

Relational databases

Problems they solve
Strengths and weaknesses

What should a database contain?

- No fixed criteria – largely a matter of preference
 - Generally, things that are included in a database are interrelated in some way
 - Including them together facilitates examination of the relationships between them
 - Can also contain reference information
- Could be:
 - All the data collected from a single experiment
 - All the data collected from a single study (multiple experiments)
 - All the DNA sequences for all the genes in the human genome
 - All the DNA sequences for proteins across all organisms

Many different data types can be held in a database

- Obviously, text and numbers
- Less obviously
 - DNA sequences
 - GIS data (points, lines, polygons)
 - Binary large objects (BLOBs)
 - Sound files
 - Images
 - Video
 - pdf files of journal articles

Database Management Systems (DBMS's)

- A DBMS is a program specifically designed to create, modify, and use a database
- DBMS's organize data in tables
 - Stacked data format
 - Each column is a variable, or **field**
 - Each row is a **record** = all known information about a single data point
- If all data are kept together in one big table, then the database is a **flat file** database
- If different tables are used, which can be joined together through one or more matching columns, then a **relational** database is used
- Although Excel can be used for a flat file database, it is a spreadsheet program, not a DBMS

Flat files

- Flat files are a stacked data arrangement, with all of the data in a single file
- See any problems here?

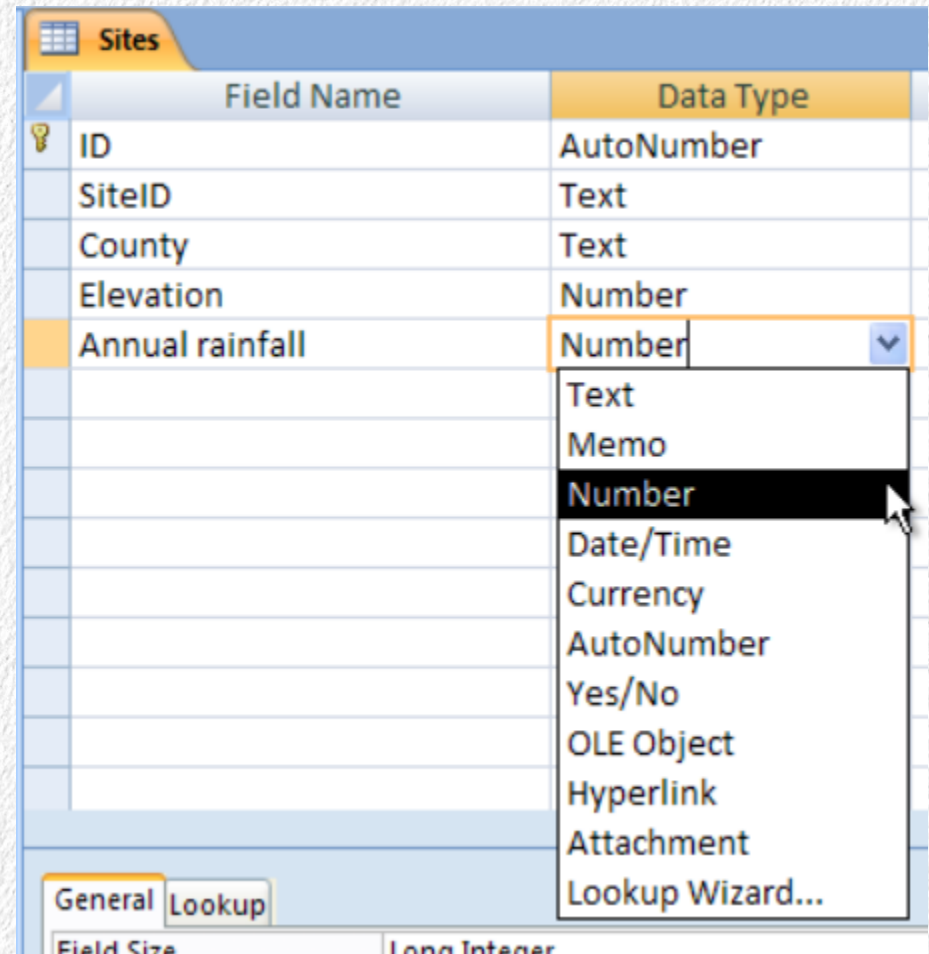
	A	B	C	D	E	F
1	Site ID	County	Elevation	Annual rainfall	Soil sample ID	Nitrogen
2	Sky Oaks	San Diego	1418	53	1	17.0
3	Sky Oaks	San Diego	1418	53	2	17.6
4	Sky Oaks	San Diego	1418	53	3	13.8
5	Sky Oaks	San Diego	1418	53	4	17.0
6	Sky Oaks	San Diego	1418	53	5	11.4
7	Sky Oaks	San Diego	1418	53	6	19.8
8	Sky Oaks	San Diego	1418	53	7	16.4
9	Sky Oaks	San Diego	1418	53	8	21.0
10	Sky Oaks	San Diego	1418	53	9	15.6
11	Sky Oaks	San Diego	1418	53	10	19.6
12	Sky Oaks	San Diego	1418	53	11	13.9
13	Santa Margarita	Riverside	338	36	1	19.0
14	Santa Margarita	Riverside	338	36	2	19.0
15	Santa Margarita	Riverside	338	36	3	16.1
16	Santa Margarita	Riverside	338	36	4	22.2
17	Santa Margarita	Riverside	338	36	5	22.4
18	Santa Margarita	Riverside	338	36	6	16.5
19	Santa Margarita	Riverside	338	36	7	17.2
20	Santa Margarita	Riverside	338	36	8	13.4
21	Santa Margarita	Riverside	338	36	9	21.0
22	Santa Margarita	Riverside	338	36	10	16.3
23	Santa Margarita	Riverside	338	36	11	16.8

Advantages of a flat file database over Excel

- DBMS's are less flexible than spreadsheets, which is good for database management
 - Fields have to be assigned a data type – helps prevent data entry errors, saves space
 - Additionally, can limit the data values that can be entered
 - The basic unit is the record, not the cell – can't accidentally sort one field but not the others
- Data management benefits from these restrictions:
 - Easier to set up data entry forms that prevent entry errors
 - Large data sets can be stored with smaller file sizes
 - Searching, filtering, subsetting data is easier, faster

Variable types in Access

- Variables in databases are called “fields”
- Fields have a specific data type assigned to them
- Only data of the appropriate type can be entered in the field



Problems with variable types in spreadsheets

Sample ID	Nitrogen	Nitrogen (mg)	Nitrogen (mg)
1	10 mg	10	10 mg
2	12 mg	12	12
3	9 mg	9	9
Average	#DIV/0!	10.33	10.50

One text cell included with two numeric cells

Mistake, not obvious

Obvious mistake

Including a text label makes this cell text

- Spreadsheets do treat different variable types differently
- But, spreadsheets do not enforce a variable type on a column
- Mixing different variable types is allowed
- Calculations may not be accurate if they are mixed

Field type: definitions

Field type	Definition	Additional properties
Text	Combinations of text or numbers, but treated as non-numeric	Number of characters (up to 255)
Memo	Combinations of text or numbers, but treated as non-numeric	Up to 63,999 characters
Number	Numbers only (no text characters)	Number type (integer, long integer, single, double)
Date/time	Dates and times	Format
Currency	Money	
Autonumber	Numbers that automatically increase with each new record added	
Yes/No	Only contains yes or no (or equivalently, 1 or 0)	
OLE object	Any file type supported, can be linked or embedded in database	
Hyperlink	Link to a URL	
Attachment	External files that are stored in the database	

Relational databases

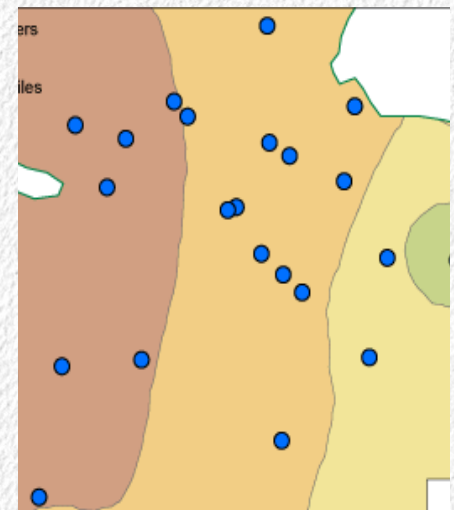
- Relational Database Management Systems (RDBMS) organize data into two or more tables that can be joined based on one or more matching field
- Matching two or more tables based on columns of matching data is a **relational join**
 - Excel (spreadsheets) cannot do these
- Relationships can be:
 - One to one = one row in table 1 matches one row in table 2
 - One to many = one row in table 1 matches many rows in table 2
- Example: soil nitrogen measurements

One to one relationships

- Example – different measurements of the same soil sample made at different times, different locations
- Field measurements: location, slope, aspect
- Chemical characteristics of soil samples are measured in the lab
- Time since last fire is taken from a computer mapping program (a Geographic Information System, or GIS)
- We will have three different tables, one for each type of data – need to join them together

Point ID	Lat.	Long.	Slope	Aspect
1	33 ° 24'	116 ° 18'	11	14 °
2	33 ° 16'	116 ° 18'	8	127 °
3	33 ° 15'	116 ° 15'	11	99 °
4	33 ° 18'	116 ° 22'	11	168 °
5	33 ° 27'	116 ° 19'	7	249 °
6	33 ° 7'	116 ° 21'	9	210 °
7	33 ° 23'	116 ° 19'	14	219 °
8	33 ° 31'	116 ° 22'	10	234 °

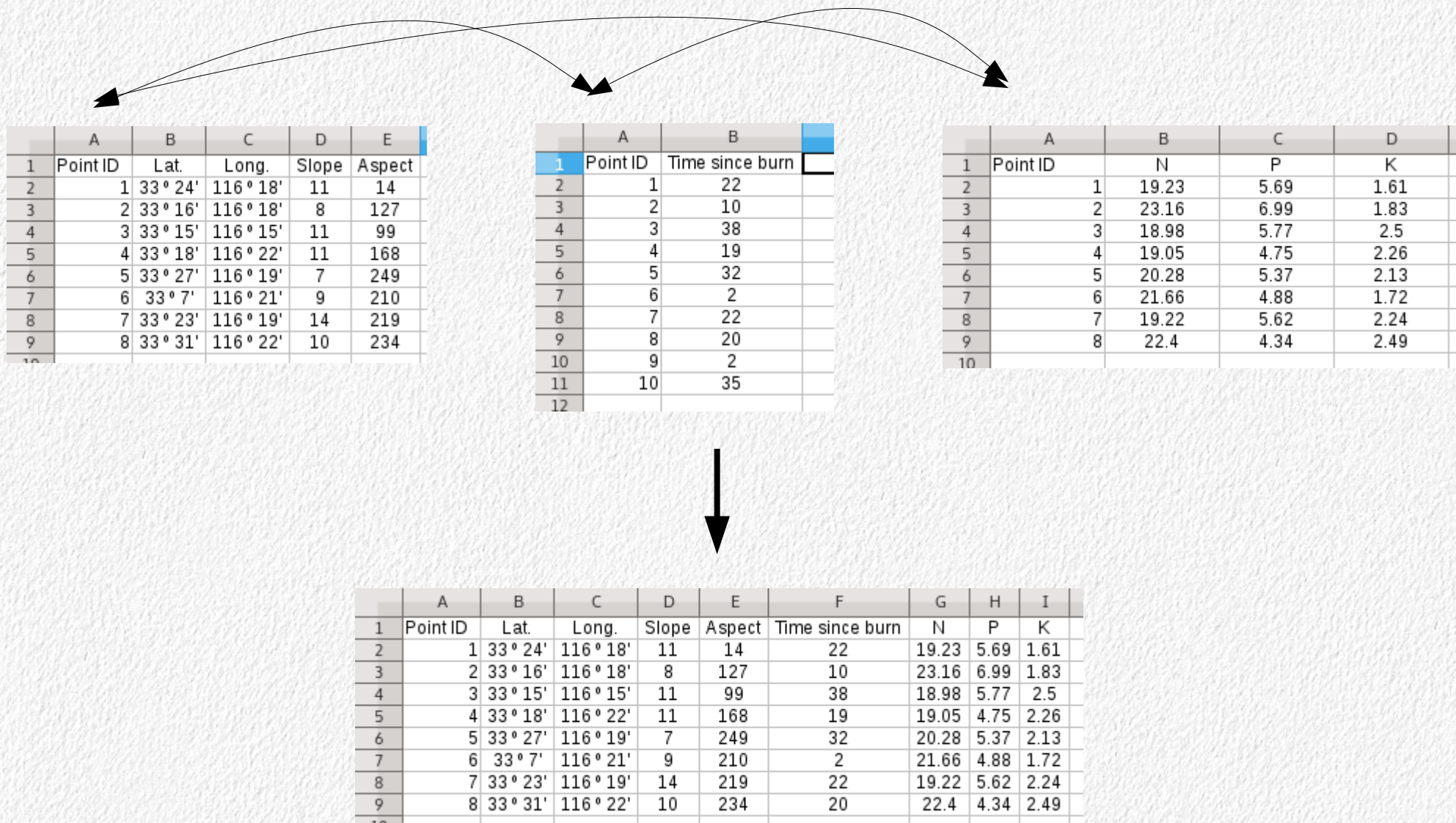
Map of time since burn



Soil extraction in the lab



Tables joined by matching columns



One to many relationship

- One table with site-level information, one with sample-level information

Site ID	County	Elevation	Annual rainfall
Sky Oaks	San Diego	1418	53
Santa Margarita	Riverside	338	36

- Joined together by one or more common fields
- The match between fields of different tables defines the relationship between them

Site ID	Soil sample	Nitrogen
Sky Oaks	1	17.0
Sky Oaks	2	17.6
Sky Oaks	3	13.8
Sky Oaks	4	17.0
Sky Oaks	5	11.4
Sky Oaks	6	19.8
Sky Oaks	7	16.4
Sky Oaks	8	21.0
Sky Oaks	9	15.6
Sky Oaks	10	19.6
Sky Oaks	11	13.9
Santa Mar	1	19.0
Santa Mar	2	19.0
Santa Mar	3	16.1
Santa Mar	4	22.2
Santa Mar	5	22.4
Santa Mar	6	16.5
Santa Mar	7	17.2
Santa Mar	8	13.4
Santa Mar	9	21.0
Santa Mar	10	16.3
Santa Mar	11	16.8

Joined

	A	B	C	D	E	F
1	Site ID	County	Elevation	Annual rainfall	Soil sample ID	Nitrogen
2	Sky Oaks	San Diego	1418	53	1	17.0
3	Sky Oaks	San Diego	1418	53	2	17.6
4	Sky Oaks	San Diego	1418	53	3	13.8
5	Sky Oaks	San Diego	1418	53	4	17.0
6	Sky Oaks	San Diego	1418	53	5	11.4
7	Sky Oaks	San Diego	1418	53	6	19.8
8	Sky Oaks	San Diego	1418	53	7	16.4
9	Sky Oaks	San Diego	1418	53	8	21.0
10	Sky Oaks	San Diego	1418	53	9	15.6
11	Sky Oaks	San Diego	1418	53	10	19.6
12	Sky Oaks	San Diego	1418	53	11	13.9
13	Santa Margarita	Riverside	338	36	1	19.0
14	Santa Margarita	Riverside	338	36	2	19.0
15	Santa Margarita	Riverside	338	36	3	16.1
16	Santa Margarita	Riverside	338	36	4	22.2
17	Santa Margarita	Riverside	338	36	5	22.4
18	Santa Margarita	Riverside	338	36	6	16.5
19	Santa Margarita	Riverside	338	36	7	17.2
20	Santa Margarita	Riverside	338	36	8	13.4
21	Santa Margarita	Riverside	338	36	9	21.0
22	Santa Margarita	Riverside	338	36	10	16.3
23	Santa Margarita	Riverside	338	36	11	16.8

Common types of RDBMS you may encounter

- Desktop software
 - MS Access, FileMaker
 - Meant to be used by one person at a time
 - These RDBMS's keep databases in single files that can be copied and moved around easily
- Structured Query Language distributed systems
 - MS SQL Server, MySQL, Postgresql, Oracle
 - Run from a central server over a network
 - Data stored centrally – changes are available to all users at once
 - Usually used for large, shared databases, web applications

Some problems in data entry and management addressed by RDBMS

- Data accuracy problems
- Reducing redundancy in data entry
- Organization

Accuracy – avoiding data entry errors

- Less need for redundant entries
- Use explicit variable types
- Pick lists, data entry forms
- Avoiding leaving blanks

Redundancy in data entry

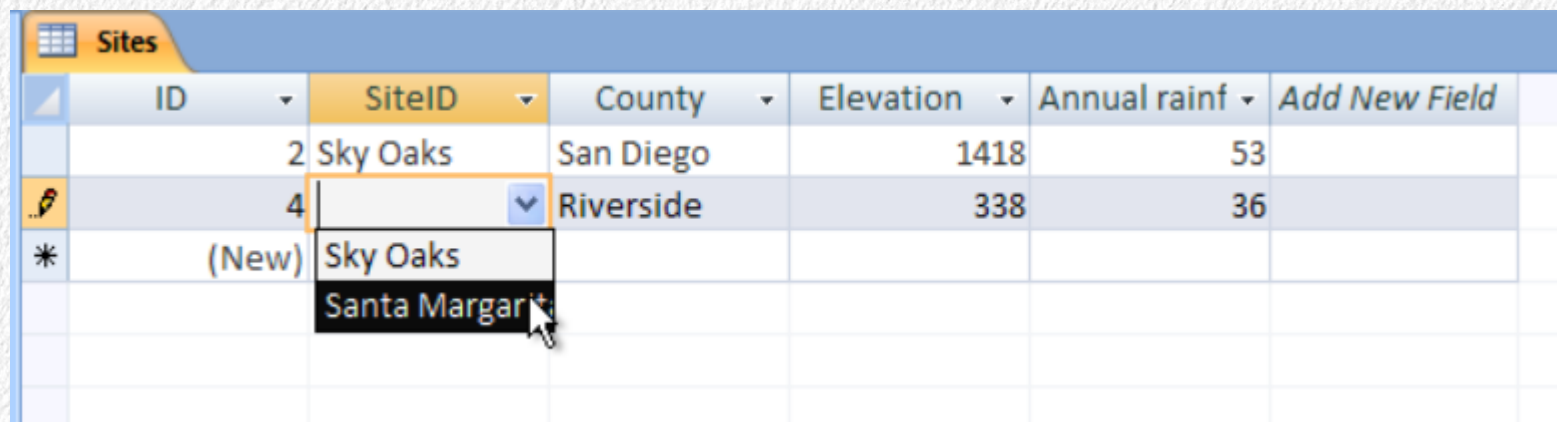
- Redundancy is bad
 - More opportunities for data entry error
 - Larger file sizes for no good reason
- We can avoid redundancy with a well-designed RDB
- Clearly true for one to many relationships – only need to enter values once rather than repeatedly
- Also true for one to one relationships, if there are several tables

Avoiding typos


- Typographical errors (typos) are data entry errors
 - Hit the wrong key
 - Used different labels for the same thing
- RDBMS's help you avoid them
 - Variable types
 - Lookups

Lookups

- Computers are literal – Sky Oaks, Sky Oaks , Sky Oaks, Sky Oakes, SkyOaks, and Sky oaks are all different
- RDBMS's allow you to limit entry to particular values
- Can be selected from a drop-down list
- No typos – only error possible is to select the wrong value from the list



The screenshot shows a database application window titled 'Sites'. It contains a table with the following columns: ID, SiteID, County, Elevation, Annual rainf, and Add New Field. The table has three rows of data. The first row has ID 2, SiteID 'Sky Oaks', County 'San Diego', Elevation 1418, and Annual rainf 53. The second row has ID 4, SiteID (with a dropdown arrow), County 'Riverside', Elevation 338, and Annual rainf 36. The third row is a new entry with ID '(New)', SiteID 'Sky Oaks', and County 'Santa Margarita'. A mouse cursor is hovering over the 'Santa Margarita' option in the dropdown menu.

ID	SiteID	County	Elevation	Annual rainf	Add New Field
2	Sky Oaks	San Diego	1418	53	
4		Riverside	338	36	
(New)	Sky Oaks				
	Santa Margarita				

Enforcing completeness in data entry

- Missing information can be a big problem
 - Can cause the entire record to be unusable
- Can tell the RDBMS to:
 - Automatically date/time stamp entries
 - Require fields to be filled in before a record is accepted
- Data forms make it easier for data entry to be done consistently
- Data forms can be designed to look the same on the screen as they look on paper – less likely to miss an entry

Data entry forms

- A single record entered at a time
- Joined tables can both be displayed
- This example shows a form for sites with a subform for samples
- All samples that match the site ID are displayed in the subform
- More samples can be added to a site
- A new site can be added, then samples added for that site

The screenshot shows a data entry form titled "Sites" with a subtitle "Combined information about sites and samples". The form contains several input fields for site information:

- ID: 2
- SiteID: Sky Oaks (dropdown menu)
- County: San Diego (dropdown menu)
- Elevation: 1418
- Annual rainfall: 53

Below these fields is a subform titled "SoilSamples subform" which displays a table of soil samples for the selected site (Sky Oaks). The table has three columns: Soil sample ID, Site ID, and Nitrogen. It shows five records with sample IDs 1 through 5, all for Sky Oaks, with nitrogen levels of 17.0, 17.6, 13.8, 17.0, and 11.4 respectively. The subform includes a scroll bar on the right and a status bar at the bottom showing "Record: 1 of 13", "No Filter", and a "Search" button.

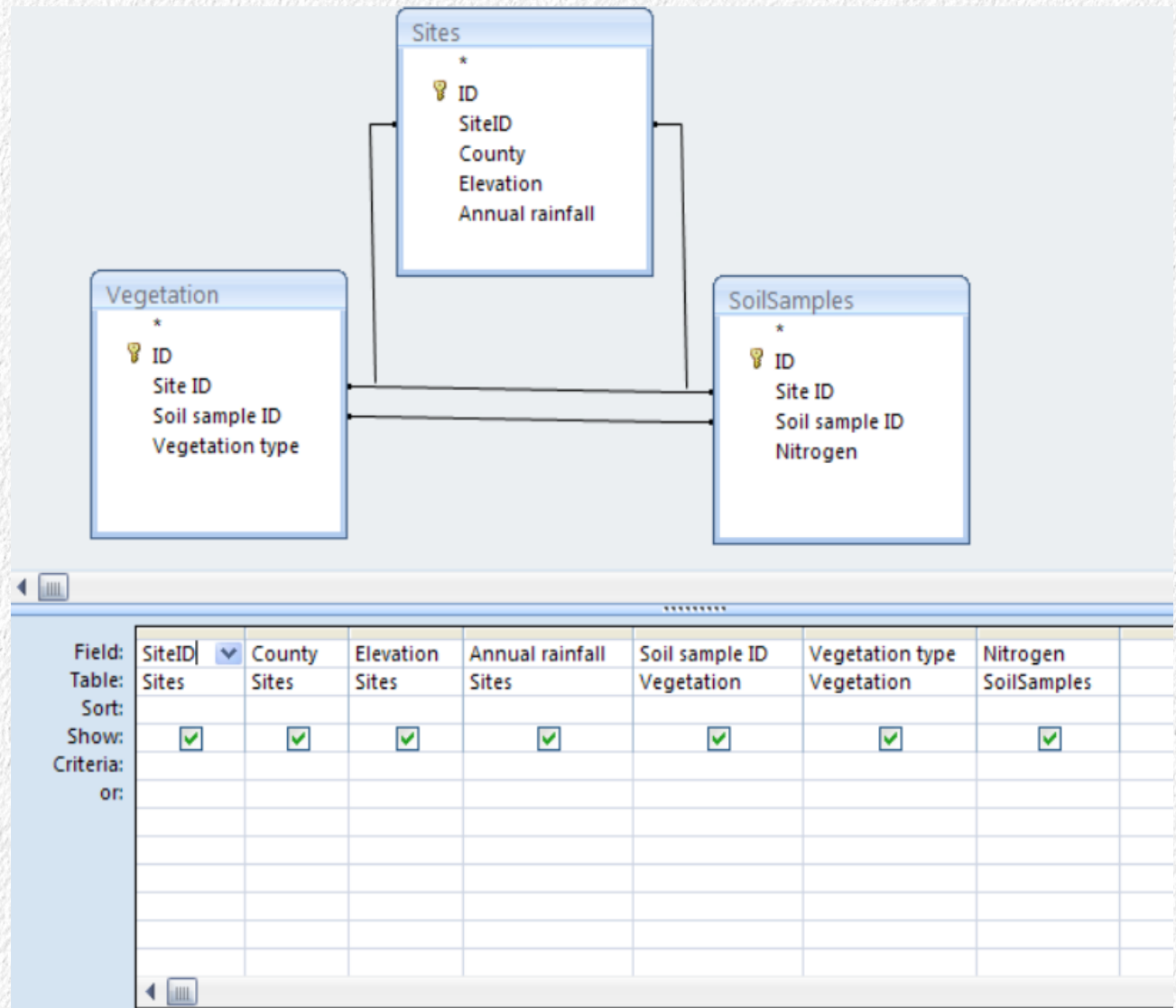
Soil sample ID	Site ID	Nitrogen
1	Sky Oaks	17.0
2	Sky Oaks	17.6
3	Sky Oaks	13.8
4	Sky Oaks	17.0
5	Sky Oaks	11.4

Keeping data organized

- Subsets of data records can be extracted for use with “queries”
- Once constructed, queries can be saved for later use
- Changes in a data table are automatically reflected in all queries that use the table

Example of a query – joining tables together and displaying the result

- The “Design View” for the query – define relationships, select the variables to display
- Join Sites to Vegetation and SoilSamples based on matching SiteID's
- Join Vegetation and Soil Samples based on matching SiteID and Soil sample ID's (must be unique!)
- Select variables from any of the three tables to display



Executed query – datasheet view

Query1							
SiteID	County	Elevation	Annual rainf	Soil sample	Vegetation 1	Nitrogen	
Sky Oaks	San Diego	1418	53	1	CSS	17.0	
Sky Oaks	San Diego	1418	53	2	Chaparral	17.6	
Sky Oaks	San Diego	1418	53	3	Riparian	13.8	
Sky Oaks	San Diego	1418	53	4	Riparian	17.0	
Sky Oaks	San Diego	1418	53	5	CSS	11.4	
Sky Oaks	San Diego	1418	53	6	CSS	19.8	
Sky Oaks	San Diego	1418	53	7	Chaparral	16.4	
Sky Oaks	San Diego	1418	53	8	Chaparral	21.0	
Sky Oaks	San Diego	1418	53	9	Chaparral	15.6	
Sky Oaks	San Diego	1418	53	10	Riparian	19.6	
Sky Oaks	San Diego	1418	53	11	Riparian	13.9	
Santa Margarita	Riverside	338	36	1	Riparian	19.0	
Santa Margarita	Riverside	338	36	2	Chaparral	19.0	
Santa Margarita	Riverside	338	36	3	Chaparral	16.1	
Santa Margarita	Riverside	338	36	4	CSS	22.2	
Santa Margarita	Riverside	338	36	5	Chaparral	22.4	
Santa Margarita	Riverside	338	36	6	CSS	16.5	
Santa Margarita	Riverside	338	36	7	Chaparral	17.2	
Santa Margarita	Riverside	338	36	8	Chaparral	13.4	
Santa Margarita	Riverside	338	36	9	CSS	21.0	
Santa Margarita	Riverside	338	36	10	CSS	16.3	
Santa Margarita	Riverside	338	36	11	CSS	16.8	

Data in a query can be edited to update the underlying tables if...

- There is a one to one relationship between the tables
- Only one or two tables are used
- If there are one to many relationships, or three or more tables joined, it won't be possible to edit the data in the query

Query to show only Sky Oaks data

Field:	SiteID	County
Table:	Sites	Sites
Sort:		
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:	"Sky Oaks"	
or:		

CombinedSiteSoilVeg							
SiteID	County	Elevation	Annual rainf	Soil sample	Vegetation 1	Nitrogen	
Sky Oaks	San Diego	1418	53	1	CSS	17.0	
Sky Oaks	San Diego	1418	53	2	Chaparral	17.6	
Sky Oaks	San Diego	1418	53	3	Riparian	13.8	
Sky Oaks	San Diego	1418	53	4	Riparian	17.0	
Sky Oaks	San Diego	1418	53	5	CSS	11.4	
Sky Oaks	San Diego	1418	53	6	CSS	19.8	
Sky Oaks	San Diego	1418	53	7	Chaparral	16.4	
Sky Oaks	San Diego	1418	53	8	Chaparral	21.0	
Sky Oaks	San Diego	1418	53	9	Chaparral	15.6	
Sky Oaks	San Diego	1418	53	10	Riparian	19.6	
Sky Oaks	San Diego	1418	53	11	Riparian	13.9	

Query to show Sky Oaks, with nitrogen levels over 17

Field:	Vegetation type	Nitrogen
Table:	Vegetation	SoilSamples
Sort:		
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:		>17
or:		

CombinedSiteSoilVeg							
SiteID	County	Elevation	Annual rainf	Soil sample	Vegetation t	Nitrogen	
Sky Oaks	San Diego	1418	53	1	CSS	17.0	
Sky Oaks	San Diego	1418	53	2	Chaparral	17.6	
Sky Oaks	San Diego	1418	53	4	Riparian	17.0	
Sky Oaks	San Diego	1418	53	6	CSS	19.8	
Sky Oaks	San Diego	1418	53	8	Chaparral	21.0	
Sky Oaks	San Diego	1418	53	10	Riparian	19.6	

Why don't we use RDBMS's all the time?

- Complexity, unfamiliarity, inflexibility
- Excel, the Swiss Army Knife, does well enough most of the time, allows us to do some things better or more simply
 - Attractive layout of data tables
 - More graphing options
 - Complex calculations stored within the spreadsheet
- Be aware of the uses of relational databases, in case you have an application for which Excel will not suffice
- Data can be moved back and forth between spreadsheets and databases – in combination they can be very powerful