Chapter 10. Implementation: Building the Database

The IT work involved in building the database is one of the most difficult and complex parts of implementing a data warehouse environment. There are many resources to provide guidance for the project team and technical staff to help accomplish this goal. For the purposes of this text, the focus of database building will be at a high level. Rather than share the nitty-gritty design details, a summary is all that you need. However, it is important for all managers (business and IT) to have a basic understanding of what is happening during this part of the project.

This chapter provides a representative sample of the work that needs to be done, but is not intended to be an all-inclusive reference. It is also important to know how you can help and what is expected of you during this very technical part of the project. This chapter provides both a foundation and insight into how managers and other members of the business community should be involved.

10.1. Extract, Transform, and Load (ETL) Fundamentals

The ETL (extract, transform, and load) work is the most frequently underestimated work that is done on a data warehouse project. It is also the least understood set of tasks by everyone other than the developers themselves. It would seem simple: Copy the data, move it around a little bit, and load it for use. In reality, the ETL process is rarely straightforward and simple. Having a good understanding of the work that must be done and the condition of the existing data will help the project team to better estimate the effort and length of time needed to build the data warehouse.

10.1.1. What Work Is Being Done?

If the work is not that simple, what is actually happening? There are several major steps to developing an ETL system. While it is easy to move data around, the goal here is to create a complete system that can be turned over to the operations group to run, and that is not as simple to accomplish. The same characteristics of any other production system must apply to the data warehouse too. The goal is to build a full system that runs without manual intervention.

CHARACTERISTICS OF A PRODUCTION SYSTEM

A full production system must be able to do the following:

- Gracefully handle exceptions that are identified in the incoming data.
- Provide warning and error messages to flag conditions in the data that require further attention. These may be conditions that can be programmatically addressed in the system or they may require human intervention to resolve.
- Ensure restart and fallback capabilities in case the system is interrupted due to processing errors or environmental issues (e.g., a computer goes down).
- Provide an audit trail to trace how data flows through the system. This is needed to help track problems back through the ETL system and/or the underlying source systems.
- Include backup and recovery of the database itself.

Several major steps are involved in the development of a production ETL system:

1. **ETL system requirements**: Up to this point, the requirements and design components of the project have focused on what the end result must look like. The data model reflects how the data is to be stored. Many detailed requirements have already been collected during other project activities, such as individual data element

names and definitions. Data profiling activities should yield insight into what the current data looks like, and strong data governance may have already determined how each data element is to be handled. Additional requirements for the ETL system must also be defined. Examples of these requirements include processing rules, guidelines for compliance with legal requirements, a processing window, and what the audit trail must include.

2. **ETL system design**: The dimensional model is the target that the ETL system will build. The ETL system design provides the details about how to get from where the data is now to this target dimensional model. Some organizations require that every little detail be defined, including all of the specific rules for building the dimension, and that fact tables be documented prior to starting to build the ETL system. Others sketch a quick high-level data flow on a napkin and then start building. There must be some balance here. It is impossible to track down every little detail that may be discovered in the data prior to starting work on the ETL system. However, it is important to create specifications defining how the ETL system will work. These are developed using the results from the requirements, including data element definitions, data mapping, and insight from source data analysis. This is even more critical if the construction work is to be done by third-party developers. Part of the overall design should also address functionality to keep the ETL system itself running, including an audit trail and backup/recovery capabilities.

3. **ETL system construction**: This is the actual development of the system itself, which may include writing programs or using technology to perform the work. This work can be divided up among different people or even different teams. Using the cohesive design, each team can work on its part. There may be some dependencies between the teams, but much of the work can be done at the same time.

4. **ETL system testing**: Because there are typically different people working on different parts of the system, it is important to conduct thorough and complete testing of the entire system. This also provides the opportunity to identify any bottlenecks and improve the system's performance. In order to prepare for testing, a series of test cases need to be developed to provide realistic conditions to determine whether the system is working properly. These test cases

must represent actual business situations and need to be defined by representatives from the business community.

As discussed so far in this book, many of the facets of building a data warehouse are different from other traditional systems design and development. The creation of the ETL system itself is the most like any other systems development effort. The organization must apply its systems development methods to create a robust ETL system. This is not the time to abandon discipline and simply move data quickly.

10.1.2. ETL System Functionality

What exactly does this ETL system need to do? The system will provide several common functions. While the concepts are consistent across the marketplace, the specifics are unique to each individual organization and its own source systems. The following sections describe the functionality and flow of an ETL system to directly populate dimensional data structures, called *presentation servers* or *data marts*. Additional steps would be needed to populate a normalized data warehouse, but most of the functionality would be similar. Let's look at several of these common functions.

10.1.2.1. Extraction

Extracting the data can be a lot harder than it sounds. The business needs information about the latest transactions, or those that have been processed since the last time the ETL system was run. It is also necessary to get the reference data that describes those transactions. For example, if an existing customer moved but has since purchased more products, you need to know the details about the customer's new location and still get the purchase transaction(s).

The challenges in extracting data are directly related to the ability of a source system to provide new business transactions and supporting reference data. Some systems identify when changes have happened and can easily share the data. Other systems effectively do what is necessary to complete the transaction but are not designed to keep track of what has changed. This may mean that the data warehouse gets a copy of the entire customer file, and the ETL system needs to figure out what has changed. Clearly, there is a significant difference in the amount of work required to sift through the entire customer file versus getting details about only those customers who have updated information.

The ETL developers need to coordinate their work with that of IT teams who work with the source systems where data is to be extracted. These other

application development and support teams may be overwhelmed by their own work. If so, the requests from the data warehouse team are just more work added to an already overloaded schedule.

Include in the project plan tasks for the source system IT support teams. Get estimated effort and lead time requirements from their manager to ensure that the appropriate resources are available.

10.1.2.2. Transformation

The term *transformation* represents a wide variety of functions that are performed to take the data as it is and get it ready for what you need to support the business decision-making process. Some of these functions are direct and straightforward, while others can be complex and challenging. There are two distinctly different sets of work to be done by developers: one to build and maintain the dimensions, and another to build and maintain the fact tables.

The dimensions provide the data needed to enable selection and grouping of data in many different ways. Creating and maintaining the dimensions often involves using data from multiple source systems. There may be some sources that are used purely to get descriptive data for the dimension. The functionality is easier to understand in a specific context, so let's look at common functions needed to build and maintain a Customer dimension:

- **Validate the customer**: Determine whether the incoming customer is already known.
- **Identify changes to known customers**: If the customer is known, have there been any changes to that customer information? This may require looking at data from several sources—perhaps the sales transactions, the corporate marketing customer database, and the accounting systems.
- **Handle changes to known customers**: For existing customers, make sure that changes are handled appropriately. Sometimes data is updated to reflect the current values. In other cases, historical versions of the customer are needed. This concept, first described in Chapter 7, is called *slowly changing dimensions*. Many other data warehouse books provide technical details about developing and maintaining slowly

changing dimensions. Needless to say, this simply means that there is more work to be done and a little more complexity to be managed.

- **Identify new customers**: If this is a new customer, then look at the sources mentioned above to determine whether the customer is identified in those systems. If this is a new customer, then a new data warehouse identifier is assigned. This is called assigning a surrogate key.

- **Build a full description for new customers**: Once a new customer is identified, data must be located to fully populate all of the customer attributes. This may require looking at a variety of data sources. It also requires guidelines that prioritize the sources. Perhaps the customer master database from the marketing department should always be the first place to look for customer information. If the customer is not found there, then use the name and address data from the source where the customer was first identified and then request demographic data from a third-party data provider. The source must be selected for each data element.

- **Validate data relationships**: Follow the defined business rules to ensure that all of the data relationships are correct. For example, check the city, state, and zip code values to ensure that they represent a valid combination. This may include standard relationships such as the geography, as well as internal relationships such as customers living in Illinois belonging to the Midwest sales region.

- **Map customer data from multiple sources**: When data is being processed from more than one source system, there must be a method to associate the data from each source. Each source system may have different customer identifiers, sometimes called the production identifiers (IDs) or codes. For example, there must be a mapping indicating that the production ID for the customer "ABC" in the sales system is the same customer identified as "1003XJ" in the accounting system. Creating and maintaining this map is a critical part of the overall transformation process that enables integration of data from different sources/systems.

- **Identify and eliminate duplicate customers**: It is common to find the same customer in each of the different data sources. However, it may be the case that the same customer can be in data from the same data source. For example, there may be data for a customer named Laura Reeves and for a customer named L. L. Reeves. By looking at the last name, variations of the first name and initial, and other differentiating data such as date of birth and home address, you can

determine whether this is indeed the same person. This processing is often called *de-duplicating* or *de-duping* the data.

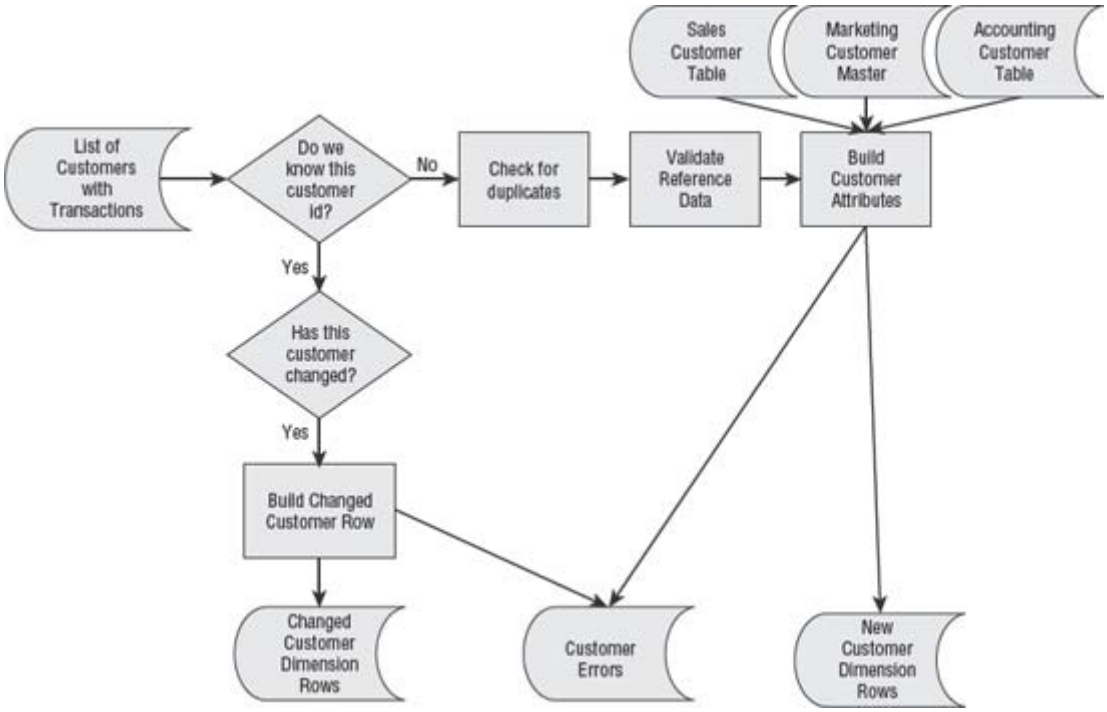
- **Restructure the data**: The customer data must be reorganized to fit into the target database design.

- **Handle errors**: Throughout the processing of the customer reference data, rules are applied to accomplish the necessary function. If for some reason there is a problem processing the data, then a determination must be made about what to do next. Previous data profiling may have already captured what should be done. Some errors are minor and may simply require a default value being placed into the customer loyalty field. In other cases, the error may be severe enough that a customer cannot be identified anywhere. The business may require that the customer (and then any associated transactions) be placed in a "parking lot" for further review and *not* loaded into the data warehouse at all. The number of errors and how they are handled depends upon how the business needs to see the data. Figure 10-1 shows a high-level view of the processing required to build this simple example of a Customer dimension.

Similarly, there are common functions that are performed to process incoming transaction data to produce the fact tables. These are explained in the context of processing customer sales transactions. Common functions include the following:

- **Isolate transactions of interest**: Depending upon how the data is provided from the source system, some processing may be required to isolate the transactions that are needed to populate the sales fact able. The sales system may process other transactions such as product returns or inventory adjustments. If these are not needed to create the sales fact table, then they can be ignored for this step (they may be needed for other fact tables).

Figure 10.1. High-level Customer dimension data flow diagram



- **Check the existence of dimension reference data for each sales transaction**: This means that the critical descriptive data such as the product, customer, and sales date are valid. Transactions can flow through the sales system with missing or invalid customer identifiers. There must be a set of rules for how to handle each special case that is observed on the transaction. It is also critical to ensure that there is a row in the dimension table for each instance that is used for a transaction. This is known as *referential integrity*.

- **Assign appropriate data warehouse identifiers**: Once the critical reference data has been validated, the identifiers must be changed from those used by the underlying source systems to those used in the data warehouse environment, the surrogate keys.

- **Validate fact fields**: Although the actual facts will be specific to each transaction, there are often general guidelines that can be checked to determine if the fact is accurate. For example, a sales transaction of zero units may indicate that this is not an actual sale. A single sales transaction for an amount that is greater than the average weekly sales for the entire company is probably a data error.

- **Translate fact values for ease of use**: The business measures needed for reports may be different from how the data is stored in source systems. It is helpful to convert data to a common unit of measure for reporting. If a sales transaction records the sale of one case, this may be a case of 24 individual units or perhaps 36 individual units. Converting all of the sales to the individual units ensures consistent and meaningful reporting. Additionally, many facts work together. For example, the number of units sold, price, and dollar amount sold are all useful facts. Only two of these need to be stored physically; the third can be calculated on-the-fly. To make this as fast as possible, it is helpful to store the two facts that are used most often.

- **Unravel source system logic**: In many instances, the source system has core data stored in a manner to help that system run fast and enable each module to be relatively self-contained. This may mean that the pieces that are needed to get a complete picture of a sales transaction may be stored in several places or tables. To make things more challenging, there may be no direct links between these tables, although there may be a series of translation and lookup tables that are needed to get to the bottom of things. This may be why the data is so hard for you to use in the source system. The details are all there, but you need to understand the "secret handshakes" in order to get to the real numbers. The ETL system can apply this logic to provide the real sale numbers to be loaded into a fact table. This is done once and then everyone can access these facts without having to learn all of these details.

- **Perform complex calculations**: Many calculations can be done when creating a report or accessing data in an ad hoc manner. However, some calculations may be too complex and long-running to perform on-the-fly. The ETL system can perform these calculations and store the results as facts, which can then be directly accessed for reporting.

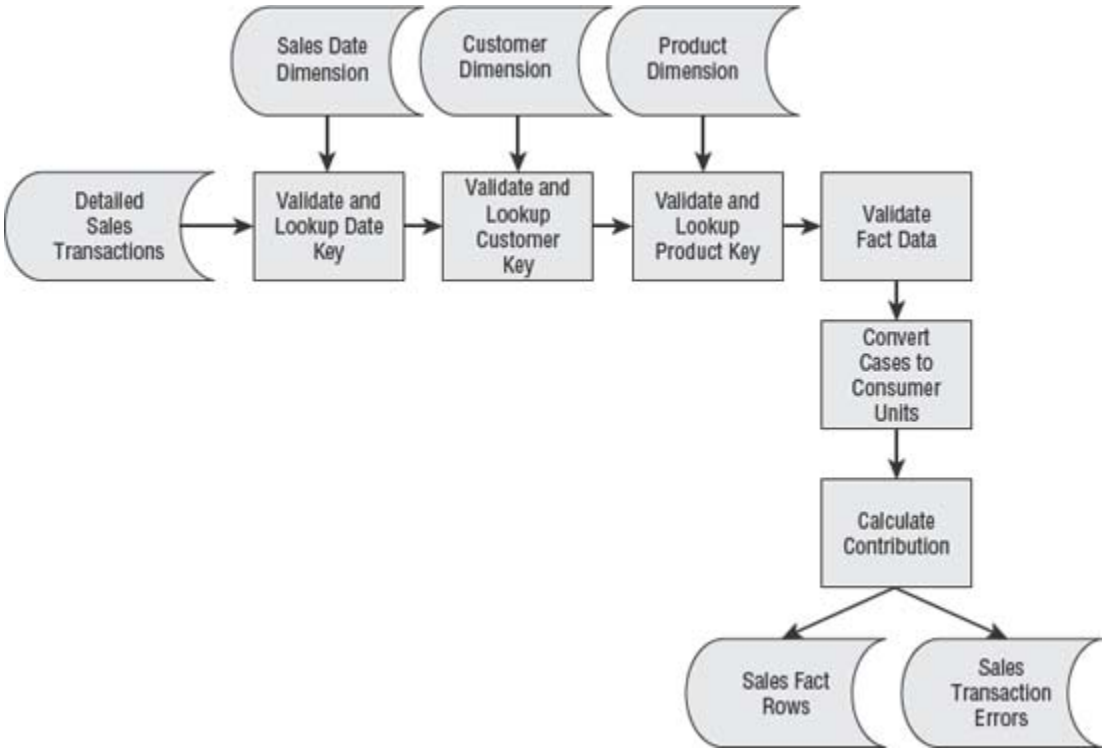
Figure 10-2 shows a general data flow diagram to build the sales fact table just described.

In addition to the functions described here, a great deal of work is also required to address anomalies and challenges in the data. Often these challenges surface when the ETL system is being developed. No matter how carefully the data is profiled in advance or how much detail is included in the design, new and unexpected things will show up in the data. The goal is to design the ETL system to be able to gracefully identify and address the unexpected.

10.1.2.3. Load

After the extracted data has been processed, the final step is to load the data into a database. This may be loading directly into a star schema in a presentation server or loading data into a third normal form data warehouse.

Figure 10.2. High-level sales fact table data flow diagram



10.2. The Business Role in ETL

The participation of the business community is fairly obvious when gathering business requirements. Business participation in the construction of the data warehouse is less obvious but just as important. At this point in the project, the business people who are needed are those with in-depth knowledge about the current data and reporting methods. After looking at why these business people are needed, this section highlights the areas in which business involvement is necessary.

10.2.1. Why Does the Business Need to Help?

There will be a variety of skill and experience levels on the development team. Without guidance from the business, you are expecting the project team to make these critical decisions about the data. If there is limited access to the business community, and pressure to deliver something, the project team is likely to forge ahead without business input. Consider what really happens: The technical team members decide what they think is better or worse, and then one of the ETL developer makes the decisions. *Do you really want the business performance results to be reported from a database that was loaded with rules defined by the technical team's best, and, it is hoped, educated, guess?*

Such an approach rarely delivers results that the business really wants. This often results in a subsequent project to redefine, rebuild, and correct the data to reflect what is really needed. Many organizations continue to apply pressure and do not commit business resources, yet fund second, third, and fourth efforts to get it right. Make the investment now so that there will not be a need to redo the data warehouse repeatedly. In the following sections, you'll look at what needs to happen now, to get it right.

10.2.2. Defining Business Rules

The overall design of the ETL system is clearly the responsibility of IT, so what role does the business play in the design process? The business must define and provide the rules about how to process the data. Fortunately, there are dozens of data elements that do not require much work. These data elements can be validated and then flow directly into the data warehouse. However, it is not always that straightforward. Some of the things where business input may be needed include the following:

- **Helping to pick the data source**: Suppose there are three different systems that have the customer's home address. Which one should be used? Perhaps you should always use the accounting system first, then look in the sales database, and, only if it is not found there, go to the customer feedback system. Candidate source systems will be researched by the IT team members, and their findings can help with the decision-making process.

- **Defining processing rules**: Representatives from the business community need to work hand-in-hand with the design team to define the rules for handling the data. What should happen when looking at a sales transaction and the customer does not exist in the reference data? Should the sales transaction be loaded with a default customer identifier? Should that sales transaction be set aside for further research before it is loaded? The business community must provide the input so the appropriate logic can be built into the ETL system.

- **Defining integration rules**: The business groups are often already using the data that will be loaded into the data warehouse. Suppose reports are being produced with data from multiple sources. How is this integration being done currently? Perhaps it is being processed in an existing reporting system. Perhaps it is pulled together manually to create the reports. The person who is doing this work has the most knowledge about the details regarding the matching rules and how to handle exceptions. The data warehouse team must work with the individuals who have this in-depth knowledge to determine whether this same logic should be applied in the ETL system. Sometimes there are no existing integration rules to even start with. In that case, the IT and business representatives need to work together to ensure that the data will be integrated appropriately.

USING EXISTING LOGIC VS. STARTING OVER

Over a number of years, very complex logic can be embedded into reporting systems. Sometimes the logic is applied to load data into a database, but in many cases the logic is applied when reports are

created. This means that if the logic changes, it needs to be changed everywhere. Unfortunately, over time, there can be a divergence of this logic across different reports. In addition, this logic can be a series of rules that are applied in layers. For example, the logic may first get the subset of new customers. Next, logic is determined using a series of codes to identify the source of their business (agents, the web, or other channels). Finally, a complex formula is applied to determine their sales potential. This is a simple example to communicate the concept. In reality, business logic can be extremely complex, with many layers.

The challenge for the data warehouse team is to determine where to get these rules. Some organizations can have dozens of SAS programs with many lines of code to apply the business logic, and the original author of the logic is no longer with the company. The IT team members can reverse engineer the programs and/or reports to derive the existing business rules. Then, the business team members need to review those rules to determine whether they are still appropriate. However, another option is to start over and develop rules from scratch. This can be a better choice if the rules have been in place for a long time, the business itself has changed significantly, or the rules were not clearly defined and applied consistently in the current environment. Which option to pursue should be a joint decision between IT and the business.

10.2.3. Defining Expected Results–The Test Plan

A comprehensive test plan is needed to ensure that the data is correct. The business must provide a set of test cases that can be used during development and testing. This gives the baseline for comparison with the data warehouse. For example, to validate earned premium, the business can provide a report with actual earned premium by state by calendar month for the last six months. This can be used to compare with the data pulled from the data warehouse.

If there are major changes in the business rules, an existing report will not suffice for comparison. A subset of raw data may need to be pulled and manipulated manually to provide the results for some test cases.

Some of the best people to help with the initial data validation are those individuals who, if required to use the new database, would immediately look for flaws. They would compare reports from the data warehouse with reports

that they have been using themselves for years. This business participation also helps others to have more confidence in the data because people they respect have helped validate it.

Some organizations have separate testing and quality assurance groups. This requires that the test cases and instructions to run the system be provided. The people performing these tests may have very little exposure to data warehousing or the data you are loading. This requires that more care be given to developing and documenting these test cases.

10.2.4. Development Support

Once the design work has been done, development can begin in earnest. This does not mean that the business representatives are off the hook. Although a great deal of thought goes into all the work leading to a complete ETL design, there may still be things that require business input.

As the team begins to work with the actual data, performing good detailed data analysis, there will be more surprises. These can be minimized with good data analysis, ETL requirements, and ETL design work up front. Data that was expected may not exist. Data that was not expected will show up. There will be many exceptions. Often, with a little legwork, the technical team can figure out what happened and adapt the development appropriately.

There will also be plenty of times when the business needs to be consulting about what to do next. How should it handle invoices for customers that do not exist in the customer master file? Although this should not happen, strange things are often found in the real data. Adjustments may need to be made to the underlying source system, but the ETL system must know what to do if this ever happens again. Should the invoice be loaded with an "unknown" customer? Should the invoice be held for further research? The development team can do either, but the business needs to define what they are expecting to see.

The business team members will have fewer daily tasks assigned to them, but they will be "on call." When a problem is found or questions arise about how to understand a business rule that was defined during the design process, the team needs to be able to get answers quickly. They do not want to wait for permission to ask a question that may take five minutes to answer.

10.2.5. Testing the ETL System–Is the Data Right?

Initial testing of the ETL system will be performed by the technical team members. This ensures that the programs and processes run without errors and that they correctly perform their designed function. This often includes basic checking to ensure that the data flows properly. The team will make sure that if 100,000 transactions came in, then 100,000 transactions were processed. Other checks may include confirming that the total units shipped on the incoming transactions matches the total units shipped from the resulting fact table. The basic testing may also ensure that each customer is assigned a credit profile and that each product has a package type.

Testing that the programs and processes run correctly is only part of the overall testing and quality checks that must be done. Additional work is needed to ensure that the data itself is accurate. The ETL system may have correctly processed those 100,000 transactions, but if all of the sales were off by a decimal point, then the tests previously mentioned would not detect it. The business must be an integral part of the testing and quality assurance process to ensure that the database is correct.

A great deal of work is required to validate the database when it is first loaded. There will also be ongoing work to ensure that each time the ETL system runs, accurate data is loaded. Who decides if the data is right? The comprehensive test plan developed earlier by the business provides the basis to assess the data. Representatives from the business community may be involved in running the test cases, but their involvement is critical in evaluating the results. This needs to be someone who has the knowledge to recognize that an increase in market share of 14% is not possible, as changes in market share are usually measured in tenths of a percent.

The ETL system should include many validation steps to check for data problems before the data is loaded. Leverage your data access or business intelligence tool to help with data validation. These tools are designed to identify exceptions, compare results with previous periods, and compare results to thresholds. Develop a set of queries or reports that can be run after the data has been loaded as a final data quality check.

10.2.6. Why Does It Take So Long and Cost So Much?

If all of the systems in the organization contained clean, reliable data with consistent production identifiers, then developing a robust ETL system would not take so long. However, it is rare to find an organization that has such clean, integrated systems already in place. Therefore, with the reality that the source systems were designed without the requirement to integrate easily with one another, the data warehouse must overcome these problems.

A variety of factors contribute to the time it takes to build the ETL system. Several of the biggest contributors include the following:

- **Long decision cycle**: Many choices need to be made throughout the design and development of the ETL system. Often, decisions can be made based upon the requirements, or a single person or area needs to be consulted first. However, in many instances the decision impacts multiple areas. It can take a long time to get the appropriate people together to understand the choice, come to an agreement, and commit to the direction.

- **New perspectives on the data**: As part of building the data warehouse, the data may be looked at in ways that have never been attempted before. As the business and the marketplace evolve, new requirements also emerge. These may result in identification of anomalies and challenges in the underlying source systems. As new perspectives are defined, the data may not yield meaningful or helpful results. For example, it may take several iterations to define the business rule to identify long-term customers. How many years are needed to consider someone a long-term customer? What if the customer leaves and then comes back again—do you count all of the years or just consecutive years?

- **Lack of business input**: There are also many demands on the business community. Business analysts who work with the data today are often swamped with demands for data, reports, and analyses. In addition to a full load of regular reporting, there are often ad hoc requests that need attention. However, these analysts have the most knowledge about how data is manipulated and which calculations are currently used. Sometimes, the team must go through a series of meetings before identifying the person who can really help.

- **A lot of hard work**: Often, developing the ETL system takes a lot of time and money simply because there is a lot of work that must be done. This can be due to the complexity and volume of data or because of a lack of well-structured and reliable data to begin with. Obviously, if the team is pulling data from two data sources, the integration work is much smaller than integrating data from ten different systems.

- **Insufficient data profiling**: Whether the organization is using a sophisticated data profiling tool or performing detailed data analysis by running queries against the database, it is important to understand what data is being stored. When the actual data contents have not been studied, problems will arise during ETL development. The types of data problems that emerge include finding that the data element is empty or that the contents of a data element are not what was expected. Thorough study of the data prior to designing and building the ETL system should obviate these issues.

- **Indirect communication**: Looking at the day-to-day work, questions often arise for which the team needs input from another IT resource or someone from the business community. If the project team must go through a liaison to gain access to other IT or business people, this can greatly lengthen the project schedule. While it may not seem like a big deal to wait a couple of days to meet with a key IT or business person, over the life of the project this can add weeks to the schedule—just waiting to have the opportunity to ask a question. Moreover, if the question cannot be answered by that individual, then the process begins again to gain access to the next person in the chain. It is much better to create an environment where the data warehouse team is provided an open door directly to anyone (within reason!). This requires that both business and IT management communicate the importance of the data warehouse project to everyone else. It also means that the data warehouse team must act responsibly and respect everyone's time.

- **Experience level of the team**: There is a lot to learn when working on a data warehouse project, especially for the first time. It takes some time to learn data warehousing concepts and principles. It also takes some time to learn and *become efficient* with using any new technology.

- **Lack of access to the right IT people**: Usually, several key people have knowledge of how a system works. These IT people are in high demand, and the data warehouse team may have trouble scheduling time with them. These individuals often play a critical role in keeping the operational system running and may be involved in other systems development projects.

Each of these items can increase the amount of time needed to design and build the ETL system. This is also the part of the project where you are likely to have the largest number of resources all working on the project. A small core group can gather requirements and develop the data model, but now many more people can work at the same time to build the ETL system. Small organizations may still have a small core team (likely the same core team),

whereas large enterprises may have a dozen or more people working on this. All this work translates into dollars, for internal business and IT staff and third-party consultants.

Use what you know about the organization to anticipate common project impediments. Try to build time into the project schedule to deal with these. For example, if the team is inexperienced, then include time and resources for education and mentoring.

10.3. Balancing Requirements and Data Reality

Organizations often have great ideas about what they want to do and how they would like to use data to help them. However, the data warehouse is limited by the data that the organization actually has. The business requirements can be identified using the techniques discussed in Chapter 6. Many of the shortfalls of the current data are found during data profiling and/or detailed data analysis performed while data modeling. If the data is not captured anywhere, then it cannot be included in the database. Additional data challenges are discovered during the development of the ETL system.

It takes a lot of detailed, tedious work to track down and resolve all of these individual data issues. It is important to ensure that issues are well understood so that decisions can be made about how to deal with problems that arise. In some cases, getting to the bottom of the problem itself may take a lot more research. A decision must be made whether to work on the problem or to postpone it for the future. This must be a joint business and technical decision. Some problems can be put off with little or no immediate impact, but some data issues must be resolved in order to meet the overall objectives of the project.

For example, suppose the organization has been collecting customer demographic data for years. When customers call in, they are asked if they are willing to complete a short survey. This short survey collects additional demographics about each customer household. While it sounds interesting to use for analysis, most customers did not participate, so only 15% of the customers have any data. To make matters worse, the entry screens required the answers to be keyed in, rather than using a set list of options, so the data that has been collected has many different values and will require a lot of cleaning to make it useful. The question at hand is whether this is worth the effort.

Because the demographic analysis is not an immediate priority, and the work required is significant, this was postponed to a subsequent iteration. In the meantime, a better data solution is to modify the survey entry screen to capture pre-set options so that the data is consistent. In addition, the top five most important questions need to be included in the initial conversation with the customer, rather than as an optional survey. These decisions need to be based on a cost-benefit analysis—not a multi-week effort, but simply a checkpoint to ensure that resources are used wisely to deliver the most value in a timely manner.

Often, the data problems identified when working on a data warehouse project are data quality problems in the underlying source systems and/or business processes. It is important to dig down to find out the root cause of data quality problems. Then, decisions can be made to eliminate the problems from recurring.

10.3.1. Discovering the Flaws in Your Current Systems

As a by-product of the detailed work that is done to extract and then transform the data, many anomalies and unusual data handling and storage techniques are uncovered. Sometimes fundamental flaws are identified regarding how the source application system works. If these flaws impact how business is conducted, they must be addressed immediately. For example, if the formula used to calculate sales tax is not correct, this must be addressed. This would need to be corrected in the system, and follow-up research is also required to determine other ramifications. If the tax was too high, does that indicate that a refund must be paid to customers? If the tax amount was too low, what are the implications with the IRS? Addressing problems like this is difficult and highly charged with emotion. This may slow the project while the situation is being addressed. While this is usually unpleasant, it is imperative to discover and correct serious flaws.

More often, less serious problems are found while building the ETL system. The application system functions properly and business is successfully completed, but this type of problem is after the fact, in how data is reported and tracked. Many organizations have copies of production data that are used to support reporting. While these meet some basic needs, it is not enough because you are building a data warehouse environment.

Over the years, a lot of logic has been embedded into these reporting environments (databases, files, spreadsheets and/or report programs). Often the logic has not been well documented and the authors may no longer work in the group or for the organization. There is often sophisticated

and complex logic applied in two primary ways. The first type of logic performs calculations. This is the most obvious and easy to understand. Calculations are used to determine commissioned sales, financial contribution, and gross margin. The second type of logic has just as much impact on the results as the specific formula that is used. It is this second type of logic that has been included and/or excluded from participating in the calculation. For example, all sales to non-profit organizations are not eligible for sales commission. This means that while the formula is the sum of sales dollars, the sales to customers designated as non-profit are excluded from the summation. The logic regarding what is included and excluded from reports and business measurements is much more difficult to track down, but just as important. Over the years, the business itself may have changed, but all of the underlying logic in the many different reports may not reflect the current view.

10.3.2. Applying New Business Rules

As the data warehouse is built, the data will be cleaned and transformed using the currently defined rules. This often means that the data warehouse results will not match reports from the older systems. Clearly, the processes and results must be validated to ensure that the new rules have been implemented correctly. However, because the new and old systems are applying *different rules*, the results will not match.

This is often difficult for people to understand, and it creates concern because the reports that have been relied upon for years have been exposed as no longer accurate. In most cases, the reports were accurate and served the purpose as defined when they were developed. It is time to move on to how the business is being tracked and measured today. For most people, it is hard to give up the current way of doing things. It is comfortable; and while not optimal, it is well understood. The changes to business rules should be driven from business requirements and decisions made during design and development of the ETL system. It is helpful to have the business people who were involved in making these decisions communicate what has changed and the rationale behind the changes. Moving forward requires learning and changing—two things that many of us don't enjoy.

Don't focus on why the old system was wrong, but emphasize how much better it is to use new measurements, formulas, and ways of looking at the business. This new perspective has evolved as a result of what was learned in the past. You are leveraging what was done in the past to make things better for the future.

10.3.3. Working Toward Long-Term Solutions

Depending upon what you find, some issues may simply be too big to address at this time. If a problem originates in a major application system, the solution is likely to be much bigger than the data warehouse. Often the team that is responsible for the application system heartily agrees that there is a problem and they often agree that the problem must be addressed. However, their priorities may differ from those of the data warehouse team. The effort to address the underlying problem may be slated with the next maintenance release, which is scheduled to start 24 months out. The data warehouse may not be able to wait that long. Therefore, many techniques are employed and a lot of work is done by the ETL system to overcome these problems with underlying systems. Over the years, real change is so infrequent that too often data warehouse teams don't even think to ask for modifications and enhancements to the underlying source systems.

Don't live with the status quo. Regardless of how clever the data warehouse team is, it will be better for the long term to have clean data captured as close as possible to the business interaction or transaction itself. The following guidelines should help:

- What is the real source of this data problem and where should it be fixed?
- Encourage the data warehouse team to look for long-term solutions, rather than short-term patches.
- Follow up to ensure that modifications and enhancements are actually made to the source application systems.

In order for your organization to begin to shift in that direction, there must be a commitment and understanding with the management team, both business and IT, that this will be an investment, a change in how things are done, but it will result in long-term benefits. These benefits include saving the cost to fix data problems later, and improved data quality that results when data is captured closest to where the real details of that business interaction are known.

10.3.4. Manually Including Business Data

There are instances where critical business data is not currently being captured in a formal operational system. This may be data that is needed to

complete the required reports and analyses. Often this data is stored in spreadsheets on someone's hard drive or on a network shared drive. It is not acceptable to have anyone manually enter data into the data warehouse.

The first thing to do is explore the need for a small operational system to support the business function that captures this data. If the need is identified, a separate project should be initiated to address that need. Sometimes, the only need is to provide a place to collect this data in an organized, systematic manner. A small subsystem can be developed to provide a user interface that enables the data to be entered into a staging data table, rather than having the data entered on a spreadsheet. This staging data can be used as input to the ETL system. This adds more checks and balances to the data entry process and ensures that the data is integrated in a consistent and approved manner with the rest of the data in the data warehouse.

10.4. Tracking Progress–Are We There Yet?

The design and development of the ETL system is very involved and can take quite a bit of time. It does not help to have only three tasks on a project plan: design, develop, and test. While these are the big tasks, it can be difficult to track real progress at such a high level. This may be all that is needed to share progress with upper management, but at a working level, more detail is needed to appreciate the work that has already been done and to see how much is left to do.

The tasks can be broken down to list each dimension and fact table, or even further if necessary to achieve small enough units of work to be done. As the detailed design, development, and testing is completed for each, it can be tracked, allowing a more accurate assessment of what percentage of the work has been completed. It is helpful to know that the ETL system is 60% complete; it is more interesting to know that eight of the ten dimension tables have been developed and one of the five fact tables is done.

Progress can even be measured in the number of known data problems. Then you can see progress as these are resolved. Without this, you don't even know how many known issues still need to be dealt with. Are there 25 or 125? Encourage the team to keep a log of all open questions. This can be used to track what still needs to be addressed. Assign a tracking number, a general category (perhaps the name of the dimension or data source to which the question relates), include a description of the problem or question, the date it was identified, the person responsible for finding the answer, and the status (opened, closed, deferred).

Over time, include notes about what has been learned and any decisions that have been made. This also serves as an audit trail showing how data issues have been resolved. This type of a log can easily have more than 100 items. The items can be prioritized to help the project team focus on the most important issues first.

At some point you are likely to get down to a dozen issues that remain open. You need to assess which of these must be addressed. From a systems perspective, can you move forward without having an answer? From a business perspective, is this issue critical? What happens if you can't find a resolution? If there is little or no impact (on systems or business), then this may be postponed to a future iteration.

10.5. What Else Can You Do to Help?

While the ETL developers are working diligently, the rest of the team can feel helpless. However, there are several concrete steps during the development of the ETL system that you can take to contribute to the project's success.

10.5.1. Encouragement and Support

Project teams are usually very aware of the time pressure to deliver results. They put in a lot of effort, often giving some of their own time to keep things moving forward. With so much work to be done, it can feel like a thankless job to the ETL developer. At times, more problems are identified than are being resolved. After a while, it can seem like the work will never end. It may still be weeks before the ETL system is complete.

Too often, the ETL developers only hear complaints and other negative feedback. Every time they need something, it feels like they are interrupting other more important work. The only time you hear from them is when they have found another problem. This is when mangers and members of the business community need to step forward with a few words of encouragement. This can be a simple e-mail to thank them for their efforts. Even better, stop by in person to thank them. If it has been a particularly bad week, bring in bagels, donuts, or even homemade cookies. You don't need to spend a lot of money, but a simple acknowledgment can give the team a sorely needed boost.

This type of support is often viewed as "management's job." However, it can also be given to the team by anyone in the business community. Remind the project team why this matters to you and how it will help you with your work every day.

Business representatives can help the project team by keeping a positive attitude, even when other demands are

looming. When asked, set aside time to answer questions and focus on what the data warehouse project needs.

10.5.2. Ensuring Continued Business Participation

The construction of the database often takes longer than any other part of the project. For smaller projects this may still be weeks, but for many large initiatives, this can take months. Database construction is not as glamorous and exciting as brainstorming requirements and developing a data model. This is detailed, technical work. Many business team members, as well as the business community in general, can lose interest in the data warehouse.

Once the design work is done, the time demands on business partners decreases. Other daily work easily fills in and begins to take precedence over the remaining data warehouse work. While there is not a need for the business team members to develop programs and technical processes, their continued input is still highly valuable. Business input is needed to answer questions, clarify data requirements, make decisions about business rules for processing, and assist with testing and validating the data. It can be difficult to estimate when this participation will be needed for the next several months. The project team may have a better sense of what to expect in the next couple of weeks. While the participation is not constant, it is still important to ensure a successful data warehouse. Remember that any delays in helping the project team will add up and delay completion of the entire data warehouse.

COPING WITH UNCERTAINTY

The one thing that is guaranteed on a data warehouse project is uncertainty. There is the uncertainty that results from the very nature of trying to keep up with the pace of business and the changing marketplace. There is the uncertainty that results from scrutinizing the organization's data in ways that have never been done before. The most important step toward dealing with this much uncertainty is openly acknowledging that it exists. There are also several things that everyone can do to help:

- *Be patient* with each other as the project evolves.
- *Remain flexible*–Realize that it may be best to get started with some work before you have every "t" crossed and every "i"

dotted. Even if you achieved that level of detail, many things will change as soon as you move forward anyway. This does not give the project team carte blanche to run at full speed without any planning; it simply means that you must find a happy medium: Do enough planning and design to ensure that you have a strong direction, yet be willing to adapt as you go.

- *Communicate openly and frequently*–This is the key to ensuring that progress is being made toward the ultimate goal: a sustainable data warehouse that helps the business realize concrete value for the organization.
- *Apply sound project management*–Continue to utilize standard project management techniques to run the project. This includes maintaining the project plan, watching for scope creep, administering change control as needed, continuing to conduct periodic status meetings, and publishing status reports.

10.5.3. Proactive Communication

Project communication should be driven from the project team out. The project manager is the spokesperson to share progress, concerns, and changes in timeline, deliverables, or costs. This is often through the standard project office or project management channels of the organization. Usually, weekly updates are submitted and included with all of the other initiatives across the organization. A one-page snapshot with a few bulleted points and a green/yellow/red indicator is *not* sufficient communication between the project team and the key stakeholders of the data warehouse.

During the early requirements gathering and data modeling phases of a project, there is natural and frequent interaction between the project team and the business community. Now that the team is focused on ETL system development, the nature of the communication must change. There will be many weeks when there are no major issues that need attention and development work continues. For very large projects, this may go on for several months. It seems that there is nothing specific to say, other than "we are still working."

While regular status reports are published, there is often a drop-off in other communications. There are often no executive briefings, and no meetings with business sponsors and drivers. While these should continue, experience

shows that when there is no major deliverable or milestone accomplishment to report, project teams are quiet. If you don't hear from the project team, be proactive. Request a meeting or ask for an update.

Another reason to stay in close communication is that business and IT management can help remove many roadblocks for the team. Too often, project teams feel that they should be able to solve everything themselves. For example, one project team was running preliminary tests for one part of the ETL system. Every night, the server crashed and their processes never completed. With some investigation, the team discovered that the machine they were assigned for development and early testing was also being used by three other projects. The machine was overburdened, and did not have the resources to support all of these projects. From the ETL developer's perspective, they set up a schedule to take turns checking on the server throughout the night.

Over time, this decreased the productivity of the team (because they were all tired) and the testing took much longer than expected. With a broader perspective, other servers were available and could have been allocated to the data warehouse team. When you are too close to the problem, sometimes you don't see alternative solutions. Regular communication between the project team, business, and IT management can help identify these opportunities. Therefore, if you don't hear from the team, then seek them out to see how they are doing.

10.6. In Real Life

Let's check in with the two companies we have been tracking now that they have moved into design and development of their ETL systems.

10.6.1. Building the Data Warehouse at Giant, Co.

Job functions and roles are highly specialized at Giant Co. Everyone does their part and rarely looks beyond their immediate task. The organization has one group of people assigned to develop the technical and functional specifications for building the ETL system, but there was some trouble getting access to the right business people to develop the specifications. Therefore, the team just made the best decisions they could to complete the specs on time. After, with iterative development, there will be more work in the future. The developers are highly skilled and march forward to build what has been designed. However, as work progresses, many things crop up with the real data. The specifications do not include any instructions about how to handle these issues, and it takes too long for questions to be answered. The developers do what they can, but often data is excluded because there is no direction for handling it. This keeps the schedule on track, but as more and more data is left out, overall value begins to diminish. The entire project is still taking longer than expected.

A great deal of animosity also grew between the design and development teams. Changes continue to be made to the specifications to address discrepancies in the data. The developers feel that if the design team had done a better job this would not have happened. Likewise, the ETL designers feel that if the data modelers and analysts who gathered the requirements had done their jobs better, then there would not be so many changes.

With the situation deteriorating, the project manager called a time-out. The project team met to review the status and what would need to happen to get the project back on track. The project manager then met with the business and IT sponsors to share these findings. Although it was hard, it was decided

to take a step back and develop a new plan of attack. These activities included the following:

- Compiling an inventory of what parts of the ETL system were working and what were not
- Comparing the project scope with the data currently being worked on, studying what needs to be added back in to meet the business objectives
- Documenting known data problems
- Revising the project plan, including cost and time estimates to complete the rest of the project
- Gaining approval for the revised project plan
- Developing specifications for remaining ETL work
- Assigning appropriate business people to assist with the specification and support the rest of the process
- Conducting team-building sessions to defuse the situation

While it was not easy to decide to revisit the requirements for the ETL system, it enabled the team to get back on track, with management support. Much of the work could be used. The experience and lessons learned were also documented while the issues were still fresh.

10.6.2. Agile, Inc., Builds a Data Warehouse Quickly

There is a great deal of pressure to move quickly at Agile, Inc., to get the data warehouse up and running. The longer it takes before data starts moving, the more anxious the business gets. With the business requirements gathered and the database designed, the team gives in to the pressure to start building the database, with minimal time spent writing any specifications for the ETL system. The general consensus is that the team knows where the data is today and what it should look like, so it's time to get started. Everyone is empowered to make decisions and get their work done.

Sometimes business input is solicited to address data questions. This is not well received because everyone is so busy pulling data and producing reports for the upcoming quarterly management meetings. Therefore, the ETL

developers just forge ahead and don't bother the business with questions. After a short development time, data is loaded and deemed ready for use. However, now that the business begins to look at the data, serious problems are identified. It turns out that many of the assumptions used to process the data were not valid. While the team met the short timeframe, the data is not usable.

After a great deal of finger pointing, a team is put together to conduct a review. The purpose is to determine whether it would be better to clean it up or start over. Because very little was documented and there was no cohesive system design for the ETL processes, it is determined to simply start over. This does not mean starting from scratch—the project team learned a lot about the data and about ETL development. Before diving in to build more processes, a joint business and IT data governance team is formed to study the data and develop standard definitions and guidelines regarding how data elements are to be handled. With this joint team to help answer data questions, the new project includes time to develop complete specifications. The project is going much better and the preliminary results are good. The data will be trustworthy and help meet the business requirements.

10.7. Summary

The ETL system is at the heart and soul of the data warehouse and must be driven by detailed requirements for the data. Taking the time to understand and document these requirements helps to design and develop a robust production system. A lot of work is done by the ETL system to extract, transform, and load the database. This chapter introduced the basic functionality of these steps.

While much of this work is technical in nature, the business perspective must still be represented. Business representatives must be involved in defining business rules for handling the data and a test plan and detailed test cases. Once development begins, the business must continue to support the ETL developers by answering questions about the data promptly and enabling cross-functional group decisions when necessary.

The business needs to continue to provide input and guidance to the technical team members to ensure that the data is prepared properly. Continued partnership between business and IT can help each data warehouse project to successfully load high-quality data. With the data loaded, it is time to take a closer look at how it can be used. The next chapter focuses on delivering data into the hands of the business community.