Attribute Data and Tables

Topics

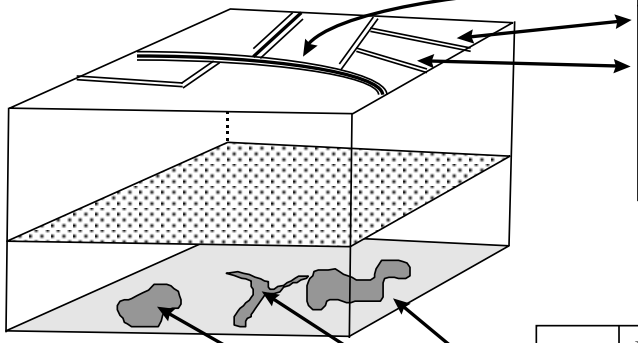
- Databases and database management system
- SQL (Structured Query Language)
- Join and relate

GIS Databases

- GIS databases store attributes and locations

Location

Attributes



ID	Age	Type	Pavement
1	12	4Ln	Bitumen
2	2	2Ln	Concrete
3	95	1Ln	Dirt
etc.			

ID	Lake Size	Quality
99	93.2	4
100	65.7	3
122	55.3	0
etc.		

Database and DBMS

- Databases store data
- **D**ata**b**ase **M**anagement **S**ystem (DBMS)
  - Software just like ArcGIS
  - Manage primarily attribute data
  - Commercial DBMS
    - Oracle, Microsoft Access, SQL Server, DB2
- What can DBMS do?
  - Store data and relationships among data (modeling and storage)
  - Provide fast and efficient access to large amounts of data (retrieval/ query)
  - Update, modify and transform data (manipulation)
  - Support multiple users with different levels of access privileges (access control)

Relational Databases

- Flat files and early DBMS
  - Flat files (redundancy & relationships)
  - Hierarchical databases
    - XML …
  - Network databases
- Relational DBMS (RDBMS)
  - Both data and relationships are stored as tables (everything is a table)
  - Built on solid mathematical foundation (relational algebra)
  - Easy to query
  - Relatively easy to create and maintain

Flat File

Data in a "text" or other lightly formatted file.

Little structure, cross-referencing or linking among entries.

Often in a row/column format

Advantages:
Transparent, easily transportable

Disadvantages:
Little structure, few error safeguards

Forests

Forest Name	Location	Size
Nantahala	North Carolina	184,447
Cherokee	North Carolina	92,271

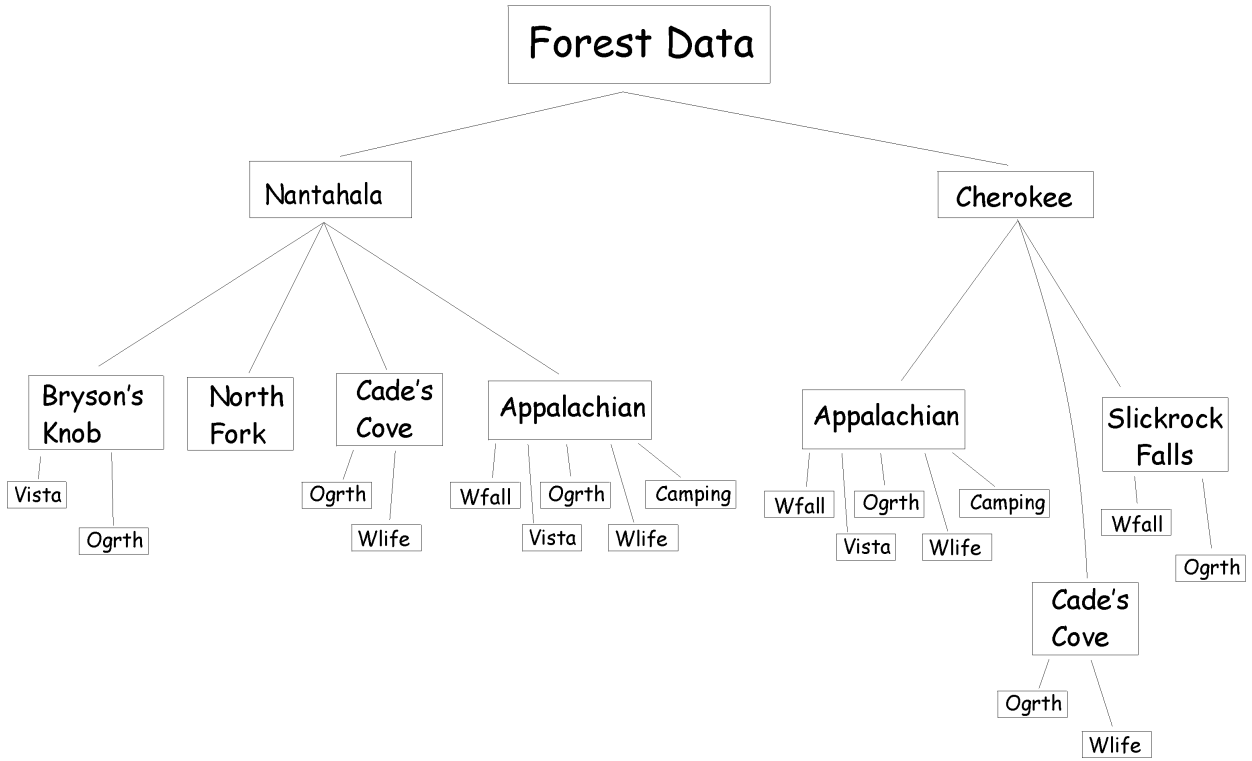
Trails

Trail Name	Difficulty	Forest	Feature
Bryson's Knob	E, M	Nantahala	Vista, Ogrth
Slickrock Falls	M	Cherokee	Wfall, Ogrth
North Fork	M	Nantahala	-
Cade's Cove	E	Cherokee, Nantahala	Ogrth, Wlife
Appalachian	M, D	Nantahala, Cherokee	Wfall, Ogrth, Vista, Wlife, Cmp

Recreation Features

Feature	Description	Activities
Wfall	Waterfall	Photography, Swimming
Ogrth	Old-Growth Forest	Photography, Hiking
Vista	Scenic overlook	Photography, viewing
Wlife	Wildlife Viewing	Photography, Birding
Cmp	Camping	Camping

Hierarchical Database



Data organized with parent-child connections in a tree-like structure

Branches group successively more similar data

Advantages:
Logical structure, quick searches for related items

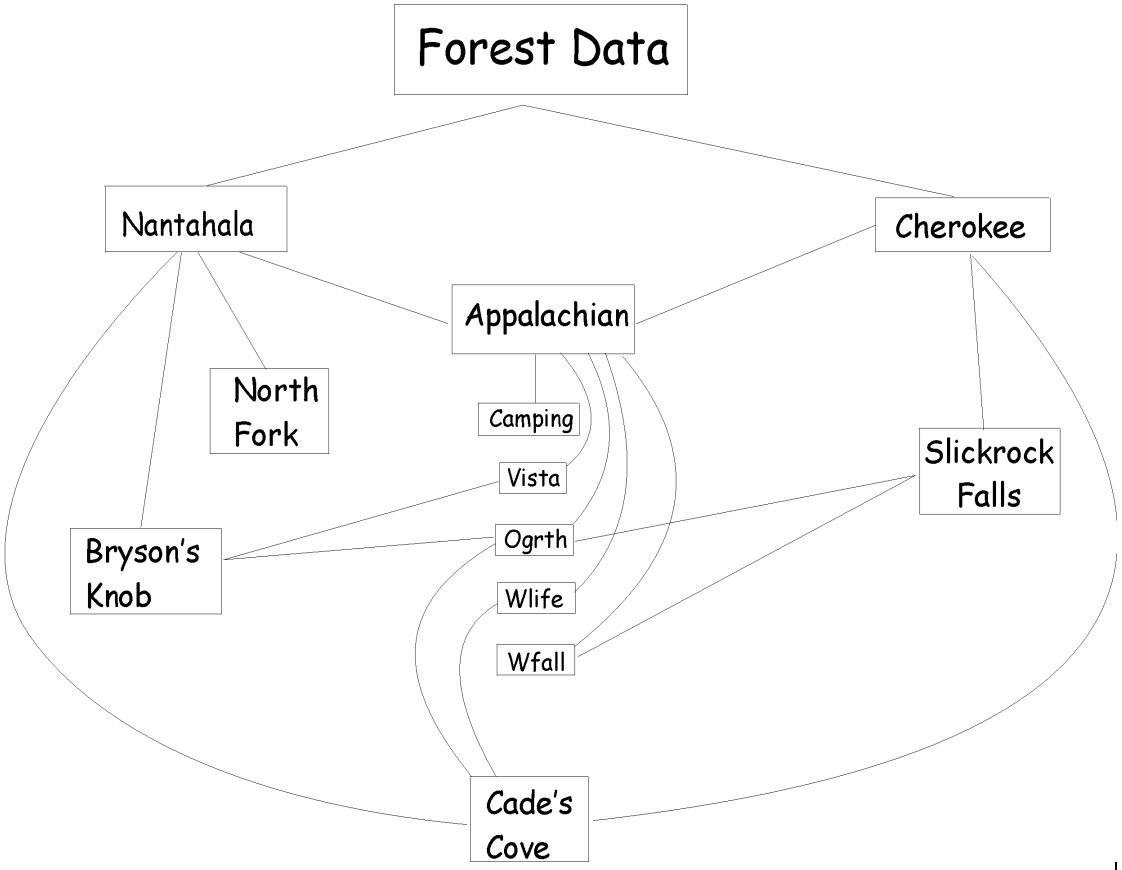
Disadvantages:
Significant effort required to create the tree structure.
Slow searches across branches

Data elements connected in a cross-linked structure

Advantages: Quick searches, reduced (often no) duplication.

Disadvantages: Significantly complex structuring – maintenance is difficult

Network Database



Relational Database

Minimal row-column structure

Items/records with specified domains (possible values)

Advantages:
Minimum structure, easy programming, flexible

Forests

Forest Name	Forest-ID	Location	Size
Nantahala	1	N. Carolina	184,447
Cherokee	2	N. Carolina	92,271

Trails

Trail Name	Forest-ID
Bryson's Knob	1
Slickrock Falls	2
North Fork	1
Cade's Cove	1
Cade's Cove	2
Appalachian	1
Appalachian	2

Characteristics

Trail Name	Feature	Difficulty
Bryson's Knob	Vista	E,M
Bryson's Knob	Ogrth	E,M
Slickrock Falls	Ogrth	M
Slickrock Falls	Wfall	M
North Fork	-	M
Cade's Cove	Ogrth	E
Cade's Cove	Wlife	E
Appalachian	Wfall	M,D
Appalachian	Ogrth	M,D
Appalachian	Vista	M,D
Appalachian	Wlife	M,D
Appalachian	Cmp	M,D

Recreational features

Feature	Description	Activity1	Activity2
Wfall	Waterfall	Photography	Swimming
Ogrth	Old-Growth Forest	Photography	Hiking
Vista	Scenic Overlook	Photography	Viewing
Wlife	Wildlife Viewing	Photography	Birding
Cmp	Camping	Camping	-

Disadvantages:
Relatively slow, a few restrictions on attribute content

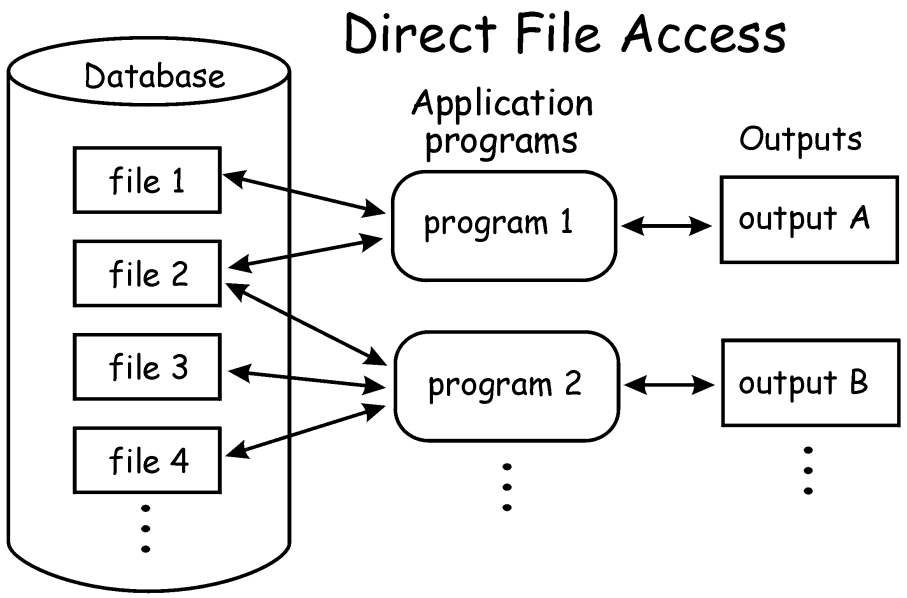
Flat File vs. Database

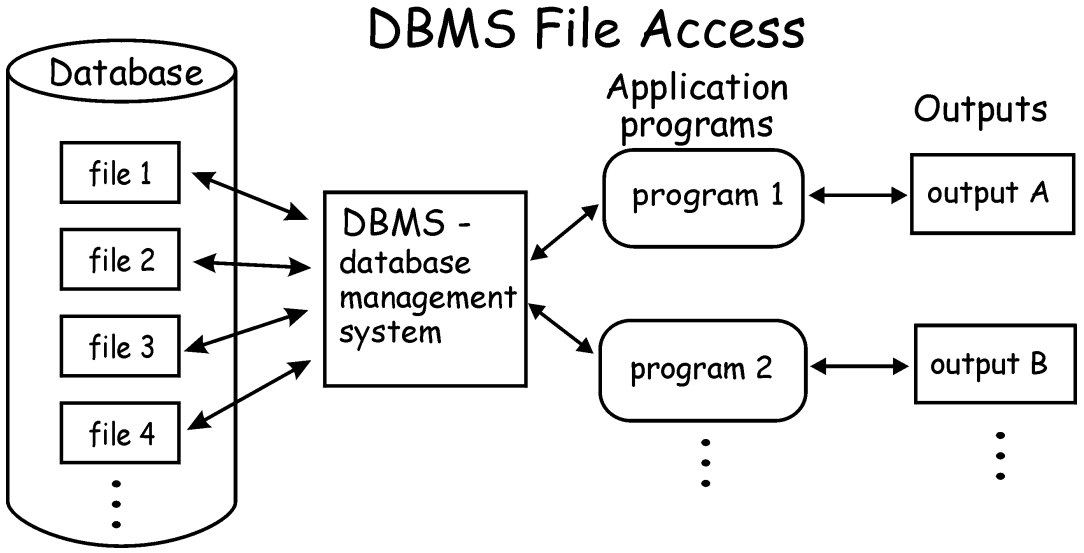
Structure among data

Applications independent of database

Multiple users (simultaneous updates)

Data security (different views)





Tables

- Data in a database are organized as tables
- Each table have a **unique name**
- A database typically has many tables

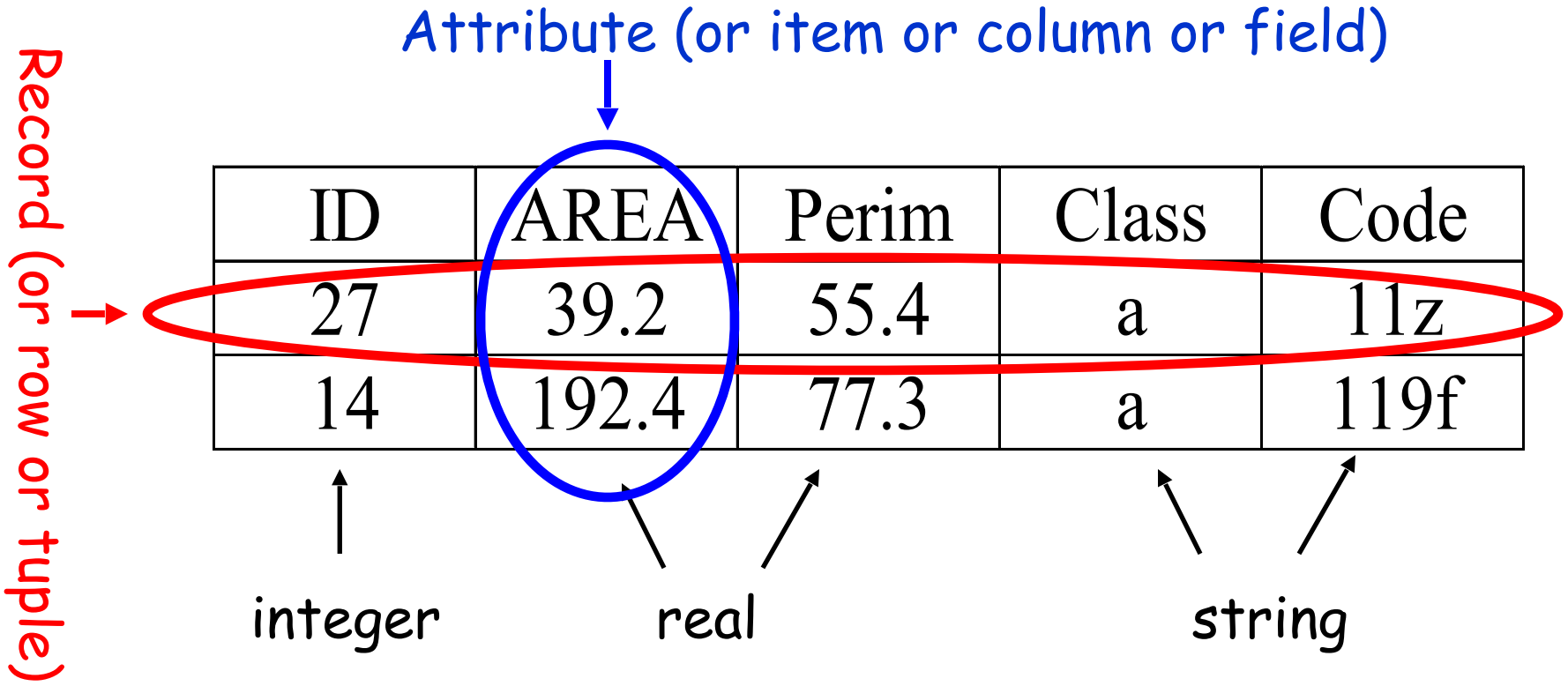
ID	Name	Major
100	Sam	Geography
200	Kevin	Economics
300	Steve	Computer Science

Students

ID	Name	Department
10000	Introductory GIS	Geography
20000	**Macroeconomics**	Economics
30000	Programming Language	Computer Science

Classes

Table Elements and Terms



A table (or *relation*) is a **set** of **tuples**

A field has a type and domain that restrict the values it may have

Main Functions of RDBMS

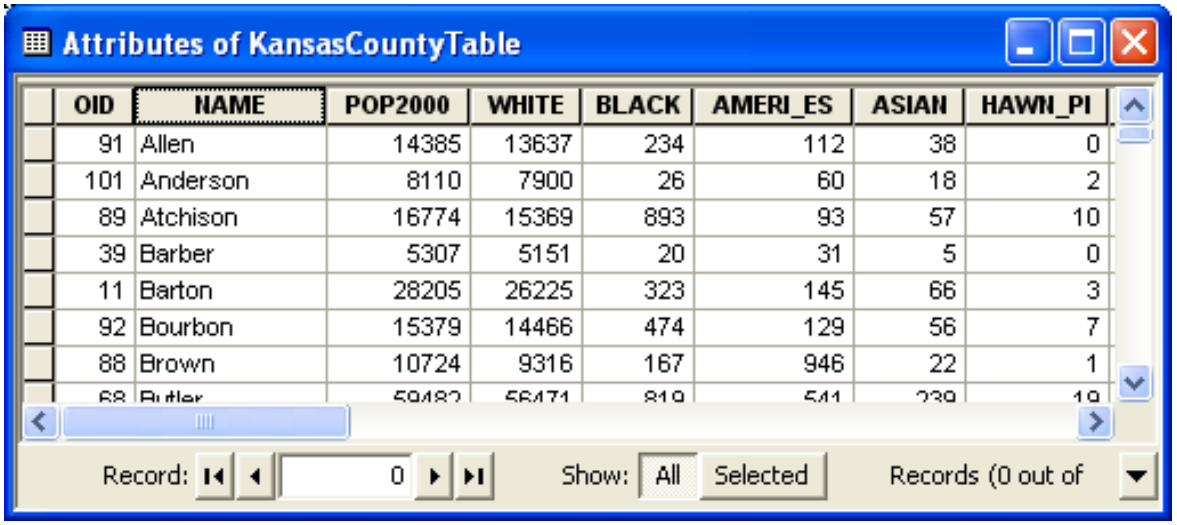
- Query / Selection
  - Choosing a subset of records and attributes which meet certain conditions
- Join
  - associate data in two tables
- Sort and calculate statistics

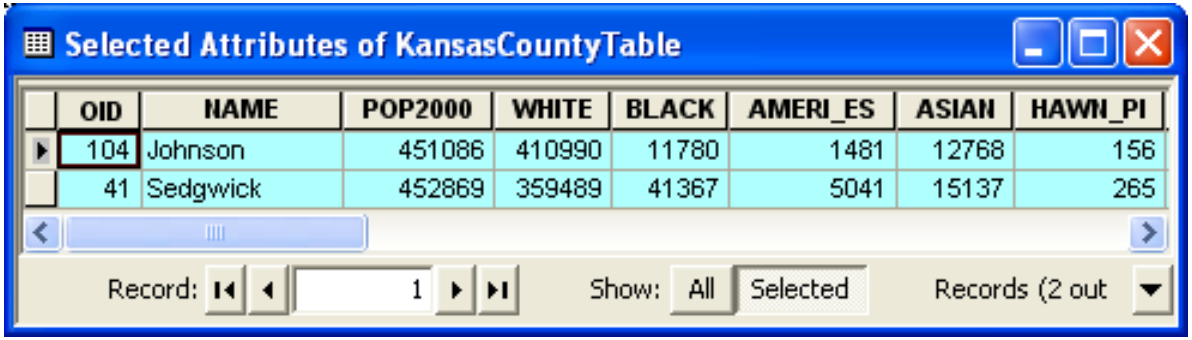
Structured Query Language (SQL)

- A language used to retrieve data from relational databases
- Most databases understand standard SQL
- SQL queries generally return results as a table, just like the tables that store the original data

The SELECT Command in SQL

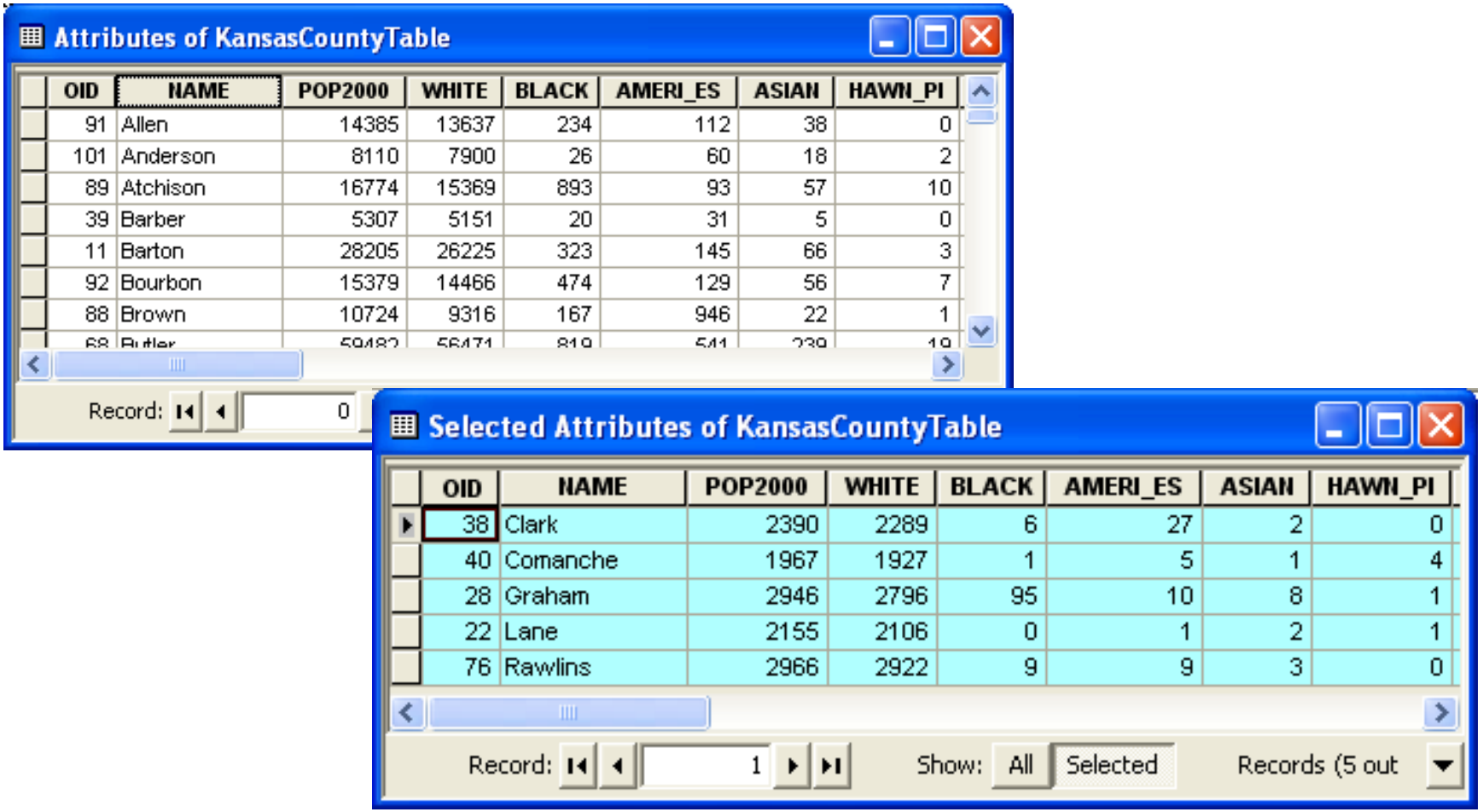
- SELECT *column(s)* FROM *table(s)* WHERE conditions
- SELECT * FROM KansasCountyTable WHERE POP2000 > 200000





The WHERE Clause

- SELECT * FROM KansasCountyTable WHERE POP2000 < 3000 AND AGE_65_UP > AGE_5_17



Attributes of KansasCountyTable

OID	NAME	POP2000	WHITE	BLACK	AMERI_ES	ASIAN	HAWN_PI
91	Allen	14385	13637	234	112	38	0
101	Anderson	8110	7900	26	60	18	2
89	Atchison	16774	15369	893	93	57	10
39	Barber	5307	5151	20	31	5	0
11	Barton	28205	26225	323	145	66	3
92	Bourbon	15379	14466	474	129	56	7
88	Brown	10724	9316	167	946	22	1
68	Butler	59482	56471	819	541	239	19

Record: 0

Selected Attributes of KansasCountyTable

OID	NAME	POP2000	WHITE	BLACK	AMERI_ES	ASIAN	HAWN_PI
38	Clark	2390	2289	6	27	2	0
40	Comanche	1967	1927	1	5	1	4
28	Graham	2946	2796	95	10	8	1
22	Lane	2155	2106	0	1	2	1
76	Rawlins	2966	2922	9	9	3	0

Record: 1 Show: All Selected Records (5 out

Expressions

- Conditions are formed as expressions in the WHERE clause in SQL
  - WHERE POP2000 < 3000 AND AGE_65_UP > AGE_5_17
- A simple expression consists of two operands and one logical operator
    - [Operand] [Logical Operator] [Operand]
- An operand can be
  - A field/column
  - A constant (number or string)
- A logical operator can be any of the followings:
  - =, >, <, <>, >=, <=
  - Like (for using wildcards (%, *, ?) in strings)

Simple Expression

- Where Area > 20.0

Area > 20.0

ID	Area	Landuse	Municip
1	10.5	Urban	City
2	330.3	Farm	County
3	2.4	Suburban	Township
4	96.0	Suburban	County
5	22.1	Urban	City
6	30.2	Farm	Township
7	4.4	Urban	County

Complex Expressions

- A complex expression is a set of expressions connected by Boolean connectors
- Boolean connectors are
  - AND, OR (binary connector)
  - NOT (unitary connector)
- Find the counties whose 2000 population is greater than 30,000 and median house income is less than 25,000
  - WHERE (pop2000 > 30000) AND (medhinc < 25000)
  - ‘pop2000’ and ‘medhinc’ are fields in the population table

Boolean Connectors—AND, OR

- Where landuse = ‘urban’ AND Municip = ‘City’
- Where landuse = ‘urban’ OR Municip = ‘City’

ID	Area	Landuse	Municip
1	10.5	Urban	City
2	330.3	Farm	County
3	2.4	Suburban	Township
4	96.0	Suburban	County
5	22.1	Urban	City
6	30.2	Farm	Township
7	4.4	Urban	County

AND typically decreases the number of records selected
OR typically increases the number of records selected

Boolean Connectors--NOT

- WHERE NOT (landuse = ‘urban’)
- WHERE landuse <> ‘urban’

ID	Area	Landuse	Municip
1	10.5	Urban	City
2	330.3	Farm	County
3	2.4	Suburban	Township
4	96.0	Suburban	County
5	22.1	Urban	City
6	30.2	Farm	Township
7	4.4	Urban	County

Complex Expression

[(Landuse = Urban) AND (Mill Rate = B)] OR
{NOT(Municip = City) AND (Density > 200)}

ID	Area	Landuse	Municip	Density	Mill Rate
1	10.5	Urban	City	1,112.2	A
2	330.3	Farm	County	1.9	C
3	2.4	Suburban	Township	237.5	C
4	96.0	Suburban	County	98.1	A
5	22.1	Urban	City	916.2	B
6	30.2	Farm	Township	3.7	A
7	4.4	Urban	County	153.8	D

Connector's Order Is Important

- NOT (A and B) is not the same as NOT (A) AND NOT (B)

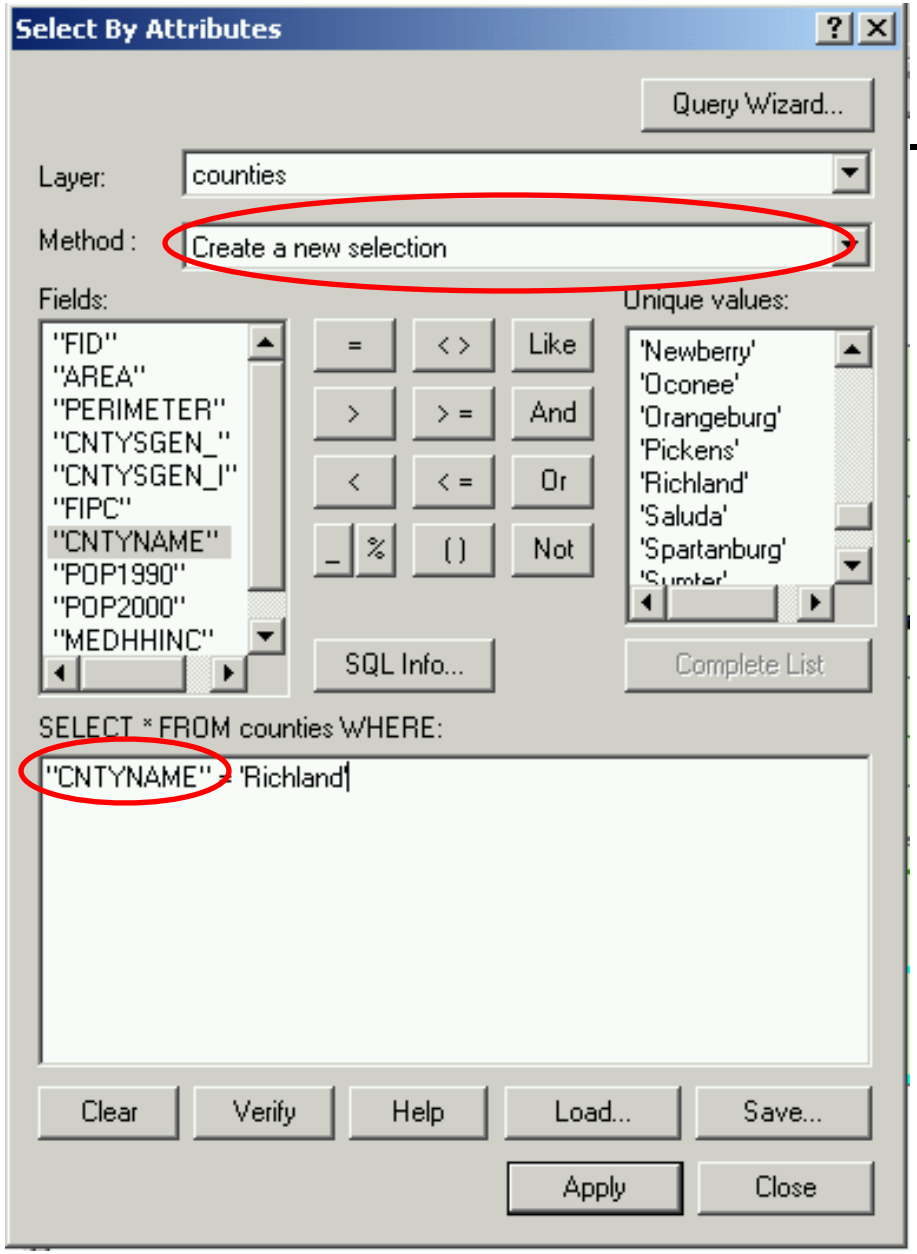
NOT [(Landuse = Urban) AND
(Municip = County)]

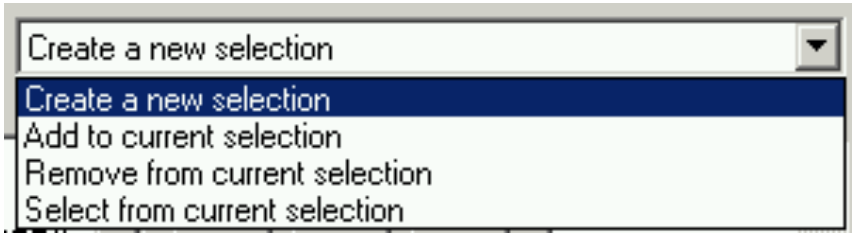
ID	Area	Landuse	Municip
1	10.5	Urban	City
2	330.3	Farm	County
3	2.4	Suburban	Township
4	96.0	Suburban	County
5	22.1	Urban	City
6	30.2	Farm	Township
7	4.4	Urban	County

[NOT (Landuse = Urban)] AND
[NOT (Municip = County)]

ID	Area	Landuse	Municip
1	10.5	Urban	City
2	330.3	Farm	County
3	2.4	Suburban	Township
4	96.0	Suburban	County
5	22.1	Urban	City
6	30.2	Farm	Township
7	4.4	Urban	County

Query in ArcGIS





Two sets are associated with a table.
- Whole set – all the features in the table.
- Selection set—the set of selected features from previous queries

Queries can be performed from
- Whole set (all the features in the table)
- Selection set (features in the selection set)

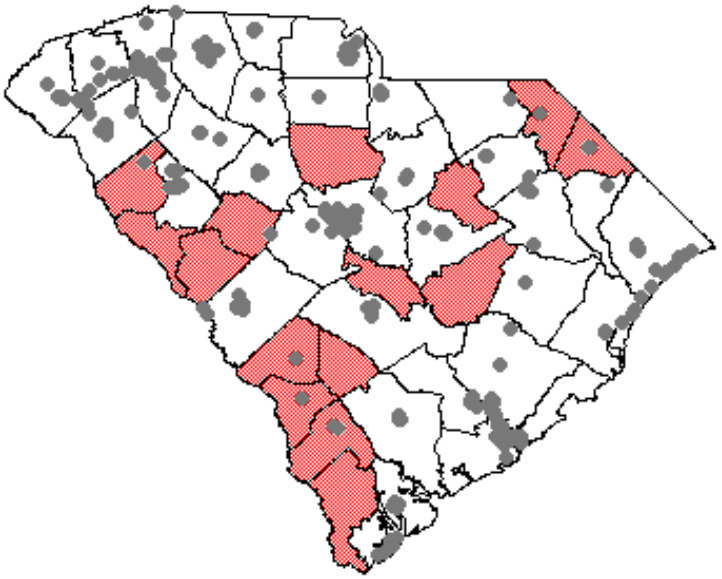
Query results can be
- Appended to the selection set
- Removed from the selection set
- A new selection set (previous selection set is discarded)

Query in GIS

- Attribute queries highlight the selected records instead of creating a new table
- Corresponding features are also highlighted on maps.
- Other types of queries:
  - Spatial queries and compound queries

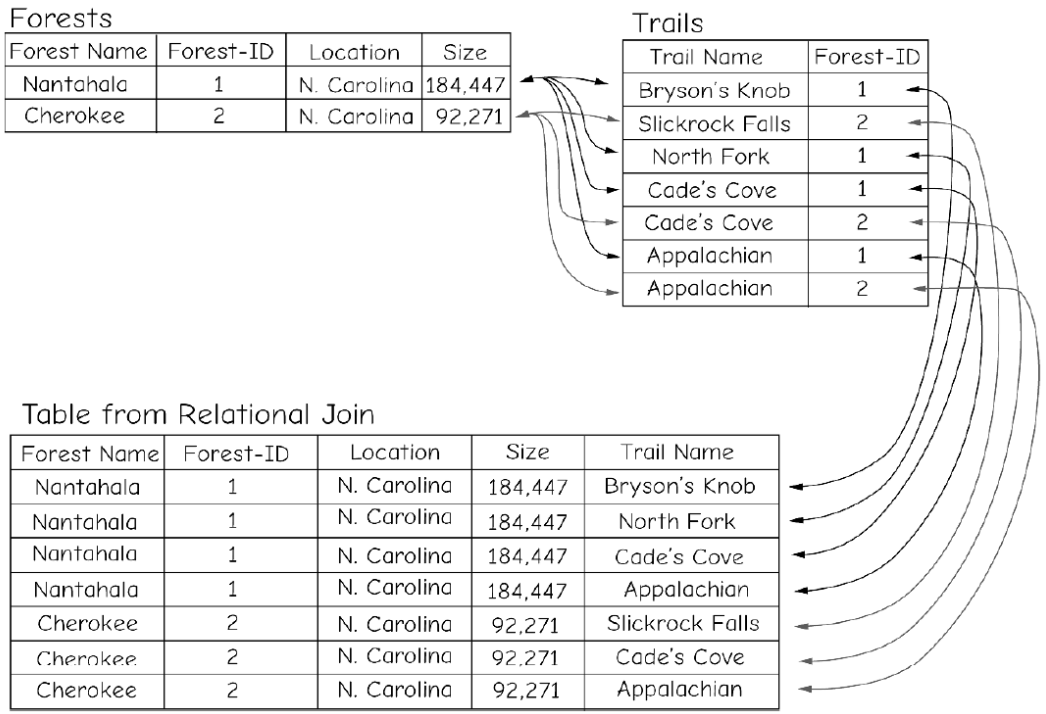
Attributes of Cntysgen

Shape	Fipc	Cntyname	Pop1990	Pop2000	Medhhi
Polygon	69	Marlboro	29361	30279	1806
Polygon	55	Kershaw	43599	49946	2828
Polygon	39	Fairfield	22295	22688	2148
Polygon	31	Darlington	61851	68346	2264
Polygon	1	Abbeville	23862	24611	2317
Polygon	47	Greenwood	59567	63971	2358
Polygon	61	Lee	18437	19009	1817
Polygon	41	Florence	114344	130492	2426
Polygon	67	Marion	33899	37001	1782
Polygon	79	Richland	285720	315896	2884
Polygon	63	Lexington	167611	215232	3291
Polygon	85	Sumter	102637	113882	2238



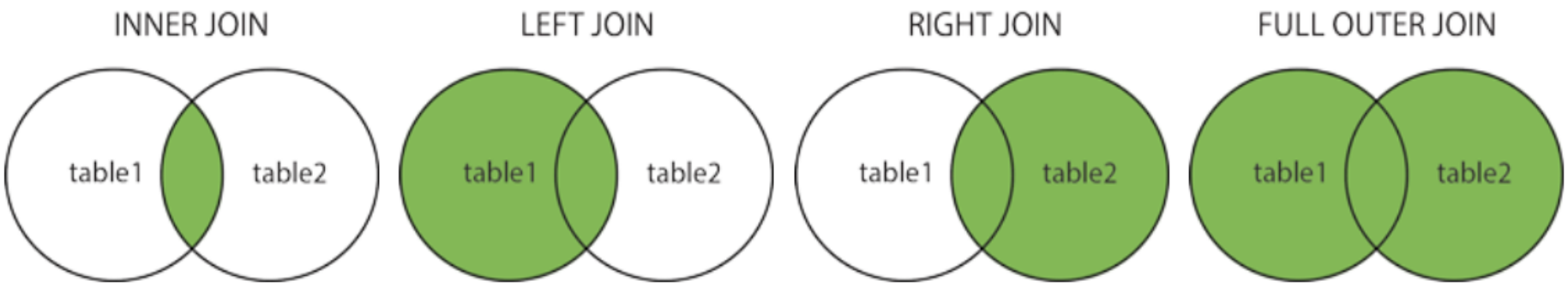
Join Tables

- Relational databases organize data in ways that reduce duplication
- Select and combine rows from two tables based on *a common field* as long as *a certain condition* on the common field satisfies



Join Tables

- (INNER) JOIN: Returns rows that have matching values in both tables
- LEFT (OUTER) JOIN: Returns all rows from the left table, and the matched rows from the right table
- RIGHT (OUTER) JOIN: Returns all rows from the right table, and the matched rows from the left table
- FULL (OUTER) JOIN: Returns all rows when there is a match in either left or right table



left_join()

right_join()

inner_join()

full_join()

Inner Join

```
SELECT  ID, NAME, AMOUNT, DATE
  FROM CUSTOMERS
  JOIN ORDERS
  ON CUSTOMERS.ID = ORDERS.CUSTOMER_ID;
```

Table 1 − CUSTOMERS Table is as follows.

ID	NAME	AGE	ADDRESS	SALARY
1	Ramesh	32	Ahmedabad	2000.00
2	Khilan	25	Delhi	1500.00
3	kaushik	23	Kota	2000.00
4	Chaitali	25	Mumbai	6500.00
5	Hardik	27	Bhopal	8500.00
6	Komal	22	MP	4500.00
7	Muffy	24	Indore	10000.00

Table 2 − ORDERS Table is as follows.

OID	DATE	CUSTOMER_ID	AMOUNT
102	2009-10-08 00:00:00	3	3000
100	2009-10-08 00:00:00	3	1500
101	2009-11-20 00:00:00	2	1560
103	2008-05-20 00:00:00	4	2060

ID	NAME	AMOUNT	DATE
3	kaushik	3000	2009-10-08 00:00:00
3	kaushik	1500	2009-10-08 00:00:00
2	Khilan	1560	2009-11-20 00:00:00
4	Chaitali	2060	2008-05-20 00:00:00

Left Join

```
SELECT  ID, NAME, AMOUNT, DATE
   FROM CUSTOMERS
   LEFT JOIN ORDERS
   ON CUSTOMERS.ID = ORDERS.CUSTOMER_ID;
```

Table 1 − CUSTOMERS Table is as follows.

ID	NAME	AGE	ADDRESS	SALARY
1	Ramesh	32	Ahmedabad	2000.00
2	Khilan	25	Delhi	1500.00
3	kaushik	23	Kota	2000.00
4	Chaitali	25	Mumbai	6500.00
5	Hardik	27	Bhopal	8500.00
6	Komal	22	MP	4500.00
7	Muffy	24	Indore	10000.00

Table 2 − ORDERS Table is as follows.

OID	DATE	CUSTOMER_ID	AMOUNT
102	2009-10-08 00:00:00	3	3000
100	2009-10-08 00:00:00	3	1500
101	2009-11-20 00:00:00	2	1560
103	2008-05-20 00:00:00	4	2060

ID	NAME	AMOUNT	DATE
1	Ramesh	NULL	NULL
2	Khilan	1560	2009-11-20 00:00:00
3	kaushik	3000	2009-10-08 00:00:00
3	kaushik	1500	2009-10-08 00:00:00
4	Chaitali	2060	2008-05-20 00:00:00
5	Hardik	NULL	NULL
6	Komal	NULL	NULL
7	Muffy	NULL	NULL

Right Join

```
SELECT  ID, NAME, AMOUNT, DATE
  FROM CUSTOMERS
  RIGHT JOIN ORDERS
  ON CUSTOMERS.ID = ORDERS.CUSTOMER_ID;
```

Table 1 − CUSTOMERS Table is as follows.

ID	NAME	AGE	ADDRESS	SALARY
1	Ramesh	32	Ahmedabad	2000.00
2	Khilan	25	Delhi	1500.00
3	kaushik	23	Kota	2000.00
4	Chaitali	25	Mumbai	6500.00
5	Hardik	27	Bhopal	8500.00
6	Komal	22	MP	4500.00
7	Muffy	24	Indore	10000.00

Table 2 − ORDERS Table is as follows.

OID	DATE	CUSTOMER_ID	AMOUNT
102	2009-10-08 00:00:00	3	3000
100	2009-10-08 00:00:00	3	1500
101	2009-11-20 00:00:00	2	1560
103	2008-05-20 00:00:00	4	2060

ID	NAME	AMOUNT	DATE
3	kaushik	3000	2009-10-08 00:00:00
3	kaushik	1500	2009-10-08 00:00:00
2	Khilan	1560	2009-11-20 00:00:00
4	Chaitali	2060	2008-05-20 00:00:00

Full Join

```
SELECT  ID, NAME, AMOUNT, DATE
   FROM CUSTOMERS
   FULL JOIN ORDERS
   ON CUSTOMERS.ID = ORDERS.CUSTOMER_ID;
```

Table 1 − CUSTOMERS Table is as follows.

ID	NAME	AGE	ADDRESS	SALARY
1	Ramesh	32	Ahmedabad	2000.00
2	Khilan	25	Delhi	1500.00
3	kaushik	23	Kota	2000.00
4	Chaitali	25	Mumbai	6500.00
5	Hardik	27	Bhopal	8500.00
6	Komal	22	MP	4500.00
7	Muffy	24	Indore	10000.00

Table 2 − ORDERS Table is as follows.

OID	DATE	CUSTOMER_ID	AMOUNT
102	2009-10-08 00:00:00	3	3000
100	2009-10-08 00:00:00	3	1500
101	2009-11-20 00:00:00	2	1560
103	2008-05-20 00:00:00	4	2060

ID	NAME	AMOUNT	DATE
1	Ramesh	NULL	NULL
2	Khilan	1560	2009-11-20 00:00:00
3	kaushik	3000	2009-10-08 00:00:00
3	kaushik	1500	2009-10-08 00:00:00
4	Chaitali	2060	2008-05-20 00:00:00
5	Hardik	NULL	NULL
6	Komal	NULL	NULL
7	Muffy	NULL	NULL
3	kaushik	3000	2009-10-08 00:00:00
3	kaushik	1500	2009-10-08 00:00:00
2	Khilan	1560	2009-11-20 00:00:00
4	Chaitali	2060	2008-05-20 00:00:00

Associating Tables

- Link/connect two tables by a common column
- Two ways:
  - Joins
  - Relates

Parcel (before Join)

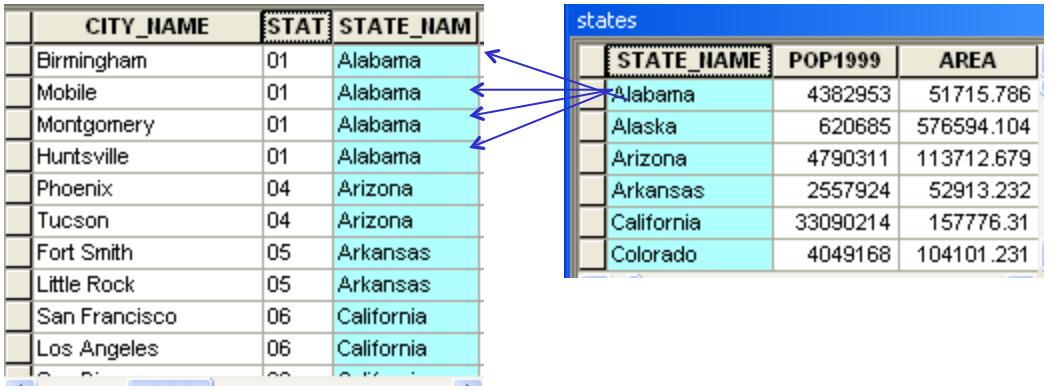
OBJECTID*	SHAPE*	PARCEL_ID	ZONE_CODE	SHAPE_Length	SHAPE_Area
1	Polygon	6750	601	512.602492	13042.492751
2	Polygon	6724	601	372.992656	6203.424403
3	Polygon	6724	603	353.692046	5446.766292
4	Polygon	6725	603	313.013884	5380.550025

ZoneCodeDesc

OBJECTID*	ZONE*	DESCRIPTION
1	601	Commercial
2	602	Institutional
3	603	Residential

Common fields:
ZONE_CODE
ZONE

Join or Relate Attribute Tables in ArcGIS



CITY_NAME	STAT	STATE_NAM
Birmingham	01	Alabama
Mobile	01	Alabama
Montgomery	01	Alabama
Huntsville	01	Alabama
Phoenix	04	Arizona
Tucson	04	Arizona
Fort Smith	05	Arkansas
Little Rock	05	Arkansas
San Francisco	06	California
Los Angeles	06	California

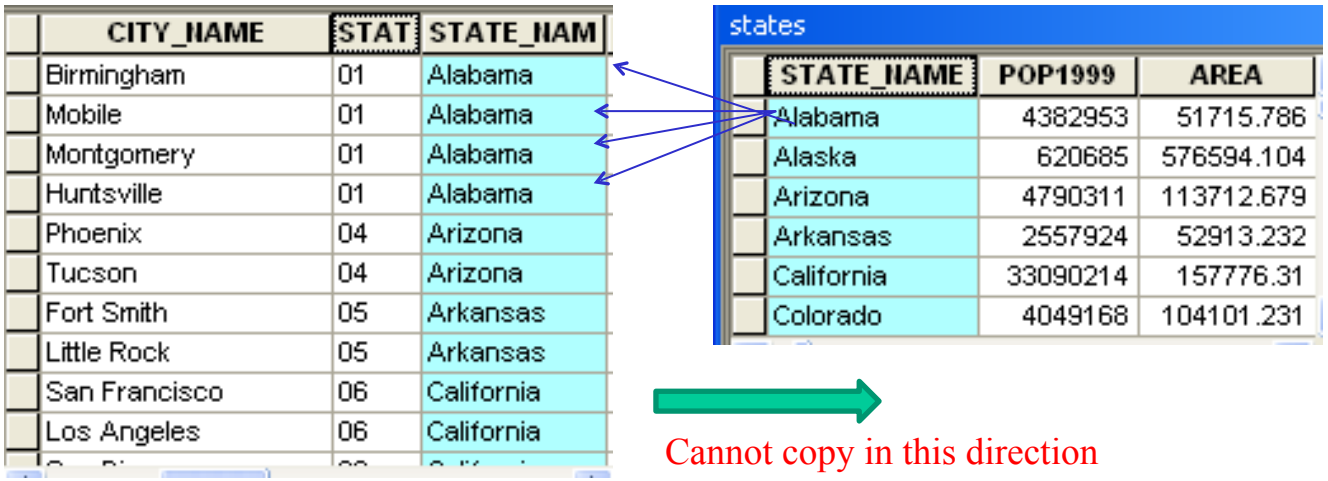
states

STATE_NAME	POP1999	AREA
Alabama	4382953	51715.786
Alaska	620685	576594.104
Arizona	4790311	113712.679
Arkansas	2557924	52913.232
California	33090214	157776.31
Colorado	4049168	104101.231

- Join operation
  - **Combines** two tables
    - E.g., cities and states
  - The two tables have a ***common field***
- Join operation in ArcGIS
  - **append** the attributes of ***B*** (the source/from table) to the attributes of ***A*** (the target/to table), if the common field *is equal*
  - works only for a many-to-one (m:1) relationship ("m" on the target/to table)
    - 1:1 is a special case of m:1

Join Tables in ArcGIS

- The appending method will not work for the other direction
  - Cannot add new rows to a table as each row is a geographical feature!
- 1:m or m:n relations require the **relate** tool



CITY_NAME	STAT	STATE_NAM
Birmingham	01	Alabama
Mobile	01	Alabama
Montgomery	01	Alabama
Huntsville	01	Alabama
Phoenix	04	Arizona
Tucson	04	Arizona
Fort Smith	05	Arkansas
Little Rock	05	Arkansas
San Francisco	06	California
Los Angeles	06	California

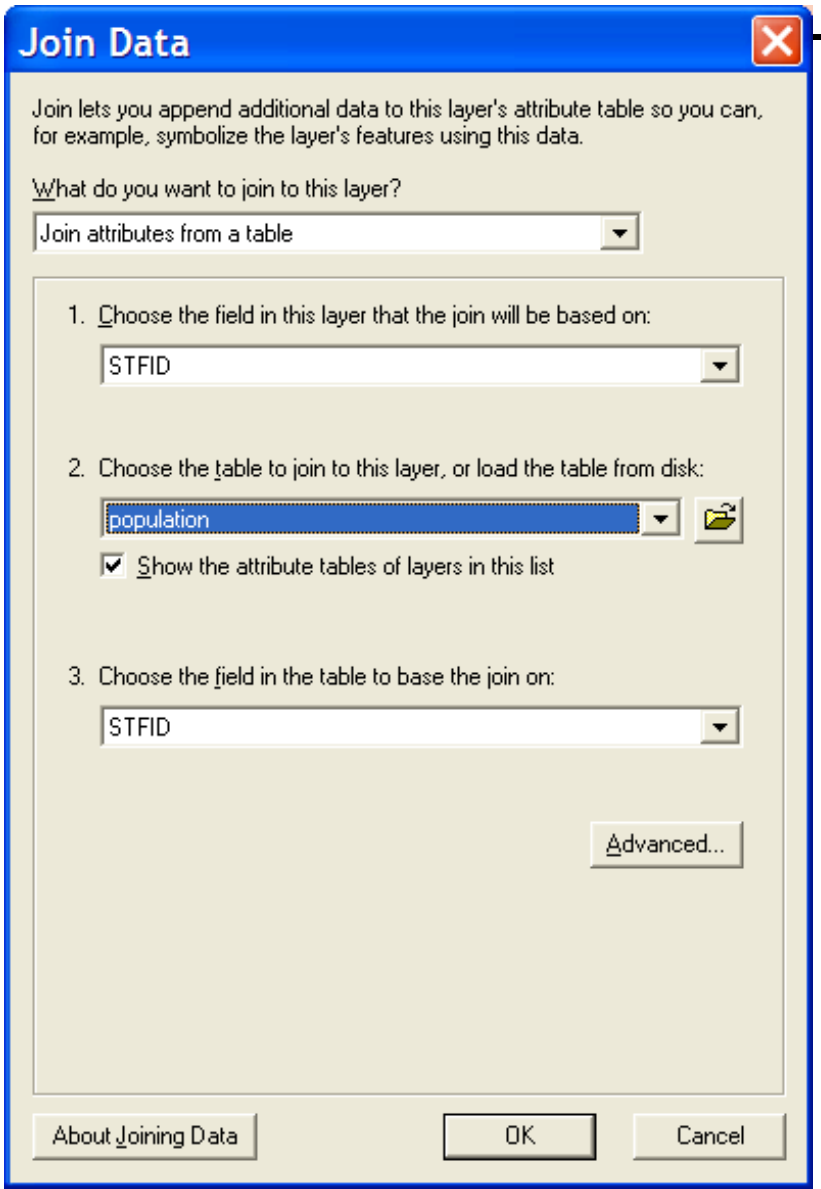
states

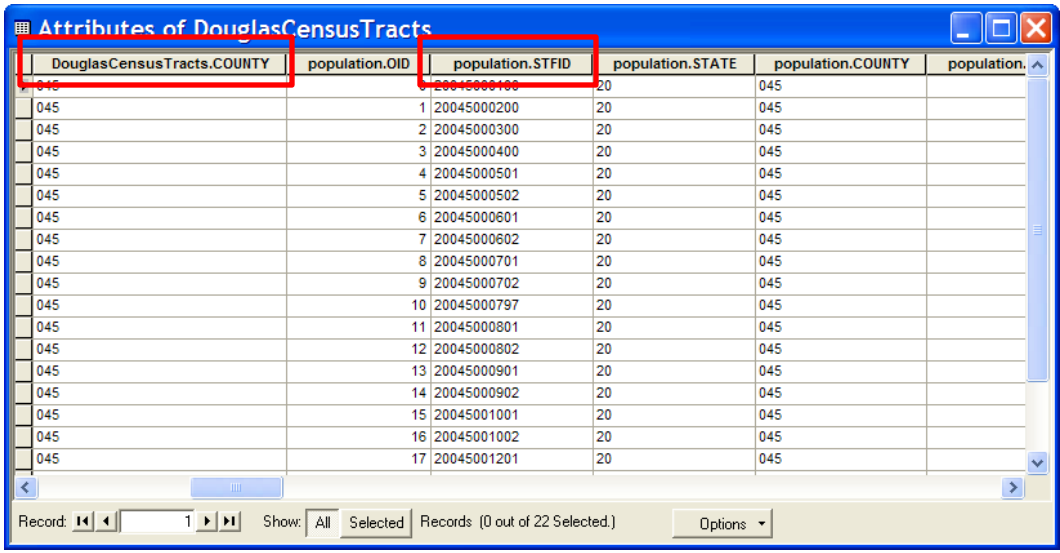
STATE_NAME	POP1999	AREA
Alabama	4382953	51715.786
Alaska	620685	576594.104
Arizona	4790311	113712.679
Arkansas	2557924	52913.232
California	33090214	157776.31
Colorado	4049168	104101.231

Join Tables

- Join--append the attributes from one table (from table) onto another table (to table)
  - Based on a common field
  - The names of the common fields are NOT necessary the same
  - Based on the equal relationship
  - Attributes copied
- Cardinality of relationships (to:from)
  - Good for 1:1 and N:1

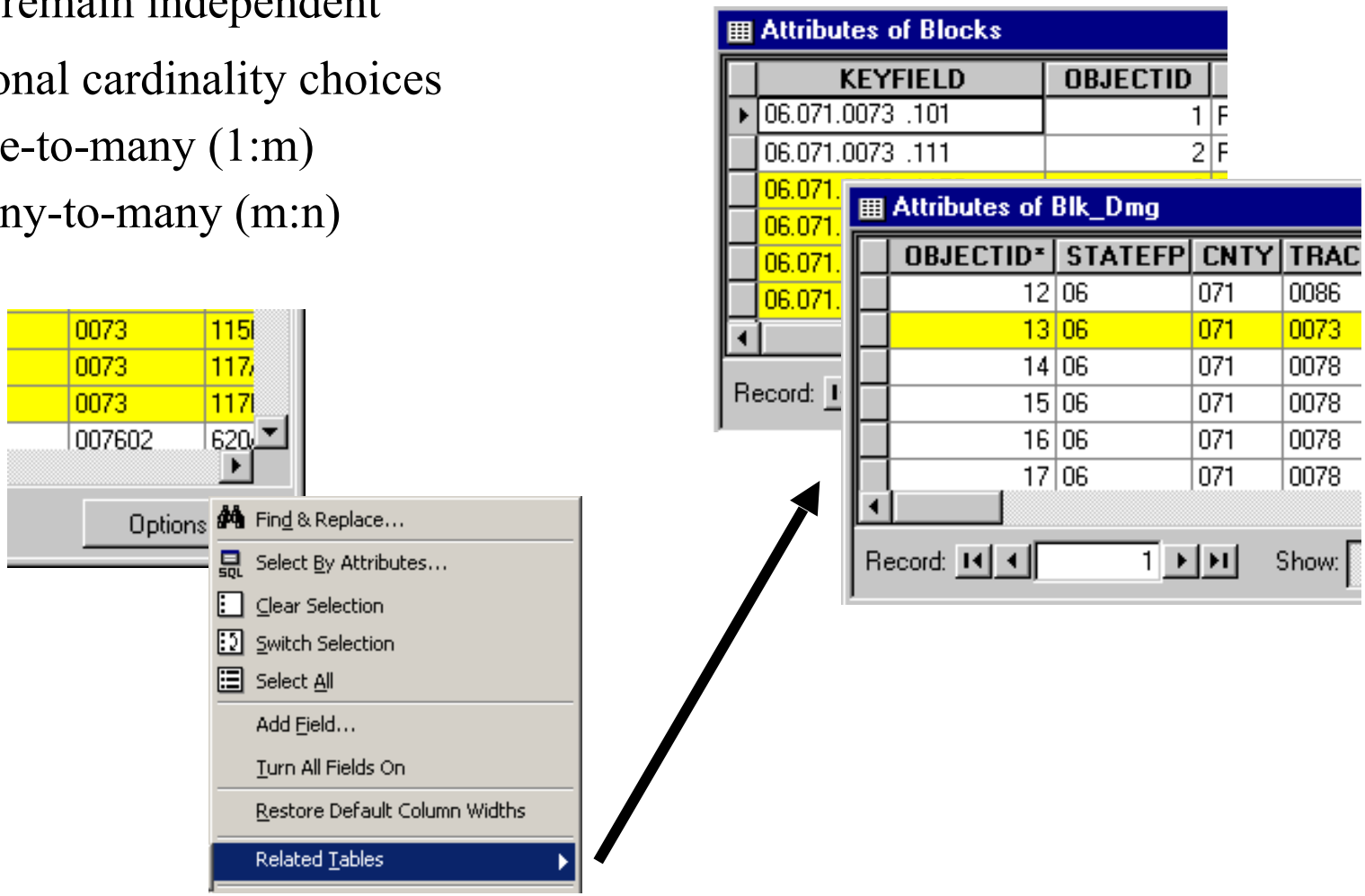
Join Attribute Tables in ArcMap



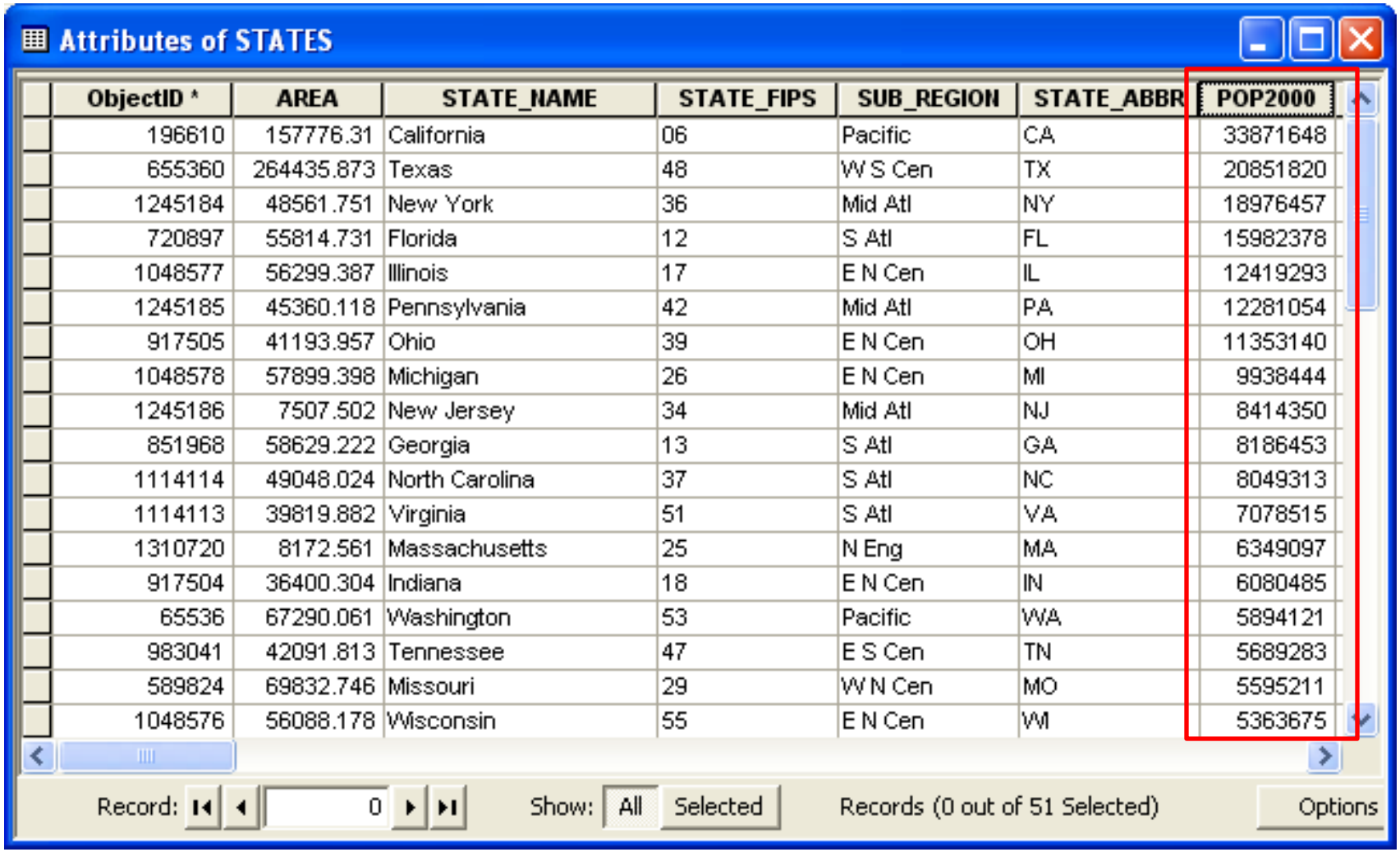


Relate Tables

- Relate rows in two tables using a common field
- Tables remain independent
- Additional cardinality choices
  - One-to-many (1:m)
  - many-to-many (m:n)



Sorting



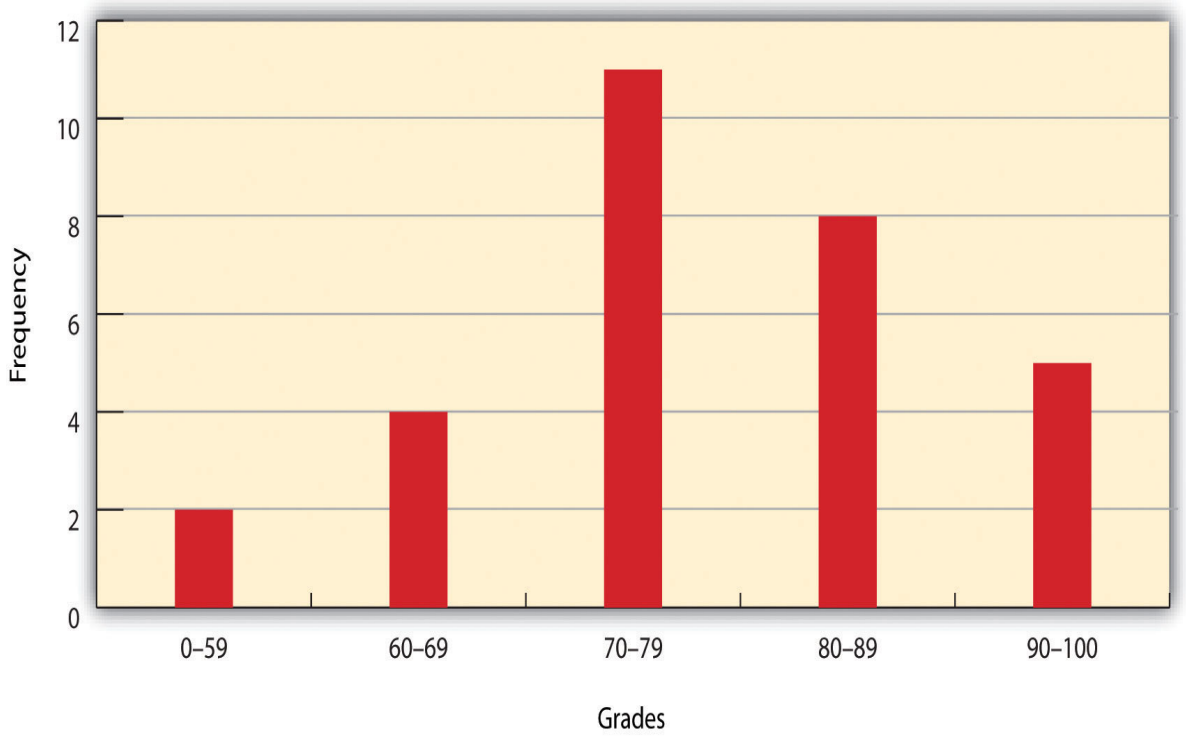
ObjectID *	AREA	STATE_NAME	STATE_FIPS	SUB_REGION	STATE_ABBR	POP2000
196610	157776.31	California	06	Pacific	CA	33871648
655360	264435.873	Texas	48	W S Cen	TX	20851820
1245184	48561.751	New York	36	Mid Atl	NY	18976457
720897	55814.731	Florida	12	S Atl	FL	15982378
1048577	56299.387	Illinois	17	E N Cen	IL	12419293
1245185	45360.118	Pennsylvania	42	Mid Atl	PA	12281054
917505	41193.957	Ohio	39	E N Cen	OH	11353140
1048578	57899.398	Michigan	26	E N Cen	MI	9938444
1245186	7507.502	New Jersey	34	Mid Atl	NJ	8414350
851968	58629.222	Georgia	13	S Atl	GA	8186453
1114114	49048.024	North Carolina	37	S Atl	NC	8049313
1114113	39819.882	Virginia	51	S Atl	VA	7078515
1310720	8172.561	Massachusetts	25	N Eng	MA	6349097
917504	36400.304	Indiana	18	E N Cen	IN	6080485
65536	67290.061	Washington	53	Pacific	WA	5894121
983041	42091.813	Tennessee	47	E S Cen	TN	5689283
589824	69832.746	Missouri	29	W N Cen	MO	5595211
1048576	56088.178	Wisconsin	55	E N Cen	WI	5363675

Calculate New Fields

- Create a new field
  - MinorityPercentage = (Population – White) / Population
- Descriptive statistics for numeric fields
  - Distribution: histogram
  - Central tendency: mean, median, mode
  - Dispersion: range, variance, standard deviation

Measures of Distribution

- Commonly illustrated using a histogram
- A summary of the frequency of values over the range of the dataset



Measures of Central Tendency

- Mean - More commonly referred to as the average, is the most often used measure of central tendency
- Mode - The mode is the measure of central tendency that represents the most frequently occurring value
- Median - The value in the middle of the sorted values when there are an odd number of observations

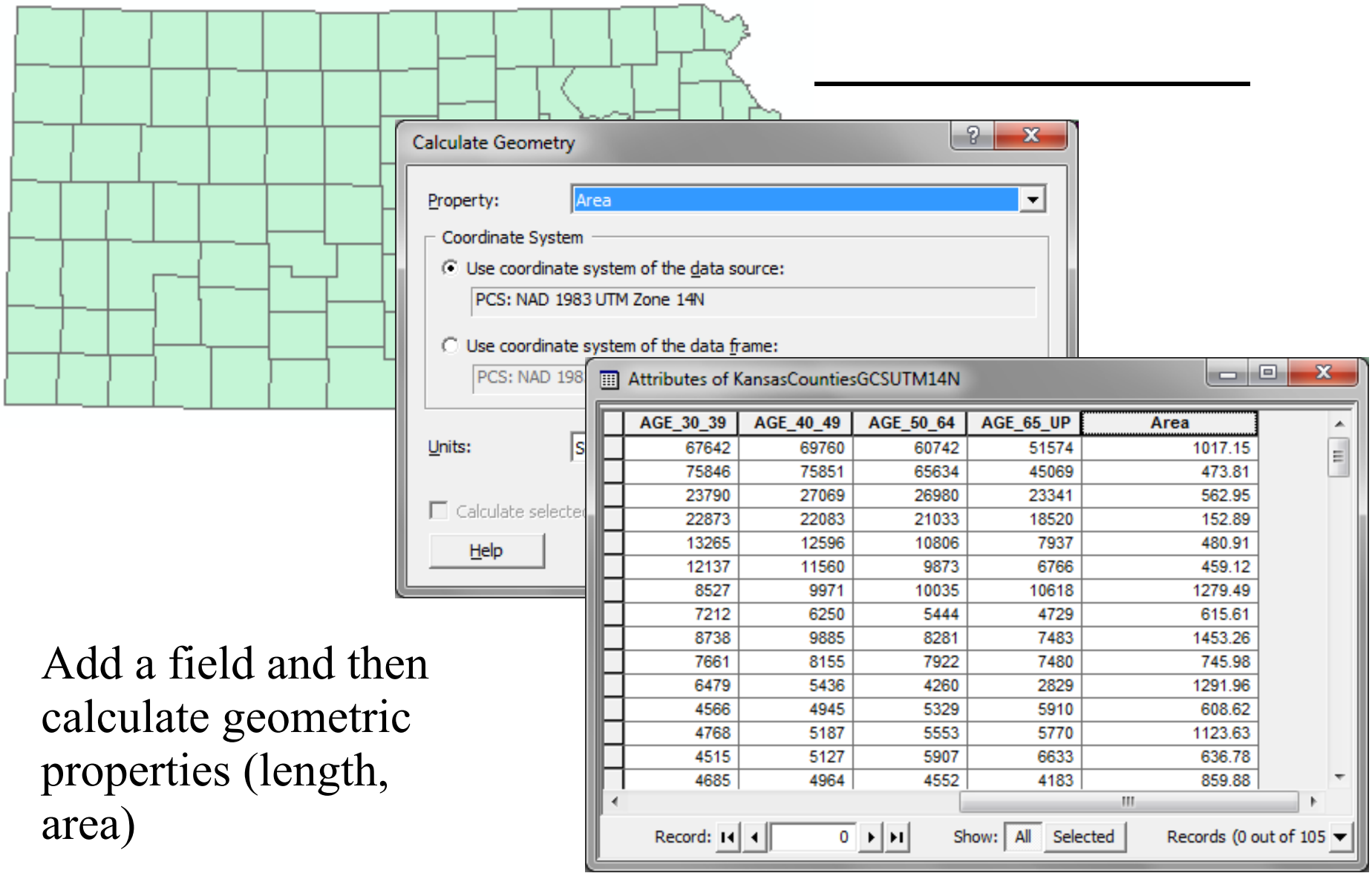
Measures of Dispersion

- Describe the spread of data around the mean
- The simplest measure of dispersion is the range
  - Max. value minus min. value
- Other measure of dispersion are the variance (s^2) and standard deviation (s)

$$S^2 = \frac{\sum (x-\bar{x})^2}{n}$$

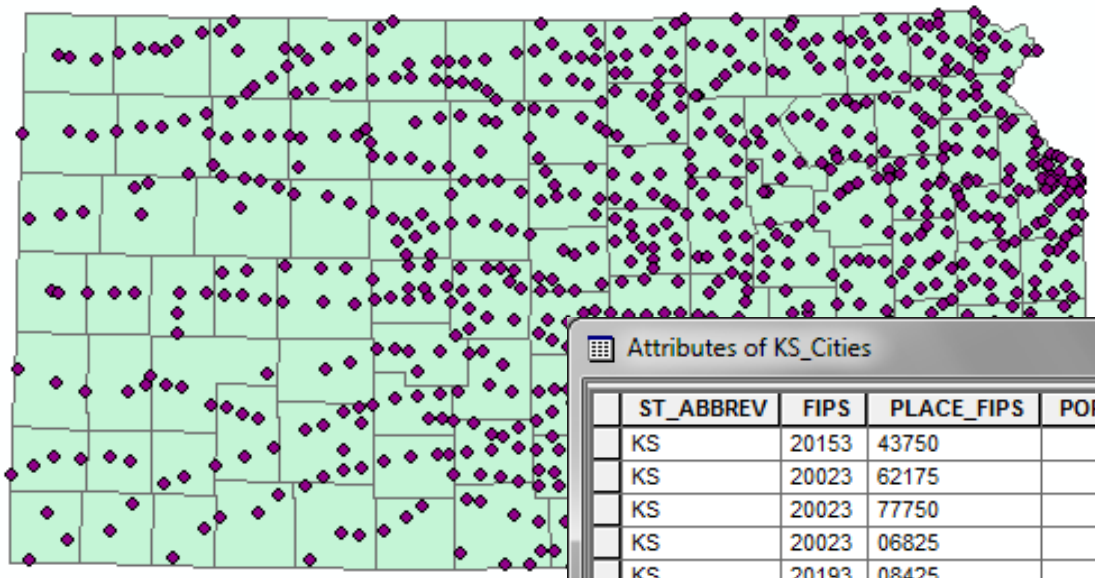
$$S = \sqrt{\frac{\sum (x-\bar{x})^2}{n}}$$

Calculating Feature Geometry Property



Add a field and then calculate geometric properties (length, area)

Getting Feature Coordinates



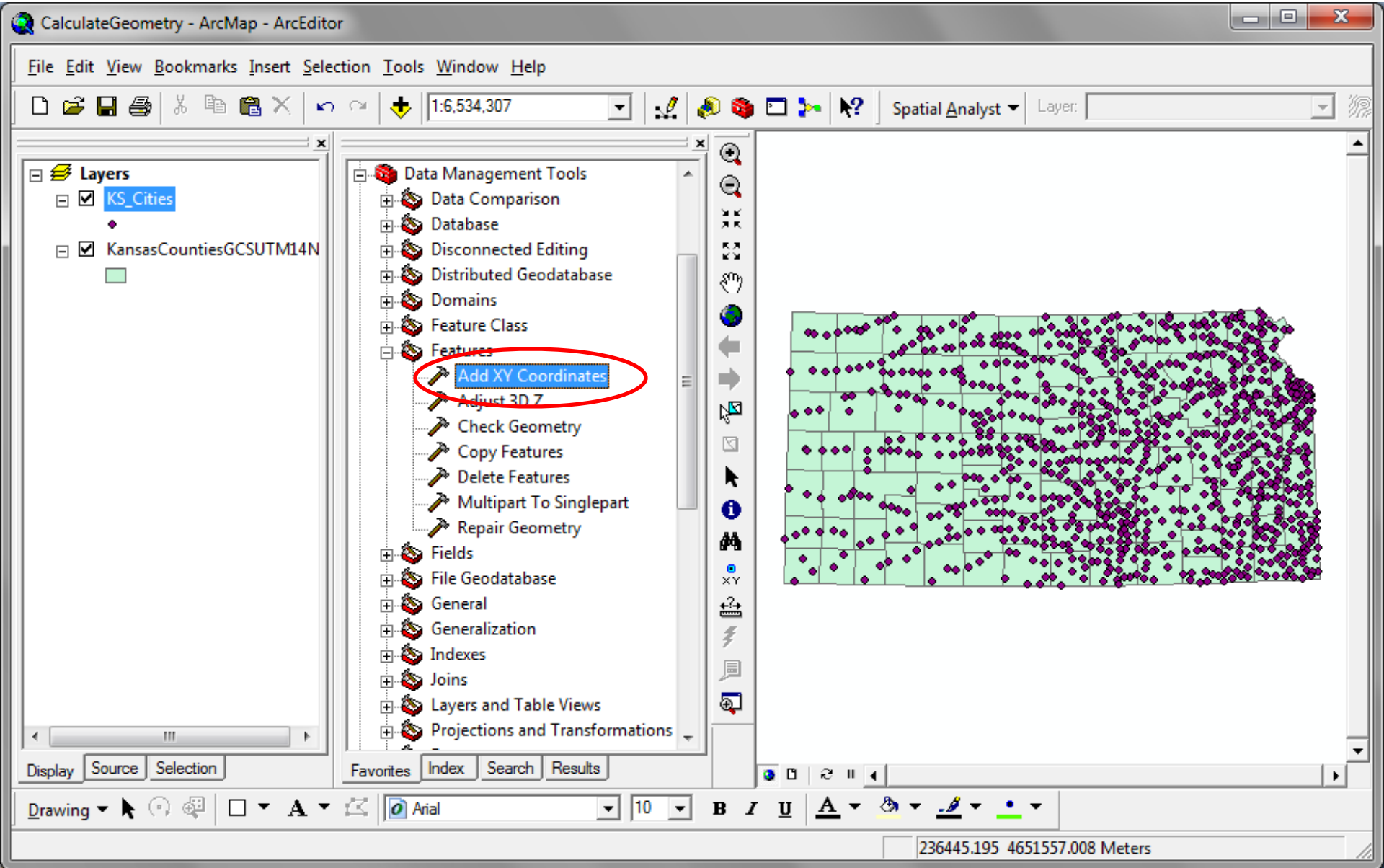
Add fields X and Y. Then calculate the geometry

Attributes of KS_Cities

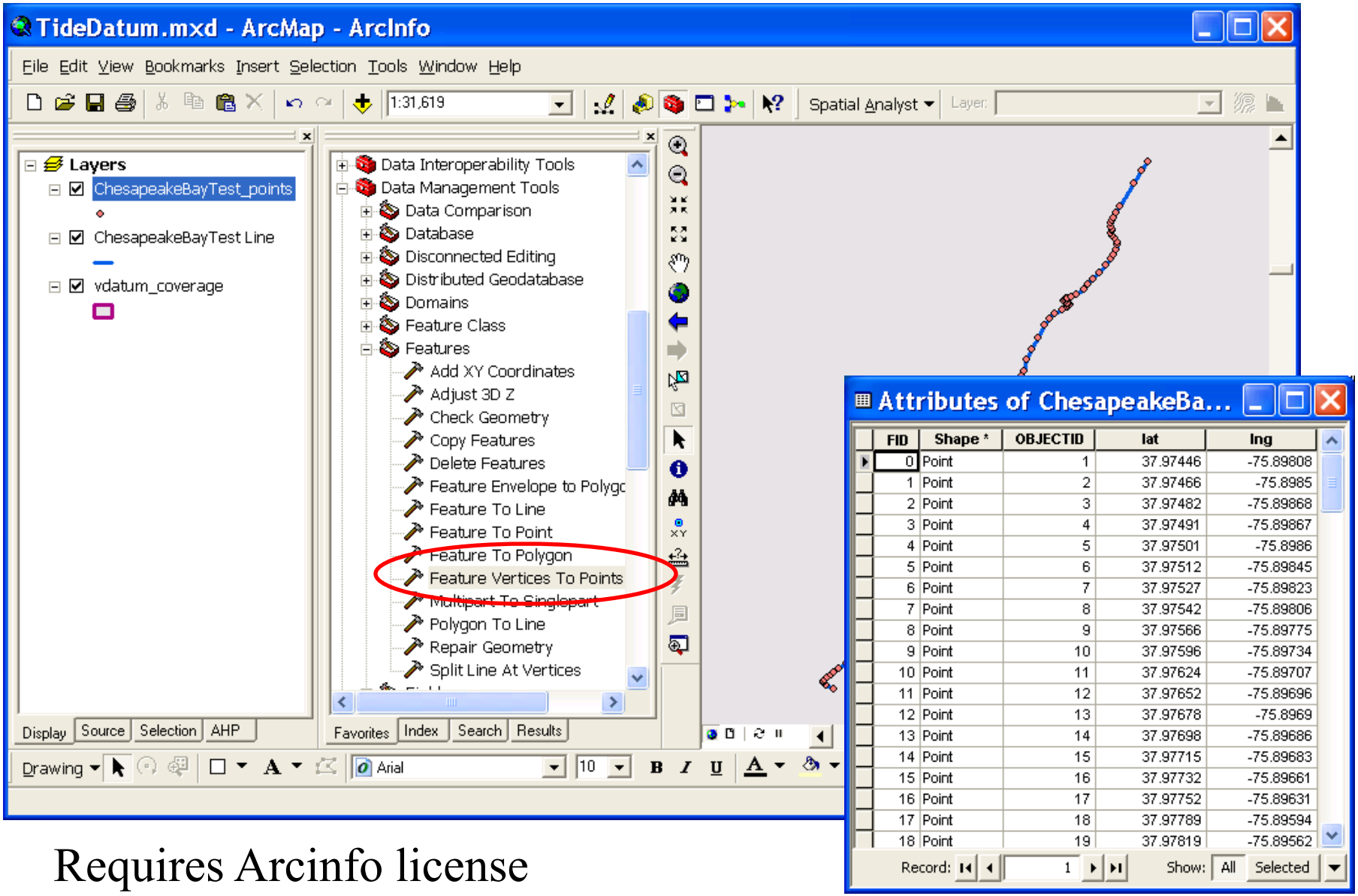
ST_ABBREV	FIPS	PLACE_FIPS	POP_98	STATUS	X	Y
KS	20153	43750	165		-101.3764	39.7857
KS	20023	62175	1476	County Seat	-101.8004	39.772
KS	20023	77750	0		-101.7144	39.7643
KS	20023	06825	458		-101.5336	39.7507
KS	20193	08425	275		-101.377	39.3666
KS	20181	26875	4669	County Seat	-101.7115	39.3507
KS	20181	35975	263		-102.0381	39.3333
KS	20181	19925	0		-101.5413	39.3373
KS	20199	74750	76		-101.5922	38.9116
KS	20199	64375	850	County Seat	-101.7531	38.898
KS	20199	76675	0		-101.9652	38.8668
KS	20071	33150	174		-101.7914	38.4766
KS	20203	39550	1687	County Seat	-101.3592	38.4797
KS	20203	63900	0		-101.5443	38.4734
KS	20071	71450	846	County Seat	-101.7535	38.4699

Record: 0 Show: All Selected Records (0 out of 923 Selected)

Getting Feature Coordinates



Getting Feature Coordinates



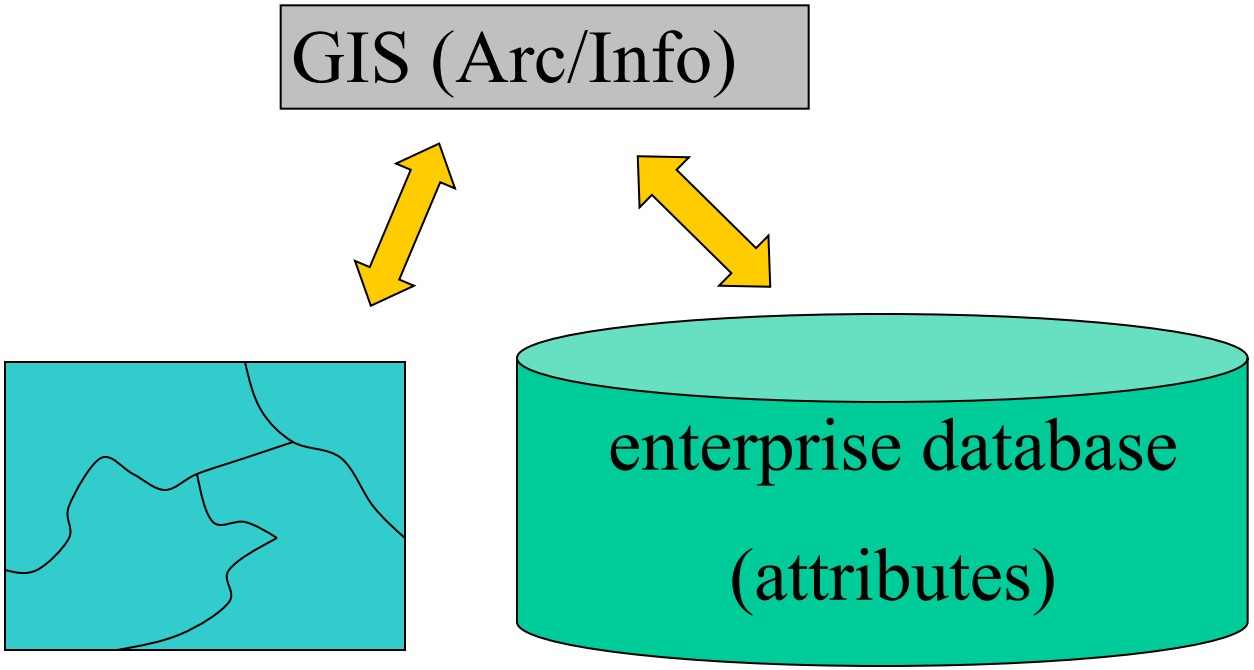
Requires Arcinfo license

SQL and Relation Algebra

- Relational algebra (RA) consists of operators that manipulate tables in a relational database
- Relational operators
  - Restrict, project, union, intersection, difference, product, join, divide
- SQL is essentially built on relational algebra
- SQL queries are translated into sequence of relational operators

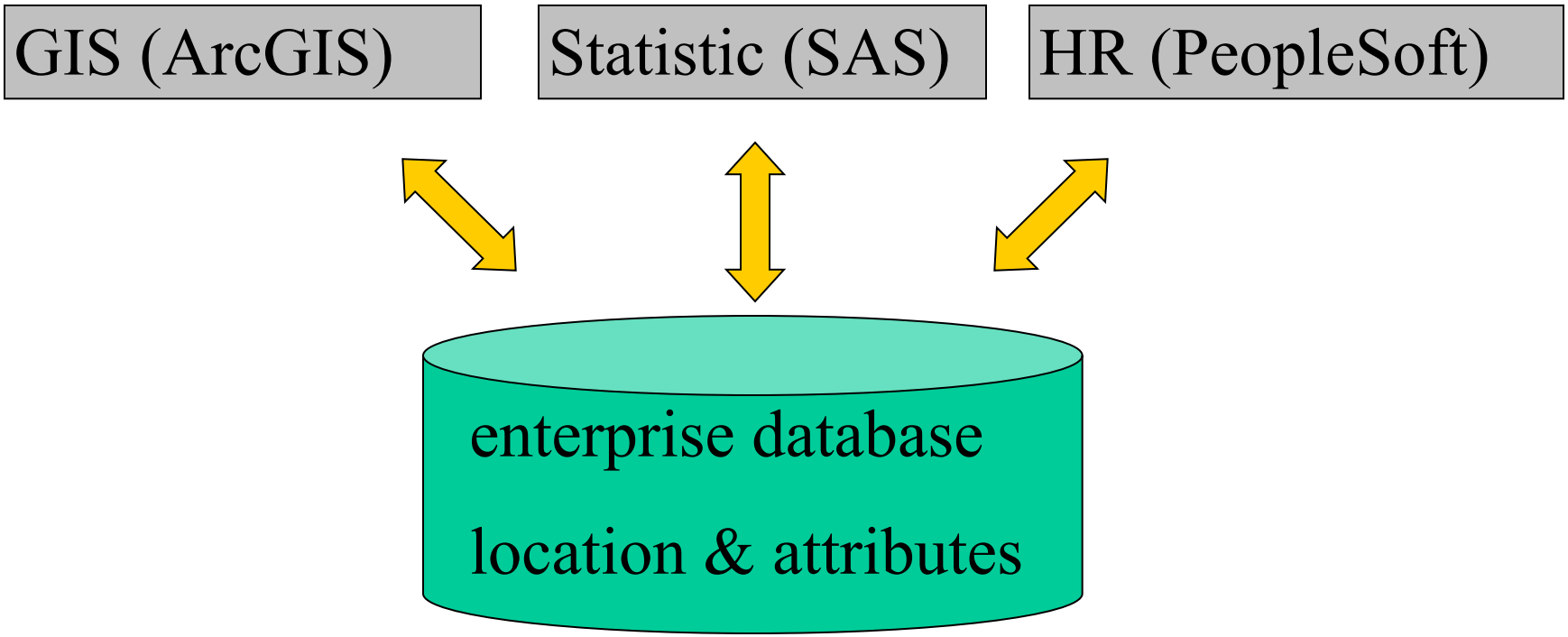
Old GIS Databases

- A hybrid system where locations are stored outside the relational database



Modern GIS Databases

- Both attributes and locations are stored inside a relational database



Multi-tiered Architecture

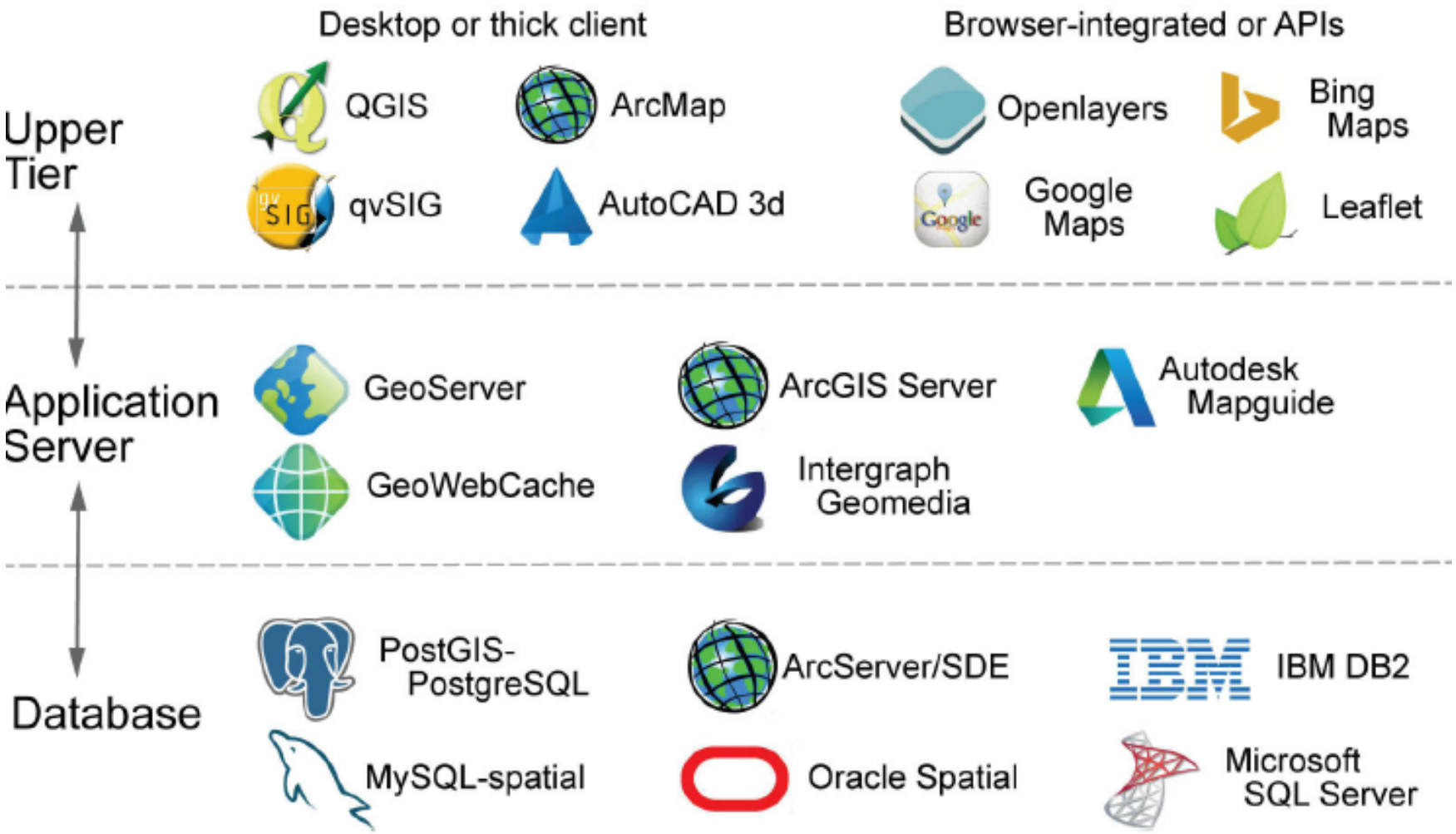


Figure 8-5: Multi-tiered architecture, and common software alternatives. Data are stored and accessed

Database

- Databases are significant infrastructures for many organizations
- Knowledge on database gives you an edge on finding GIS jobs
- Open source DBMS
  - PostgreSQL and PostGIS
  - MySQL
  - SpatialLite
- Commercial DBMS
  - Oracle Corporation (Oracle)
  - Microsoft Corporation (SQL Server)
  - IBM Corporation (DB2)
- GEOG 528, Spatial Databases