

**SQL and SSQL**

IT4GIS  
Keith T. Weber, GISP  
GIS Director  
ISU-GIS Training and Research Center

Pocatello | Idaho Falls | Meridian | Twin Falls

**Idaho State**  
UNIVERSITY

---

---

---

---

---

---

---

---

**Definitions to get started**

- SQL = Structured Query Language
- SSQL = Spatial SQL
- GPL = Graphical Presentation Language

Pocatello | Idaho Falls | Meridian | Twin Falls

**Idaho State**  
UNIVERSITY

---

---

---

---

---

---

---

---

**SQL- A Review**

- SQL is a simple language used to query (question) an ODBC-compliant database and retrieve data.
  - SQL is *not* simple or standard
  - S = structured

Pocatello | Idaho Falls | Meridian | Twin Falls

**Idaho State**  
UNIVERSITY

---

---

---

---

---

---

---

---

## Structure

- The most basic SQL statement is:
  - **SELECT** \* from *database.table*
- Let's dissect this statement
  - **SELECT** is the command
  - \* is a wildcard = i.e., everything and anything
  - Database.table is the target of the query

Pocatello | Idaho Falls | Meridian | Twin Falls




---

---

---

---

---

---

---

---

## A Little More...

- The previous SQL statement selected everything from a table
- But, how do we select only a portion of a table?
  - The **WHERE** CLAUSE

Pocatello | Idaho Falls | Meridian | Twin Falls




---

---

---

---

---

---

---

---

## WHERE CLAUSE



- **WHERE** *conditional operator*
- For example:
  - **SELECT** \* from database.table **WHERE**  
CITY\_NAME = 'Pocatello'



Pocatello | Idaho Falls | Meridian | Twin Falls




---

---

---

---

---

---

---

---

## Types of Conditional Ops

- **Simple** (as in the previous example)
- **Compound**
  - Let's say we want to select and work with all records describing Pocatello and Blackfoot
- We could select and work with them individually using two discrete **Simple** statements or use **Conditional** operators

Pocatello | Idaho Falls | Meridian | Twin Falls



---

---

---

---

---

---

---

---

## Combining Statements using Conditional Operator Expressions

- Instead of:
  - **SELECT** \* from database.table **WHERE**  
CITY\_NAME = 'Pocatello'
  - ...do some work, and then
  - **SELECT** \* from database.table **WHERE**  
CITY\_NAME = 'Blackfoot'
  - ...do some more work

Pocatello | Idaho Falls | Meridian | Twin Falls



---

---

---

---

---

---

---

---

## We Can Use...

- A **Compound** expression combining two or more single expressions using either:
  - **AND**
  - **OR**
- In our example, which shall we use?

Pocatello | Idaho Falls | Meridian | Twin Falls



---

---

---

---

---

---

---

---

## OR

- **SELECT** \* from database.table **WHERE**  
CITY\_NAME = 'Pocatello'  
**OR**  
CITY\_NAME = 'Blackfoot'

---

---

---

---

---

---

---

---

## Why OR?

- Before a record (entity) is returned as a result of a query, the record must satisfy EACH **WHERE** clause if **AND** is used.
- When **OR** is used, a record must satisfy only one of the **WHERE** clauses.

---

---

---

---

---

---

---

---

## This is SQL

- What is SQL?
  - **Spatial** Structured Query Language
  - Or SQL for Spatially-enabled relational databases (i.e., object-relational databases)
    - Informix
    - Oracle
    - IBM DB2
    - MS SQL Server
    - PostgreSQL

---

---

---

---

---

---

---

---

## An Example

- **SELECT** *residence.geometry*  
FROM residence  
**WHERE** Type = 'single family'

What is different about this expression?  
*residence.geometry*

---

---

---

---

---

---

---

---

## Why is *\*.geometry* important?

- Until now, we have been returning all fields  
– (SELECT \* FROM...)
- Now, we only want to see the *geographic feature(s)* returned by the query
- To do that, we instruct SSQL to select the geometry (*.geometry*) of the TABLE of interest (residence)

---

---

---

---

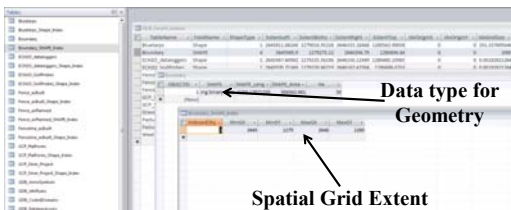
---

---

---

---

## GIS Layers are Tables?



---

---

---

---

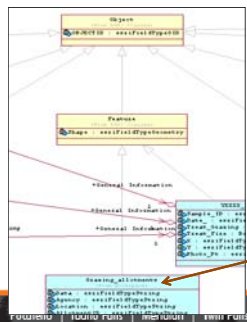
---

---

---

---

## Object Relational



- OBJECTID inherited from Object class
- SHAPE inherited from a class called Feature

This could be "Boundary"

---

---

---

---

---

---

---

---

## Geometry Data Type

- Earlier we talked a lot about data types for more traditional attributes (e.g., long integer, text, etc.)
- Recall, RDBMS can store OBJECTS natively
- What *data type* is used to store OBJECTS?

---

---

---

---

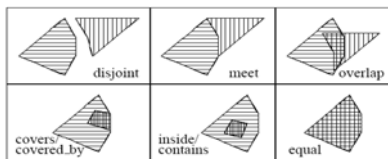
---

---

---

---

## SSQL with Topology



- Similar to intersect and union
- These are conditional operators that are written into the **WHERE** clause

---

---

---

---

---

---

---

---



## Key Concepts

- SQL is highly structured
- Spatial SQL builds upon SQL but remains within the general framework
- SSQL requires an object relational, spatially enabled database
- The \*.geometry table is queried to return features
  - which are stored in the table as a LOB field
  - Along with other topological attributes

Pocatello | Idaho Falls | Meridian | Twin Falls

Idaho State  
UNIVERSITY

---

---

---

---

---

---

---

---

## Questions?

- Your assignment is:
  - Review basic SQL (as necessary)
  - Read Egenhofer's early (1994) manuscript on Spatial SQL theory
  - Review the Spatial Data Extender white paper
  - Review the ArcGIS SSQL Help (use the link)
  - Complete the IBM DB2 SSQL exercise

Pocatello | Idaho Falls | Meridian | Twin Falls

Idaho State  
UNIVERSITY

---

---

---

---

---

---

---

---