Data Preparation 101: How To Use Query-By-Example To Get Your Research Dataset Ready For Primetime

Paul J. Lazarony, California State University, Northridge, USA Donna A. Driscoll, California State University, Northridge, USA

ABSTRACT

Researchers are often distressed to discover that the data they wanted to use in their landmark study is not configured in a way that is usable by a Statistical Analysis Software Package (SASP). For example, the data needed may come from two or more sources and it may not be clear to the researcher how to get them combined into one analyzable dataset. Fortunately, there is hope. The query facility within database management software (DBMS), such as Microsoft Access, is particularly well-equipped to do the data reconfigurations necessary to get datasets ready for a SASP. Unfortunately, most researchers have either never been exposed to a DBMS or are unaware of the powerful data-transforming queries that they can perform. The purpose of this paper is to introduce researchers to some very helpful and relatively easy to learn techniques for solving common dataset misconfiguration problems. It is presented in three sets of hands-on examples: (1) how to import a dataset from an electronic spreadsheet into a database table, (2) how to use an easy-to-learn DBMS facility called Query-By-Example (QBE) to perform specific data reconfiguration tasks, and (3) how to import the reconfigured dataset from the DBMS into a SASP. The software used is MS Excel, MS Access, and SPSS (originally called Statistical Package for the Social Sciences). The examples are presented step-by-step so that the reader can follow along using files downloadable from the authors' website

Keywords: Database; Dataset; Query; Research; Spreadsheet; Statistical Analysis

INTRODUCTION

he authors of this paper are information systems faculty who have discovered that database techniques can be very helpful in dealing with common problems associated with research datasets. This paper is the second in a series that attempts to make these techniques more accessible to researchers in other fields.

Reconfiguration of datasets in a database can be approached in two basic ways: (a) Structured Query Language (SQL) and (b) Query-By-Example (QBE). SQL is a high-level computer language that involves writing computer code to do queries of a database; QBE allows the user to formulate queries using a graphical user interface (which allows the use of dropdown menus, toolbars, and wizards) and is thus much more user-friendly.

Our first paper in this series focused on the first of these approaches. Specifically, we used <u>SQL</u> to efficiently correct, recode, and reorganize a dataset downloaded from <u>one</u> table that was embedded in a web page. The current paper focuses on the second approach. Specifically, we will demonstrate how to use <u>QBE</u> to combine data stored in <u>multiple</u> tables in a variety of useful ways.

This paper is presented in three phases, each of which includes hands-on examples. In **Phase I**, we demonstrate the steps required to import a dataset stored in an **MS Excel** file into an **MS Access** database table. Then, in **Phase II**, we take the reader through a series of five examples that illustrate how to combine data that exist



1

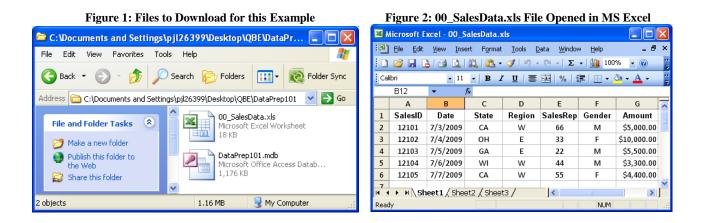
in multiple tables. Finally, in **Phase III**, we show how to import the newly combined dataset from an **MS Access** table into **SPSS** for statistical analysis.

As previously mentioned, this example is designed so that the reader may follow along step-by-step. The Web page containing the data can be found at www.csun.edu/~pjl26399/dataprep101. The following software will also be required to complete the example: MS Excel, MS Access, and SPSS. If SPSS is not available, Phase III could also be accomplished using another statistical analysis package such as SAS or Minitab. The dataset could also be returned to MS Excel for statistical analysis.

PHASE I: IMPORTING A DATASET FROM MS EXCEL INTO AN MS ACCESS TABLE

Although datasets could come to a researcher in a variety of forms, one of the most commonly used current formats is an **MS Excel** file. Depending on the particular dataset and the complexity of the required analysis, a researcher may sometimes choose to leave the data in **MS Excel** and simply do the analysis right there. At other times, however, it could be difficult to use the data in its original form. The following steps illustrate the ease of importing an **MS Excel** file into an **MS Access** table:

- A. Download the **DataPrep101.mdb** and **00_SalesData.xls** files from the Web page shown above. Figure 1 shows the files after download. Be sure to save them in a new folder on your computer's hard drive, named **DataPrep101**.
- B. Open the **00_SalesData.xls** in **MS Excel**. The file should look like Figure 2 below:



- C. The goal is to import the **00_SalesData.xls** spreadsheet file into a table in the **DataPrep101.mdb** database file. To complete this task, follow the steps below:
- 1. Open the **DataPrep101.mdb** file in MS Access. Your screen will look similar to Figure 3 below. Notice the Tables and Queries listed in on the left-hand side of the screen. We will be using these objects later in this paper (in **Phase II**).
- 2. Notice the **Security Warning** in the middle of the screen. Click the **Options** button. The Microsoft Office Security Options dialog box will appear. See Figure 4.
- 3. The default is to leave the database file as read-only. Since changes to the database will be necessary, will need to enable all content. Click the radio button next to the **Enable this content option**. Then click **OK**. You will need to repeat these steps (2 & 3) each time you open the database file.
- 4. To import the **00_SalesData.xls** spreadsheet file, click the **External Data** tab, then click the Excel button on the toolbar. The Get External Data-Excel Spreadsheet dialog box will appear. See Figure 5.



Figure 3: DataPrep101.mdb File Opened in MS Access

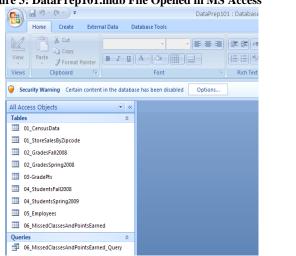




Figure 5: Database Window with Get External Data-Excel Spreadsheet Dialog Box

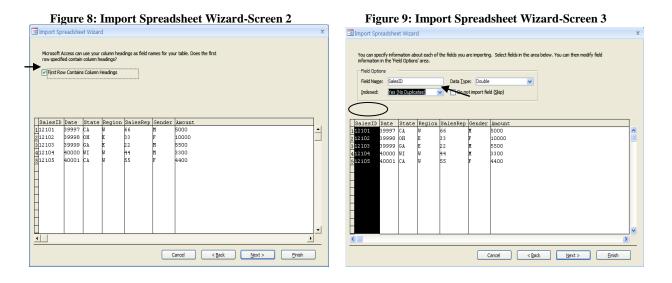
	DataPrep101 : Database (Access 2002 - 2003 file format) - Microsoft Access
Home Create External Data	Database Tools
Saved Imports Access Excel SharePoint List More ~	e Saved Excel SharePoint Text File Create Manage Work Synchronize Move to
Import	Get External Data - Excel Spreadsheet ? 🔀
All Access Objects 📀	Select the source and destination of the data
Tables	
01_CensusData	Specify the source of the data.
01_StoreSalesByZipcode	
02_GradesFall2008	Ele name: D:\ Browse
02_GradesSpring2008	
03-GradePts	Specify how and where you want to store the data in the current database.
04_StudentsFall2008	Import the source data into a new table in the current database.
04_StudentsSpring2009	If the specified table does not exist, Access will create it. If the specified table already exists, Access might overwrite its contents
05_Employees	with the imported data. Changes made to the source data will not be reflected in the database.
06_MissedClassesAndPointsEarned	Append a copy of the records to the table: 01_CensusData
Queries	If the specified table exists, Access will add the records to the table. If the table does not exist, Access will create it. Changes made to the source data will not be reflected in the database.
06_MissedClassesAndPointsEarned_Query	
	Link to the data source by creating a linked table.
	Access will create a table that will maintain a link to the source data in Excel. Changes made to the source data in Excel will be reflected in the linked table. However, the source data cannot be changed from within Access.
	OK Cancel

- 5. Click the **Browse** button. Locate the **00_SalesData.xls** spreadsheet file; click **Open**. The **Get External Database** dialog box should now look similar to Figure 6.
- 6. Leave the radio button checked next to **Import the source data into a new table in the current database**. Click **OK**. The **Import Spreadsheet Wizard** will appear (see Figure 7).



Figure 6: Get External Data Dialog Box-Screen 2	Figure 7: Import Spreadsheet Wizard-Screen 1	
Get External Data - Excel Spreadsheet 🔹 🛛 🔀	Import Spreadsheet Wizard	х
Select the source and destination of the data	Your spreadsheet file contains more than one worksheet or range. Which worksheet or range would you like?	
Specify the source of the data. Ele name: C:/Documents and Settings/p/26399/Desktop/Q8E/DataPrep101/00_Sales/Data.xis Browse	Show Worksheets Show Named Banges Show Named Banges	
Specify how and where you want to store the data in the current database.		
 Import the source data into a new table in the current database. If the specific table does not solv, Access will act it. If the specific table aready exists, Access singht overwrite its contents with the imported data. Charges made to the source data will not be reflected in the database. Append a copy of the records to the table: 		

- 7. Notice that the all of the column headings and data from the **00_SalesData.xls** spreadsheet file are displayed in the wizard. Click **Next**. The second screen of the wizard appears (see Figure 8).
- 8. This second screen assumes that the first row of data from our spreadsheet file contains column headings. Since this assumption is correct, leave the checkmark next to **First Row Contains Column Headings**. Click **Next**. The third screen of the wizard appears (see Figure 9).



- 9. This third screen of the wizard allows you to change the data type of each field (column), if needed, as well as indicate how the table will be indexed. In our example, the only change we need is to have the SalesID column be set as a unique key field (so that it might later be used to link to a SalesID field in another table). To accomplish this, first click on the **SalesID** column heading. Second, click the down arrow next to **Indexed** and select **Yes (No Duplicates)**. Click **Next**. The fourth screen of the wizard appears (see Figure 10).
- 10. Click the radio button next to **Choose my own primary key**. This will select the **SalesID** as the primary key (or unique field) for this table. Click **Next**. The fifth and final screen of the wizard appears (see Figure 11).



Figure 10: Import Spreadsheet Wizard-Screen 4	Figure 11: Import Spreadsheet Wizard-Screen 5
Import Spreadsheet Wizard	Import Spreadsheet Wizard
Morosoft Access recommends that you define a primary key for your new table. A primary key is used to unquely identify each record in your table. It allows you to retrieve data more quickly. I are are an Compose any own primary key. Origonee my own primary key. Origonee my own primary key.	That's all the Information the wizard needs to import your data. Import to Table: 0Selectiva
SalesID Date State Region SalesRep Gender Amount 11101 3997 KA # 66 M S000 2 12102 3998 DH E 33 F 10000 3 1 3 1 3 1 <td></td>	
	I would like a wizard to gnalyze my table after importing the data.
Cancel (Back (Best > Enish	Cancel (gad Birst) Brish

11. In the **Import to Table** text box, type in the new table name: **00_SalesData**. Click **Finish**. In the next dialog box, click **Close**. Congratulations, the **00_SalesData** table now appears in the list of tables of the left-side of the screen. Double click the new table name (**00_SalesData**) and the new table appears on the right (see Figure 12).

Figure 12: Database File Showing the Newly Imported 00_SalesData Table and Dataset

	U		• •		(Access 2002 -	2003 file for		oft Access
	atabase Tools	Datasheet						
Saved Imports	Saved Excel	SharePoint	ext File 1ore * E-	eate Manage mail Replies Collect Data	Work Synchi Online	💷 🖓 Cac		Move to SharePoint
All Access Objects 📀 «	00_SalesD	ata						
Tables	🔺 SalesID	🔹 Date 👻	State 👻	Region 👻	SalesRep 👻	Gender 👻	Amount 👻	Add New Field
00_SalesData	121	01 7/3/2009	CA	W	66	М	\$5,000.00	
01_CensusData	121			E	33	F	\$10,000.00	
01_StoreSalesByZipcode	121			E	22		\$5,500.00	
02 GradesFall2008	121			W	44		\$3,300.00	
02 GradesSpring2008	121	05 7/7/2009	CA	W	55	F	\$4,400.00	
03-GradePts	*							
04_StudentsSpring2009								
05_Employees								
06_MissedClassesAndPointsEarned								
Queries *								
06_MissedClassesAndPointsEarned_Query								

- 12. To close the table, just press the **X** in the upper right corner of the table.
- 13. **Phase I** is now complete. The reader may now continue on to **Phase II** or close the **DataPrep101.mdb** file.

PHASE II: FIVE EXAMPLES OF HOW TO USE MS ACCESS' QUERY BY EXAMPLE TO RECONFIGURE DATASETS

Although the QBE facility in MS Access can be used to handle most types of data misconfigurations, some situations tend to occur fairly frequently and, unfortunately, without the use of such database techniques, can be either complete barriers to further progress on a project or a problem that ends up being dealt with by the brute force approach of fixing the problems by hand at a substantial cost in time, money, or both.



In **Phase II**, we focus on five relatively common data problems that can be solved using QBE in a tiny fraction of the time it would take to do it the old-fashioned (and labor-intensive) way:

- A. combining the columns from two related tables (using a **JOIN** query),
- B. adding the rows in one table into another table (using an **APPEND** query),
- C. recoding a column in one table (using data stored in a second **CONVERSION** table),
- D. dealing with tables having overlapping data (using **UNMATCHED** and **APPEND** queries), and
- E. allowing the matching of values in one column that reference the values of another column in the same table (using a **SELF-JOIN** query).

These five situations are summarized in Table 1, along with a simple example of each that will then be described in detail. The reader is encouraged to work through these examples using the same database available from our website. (Please see the Introduction section of this paper for the file location.)

Situation Type		Example				
	Description	Illustration				
JOIN query The columns of data might be in two separate datasets that have one common linking	One dataset of census data by zipcode and another of store sales data by store# which includes the zipcode for each store.	Census Data: Zipcode Average Household Income Store Sales by Zipcode:				
column.	-	Store # Sales Zipcode				
APPEND query There might be two sets of rows having column sequences that are different from each other and that need to be combined into one larger dataset.	Two datasets of grade data in which the column names and column order are different and need to be aggregated into one comprehensive dataset.	Grades Fall 2008: SemesterID SectionNo StudentID CourseNo CourseTitle LetterGrade Grades Spring 2009: SemesterID SemesterID StudentNum TicketNo CourseNo ClassName Grade				
CONVERSION query One or more of the existing columns is not in analyzable form and needs to be recoded.	A grades dataset uses letters for grades, but the planned analysis requires numeric grade points. A second dataset containing letter grades and the corresponding grade points. The letter grades need to be converted into grade points.	Grades Fall 2008: SemesterID SectionNo StudentID CourseNo CourseTitle LetterGrade GradePts: LetterGrade GradePt				
Combining overlapping lists:UNMATCHEDandAPPEND queries.There may be two sets of rows	Two lists of students for two different semesters need to be combined into one comprehensive list of unique students which does	Students in Major Spring 2008: StudentID LastName FirstName				
places, or things existing at	not include any duplicate records.	Students in Major Fall 2008:				
two different points in time that need to be combined into one comprehensive list.		StudentID LastName FirstName				
SELF-JOIN query There might be two columns in the same dataset in which the values in one column reference the values in another.	The researcher wants to test whether an employee's stated opinion about the importance of home recycling is affected by his manager's stated opinion of home recycling. The available employee dataset includes columns of Employee ID, Manager's Employee ID, and Employee Home Recycling Importance, measured on a Likert- scale. A new Manager's Home Recycling Importance column	Employee:EmployeeIDLastNameFirstNameManagerIDHome Recycling Importance10GarciaMonica120SmithMary10130DoeJohn10140WongSteven550NorthCarla40560RobinsonKazetta405				
	The columns of data might be in two separate datasets that have one common linking column. APPEND query There might be two sets of rows having column sequences that are different from each other and that need to be combined into one larger dataset. CONVERSION query One or more of the existing columns is not in analyzable form and needs to be recoded. Combining overlapping lists: UNMATCHED and APPEND queries. There may be two sets of rows representing lists of people, places, or things existing at two different points in time that need to be combined into one comprehensive list. SELF-JOIN query There might be two columns in the same dataset in which the values in one column	Join query The columns of data might be in two separate datasets that have one common linking column.One dataset of census data by zipcode and another of store sales data by store# which includes the zipcode for each store. column and state to be combined into one larger dataset.APPEND query There might be two sets of other and that need to be combined into one larger dataset.Two datasets of grade data in which the column names and column order are different and need to be aggregated into one comprehensive dataset.CONVERSION query One or more of the existing columns is not in analyzable form and needs to be recoded.A grades dataset uses letters for grades, but the planned analysis requires numeric grade points. A second dataset containing letter grades and the corresponding grade points. The letter grades need to be converted into grade points.Combining overlapping lists: UNMATCHED and APPEND queries. There may be two sets of rows representing lists of people, places, or things existing at two different points in time that need to be combined into one comprehensive list.Two lists of students for two different semesters need to be combined into one comprehensive list.SELF-JOIN query There might be two columns in the same dataset in which the values in one column reference the values in another.The researcher wants to test whether an employee's stated opinion about the importance of home recycling. The available employee dataset includes columns of Employee ID, Manager's Employee ID, and Employee Home Recycling ID, and Employee Home Recyclin				

Table 1: Five Database Problems that can be Solved using QBE



- A. Combining the Columns from Two Related Tables (using a JOIN query).
 - Open the DataPrep101.mdb file. Be sure to turn off the security features (see Figures 3 & 4). The following tables will be used for this example: (1) 01_StoreSalesByZipCode and (2) 01_CensusData. Figures 13 and 14 show the datasets for these two tables. Notice that the two tables have a common column, zip code. The goal will be to have a one dataset with the columns of Store#, Sales, Zip, and AvgHouseholdIncome.

	Figure 13: 01_StoreSalesByZipCode Table									
	101_StoreSalesByZipcode									
	Store#	*	Sales 👻	Zip	-					
		1	\$10,500,000	20000						
		2	\$21,680,000	30000						
		3	\$15,583,000	20000						
		4	\$26,400,700	30000						
*	(Ne	w)								

	Figure 14: 01_CensusData Table								
	01_CensusD	ata							
	Zipcode	-	AvgHouseholdIncome 👻						
	20000		\$70,000						
	30000		\$80,000						
*									

2. A JOIN query will be used to achieve this goal. A JOIN is a specific form of SELECT query that combines the data from two or more tables based upon common fields. Here's the process: Click the **Create** tab, then click on the **Query Design** button on the toolbar (see Figure 15). The **Show Table** dialog box appears (see Figure 16).

Figure 15: Creating a Query Using the Query Design Facility

8	19 -	(°" -) = _					DataF	rep101 :	Databa	se (Access 2002	- 2003 file	e format) - N	licrosoft Ad
9	Home (Create	External	Data	Databa	ise Tools							
Table	Table Templates	SharePoint • Lists •	Table Design	Form	Split Form	<u></u>	PivotChart	Form Design	Report	 Labels Blank Report Report Wizard 	Report Design	Query Que Wizard Desig	ry Macro
	Ta	bles					orms			Reports		Oth	er

Figure 16: Show Table Dialog Box

Show Ta	ble						?	Þ	<
Tables	Queries	Both				 			1
01_Cer 01_Sto 02_Gra 02_Gra 03-Gra 04_Stu 04_Stu 05_Em	nsusData reSalesBy desFall20 desSpring dePts dentsFall2 dentsSprii	08 2008 2008 1g2009	ntsEarr	ned					
			(Add		lose		J

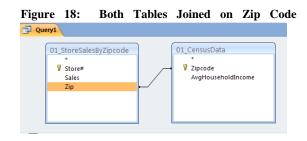


- In the Show Table dialog box, select 01_StoreSalesByZipcode table; click Add. Next, select the 01_CensusData table; click Add. Click Close to close the dialog box. Your screen should now look like Figure 17.
- 4. The next step is to JOIN the two tables on the common field of zip code. This JOIN is imperative to the goal of creating one dataset. Click and drag the **Zip** field from the **01_StoreSalesByZipcode** table to the **Zipcode** field of the **01_CensusData** table. Your screen should now look like Figure 18.

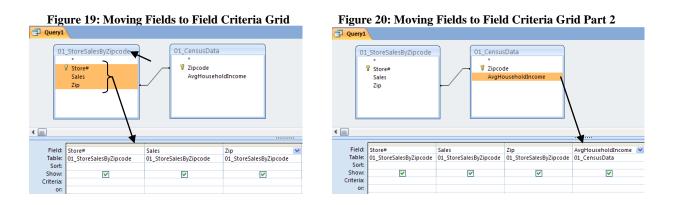
rigure	1/.	Query	Design	VIEW	with	Dom	Table
🗗 Query1							
8	StoreSales * Store# Sales Zip	ByZipc	01_Cens * Zipc Avgi		come		
▲ []							
Field: Table: Sort: Show: Criteria: or:				E]		

Query Design View with Both Tables

Figure 17.



- 5. Now the fields from the tables need to be brought down to the field criteria grid. Double click the title bar of the **01_StoreSalesByZipcode** table box. Click and drag the highlighted fields from the table box down to the field criteria grid (see Figure 19).
- 6. From the **01_CensusData** table box, click and drag the **AvgHouseholdIncome** field to the field criteria grid. The **Zipcode** field does not need to be brought down, as it would only duplicate the zip codes in two columns. Your screen should now look like Figure 20.



7. The query is ready to be run. To accomplish this, click the **Run** (!) button on the toolbar (see Figure 21). Figure 22 shows the query results with data from both of the tables. *Phase III* will demonstrate how to import this new dataset into **SPSS**.



Figure 21: The Query Run Button



	Figure 22: Query Results with Data from Both Tables										
i.	Query1										
	Store#	-	Sales 👻	Zip 👻	AvgHouseholdIncome 🕞						
		1	\$10,500,000	20000	\$70,000						
		3	\$15,583,000	20000	\$70,000						
		2	\$21,680,000	30000	\$80,000						
		4	\$26,400,700	30000	\$80,000						

Click the Save button (see Figure 23). The Save As dialog box appears. Name this new query: 01_Join. Click OK. The new query now shows up on the query list on the left-hand side of the screen (see Figure 24).

Figure 23: Q	Figure 24: 01_Join in Query List								
■ 1 · (1 ·) +	DataPrep101 : Database (Access 2002 - 2003 file								
Home Create External Data	Database Tools	Tables *							
Cut		00_SalesData							
Calibri	 11 · ■ 書 書 譯 課 / · · · · · · · · · · · · · · · · · ·	01_CensusData							
View Paste Format Painter		01_StoreSalesByZipcode							
Views Clipboard 🕞	Font 🕞 Rich Text Records	02_GradesFall2008							
All Access Objects 💿 <	« 🔁 Query1	02_GradesSpring2008							
Tables	Store# Sales Zip AvgHouseholdIr	03-GradePts							
00_SalesData	1 \$10,500,000 20000	04_StudentsFall2008							
01_CensusData	3 Save As	04_StudentsSpring2009							
01_StoreSalesByZipcode	4 Query Name:	05_Employees							
02_GradesFall2008	* (New) 01_Join	06_MissedClassesAndPointsEarned							
02_GradesSpring2008		Queries ×							
03-GradePts	OK Cancel	1 01_Join							
04_StudentsFall2008		06_MissedClassesAndPointsEarned_Query							

9. Part A is now complete. The reader may now continue on to part B or close the **DataPrep101.mdb** file.

B. Adding the Rows in One Table into Another Table (using an APPEND query).

Open the DataPrep101.mdb file. Be sure to turn off the security features (see Figures 3 & 4). The following tables will be used for this example: (1) 02_GradesSpring2008 and (2) 02_GradesFall2008. Figures 25 and 26 show the datasets for these two tables. Notice that the two tables have all fields matching, however they are not in the same order, nor do all of the column headings match. The goal will be to have a one new table (named: 02_GradesAll) that includes all four records for 2008.

	02_GradesSpring2008									
	SemesterID 👻	SectionNo 🝷	StudentID 🝷	CourseNo 👻	CourseTitle 🚽	LetterGrade 👻				
	20081	12456	384473999	SOC100	Principles of Sociology	В				
	20081	23500	245894533	ACCT200	Financial Accounting	C				
*										

Figure 25: 02_GradesSpring2008 Table

ŀ	'igure 26: 02_(GradesFall2008	3 Table	

	U2_GradesFall2008										
	SemesterID 👻	StudentNum 👻	TicketNo 👻	CourseNo 👻	ClassName 🔹	Grade 👻					
	20082	269475553	22679	EDUC350	Educational Psychology	B-					
	20082	684688392	45833	MATH200	College Algebra	C+					
*											

© 2011 The Clute Institute

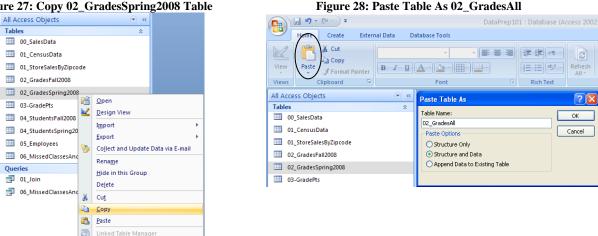
김 للاستشارات

- 2. Let's say that we want to be able to create the complete list of grades while leaving both of the original semester tables intact. We can accomplish this by first copying one of the two tables, say Spring, and then inserting duplicates of the fall records into the new table. Here's how it goes: The first step is to copy the **02_GradesSpring2008** table. On the left-hand side of the screen, under the Tables list, locate the **02** GradesSpring2008 table name (see Figure 27). Right click on it to make the shortcut menu appear. Choose Copy from the list.
- 3. Click the **Paste** button (see Figure 28), the **Paste Table As** dialog box appears. Name the new table: 02_GradesAll, then click OK. The 02_GradesAll table now shows up on the Tables list on the left-hand side of the screen. Remember, this table only includes the two records from the 02_GradesSpring2008 table. The next step is to add the two records from the 02 GradesFall2008.

Figure 27: Copy 02_GradesSpring2008 Table

Ta<u>b</u>le Properties

1



4. An APPEND query will be used to achieve this goal. Click the Create tab, then click on the Query Design button on the toolbar (see Figure 15). The Show Table dialog box appears, click Close. Click the Append button on the toolbar (see Figure 29). The Append dialog box appears. Choose the **02** GradesAll table as the file to append (or add records to), click OK.

Figure 29:	Append to 02_GradesAll Table
	Query Tools DataPrep101 : Database (Access 2002 - 2003 file f
	atabase Tools Design
SQL ! .	📰 🗙 🛈 Union 🕂 🖓 🖅 Insert Columns
	🛄 🌕 🎯 Pass-Through 📃 🔿 Delete Rows 🐰 Delete Columns
View Run Select Make Append Update Cr Table	rosstab Delete 🖉 Data Definition Show 🏠 Builder 🏢 Return: All 🔻
Results Quen	y Type Query Setup
All Access Objects 💿 «	P Query1
Tables	Append ?X
00_SalesData	
01_CensusData	
01_StoreSalesByZipcode	Table Name: 02_GradesAll ⊙ Current Da 02_GradesAll
02_GradesAll	O Current Da do La GodesFall2008 O Another Da 02_GradesFall2008
02_GradesFall2008	Eile Name: 03-GradePts
02_GradesSpring2008	04_StudentsFall2008 04_StudentsSpring2009
03-GradePts	

- 5. Click the **Show Table** button on the toolbar (see Figure 30); the **Show Table** dialog box appears. Highlight the **02** GradesFall2008 table from the list, click Add.
- 6. The next step is to add the fields from the **02_GradesFall2008** table box to the field criteria grid (see Figure 31). Double click the title bar of the 02_GradesFall2008 table box. Click and drag

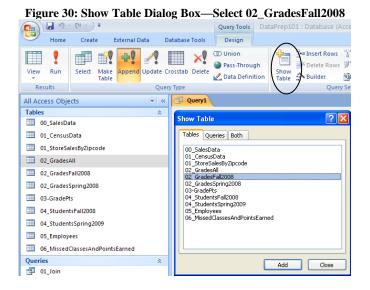


© 2011 The Clute Institute

www.manaraa.com

the highlighted fields from the table box down to the field criteria grid. Notice that only the **SemesterID** and **CourseNo** fields are currently matching up with the fields in the **02_GradesAll** table because the fields were not named consistently for the two semesters. Now the remaining four fields in **02_GradesFall2008** need to be manually matched up with the corresponding field names in **02_GradesAll**.

7. On the **Append To** row (see Figure 32), click the cell under **StudentNum**, click the **Down Arrow** and choose **StudentID** from the list. Continue with the three remaining field pair (**TicketNo** =**SectionNo**, **ClassName** = **CourseTitle**, and **Grade** = **LetterGrade**).





	GradesFall2008					
◀ []		4				
Field:	SemesterID	StudentNum	TicketNo	CourseNo	ClassName	Grade 💌
Table:	02_GradesFall2008	02_GradesFall2008	02_GradesFall2008	02_GradesFall2008	02_GradesFall2008	02_GradesFall2008
Sort:						
	SemesterID			CourseNo		
Criteria:						
or:						



02	2_GradesFall2008					
	*					
	SemesterID					
	StudentNum FicketNo					
	CourseNo					
	ClassName					
	Grade					
	onade					
< [m]						
4						
Field:	SemesterID	StudentNum	TicketNo	CourseNo	ClassName	Grade
Field: Table:	02_GradesFall2008	StudentNum 02_GradesFall2008	TicketNo 02_GradesFall2008		ClassName 02_GradesFall2008	
Field: Table: Sort:	02_GradesFall2008	02_GradesFall2008	02_GradesFall2008	CourseNo 02_GradesFall2008	02_GradesFall2008	02_GradesFall2008
Field: Table: Sort: Append To:	02_GradesFall2008 SemesterID			CourseNo		02_GradesFall2008
Field: Table: Sort: Append To: Criteria:	02_GradesFall2008 SemesterID	02_GradesFall2008	02_GradesFall2008	CourseNo 02_GradesFall2008	02_GradesFall2008	02_GradesFall2008 LetterGrade 02_GradesAll.*
Field: Table: Sort: Append To:	02_GradesFall2008 SemesterID	02_GradesFall2008	02_GradesFall2008	CourseNo 02_GradesFall2008	02_GradesFall2008	02_GradesFall2008 LetterGrade 02_GradesAll.* SemesterID
Field: Table: Sort: Append To: Criteria:	02_GradesFall2008 SemesterID	02_GradesFall2008	02_GradesFall2008	CourseNo 02_GradesFall2008	02_GradesFall2008	02_GradesFall2008 LetterGrade 02_GradesAll.* SemesterID SectionNo
Field: Table: Sort: Append To: Criteria:	02_GradesFall2008 SemesterID	02_GradesFall2008	02_GradesFall2008	CourseNo 02_GradesFall2008	02_GradesFall2008	02_GradesFall2008 LetterGrade 02_GradesAll.* SemesterID
Field: Table: Sort: Append To: Criteria:	02_GradesFall2008 SemesterID	02_GradesFall2008	02_GradesFall2008	CourseNo 02_GradesFall2008	02_GradesFall2008	02_GradesFall2008 Letter/Grade 02_GradesAll.* SemesterID SectionNo StudentID

Figure 32: On the Append Row, Select the Matching Field Names

- 8. The final step is to run this APPEND query. Click **Run** (!). A dialog box appears warning that "You are about to append 2 rows (see Figure 33). Since this is what we want to do, click **Yes**.
- 9. Congratulations, you have successfully combined the two data sets. Click the **Save** button and save this query as **02_CombiningDataSets** (see Figure 34). Click **OK**.

Figure 33: You Are About to Append 2 Row(s) Dialog Box Microsoft Office Access You are about to append 2 row(s). Once you click Yes, you can't use the Undo command to reverse the changes. Are you sure you want to append the selected rows? Yes No

Figure 34: Save the Append Query

Save As		[? 🗙
Query Name:			
02_CombiningDataSets			
	ок	Can	cel

10. The **02_GradesAll** table now includes all four grade records from Spring and Fall 2008 semesters (see Figure 35).

Figure	e 35: App	ended 02_	GradesAll	Table

02_GradesAll					
SemesterID 👻	SectionNo 🕞	StudentID 👻	CourseNo 👻	CourseTitle 🔹	LetterGrade 👻
20081	12456	384473999	SOC100	Principles of Sociology	В
20081	23500	245894533	ACCT200	Financial Accounting	С
20082	22679	269475553	EDUC350	Educational Psychology	B-
20082	45833	684688392	MATH200	College Algebra	C+

11. Part B is now complete. The reader may now continue on to part C or close the **DataPrep101.mdb** file.

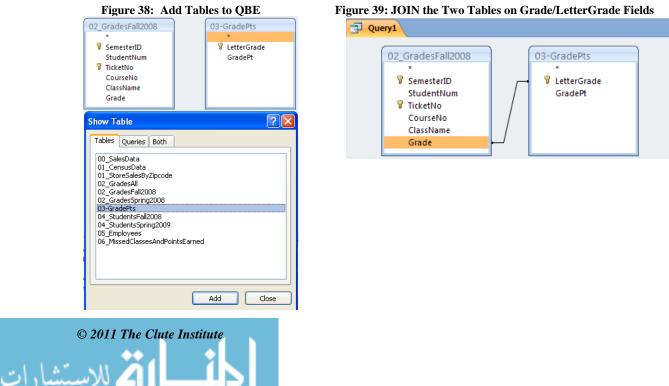


- С. Recoding a Column in One Table (using data stored in a second CONVERSION table).
 - 1. Open the DataPrep101.mdb file. Be sure to turn off the security features (see Figures 3 & 4). The following tables will be used for this example: (1) **02_GradesFall2008** and (2) **03_GradePts**. Figures 36 and 37 show the datasets for these two tables. Notice that the 02 GradesFall2008 table has a Grade column and the 03_GradePt table includes both a LetterGrade and a GradePt field. Our goal here will be to create one new dataset that contains all of the fields from the 02 GradesFall2008 table and also includes the GradePt column.

	Figure 36: 02_GradesFall 2008 Table										
	02_GradesFall2008										
Ζ	SemesterID 👻	StudentNum 👻	TicketNo 👻	CourseNo 🝷	ClassName 👻	Grade	-				
	20082	269475553	22679	EDUC350	Educational Psychology	B-					
	20082	684688392	45833	MATH200	College Algebra	C+					
*											

Figure 37: 03_GradePts Table										
	03-GradePts									
	etterGrade 👻	GradePt 👻								
	Δ.	4								
	A-	3.7								
	В	3								
	В-	2.7								
	В+	3.3								
	¢	2								
	C-	1.7								
	C+	2.3								
	Þ	1								
	p-	0.7								
	D+	1.3								
	Ł	0								

- 2. Click the **Create** tab, then click on the **Query Design** button on the toolbar (see Figure 15). The Show Table dialog box appears.
- 3. Choose the 02_GradesFall2008 table from the list; click Add (see Figure 38). Next, choose the 03_GradePts table from the list; Click Add. Click Close.
- The next step is to JOIN the two tables on the common field of Grade/LetterGrade (see Figure 39). 4. To do this, click and drag the Grade field from the 02 GradesFall2008 table to the LetterGrade field of the 03 GradePts table.



13

- 5. Now the fields from the tables need to be brought down to the field criteria grid. Double click the title bar of the **02_GradesFall2008** table box. Click and drag the highlighted fields from the table box down to the field criteria grid.
- 6. From the **03_GradePts** table box, click and drag the **GradePt** field to the field criteria grid. The **LetterGrade** field does not need to be brought down, as it would only duplicate the **Grade** field from the other table. Your screen should now look like Figure 40.

	Figure 40: Moving Fields to Field Criteria Grid									
Query1	C2_GradesFall2C * SemesterID StudentNum TicketNo CourseNo ClassName Grade		03-GradePts * VetterGrade GradePt							
4										
Field: Table: Sort:	SemesterID 02_GradesFall2008	StudentNum 02_GradesFall2008	TicketNo 02_GradesFall2008	CourseNo 02_GradesFall2008	ClassName 02_GradesFall2008	Grade 02_GradesFall2008	GradePt 03-GradePts			
Show: Criteria:	V			✓		V	V			

7. The final step is to run this query. Click **Run** (!). Congratulations, you have successfully used a CONVERSION table to combine the two datasets. The results of the query are in Figure 41. Click the **Save** button and save this query as **03_ConversionTable.** Click **OK**.

Figure 41	: Results	of the 03	_ConversionTal	ble Ouerv
I Igui C HI	. itcourto	or the ob_		Jie Query

	SemesterID 👻	StudentNum 👻	TicketNo 👻	CourseNo 👻	ClassName 👻	Grade 🚽	GradePt 👻	
	20082	269475553	22679	EDUC350	Educational Psychology	B-	2.7	
	20082	684688392	45833	MATH200	College Algebra	C+	2.3	
*								

- 8. Part C is now complete. The reader may now continue on to part D or close the **DataPrep101.mdb** file.
- D. Dealing with Tables Having Overlapping Data (using UNMATCHED and APPEND queries).
 - 1. Open the **DataPrep101.mdb** file. Be sure to turn off the security features (see Figures 3 & 4). The following tables will be used for this example: (1) **04_StudentsFall2008** and (2) **04_StudentsSpring2009**. Figures 42 and 43 show the datasets for these two tables. Notice that the two tables have exactly the same fields, however some of the rows are overlapping (i.e., showing in both tables). The ultimate goal is to have one complete list of unique students in a table named **04_StudentsAll**.

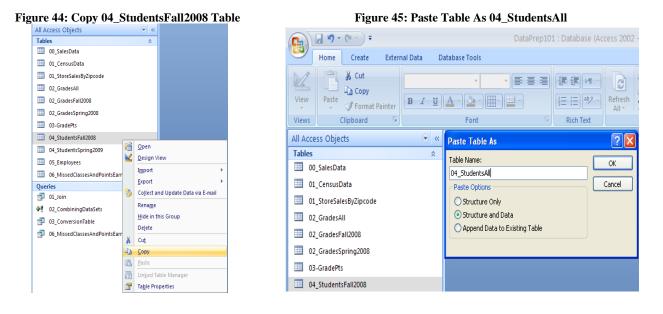


Figure 42: 04_StudentsFall2008 Dataset

04_StudentsFall2008		
StudentID 👻	LastName 🕞	FirstName 👻
111111111	One	John
222222222	Two	Sierra
33333333	Three	Troy
44444444	Four	Montana
55555555	Five	Stefan
666666666	Six	Damian
777777777	Seven	Donzie

F	Figure 43: 04_StudentsSpring2009 Dataset						
	04_StudentsSpri	ing20	009				
	StudentID	-	LastName 🔹	FirstName 🕞			
	333333333		Three	Troy			
	44444444		Four	Montana			
	55555555		Five	Stefan			
	66666666		Six	Damian			
	777777777		Seven	Donzie			
	888888888		Eight	Oriel			
	999999999		Nine	Nancy			

- 2. The first step is to copy the **04_StudentsFall2008** table. On the left-hand side of the screen, under the **Tables** list, locate the **04_StudentsFall2008** table name (see Figure 44). Right click on it to make the shortcut menu appear. Choose **Copy** from the list.
- 3. Click the Paste button (see Figure 45); the Paste Table As dialog box appears. Name the new table: 04_StudentsAll, then click OK. The 04_StudentsAll table now shows up on the Tables list on the left-hand side of the screen. Remember, so far this new table only includes the records from the 04_StudentsFall2008 table. The next step is to find the unmatched records from the 04_StudentsSpring2009 table (LastNames Eight and Nine).



- 4. Click the **Create** button then click the **Query Wizard** button, the **New Query** dialog box appears (see Figure 46). Select **Find Unmatched Query Wizard** from the list; click **OK**. The next step is to choose the table with the new data (in this case **04 StudentsSpring2009**).
- 5. In the first dialog box of the **Find Unmatched Query Wizard**, select the **04_StudentsSpring2009** table from the list, click **Next** (see Figure 47). The next step is to choose the table that contains the related records (in this case, **04_StudentsAll**).



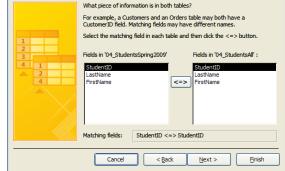
Figure 46: Launc	h the Find Unmatched Query Wizard	Figure 47: Unmatched Query Wizard-Screen 1
	DataPrep101 : Database (Access 2002 - 2003 file format) - Mi	Find Unmatched Query Wizard
Home Create External Data	Database Tools	
Table Table SharePoint Table Form	Split Multiple Blank Form - Form Form Series - Design Form - Report Report Wizard Design - Report Wizard Design - Report Wizard Design - Report - R	
All Access Objects	New Query	2 Which table or query contains records you want in the query results?
Tables A III 00, SalesData III, ConsuBata III, StoreSalesByZipcode O.2, GradesAll III 02, GradesFall2000 III 02, GradesFall2000 III 02, GradesFall2000 III 02, GradesFall2000 III 04, StudentsFall2000 III 04, StudentsFall2000	The query Simple Query Witard Prod Dynamic Costs & Query Ward Prod Dynamic Costs & Que	4 1 3 Table: 02_GradesFinig2008 Table: 03_Gradefs Table: 04_Studentsal Table: 04_Studentsal Table: 05_Employees Table: 05_Employees Table: 05_MissedClassesAndPointsEarned View OTables Queries Bables Queries
04_StudentsSpring2009 05_Employees 06_MissedClassesAndPointsEarned	OK Cancel	Cancel < Back Next > Enioh

- 6. In the second dialog box, select the **04_StudentsAll** table from the list; click **Next** (see Figure 48). The next step is to identify which fields are matching in both tables.
- 7. In the third dialog box, make sure that the **StudentID** field is selected on both sides (indicating that the records in the two tables should be compared on the basis of this key field); click **Next** (see Figure 49). The next step is to identify which fields should show in the query results.

Figure 48: Unmatched Query Wizard-Screen 2







- 8. In the fourth dialog box, move all three fields (**StudentID**, **LastName**, and **FirstName**), from the left to the right, by clicking the >> button; click **Next** (see Figure 50).
- 9. This query must be named. In the next dialog box, type in **04_Spring2009NewStudents**; click **Finish** (see Figure 51). The query results are now showing on the screen (see Figure 52). Notice that only the unmatched records (i.e., the new students for Spring 2009) are in the list.



16

للاستشارات



Find Unmatched Quer	y wizard						
	What would you like to name your query? 04_Spring2009NewStudents						
	That's all the information the wizard needs to create your query. Do you want to view the query results, or modify the query design?						
	 O view the results. O Modify the design. 						
	Cancel < Back Next > Finish						

© 2011 The Clute Institute

ŀ	Figure 52: 04_Spring2009NewStudents Query Results							
	04_Spring2009NewStudents							
	St	udentID	*	LastName	Ŧ	FirstName 🔻		
	8888	88888		Eight		Oriel		
	99999	99999		Nine		Nancy		

- 12. Now that the unmatched records have been identified, the next step is to APPEND (or add) these two records to the **04_StudentsAll** table. Click the **Create** tab, then click on the **Query Design** button on the toolbar (see Figure 15). The **Show Table** dialog box appears, click **Close**. Click the **Append** button on the toolbar (see Figure 29). The **Append Query** dialog box appears. Choose the **04_StudentsAll** table as the file to append (or add records to); click **OK** (see Figure 53).
- Click the Show Table button on the toolbar. The Show Table dialog box appears. Click on the Queries tab and select the 04_Spring2009NewStudents query from the list (see Figure 54). Click Add. Click Close. Your screen should now look similar to Figure 55.

Figure 53	Append	Query	Dialog Box	
-----------	--------	-------	-------------------	--

Append		? 🔀
Append To		ОК
Table Name:	04_StudentsAll	
💿 Current Da		Cancel
O Another Da	atabase:	
File Name:		
	Browse	
	· · · · · · · · · · · · · · · · · · ·	

Figure 54: Show Table Dialog Box

Show Table	×
Tables Queries Both	_
01_Join 03_ConversionTable 104_Spring2009NewStudents	
06_MissedClassesAndPointsEarned_Query	
Add Close	

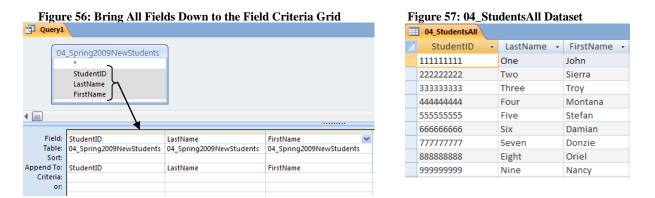
Figure 55: Append Query

Query1	
04	_Spring2009NewStudents
	StudentID LastName FirstName
◀ 📖	
Field:	~
Table:	
Sort: Append To:	
Criteria:	
or:	

- 14. The next step is to add the fields from the **04_Spring2009NewStudents** query box at the top of the screen to the field criteria grid at the bottom (see Figure 56). Double click the title bar of the **04_Spring2009NewStudents** query box. Click and drag the highlighted fields from the query box down to the field criteria grid.
- 15. The next step would normally be to select the matching fields on the **Append To** row. Since the three field names are identical in the **04_Spring2009NewStudents** query and the **04_StudentsAll** table, they already match.
- 16. The final step is to run this APPEND query. Click **Run** (!). A dialog box appears warning that "You are about to append 2 rows." Click **Yes**.
- 17. Congratulations, you have successfully combined the two data sets. Click the **Save** button and save this query as **04_2008And2009Students**. Click **OK**.



18. If you view the **04_StudentsAll** table, it will now contain the list of nine unique students from both semesters (See Figure 57).



- 19. Part D is now complete. The reader may now continue on to part E or close the **DataPrep101.mdb** file.
- *E.* Allowing the Matching of Values in One Column that Reference the Values of Another Column in the Same Table (using a SELF-JOIN query).
 - 1. Open the **DataPrep101.mdb** file. Be sure to turn off the security features (see Figures 3 & 4). The following table will be used for this example: **05_Employees** (see Figure 58). Notice that the **ManagerID** is actually the **EmployeeID** of the manager. The statistical goal is to see if the employee's opinion of **HomeRecyclingImportance** is influenced by (i.e., correlated with) the opinion of their manager. To this requires a dataset that contains the columns: **EmployeeID**, **EmpOpinion**, **ManagerID**, and **MgrOpinion**. Of course, the reader will correctly note that the **05_Employees** table only contains the **HomeReyclingImportance** opinion of the Employee, not the Manager. But, since managers are, themselves employees, their opinions on recycling are also stored in the table. For example, Mary Smith's (Employee #20) Rating is 2 and her manager's (Monica Garcia, Manager #10) rating is 1. The process for extracting data from a table in this way involves making a virtual copy of the original **05_Employees** table and then linking the copy to the original and is called a SELF JOIN.

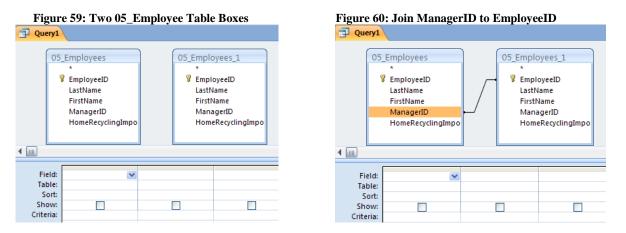
mployees	•				
yeelD 👻	LastName 👻	FirstName 👻	ManageriD 👻	HomeRecyclingImportance	Ŧ
10	Garcia	Monica			1
20	Smith	Mary	10		2
30	Doe	John	10		1
40	Wong	Steven			5
50	North	Carla	40		4
60	Robinson	Kazetta	40		5
	oyeelD 20 30 40 50	Imployees Imployees	DyeelD LastName FirstName Image: Garcia Monica 20 Smith Mary 30 Doe John 40 Wong Steven 50 North Carla	oyeelD LastName FirstName ManagerlD IO Garcia Monica 20 Smith Mary 30 Doe John 40 Wong Steven 50 North Carla	OyeelDLastNameFirstNameManagerlDHomeRecyclingImportance10GarciaMonica1020SmithMary1030DoeJohn1040WongSteven1050NorthCarla40

Figure 58: 05_Employees Table Dataset

To start, click the Create tab, then click on the Query Design button on the toolbar (see Figure 15). The Show Table dialog box appears. Select the 05_Employees table from the list. Click Add, then click Add again. You should now have two 05_Employees table boxes showing (see Figure 59).



3. The next step is to JOIN the two copies of the table on the common field. This JOIN is unique, in that **ManagerID** and **EmployeeID** will be the common fields, therefore allowing us to create a SELF JOIN. Click and drag the **ManagerID** field from the **05_Employees** table box to the **EmployeeID** field of the **05_Employees_1** table box. Your screen should now look like Figure 60.



- 4. The next step is to bring down the required fields to the field criteria grid (see Figure 61). From the **05_Employees** table box double-click the **EmployeeID**, then double-click the **HomeRecyclingImportance**. Next, from the **05_Employees_1** table box double-click the **EmployeeID**, then double-click the **HomeRecyclingImportance**.
- 5. The next step is to rename the two **HomeRecyclingImportance** fields to reflect the identity of the opinion and to rename the second **EmployeeID** field to **ManagerID** (see Figure 62). Type in **EmpOpinion:** in front of the first **HomeRecyclingImportance** field. Type in **ManagerID:** in front of the second **EmployeeID**. Finally, type in **MgrOpinion:** in front of the second **HomeRecyclingImportance** field.

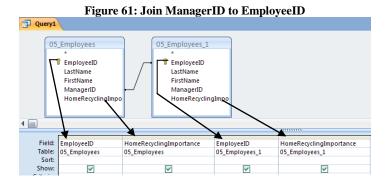


Figure 62: Renaming of the Fields

Query1					
	Employees * EmployeeID LastName FirstName ManagerID HomeRecyclingImpo		05_Employees_1 * EmployeeID LastName FirstName ManagerID HomeRecyclingImpo		
◀ 📖					
	Field: EmployeeID		ion: HomeRecyclingImportanc	e ManagerID: EmployeeID	MgrOpinion: HomeRecyclingImportance
Table:	05_Employees	05_Emplo	yees	05_Employees_1	05_Employees_1
Sort:					
Show:			 Image: A set of the set of the	✓	

김 للاستشارات

6. The final step is to run this query. Click **Run** (!). Congratulations, you have successfully used a SELF JOIN by using two copies of the same table. The results of the query are in Figure 63. It appears that the opinion of the manager does impact the opinion of the employee. Click the **Save** button and save this query as **05_EmployeeAndManagerOpinions.** Click **OK**.

ŧ.	05_Employee/	AndManagerOpinions			
4	Employee 👻	EmpOpinion 👻	ManagerID 👻	MgrOpinion	
	20	2	10		1
	30	1	. 10		1
	50	4	40		5
	60	5	40		5

Figure 63: 05_EmployeeAndManagerOpinions Query Results

7. **Phase II** is now complete. The reader may now continue on to **Phase III** or close the **DataPrep101.mdb** file.

PHASE III: IMPORTING A RECONFIGURED DATASET FROM MS ACCESS INTO SPSS.

Now that the five common ways of reconfiguring datasets using **QBE** have been covered, this third and final phase of the paper explains how to bring a reconfigured dataset from **MS** Access into **SPSS** for statistical analysis. Specifically, the reader will be importing the **06_MissedClassesAndPointsEarned_Query** dataset from **MS** Access into **SPSS** and run a regression on this data. Although **SPSS** will be used in this example, many other statistical analysis packages (including **MS** Excel, SAS, or Minitab) could also be used.

A. The following steps will be used in importing the 06_MissedClassesAndPointsEarned_Query from MS Access into SPSS. Figure 64 shows the query results in MS Access.

1	1155eu Classes Allul	onnisizarneu
1	06_MissedClassesAndPointsEarn	ed_Query
	NumberOfMissedClasses 👻	TotalPointsEarned •
	0	900
	0	850
	0	950
	0	800
	1	750
	1	700
	2	720
	2	690
	2	670
	3	640
	3	600
	4	580
	4	550
	5	450

Figure 64: 06_MissedClassesAndPointsEarned_Query Results

- 1. Launch **SPSS** so that a new data editor window appears (see Figure 65).
- Before the query may be imported, a "Microsoft Data Source" must first be established. This is done by the use of SPSS's Database Wizard. To launch the wizard, from the menu select File/Open Database/New Query (see Figure 66).



Fi	igure 65: New SPSS Data Editor Window									
9 1	Untitled	11 [Data	iSet0]	- SPSS Stat	istics [ata Editor	e la			
Eile			-	-	Analyze		Utilities	Add- <u>o</u> ns	-	
₿		I			? M	1	*			V
1:								1	ible:0 of0 \	
	1		var	var		var	var	Va		var 🔺
										333
	3									
	4	4	3							•
Dat	a View	Variab	le Viev	v						
					SF	SS Statistics	Processor	ris ready		

- 3. The **Database Wizard** appears. Select **MS Access Database** from the list; click **Next** (see Figure 67).
- 4. The **ODBC Driver Login** window appears. The next step is to locate the **DataPrep101.mdb** file. Click the **Browse** button; locate the file; click **OK** (see Figure 68).

Figure 67: Database Wizard	Figure 68: Selecting the Database
Database Wizard	📴 ODBC Driver Login
Welcome to the Database Wizard! This wizard will help you get data from a non-SPSS Statistics data source. You can choose the data source, specify which cases are retrieved, aggregate and sort the data before retrieved, and specify variable names and properties here. Some features are available only when connected to SPSS Statistics Server. From which data source would you like to retrieve data?	Data Source: MS Access Database Database:
WebASE Files MS Access Database Wiso Database Samples Wiso Database Samples Add COBC Data Saure	
< Beck Next > Finish Cancel Help	

- 5. The **Database Wizard-Select Data** window appears (see Figure 69). Notice that all of the tables are showing on the left-hand side of the window, however, none of the queries are visible. Click the **Checkbox** next to **Views** and the queries are also included in the list. (see Figure 70).
- 6. Select the **06_MissedClassesAndPointsEarned_Query** from the list, click the **>** to move the two fields from the query to the right-hand side of the screen. Your screen should now look like Figure 71.



Figure 69: Database Wizard-Select Data-Screen 1	Figure 70: Database Wizard-Select Data-Screen 2
🖬 Database Wizard	🔂 Database Wizard 🛛 🔀
Select Data Select Data Selecting a table selects al of the fields. The Selecting a table selects al of the fields. The Selecting a table selects al of the fields.	Select Data Set the fields you want to refrieve. Then click the arrow button or drag the fields to the Retrieve Fields lat. Te: Selecting a table selects all of its fields. The Dispersion of the dispersion of the dispersion of drag the fields to the Retrieve Fields in this Order:
Sort field names Show: Viables	Sort field names Show: ♥ Jables ♥ Vjews Synonyms System tables
< Back Med > Frish Cancel Hep	< Back Med > Finish Cancel Help

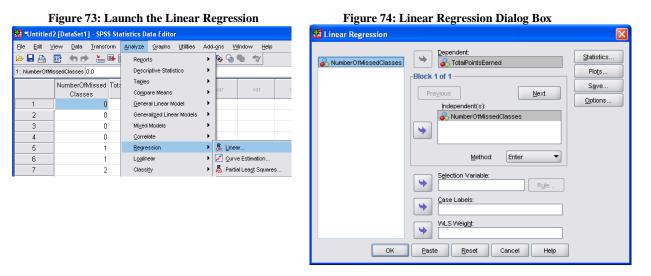
7. Notice that both the NumberOfMissingClasses and the TotalPointsEarned fields are now listed. Click Finish. The data is now showing in the SPSS Data Editor (see Figure 72). Congratulations, you have successfully imported the dataset into SPSS. Follow these same steps to bring in any table or query from an MS Access database.

Figure 71: Database Wiza	Figure 72: Data in SPSS Data Editor				
🚰 Database Wizard		×	🔛 *Untitle	d2 [DataSet1] - SPSS	Statistics Data Editor
Select Data				<u>V</u> iew <u>D</u> ata <u>T</u> ransfor	- · - ·
Select the fields you want to retrieve. Then click the arrow button or drag th	e fields to the Retrieve Fields list.		🗁 📙 🔔	- 📴 🔸 🔶 🔚 🖩	• 📭 🗛 🔸 🛔
Tip: Selecting a table selects all of its fields.			1 : NumberOfM	lissedClasses 0.0	
Available Tables: 문법 00_SalesData 문국] 01_CensusData	Retrieve Fields in this Order:	*5		NumberOfMissed T Classes	otalPointsEarned
🕀 🖓 01_Join	문_ 06_MissedClassesAndPointsEarned_Query: TotalPointsEarned		1	0	900
문 🌚 01_storeSalesByZipcode 문 👻 02_GradesAll		*	2	0	850
02_GradesAll	*	¥	3	0	950
문 육 02_GradesSpring2008 문 육 03-GradePts		•	4	0	800
E 🖶 03_ConversionTable			5	1	750
문 육 이4_Spring2009NewStudents 문 육 이4_StudentsAll			6	1	700
⊕ ⊕ 04_StudentsFall2006			7	2	720
용 육 이4_StudentsSpring2009 용 육 영 05_EmployeeAndManagerOpinions				_	690
E Toployees				2	
G MissedClassesAndPointsEarned			9	2	670
8-9106_MissedClassesAndPointsEarned_Query			10	3	640
			11	3	600
· · · · · · · · · · · · · · · · · · ·		_	12	4	580
🗌 Sort field names Show: 🗹 Iables 🗹 Views 🗌 Syno	nyms System tables		13	4	550
			14	5	450
< Back	Finish Cancel Help				

- B. The next step is to run the regression in SPSS. In this dataset the **TotalPointsEarned** is the dependent variable and the **NumberOfMissingClasses** is the independent variable. Follow these steps to run the regression.
 - 1. From the menu, select **Analyze/Regression/Linear** (see Figure 73). The Linear Regression Dialog Box Appears (see Figure 74).



2. On the right-hand side of the screen, select **TotalPointsEarned** and click the **■** next to the **Dependent:** variable. Next, select the **NumberOfMissingClasses** and click the **■** next to the **Independents(s):** variable. Your screen should look similar to Figure 74.



- 3. The regression is now ready to be run. Click **OK**. The results of the regression are now displayed in the **SPSS Statistics Viewer** (see Figure 75). By looking at the right-hand column of the **Coefficients** table, it is clear that independent variable **NumberOfMissingClasses** is shown to have a significant relationship (i.e, .000) with the dependent variable, **TotalPointsEarned**. This result could be interpreted as meaning that the more classes a student misses, the lower their total class points earned will be. Obviously this is a contrived dataset, however the two authors are also professors and we would like to believe that this is true!
- 4. Now that the regression is complete, it is time to store the results. Save the SPSS Statistical Viewer file as **06_MissedClassesAndPointsEarned.spo**; save the data file as **06_MissedClassesAndPointsEarned.sav**. Close **SPSS**.

		Model Sur	nmar	y					
Model	R	R Square		sted R luare	Std. Error of the Estimate				
1	.950ª	.902	00	.894	45.226				
a. Pre	edictors: (Co	nstant), Numb	erOfM	issedClass	es				
				ANOVA ^b					
Model		Sum of Squares		df	Mean Square	F	Sig.		
1	Regression	361	1	227176.361	111.066	.000	а		
	Residual	24545.0	068	12	2045.422				
	Total	251721.4	129	13					
	,	nstant), Numb iable: TotalPoi			es				
				Coef	ficientsª				
			Ur	nstandardiz	ed Coefficients	Standardize Coefficient			
Model				В	Std. Error	Beta		t	Siq.
1	(Constant)			854.836	18.765		4	5.556	.000
	NumberOfM Classes	issed		-78.433	7.442	96	50 -1	0.539	.000



CONCLUSION

The purpose of this paper was to help researchers deal with ill-behaved data through the use of relational database software, with particular focus on QBE (Query-By-Example). We believe we have accomplished this goal through a demonstration consisting of the following steps:

- How to import a misconfigured dataset from MS Excel into an MS Access table.
- Five examples of how to use **MS Access'** QBE to reconfigure datasets.
- How to import a reconfigured dataset from MS Access into SPSS.

The broader purpose of this paper is to demystify database software and encourage the reader to learn more about the usefulness of database tools and techniques in their own field of expertise. To this end, we have included in our reference list a number of database theory and MS Access texts that could be used for further explanations of database topics. We hope that they will prove helpful.

AUTHOR INFORMATION

Dr. Paul J. Lazarony is an Associate Professor of Information Systems at California State University, Northridge. His current research interests are assessment, database, enterprise-wide systems, and IT project management. He received his PhD in Computer Technology and Business Education from the Ohio State University.

Dr. Donna A. Driscoll is a Professor of Information Systems at California State University, Northridge. Her current research interests are database, information assurance, and web-based systems development. She received her PhD in Business Administration (Accounting and Research Statistics) from the University of Southern California.

REFERENCES

- 1. Groh, M. et al. (2007). Access 2007 Bible, Wiley Publishing, Indianapolis, IN.
- 2. Hoffer, A. and Prescott, M. (2009). *Modern Database Management*, Ninth Edition, Pearson Prentice Hall, Upper Saddle River, NJ.
- 3. Kroenke, D. and Auer, D. (2010). *Database Concepts*, Fourth Edition, Prentice Hall, Boston, MA.
- 4. Kroenke, D. and Auer, D. (2010). *Database Processing: Fundamentals, Design, and Implementation*, Eleventh Edition, Prentice Hall, Boston, MA.
- 5. Post, G. (2005). Database Management Systems, Third Edition, McGraw-Hill, Boston, MA.
- 6. Rob, P. and Coronel, C. (2007). Database Systems: Design, Implementation and Management, Seventh Edition, Thomson Course Technology, Boston, MA.
- 7. Shelly, G. and Cashman, T. (2007). *Microsoft Office Access 2007: Comprehensive Concepts and Techniques*, Thomson Course Technology, Boston, MA.

