PROJECT 3:

The San Lucas Arrest Module Application SLAMA

CSE100/INFO 100

Fluency with Information Technology

Introduction

This next section of this course is an overview of database technology and relational database design. While learning the concepts behind database design is helpful, your really only master those concepts by working through an actual project.   This project is intended to improve your understanding of databases and to provide an opportunity to study how the information system of an organization is created to serve its specific needs. This project will also allow you to reflect on how to improve an existing system.

Your goal is to build part of the database application for the police department in San Lucas, a small imaginary city in Washington State. The application is called the San Lucas Arrest Module Application (SLAMA). This database application will allow police officers to record their arrest information in a quick and efficient manner using a number of forms (user interfaces) that help place the data in the correct location in the database. There are many steps involved in the arrest process. An arrest is complete when a suspect is no longer free to walk away and all paperwork of the initial charge against him or her is completed. This project will be limited to the arrest process itself and only collect data about the arrest up to the point of arraignment, but not including arraignment. Your job will be to concentrate on the area of the application that deals with DUIs: driving a vehicle while under the influence of drugs or alcohol.

An important aspect of the database system is to assure some measure of security and privacy for the suspects. Remember, one is innocent until proven guilty in a court of law! This database system is intended to support collection of information needed during that legal process by attorneys and the court. To gather some background information on the arrest process for DUI arrests, read the following two articles from FIND LAW, a legal resource for the public:

Chronology: The Arrest Process
http://public.findlaw.com/dui/life_events/le5_2chronology.html

A Walk Through Two Drunk Driving Cases: How the Criminal Justice System Works
http://public.findlaw.com/dui/nolo/ency/4714A4D6-9500-4BDF-89AC9DFBF832DCE8.html

Understanding what things (items, people, events, etc) are important to an organization and need to be tracked is the first step in any database system design. The next step is to identify how all of those things associate with one other. By exploring how an organization works and how information flows through it, you can identify those items most important to the system.

Context

In a real world design process, the database developer or team spends many hours interviewing the system users and tracking organization flow in order to identify the many items, events, actions and people that are to be tracked in the information system. In these interviews a model of the system will be developed. *This section and the class lectures on the project are intended to replace many of those interviews.*

This database is intended to support the arrest and booking process of suspects and allow police officers to capture details of the arrest normally written in paper forms (which are still used, but hard to search!). Make sure to read the articles listed in the Introduction for more context.

During the arrest and booking process, a suspect will give personal information, be fingerprinted, have their personal items taken and inventoried, be allowed a phone call and be placed in a jail cell while the officer checks their arrest history and a decision is made about bail amounts. If any tests not performed in the field, but related to the arrest, are needed (such as blood or urine tests for drugs or alcohol) then those samples will be acquired at this time as well. When the test results come back from the lab, the officer is notified and the data entered to the database.

SLAMA does not deal with payroll, insurance or other issues related to employment.

Look at the flow of information through the arrest as a guide to the design.

Arrival at Police Headquarters and start of the official arrest and booking process:
When a suspect arrives to the station with the officer, a desk clerk interviews the officer to gather a small amount of personal data that is entered into the suspect's record. If the suspect has been arrested before, the desk clerk will do an initial verification of the information in the existing record. The desk clerk is then responsible for starting the current arrest record that will include the suspect and officer key information. While this is happening, the suspect is photographed, fingerprinted and personal items inventoried. They are then escorted to a jail cell while the officer completes their arrest information for the database.

Completing the Booking process:
The officer fills out a form to complete the arrest record, beginning to end. The form has spaces for investigation and arrest notes as well as identifying the particulars of the type of arrest, any field sobriety tests that were given or need to be done at the station.

If a non-field test is ordered by the officer to determine blood alcohol content (as in the case of a DUI) or to determine drugs in the system (i.e. one or more check boxes for blood or drug tests have been checked) then specimens (urine, blood, etc.) will be collected from the suspect to be sent to the lab.

A label must be generated to attach to the sample. The label must include the pertinent information about the test and the person requesting it. To assure privacy a tracking number is used whenever a suspect must be identified. The number is not randomly generated, but neither is it easily identifiable with a particular suspect. It is not necessary for the lab technicians to know who the test is for, only the test to be performed. The label will include the station information, the date of the test request, and the unique tracking number for that suspect's particular sample. The tracking number will be related to the individual arrest number.

Daily Test Manifest:
Since suspects must be arraigned within 72 hours, it is vital to have the results of the tests finished within 24 hours, maximum. A daily test manifest is put together at a certain time each day. It will list all tests requested on each sample, thus allowing the lab technician to take what is needed from each sample given.

Results Returned:
The lab technician will return the paper manifest as soon as the tests are finished and enter the individual test results. The outcomes are: *positive, negative, failed* (must be repeated) for drugs, and a number to indicate the BAC level for alcohol (ex. .89, .20, etc). When the results of specimen tests are finished, the outcomes are added to the arrest record so that a report can be generated for the DA for arraignment.

Arrest Reports:
Every so often, a request for various reports will be made. To assist the Police Administration,summary statistics are generated each quarter. These include the number of arrests, by type, as well as indications of the numbers drug and alcohol outcomes. The stored information will need to be queried in order to pull together a particular view that the requesting administrator has asked for.

Analyzing all of the previous description will reveal those entities (items, people, actions and events) most important to the clinic that should be tracked.

Needs Analysis: Identifying Important Entities

The actions and processes described above give us an idea of the people, events and their associations to each other that need to be tracked. Once identified, they become components, or pieces, of the system. These and additional components are necessary to support the police station's daily operation and should be included in the basic database design. They are numbered in the following description.

From the Arrival description, it is evident that tracking Suspect information is very important! Without it, a case will be tough to create. A Suspect table **(1)** is required to hold that information. This will contain

- An ID number for the Suspect (specific to the SLAMA database)
- The suspects full name (this might be acquired in several ways-but it is obtained)
- Gender
- Date of Birth
- Address information
- ??? What else might be important?

In order to help the desk clerk enter information into this table in an intuitive, simple manner, a form **(2)** is required. The form should be attractive, be embellished with the Police Station name and address and look suitably professional. Once new suspect information has been added to the table through use of the form, the desk clerk will start the arrest record that involves both the suspect and the officer. To enter initial arrest information will require opening an additional form to start an arrest record with the appropriate information. The way to do this will be explained in the Tasks section.

Also from the arrival description, there must be a officer table **(3)** containing important information about each officer at the station. This table should hold:

- Badge Number (specific to the Police station and the SLAMA database)
- Rank
- Full name information
- Home Contact information
- Social Security Number
- Date of Birth
- Hire Date

There is a form **(4)** for entering data for this table that is attractive and consistent in design with the other forms.

Also from the Arrival and Booking descriptions is the fact that each arrest event between a suspect and an officer should be recorded and tracked in its own table structure. The Arrest table will hold this data **(5)**. This table will include:

- The Arrest ID
- The Suspect ID from the Suspect table
- The Officer badge number from the Officer table
- The Arrest date (let the system enter the current date)
- The initial arrest charge (felony DUI? DUI? Other??)
- Miranda rights check-have then been read to suspect?
- A notes field for any extra information regarding Miranda rights
- Inventory field to list out suspect personal property for storage
- A way to indicate that they have signed off on the inventory list
- Field Sobriety Tests fields along with outcomes to be recorded:
    - Breathalyzer test-to calculate initial BAC: Blood Alcohol Content
    - Alphabet (could suspect say it backwards?)
    - Walk a straight line?
    - Stand on one foot and touch nose?
    - Other tests you may know about that the professor does not? ☺
- Tests requested at station (need a field for each test done at station)
    - Alcohol
    - Drug
- Test outcome data (need a field for each test outcome)
- A tracking number field to hold generated lab sample number
- Arrest Notes
- Investigation Notes
- Remarks/Notes

The tracking number is not the unique record ID for that table (that is the arrest ID attribute). Instead it is a multi-digit number used to identify the blood and urine specimens. This number preserves the anonymity of the suspect while the specimens and tests are being processed at the police lab. One tracking number is associated with each arrest of a suspect, and so if more than one test is ordered after an arrest, the same tracking number will be used on each. However, each arrest will generate a new tracking number if tests are ordered, thus preserving the uniqueness of each arrest. There should also be a field to record the outcome of each test, once it is returned.

The start of the arrest process between a suspect and the officer by the desk clerk will require use of a form **(6)** that can start an arrest record with the appropriate suspect and officer information. The desk clerk needs to view different pieces of data than the officer. The clerk is only concerned that:

- The arrest be uniquely identified from any other that day (possibly between the same officer and suspect!
- The proper suspect ID number is noted
- The proper officer badge number is noted
- The date of the arrest in entered automatically. *The date is entered automatically when the record is started using a particular function.*
- The charge of the arrest is indicated
- An inventory is made of the suspect's personal effects and there is a way to indicate that he or she has signed off on that inventory.

For the completion of the booking process between suspect and officer, another form **(7)** for entering the additional data to the arrest record is created. It should be set up with:

- The arrest number
- The arrest date (to verify the record is not from another time, but is today). *.*
- The suspect's name: first and last from the Suspect Table
- The Officer's name: first and last from the Officer table
- Officer rank from the officer table
- A way to enter in test information done in the field (and any results such as alcohol levels, pass or fail of sobriety tests, etc)
- A way to check off that a particular in-station test should be ordered (Drug or BAC)
- The tracking number
- Notes fields (arrest, investigation, remarks, etc.)

For each test requested, a label **(8)** must be printed to place on the specimen. This label includes the date of the request and the tracking number.

From the Daily Test Manifest description, it is necessary to create **(9)** a complete manifest of all tests to be sent to the lab with the specimens that:

- Lists the tracking numbers for all requested tests
- The checked boxes for each test are used as criteria confirming that it was ordered.

The lab should check each of specimens against the test manifest to make sure all are accounted for.

The Results Returned description explains that the lab will enter results for each test when complete using a special form. A paper copy of the manifest with results will come to the desk clerk as a back. There will only be one of 4 possible results listed (positive, negative, failed OR the alcohol level in the blood). This requires a form for easy data entry by the lab technician **(10)**. The information that the technician sees, however, should be limited to the tracking number and the outcome fields for entering the results. A Daily Arrest Manifest is then created with relevant suspect, officer and arrest information **(11)**. From this, a report for each arrest can be printed out and sent to the District Attorney **(12).**

From the Arrest Reports description, there will occasionally be reports (13) generated to display particular information requested by Administration.

Once an arrest is complete and the report sent to the DA for arraignment, the arrest record can be closed and flagged for movement to an Arrest archive since no further information needs to be added. The summary statistics report (14) is generated from the Arrest Archives table. The statistics are (a) the number of arrest, (b) the number of arrests by category, (c) the numbers of tests of each type and the outcomes. In addition, there should be a summary of the number of arrests by month for each month in the quarter.

Finally, because of privacy practices in the legal field, chain of evidence laws and possibly state laws, the database must be secured and accessed by authorized personnel only. Further, it should contain an "auto close" facility that exits the database when no activity has occurred for a while. This will protect unintended access when the officer, desk clerk or lab technician leaves the computer.

Summary

The following components of the San Lucas Arrest Module Application (**SLAMA**) database were identified in the Needs Analysis section:

Suspect table	
Suspect table entry form	
Police Officer table	
Police Officer table entry form	
Arrest table	
Initial Arrest data entry form (starts a new arrest record when suspect enters station)	
Booking/Arrest record completion form (helps officer complete the arrest process)	
Blood/Urine Specimen Label	
Daily Test Manifest	
Test results entry form	
Daily Arrest Manifest	
Arrest Reports	
(Arrest Statistics for Administration)	**EXTRA CREDIT**
(Arrest Archive)	***Not Part of Project 3***
(Police Station quarterly statistics)	***Not Part of Project 3***
(Security of database)	***Not Part of Project 3***

Part A: Understanding and Designing the SLAMA Database Structure

The goals of Part A are:

1. Understanding the flow of information through the Arrest process
2. Create the basic tables and data entry forms that will make up part of the database
3. Establish the relationships among the fields of the database to allow for joining of tables if future queries
4. Start the suspect arrest process within the system

Task 1: Understanding the flow of information

The purpose of this task is to visualize the flow of information and how it should be embodied in a database. Write a narrative description *(2-3 paragraphs, minimum)* of a suspect's interaction with the arresting officer, describing the flow of information that takes place during the arrest process. The narrative begins, "<*Name of suspect*> has been arrested for driving under the influence and enters the police station with the arresting officer where he/she is seated at the first processing desk to start the arrest record.….". From here, your narrative should describe the information flow process: what information is transferred, from whom and to whom. The people and places involved in the information flow are the suspect, the desk clerk, the officer and the testing lab. You should explicitly state how information in input to the system, who or what it goes through and where it is stored.

Task 2: Build the Basic Database Table Structures and Create Simple Data Entry Forms

Now that you have tracked the information flow through the SLAMA system, it's time to build the structures that have been identified to hold the information considered important enough to track.

A. Create the structures for three tables: **tblSuspect**, **tblOfficer**, and **tblArrest**. These tables must have the fields (attributes) identified in this description and the class discussion along with the appropriate properties to properly store the data. Use input masks where appropriate so that users are given a visual clue about how to enter data. An example of an input mask might be: *###-##-####* in a field that holds Social Security Numbers.

   It will be up to you to name the fields so that they are descriptive of the attribute they hold, but don't include any spaces or punctuation. Also, describe in the notes field what data that field/attribute will eventually hold.

B. Create a Form that enters data into the tblOfficer table. Call the form **frmOfficer**. Make the form attractive and easy to read (change the color, font, layout, etc.) for the user who will eventually have to add data to the table in the system with it.

C. Create another form to enter data into the tblSuspect table. Call the form **frmSuspect**. Make the form attractive and easy to read (change the color, font, layout, etc.) for the user who will eventually have to add data to the table in the system with it.

D. Using the two forms you have created, add four sample entries to the tblSuspect and tblOfficer tables. **Use names of family members for the suspects and friends for the officers!**

Task 3: Build Relationships

The purpose of this task is to relate the information in different tables based on a single attribute so that eventually the information can be associated into an integrated whole. This is a feature that Microsoft Access provides. It is establishing the Joins between tables prior to creating any queries. If this feature were not available, you would simply have to establish the joins between tables at the time you create the queries needed.

A. Establish that suspect records have a one-to-many relationship with Arrest records, and that Officer records have a one-to-many relationship with Arrest records as well. That means that a suspect may be arrested on multiple different occasions and for different crimes and an officer will likely participate in one or more arrests. What attributes would seem the most likely to show these relationships among the tables? Think of what attribute tblSuspect and tblArrest have in common and what attribute tblArrest and tblOfficer have in common.

Task 4: Start the Arrest and Booking Process

The purpose of this task is to support the officer in data gathering process for the arrest of the suspects:

A. Create a form called **frmArrest** that contains the following information (fields) from the **tblArrest**:

-the Arrest Number,
-Suspect Number,
-Officer Badge Number,
-Arrest Date,
-the fields that indicate what the suspect has been arrested for, initially,
-an indication that the officer has or has not read the suspect their rights
-the inventory field and any fields that will take notes on the arrest type.

B. Add a button to **frmSuspect** in Design View to save the current Suspect record as a row in the Suspect table once it is complete.

C. Add another button to **frmSuspect** form that will open up **frmArrest** and show only the data related to the Suspect whose record currently shows in frmSuspect. Use the Tool Wizard to accomplish this. You will need to explore the options the wizard provides to you.

When the buttons are both added and the form saved, notice that the desk clerk can click on the command button to save the record in form view and then click another button to open frmArrest to start a new row in the Arrest table. This begins a unique arrest record between the suspect and the arresting officer. IF the suspect has had any prior arrests in San Lucas, those will also be displayed. If not, there will only be a blank form waiting for data to be entered for the first time.

- A nice additional feature (but NOT REQUIRED) to the frmArrest would be a way to move to the last record in a set of Arrests for a particular Suspect. Otherwise, as you add arrest records for an individual, you will have to go through the first arrests to get to the current one.

D. The last thing to do is start several rows in the Arrest table using various Suspects and officers you have added. Once you have added the rows using frmArrest, check tblArrest to see if the records have been started. Also check to see that the current date appears on the arrest. For part B you will generate more seed data in a database that I provide to you, but for now simply test your various forms.

Part A: Due Tuesday, December 3rd, 11 PM

Part A is due at 11 PM on December 3rd through eSubmit. The paper submission for Task 1 is due in the next lecture (Wednesday). Late papers will not be accepted.

BEFORE TURNING IN YOUR DATABASE, BE SURE TO “COMPACT AND REPAIR” IT USING THE INSTRUCTIONS GIVEN TO YOU IN LAB.

The database for Part B will be released after Part A is submitted (very early Thursday, December 5th). Stay tuned!

Part B: Finishing the Booking Process

For part B you will work on an existing database. You will be notified where to download your copy by December 5th. The tables and data are included as well as the basic forms you needed to complete part A. Your job is to complete the booking process after the arrest has been made and support the processing of samples and recording of information using queries and forms based on those queries that you will add to the database.

Specifically:

5. Generate the Booking query and form to continue the suspect processing through the system
6. Implement a tracking number for the set of tests requested in an arrest so lab testing is not directly tied to the suspect's personal data.
7. Generate test data
8. Understand and implement a query to create a main manifest list for all daily tests
9. Using the report wizard, construct a paper list of all tests for the day and the tracking numbers associated with them. Also, print labels to go on the vials of these samples.
10. Create the query that will be used to record test results for each suspect's specimen.
11. Generate the daily arrest manifest and print the arrest report of each individual.
12. Reflect on issues of Privacy/access and dissemination issues surrounding database use and creation within the law enforcement community.

Task 5: Continue Suspect's Arrest Process

The purpose of this task is to support the officer as they finish booking the suspect. You are creating the query that will give a view of all fields (attributes) that the Officer needs to see when booking the suspect.

A. Using the QBE (Query By Example) to generate the SQL needed, create a query that will select information for display from each of the table structures. Decide what information you need from each table. Your choice should limit the information to just the essential data required to finish booking the suspect. For example, selecting the suspect first and last name is essential, but the officer's home address is not. Start your list with the following fields, the rest will be up to you to determine:
   - The arrest number
   - The arrest date (to verify the record is from today).
   - The suspect's name: first and last from the Suspect Table
   - The Officer's name: first and last from the Officer table.
   - What other information still needs to be acquired?
   - What other information already gathered does the officer still need to see?

(HINT: None of the tables used in the query will have ALL of their information displayed.)

B. Save the query as **qryBooking**

C. Now, create a form in design view. Save it as **frmBooking**. This form will be used by the officer to finish the arrest record. The basis for the form will be the view just generated by qryBooking.
   - In the properties of the form, select as the Record Source the query just created.
   - A window showing the attributes selected for the query should appear. If it doesn't, you can find it under **View>Field List** (but only when you are in Design View).

- If you forgot to add a particular attribute, simply save and close the form, open the query, make the adjustments, save it and close it. Go back to the form design view and find the field list under **View>Field List**.

D. Add the various fields to the form and arrange them in a pleasing manner. Adjust the color or background of the form by exploring and finding the appropriate properties window for different form areas in design view. Edit it to look attractive and easy to read. (See the second lab on databases for help with this task)

   For example, you will want to display the officer name so they can identify the records they are working on. You may list them at the top of the form and place a divider between their name and the suspect name information. Also place a divider between the suspect information and all the fields from the arrest table.

   You cannot make changes to officer or suspect information, or change the arrest number. These have all been established either in separate tables that you are not modifying OR were created as part of the record and cannot and should not be changed (arrest number)

Task 6: Set Tracking Number
Before starting this task, let's look at the tracking number in a little more detail.

The tracking number is a unique number used to label specimens when they go to the lab for testing. The main goal is to remove all information not required by the lab technician. Since their job is to test the specimen and not make any judgment on the suspect, a tracking number is used so the number should not be easily traced to the client. In the tasks for Part B, the tracking number is the way to refer to the client. The tracking number is only generated when the tests are ordered (in other words, when the booking is finished and the decision has been made to order certain tests).

A. Open frmBooking.

B. In Design View, find the Tool Box and select the command button object to add to the form.

C. Use the wizard process to add a command button to the Booking Form for this operation. Place the command button, select Form Operation and "Refresh form data" for your options. The caption of the button should be "Order Tests" and the Name should be **cmdOrder.**

D. When you have finished adding the button, add the additional code (shown below and also found at the end of the Appendix) to the click event by going to **View>Code** on the Menu bar. Because you are using a wizard to add the button, there is a lot of code generated by Access using Visual Basic for Applications. The place to add the code to the cmdOrder_Click event is shown in the Appendix below in blue. Hopefully, you can take what you know about programming concepts and dot syntax to understand a little of what the code you are adding is doing, even if you have not used Visual Basic before.

This is what happens in the code: When the Order Tests button is clicked at the end of the booking process, the tracking number field, which is normally locked to keep someone from accidentally editing it, is unlocked. The text in the field is set to (assigned) a letter string using the code given below and relocked. The code is part of the click event handler. Sound familiar?

So, to add in the correct code piece you need to know the name of the tracking number text box and the Arrest ID text box on your form (it is the same name you used for the tracking number field and the Arrest ID field in the Arrest table). In the click event for the command button, at the point where you are shown in the Appendix, enter the code statements listed below. Replace the text in bold with the field names you have used in the tables. Do not worry that you have not used this syntax before. Simply follow the instructions and replace the bold text box names with the names of your text boxes and place the code in the area shown in Appendix A.

NOTE: If you have not made any changes to the attribute names in the Database provided to you, then the code should work as is.

E. Place the cursor at the tracking number text box

```
TrackingNumber.SetFocus
```

F. Unlock the text box

```
TrackingNumber.Locked = False
```

G. Assign the text box this value

```
TrackingNumber.Text = "SLAMA" & Hex((12*(Year(Date)- _
Year("10/02/02")) +Month(Date)) * 1000000 + ArrestNum)
```

H. Relock the text box

```
TrackingNumber.Locked = True
```

The assigned value simply adds a hexadecimal number to the letters "SLAMA". The hexadecimal number encodes the number of months since the system was built times one million plus the Arrest Number.

Task 7: Generate Test Data
Test the functionality of your system with the following:

A. Use the Arrest form that is linked from the Suspect form to start several new Arrest records. You are provided with suspect and officer test date to do this.

B. Once you have started the records, go to the Booking form and view the new data that has been added.

C. Request Drug AND/OR BAC tests for the different suspects and generate the tracking numbers to order the tests. Not all suspects need to have test ordered in the station if they failed the field sobriety tests.

Task 8: Build A Virtual Test Manifest Table using a Query

A. Construct a query that will pull all tests requested for the current date. The query information comes from the Arrest table and requires you to show:
   - The tracking number. There must be data in this field (it must exist) for the record to be returned.

   - The date. Only Arrest records for the current date are to be listed so make sure you know how to use the current date as criteria!!!!

   - The request field *when at least one test request field (the checkbox) has been checked*. This is giving you a hint about establishing query criteria. There is no programming involved, but you will need to use Logical operators.

B. Once you have this data retrieved, save the query as **qryDailyTestManifest**.

C. Look at the SQL View of the Daily Test Manifest. Write a short paragraph (2-3 sentences) saying in English what the query is doing. Identify each clause of the query and say what it is referring to (the particular field, table/query or constraint). This write up should just be a description, line by line, of the SQL and is to be put just before your reflection paragraphs for Task 11.

Task 9: Paper Manifests and Labels

A. Using the Report wizard create the main manifest list for all tests using the qryMainManifest "table" of Task 8. The list should give the tracking numbers and date, as well as the name, address and phone number for the station at the top. Name the list **rptMainManifest**.

Go into design view to make any changes that aren't taken care of properly with the wizard.

B. Using the Label wizard (part of Reports, click on New at the top to find the wizard), create labels for each specimen. The labels will also be based on your qryDailyTestManifes. The label should have:
   - The name, address and phone number of the PSHC
   - The name of the tests requested
   - The tracking number
   - The date

   Name the labels **lblDailyTests**.

Task 10: Recording Results

Generate the query that will be used to reconstruct the daily test manifest in Task 8. It is assumed that the results are written on the paper manifest at the lab as a backup but the lab technician also enters the results into the database as well.

A. Using a query construct a virtual table (a view) for the test results that is similar to qryDailyTestManifest, created in Task 8, but with the added fields for outcomes for the tests showing instead of the test request checkboxes (which you will include for criteria, but not display). The query will be based on data found in the Arrest table, since that is where all the visit data for the current day and all previous days is stored.

B. Unlike Task 8, the criteria for date should not be the current date, but instead should be equal to a date typed in by the lab technician. By making the criteria for that field a request set in square brackets: **[Enter Date Tests Were Requested]** the query will prompt the user for a date when it is run.

C. Name the Results query: **qryDailyTestResults**.

D. Create a form based on this query that will allow the lab technician to easily enter in results. The form should:
   - Prompt the technician for the date shown on the manifest and then open up when the date has been entered, showing only records from that date where the tracking number is evident.
   - Include the tracking number field, date of arrest field and the results field. You do not need to show the test request checkbox. The query above should already be set up to use it as criteria, but it does not need to be displayed for the technician on the form. This avoids any possibility that the box could accidentally be "unchecked", thus contaminating the data. (The same argument can also be given to not show the date, but for now we will)

- Generate a combo box in Design View on the form where you can enter the 3 preset outcomes for the drug test. The technician will use this combo box to select the outcome written on the test manifest and have that result text stored in the Drug Results field of the Arrest table. For the BAC results, blood alcohol level number, they can enter it into the text box for BAC results by hand.

Task 11: Reflection
Add several paragraphs just after the SQL explanation for Task 8 to answer these questions:

A. What things would you change in the system to improve it? [The point of this is that this is NOT an adequate system, indicate how you would improve it and say why]

   These changes can be structural (for the tables) or interface (change in the way the forms are displayed) or in access (getting to certain information).

B. How could this database be used in ways for which it was not intended if not properly secured to keep much of the information private? [Some arrest information is made public after an arrest, such as the suspect name and address. What information that could be stored in this system might be misused?]

C. What things would you change to improve privacy in this system. Be specific. Do not simply say "Add password protection to the database". You may include that point, but only as an additional feature to others you have thought about.

EXTRA CREDIT:

NOTE: Extra Credit tasks will only be looked at if the rest of the project requirements are met.

EC#1: 3pts
Generate a query to show the number of DUI arrests made on a certain date. The user of the query must be able to enter the date themselves when prompted.

EC#2: 3pts
Generate a query to show the total number of BAC tests that have been given (from data in your database)

EC#3: 3pts
Generate a query to show the number of arrests made by a certain officer. The user of the query must be able to enter officer last name or badge number themselves when prompted.

EC#4: 4pts
Generate a report based on a query you create to show the total number of arrests of men and women. To get full credit the report must be professional looking and offer enough information to any official looking at it that no further explanation is needed about the data shown.

Part B: Due Wednesday, December 10th, 11 PM

Part B is due at 11 PM on December 10th through eSubmit. The paper submission of the SQL interpretation and reflection paragraphs from Tasks 8D & 11 is due in the Information School office by noon on Thursday. Late papers will not be accepted.

BEFORE TURNING IN YOUR DATABASE, BE SURE TO "COMPACT AND REPAIR" IT USING THE INSTRUCTIONS GIVEN TO YOU IN LAB.

Appendix:

Code Placement for Tracking Number Creation:

```
Option Compare Database

Private Sub cmdOrder_Click()
On Error GoTo Err_cmdOrder_Click

      TrackingNumber.SetFocus

      TrackingNumber.Locked = False

      TrackingNumber.Text = "SLAMA" & Hex((12 * (Year(Date) - _
      Year("10/02/02")) + Month(Date)) * 1000000 + ArrestNum)

      TrackingNumber.Locked = True

  DoCmd.DoMenuItem acFormBar, acRecordsMenu, 5, , acMenuVer70

Exit_cmdOrder_Click:
    Exit Sub

Err_cmdOrder_Click:
    MsgBox Err.Description
    Resume Exit_cmdOrder_Click

End Sub
```