

# Post Collection Processing of Survey Data

Wayne Enanoria, PhD, MPH  
Public Health Epidemiologist

Center for Infectious Disease Preparedness  
UC Berkeley School of Public Health  
Email: [enanoria@berkeley.edu](mailto:enanoria@berkeley.edu)



# Outline

- ◆ Steps of Processing Data:
  - ◆ Defining fields
  - ◆ Creating a database
  - ◆ Format and range of permissible values
  - ◆ Creating a data dictionary
  - ◆ Coding
  - ◆ Data entry
  - ◆ Creating a dataset for analysis
  - ◆ Backing up and archiving the dataset



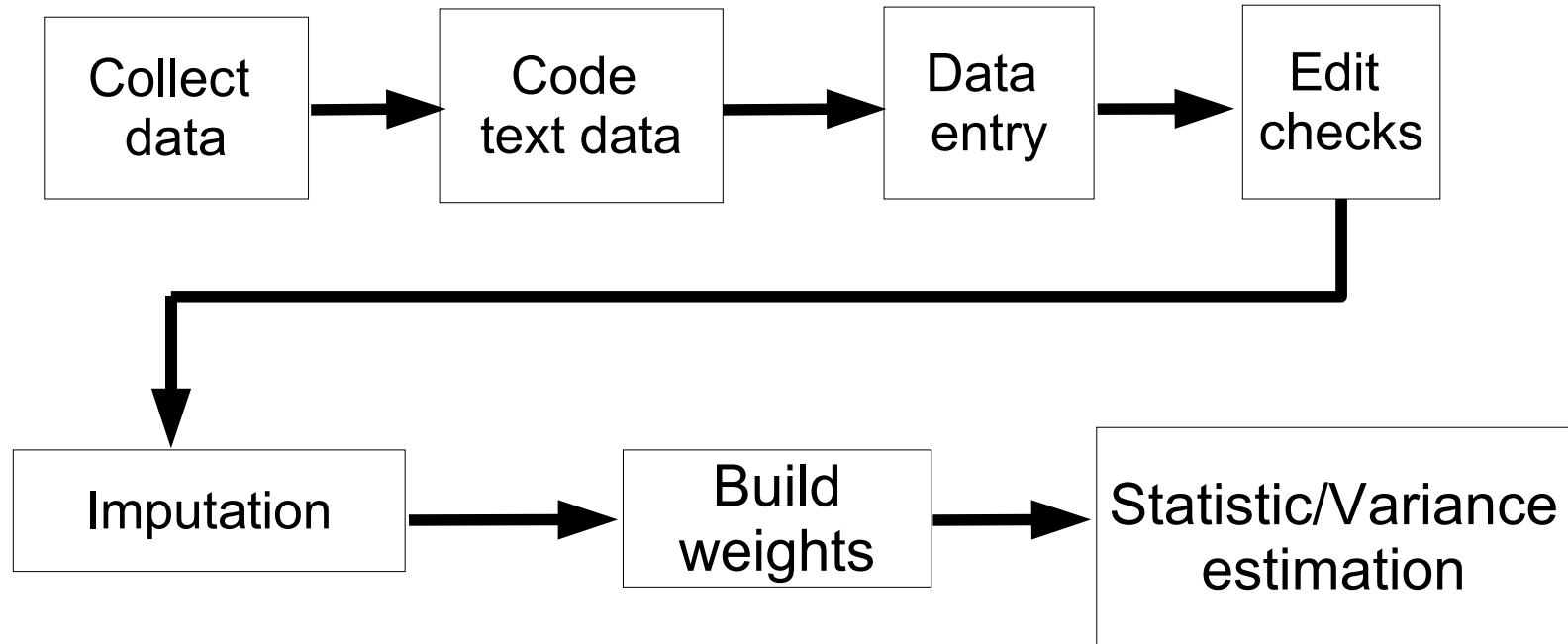


ALERT: this lecture contains my  
biases for data management!



# Data Processing “Road Map”

Adapted from Groves et al. *Survey Methodology* 2004



# Defining New Variables

- ◆ Each variable should be identified and given a name.
- ◆ The name will be used to identify variables in the database and during the analysis.
- ◆ Ideal features of a name:
  - ◆ Easily identifies the question on the form (if one is used) or type of information collected
  - ◆ Understandable, consistent and short (some software programs only allow 8 characters!)





#### 4.0 Laboratory Evaluation

Choose from the following specimens to enter for each test: whole blood, serum (acute), serum (convalescent), NP swab, NP aspirate, bronchoalveolar lavage specimen, OP swab, urine, stool, or tissue.

LABORATORY  
↓

Specimen	Date Collected (mm/dd/yyyy)	Test Requested	Result
SPECIMEN	DATE COLLECT	TEST REQUEST	RESULT



# Alternative Example: Laboratory Data

idnum	specimen0	specimen1	specimen2	specimen3	specimen4
1	Blood	Sputum	Blood	Blood	Blood
2	Blood	Sputum	Blood		
3	Sputum	Blood	Blood	Blood	Blood
4	Blood	Blood			
5	Sputum				



# Laboratory Data as One Table

idnum	specimen
1	Blood
1	Sputum
1	Blood
1	Blood
1	Blood
2	Blood
2	Sputum
2	Blood
3	Sputum
3	Blood
3	Blood
3	Blood
3	Blood
4	Blood
4	Blood
5	Sputum



# Creating a Database

- ♦ A database turns disparate pieces of data into information.
- ♦ Before a database is created, be sure to have a clear goal and purpose for its existence.
  - ♦ Who will use it? What are their needs?
  - ♦ What will the data be used for?



# Creating a Database con't

- ♦ Sometimes it helps to start with the end product in mind:
  - ♦ What do I want to say with this data?
  - ♦ What “form” must the data be in for the analyses?



# Types of Databases in Public Health

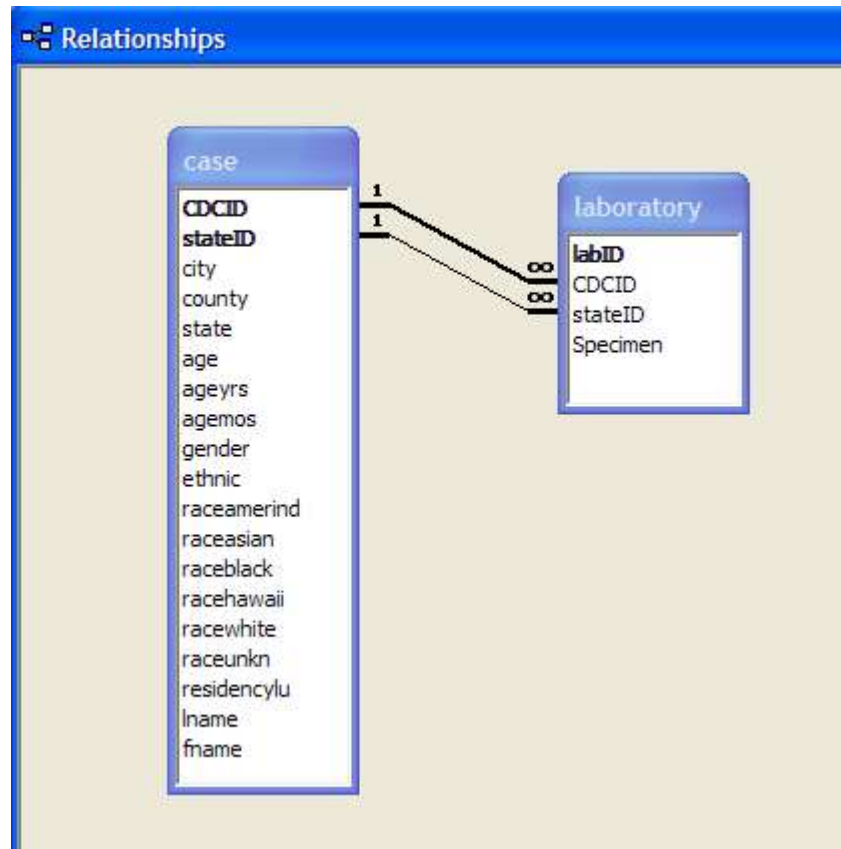
- ♦ There are two main types of databases used in public health:
  - ♦ flat-file database (spreadsheet-like type)
  - ♦ relational database



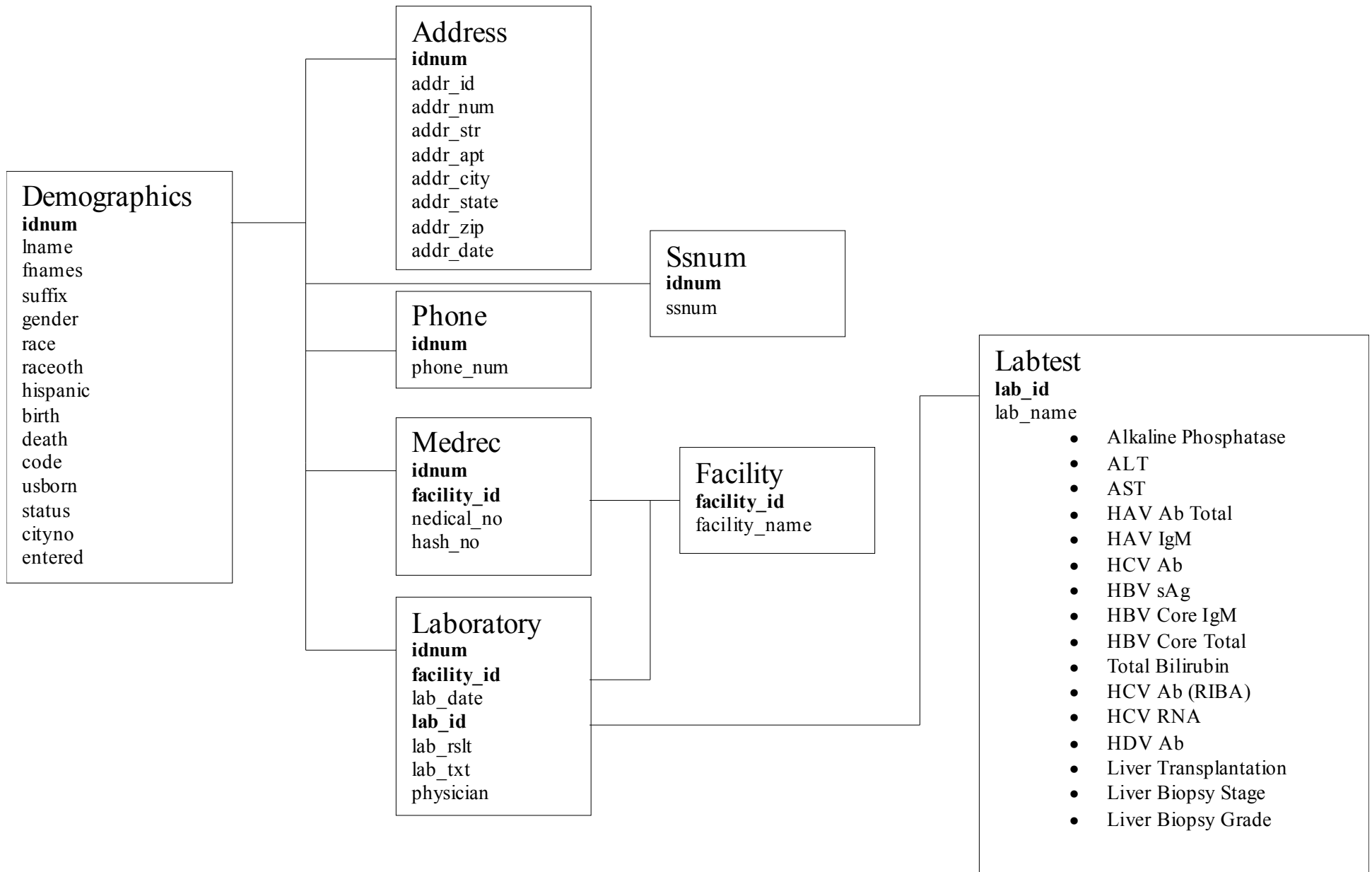
# What is a relational database?

- ♦ A relational database stores data in “relations”.
- ♦ Each relation (table) is composed of records (rows) and fields (columns).
- ♦ Software for building relational databases include:
  - ♦ Epi Info version 3.3
  - ♦ MS Access
  - ♦ MySQL
  - ♦ Oracle





# Database Structure



# Flat File vs. Relational Database?

- ♦ Choice depends on the information you are collecting.
  - ♦ Does the information contain complicated relationships?
  - ♦ If yes, what is the most efficient way to store these relationships (minimize redundancy, data entry time, chance for error, etc.)?
- ♦ A database should only be as complicated as it needs to be.



*“Everything should be as simple  
as possible, but not simpler.”*

- Albert Einstein



# Creating a Data Dictionary

- ♦ A data dictionary is a “codebook” to understand the meaning of the collected data.
- ♦ For each field, the data dictionary should include:
  - ♦ type (eg, continuous, integer, text, other)
  - ♦ format (eg, “Yes”, “No”, “Missing”)
  - ♦ permissible values (eg, date field can only include dates after January 1, 2005, or a response coded as 0, 1, or 99)



# Data Validation and Checks

- ♦ Some answers should have logical relationships to others.
  - ♦ For example, those reported as “male” should not report that they are pregnant.
- ♦ Most data management systems can perform edit checks to validate your data as it is being entered – set this up a priori.
- ♦ Most systems also allow one to control the range of permissible values that can be entered into a field.



# Data Entry Options to Consider

- ♦ Required fields
- ♦ Legal values
- ♦ Range checks
- ♦ Repeat fields
- ♦ Conditional jumps
- ♦ Programmed checks



# Controlling Data Range

- ♦ For example:
  - “Have you ever smoked?”
    - 1=“Yes”
    - 2=“No”
    - 9=“Don't Know”
- ♦ The database can be set up to allow only the values 1, 2, and 9 to be entered into the field “eversmok”.



# Implementation

- ♦ MS Access can create drop-down menus to control range of values.
  - ♦ Use the Lookup Wizard when specifying data type in the creation of a table.
- ♦ Epi Info version 3.3 also has this capability.
  - ♦ Create a field using “legal value” or “comment legal value” options.
- ♦ MySQL
  - ♦ ENUM option



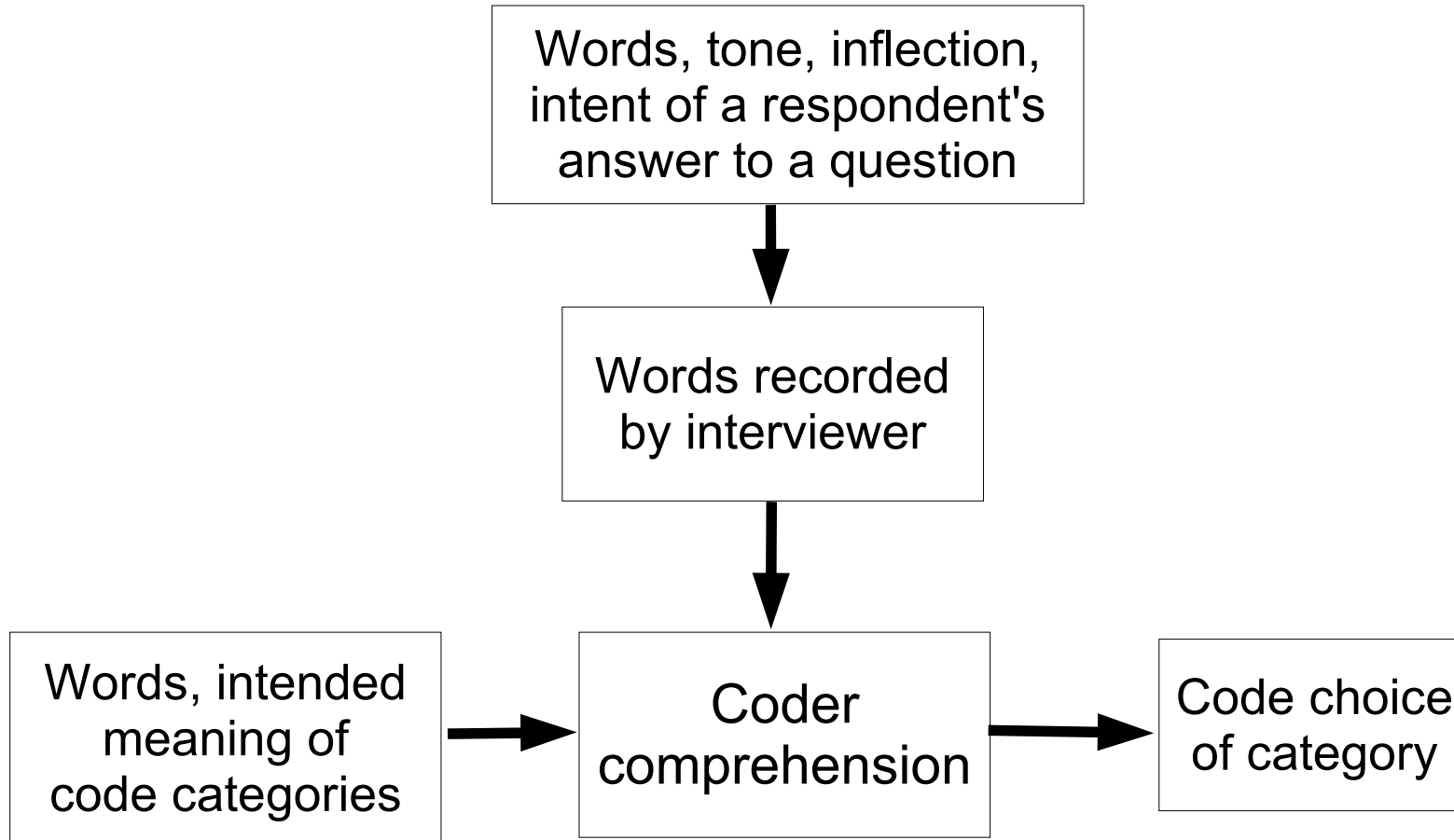
# Coding

- ♦ Most statistical routines require that non-numeric information be coded into numeric answers.
- ♦ Coding is both an act of translation and an act of summarization.
- ♦ These numbers then become the values in a field of the electronic data file eventually produced.



# Comprehension and Judgement Task of the Coder

Adapted from Groves et al. *Survey Methodology* 2004



# Coding Example

- ♦ For example:
  - “Have you ever smoked?”
  - 1=“Yes”
  - 2=“No”
  - 9=“Don't Know”



# Coding Open-Ended Questions

- ♦ Coding answers to open-ended questions can be less straight forward.
- ♦ For example:  
“What is your occupation?”



# Coding Missing Data

- ♦ Missing data should be assigned a value that is not a possible numeric value.
  - ♦ Example: coding missing data with “99”. Missing values for age will be analyzed as age=99 years.
- ♦ It is best to code “Don't Know” differently from missing data.
  - ♦ In the analysis, all these responses are treated as missing, but the reason the data is missing is retained.



# Useful Attributes of Codes

Source: Groves RM et al. *Survey Methodology*, 2004, Wiley Series.

- A unique number, used later for statistical computing.
- A text label to describe all answers assigned to the category.
- All responses should be able to be assigned to a category.
- Mutual exclusivity; no single answer should be assignable to more than one category.
- A number of unique categories that fit the purposes of the analyst.



## Coding Example (2)

- ◆ For example:  
“In your opinion, what is the worst part about being homeless?”



# Coding Example (2) con't

- ♦ Answers:

Weather-Getting wet

Not enough resources

Not being secure, Uncertain about the future

Dealing with people that have no desire to change

Personal appearance, not being able to support yourself

Having issues with other people

Not having money to pay rent, no job

Refused

Uncertain when you will wake up, no where to wash

Being more concerned about others

Discrimination

Not being able to jump right to the bathroom or shower

Not having shelter, the weather



## Coding Example (2) con't

- ♦ Some categories for coding this example could be:
  - ♦ Lack of shelter (physically being homeless)
  - ♦ Emotionally being homeless (feelings of instability)
  - ♦ Problems with others
  - ♦ No resources
  - ♦ Hygiene/Cleaning issues
  - ♦ Refused



# Coding Example (2) con't

- ♦ Answers:

Knowing you can do better

Not being able to shower, Lacking a permanent address

Unpredictability

Not being able to entertain friends and take baths

Not being able to cook, trying to stay clean

Not having what you need to live with

Inconvenience involved with it

Police harassment

Refused

It puts you into a category that you don't feel good about yourself

The instability, not having any place to store your things

Having to deal with other homeless people

Finding somewhere to lay your head

No safe place to stay, weather

Refused

Being viewed as homeless

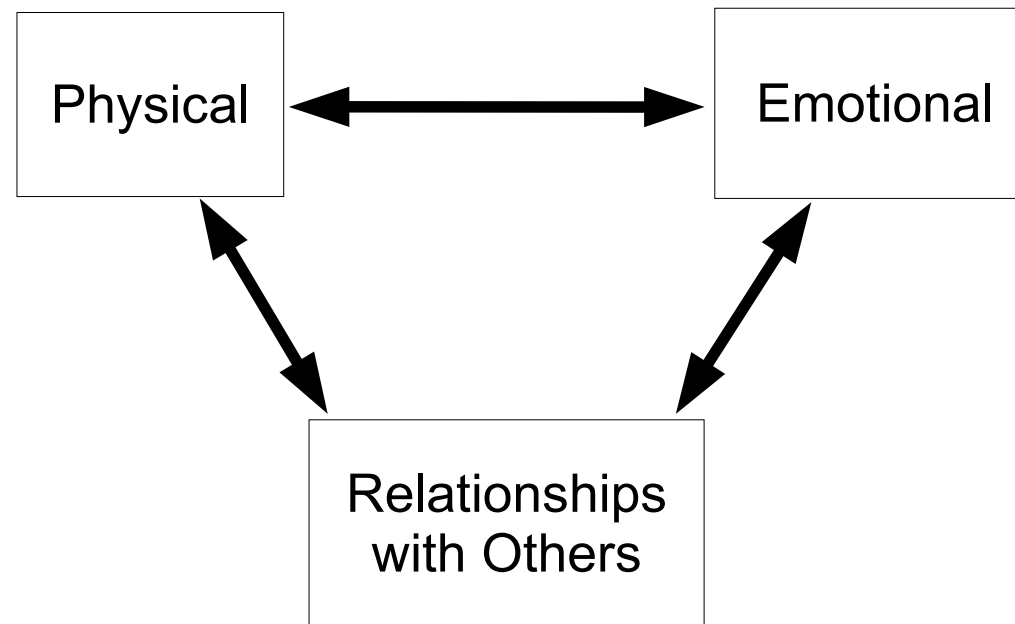


## Coding Example (2) con't

- ♦ Revision of categories for coding:
  - ♦ Lack of shelter (physically being homeless)
  - ♦ Emotionally being homeless (feelings of instability)
  - ♦ Problems with others (safety)
  - ♦ No resources
  - ♦ Hygiene/Cleaning issues
  - ♦ Refused
  - ♦ Low self-esteem
  - ♦ No means to cook



# Concepts from Responses

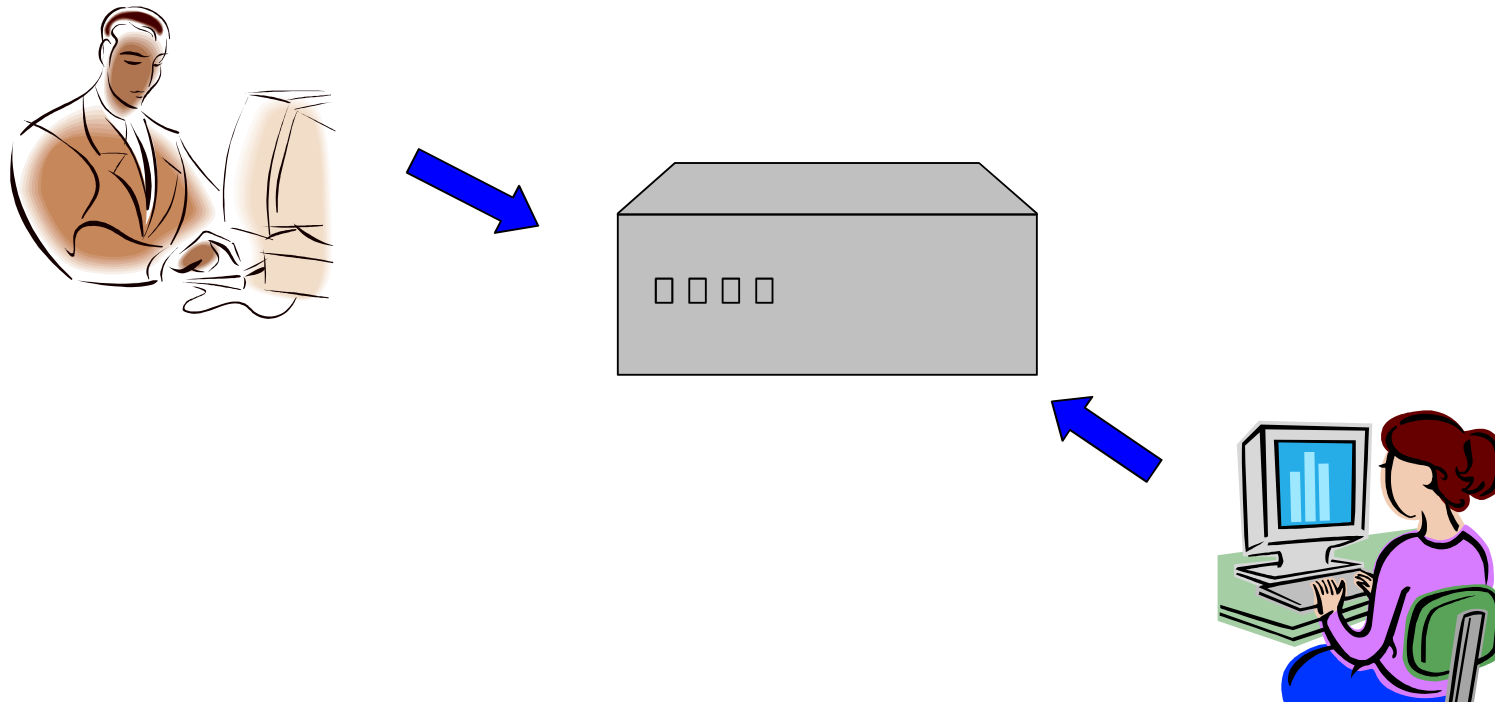


# Data Entry

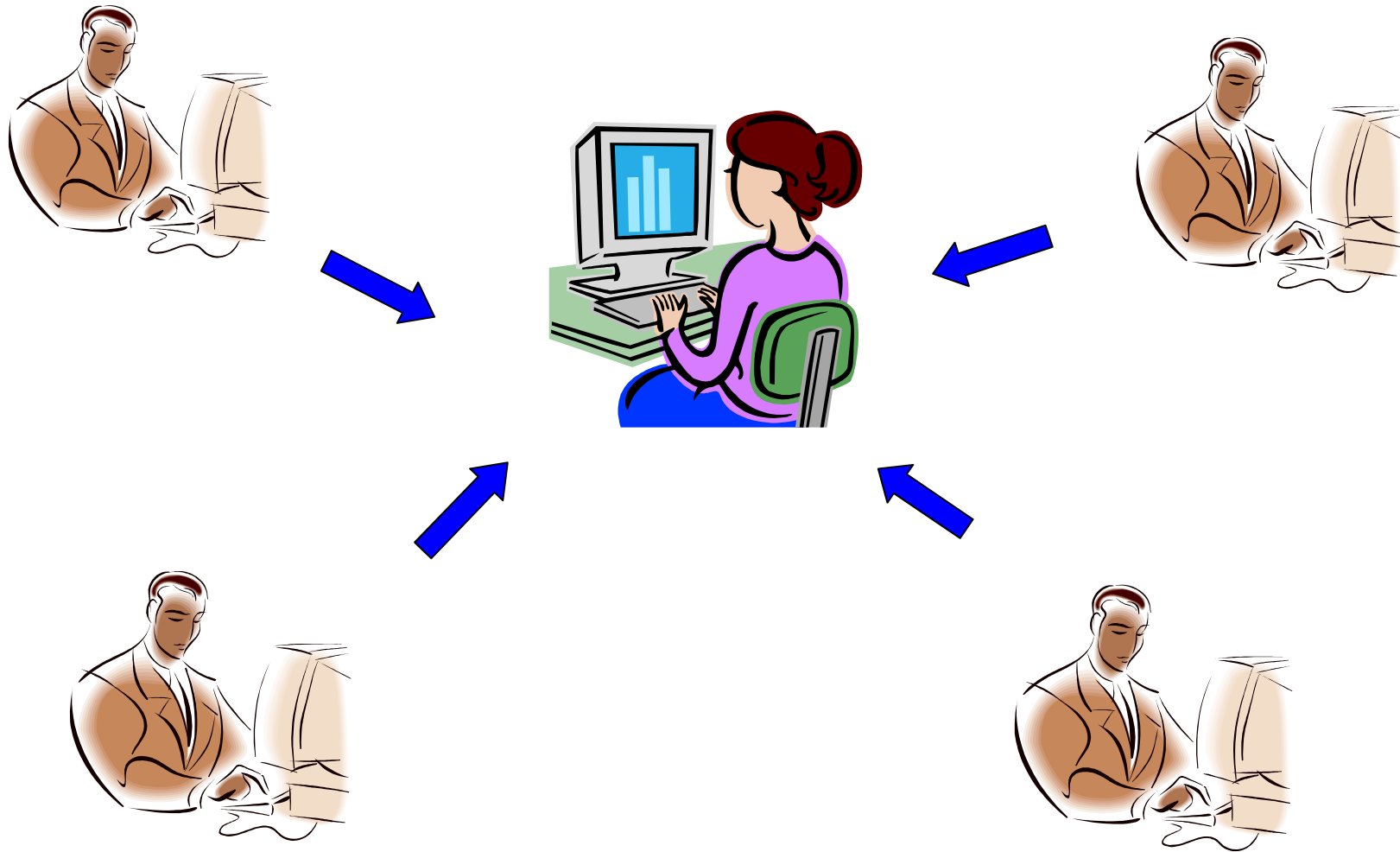
- ♦ Goal: enter the data efficiently and accurately into the database.
- ♦ Minimize data entry time by instituting proper tab orders, hot key short cuts, proper “look and feel” of the data entry screen.



# Data Entry Options: One Database



# Data Entry Options: Multiple Databases



# Advantage of One Database

- ♦ You can quickly run summary statistics and interim analyses of the data as it is being entered.
- ♦ All of the data is in one location; you can easily check for duplication of data entry.
- ♦ Useful if data enterers are in multiple locations (e.g., different floors, offices buildings, etc.) but assumes that everyone has access to the same database/server.



# Advantage of Multiple Databases

- ♦ If the database gets corrupted or unreadable, you only have to re-enter that person's records!
- ♦ Useful if the data entry staff has varied levels of computer skills or knowledge of data entry.
- ♦ You will not lose ALL of your data (just some)!
- ♦ Data can be appended in the analysis phase of the project.

Remember to keep track of who is entering which records into the database!

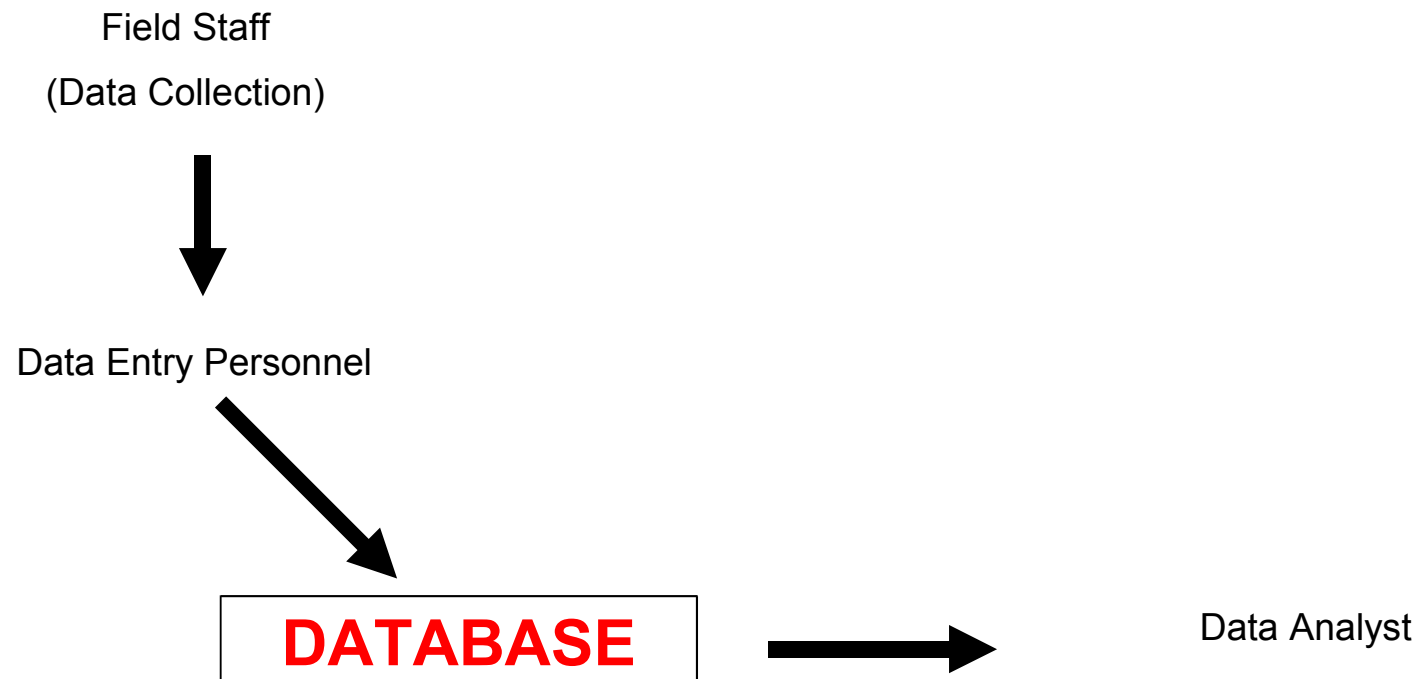


# Staff Training

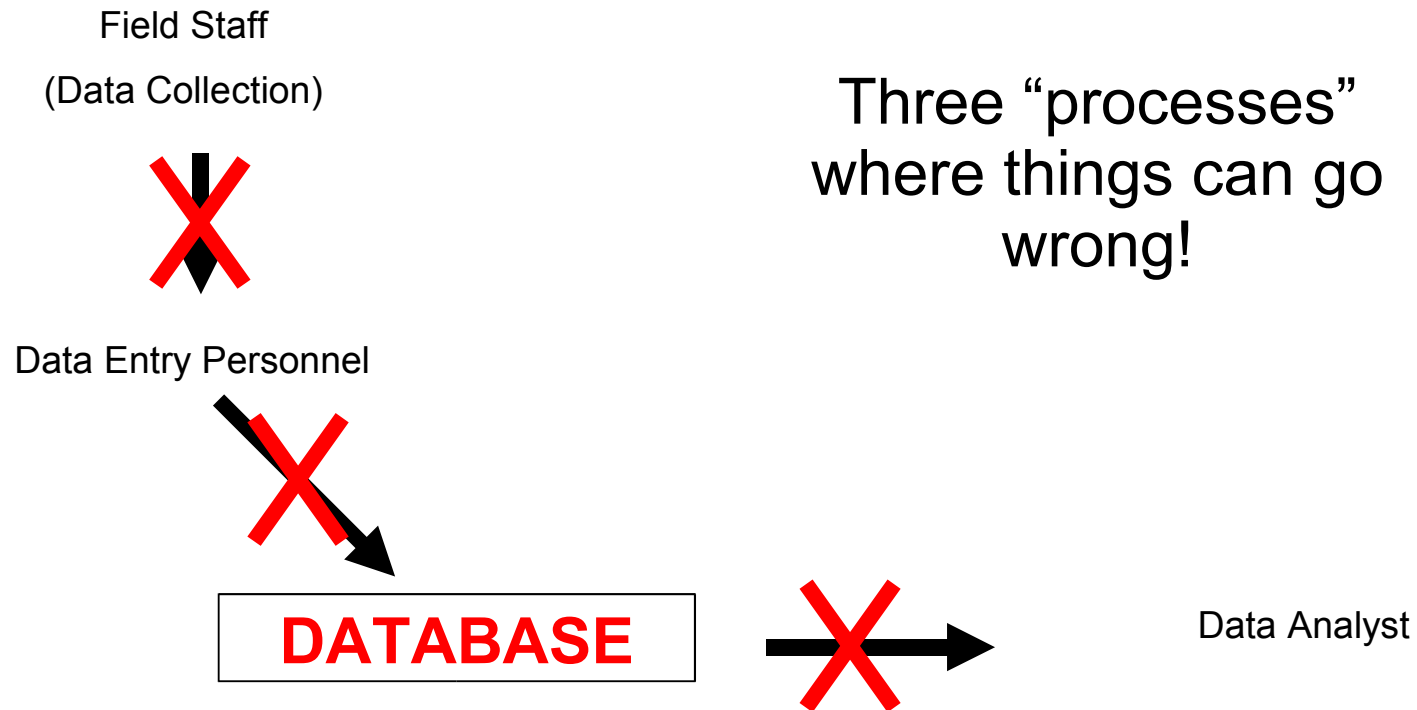
- Train data entry staff on how to enter the data.
  - data entry and navigation
  - what to enter and how
  - making backups
- Record everything that happens!



# Project Personnel



# Three Processes



# Documentation of the “Process”

- ♦ Keep a notebook and record **EVERYTHING** that happens!
  - ♦ decisions that are made about entry
  - ♦ change in personnel/staff
  - ♦ problems with data entry and solutions
  - ♦ provide dates
- ♦ Organize all data collection forms for easy retrieval.
- ♦ Documentation takes a lot of time, but it does pay off in the analysis phase of the project.

The notebook will be your memory!



# Correct Errors in the Data

- ♦ Errors in the database should be identified and corrected.
- ♦ Double data entry is one approach to minimize data entry errors.
  - ♦ data are entered twice
  - ♦ the two entries are compared for each variable and presents a list of values that do not match.
  - ♦ discrepant results are checked against the original data
- ♦ An alternative to this is to recheck or reenter a random proportion of the data.



# Test Data Entry Procedures

- ♦ It is important to always test your data entry procedures.
  - ♦ check your coding
  - ♦ check your data validation and checking procedures
  - ♦ revisions to your data dictionary
- ♦ Take five questionnaires and enter them into your newly created database...you will learn a lot!



*“For the things we have to learn  
before we can do them,  
we learn by doing them.”*

-Aristotle



# Creating a Dataset for Analysis

- ♦ In creating a dataset, think again about what you want to know.
  - ♦ Determine an appropriate analytical method.
  - ♦ Think about the “form” that the data should be in for this.
  - ♦ Query your database to get the data into this form.
- ♦ Most relational databases can export data to an ASCII format.
- ♦ All statistical programs can import ASCII text files.



# Merging Options in Statistical Software Packages

- ♦ Stata: merge, joinby, odbc
- ♦ SAS: merge, PROC SQL, SAS/Access software
- ♦ R: merge(), RODBC library
- ♦ Epi Info version 3.3: merge, relate



# Backing Up and Archiving

- ♦ Be sure to back up the data on a regular basis!
  - ♦ Some servers can do this automatically (eg, every evening).
- ♦ It's good archive a dataset with its documentation (data dictionary, etc.) and any necessary files for interpreting the data.



# Other CIDP Resources

- ◆ Relational Database Management Systems for Epidemiologists
  - ◆ [http://www.idready.org/rdbms/database\\_RDBMS.html](http://www.idready.org/rdbms/database_RDBMS.html)



# Summary

- ♦ Steps of Processing Data:
  - ♦ Defining fields
  - ♦ Creating a database
  - ♦ Format and range of permissible values
  - ♦ Creating a data dictionary
  - ♦ Coding
  - ♦ Data entry
  - ♦ Creating a dataset for analysis
  - ♦ Backing up and archiving the dataset

