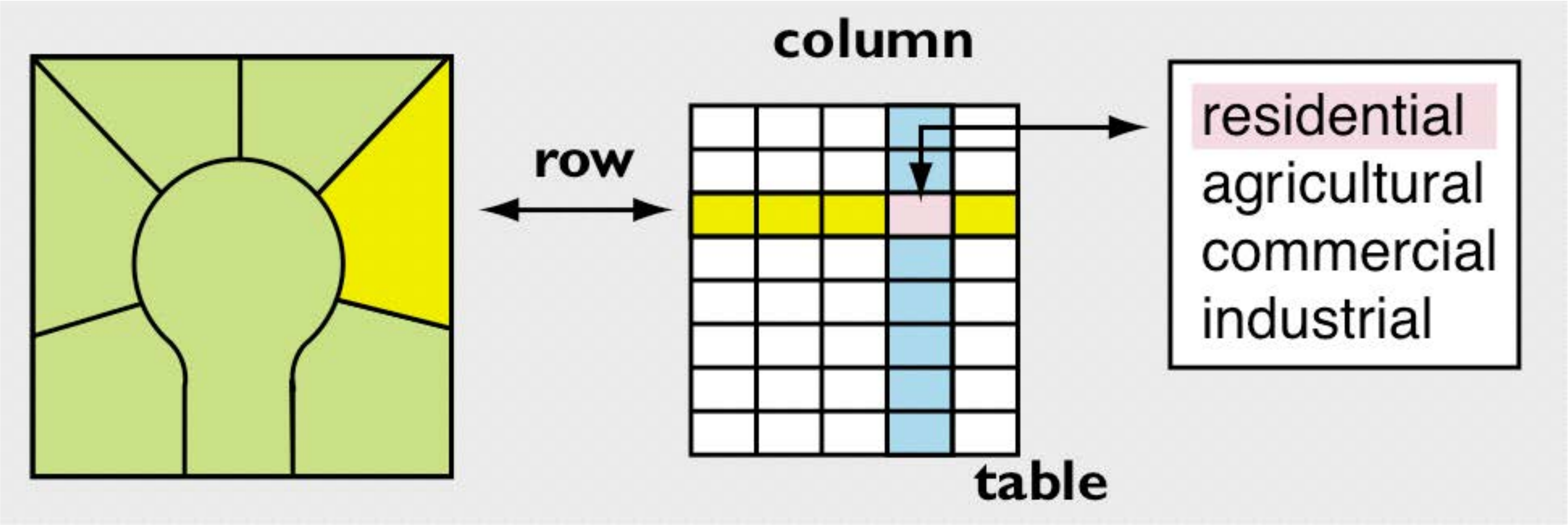
Attribute Tables and Tabular Data

Outline

- Attribute tables
- “Joins” and “relates”
- Relational databases
- Importing tabular data

Attributes

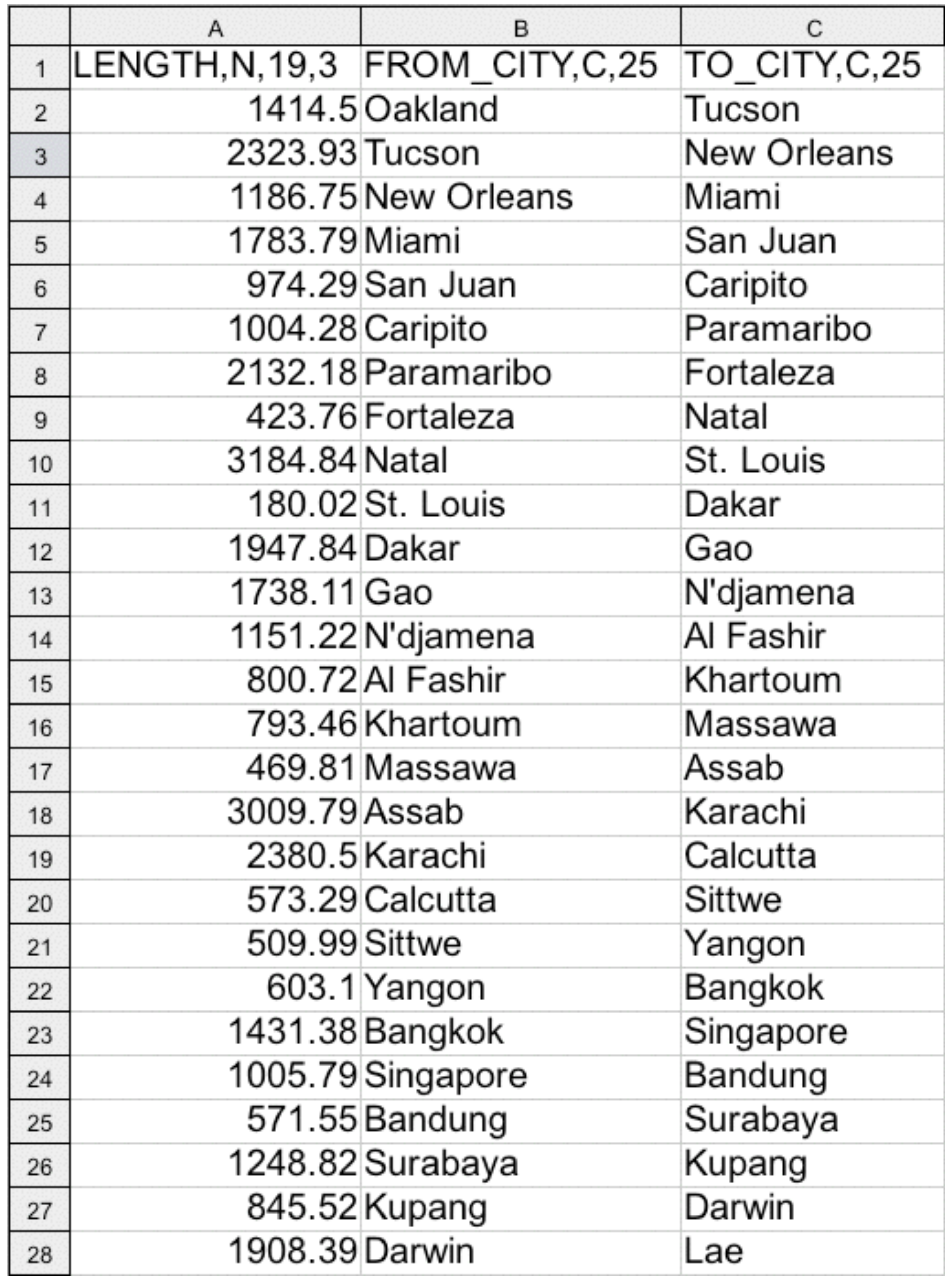
stored/organized in **tables**



- **row** = spatial feature
- **column** = attribute
- $\text{row}_i \cap \text{column}_j$ = value of attribute j for feature i

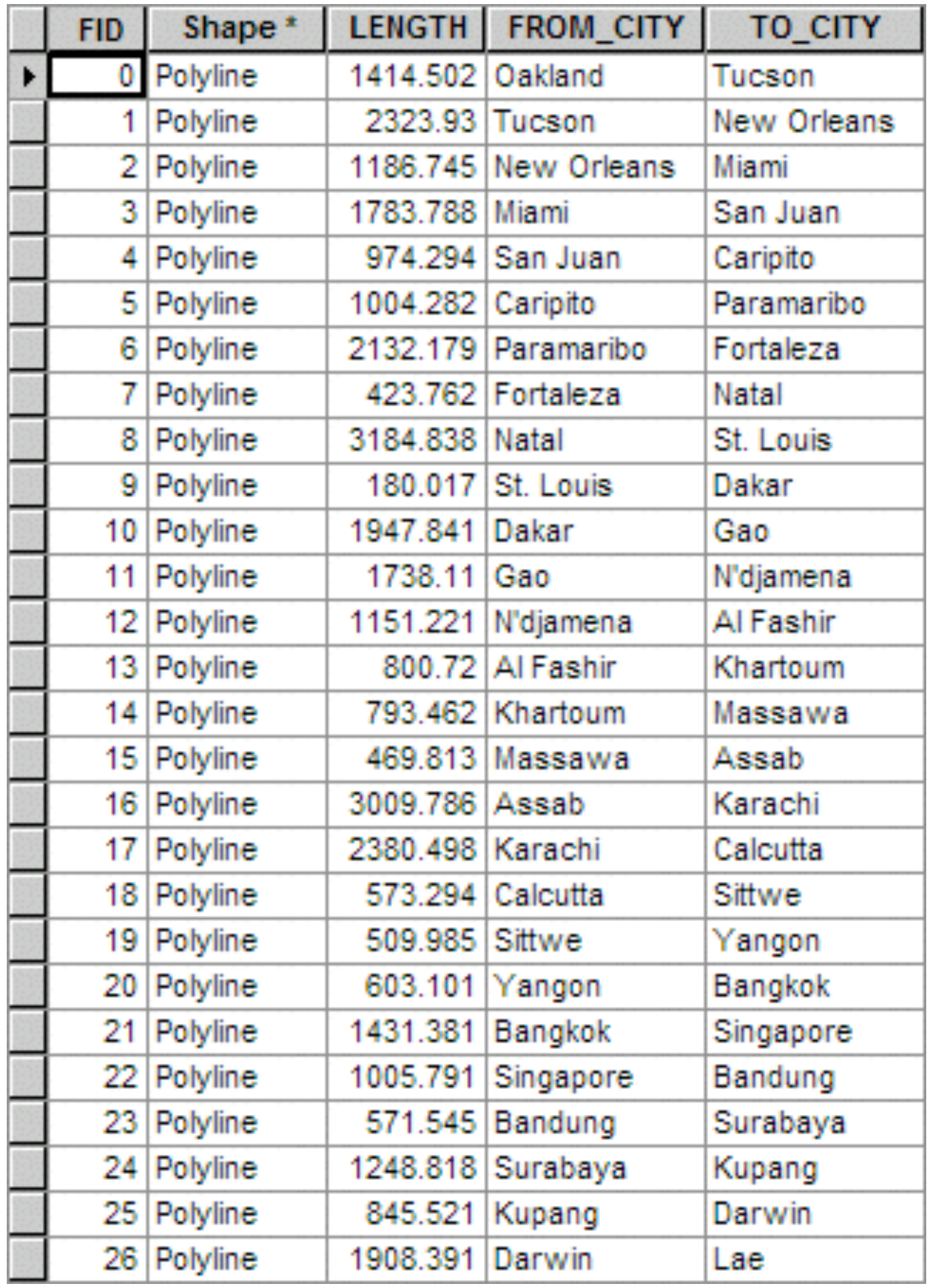
Rendering the Attribute Table

in spreadsheet



	A	B	C
1	LENGTH,N,19,3	FROM_CITY,C,25	TO_CITY,C,25
2	1414.5	Oakland	Tucson
3	2323.93	Tucson	New Orleans
4	1186.75	New Orleans	Miami
5	1783.79	Miami	San Juan
6	974.29	San Juan	Caripito
7	1004.28	Caripito	Paramaribo
8	2132.18	Paramaribo	Fortaleza
9	423.76	Fortaleza	Natal
10	3184.84	Natal	St. Louis
11	180.02	St. Louis	Dakar
12	1947.84	Dakar	Gao
13	1738.11	Gao	N'djamena
14	1151.22	N'djamena	Al Fashir
15	800.72	Al Fashir	Khartoum
16	793.46	Khartoum	Massawa
17	469.81	Massawa	Assab
18	3009.79	Assab	Karachi
19	2380.5	Karachi	Calcutta
20	573.29	Calcutta	Sittwe
21	509.99	Sittwe	Yangon
22	603.1	Yangon	Bangkok
23	1431.38	Bangkok	Singapore
24	1005.79	Singapore	Bandung
25	571.55	Bandung	Surabaya
26	1248.82	Surabaya	Kupang
27	845.52	Kupang	Darwin
28	1908.39	Darwin	Lae

in ArcMap



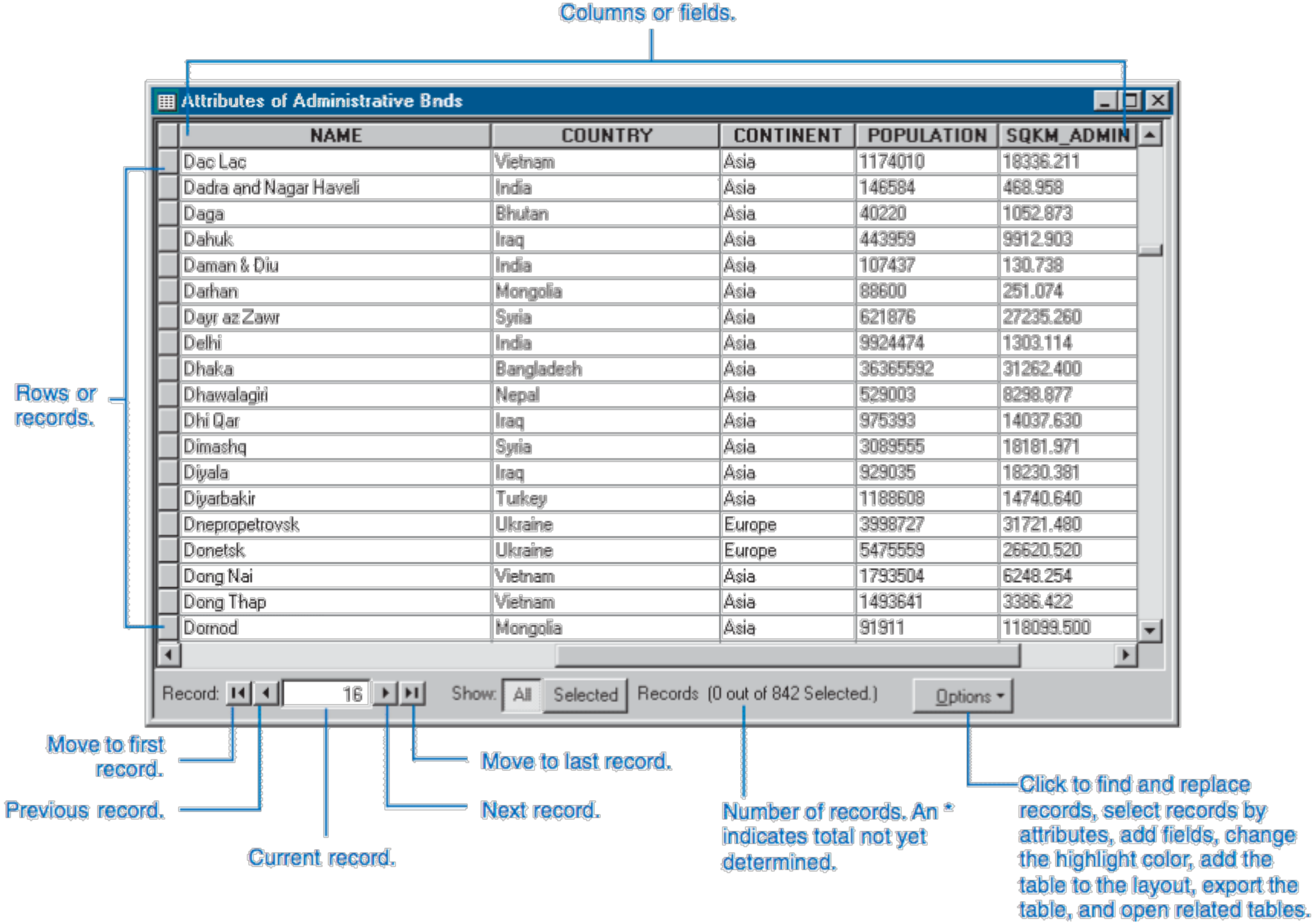
FID	Shape *	LENGTH	FROM_CITY	TO_CITY
0	Polyline	1414.502	Oakland	Tucson
1	Polyline	2323.93	Tucson	New Orleans
2	Polyline	1186.745	New Orleans	Miami
3	Polyline	1783.788	Miami	San Juan
4	Polyline	974.294	San Juan	Caripito
5	Polyline	1004.282	Caripito	Paramaribo
6	Polyline	2132.179	Paramaribo	Fortaleza
7	Polyline	423.762	Fortaleza	Natal
8	Polyline	3184.838	Natal	St. Louis
9	Polyline	180.017	St. Louis	Dakar
10	Polyline	1947.841	Dakar	Gao
11	Polyline	1738.11	Gao	N'djamena
12	Polyline	1151.221	N'djamena	Al Fashir
13	Polyline	800.72	Al Fashir	Khartoum
14	Polyline	793.462	Khartoum	Massawa
15	Polyline	469.813	Massawa	Assab
16	Polyline	3009.786	Assab	Karachi
17	Polyline	2380.498	Karachi	Calcutta
18	Polyline	573.294	Calcutta	Sittwe
19	Polyline	509.985	Sittwe	Yangon
20	Polyline	603.101	Yangon	Bangkok
21	Polyline	1431.381	Bangkok	Singapore
22	Polyline	1005.791	Singapore	Bandung
23	Polyline	571.545	Bandung	Surabaya
24	Polyline	1248.818	Surabaya	Kupang
25	Polyline	845.521	Kupang	Darwin
26	Polyline	1908.391	Darwin	Lae

Table Characteristics

- All tables
  - Row order doesn't matter
    - rows can be ordered on any column value(s)
  - Columns are typed
    - ArcGIS: integer, real, text, date

- Feature attribute tables
  - 1 row per feature
    - table row ← feature ID → geometry object
  - 1 table per feature class
    - shapefile, coverage, geodatabase feature class, ...

Attribute Manipulation in ArcGIS



Attributes of Administrative Bnds

NAME	COUNTRY	CONTINENT	POPULATION	SQKM_ADMIN
Dac Lac	Vietnam	Asia	1174010	18336.211
Dadra and Nagar Haveli	India	Asia	146584	468.958
Daga	Bhutan	Asia	40220	1052.873
Dahuk	Iraq	Asia	443959	9912.903
Daman & Diu	India	Asia	107437	130.738
Darhan	Mongolia	Asia	88600	251.074
Dayr az Zawr	Syria	Asia	621876	27235.260
Delhi	India	Asia	9924474	1303.114
Dhaka	Bangladesh	Asia	36365592	31262.400
Dhawalagiri	Nepal	Asia	529003	8298.877
Dhi Qar	Iraq	Asia	975393	14037.630
Dimashq	Syria	Asia	3089555	18181.971
Diyala	Iraq	Asia	929035	18230.381
Diyarbakir	Turkey	Asia	1188608	14740.640
Dnepropetrovsk	Ukraine	Europe	3998727	31721.480
Donetsk	Ukraine	Europe	5475559	26620.520
Dong Nai	Vietnam	Asia	1793504	6248.254
Dong Thap	Vietnam	Asia	1493641	3386.422
Dornod	Mongolia	Asia	91911	118099.500

Record: 16 Show: All Selected Records (0 out of 842 Selected.) Options

Columns or fields.

Rows or records.

Move to first record.

Previous record.

Current record.

Next record.

Move to last record.

Number of records. An * indicates total not yet determined.

Click to find and replace records, select records by attributes, add fields, change the highlight color, add the table to the layout, export the table, and open related tables.

Attribute Querying in ArcGIS: “Select by Attributes”

simplified SQL query

- columns: all
- table: current
- constraint:

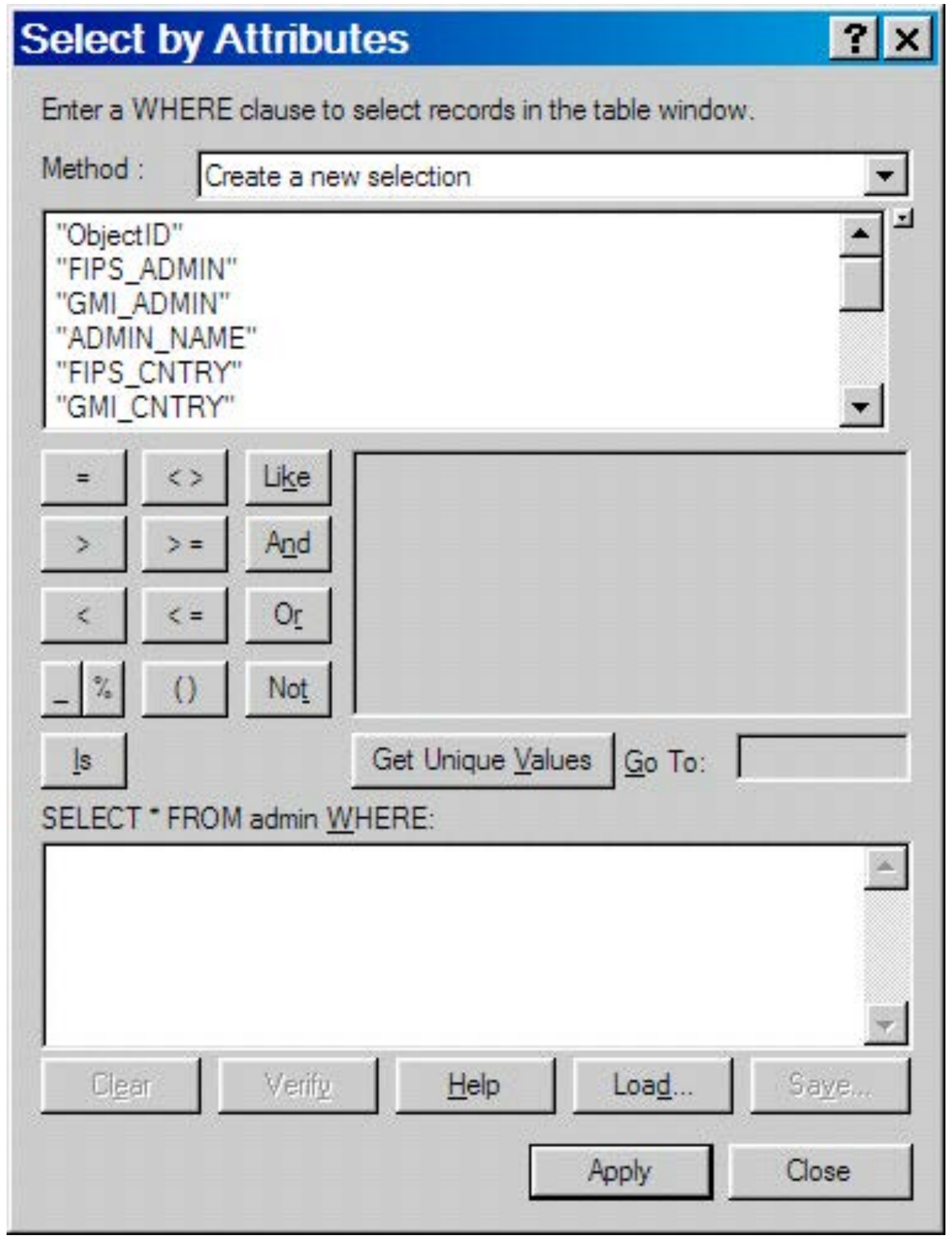
  column {op} value

  – choose from available

  or

  – specify directly

not quite a database!



Select by Attributes

Enter a WHERE clause to select records in the table window.

Method : Create a new selection

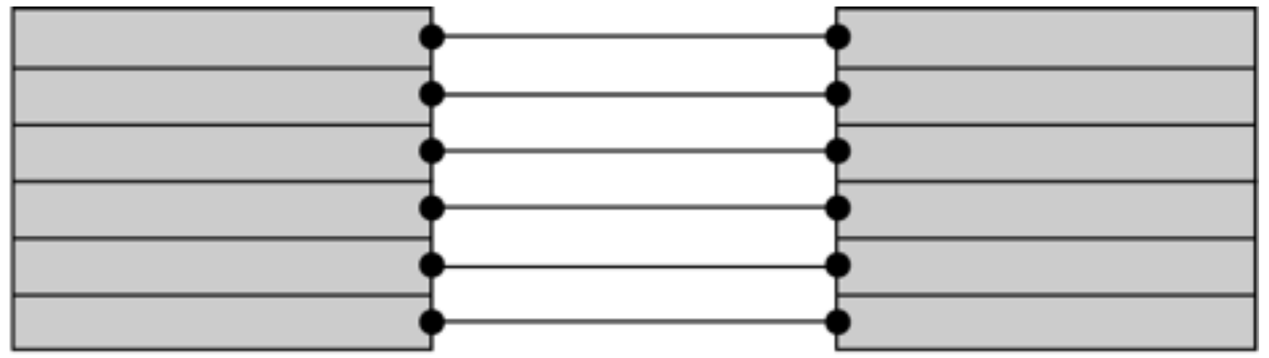
"ObjectID"
"FIPS_ADMIN"
"GMI_ADMIN"
"ADMIN_NAME"
"FIPS_CNTRY"
"GMI_CNTRY"

SELECT * FROM admin WHERE:

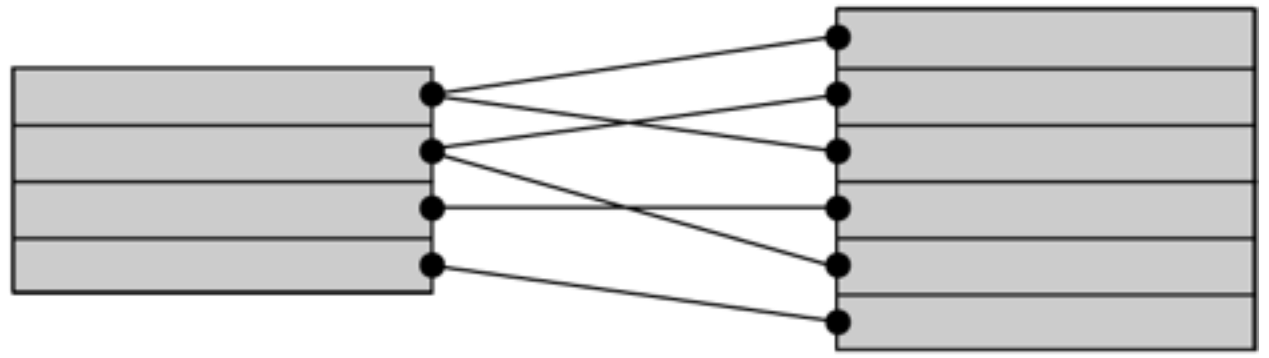
Connecting Tables

What if attributes are in more than 1 table?

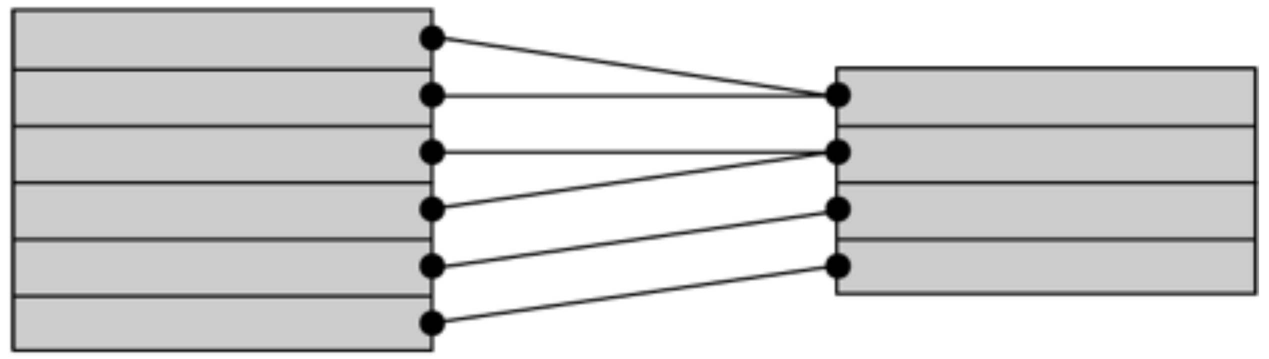
- Create **relationships** between tables
  - **cardinality**: #rows$_{\text{(table1)}}$ ↔ #rows$_{\text{(table2)}}$



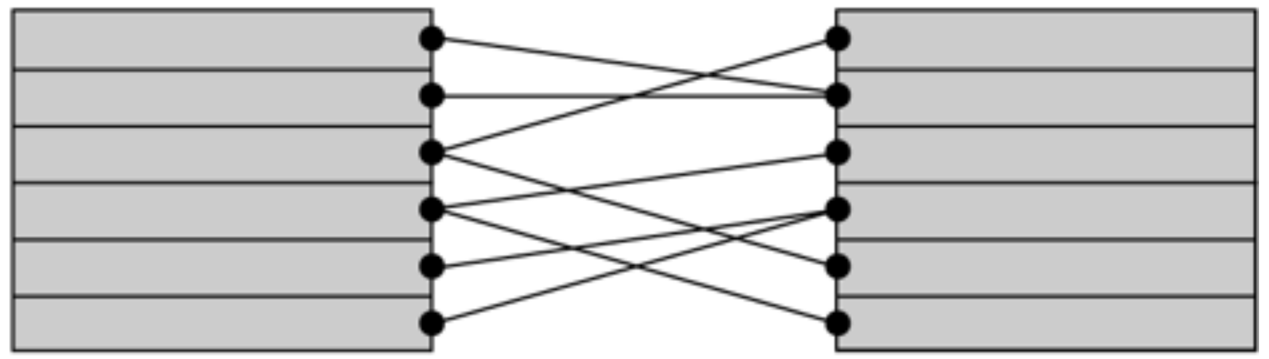
One-to-one relationship



One-to-many relationship



Many-to-one relationship



Many-to-many relationship

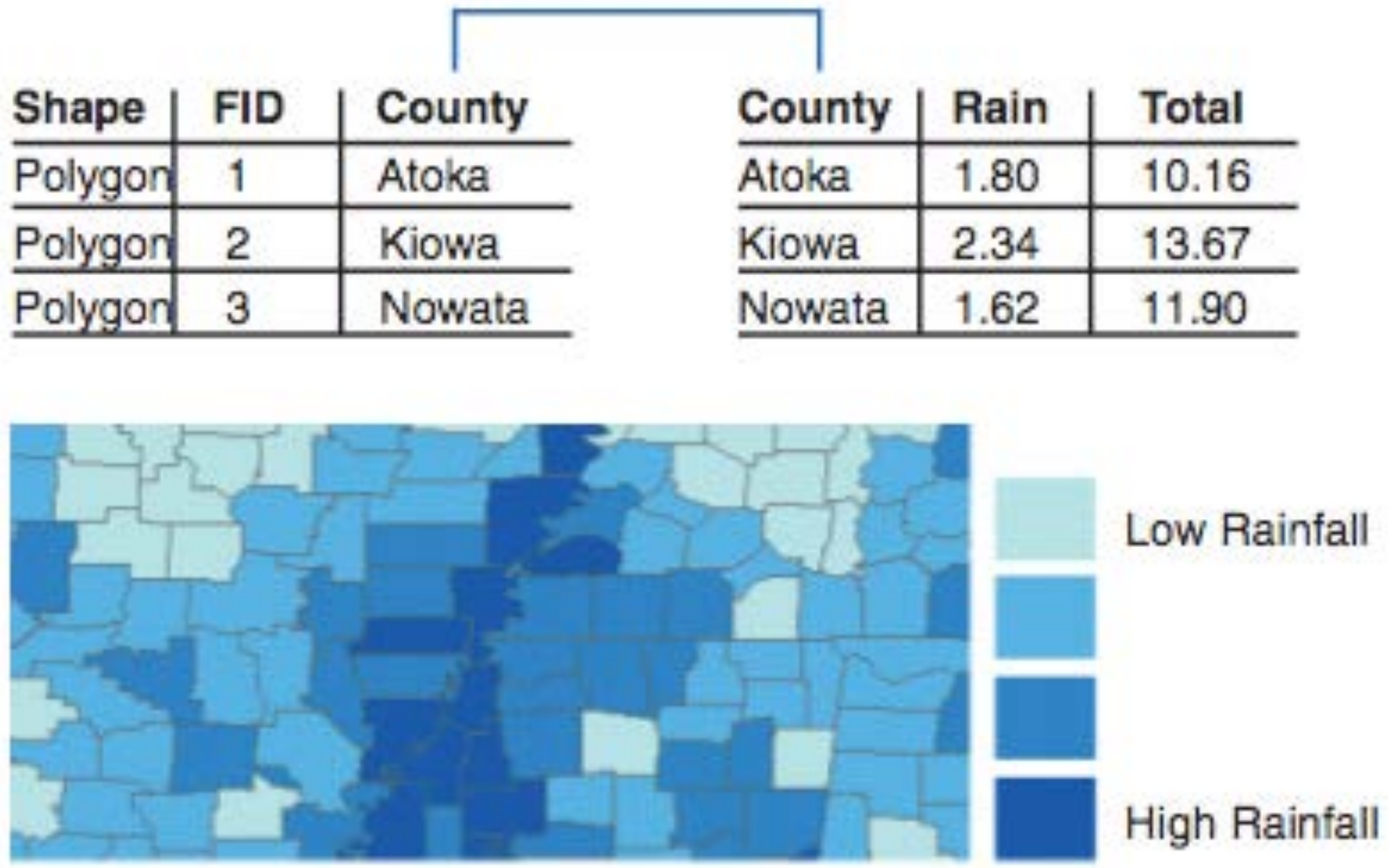
Connecting Tables in ArcGIS

- Connect tables using common key values
  - **join:** concatenates
  - **relate:** links (but keeps separate)

- ArcGIS, **not** database terminolgy!

Joins

- 1 to 1

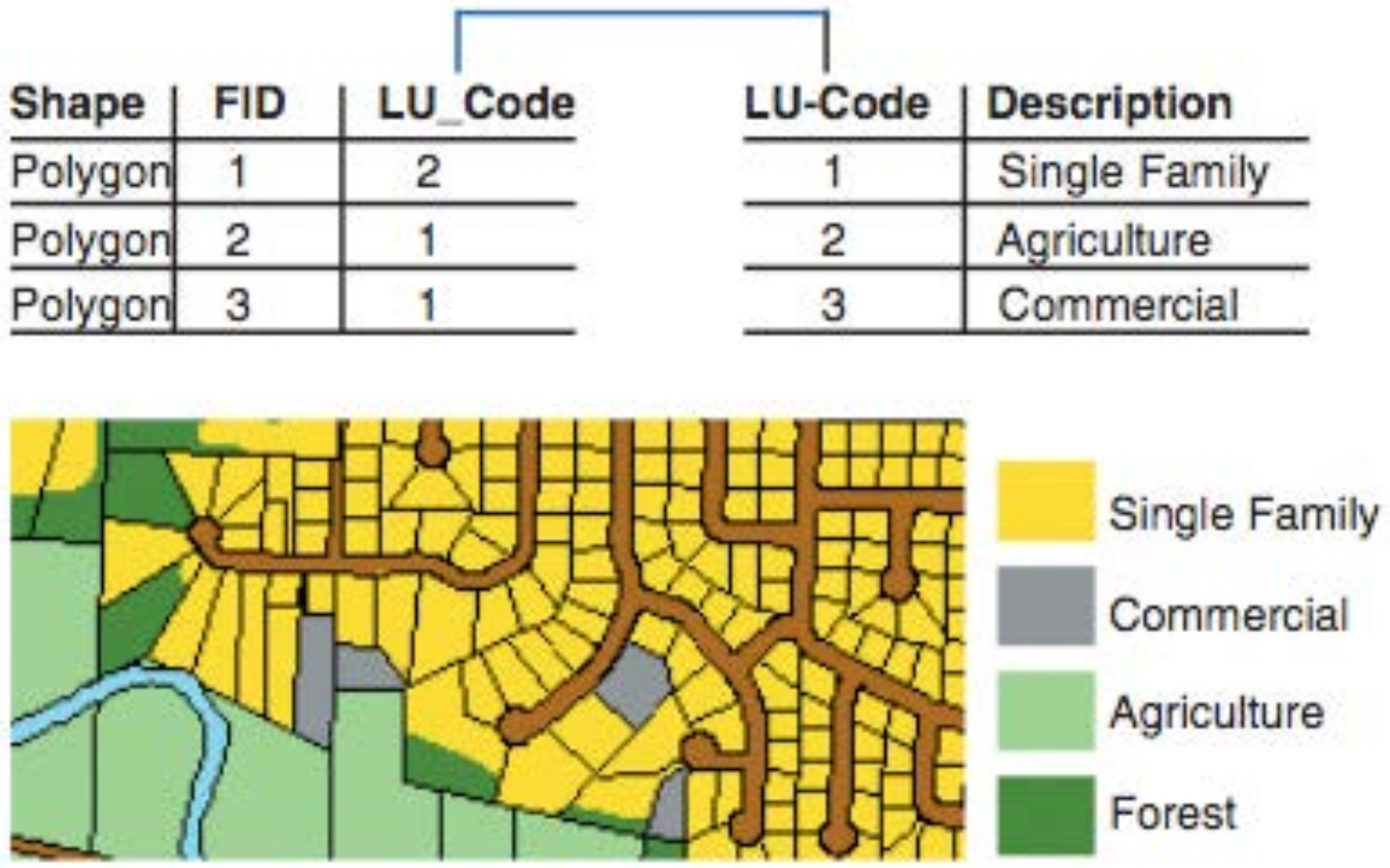


Shape	FID	County
Polygon	1	Atoka
Polygon	2	Kiowa
Polygon	3	Nowata

County	Rain	Total
Atoka	1.80	10.16
Kiowa	2.34	13.67
Nowata	1.62	11.90

Symbolizing features based on joined rainfall data.

- Many to 1



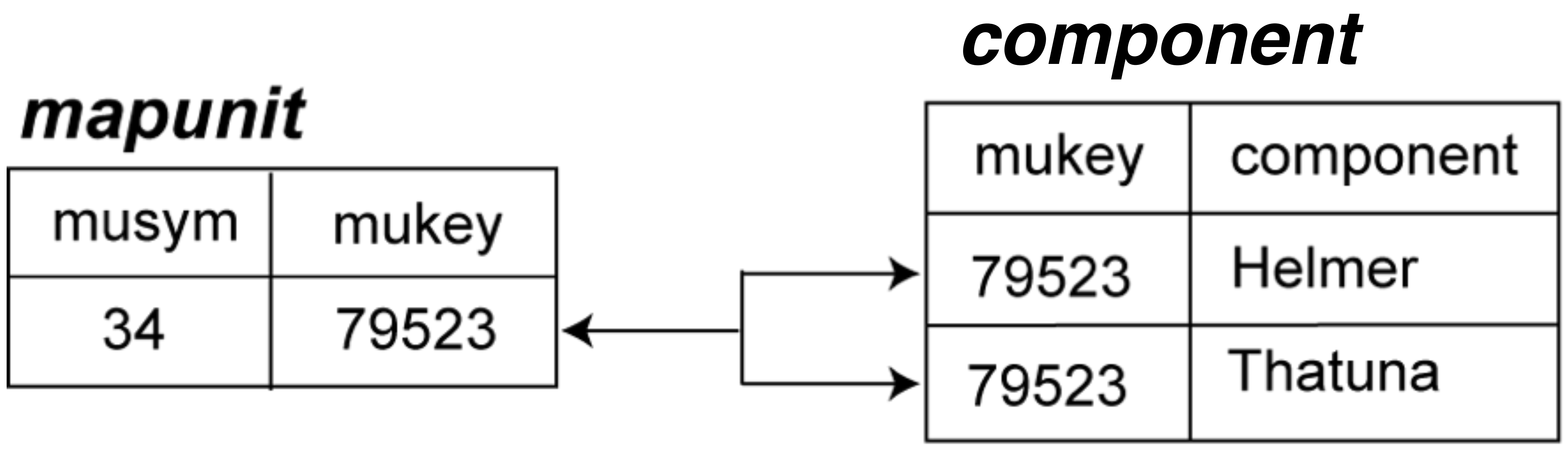
Shape	FID	LU_Code
Polygon	1	2
Polygon	2	1
Polygon	3	1

LU-Code	Description
1	Single Family
2	Agriculture
3	Commercial

Many polygons share the same land use description.

Relate

- 1 to many
  - – e.g. one soil map unit → two soil components



mapunit

musym	mukey
34	79523

component

mukey	component
79523	Helmer
79523	Thatuna

- NB: can’t **join** 1-to-many: why?
  - – would have to replicate features

Summary: Tables in ArcGIS

ArcGIS isn’t quite a database

- Enforces 1 feature ↔ 1 attribute table row
  - Joins and relates are part of map, not data
- Can’t query multiple tables simultaneously
  - Have to explicitly join or relate them first
- Can’t save queries and run them later

What’s a Database?

- A collection of data?
  - e.g. all the data for your group project
- A computer information system?
  - file?
  - spreadsheet?
  - GIS
- Date’s criteria: a database enables/enforces:
  - integration
  - sharing
  - persistence
  - entities, attributes, and relationships

» Date, C.J., “An introduction to database systems (7th ed.)” ISBN 0-201-38590-2

Tables in a Database

- Table = **entity**
  - e.g. professors
- Row = **instance** of an entity
  - e.g. Frew
  - also called: **tuple**
- Column = **attribute** of an entity
  - e.g. shoe size

A database is **really picky** about what you put in a table...

Relation (Table) Rules

- Only one value in each cell (intersection of row and column)
- All values in a column are about the same subject
- Each row is unique
- Column order doesn’t matter
- Row order doesn’t matter

Usually Need More than 1 Table

- Avoid **redundancy**:
  if single table,
  then attribute values **shared** by >1 instance
  must be **repeated** in each instance
  - e.g. 58 students taking 263
  - 263 meets in Bren 1414
  - 58 student records have Bren 1414 as meeting place
  - what happens when class moves?

- Consequences of redundancy
  - more sensitive to typos and transcription errors
  - fragile updates: have to change multiple copies

How Databases Use Multiple Tables

- Eliminate redundancy
  by **normalizing** single table into multiple tables
  - Each table = single kind of thing
  - Each row = single thing

- Preserve relationships
  by **references** between tables
  - Collapse redundant attributes into single **key**
    (attribute shared between tables)
  - Relationships implied by matching key values

Keys

A **key** uniquely identifies,
and can therefore be used as a reference to,
a single row

- **Primary** key
  - attribute whose value uniquely identifies a row
    - Data values that are naturally unique
      - may be more than 1 attribute
    - Arbitrary/synthetic value
      - e.g. auto-incrementing counter
- **Foreign** key
  - attribute whose value corresponds to another row's (usually in another table) primary key
  - **Foreign keys are how databases maintain explicit relationships between rows, within or between tables**

Normalization

- Step 0: non-normalized single table

PIN	Owner	Owner Address	Sale date	Acres	Zone code	Zoning
P101	Wang Chang	101 Oak St. 200 Maple St.	1-10-98	1.0	1	residential
P102	Smith Jones	300 Spruce St. 105 Ash St.	10-6-68	3.0	2	commercial
P103	Costello	206 Elm St.	3-7-97	2.5	2	commercial
P104	Smith	300 Spruce St.	7-30-78	1.0	1	residential

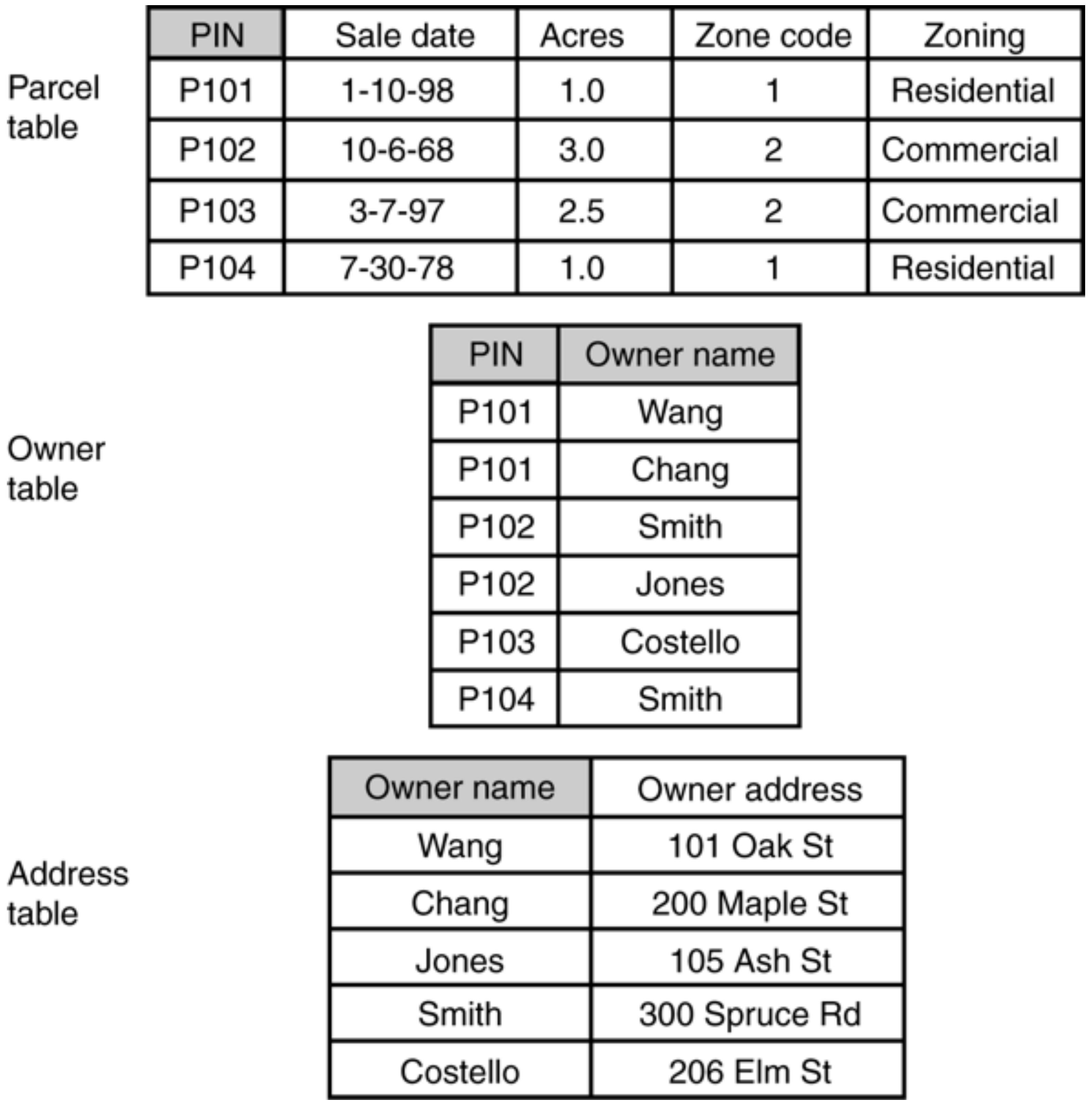
Normalization

- First Normal Form
  - – no multi-valued attributes

PIN	Owner	Owner address	Sale date	Acres	Zone code	Zoning
P101	Wang	101 Oak St	1-10-98	1.0	1	residential
P101	Chang	200 Maple St	1-10-98	1.0	1	residential
P102	Smith	300 Spruce Rd	10-6-68	3.0	2	commercial
P102	Jones	105 Ash St	10-6-68	3.0	2	commercial
P103	Costello	206 Elm St	3-7-97	2.5	2	commercial
P104	Smith	300 Spruce Rd	7-30-78	1.0	1	residential

Normalization

- Second normal form
  - separate entities: parcel, owner, address



Parcel table

PIN	Sale date	Acres	Zone code	Zoning
P101	1-10-98	1.0	1	Residential
P102	10-6-68	3.0	2	Commercial
P103	3-7-97	2.5	2	Commercial
P104	7-30-78	1.0	1	Residential

Owner table

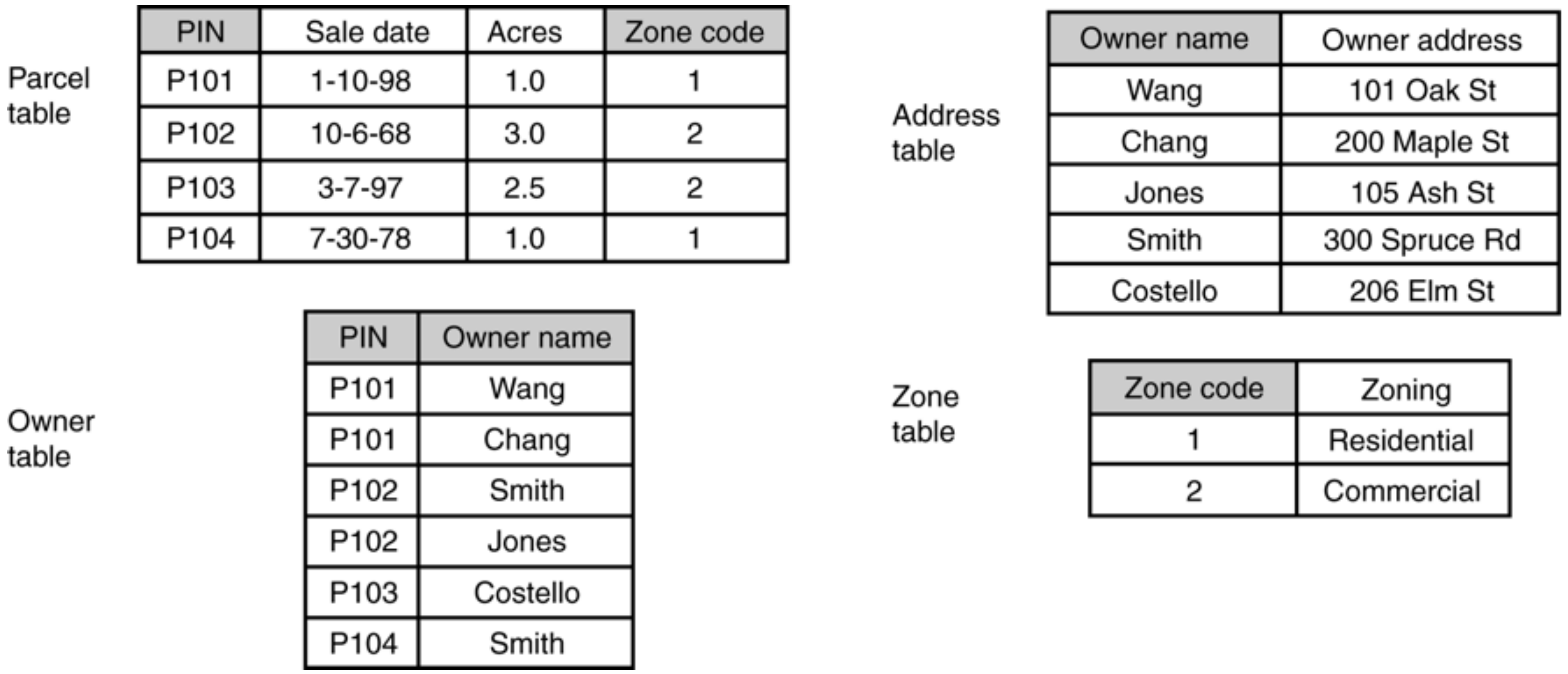
PIN	Owner name
P101	Wang
P101	Chang
P102	Smith
P102	Jones
P103	Costello
P104	Smith

Address table

Owner name	Owner address
Wang	101 Oak St
Chang	200 Maple St
Jones	105 Ash St
Smith	300 Spruce Rd
Costello	206 Elm St

Normalization

- Third normal form
  - – separate remaining non-key dependencies



Parcel table

PIN	Sale date	Acres	Zone code
P101	1-10-98	1.0	1
P102	10-6-68	3.0	2
P103	3-7-97	2.5	2
P104	7-30-78	1.0	1

Owner table

PIN	Owner name
P101	Wang
P101	Chang
P102	Smith
P102	Jones
P103	Costello
P104	Smith

Address table

Owner name	Owner address
Wang	101 Oak St
Chang	200 Maple St
Jones	105 Ash St
Smith	300 Spruce Rd
Costello	206 Elm St

Zone table

Zone code	Zoning
1	Residential
2	Commercial

Advantages of Databases over Files

- Avoids redundancy and duplication
- Reduces data maintenance costs
- Applications are separated from the data
  - Applications persist over time
  - Support multiple concurrent applications
- Better data sharing
- Security and standards can be defined and enforced

Disadvantages of Databases over Files

- Expense
- Complexity
- Performance – especially complex data types
- Integration with other systems can be difficult

References

- Chang, K.T., “Introduction to Geographic Information Systems, 5th ed.” ISBN 007729436X
- Date, C.J., “An Introduction to Database Systems, 7th ed.” ISBN 0201385902