Creating A SARS Database

Wayne Enanoria, PhD, MPH
Center for Infectious Disease Preparedness
University of California at Berkeley

Table of Contents

Introduction

Epi Info is one program you can use to create a relational database. The main advantage of using Epi Info is that you do not have to worry about establishing table structure and relationships. The program does all of this planning for you when you first enter the data entry program. This document is designed to give you a detailed description of the table structure for the SARS database that you created in Workshop I. The screenshots come from my sarscasereport.mdb project under Workshop I directory.

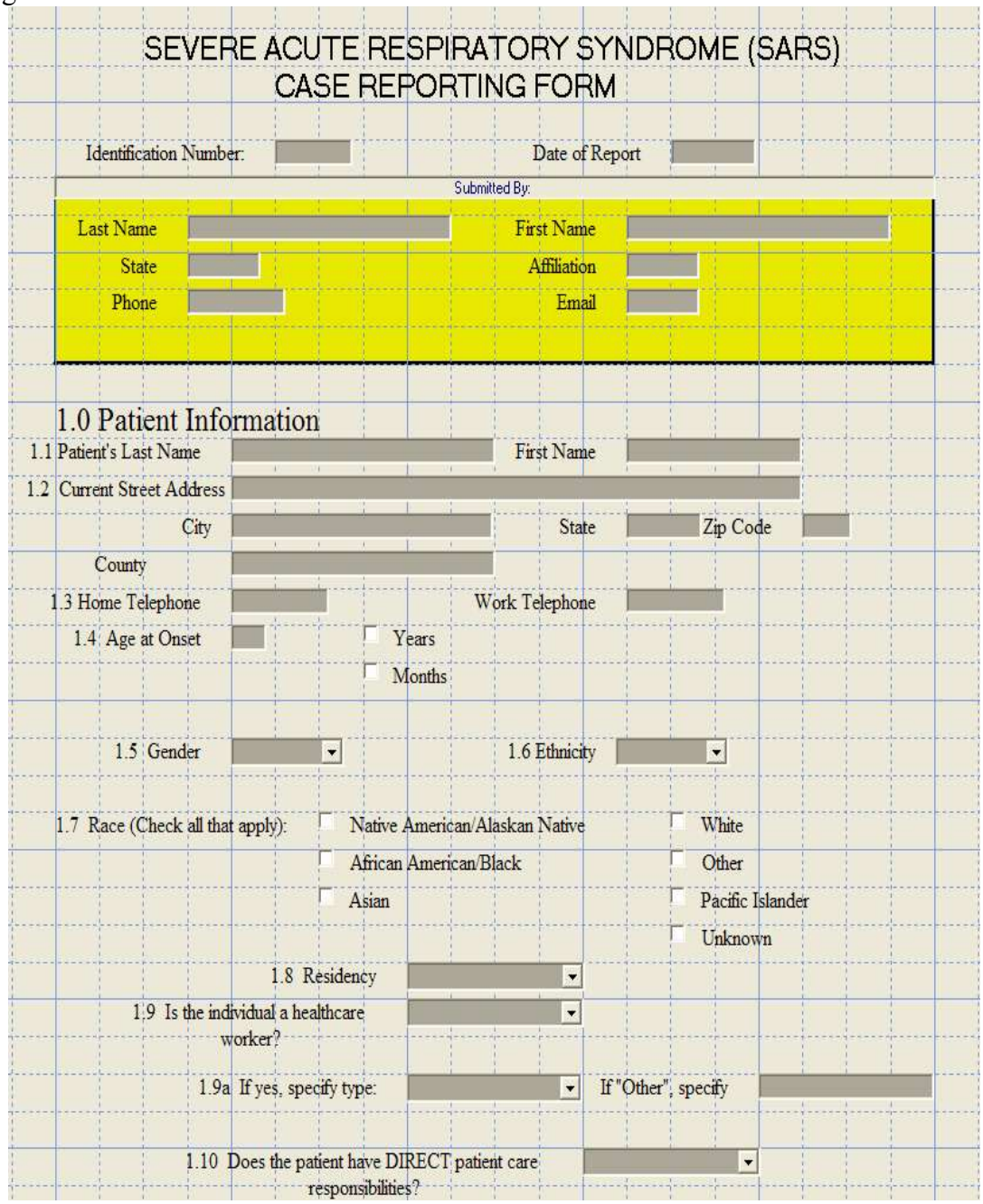
Structure of the View

The SARS database that I created contains the following four pages stored in one view called “case”:

1. Patient Information
2. Signs and Symptoms
3. Risk Factors
4. Laboratory Evaluation

Page One: Patient Information

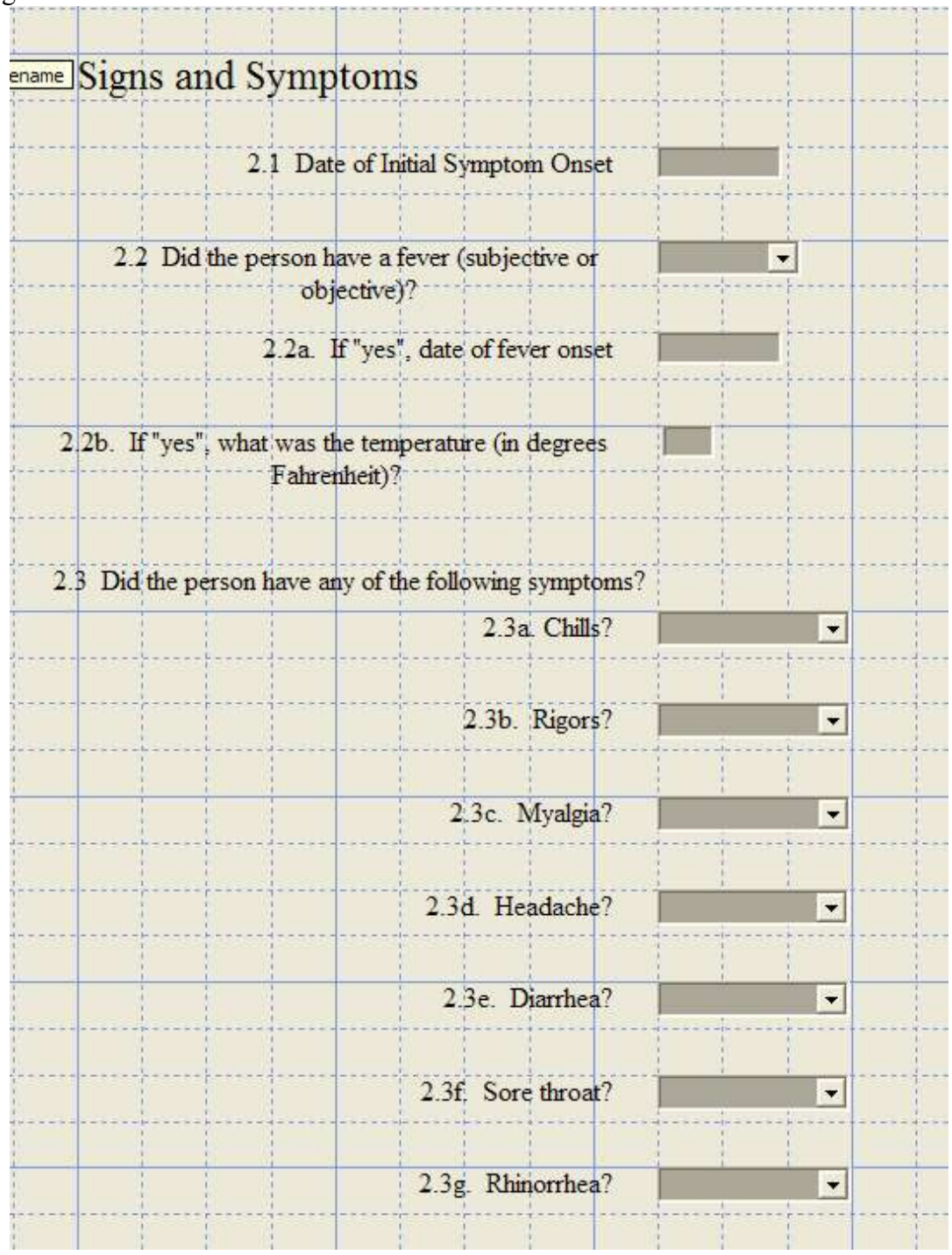
Page one of the main view called "case" looks like this:



Notice that I created the following fields as drop-down menus (legal values) where the person entering the data can only enter in particular fields: gender, ethnicity, residency, hcw (question 1.9), hcwother (question 1.9a), ptcare (question 1.10).

Page Two: Signs and Symptoms

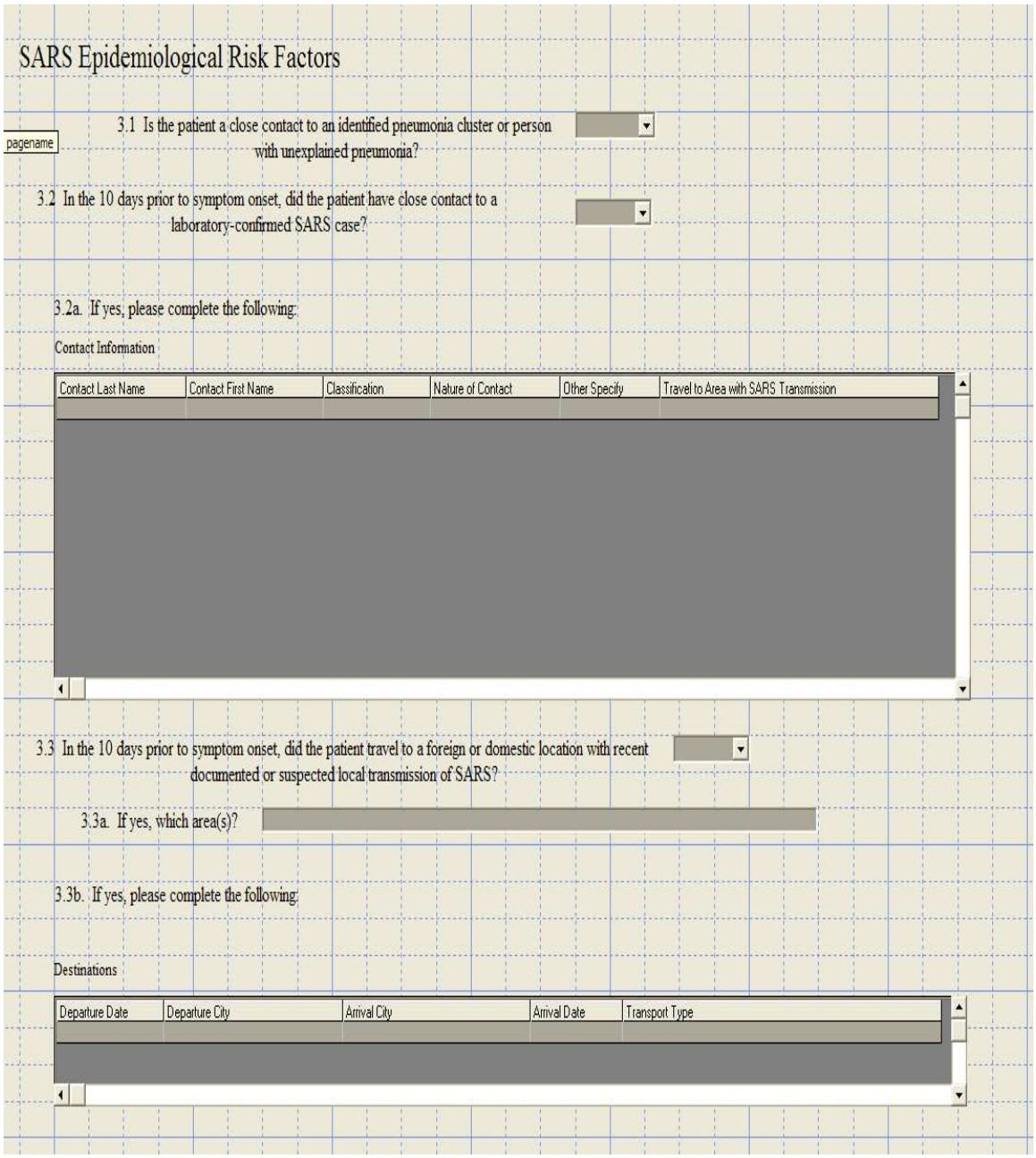
Page two of the main view called “case” looks like this:



The following fields have drop-down menus (legal values): fever, chills, rigors, myalgia, headache, diarrhea, sore throat, rhinorrhea.

Page Three: Risk Factors

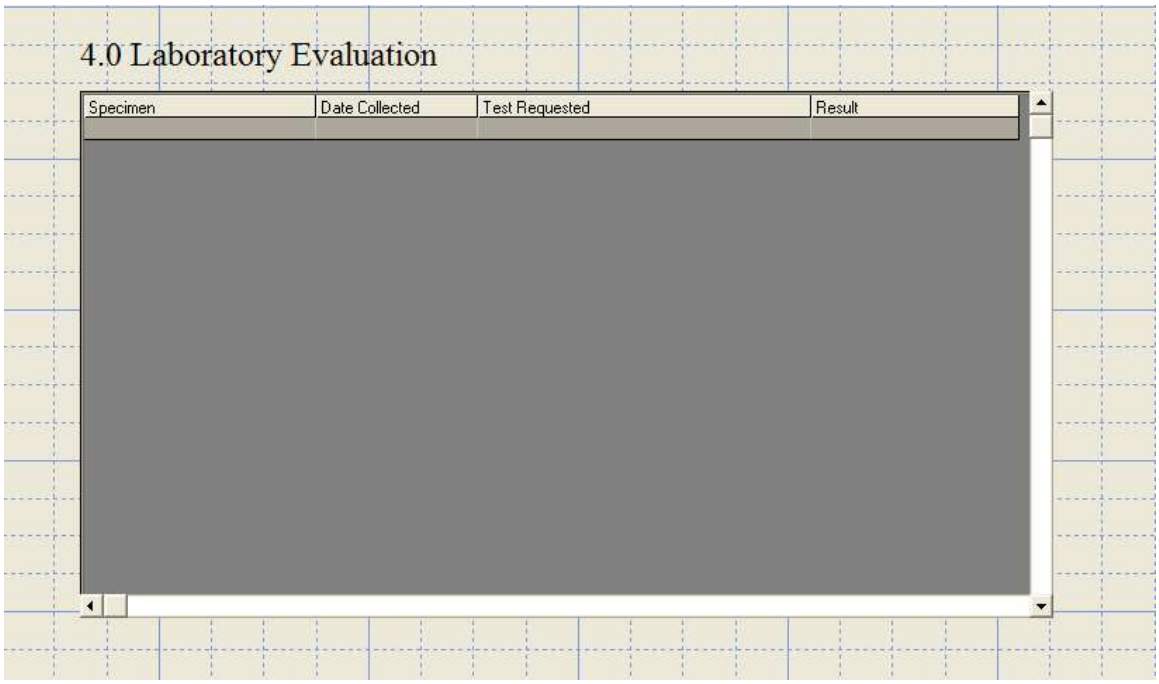
Page three of the main view called "case" looks like this:



This page of the view contains two grids: one called "Contact Information" and another called "Destinations" that enable the data enterer to view multiple contacts and multiple destinations for one case at the same time (in a spreadsheet-like view).

Page Four: Laboratory Evaluation

Page Four of the main view called "case" looks like this:

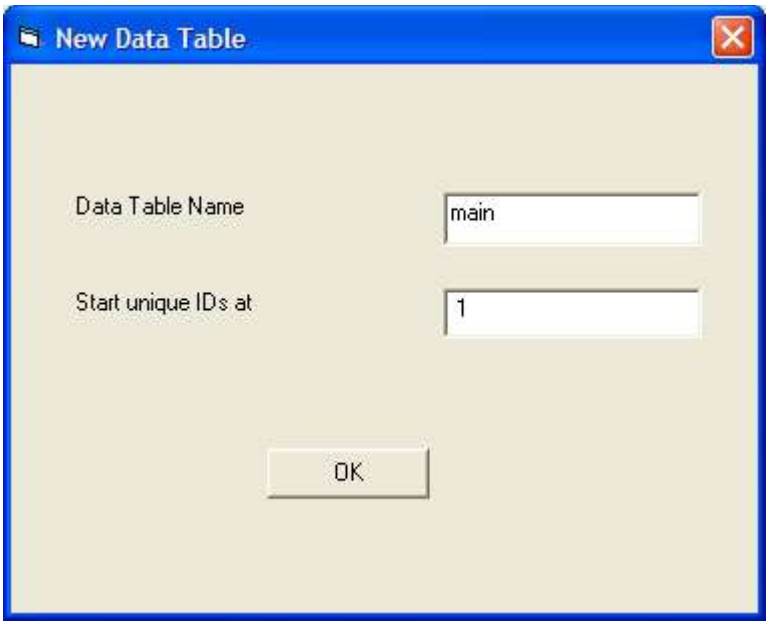


The page consists of one grid called "4.0 Laboratory Evaluation". One can enter in multiple lab specimens per case and be able to see all of them in a spreadsheet-like view.

One could have set up result field as a drop-down (legal values) field that controlled the responses that could be entered into this field. For example, I could have set it up as a legal values field that only allowed the following responses: "positive", "negative", "unknown", "not tested". But in the screenshot as you see it here, the data enterer would have to type out the response itself (rather than selecting it from a drop-down menu).

Structure of the Tables

When you enter data for the first time in a database that you have created, Epi Info will create all of your tables for you. Information for one case can be stored on one table or multiple tables, depending on how the information relates to one another. It will assign a unique identification number to each record in your database as you are entering it so that it can keep track of all the information for each case that is entered on all of the tables. A dialog box comes up that asks you for some information: (1) the name of the data table and (2) the starting number for the unique identification numbers. The screenshot on the next page is an example of this dialog box.



So in this example, I am naming my data table “main” and I would like Epi Info to start the unique identification numbers at 1. Epi Info will then number the records sequentially from this starting number (eg, 1, 2, 3, ...).

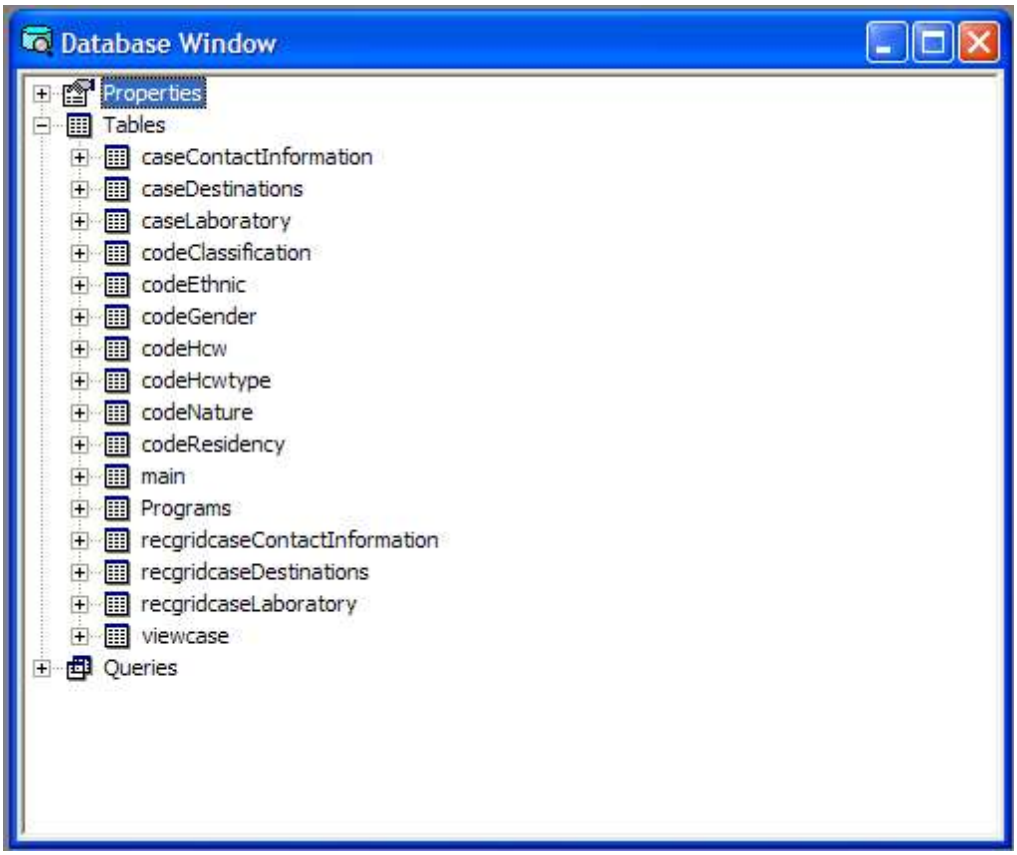
By default, the data table name will be the same as the name you gave to the view (in this example, “case”) but Epi Info does not like to name the data table “case”. If you try to name the table name “case”, you will get an error message that reads “Table name may conflict with reserved word. Please try another name.” Epi Info will then rename the table name automatically to “case1”. Avoid this all together by naming the table “main” instead of “case”.

In addition to this table, Epi Info will also create all of the related tables depending on how you set up the view. For example, let's take a look at our database using the Visualize Data program.

Open the SARS Database in the Visualize Data Program

1. From the main Epi Info menu, select “Utilities” --> “Visualize Data”.
2. From the File menu, select “Open” and open the SARS database that I created that contains data (sarswkshop2.mdb).

Once you open up the sarswkshop2.mdb project file, you can look at any of the existing tables and their contents. For example, once you open the Epi Info project file, you see the following dialog box:



In this database, there are 16 tables stored in the project file.

Legal Value Tables

Each field that you created as legal values (where only certain responses could be entered into the field) was created as a separate table. If you look at the tables, you will notice that all of the legal value fields you created are all given some name with the prefix "code". For example, when you created the field called "Gender", Epi Info entered the legal values for this field as a separate table called "codeGender". Similarly, the legal value fields "Classification", "Ethnic", "Hcw", "Hcwtype", "Nature", and "Residency" all have tables for these values named "codeClassification", "codeEthnic", "codeHcw", "codeHcwtype", "codeNature", and "codeResidency" respectively. You can view these tables as validation tables which "validate" the responses entered into the associated fields.

Data Tables

The main data table (called "main") that stores most of the information can also be viewed in the Visualize Data program. If you double-click on the table called "main", you will see all of the fields for the main data table as well as all of the records that have been entered. If you notice, each record has a value ("male" or "female") entered in the

field called “Gender”. The entries for this field on the main form are validated with the codeGender table.

Grid Tables

The grids on the view are each saved as separate tables. The name of these tables are named as default [*view name*][*grid name*]. For example, the data stored in the grid called “ContactInformation” is saved in a table called “caseContactInformation” (the view was named “case”). Similarly, the grids called “Destinations” and “Laboratory” have their data stored in tables called “caseDestinations” and “caseLaboratory”, respectively.

Tables Containing View Information

Information pertaining to how fields look in the view are also saved in tables within the project file. For example, the name of the field, the question/prompt, the page number the field appears on, field type, special formatting, tab order, prompt font, etc., all get stored in a table called “viewcase” (or more generally, [*view*][*viewname*] – each table has the prefix “view”). Refer to the Epi Info manual for a more detailed description of what gets stored in these tables.

Similar to these tables are the tables containing information about how the grids appear in the view. Each of these tables have a prefix “recgrid” followed by the view name and the name of the grid. For example, the specifications of the ContactInformation grid is stored in the data table called “recgridcaseContactInformation” (remember [*recgrid*][*viewname*] [*gridname*].