Access 2010: Cleansing Excel data for import into Access

Practical workbook

University of
BRISTOL

IT SERVICES

Aims and Learning Objectives

By the end of this workbook, you will be able to:

- explain the problems that can be encountered if data is not consistently entered
- cleanse Excel data for import into a simple, multi-table Access database;
- import cleansed Excel data into multiple Access tables;
- create relationships between the tables;
- modify the tables so that data input via Access is simplified.

Document information

Course documents and files

This document and all its associated practice files are available on the web. To find these, go to http://www.bristol.ac.uk/it-services/learning/resources and in the Keyword box, type the document code given in brackets at the top of this page.

Related documentation

Other related documents are available from the web at:

http://www.bristol.ac.uk/it-services/learning/resources

Introduction

This document provides you with experience of the tasks that you will have to do if you want to import data held in Excel into Access tables. The reality is that this process will be time-consuming and fiddly and in real life almost certainly a lot more difficult than in the following tasks.

There are several important things to keep in mind when doing this for real:

- Make sure that you have designed a viable data model (i.e. that you have the right tables and the right relationships)

- Keep backups at all times, just in case you scramble your Excel data

- Be extra careful when manually entering key field data – get this wrong and you'll be joining the wrong records together in Access

Prerequisites

Experience of the currently-supported versions of Excel and Access are required.

You will get more out of this document if you have done one of the following:

- Attended "Access 2010: An introduction" (ITS-ACC10-1) and "Access 2010: Building Access databases" (ITS-ACC10-2).

- Worked through course documents for the above two courses.

Contents

Document information

Task 1   Spreadsheet examples

Objectives   To look at a spreadsheet where the data has been cleansed and to look at a more usual spreadsheet example with multiple examples of bad practice.

Comments   Entering data into a correctly-designed database precludes many of the problems that are faced when using spreadsheets for data entry, such as having to type the same data over and over again and threat of typos.

Spreadsheet with cleansed data

It is useful to see an example of good practice so that you can better identify bad practice when you come across it.

1.1   To see an example of good practice:

➢ Open the Excel file **PlanningYourDatabaseGood.xlsx**.

➢ Sort the spreadsheet by the **Title** column.

➢ Notice that every entry, where the same artist appears more than once, is spelled and formatted exactly the same.

➢ Sort the spreadsheet by the **Format** column and repeat the above scrutiny.

➢ Do the same for the **Label** column.

➢ Have a general look through and close the file.

Spreadsheet without cleansed data

1.2   To see an example of the more usual type of spreadsheet:

➢ Open the Excel file **PlanningYourDatabaseBad.xlsx**.

➢ Do the same as above and jot down the problems you find in the box below.

```



```

Note   Check how many Beatles records are grouped together in the sort – there are actually six Beatles records in the Spreadsheet. Also, see if you can find, by sorting or searching, out how many recordings of steam locomotives there are.
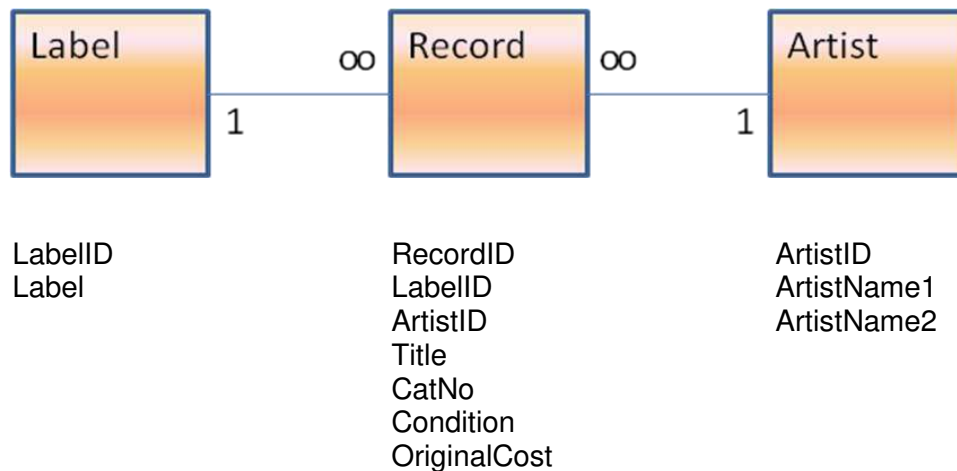
Task 2   Prepare cleansed data for import into Access

Objectives   To split cleansed Excel into separate sheets so that it can be input into Access tables.

Comments   The data in the Excel spreadsheet has already been cleansed. As you saw in the previous task, things need to be spelled and entered consistently. Time spent doing this makes the following tasks much more likely to be successful.

The identified data model and fields

Not only has the data already been cleansed to preclude inconsistent data or typos from causing havoc later, but we've already worked out what tables our database needs, as well as which data entry fields go where. Further documentation exists to help you work out your database design and to help you understand the following diagram (it's all quite simple, really – honest!) – see the **Related documentation** section on the rear of the title page. The important thing to understand here is that each table must have a key field – basically a unique code for each record – which is what causes most of the complexity in the following tasks.

LabelID	RecordID	ArtistID
Label	LabelID	ArtistName1
	ArtistID	ArtistName2
	Title	
	CatNo	
	Condition	
	OriginalCost	

Split the spreadsheet data into three worksheets

The main issues are that we need to split the current Artist field into two fields so that sorting can be done on group name or artist surname and we need to add key columns to the spreadsheet so that the data, when split, will join up again once imported into the database.

2.1   To view the spreadsheet that we need to cleanse:

➢ Open **PlanningYourDatabaseToBeCleansed.xlsx**.

➢ Look at the data and think about the problems inherent in cleansing large data sets. Remember, this is just a small amount of example data here!

2.2   To add the LabelID column:

➤ Click in the first cell under the column heading in the **Label** column (cell **D2**) and sort the data in ascending order (**Sort A to Z**).

➤ Now insert a new column and in the first cell in the column, type **LabelID** (that's a lowercase "L" first, followed by an uppercase "i").

2.3 To add in unique identifiers for each label:

➤ For the first label (Afro Strut), type **1** in the **LabelID** column.

➤ Add **2** for the next label, and increment by one for each new label (not forgetting to give the same number where more than one record is on the same label).

Note You should end up giving ZTT the unique identifier of 38 – if you have a different number, go back and double-check what you have done.

2.4 To split the Label data from the main set of data:

➤ Highlight the **LabelID** and Label columns.

Figure 1 – highlight the LabelID and Label columns thus (it doesn't matter if you have them the other way around)

➤ Copy (DO NOT CUT) the two columns.

➤ Click on the **Insert Worksheet** tab at the bottom of the window and paste the columns into the new worksheet.

➤ Return to the main worksheet and delete the **Label** column.

Warning! The main worksheet should now have the LabelID column but not the Label column and the new worksheet should now have both the LabelID and Label columns. If this is not the case, judicial use of the Undo button would be a good idea!

2.5 To add the ArtistID field and add unique identifiers for each artist:

➤ Do exactly the same as you did for LabelID in sections 2.2 and 2.3, remembering to name the new column **ArtistID** – and don't forget to sort by **Artist** first.

Note	You should end up giving Wishbone Ash the unique identifier of 48 – if you have a different number, go back and double-check what you have done.

2.6 To split the Label data from the main set of data:

➢ Highlight the **ArtistID** and **Artist** columns.

➢

Figure 2 - highlight the ArtistID and Artist columns thus (it doesn't matter if you them the other way around)

➢ Copy (DO NOT CUT) the two columns.

➢ Click on the **Insert Worksheet** tab at the bottom of the window (not on Sheet1, which contains the label columns) and paste the columns into the new worksheet.

➢ Return to the main worksheet and delete the **Artist** column.

Warning!	The main worksheet should now have the ArtistID column but not the Artist column and the new worksheet should now have both the ArtistID and Artist columns. If this is not the case, as before, judicial use of the Undo button would be a good idea!

Figure 3 – how the main datasheet should now look

Remove duplicate values from Label and Artist worksheets

Note   Where there is duplication, this is called data redundancy. In a relational database, you will only have to enter the label and artist details once, instead of over and over.

2.7   To remove duplicate values from the Label worksheet:

➢ Open the worksheet where you pasted the label columns and select all of the rows in both columns, including the header rows.

➢ In the **Data** tab in the Ribbon select **Advanced** from the **Sort &Filter** section.

➢ Select **Filter the list, in place** and tick the **Unique records only** box, followed by **OK**.

➢ Keeping the selection highlighted, copy the selection and open a new worksheet.

➢ Paste the selection into the new worksheet and delete the old worksheet.

2.8   To remove duplicate values from the Artist worksheet:

➢ Do the same as above for **ArtistID** and **Artist**.

Split the Artist column into two separate columns

Note   With large data sets there are ways to automate the splitting of one field into two or more separate fields, though all ways will have inherent problems to some extent. The following method is fine with small data sets, but with larger data sets you will probably want to explore Excel further – try Googling to see what options exist within Excel.

2.9   To split the Artist column into two separate columns:

ArtistID	ArtistName1	ArtistName2
1	ALAN	STIVELL
2		ATOMIC ROOSTER
3	CAT	STEVENS
4	CHRIS	MILLER & KEN CAMPBEL
5	DAVID	ROSE & HIS ORCHESTR
6		DR FEELGOOD
7	ELIZABETH	PAGE & THE HOLBOURI
8		FAMILY

37	THE	GREENE STRING QUAR
38	THE	MOODY BLUES
39	THE	MOONFLOWERS
40	THE	MOTORS
41	THE	SEARCHERS

Figure 4 – how the worksheet with artist details should look when completed

➢ In the new worksheet that includes the **ArtistID** and **Artist** columns highlight the **Artist** column.

➢ Copy the column, click into cell **C1** and paste the column so that there are two columns exactly the same side by side.

➢ Rename the first **Artist** column as ***ArtistName1*** and the second as ***ArtistName2***.

➢ Edit both columns so that artists' first names and the word "The", if preceding a group name, only appear in **ArtistName1**.

➢ Edit both columns so that group names and artist surnames appear only in **ArtistName2**.

Note Jethro Tull is a group name, not a person.

Task 3 Create Access tables from Excel data

Objectives To import the data into three tables in a new, blank Access database.

Comments Just in case everything went horribly wrong during the previous tasks, there is a cleansed spreadsheet that you can use, named PlanningYourDatabaseCleansed.xlsx. If all went well, continue using the file from the previous task.

3.1 To begin the import process:

Note The assumption is that you know how to create a new database in Access. If you don't, see the document Access 2010: Building Access databases.

> ➢ Close and save the **Excel** file, then open **Access** and create a new, blank database and name it as ***RecordDatabase.accdb***.

> ➢ Close the open table that Access annoyingly opens automatically, go to the **External Data** tab and select **Excel**.

> ➢ Browse to the **Excel** file, leave the defaults as they are and click on **OK**.

> ➢ When the wizard starts select the first sheet (probably called **qryPlanningYourDatabase** or similar) and click on **Next**.

> ➢ Ensure that **First Row Contains Field Names** is selected and click on **Next**.

> ➢ Highlight **ArtistID** by clicking on the fieldname and select ***Long Integer*** as the **Data Type**, then highlight **LabelID** and select ***Long Integer***.
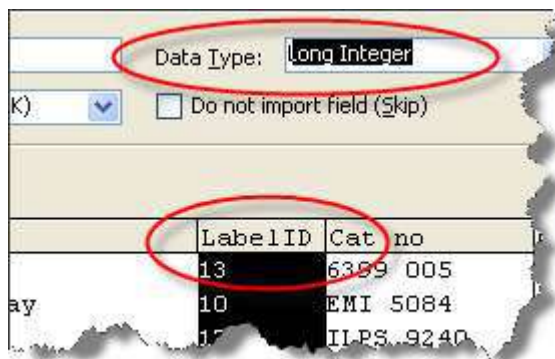

Figure 5 – click on LabelID and select Long Integer

> ➢ Highlight **Cost** and select ***Currency***, then click **Next**.

Note Both ArtistID and LabelID constitute foreign keys in the Record table, and both are to be linked to AutoNumber fields. They must be set as Long Integer, as this is the only format that can be linked to an AutoNumber field.

> ➢ Ensure that **Let Access add primary key** is selected and click **Next**.

> ➢ Name the table ***Record*** and click **Finish** followed by Close.

3.2 To view the imported data in the new table:

> ➢ In the **Database** window, double-click on **Record** to view the data, then close the table.

3.3 To import label data as a new table:

 ➢ In the **External Data** tab select **Excel**.

 ➢ Browse to the **Excel** file, leave the defaults as they are and click on **OK**.

 ➢ When the wizard starts select the first sheet (called either **Sheet1** or **Sheet3** depending on whether you are using the original Excel file or the if-it-all-went-horribly-wrong backup file) and click on **Next**.

 ➢ Ensure that **First Row Contains Fieldnames** is selected and select **"** as the **Text Qualifier**.

 ➢ Store to a new table and in **Field Options** ensure that the **Data Type** option for **LabelID** is set to *Long Integer*.

 ➢ Change the **Indexed** option for **LabelID** to *Yes (No Duplicates)*.

 ➢ When asked, select **Choose my own primary key** and choose *LabelID*.

 ➢ Save the table as *Label*, view the data in it by double-clicking on it in the **Database** window, then close the table.

3.4 To import artist data as a new table:

 ➢ Do exactly the same as above making sure that **ArtistID** is set to *Long Integer* and the **Indexed** option is set to *Yes (No Duplicates)*.

Note The sheet will be titled as either Sheet2 or Sheet4.

 ➢ Select **ArtistID** as the primary key and name the table as *Artist*.

Task 4 Add relationships and test the data

Objectives To add relationships to test the imported data.

Comments You'll see some obvious issues, such as only being able to view the foreign key code in the Record table rather than the more useful artist and label names, but we'll sort this out in the next task.

4.1 To add relationships so that the data can be tested:

Note The primary key fields for both Artist and Label tables have to be added manually each time a new record is added to either. It is possible to turn these fields into AutoNumber fields so they will be automatically filled in as each new record is added. The ability to do this is dependent on several factors – such as not adding or deleting any records before getting around to doing it (!) – and is outside the scope of this training course. Expert help is required to do this – please consult your IT Support Staff.

➢ Close any tables you have open, then select the **Database Tools** tab, select **Relationships**, add all three tables and then close the **Show Window** popup.
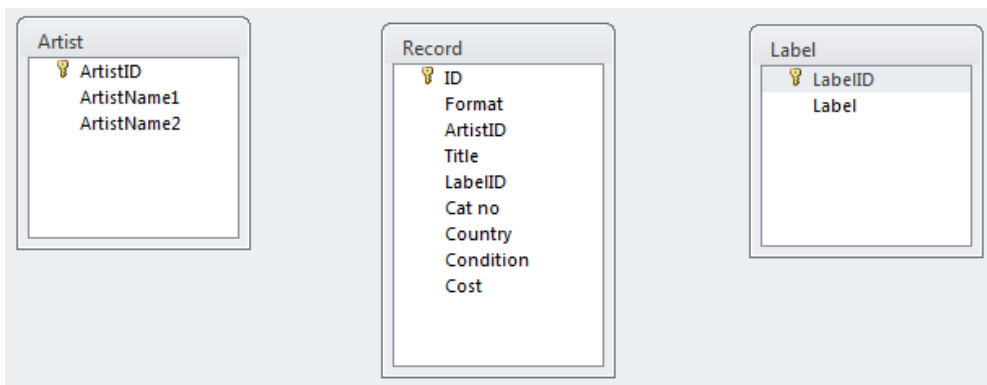

Figure 6 – the Relationships window should look similar to above

➢ On the **Artist** table, click on **ArtistID**, keep the mouse button pressed, then drag across to the **ArtistID** field on the **Record** table and let go.

➢ In the **Edit Relationships** dialogue that opens, double-check that the two named fields both say **ArtistID** – if not, click **Cancel** and try again.

➢ Tick **Enforce Referential Integrity** and then tick **Cascade Update Related Fields** and click **Create**.
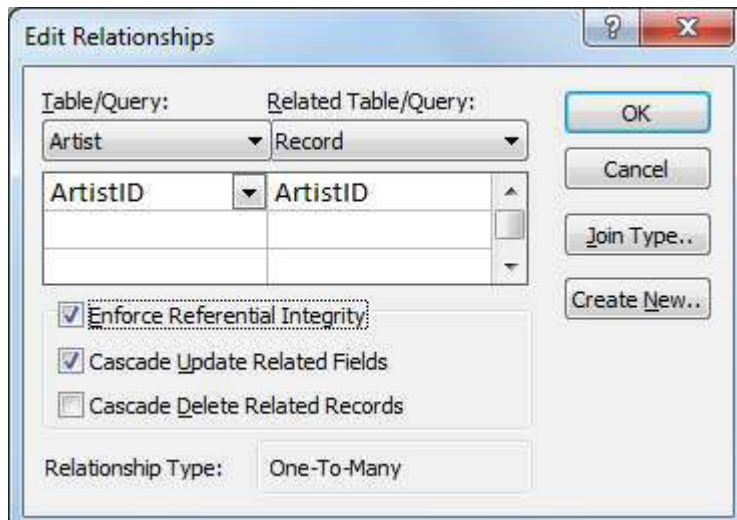
Figure 7 – tick Enforce Referential integrity and Cascade Update Related Fields

> ➢ Do the same for **LabelID** so that the relationships are the same as shown in Figure 8.
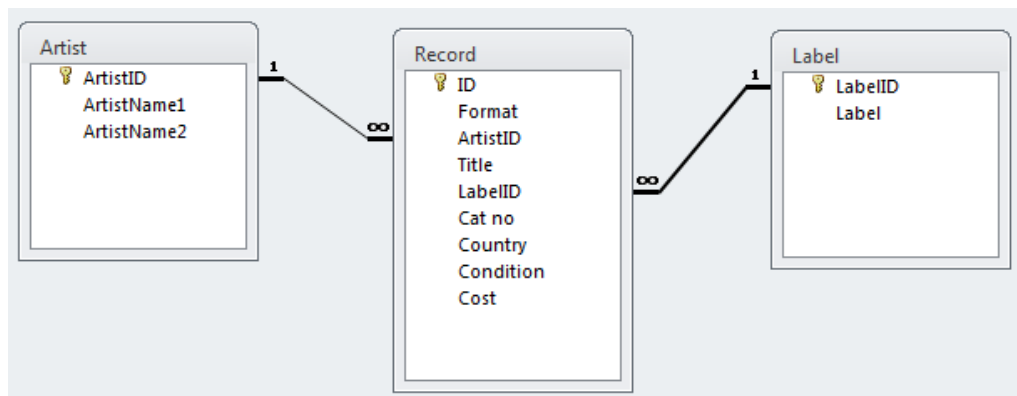

Figure 8 – how the Relationships window should look when finished

> ➢ Close the **Relationships** window and choose **Yes** when prompted whether you want to save the changes to the layout.

4.2 To see what adding relationships has done:

> ➢ Open the **Artist** table and notice the new column to the left of **ArtistID**.
>
> ➢ Click randomly on the small boxed **+** signs to see the related records (Jethro Tull has more than one related record).
>
> ➢ Do the same with the **Label** table (Virgin and Charisma have more than one related record).

Note There are obviously still some drawbacks with the tables as created and linked, but this is only the first stage of many in creating a user-friendly database. However, the first stage has been successfully completed – you did not have to retype any of the data from the original spreadsheet.

Task 5   Add dropdowns that show artist and label details

Objectives   To create dropdown lists in the Record table that show data from the Artist and Label tables. To sort the lists in alphabetical order

Comments   This is a two-stage process in that we have to first cerate what is called a lookup, then we have to sort it into alphabetical order to make it data entry-friendly. We also have to join ArtistName1 and ArtistName2 together.

5.1   To add dropdown lists to the foreign key fields in the Subject table:

➢ Open the **Record** table in **Design View** (you may have to click on the **Home** tab in the **Ribbon** to see this option) and highlight the **ArtistID** field.

➢ Under the **Field Properties**, select the **Lookup** tab.

➢ Select **Combo Box** from the dropdown list, click in **Row Source** and select **Artist**.
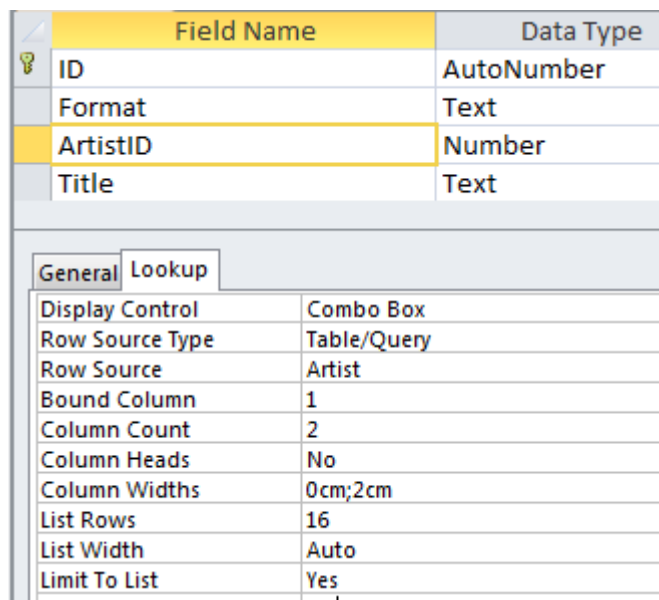

Figure 9 – make sure that you do this exactly as explained

➢ Update **Column Count** to *2*, then click in **Column Widths** and enter *0cm;2cm* - this means that the dropdown list will show only the data in the second field (the faculty name), but the data that is really being put into the foreign key field is from the first field (the faculty code) because this is the bound field.

➢ Look at the table in **Datasheet View** (save changes when prompted) and change one or two of the values in the **Artist** field using the new dropdown list – don't forget to change them back afterwards!

➢ Go back to **Design View** and select **LabelID**.

➢ Do the same as above, except choose **Label** as the **Row Source** and in **Column Widths**.

➢ Check the table in **Datasheet View** again, then close the **Record** table.

5.2 To join the two ArtistName fields together and list the names in alphabetical order:

➢ Open the **Record** table in **Design View**, highlight the **ArtistID** field and click on the **Lookup** tab in **Field Properties**.
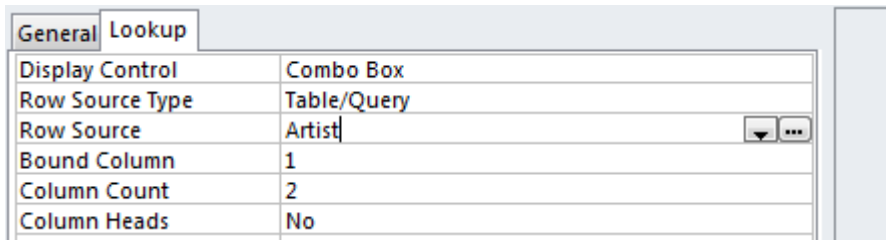
General Lookup	
Display Control	Combo Box
Row Source Type	Table/Query
Row Source	Artist
Bound Column	1
Column Count	2
Column Heads	No

Figure 10 – clicking on the three dots invokes the Query Builder

➢ Click in **Row Source** and notice the box with three dots that appears at the far-right – click on this and select **Yes** on the message box that pops up.

➢ Double-click on all three fields to add them to the **QBE Grid** in the same order as they appear in the table.

➢ In the **ArtistName2** column, click on **Sort** and select *Ascending*.

Note Make sure that you choose the correct field above – sorting by ArtistName1 will not be much use!

➢ Alter **ArtistName1** to read *ArtistName1 & " "& ArtistName2*.

➢ Close the query (save when prompted) and look at the contents of **Row Source** – the table name has been replaced with a SQL statement.

5.3 To list the labels in alphabetical order:

➢ Highlight the **LabelID** field, click on **Row Source** in the **Lookup** tab.

➢ Click on the button with three dots (and click on **Yes**) and pull down both fields into the query.

➢ Sort ascending by **Label**, then close and save the query.

5.4 To test the updated lookup:

➢ Switch to **Datasheet View**, saving when prompted, and look at the dropdown list.

➢ Select a record at random and clear the entry from the **LabelID** field.

➢ Enter *c* and notice that the dropdown list goes to the first label starting with "c".

Note: any forms created will have dropdowns sorted alphabetically because they inherit their properties from the table as at the point of creation.