST2', 83900),'Pruggerner Platz 4' ,441783,335079);
INSERT INTO POINT_BSP3 (OBJECTID,ADRESSE,RECHTSWERT,HOCHWERT) values (sde.version_user_ddl.next_row_id('TEST2', 83900),'Schladminger Ried 5',430494,325517);
```

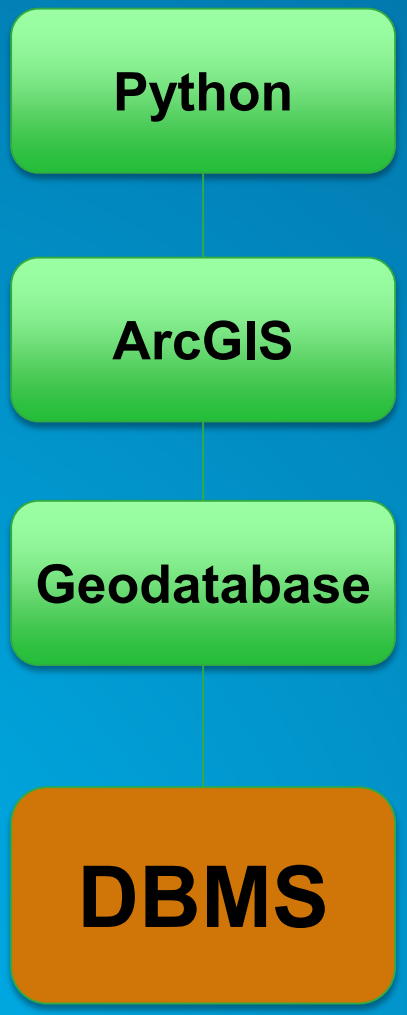
ST_Functions

- **Functions to work with the Geometry adn data**

ST_AsBinary	ST_LineFromShape (PostgreSQL only)	ST_MPolyFromShape (PostgreSQL only)	ST_PointFromText (Oracle only)
ST_AsText	ST_LineFromText (Oracle only)	ST_MPolyFromText (Oracle only)	ST_PointFromWKB
ST_Curve (Oracle only)	ST_LineFromWKB	ST_MPolyFromWKB	ST_PolyFromShape (PostgreSQL only)
ST_GeomCollection	ST_LineString	ST_MultiCurve (Oracle only)	ST_PolyFromText (Oracle only)
ST_GeomCollFromShape (PostgreSQL only)	ST_MLineFromShape (PostgreSQL only)	ST_MultiLineString	ST_PolyFromWKB
ST_GeomCollFromWKB (PostgreSQL only)	ST_MLineFromText (Oracle only)	ST_MultiPoint	ST_Polygon
ST_Geometry	ST_MLineFromWKB	ST_MultiPolygon	ST_Surface (Oracle only)
ST_GeomFromShape (PostgreSQL only)	ST_MPointFromShape (PostgreSQL only)	ST_MultiSurface (Oracle only)	ST_Transform
ST_GeomFromText (Oracle only)	ST_MPointFromText (Oracle only)	ST_Point	
ST_GeomFromWKB	ST_MPointFromWKB	ST_PointFromShape (PostgreSQL only)	

- **http://desktop.arcgis.com/en/arcmap/latest/manage-data/using-sql-with-gdbs/a-quick-tour-of-sql-functions-used-with-st-geometry.htm**

Guidelines for using SQL and the geodatabase



- Understanding the geodatabase system and their structure
- Avoid changing data that affectes geodatabase software behavior
- Geodatabase awareness
  - You have it
  - The database does not

Guidelines for using SQL and the geodatabase

	GDB System tables	Simple FC / Tables	Complex FC / Tables
QUERY	✓	✓	
Edit/Update		✓	
Insert		✓	

Guidlines for using SQL and the geodatabase

- DO NOT update the OBJECTID(row_id) value

- DO NOT modify geometries for feature classes participate in non simple data as
  - Topologies, geometric networks, network datasets, terrains, parcel fabric,
  - Geodatabase replication, schematic datasets, feature-linked annotation, …
- DO NOT update attributes that define geodatabase behavior
  - Enable/Disable attributes, ancillary attributes, weiight attributes, …

  - Use Is_Simple to check
  - http://desktop.arcgis.com/en/arcmap/latest/manage-data/using-sql-with-gdbs/is-simple.htm

Guidlines for using SQL and the geodatabase

- Do perform spatial operations
- Do query spatial and attribute information
- Do INSERT, UPDATE and DELETE geometries
  As long you pay attention to behavior
- Do INSERT, UPDATE and DELETE attribute data
  As long you pay attention to behavior
- Do write efficent SQL

Resources

- Comprehensive documenattion covering
  - Accessing dataset properties
  - Editing geodatabase data
  - Esri spatial and raster type reference

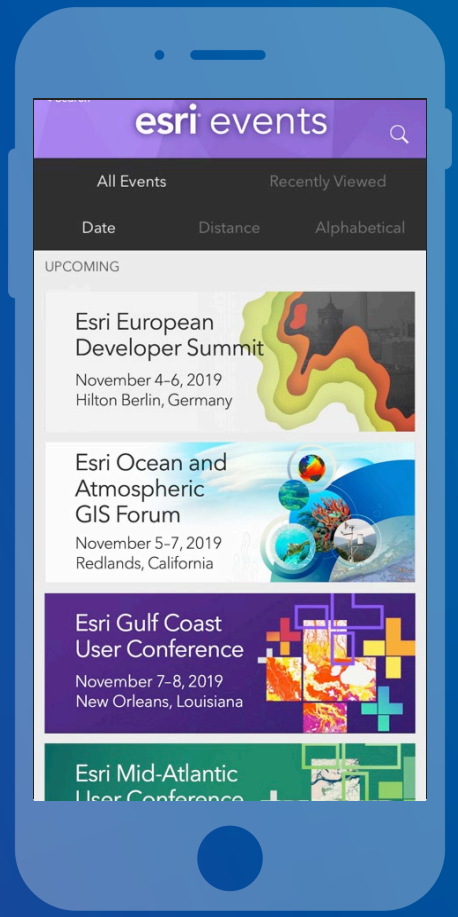
- Get startet at
  - http://desktop.arcgis.com/en/arcmap/latest/manage-data/using-sql-with-gdbs/sql-and-enterprise-geodatabases.htm

  - http://desktop.arcgis.com/en/arcmap/latest/manage-data/using-sql-with-gdbs/xml-column-queries.htm

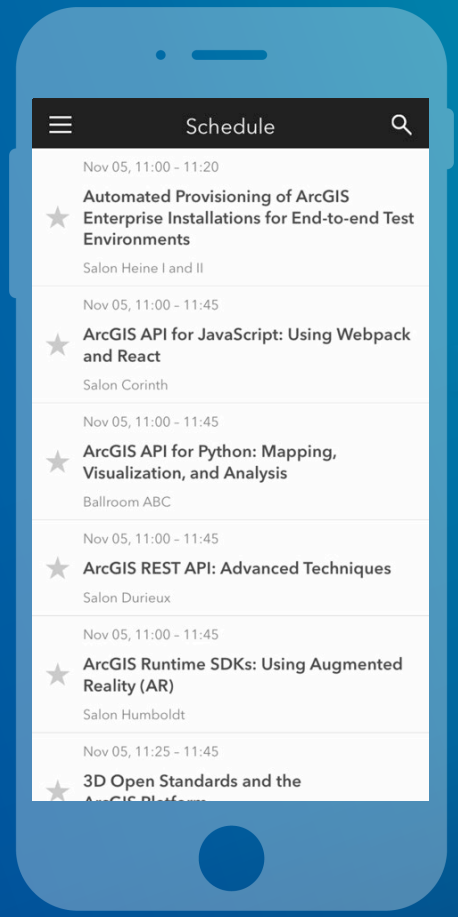
Questions?

Please Take Our Survey on the App

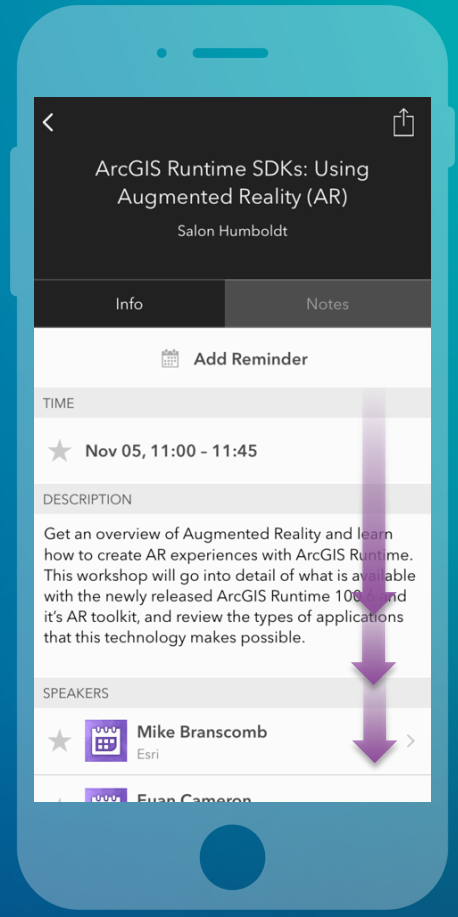
Download the Esri Events app and find your event



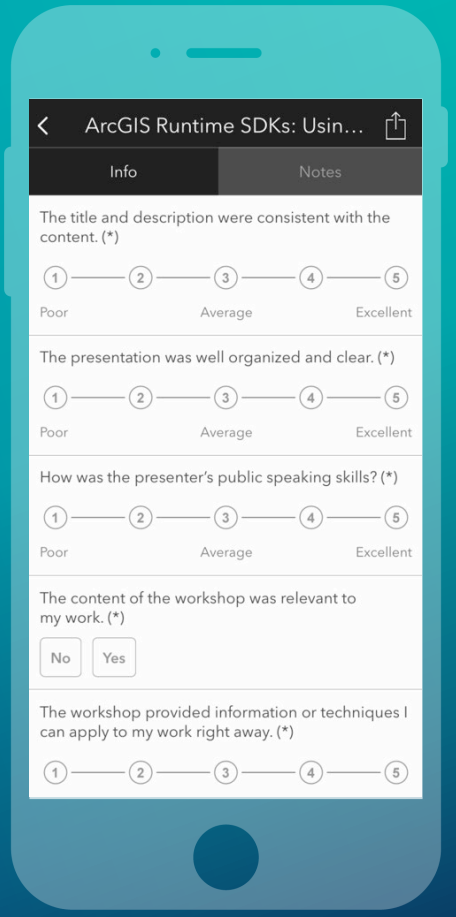
Select the session you attended



Scroll down to find the feedback section



Complete answers and select "Submit"



esri

THE SCIENCE OF WHERE