Extraction, Transformation, and Loading in a Data Warehouse Course

Edward A. Boyno, Ph.D.[1]
Dept. of Computer Science, Montclair State University
Upper Montclair, NJ 07043 USA

Abstract

This paper describes my experiences teaching a unit on Extraction, Transformation, and Loading as part of a course on Data Warehousing. After a brief discussion of the process in general, it concentrates on one of the more interesting parts of that process, Data Cleansing. Some of the problems that may occur during Data Cleansing are discussed as well as some of the techniques that might be used to address them. Student exercises, some elementary, some more challenging, are presented and discussed. Finally, the paper addresses some of the problems I encountered and possible future work.

Keywords: extraction, transformation, loading, ETL, data warehouse, data cleansing, data scrubbing, CS education, IS education

1. INTRODUCTION

A central concern in any data warehousing initiative is the population of the warehouse with data from multiple, heterogeneous data sources. This process is commonly referred to as "extraction, transformation and loading of data" or "ETL." Clearly, these ETL concerns must also be of central interest to the teaching of a course in data warehousing. This paper will briefly discuss ETL in general in Part 2 and then, in Part 3, will concentrate on some of the specific problems that must be addressed during one particular phase, the "scrubbing" or "data cleansing" phase. Data Cleansing is generally very interesting to students because the problems associated with it are easily (and sometimes humorously) presented.

After discussing some of these problems, the paper will look at some of the possible tools and techniques that might be used to address them. Part 4 will present some exercises that proved useful in the course; part 5, will address some of my experiences while teaching the subject and part 6 will present some future plans.

2. ETL IN A NUTSHELL

I prefer the term "data migration" to describe the process by which a data warehouse is initially populated and subsequently updated. I view it as several separate but interdependent steps:

Extraction: The extraction of data from a collection of usually heterogeneous sources. Sources of data might include ASCII files; legacy databases; mainframe data perhaps in VSAM files or other proprietary systems; files formatted for commercial DBMS', both relational and otherwise, and possibly even the Internet

Conditioning:(sometimes called data transformation): The conversion of data from the source data type to the target data type. Examples might include changing EBCDIC to ASCII; packed decimal to float etc. We may also include in this step some other transformations like changing "M" to "Male."

[1] boynoe@mail.montclair.edu

Conditioning is pretty much mechanical and easy to automate.

Scrubbing: (sometimes called data cleansing): Making sure that the data meets all the validation rules that have been decided upon by the warehouse designers. Problems that can arise during this step include null or missing data; violations of data type (as, for example, the placement of numeric data in non-numeric fields or visa-versa); non-uniform date formats; invalid data (a supplied zip code must actually exist and must be correct for the given city and state), and many others. Data Scrubbing is the major topic of this paper.

Transformation: putting the data into a form that is more appropriate for data mining operations. Some of the actions that may occur during this phase are:

Smoothing: the act of using statistical techniques to remove irrelevant data points from the dataset.

Aggregation: the act of calculating summary data.

Generalization: the act of replacing finely grained data with "higher level" data usually along some well defined hierarchy. For example, warehouse designers may decide that actual ages are unnecessary and would replace them with categories like "young," "middle-aged" and "elderly."

Normalization: the act of transforming data in specific and clearly defined ways. For instance, numerical data may be transformed so that all data lies in the range of 0 to 1.
Data may be normalized across both its range and in its distribution.

Attribute Construction: the act of adding new attributes to aid in data mining.

Loading and Refreshing: The actual placement of the data in the warehouse. "Loading" here refers to the initial build of the warehouse and refreshing to the process of updating the warehouse.

Other operations that may be performed during data migration include:

Validating: Making sure that the data has maintained its integrity during the transformation process.
Auditing: Attempting to uncover unusual facts.

Merging: Choosing a schema if a target datum is found in multiple sources.

3. DATA SCRUBBING

There are problems to be overcome in every phase of the process, but I will limit my discussion to those that arise during data scrubbing. As noted, these anomalies are easily understood and generally interesting to students. For discussion purposes, I divide the problems into two broad categories: those dealing with primary keys and those that do not.

Primary key problems:

1. Records may have the same primary key but might have different data. This can occur if primary keys are reused or when two organizations or units merge.
2. The same entity might occur with different primary keys. This can easily arise when different segments of an entity design databases independently of one another
3. Data may have a primary key in one system but not in another. The classic example of this kind of error occurs when an entity is represented in more than one source database. It is quite possible that the entity is central to one of the databases and therefore has a primary key field or fields while the same entity is so peripheral to the purposes of the other database that it is not dignified by being given a primary key.
4. Primary Keys may be intended to be the same but might occur in different formats. Probably, the most widespread instance of this error occurs when social security numbers are used as primary keys: are they character data or numeric; if character data, do they contain dashes?

Non primary key problems:

1. Data may be encoded differently in different sources. The domain of a "sex" field in some database

may be {'F', 'M'} or as {Female", "Male"} or even as {1,0}.

2. There are often multiple ways to represent the same piece of information. "UVA", "University of Virginia", "Univ. of Virginia" and "Virginia, Univ. of " can all be found in the literature as representing that eminent institution founded by Thomas Jefferson.
3. Sources might contain invalid data. A point of sale terminal may require that the sales clerk enter a customer's telephone number. If the customer does not wish to give it, clerks may enter 999-999-9999.
4. Two fields may contain different data but have the same name. There are a couple of ways in which this can happen. "Total Sales" probably means fiscal year sales to one part of an enterprise and calendar year sales to another. The second instance can be much more dangerous. If an application is used by multiple divisions, it is likely that a field that is necessary for one business unit is irrelevant to another and may be left blank by the second unit or, worse, used for otherwise undocumented purposes.
5. Required fields may be left blank. Clerical errors account for most of these, but Zip Codes did not come into use until 1963, so addresses recorded before then will not have them.
6. Data may be erroneous or inconsistent. The Zip Code may be for CA but the State is listed as NY.
7. Data might violate business rules. The listed minimum rate of a variable rate loan might actually be higher than the listed maximum rate
8. Data might be stored in one field that ought to be in multiple fields. In the US, shoe sizes should probably be 2 fields expressing length and width, but is usually recorded as one.
9. The data may contain null values. Null values can occur for a wide variety of reasons, the most common of these are:
   a. Data that is genuinely missing or unknown,
   b. An attribute does not apply to an entity,
   c. Data that is pending, or
   d. Data that is only partially known.

Handling null data is further complicated by the variety of ways in which such data can be represented:

System nulls,
Default values, or
User defined nulls.

4. A FEW SOLUTIONS

Once one has decided to cleanse data, a decision that is not in many instances automatic and a subject worthy of extensive discussion all by itself, there is a kind of generic set of tools and techniques that might be used (usually in combination) to solve the problems presented by the need to cleanse data.

The question of how to cleanse the data actually encompasses two possibly separate problems: the initial load and refreshment. There are a variety of tools and techniques which may be used in one or the other or both of these operations.

Techniques:

Using Reference data: A reference system is a central repository of data standards. These standards are distributed to each "feeder" data source which then imposes the standards on itself. All the data which is subsequently moved to the warehouse will be in the correct format.

The essence of this technique is the ability to define and maintain the reference system. Given this ability, this system has several benefits: much of the warehouse processing, even the routine transformations to the physical data, can be transferred to the feeder databases; cross feeder processing is not a problem and other elaborate data reconciliation processes are not needed.

This system cannot, of course, be used with data that is not under the control of the enterprise as is the case with legacy systems and the Internet

Domain Mapping: Domain mapping requires the warehouse designers to construct a uniform set of allowable domains and a uniform set of functions that transform the values in the feeder sources into these values. This technique can clearly be applied to all sources, but requires a close investigation of both the technical and business aspects of all the feeder sources. The need for this analysis can grow rapidly as the number of feeder sources increases and can add greatly to the cost of maintaining the warehouse.

Tools:

Using Domain Experts: The tool of last resort but sometimes a necessary one. There are simply some decisions that cannot be made automatically. The goal of data cleansing is to reduce the use of human experts to an absolute minimum.

Parsing and Fuzzy Matching: In many instances it is possible to apply standard parsing techniques to identify syntactic elements of a record allowing warehouse designers to map source data to a standard or to decide when, in fact, data matches. Similarly fuzzy techniques can be used to decide that two data elements are probably the same.

Designating a Preferred Source: We can use this technique to implement the "reference data" technique simply by designating one source to be the reference database.

Rule Based Data Cleansing: Every data cleansing rule has two parts: one that tests for the presence of an error and one that specifies which action is to be taken. Rules can be used to simply count errors without attempting to repair them (auditing), to remove records which violate the integrity rules (filtering) and of course to repair erroneous data (correcting). The trick, of course, is finding and implementing the rules.
Rules can be implemented in code or by using the rule based systems that come with all major commercial DBMS'. An excellent discussion of rules can be found in Duncan and Wells (Duncan 1999).

Handling Null Data: These same techniques and tools can be used to deal with missing data. A special technique put forward by David McGoveran (1993, 1994a, 1994b, 1994c) might also be useful. Rather than insert null data in a cube, a special null table or tables is (are) created. Every occurrence of a null value causes a row to be placed in a null table with a surrogate key and an optional explanation of the type of null. This key value is placed in the base table as a foreign key.

5. EXERCISES

Students worked with a warehouse that nominally dealt with library records. The design of the warehouse, a classic star schema, was given. The basics of warehouse design were presented in another part of the course.
The exercises that were developed, tested and given to students are shown below. They were based on three datasets. The first two were a pair of 5000 or so row data sets that I had constructed with certain known errors. The first dataset (theoretically about library patrons) did not contain a primary key field forcing students to provide a surrogate key field. The second did have a primary key field that was not the same as students had invented for the first file. The data was constructed so it would not be possible to completely resolve the inconsistent data without the use of a human expert, so among other things, students had to develop policies and strategies for what records would in fact be referred to an expert. The third dataset was taken from the University of California at Irvine KDD archive. It was a large, very messy dataset dealing with movies presented by Gio Wiederholt for the annual KDD competition.. It can be found at www.kdd.ics.uci.edu.

Elementary Exercises:

1. Convert data from an unstructured ASCII file to an Oracle database using the Oracle Loading utility, SQLLdr.

2. Use a high level language to develop and implement techniques to cleanse the data of the following types of errors:
   i. Mixed representation of data
   ii. Duplicate data
   iii. Missing primary keys
   iv. Primary keys from multiple sources
   v. Ambiguous data

Students were given a copy of the two files and were expected to cleanse them of the errors noted above and load them into an Oracle database that I maintain for student use.

None of the exercises required very much high level theoretical work. Mixed representation data is straight forward to deal with; primary key problems were dealt with by creating a new surrogate key for the warehouse files and assigning it to records as they were loaded and exact duplicates were found by simply sorting.

The ambiguous data was found only in the addresses of the supposed library patrons, represented in the datasets as city and state only. Some cities appeared in the data in multiple alternate forms as for example N Orleans, N. Orleans and New Orleans. In order to regularize the data students processed the data as follows:

Students constructed a binary search tree whose nodes contained the name of a city (city name plus state designation) and a counter. Further, each node acted as the head of a linked list which would hold synonymous city names if any occurred.

As a data record was processed, the city name was scanned for tokens. Spaces and punctuation marks were used as delimiters. Students decided on their own that in any city name that had more than one token in it, the second token was probably the most important and should be designated as the principal token. If a city just consisted of one word, that word was used as its principal token. Cities with the same name but in different states were considered to have different principal tokens. As a city name was encountered for the first time it was entered into the binary search tree according to its principal token.

If a city name could be identified as already being present in the tree, the counter was incremented. This is simple for single word names, but more interesting for names with multiple tokens.

If a multiple-token name was seen, its principal token was compared to the principal tokens of previously catalogued names and if a match was found, the secondary tokens were compared. If they matched, the name counter was again incremented, if not, as in the case of N Orleans versus New Orleans, the user (playing the role of Domain Expert) was asked to decide if an actual match had been found. If yes, the variant form was added to the linked list and the counter incremented; if no, a new node was created in the tree.

More Advanced Exercises:

These exercises included many of the same issues as above plus the following:

1. Parsing semi-structured data to determine its field structure

2. Dealing with null data.

3. Loading large amounts of data

Except for the need to do some more extensive parsing and developing a strategy for dealing with missing data, there really wasn't a lot of theoretical knowledge required here either. With a little more time to spend on these issues or more experienced students, I think that all students could have performed the exercises.

6. EXPERIENCES AND OBSERVATIONS

The ETL material was first presented as part of a graduate course on data warehousing. Some of the students had experience with DMBS' some not. Not all of the students experienced with DBMS' had completed formal course work. All of the students were competent programmers, but, unfortunately only a few had experience with parsing or with fuzzy techniques. Familiarity with DBMS' did not seem to effect students performance on the ETL material; the elementary parsing techniques that were required to perform the elementary exercises was easily and quickly taught. The mathematical background of the students varied widely as well, and this proved to be a much greater handicap. Exploring many of the basic data transformation operations would have required a lot of background work on my part. I probably should have anticipated this difficulty, and it is an issue that would have to be addressed in any future attempts to teach the material.

Another serious difficulty that arose, from my point of view, was the sheer lack of time available to handle all the issues that were raised. There is a limit on how many programming projects you can expect of students even in a graduate course.

The third difficulty worth mentioning was the development of the datasets. I really wanted to control the types of errors that students would encounter in the elementary exercises, so I constructed the datasets myself by editing and anonymizing some of my old student data. This was a lot of work but produced datasets that have easily stated and controlled errors. They are however really quite small and don't present much of challenge to the students in the sense that inefficient processing really doesn't punish them very much. The UCI archives are a rich source of datasets, but they are really intended for a different purpose and obviously can't be as carefully controlled.

On the bright side, students found the material quite stimulating and were eager to present data cleansing horror stories from their own experience. All the students were able to complete the basic exercises in one way or another. Some of their work was quite good. Four students attempted the advanced exercise and produced a rather nice data cube from the Wiederhold data.

7. FUTURE PLANS

The most easily corrected problem that I had with the ETL material was the lack of time. There simply wasn't time for students to accomplish any more than some basic data cleansing exercises. Next time I would offer it in no less than half a semester perhaps in conjunction with some more extensive discussion of data transformation techniques and possibly some data mining. With an entire semester to spend, I think there would also be enough time to address some of the student's mathematical shortcomings as well.

On a more technical level, there are several things that need to be done. Clearly it would desirable to increase the size of my "controlled" data sets and to increase the number of kinds of errors that were available for the students to practice on. Specifically my data sets do not yet contain any numerical attributes thereby constraining the types of errors I can introduce. In particular I cannot introduce numerical null values, a rich source of problems in both data cleansing and data transformation.

Other types of errors whose introduction would seem to be straightforward and which would lead to a valuable learning experience are: instances of invalid data; fields used for different purposes in different data sets and inconsistent data.

8. REFERENCES

Bohn, K., 1997, "Converting Data for Warehouses." DBMS, June, p. 61.

Boyno, E., 2001, "A Syllabus in Data Warehousing." Proceedings of the Information Systems Education Conference, Cincinnati, November.

Cataldo, J. A., 1997, "Care and Feeding of the Data Warehouse." Database Programming and Design, December, p. 36.

Corey, M., M. Abbey, I. Abramson, and B. Taub, 1998, Oracle 8 Data_Warehousing. Osborne/McGraw-Hill, Berkeley, CA.

Duncan, K. and D. Wells, 1999, "A Rule Based Data Cleansing." Journal of Data Warehousing, vol. 4 no.3, Fall, p.2.

Elmasri R. and S. Navathe, 2000, Fundamentals of Database Systems, 3rd Ed. Benjamin/Cummings, Redwood City, CA.

Gardner, S., 1998, "Building the Data Warehouse." Communications of the ACM, vol. 41 no. 9, Sept, p. 52.

Gray, P. A., 2001, "Teaching the Data Warehousing Course." ISECON, November.

Hudicka, J., 1999, "So, You Say Your Data's Clean, Huh?" Select, July, p. 16.

Inmon, W. H., 1996, Building the Data Warehouse, 2nd Ed. John Wiley, New York.

Johnson, T., and T. Dasu, 2003, "Data Quality and Data Cleaning: An Overview." ACM SIGMOD/SIGPODS, San Diego.

Kimball, R., 1996, Data Warehouse Toolkit. John Wiley, New York.

Kimball, R., 2000, "Indicators of Quality." Intelligent Enterprise, April, p. 20.

Kimball, R., 2000, "Is Your Data Correct?" Intelligent Enterprise, December, p. 22.

McCartney, B., 2000, "Data Warehouse Loading Performance Considerations." Oracle Openworld, San Francisco, November.

McGoveran, D., 1993, "Nothing From Nothing." Database Programming and Design, December, p. 33.

McGoveran, D., 1994, "Classical Logic: Nothing Compares 2 u." Database Programming and Design, January, p. 54.

McGoveran, D., 1994, "You Can't Lose What You Never Had." Database Programming and Design, February, p. 43.

McGoveran, D., 1994, "It's In The Way That You Use It." Database Programming and Design, March, p. 54.

Moss, L.. 1998, "Data Cleansing: A Dichotomy of Data Warehousing?" DM Review, February. <http://www.dmreview.com/editorial/dmreview>

Pyle, D., 1999, Data Preparation for Data Mining. Morgan-Kaufman, San Francisco.

Ramakrisnan, R. and J. Gehrke, 2000, Database Management Systems, 2nd Ed. McGraw-Hill, Boston.

Redman, Thomas. A., 1998, "The Impact of Poor Data Quality on the Typical Enterprise." Communications of the ACM, vol. 41, no.2, February, p.79.

Sullivan, D., 2001, Document Warehousing and Text Mining. John Wiley, New York.

Thomsen, E., 1997, OLAP Solutions. John Wiley, New York.