**CIS 3400: DBMS I – Access Assignment #3 (*\*\* Revised 12/3\*\*)***

**Due Thursday, December 7th**

(You can hand it in as last as 12/12 with no late penalty, but there will be no resubmission option then).

This assignment uses the *Stanley EcoTours* case (Case Problem 4) in the Access textbook. The first part is a repeat of the lab work we did on forms. I have condensed the lab instructions on the form work into one document here. The second part is a structured report using a multi-table query.

For this assignment, you will use a modified version of the *Stanley EcoTour* database, called *Stanley\_Assignment\_3.accdb,* for this exercise. Please download it from the BlackBoard site; you will find it under ASSIGNMENTS / Advanced Access Lab.

# I. Starting Out

1. **Enter Your Name:** Please replace the values of GuestID 401 (‘StudentFirst’ and ‘StudentLast’) with your first and last name.
2. **Setting Referential Integrity:** Create the necessary referential integrity between the tables. Although there are some guests without reservations, that will not pose any referential integrity problem.
3. **Look at the Data:** Look at the structures for the following data:
   1. **GUEST:** I have added the *Guest\_Revenues* and *Guest*\_*Tours* fields from lab #3;
   2. **RESERVATION:** I have already inserted the lookup field for *TourID*, along with the *TourPrice*, *SingleSupplement*, and *TotalTourCost* fields.

**II. Forms, Lookups, and Calculations:**

It turns out that, although we can perform a look-up on an Access table access, we cannot automatically execute any operations with the look-up data through the table. For example, if the user selects a tour for a reservation, Access should be able to use the tour price and single supplement information to calculate the entire price. To do this, we need to use an Access object called a FORM. For now, we will create a simple form on RESERVATION, and then learn a little VBA code (Visual Basic of Applications) to extract the information from the look-up field and calculate the package cost.

We will now create a simple tabular form combination that will let the user look at a reservation, select a specific *TourID*, and update all the calculated fields for that reservation.

1. **Creating a new form**: The bulk of the work will be in creating and modifying a form for

RESERVATION:

* 1. Select RESERVATION from the table list;
  2. Select *Create / Form Wizard* on the main MS-Access toolbar;
     1. Make sure to select the table RESERVATION as the data source for the subform; ii) Copy all the fields from RESERVATION to the right side window and press **Next**; iii) **\*\*\*READ THIS CAREFULLY!!!!** Select ‘***Tabular’*** for the layout and press **Next**;

(1) Even though the Tabular and Datasheet styles look similar in their table appearance, the

Tabular style will allow us to do lookups, headers/footers, and calculations more easily; iv) Give the form a name like: **sbfReservation**

(1) I use the ‘sbf’ prefix to indicate a ‘subform’ that I will later use within another form;

* + 1. Select “Open the form to view or enter information” and press **Finish.**
    2. When you open the form, you will see that, although it is organized like the datasheet view form before, each field is clearly outlined as a separate field, and the new fields we added are empty.

(1) Also, *TourID* now has a pull-down menu that contains all the information from the lookup field

* 1. Make sure you have saved **sbfReservation.**

1. **Using Lookups to automatically add data:**
   1. Open **sbfReservation**in datasheet view;
   2. You will see that if you pull down on TourID, you get some information about the tours; please note the order of the fields in the lookup menu, and that Access will number them starting with zero (it is a common approach in computer programming:
      * 1. TourID
        2. TourName
        3. Price Per Person
        4. SingleSupplement
   3. We are going to have Access automatically plug in some values when we select the *TourID*.

Open **sbfReservation** in Design View.

* 1. Open the *Property Sheet* if it is not open, and select the DATA tab;
  2. Click on the *TourID* field in the detail line of the form to show that field’s Properties Sheet go to the EVENT tab;
  3. Find ‘On Change’ on the *Property Sheet*, and click on the ellipsis (…);
     1. When we select a lookup field in a form, it triggers an **ON\_CHANGE** event which we will use to write some VBA code;
     2. This code will take advantage of the fact that the *TourID* lookup produces a table; and we can extract values from other fields in that table. The first column (*TourID*) is column(0), and all the other fields are indexed consecutively. That’s why I had the numbering start with zero in item *13.b*  above.
  4. Select ‘Code Builder’ from the pop-up window;
  5. Access will open up VBA, and show you a procedure that looks like this:

***Private Sub TourID\_Change()***

## End Sub

*i)* Insert this simple two line code in between the start and end of the procedure;

*i)* The ‘ME.’ segment refers to the fields in the current sbfReservation form; *ii)* We will need to change the Me.SingleSupplement statement later):

## Private Sub TourID\_Change() Me.TourPrice = TourID.Column(2) Me.SingleSupplement = 0 End Sub

1. In the VBA window, click on /File / Save ….
2. Go back to your Access program, put the form in *Form View*, and try changing the TourID of a reservation. Once you leave the *TourID* field, you should see updates to *TourPrice*, *SingleSupplement*, and *TotalTourCost*.

*i)* We don’t have to do anything with *TotalTourCost* because it is a calculated field, and once those fields have values, Access automatically recalculates *TotalTourCost*.

**3) Modifying the VBA Code**

We have to adjust the VBA code to add the single supplement fee for travelers touring by themselves.

That means we have to test if the value of *People* in the reservation is 1: a) Open **sbfReservation** in Design View.

1. Open the *Property Sheet* if it is not open, and select the DATA tab;
2. Click on the *TourID* field in the detail line of the form until the Properties Sheet is showing that field;
3. Find ‘On Change’ on the *Property Sheet*, and click on the ellipsis (…) next to [Event Procedure];
4. We will edit the existing code to use an IF-THEN-ELSE structure to test for single travelers;

*i)* In this new version, the code checks to see if there is only one person (*Me.People*) on the reservation; if so, then the reservation’s single supplement is set to the tour’s single supplement. If there are more than one traveler on the reservation, the reservation’s single supplement is set to zero.

## Private Sub TourID\_Change() Me.TourPrice = TourID.Column(2) If Me.People = 1 Then Me.SingleSupplement = TourID.Column(3) Else Me.SingleSupplement = 0 End If End Sub

*f)* In the VBA window, click on /File / Save ….

*ii)* Go back to your Access program, put the form in *Form View.* Go to a reservation that has only one traveler, and update the *TourID field* to see what happens.

**4) Modifying the *People* Field**

We have set up our form to calculate the total package cost of a tour once we have selected a tour. Because of *TotalTourCost* is a calculated field, we could let Access simply redo the calculation if we change the number of people. There is one wrinkle, however; if we change the number of people from several to one, or one to several, we will need VBA code to modify the single supplement fee before recalculating the total tour cost.

1. Open **sbfReservation** in Design View.
2. Open the *Property Sheet* if it is not open, and select the DATA tab;
3. Click on the *People* field in the detail line of the form until the Properties Sheet is showing that field;
4. Find 'After Update' on the Property Sheet, and click on the ellipsis (…);
   1. If we use ‘On Change’ with a text box, the event is triggered as soon as we type one character, and we don’t want that.
   2. Hence, we need to use ‘After Update’ because that waits until the user is done editing the data in a text box before triggering the event.
5. Select ‘Code Builder’ from the pop-up window;
6. Access will open up VBA, and show you a procedure that looks like this:

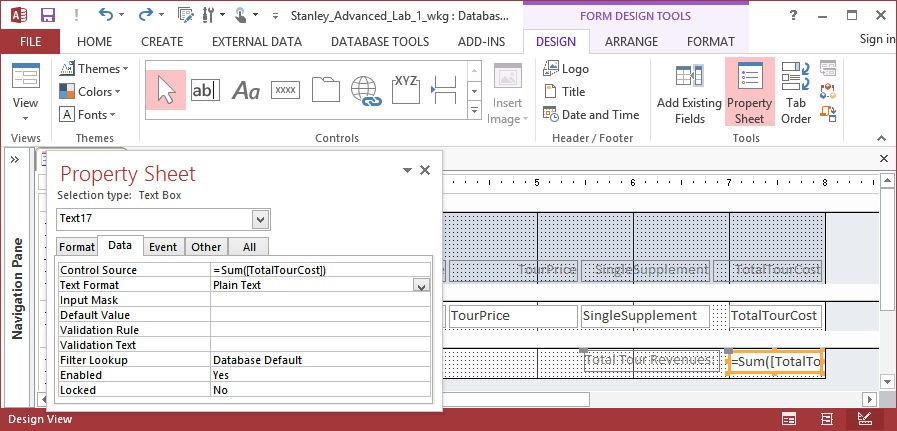
## Private Sub People\_AfterUpdate() If Me.People = 1 Then Me.SingleSupplement = TourID.Column(3) Else Me.SingleSupplement = 0 End If End Sub

*g)* In the VBA window, click on /File / Save ….

*iii)* Go back to your Access program, put the form in *Form View.* Go to a reservation and update the number of travelers to 1, or from 1 to several, to see what happens.

1. **Creating a Subtotal for Total Tour Revenues in the form:** 
   1. We will add a field that totals up the total revenues generated by all the tours. To do this, we will put a blank field in the form footer, then use a SUM() function to do the math:
   2. First, open the **sbfReservation** form footer just a bit (not too much) to create about a ¼ to a ½ inch of space;
   3. Click on the Text Box tool (the **ab|** icon circled below), and draw a rectangular box in the footer (the one that says ‘Unbound’ below). Align that box and its label better so that the unbound box is under *TotalTourCost –* see below:

d)



* 1. On the Property Sheet, click on the DATA tab put the following in the control source:

=sum([TotalTourCost])

* 1. On the FORMAT tab
     1. set the Format property to Currency ii) set the Decimal property to 0;

iii) ***\*\*\* On the OTHER tab, change the NAME property to fldTotalTourRevenues. We will use this property later in this assignment \*\*\*;***

* 1. Click on the **label box**for the subtotal;
     1. On the FORMAT tab, change the Caption property to “Total Tour Revenues”
  2. Save the form, and then reopen it. You will see the subtotal at the bottom of the form;
  3. If you change either the *TourID* or *People* field, but you will notice that fldTotalTourRevenues will not be updated until you leave the current record;
     1. You can make the recalculation occur immediately if you put the statement

***Me.recalc*** just before the END SUB statement **in both of** your VBA routines.

1. **Creating a Subtotal for Total Number of Tours in the form:**

Follow the instructions in Step 5 above on this page to create a subtotal that counts the number of reservations. Put this subtotal in the form footer also:

* 1. Use the =count(reservationID) as the data source for this control;
  2. ***\*\*\* Do not forget to change the NAME property to something you will remember; I used fldRsvCount \*\*\****

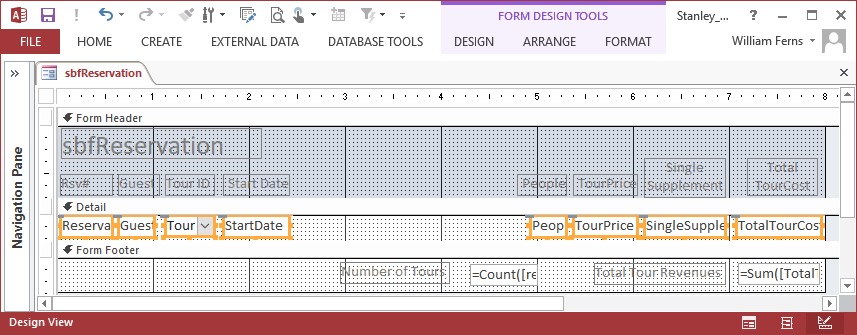
1. **Reformatting the form:**

You will see a lot of empty space inside some of the fields in the form, and we need to make space for a new field. We will learn to make some fields narrower and re-align some fields. a) Let’s start by making TourID and TourPrice narrower:

* + 1. Go to DESIGN mode.
    2. Click on the caption for *TourID*; iii) Go to /Arrange / Size / To Fit (or Alt-O; -S, -F) for fOrmat / Size / Fit iv) Then down the <Shift> key, and click on both the caption and the detail line field for *TourID*; v) Go to /Arrange / Size / Narrowest (or Alt-O; -S, -N for fOrmat / Size / Narrowest)
  1. *TourPrice*: go through the same process with *TourPrice*.
     1. To align the caption and detail line field when you are done, go to /Arrange / Align / Right

(or Alt-O; -A, - R)

* 1. If you want to move several fields / captions at once, just left-click, highlight the fields you want, and then shift them using the arrow keys or by dragging them while holding down the left-click.
  2. You can also make long captions into double lines to save horizontal space. See what I did with the captions for *SingleSupplement* and *TotalTourCost* below;
  3. Try to get your form in design view to look like this with a large space between *TourID* and *StartDate*:



1. **Adding an Automatic Lookup in the Form:**

We will now see an alternative way to display data from another table without saving it to the current table. If you remember, when we pull down the TourID, it also shows us the Tour Name. Let’s say that, besides the TourID, we want to be able to see the Tour Name on this form, but we do not want to save it with the data. We can add a display field that will show us the Tour Name from the Tour ID pulldown:

* 1. Put the form in DESIGN mode.
  2. Click on the Text Box tool (the **ab|** icon circled below), and draw a rectangular box in the detail line between *TourID* and *StartDate*; the box will now say ‘Unbound’.
  3. You will have to cut-and-paste the label box (it will say something like “Text99” to be in the form header.
  4. Change the label box to say “Tour Name”;
  5. We will use the lookup indexing method that we used to set the tour price and the single supplement, but just as a temporary display:
     1. The TourName is the second column in the TourID lookup, which means its index will be column (1);
     2. In the Unbound box, type: ***= TourID.Column(1)*** and press <Enter> iii) Go into FORM view and see what happens. Try changing the Tour ID.
  6. Go back into DESIGN VIEW and do the following:
  7. Resize and realign the new Tour Name caption and display box with the other fields;
  8. Open the PROPERTIES Sheet, and
  9. click on the new Tour Name field. ii) Click on the OTHER tab;

iii) Set the name of the new field to *fldTourName*; I use the *fld* prefix to note which fields are not fields in the database, but fields I have used for improving the interface.

# III. Creating the Form / Subform Combination

The bulk of the work will be creating a new main form that uses *sbfReservation* as a subform.

1. **Creating and Modifying** the **Main Form:**

Create the Main Form: Now we will create the main form **frmGuestsWithReservations**. a) Select GUEST from the table list;

* 1. Select *Create / Form Wizard* on the main MS-Access toolbar;
  2. Use the table GUEST as the data source for the form;
  3. First move GuestID and LastName to the window on the right;
  4. Second, copy the remaining fields to the right side window and press **Next**;
  5. This time, we will select ‘Columnar’ for the layout and press **Next**;
  6. Save the form with the name *frmGuestsWithReservations*
  7. Select “Open the form to view or enter information” and press **Finish**
  8. You will see that all the data is aligned against the left margin. Go into Design mode and do the following**.** 
     1. Widen the grid to 10”;
     2. Move the address fields (Address; City; State/Prov; Country to the right to make two columns of information.
     3. Move the remaining fields on the left (Phone / Guest\_Revenues / Guest\_Trips) up; iv) Make any other formatting changes you think are useful; try to squeeze the fields closer to the top of the form without overlapping them.
  9. View the form in FORM view to see how it looks

1. **Add the SubForm:** 
   1. Put the main form *frmGuestsWithReservations* into Design View.
   2. On the main menu bar, go to “Form Design Tools” and then the “Design” tab;
   3. On the center left of the menu bar below, click on Tools;
      1. Make sure ‘Control Wizards’ is set to ON; ii) Scroll through ‘Tools’ until you find ‘Subform/Subreport’; click on that icon.

iii) Draw a wide rectangle on the main form under your current fields; iv) At the first pop-up window, select “Use an existing form” and then “sbfReservation” v) Press Finish.

d) The design will look a bit messy, but save it, and let’s run the main form and see what happens.

1. **Run the Main Form:**

## a) Select frmGuestsWithReservations

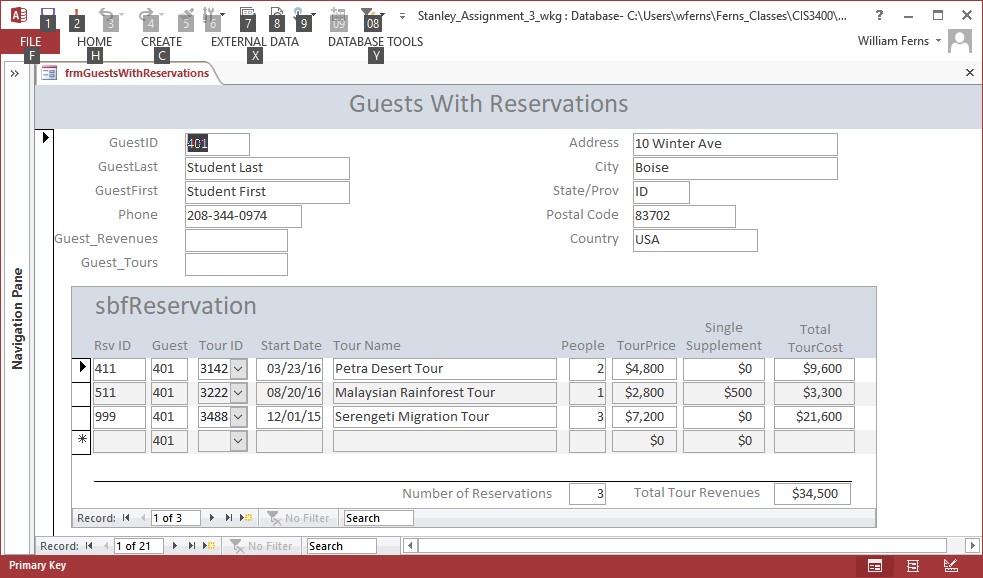
1. Notice that for the first guest, there are multiple resrevations showing;
2. Click on the field ‘LastName’ in the main form, and scroll through (using **page down** or the record locator arrows at the bottom of the form) until you find your name in that field—you will see that you have only two contracts on the subform.
3. While we are on your GUEST record, let’s update some info:
   1. For Reservation 511, change the tour to “Malasian Rain Forest”; you will see that the subform recalculates the Total Tour Cost;
   2. For Reservation 511, change the *People* field to 1; you will see that the subform adds the single supplement, recalculates the Total Tour Cost; and updates the Total Tour Revenues in the footer.

1. Let’s add a contract. Click on the empty reservation and let’s add some data:
   1. Reservation ID: 999
   2. ID: select the pulldown arrow, then scroll down until you find the record for “Serengeti

Migration Tour”; iii) Set Start Date to 12/1/2015; iv) Set the number of people to 3;

* 1. You will see that the subform inserts the Price Per Person, calculates the Total Tour Cost; and updates the reservation count and the total revenues for this guest in the footer.
  2. See Figure 1 below:

**Figure 1. Guest Form with Reservation Subform**



# IV. Updating GUEST fields from the Subform

*\*\*\* REVISION 12/3/2017 – The previous approach is no longer behaving predictably in VBA. I have updated this with another set of code. I have added the code with minimum explanation for now so that you can start using the code. I will provide an update with a fuller explanation sometime Monday. If you have already created these two temporary fields, you can delete them.*

~~The~~ *~~fldTotalTourRevenues~~* ~~and~~ *~~fldRsvCount~~* ~~fields in the subform footer are only displayed fields, but are~~ ~~not saved anywhere. We want to copy the values of those two fields in the subform footer into the~~ *~~Guest\_Revenues~~* ~~and~~ *~~Guest\_Tours~~* ~~fields of the GUEST table whenever a reservation~~ *~~sbfReservation~~* ~~is~~ ~~modified.~~

~~There is one problem, however; while we can display data from the subform on the main form, it is very~~ ~~complex to update the data on the main form from the subform. To create a link between the totals fields~~ ~~in the subform footer and the toals fields in the main form, we create two temporary fields to display the~~ ~~totals on the main form that can access the subform; we then update the and then when the user moves to~~ ~~a new record, the total fields in GUEST will be updated. We will do this on the main form, and it will~~ ~~require the last bit of VBA code we need:~~

1. *~~tmpGuestRevenues~~*~~: Click on the Text Box tool and draw a rectangular box in some empty space on~~ ~~the main form; Align that box and its label so that it is adjacent to the original Guest\_Revenues field;~~ ~~a) On the Property Sheet, click on the DATA tab, and then the ellipsis (…)~~
   1. ~~In the Expression Builder:~~
      1. ~~On the left column, click on~~ *~~sbfReservation~~* ~~ii) In the middle column, click on~~ *~~fldTotalTourValues~~* ~~(the name of the subtotal;~~ ~~iii) Press OK;~~ ~~iv) You will see the following in the control source:~~

~~=[sbfReservation].[Form]![fldTotalTourRevenues]~~

* 1. ~~On the FORMAT tab~~
     1. ~~set the Format property to Currency~~ ~~ii) set the Decimal property to 0;~~

~~iii)~~ ***~~\*\*\* On the OTHER tab, change the NAME property to tmpGuest\_Revenues \*\*\*~~****~~.~~* ~~You need~~ ~~to do this for the VBA code we will use in a few minutes to work..~~

* 1. ~~Click on the~~ ***~~label box~~*** ~~for the subtotal;~~
     1. ~~On the FORMAT tab, change the Caption property to “Guest Revenues”~~

1. *~~tmpGuest\_Tours~~*~~: Click on the Text Box tool and draw a rectangular box in some empty space on the~~ ~~main form; Align that box and its label so that it is adjacent to the original Guest\_Trips field;~~ ~~a) On the Property Sheet, click on the DATA tab, and then the ellipsis (…)~~
   1. ~~In the Expression Builder:~~
      1. ~~On the left column, click on~~ *~~sbfReservation~~* ~~ii) In the middle column, click on~~ *~~fldRsvCount~~* ~~(the name of the subtotal;~~ ~~iii) Press OK;~~

~~iv) You will see the following in the control source: =[sbfReservation].[Form]![fldRsvCount]~~ ~~c) On the FORMAT tab~~

~~i) set the Format property to fixed~~ ~~ii) set the Decimal property to 0;~~

~~d)~~ ***~~\*\*\* On the OTHER tab, change the NAME property to tmpGuest\_Tours\*\*\*.~~***~~You need to do~~ ~~this for the VBA code we will use in a few minutes to work;~~

.

1. **~~Save the form~~** ~~and view it in FORM view. Change something in the subform—either the TourID or~~ ~~the number of people. You will see that the temporary values display the new subtotals, but the actual~~ ~~data fields do not change. Next we will write the code to fix that.~~
2. **Updating the *Guest\_Revenues* field**:

***\*\*\* This is part of the revision from the previous page:***

## a) Open frmGuestsWithReservations in Design View

1. On the DESIGN VIEW menu, go directly to VIEW CODE (on the far right of the Access ribbon);

i) You may have to close the empty code window that is showing;

~~Forms![frmGuestsWithReservations]![Guest\_Revenues] = \_~~

~~[Forms]![frmGuestsWithReservations]![tmpGuest\_Revenues]~~

~~Me.Refresh~~

~~where~~

* + **~~[Forms]![frmGuestsWithReservations]~~** ~~specifies the main form~~
  + **~~![Guest\_Revenues]~~** ~~specifies the field on the main form that will be updated;~~
  + ~~[tmpGuest\_Revenues] specifies the subtotal field you created in the subform footer in~~ *~~sbfReservation~~*~~;~~
  + ~~The underscore ( \_ ) between the first two lines is the line-continuation character in VBA;~~ ~~when VBA runs into the underscore, it assumes the next line is part of the first one;~~

1. ~~You need this code in each event because they each modify the revenues~~~~for each guest, and~~ ~~hence alter the~~ *~~Guest\_Revenues~~* ~~field.~~
2. ~~Save the code, close the subform, and try the main form to see what happens.~~

~~Assuming that you see the new change working properly, we also need to add the code that updates the~~ *~~Guest\_Tours~~* ~~field in the GUEST table:~~

**~~5) Updating the Guest\_Tours field~~**

1. ~~Go back to your VBA code;~~
2. ~~In each of the events you created previously, after~~ **~~me.recalc~~** ~~but before~~ **~~me.refresh~~**~~, you will~~ ~~add the following line of code:~~

~~Forms![frmGuestsWithReservations]![Guest\_Tours] = \_~~

~~[Forms]![frmGuestsWithReservations]![tmpGuest\_Tours]~~

1. ~~You need this code in each event because in order to count the number of trips per each guest,~~ ~~and hence alter the~~ *~~Guest\_Tours~~* ~~field.~~
2. ~~Save the code, close the subform, and try the main form to see what happens.~~
3. ~~You can also open the GUEST table and see that the~~ *~~Guest\_Revenues~~* ~~and~~ *~~Guest\_Tours~~* ~~fields~~ ~~have been updated.~~
4. In each of the events you created previously, after **me.recalc**, you will add the following lines of code. I have inserted comments (the green text following single quotes) to provide some explanation of what these commands do:

***' The next two commands calculates the total revenues and tours for each guest; tmpRevenues = DSum("TotalTourCost", "Reservation", "guestid = '" & GuestID & "'") tmpTours = DCount("GuestID", "Reservation", "guestid = '" & GuestID & "'")***

***' This next line concatenates (with &) the above fields with an SQL UPDATE command***

***sqlstr = "UPDATE GUEST SET GUEST\_REVENUES = " & tmpRevenues \_***

***& ", GUEST\_TOURS = " & tmpTours \_***

***& " where guest.guestid = '" & Me.GuestID & "'"***

***DoCmd.SetWarnings (WarningsOff) ' turns off annoying warning message***

***' This next line runs the SQL Update command from within the VBA code DoCmd.RunSQL sqlstr***

***DoCmd.SetWarnings (WarningsOn) ' turns on annoying warning message***

1. **Crossing Ts and Dotting Is:** 
   1. *Again you can safely delete tmpGuest\_Revenues and tmpGuest\_Tours if you have already created them;* 
      1. ~~Go into DESIGN view and highlight the two fields;~~
      2. ~~Open the PROPERTIES window and go to the FORMAT tab;~~ ~~iii) Go to the ‘Visible’ property and set it to ‘NO’~~ iv) ~~View the form in FORM VIEW~~
   2. Use the Tab Order tool to review that your fields tab in a predictable order in both of your forms.
   3. You may want to set some fields on *sbfReservation* so that they can’t be changed by the user, because your pulldown menu on *TourID* modifies these fields: i) TourName
      1. TourPrice iii) SingleSupplement iv) TotalTourCost
   4. Go to /Properties / Data / Enabled, and switch ‘Enabled’ to ‘NO’ and see what happens.

1. **Adding a Reservation:**

Add a reservation to your student record: a) ReservationID = 9998

* 1. Tour: Pick the ‘Serengeti Migration Tour’
  2. Start Date: January 2, 2016
  3. Number of People: 1

1. **Crossing Ts and Dotting Is:** 
   1. Use the Tab Order tool to review that your fields tab in a predictable order in both of your forms.
   2. You may want to set some fields on *sbfReservation* so that they can’t be changed by the user, because your pulldown menu on *TourID* modifies these fields: i) TourName
      1. TourPrice iii) SingleSupplement iv) TotalTourCost
   3. Go to /Properties / Data / Enabled, and switch ‘Enabled’ to ‘NO’ and see what happens.

You are finished with the FORM part of the assignment. Next, you will create and design a report;

# V. Creating a Structured Report

You will use **Tutorial 7, Case 4--Steps 1 through 4 to** create a report named *rptTourReservations.* The report is similar in sophistication to the report you should submit for Milestone #4. You will also find the specifications on BlackBoard.

1. Step 2: Pay attention to the fields and the order of the fields that you include in the query. Also, please follow the instructions for ordering the query;
2. Step 3: You can simplify the process of creating a custom report from steps 3a to 3c by using the

Report Wizard

* 1. Use *qryTourReservations* as your data source;
  2. Use all the fields;
  3. View the data by *Tour*;
  4. Grouping: You do not need any additional grouping levels;
  5. Sort Order: use *StartDate* as the first sorting field, *State*/*Prov* as the second;

i) On that same window, click on “Summary Options” ii) For *People*, click SUM, and press OK, then NEXT;

* 1. Use the OUTLINE format for your report.

1. Steps 3d through 3i: follow the instructions as stated;
2. Please modify your report so that it resembles Figure 7-43, including changing the various field labels and textboxes so that they are well-formatted and informative.
3. YOU DO NOT NEED TO DO STEPS 4 and 5.

**VI. What you should submit:**

You should submit a package with all of the following items.

1. A cover page, with the number and name of the case, your name, and your section;
2. A screenshot of your main frmGuestsWithReservations and the sbfReservation form;

a. ***the GUEST record should be the record with your name***; C. A screenshot of those forms in DESIGN view;

1. A printout of your Visual Basic Code;
2. A screenshot in DESIGN view of the qryTourReservations query; F. A screenshot of in DESIGN view of your rptTourReservations;
3. The output of the report.
4. You should also send me your compressed MS-Access file; please make sure you label it:

*lastname\_firstname\_EcoTours\_Assignment\_3.accdb*

# VII. Extra-Credit: Switchboard

If you do the switchboard correctly, I will drop your lowest project grade and replace it with the grade for the switchboard.

Create a Switchbord Menu system allows the user to select:

* The data entry form you created earlier in the assignment;
* The query you created for Part V above ;
* The report your create for Part V above; please display it in *preview* mode;
* Close the switchboard and return to the database;
* Include printouts of the macros you use for the switchboard