Data Models and SQL for GIS

John Porter
Department of Environmental Sciences
University of Virginia

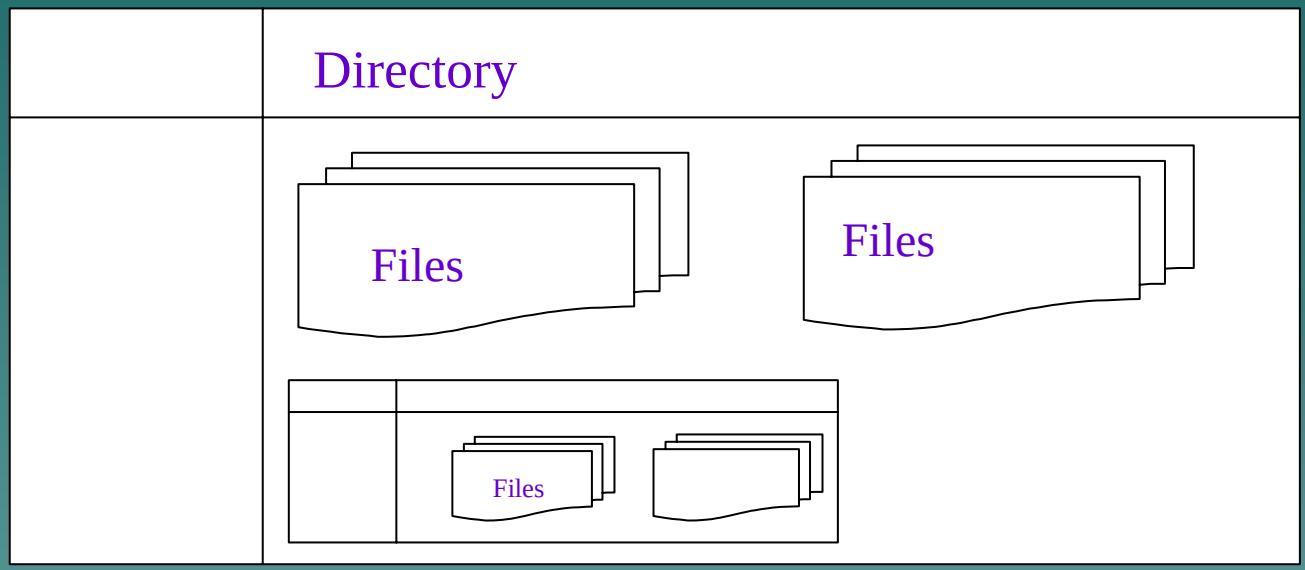
Why Talk about Databases in a GIS Class?

- The goal of GIS is to analyze and display spatial data
- Vector data are almost always stored in some form of database
  - Raster data may be stored in special data structures
- Often we need to link non-GIS data to our data layers

Database Management System Types

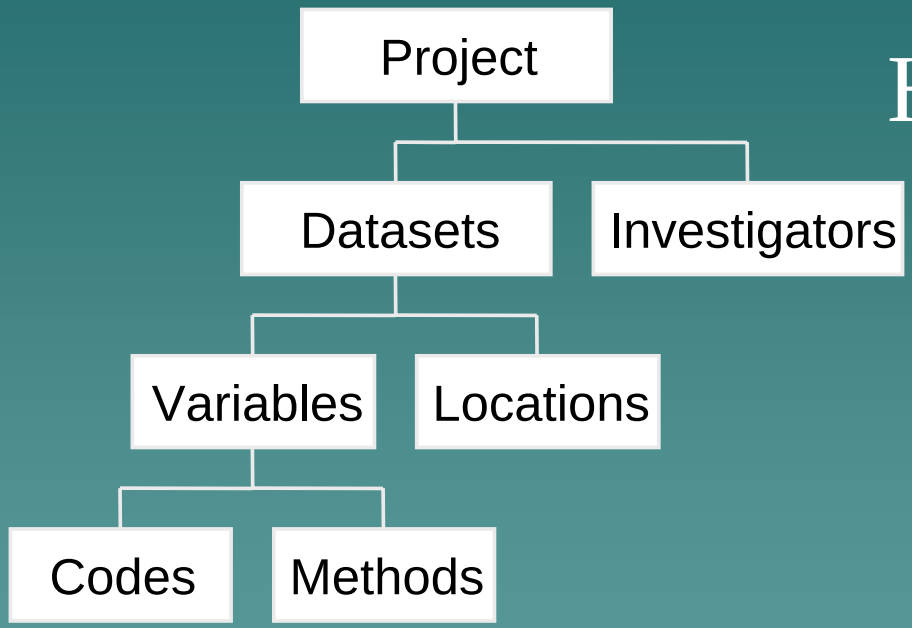
- File system-based
- Hierarchical
- Network
- Relational
- Object-oriented

File-System Based



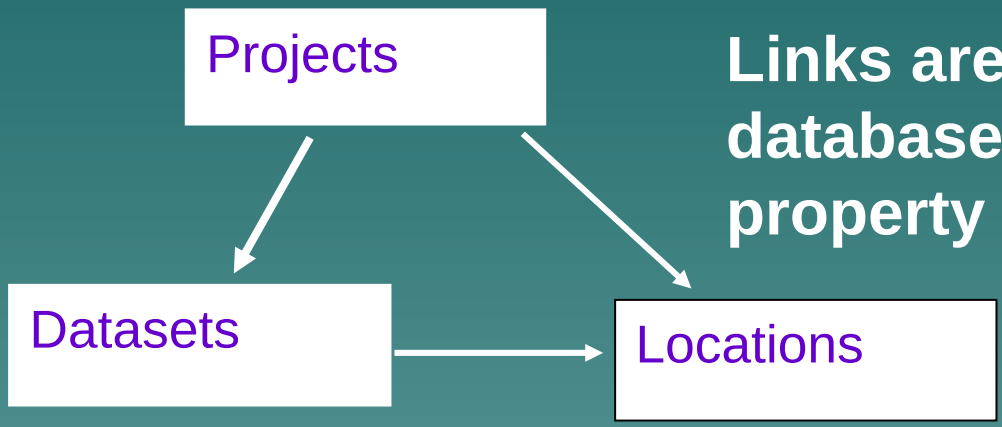
- very simple and easy to set up
- inefficient
- few capabilities

Hierarchical



Hierarchical
- efficient
- not very general
- e.g. phylogenetic structures
- geographical images

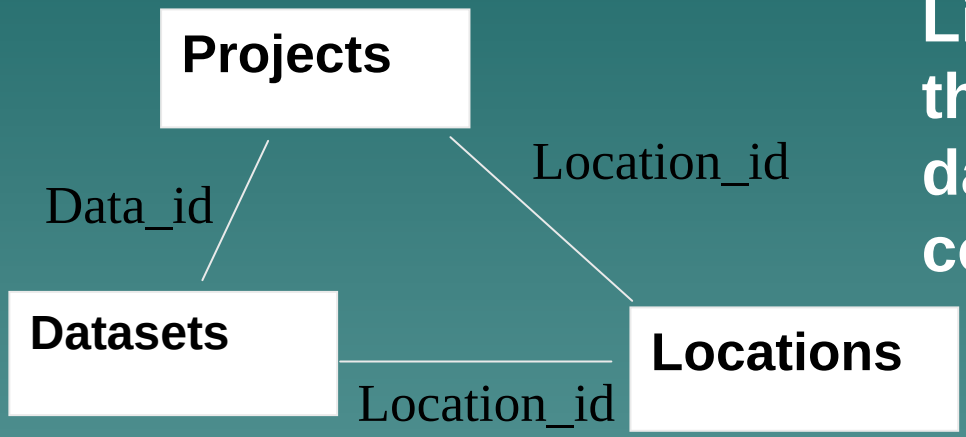
Network Database



Links are hard-coded into database. They are not a property of the data

- very flexible
- unwieldy to modify
- not widely used

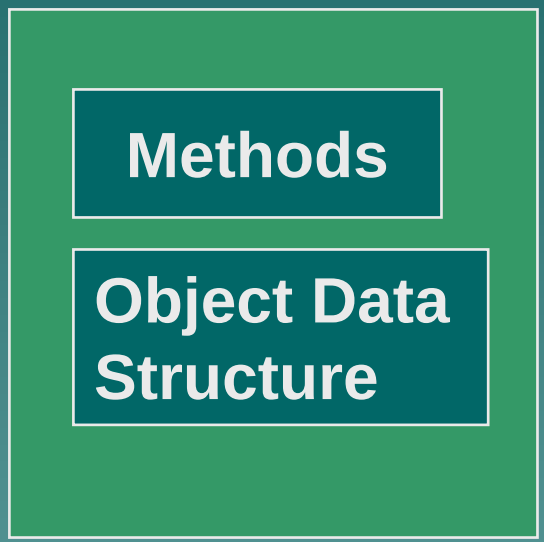
Relational Database



Linkages are through the properties of the data itself - not hard coded

- widely-used, mature
- table-oriented
- restricted range of structures

Object Oriented



- developing -few commercial implementations
- diverse structures
- extensible

Complex data structures, along with the methods to use the data are in the database

Data Modeling

- DBMS Systems are highly flexible
- Good: they can do a lot!
- Bad: they have to be told how to do it!
- A Database Management System is the CANVAS, the DATA MODEL is the painting.....

Data Modeling

- Data modeling is used to develop the database structures used in a database
- Your data model effects
  - reliability of the data
  - efficiency and speed of queries
  - the complexity of the database
- Data modeling is an art, not a science!

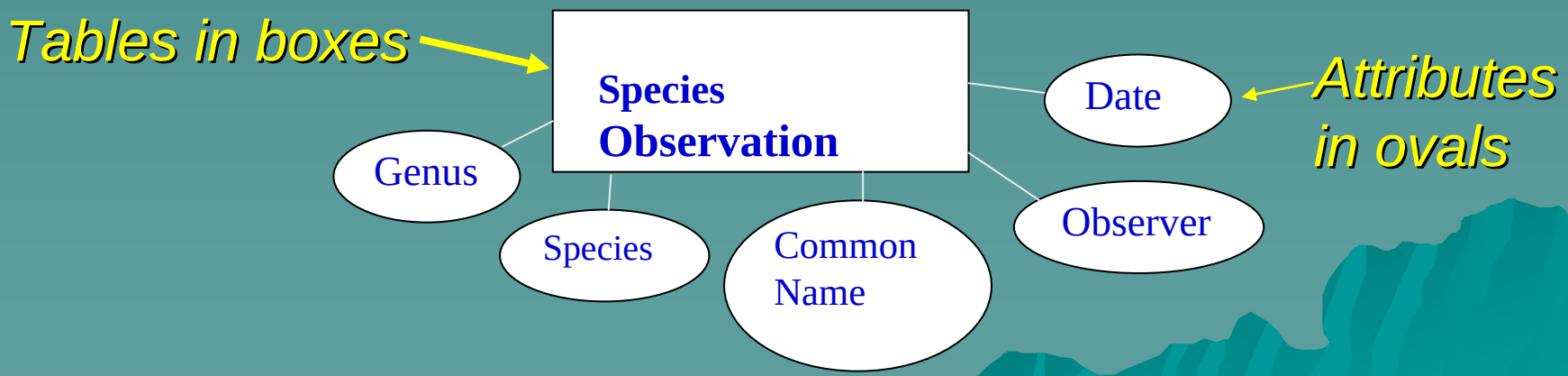
Some Terminology:

Tables contain attributes or fields (columns) and multiple observations or tuples (rows)

Spec_code	Genus	Species	Common Name
QRCALB	Quercus	alba	White Oak
QRCRBR	Quercus	rubra	Red Oak

Flat-file

Genus	Species	Common Name	Observer	Date
Quercus	alba	White Oak	Jones, D.	15-Jun-1998
Quercus	alba	White Oak	Smith, D.	12-Jul-1935
Quercus	alba	White Oat	Doe, J.	15-Sep-1920
Quercus	rubra	Red Oak	Fisher, K.	15-Jun-1998
Quercus	rubra	Red Oak	James, J.	15-Sep-1920



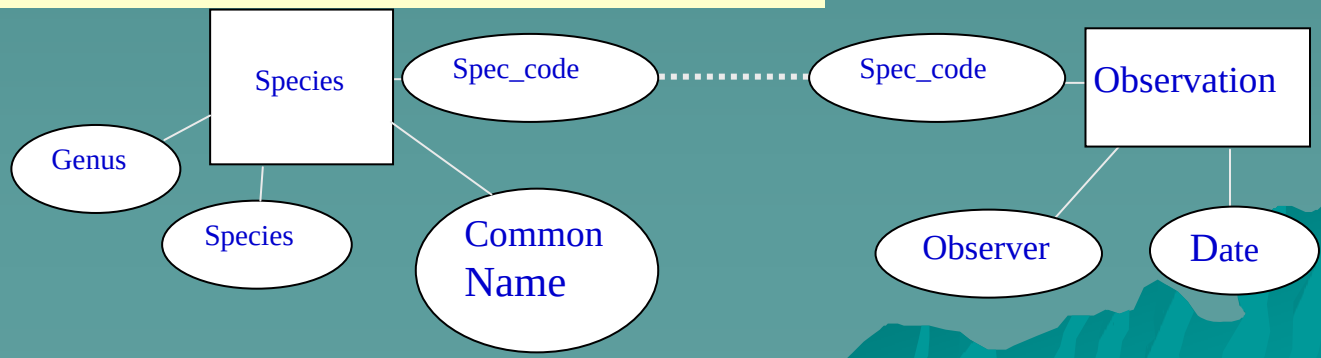
Normalization

- One widely-used approach for reducing errors within a database is to normalize your data structures
- Normalization is the process of eliminating duplicate or redundant information

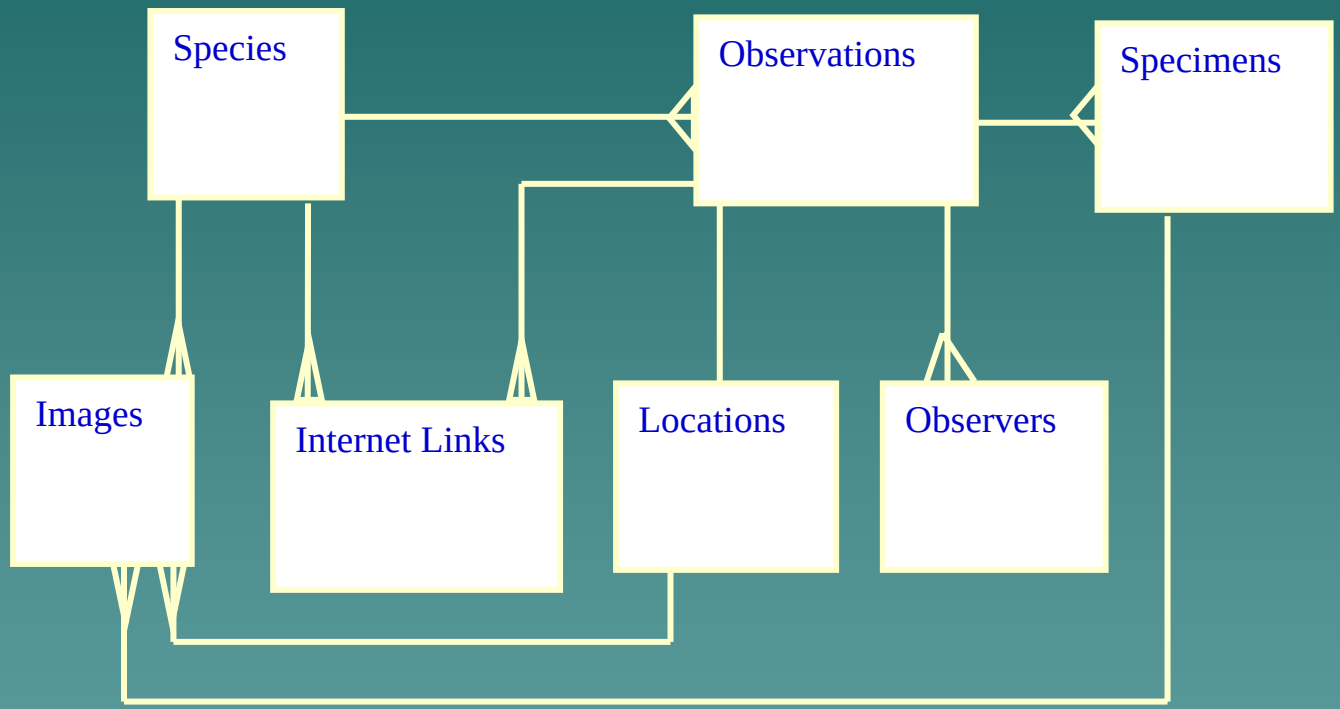
Two-table Relational Database

Spec_code	Genus	Species	Common Name
QRCALB	Quercus	alba	White Oak
QRCRBR	Quercus	rubra	Red Oak

Spec_code	Observer	Date
QRCALB	Jones, D.	15-Jun-1998
QRCALB	Smith, D.	12-Jul-1935
QRCALB	Doe, J.	15-Sep-1920
QRCRBR	Fisher, K.	15-Jun-1998
QRCRBR	James, J.	15-Sep-1920



Complex Data Model



Notation: *One-to-one*

or *One-to-many*

ArcGIS Databases

- The progenitor of ArcGIS, Arc/Info used the “Info” database
  - Info is now obsolete, but is still included in ArcGIS Workstation for compatibility
  - Coverages are stored in Info data formats (hence the ‘info’ directory)
- ArcGIS now uses its own, built-in database based on Dbase IV (.dbf) formats

Using External Databases

- The major reason for using external databases is that there may be large amounts of useful data stored in non-GIS databases
  - Addresses
  - Detailed land cover characteristics
- Often these databases are maintained outside of a GIS using database tools (e.g., UVA Integrated System)
- Access to external databases reduces the need for redundant storage and eliminates issues of concurrency

Databases for ArcGIS

- There are a number of databases that can be used with ArcGIS and the “Spatial Database Engine” (ArcSDE)
  - Oracle – powerful, but expensive
  - Microsoft SQLserver
  - IBM DB2
  - Informix
- Personal Geodatabases can use ACCESS
  - Not recommended for large applications

Structured Query Language

- So far, you've had the theory of databases, but how do you use them?
- The key to most modern databases (but NOT the INFO database) is called "Structured Query Language" or more simply, SQL (pronounced "seequel")

Why use SQL?

- Provides the tools needed to manage relational databases including:
  - Creating Tables
  - Adding Data
  - Queries / Searches
- It's a STANDARD! – multiple vendors produce products that support SQL queries

Standards – A Caveat

- Just because there are standards for SQL implementations does not mean that all databases will have all the capabilities in the SQL standard.
- Most relational databases implement some non-standard extensions or lack some features of the full standard

Examples of Variation

- MiniSQL - implements only a critical subset of SQL commands
- MySQL – fairly compatible - no sub-selects (nested selects)
- Postgres – not fully standardized, object extensions
- **“The wonderful thing about standards is that there are so many of them to choose from” - anonymous**

Critical SQL Commands

- Table Management
  - Create Table
  - Drop Table
- Editing
  - Insert
  - Update
  - Delete
- Query
  - Select

There are many other commands, but these six will allow you to do almost anything you need to do

Create Table

```
CREATE TABLE mytable (
name CHAR(40) NOT NULL,
age  INT )
```

- Creates a table named “mytable” with two fields
  - A required character field called “name”
  - An optional numeric (integer) field called “age”

Insert

```
INSERT INTO mytable
  (name,age)
VALUES ('George',20)
```

- Inserts a data row into the table
  - "name" is set to "George"
  - "age" is set to 20

Select

```
SELECT name,age FROM mytable WHERE
age = 20
```

- Searches the table for rows where “age” is 20 and returns the associated name and age. This query resulted in:

```
+--------+------+
| name   | age  |
+--------+------+
| George |   20 |
+--------+------+
1 row in set (0.02 sec)
```

Update

```
UPDATE mytable SET age=21
  WHERE name LIKE 'George'
```

- Searches the table for rows where "name" is "George" and sets age to 21. Note: if we had more than one row with name "George" all would be set to age=21.

```
+--------+------+
| name   | age  |
+--------+------+
| George |   21 |
+--------+------+
1 row in set (0.02 sec)
```

Delete (a row from a table)

```
DELETE FROM mytable WHERE
  name LIKE ‘George’AND age =
  21
```

- Searches the table for rows where “name” is “George” and age is 21 and deletes them

Drop Table (delete a table)

`DROP TABLE mytable`

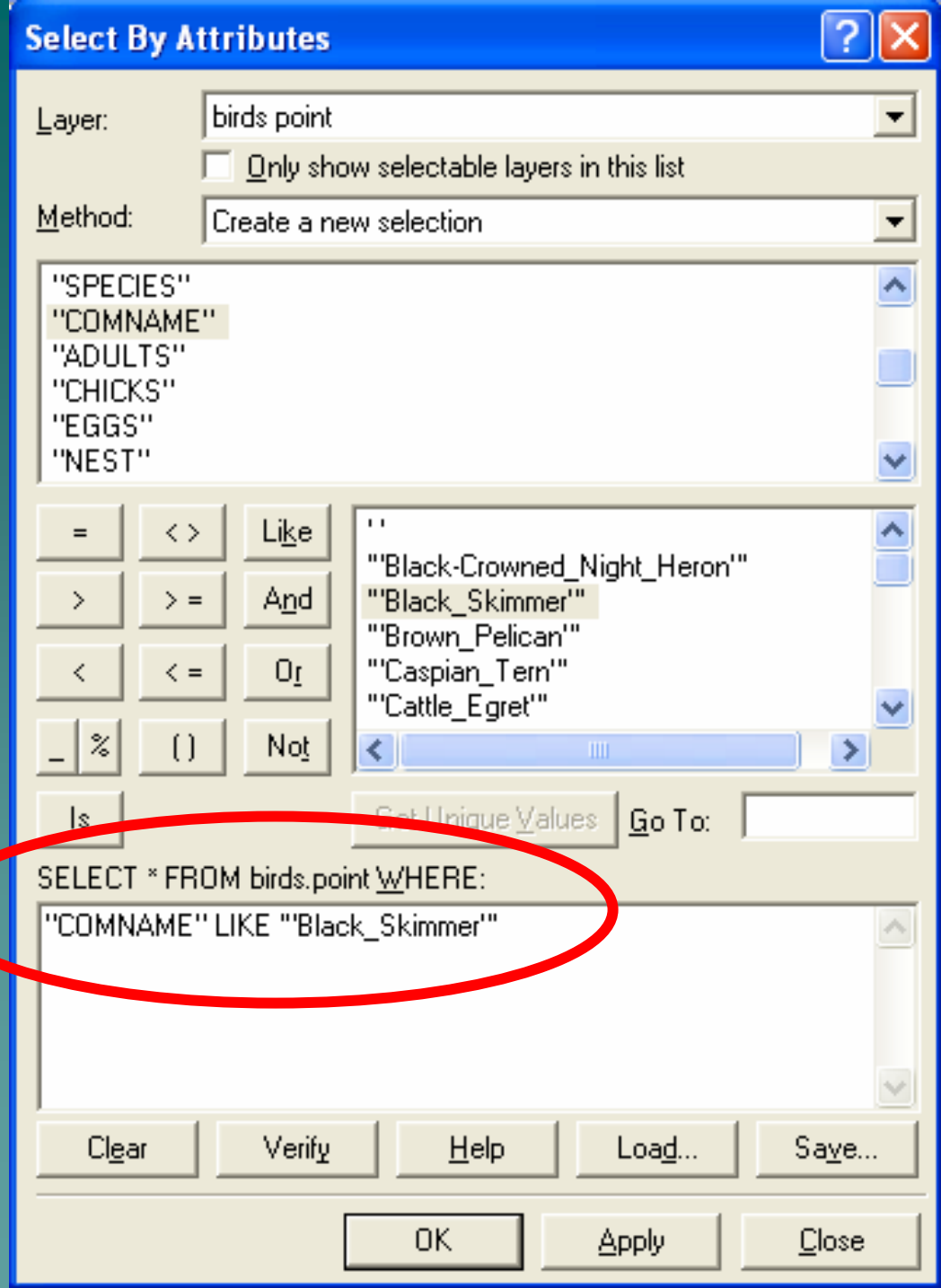
- Completely eliminates table “mytable.” All data in the table is lost.

Why SQL?

- Despite its power to manipulate data, SQL makes a poor user interface
  - Few ecologists will want to take the time to learn SQL
  - Effective use also requires knowledge of the underlying fields and tables
- For this reason, most SQL is imbedded into programs where it is hidden from the users

SQL in ArcGIS

- SQL syntax is used in several of the ArcGIS tools
  - SELECT BY ATTRIBUTE generates an SQL query



Other SQL functions

- ArcGIS won't let you do updates etc. using SQL statements
  - That could damage what ArcGIS needs to do it's work
  - However, you can modify the data using the attribute CALCULATOR

Putting the Relations in Relational Databases

- SELECT statements are not restricted to single tables. For example:

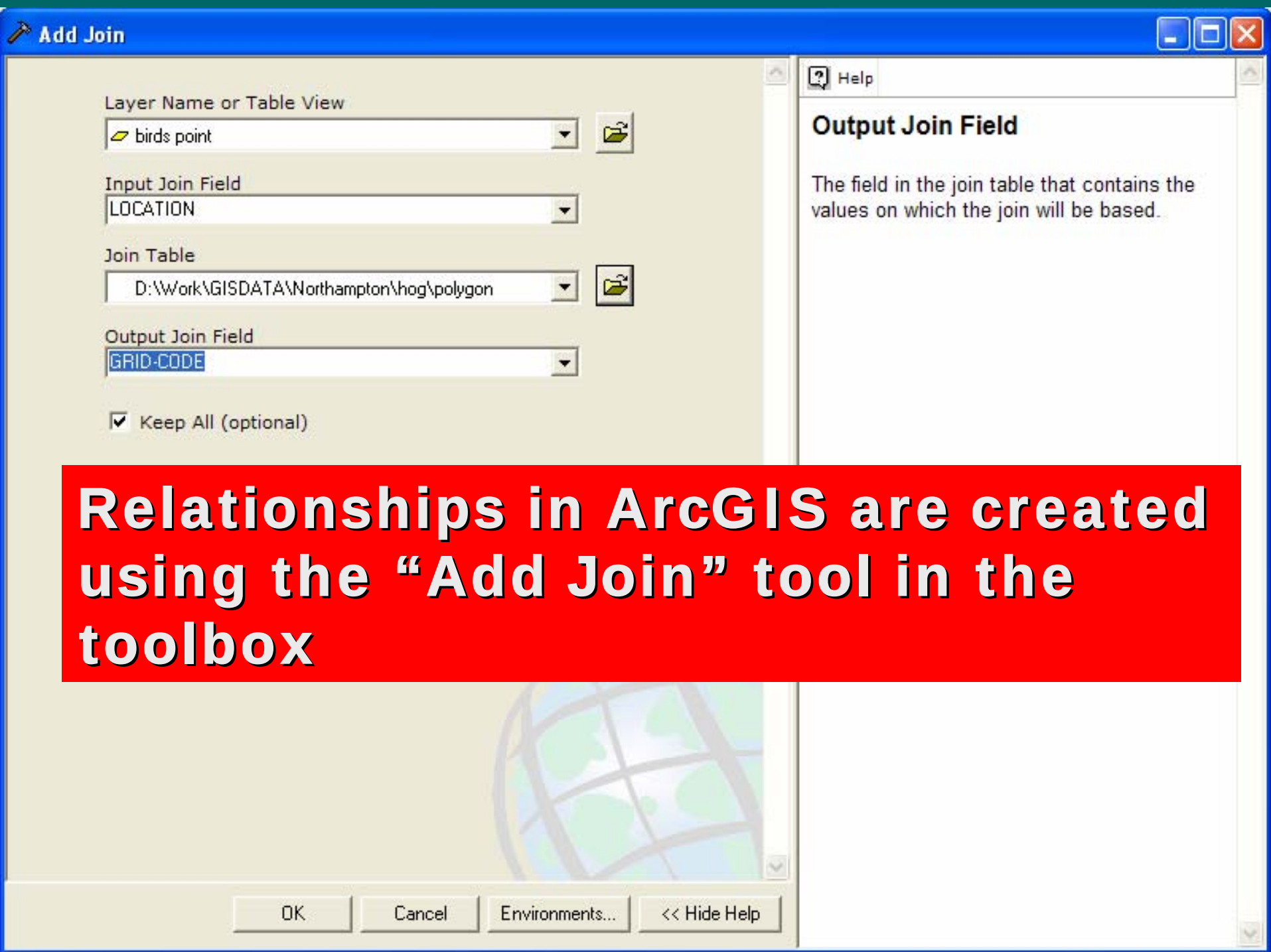
```
SELECT DISTINCT
mytable.age, yourtable.address
FROM mytable, yourtable
WHERE mytable.name LIKE
  yourtable.name
```

Multi-table selects create a “join”

Relational SELECT

```
SELECT DISTINCT
mytable.age, yourtable.address
FROM mytable, yourtable
WHERE mytable.name LIKE yourtable.name
```

- Accesses two different tables: “mytable” and “yourtable”
- Returns “age” from mytable, and “address” from yourtable where the “name” field in the two tables match.
- DISTINCT means that if the same age and address shows up in multiple rows, only the first instance will be displayed.



Relationships in ArcGIS are created using the "Add Join" tool in the toolbox

SQL Exercise

- Now it's time for you to try out your SQL skills using the web pages:

http://www.sqlcourse.com

http://www.sqlcourse2.com

  - Do part 10 (table joins)