Building ETL Processes for Business Intelligence Solutions Built on Microsoft SQL Server

Extract, transform, and load your operational databases into a powerful data warehouse for integrated analysis.

By Eric Johnson and Joshua Jones

CA ERwin

TABLE OF CONTENTS

INTRODUCTION

If you work with a business intelligence solution, then chances are pretty good that you need to pull data into a data warehouse or data mart. On the surface, this seems like it would be an easy task. All you have to do is gather data from various systems and load it into the data warehouse. Maybe you have to map a few columns because things are stored in different places, but all data is simple to interpret because it's all stored in a nice relational database. Wouldn't it be nice if that was the case? In the real world, data is stored in many different places in your organization. Some of it may be in relational databases, but some may be in flat files with no referential integrity rules. Pulling data from these sources and getting it into the common format of your data warehouse can be a monstrous task, one that takes you weeks to develop.

When it comes to designing a data warehouse, there are quite a few traditional data modeling processes that are useful. When you design a data model, you will typically gather requirements, identify entities and attributes based on the data, normalize the model, determine relationships, and document the model using a modeling tool such as CA ERwin DM. But now you have to work at getting data into the physical database described by the model.

In general terms, this process of gathering data from several disparate data sources, changing it to a common format, and loading it into a data warehouse, has come to be called Extract, Transform, and Load (ETL). There are many tools available that help you to build ETL processes, but you still need to have a firm understanding of the data in order to be successful.

Like any design program, ETL products give you the tools you need to design an ETL process but you must understand the tools on order to make them work for you. If your business intelligence solution is using SQL Server 2005 or higher, you are likely using SQL Server Integration Services (SSIS) to build your ETL packages. What follows is an in-depth look at the components of an ETL process. Examples will be given using SSIS, but any ETL tools should have similar constructs to the one shown.

EXTRACTING DATA

The first step in the ETL process is the extraction of data from all the sources that contain information you need to have housed in your data warehouse. While this seems like the easy step of the whole process, it can often be jam-packed with difficulties. First, consider all the different places that data can live in your organization. Here is a short list of the different locations from which you may need to extract data.

- SQL Server Databases
- Oracle Databases
- Microsoft Access
- Microsoft Excel
- Delimited Files
- Fixed Length Files
- Lotus Notes Databases
- Lotus 1-2-3
- MySQL
- Sybase

This list is by no means exhaustive, but gives you an idea of the types of places you are likely to find data. Some of these sources are relational, while others are just single flat files that probably don't have any data integrity rules. Remember, just because the data is coming from a relational database management system, it is not necessarily being stored in a normalized relational manner. Databases are only as good as their design, and there are plenty that look like a huge collection of Excel spreadsheets with no discernable relationships or data integrity.

The goal of the extract step is to retrieve the data as it exists in the source systems so your ETL process can work with it. To that end, you need to understand how the data looks and make every effort to read the data into something resembling a record set. If you are pulling from a flat file, then you need to understand how the data is stored and delimited so you can turn it into columns and rows.

Using SSIS, this is done by setting up Data Connections and creating Data Sources in an SSIS package as shown in Figure 1. In this example, the Connection Manager, shown on the button of the screen, is what allows you to establish connections to the data sources. The two sources, shown in the package itself, allow you to tell the package what format the data takes on so it can be read and used in the ETL process.

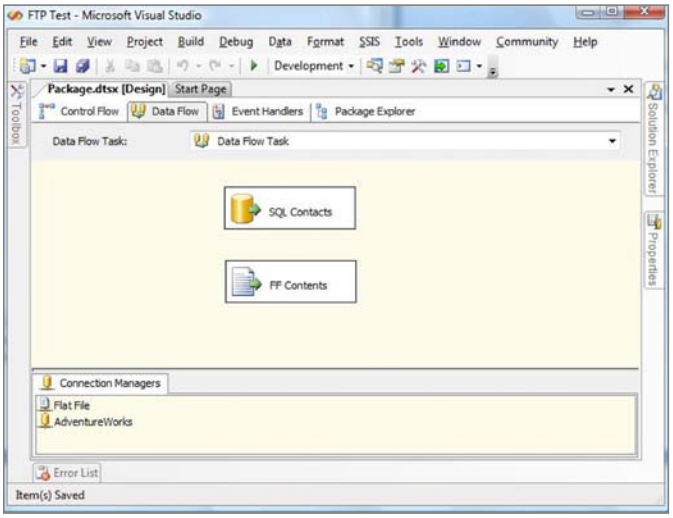


Figure 1: *An SSIS package that connects to SQL Server and Flat File data sources.*

If you're lucky, most of your data sources will fit a pre-defined type, such as the ones discussed here, but luck can be a fickle friend. You may end up with data that is stored in files with no discernable way of extraction. In these cases, you may need to brush up on your programming skills in order to write custom extraction processes. This can often be time consuming, but is sometimes necessary. Just remember, whether it's a straight forward or a custom extraction process, the goal is to retrieve the data so that it is useable by the upcoming transformation phase of the ETL process.

BUILDING DATA TRANSFORMATIONS

The next and most complex portion of the ETL process is the transformation phase. At this point, you have extracted all your data from all the various data sources in your environment, but chances are good that the data looks very different than the destination schema of your data warehouse. This is where the rubber meets the road in terms of ETL. You have to build processes to translate the data in its current form to the form of the warehouse. This goes far beyond just mapping columns and tables to the correct location in the warehouse. Often, you will need to change the data itself to conform to the data types and other constraints of the data warehouse. In addition, you may have to modify data altogether in order for it to be useful.

Finally, you may even have requirements to roll-up data coming in to the data warehouse because you only care about weekly, monthly, or quarterly totals. In the sections that follow, we are going to take a look at the most common tasks that you may run into when designing your data transformations.

FILTERING DATA

One of the first things you will usually do in the transformation phase is to filter the incoming data to make sure that you only get the data you want. For example, you may do this to reduce your result set to the last month of data or to ignore rows where a particular column is null. While this can often be done by filtering your data in the extract portion of the process, it is often better to do it during transformation so you can easily find the places where you filtered data.

DATA MODIFICATION

Sometimes the data that comes in from your source needs to be changed in order to be useable in the data warehouse. This is best explained in an example. Take something simple like storing the gender of a contact; there are probably 50 ways to accomplish the storage of a value that would indicate gender. Take a look at Table 1 for some examples of how this data could be stored.

Column Name	Possible Values
Gender	M or F Male or Female 1 or 2
Sex	M or F Male or Female 1 or 2
Male	Yes or No True or False 1 or 0
Female	Yes or No True or False 1 or 0

Table 1: *Different Ways to Store*

As you can see, there are a few different ways to store the same data. This data need to be managed and mapped to your data warehouse. Let's say you have a column called gender and you store a 1 for a male and a 2 for a female. You need to create transformations to map the value of "Yes" in a column called Male to a 1 and a "No" to a 2. In addition, you may have several sources doing things in the different ways so you may have to build multiple mappings, one for each source. In SSIS, this can be done using "IF" logic in a Derived Column transformation as shown in Figure 2.

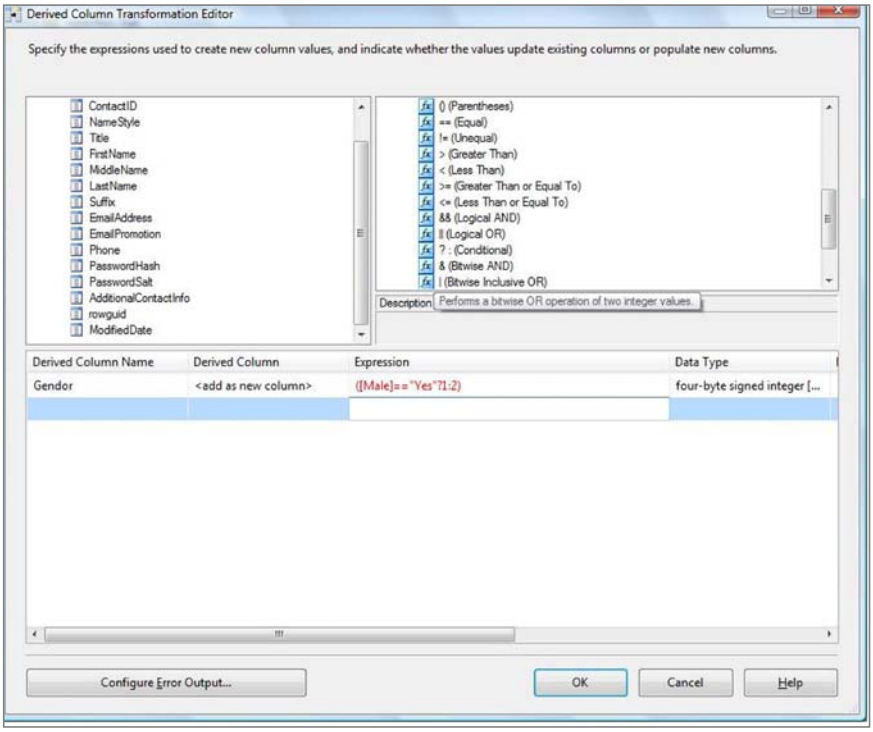


Figure 2: *A SSIS Derived Column Transformation.*

In addition to simple columns with two possible values, you may have columns with hundreds of possible values. In these cases you may need to look up a key value based on the string value that is being passed in. This is usually accomplished with a lookup transformation which passes a string in as a select filter and returns the appropriate key value to use for inserting into the destination. This makes life easier when you have more complex columns.

CALCULATING NEW VALUES

Often with data warehouses you are not concerned with storing all the details of everything that is contained in your source table. You may track orders and sales in a SQL Server database, but when you import that data, you only care about the total of each line and not the cost or quantity of the product sold. In this case, you may take the qty column and the unit_prc column and multiply them to come to a line total. The total value is written to the data warehouse and the price and quantity columns are ignored. This is a common practice in ETL and you may find yourself doing it quite a bit. Again, the best way to do this in SSIS is by using a derived column transformation.

GENERATING SURROGATE KEYS

Another common practice in ETL is the generation of surrogate keys. This is especially important when you have different primary key values in your various source systems. Usually the original primary key is stored and the new surrogate key is generated by the destination system. The two most common surrogate keys in SQL Server are Global Unique Identifiers (GUIDs) or auto incrementing numeric columns called Identities. In either case, you must generate these values during the transformation process so that the data can be stored in the data warehouse without violating data integrity rules.

SUMMARIZATION AND ROLLUP

Once you have cleaned up your data and have it in a useable state, you will probably spend some quality time summarizing data and rolling it up for storage in the data warehouse. As mentioned earlier, it is often the case that the data warehouse does not contain all the gritty details of the source data. Data is usually summarized and rolled up on a weekly, monthly, or even quarterly basis. This allows the data warehouse to focus on the macro picture and answer questions about trending or sales results.

Let's face it, you probably care less about how many monitors a customer bought and more about how many monitors were sold to all your customers in Q1 or 2008. The former you can probably get from the original database, while the latter forms intelligence that drives your business. The most common tool for this in SSIS is the Aggregate transformation. Using Aggregate, you can sum or average data, find the min or max values, count rows, or even group by specific columns.

ETL WORKFLOW CONSIDERATIONS

In any good ETL tool, you will have a great deal of control over the workflow of your process. How is performance affected when trying to process ten tables at a time versus doing them one by one? There is no correct answer, but you should be able to handle how your package does work and test it to find the optimal solution. This applies not only to transformation but to all aspects of the ETL process. SSIS uses the control flow to allow you to manage how and when tasks in your package are executed.

LOADING DATA INTO A DATA WAREHOUSE

With the heavy lifting out of the way, your data should be ready to load into the data warehouse. This isn't as simple as just inserting mass amounts of data. You really need to take several things in to account before you open the flood gates. First of all, how often do you want to load data, once week, once a month? Loading will have an impact on the data warehouse. Not only will the server seem slower when data is being loaded, data can change and locking can occur.

Speaking of changing data, you also need to decide how to handle this. Some data warehouses only accept new data, i.e. things that have been added since the last load. In others, you must update existing data with the more recent data of your data sources. You may also want to maintain an audit trail as to what has changed and what is new. While the process isn't very complicated as compared to the other parts of ETL, it is a crucial step and you need to make sure you don't cripple your data warehouse in the name of loading data.

OTHER PROCESSES RELATED TO ETL

TRANSIENT DATA SOURCES

When working with complex ETL processes, you may find it useful to save your work along the way. It is common practice to write your data to a transient data source after major steps of the process. These data sources can be flat files or even other databases. For example, you may want to store the data as it exists right after you extract it from the source. You many again want to save your data after it is cleaned up and ready to be summarized.

Saving your work also allows you to break up a complex ETL process to lessen its load on the systems. Maybe you have extracted the data and transformed it, which can happen at any time, but you only want to load on Sunday at 3:00am. Using a transient data source, you can save off the processed data all week while still only affecting the production data warehouse once a week.

MOVING REMOTE DATA

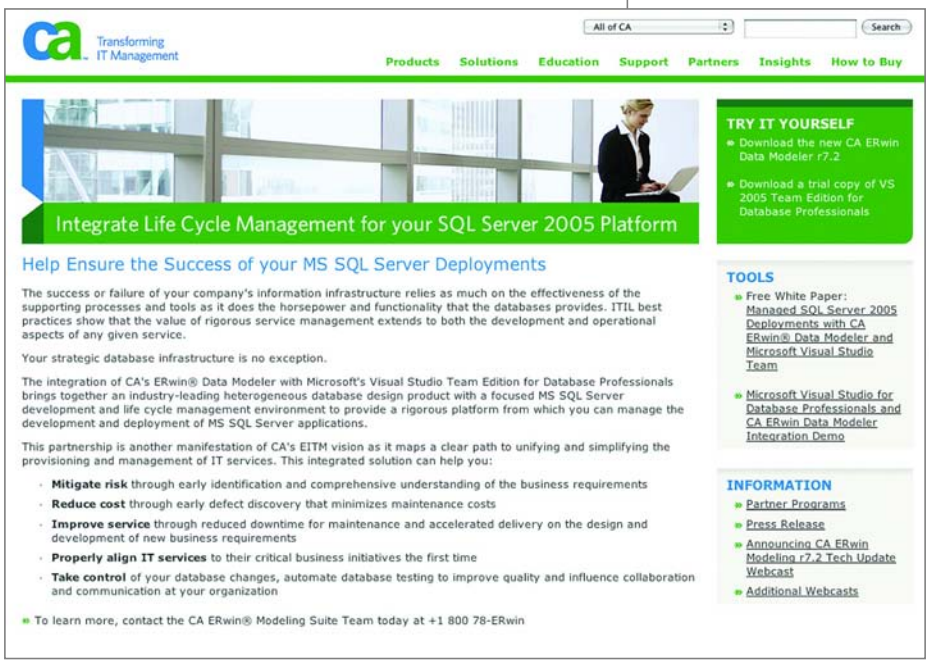
Another common hurdle you will encounter with ETL is the fact that some of your data may be remote to the data warehouse. This means you have to build a mechanism to not only extract data, but also send it to a central location to be processed further. This is often a two step process that begins be exporting your data to a transient data source such as a comma-separated file and then sending that file to a location where it can be picked up and the data can continuing into the transformation phase. SSIS offers you a few ways to do just that. You can use the built in tools to copy files to file shares or to FTP files to a server over the Internet. When all else fails, you can write a custom script to handle the relocation of your files.

CONCLUSION

You can create the greatest data warehouse in the world, but if you can't put data into it, it is completely useless. Designing and building solid ETL processes is critical to the success of your data warehouse. If you take your time and design a flexible ETL process you will find that your life becomes much easier later when you acquire a new data source. While not as flashy, ETL is just as important as the other aspects of your Business Intelligence solution, if you want a good BI solution, then you need to take the time to develop a good ETL process.

ABOUT CONSORTIO SERVICES

Josh Jones and Eric Johnson are the Operating Systems and Database Systems Consultants, respectively, for Consortio Services, as well as co-hosts for the popular technology podcast "CS Techcast" (www.cstechcast.com). They offer consulting and training services in many areas of Information Technology. Check our www.consortioservices.com for more information. Eric and Josh have written a combined 3 books including the forth coming "A Developers Guide to Data Modeling for SQL Server".



Visit: ca.com/erwin/msdn. *CA ERwin Data Modeler provides a way to create a logical structure of your data and map that structure to a physical data store.*

ca.com/erwin/msdn