

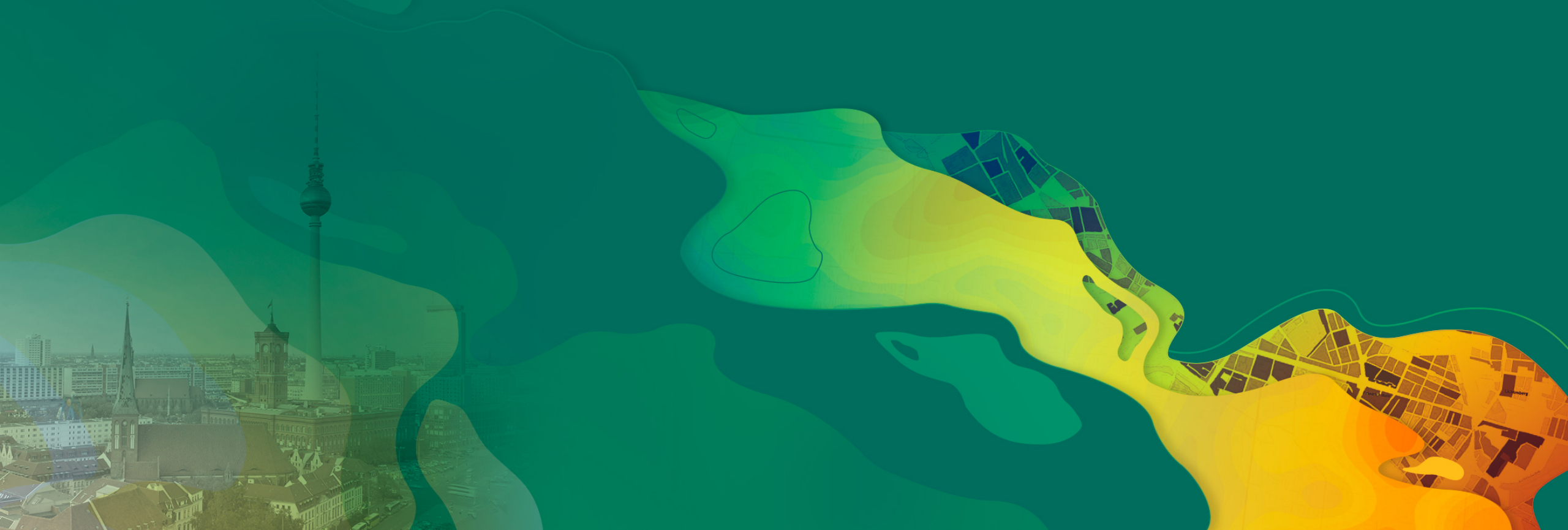


DEVELOPER  
SUMMIT

EUROPE

# Accessing Your Enterprise Geodatabase using SQL

Gerhard Trichtl

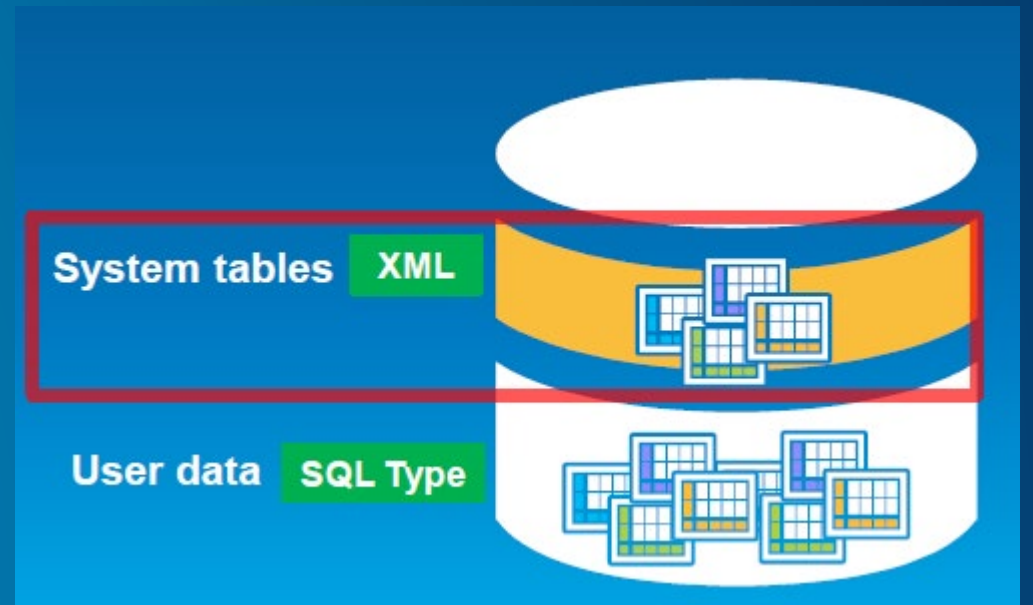


# What is the Geodatabase

- **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**

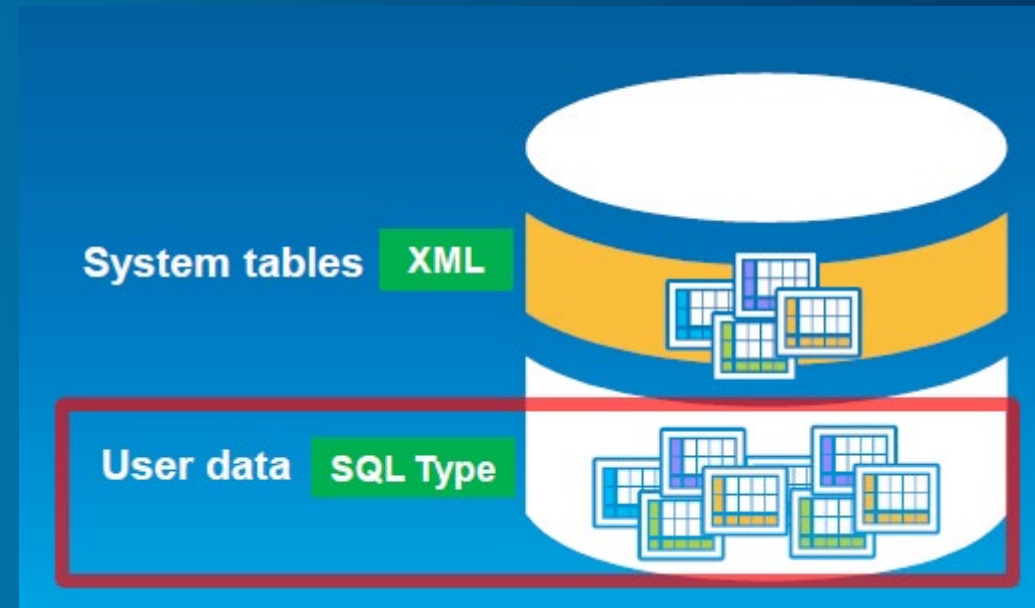
# Geodatabase system tables

- 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



# 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

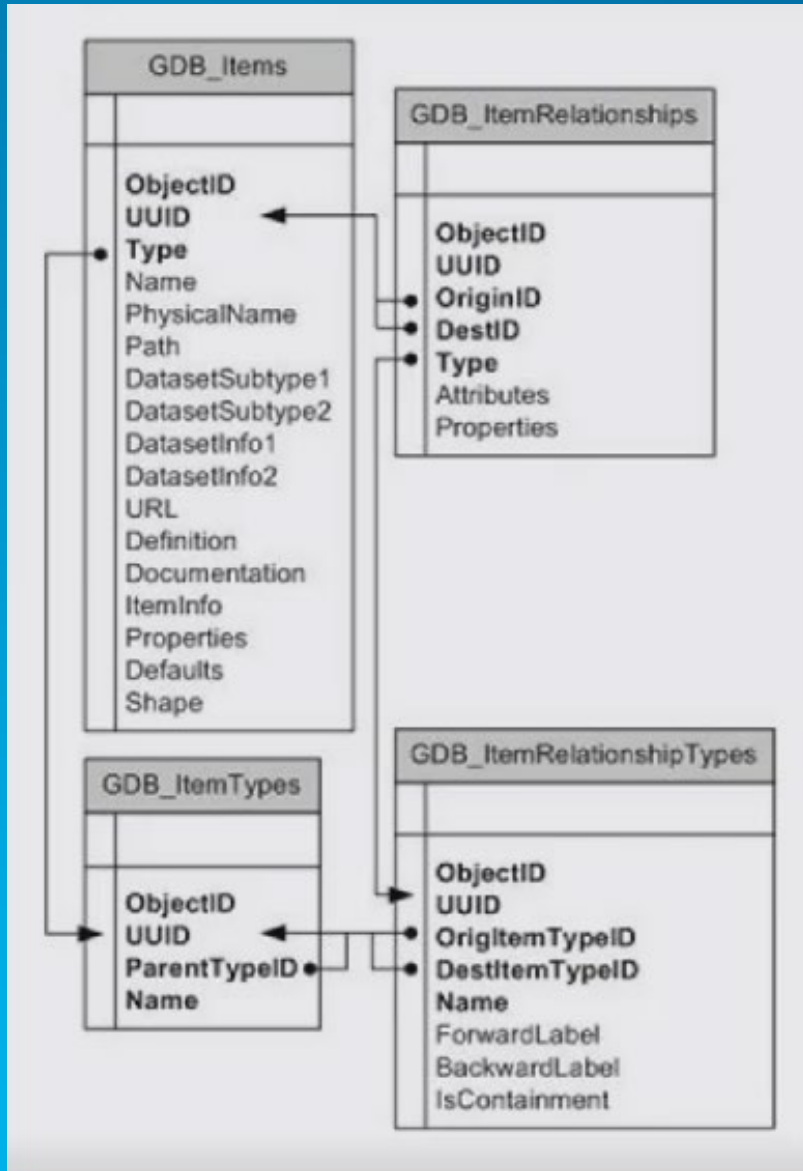


## Accessing data in a DBMS

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

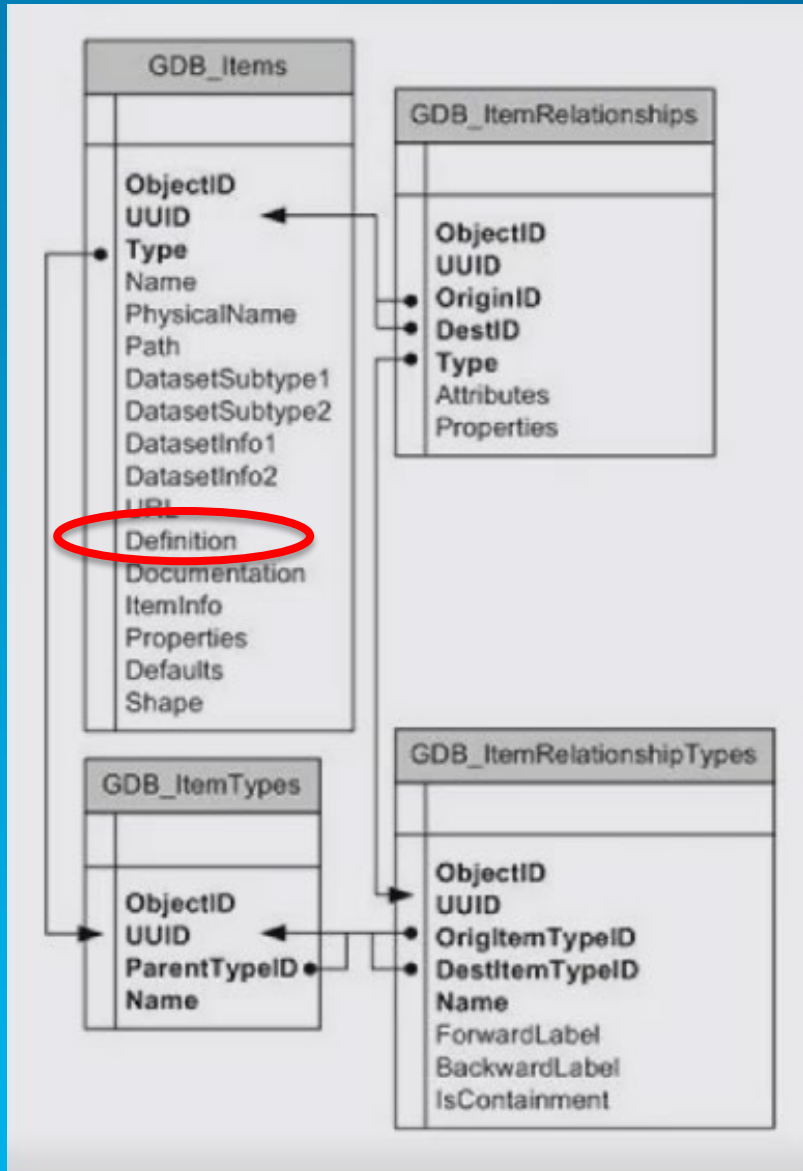
Geodatabase	Database – Simple Feature Access
DB2	ALTIBASE
Informix	Dameng
ORACLE	Teradata
PostgreSQL	Netezza
Microsoft SQLServer	DB2, Informix
SAP HANA	ORACLE, PostgreSQL
	Microsoft SQLServer, SAP HANA

# 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

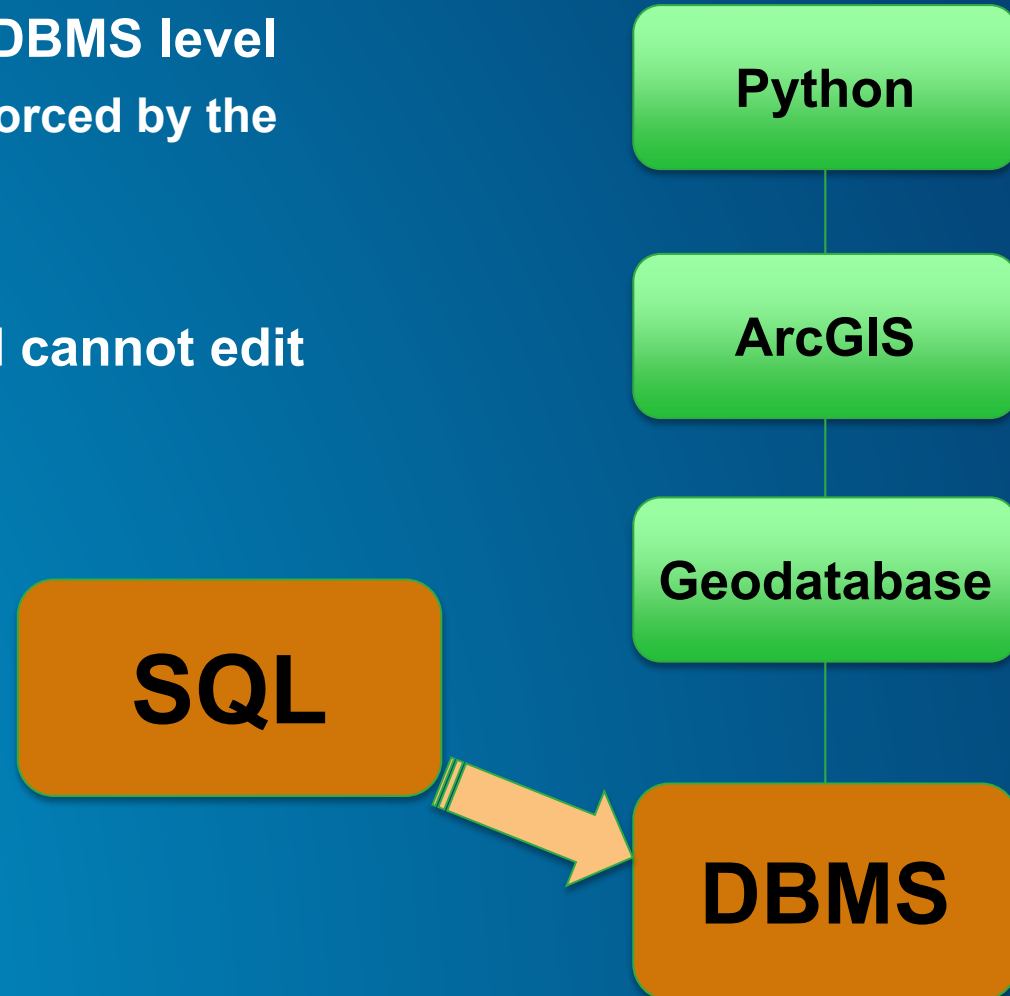
# Geodatabase schema – four main system tables



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

# 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





The image shows a screenshot of a SQL Server environment. On the left, the Enterprise Manager interface displays a tree view of a database named 'oror'. The 'Tabellen' (Tables) folder is expanded, showing a list of tables including 'sde.GDB\_ITEMS', which is currently selected. Other tables listed include 'sde.GDB\_ITEMRELATIONSHIPS', 'sde.GDB\_ITEMRELATIONSHIPTYPES', 'sde.GDB\_ITEMTYPES', 'sde.GDB\_REPLICATALOG', 'sde.GDB\_TABLES\_LAST\_MODIFIED', and several 'sde.i' tables (i2 through i5). On the right, a SQL Query window titled 'SQLQuery3.sql - w-l...RGIS\g.trichtl (59)' contains a script for selecting the top 1000 records from the 'sde.GDB\_ITEMS' table. The query is as follows:

```
/* Skript für SelectTop  
SELECT TOP 1000 [ObjectID]  
    , [UUID]  
    , [Type]  
    , [Name]  
    , [PhysicalName]  
    , [Path]  
    , [Url]  
    , [Properties]  
    , [Defaults]  
    , [DatasetSubtype1]  
    , [DatasetSubtype2]  
    , [DatasetInfo1]  
    , [DatasetInfo2]  
    , [Definition]  
    , [Documentation]  
    , [ItemInfo]  
    , [Shape]  
FROM [TestDB].[sde].[GDB_
```

# The geodatabase schema

# 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>

```
--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>

```
--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
```

# 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>

```
-- 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:**

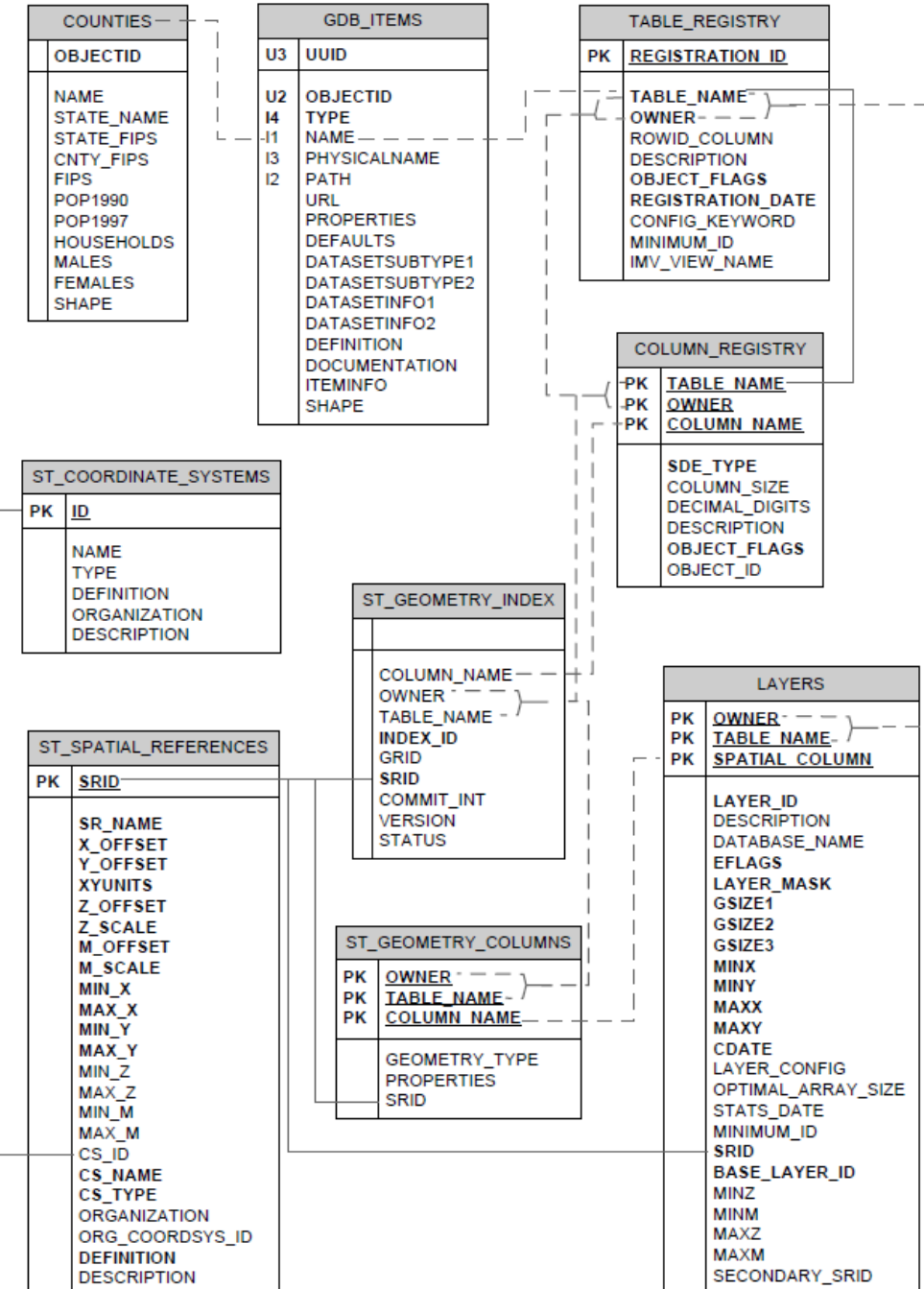
- <http://desktop.arcgis.com/en/arcmap/latest/manage-data/using-sql-with-gdbs/determining-which-data-is-versioned.htm>

```
--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 Repository Tables

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

Feature class tables in Oracle using ST\_Geometry storage



# Review additional Repository Tables

100 %

Ergebnisse Meldungen

	srid	description	auth_name	auth_srid	false_x	false_y	xyuni
1	1	NULL	EPSG	4326	-400	-400	1000
2	2	NULL	EPSG	4326	-400	-400	1000
3	3	NULL	EPSG	4326	-400	-400	1000
4	4	NULL	EPSG	4326	-400	-400	1111
5	5	NULL	SQL Server Geometry	3857	-20037700	-30241100	1000
6	6	NULL	SQL Server Geometry	31255	-5622500	-15001000	1000
7	7	NULL		0	-5622500	-15001000	1000
8	8	NULL		0	-5622500	-15001000	1000
9	9	NULL	SQL Server Geometry	31256	-5622500	-15001000	1000
10	10	NULL	SQL Server Geometry	0	-5622500	-15001000	1000
11	11	NULL		0	-17463800	-46132600	3048
12	12	NULL		0	-5622500	-15001000	1000
13	13	NULL	SQL Server Geometry	31256	-5622500	-15001000	1000
14	14	NULL	SQL Server Geometry	31287	-35830000	-29978200	1000
15	15	NULL	SQL Server Geometry	31287	-35830000	-29978200	1000
16	16	NULL	SQL Server Geometry	31287	-35830000	-29978200	1000
17	17	NULL	SQL Server Geometry	3857	-20037700	-30241100	1000
18	18	NULL		0	-5622500	-15001000	1000
19	19	NULL	SQL Server Geometry	0	-5622500	-15001000	1000

# Querying geodatabase user-data

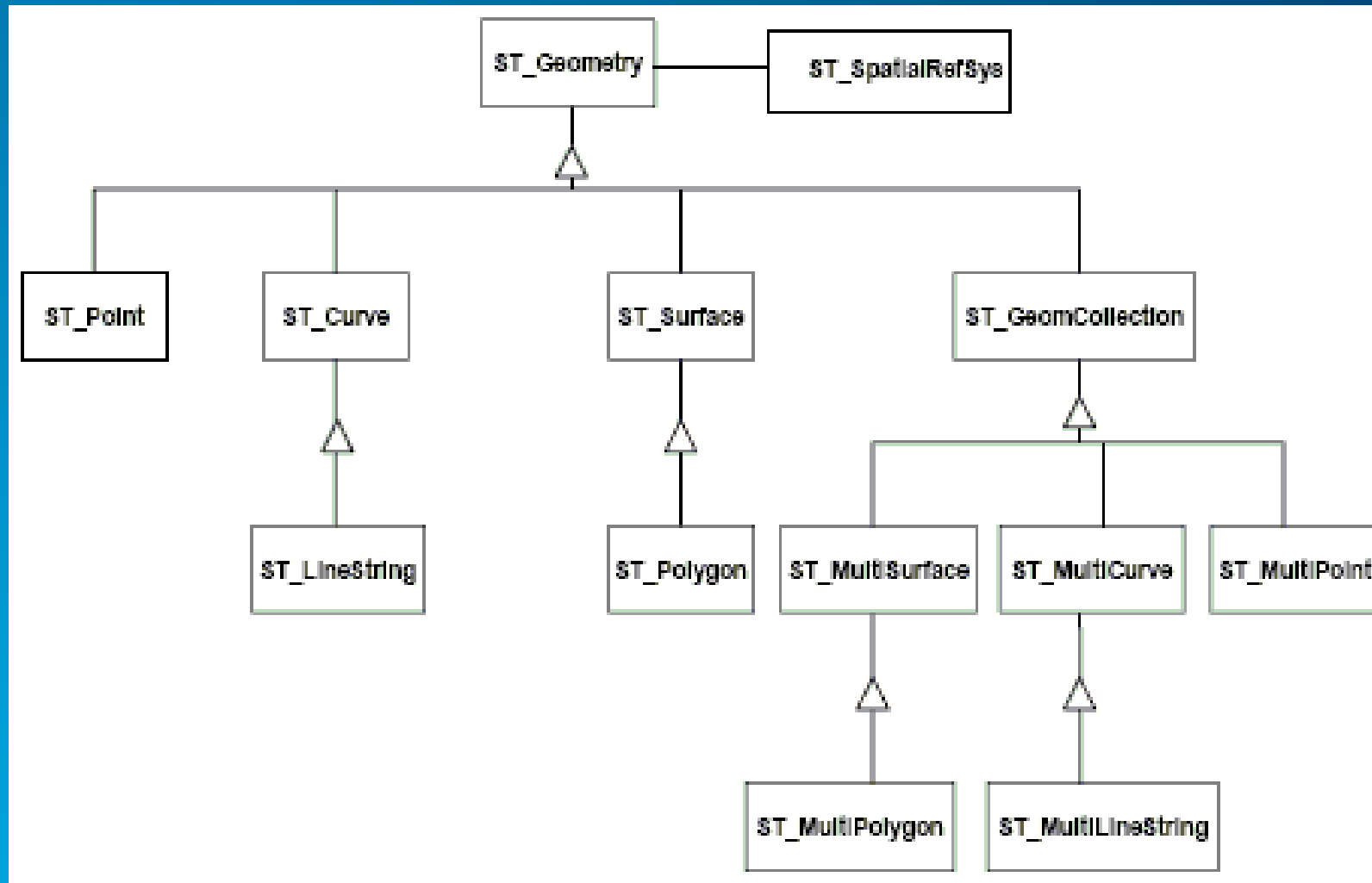
- **Why use SQL when I have a GIS?**
  - **Use power of DBMS engine to query and analyze your data**
  - **DBMS spatial methods for performing spatial analysis**
  - **Bridge between GIS and Business Intelligence / Insights**
  - **Sometimes you want a single result and not a map**

# 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 search
- **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

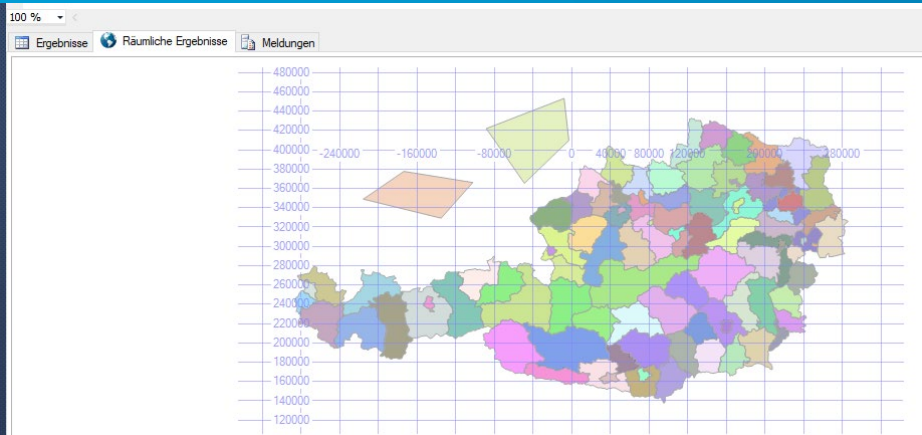


# Benefits for a Spatial Type

- **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
- **Enhances 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



New Query Layer

Connect to a database and define the query.

Connection: SQL2014\_NOSDE\_test.sde

List of Tables:

Name
NOSDE.dbo.xytable
NOSDE.test.BEZ

Columns:

Name	Type	Nullable
OBJECTID	Long	False
BKZ	Short	True
PB	Text	True
BL_KZ	Short	True
BL	Text	True
ST_KZ	Short	True

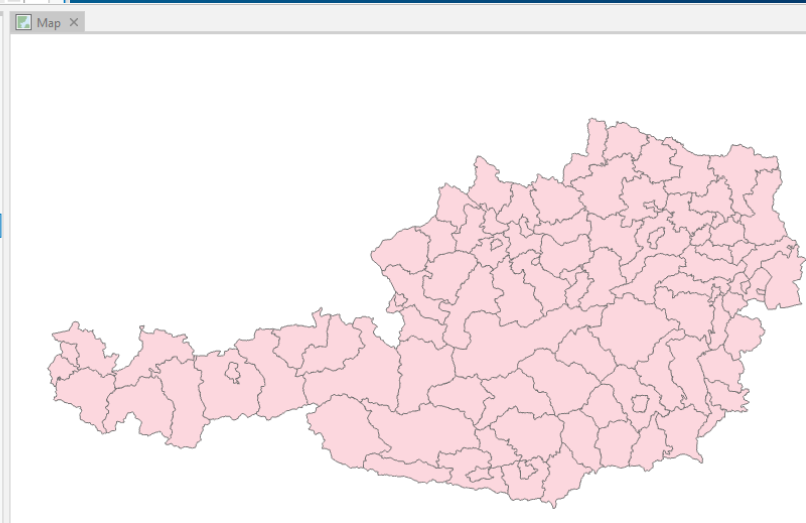
Name: Austria Only

Query: [Learn about using SQL queries](#)

```
SELECT * FROM NOSDE.test.BEZ WHERE BL IS NOT NULL
```

Let ArcGIS Pro discover spatial properties for the layer  
 Define spatial properties for the layer

< Back Next > Cancel

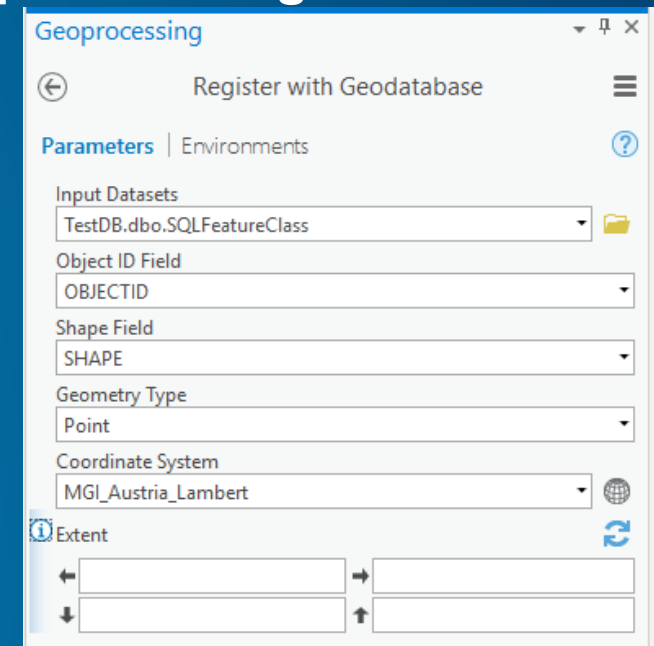
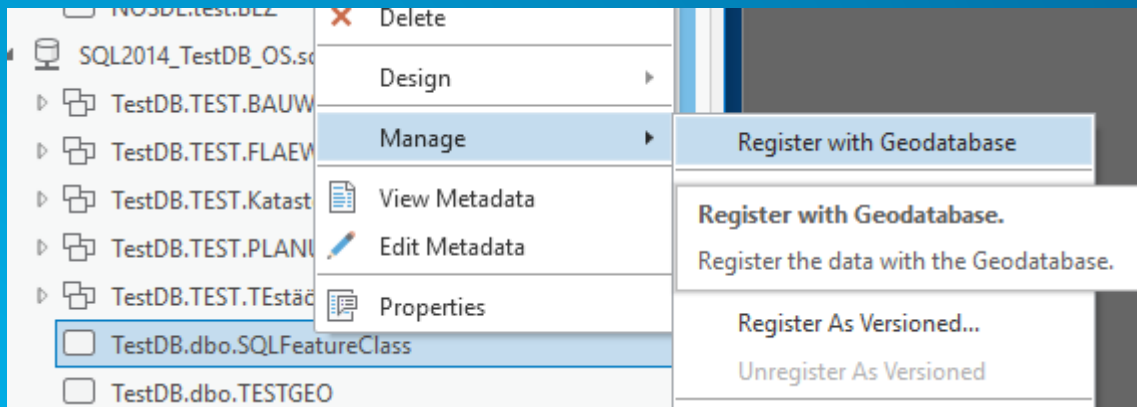


# Create geodatabase feature classes using SQL

- Use SQL to create and populate tables

```
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 whether your data can be edited
- **Editing non-versioned tables**
  - Edit tables directly
- **Editing versioned tables**
  - Edit special versioned views instead of tables

## Important how to maintain „OBJECTID“

- „OBJECTID“ is Maintained by Geodatabase, so „OBJECTID“ couldn't be directly inserted with a value
- Value needs to be inserted with the function `sde.version_user_ddl.next_row_id`, which gets the next available „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 and data**

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

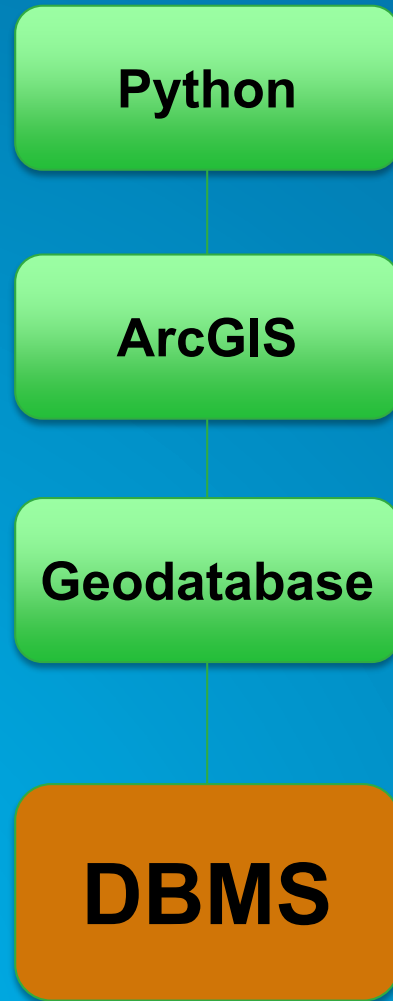
- <http://desktop.arcgis.com/en/arcmap/latest/manage-data/using-sql-with-gdbs/a-quick-tour-of-sql-functions-used-with-st-geometry.htm>



# Populate fields with Spatial functions



# 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 doesn't

# Guidelines for using SQL and the geodatabase

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

# Guidelines 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, weight attributes, ...
- Use Is\_Simple to check
- <http://desktop.arcgis.com/en/arcmap/latest/manage-data/using-sql-with-gdbs/is-simple.htm>

# Guidelines 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 efficient SQL**

# Resources

- **Comprehensive documentation covering**
  - **Accessing dataset properties**
  - **Editing geodatabase data**
  - **Esri spatial and raster type reference**
- **Get started 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>

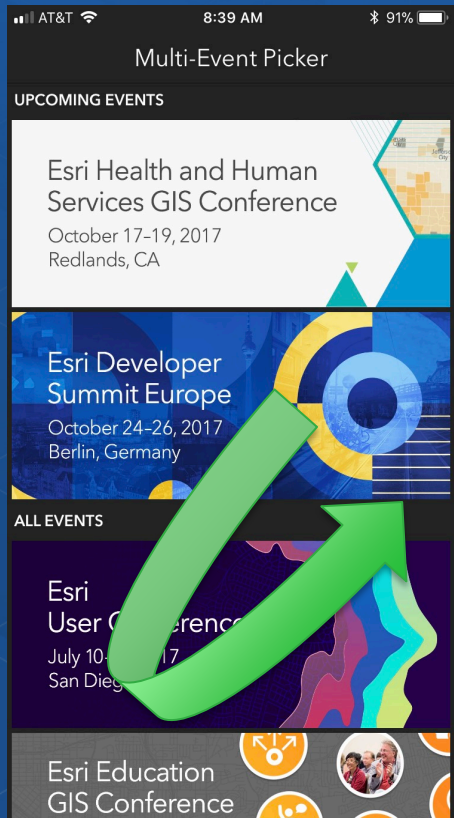
**Questions?**

Thank You to Our Sponsors

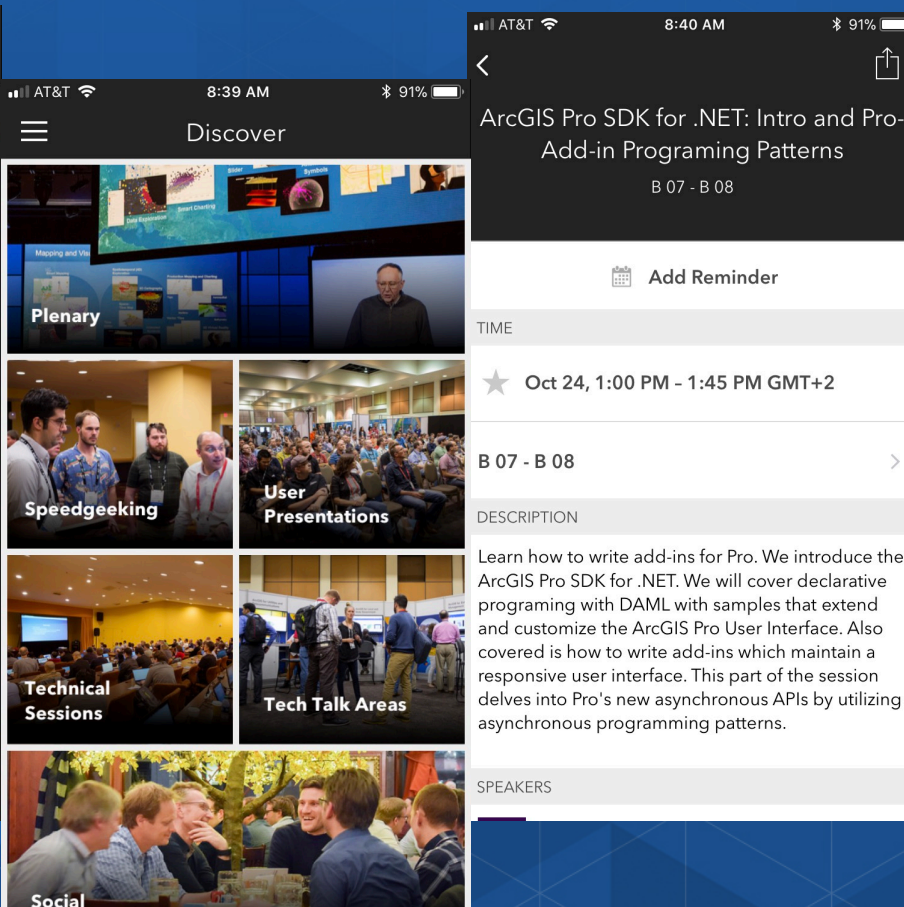
con•terra  
geo@com

# Please Take Our Survey!

Download the Esri Events app  
and go to DevSummit

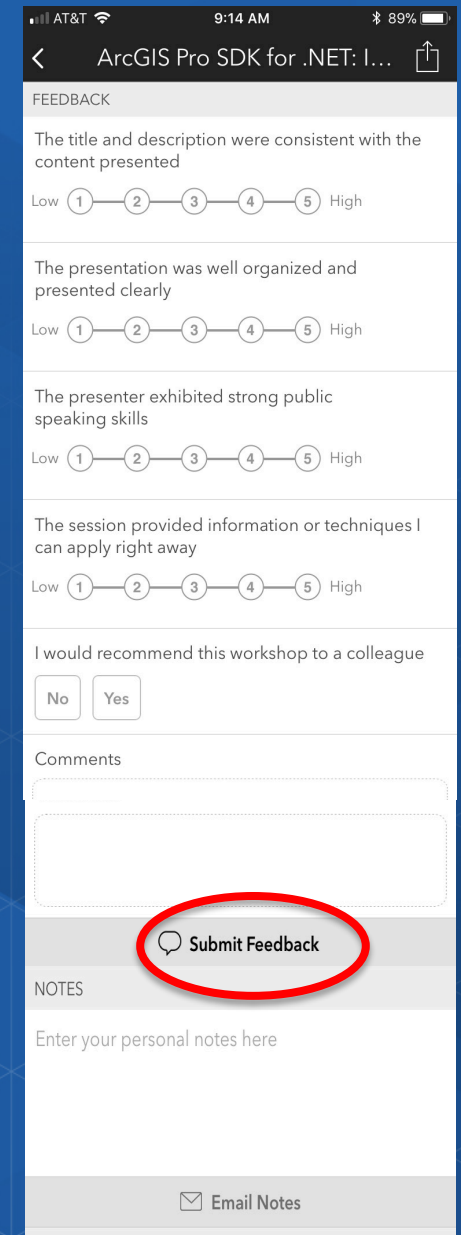


Select the session you attended



Scroll down to the  
“Feedback” section

Complete Answers,  
add a Comment,  
and Select “Submit”







esri

THE  
SCIENCE  
OF  
WHERE