1

Data Modeling: an Overview

In this chapter we will take a look at what data modeling involves and who, within any given organization, actually does the modeling. We will look at:

- ❑ The nature of data versus information
- ❑ What data modeling is
- ❑ How data modeling helps in dealing with the lifecycle of Enterprise data
- ❑ Who actually performs data modeling tasks in the organization
- ❑ What activities the data modeler participates in

What Is Data?

Data is the description of facts, rather than information, which is the useful interpretation of data. As the ultimate goal of data is to provide useful information, it must be modeled in such a way to aid that aim. While once data was represented simply by text and numbers in a database, today its abstraction into digital format has expanded to include pictures, sounds, virtual 3D objects, and complex multi-media blends of all of those things. That evolutionary growth of the very nature of data is challenging us all as to how to use technology to its maximum capability.

Even the subjects and focus areas of data have changed. We used to think of data mostly in terms of Inventory and Accounts Payable systems. Now we add Ground Tracking Systems using dynamic uplinks to satellites that help us find our way around. We see Virtual Gaming allowing people from around the world to play and communicate with each other in real time. There is Online Shopping that remembers who we are and what we might be interested in, and cameras on public buildings using face recognition software to watch for trouble.

While people seem to want to collect ever-increasing amounts of data, they also want to have it manipulated in order to provide aggregates, trends, logic-driven prioritizations, derivations, and mappings of data to other things of importance about the data (meta data). In order to support those desires we need to provide data in structures that support different functionality like:

- Online Transactional-Processing (OLTP)
- Operational Data Stores (ODS)
- Data Warehousing (DW)

Our current view of data management tells us that we need to collect and store data in the most elemental, atomic, basic, and reusable parts possible. Only by doing that can we build relationships that support the creation of newly defined information. And since we frequently need to combine those basic data elements, we need to design appropriate storage and processing facilities that allow this to be accomplished.

The process we use to discover those basic data elements, determine how they relate to each other today, and define them so that they can be recognized and used in the future, is called data modeling.

What Is Data Modeling?

Data modeling is a technique that records the inventory, shape, size, contents, and rules of data elements used in the scope of a business process. The business process scope may be as large as a multi-disciplined global corporation, or as small as the receiving of boxes on a dock. The final product is something like a map accompanied by all the backup documentation needed to interpret it with complete clarity.

Some models are built to document very high-level ideas and are termed **conceptual** models. Some are built to document the theoretical purity of data element rules and structure, called **logical** models. And maybe the best known type of data model is the type that determines the actual design of a database, called a **physical** model. This model is the basis of the code written to create tables, views, and integrity constraints. These three basic types connect to each other loosely and can support each other as different views of a single business process.

A data model does not provide the full compliment of back end code that needs to be built to support an application. It does not contain security grants, or database link code. It doesn't show sizing and space needs (although some modeling software will let you add in those aspects). The data model is a blueprint for a database, *not* the database itself. It's a cornerstone of the development effort, but only one of the building blocks of designing an application.

The process of modeling can capture a representation of something that exists, the **As-Is** state of our physical data, so that we can view how the data is structured in an existing database, and what rules the database manages in an environment today. These views are generally restricted to a single application, or a set of tables in a restricted schema like a database. It could take hundreds of them to fully document the scope of an Enterprise, for example. It could also take hundreds of logical models to describe the more theoretical rules of the data, rather than how they were physically implemented. Similarly, there could also be many conceptual models covering all the business processes, although it is more likely at the higher level of the conceptual model, than it is at the other stages, to attempt to cover all the Enterprise activities in a single view.

A model helps us to view how we manage data now or did in the past.

Data Modeling is also used to create designs for something new, the **To-Be**. Sometimes we create multiple To-Be designs, and the development team chooses which one to implement.

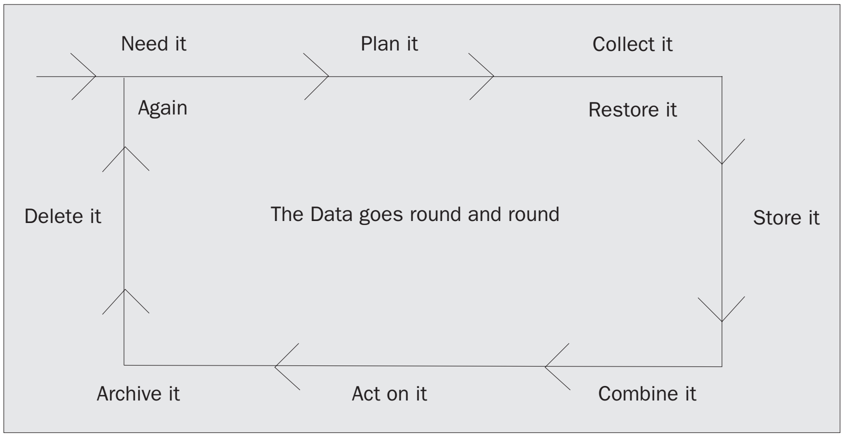
> **We model to try out options. There can be a number of solutions even for similar problems, and it isn't always easy to know the right one.**

However, whatever it is used for, data, in its elemental form, needs to be understood and should be cared for with a disciplined approach to its lifecycle. Before we continue looking at data modeling techniques, let's look at the lifecycle of a typical data element.

Data Lifecycle

What follows are the stages that new data goes through on its way from being used as a recent fact, to becoming a historic fact, and eventually becoming a forgotten fact. (A stage that we are having increasing difficulty believing in, due to repeated requests by clients to unearth their archived data or restore it from backups.)

This is not the lifecycle of an application development project, although many other processes can go through similar stages. These are the steps that any data element goes through:



Need It

This is the point in time where someone asks for some data. What was the temperature in Des Moines yesterday? How many candy bars were bought before the PG movies last week? How many parking spaces are going to be available before second shift shows up? Let's consider the example of a Credit Card Transaction Chargeback Amount. Say you are a small retail firm, and you want to begin accepting credit cards to pay for purchases. You contact several different credit cards to find out how. One of the things you discover is that different credit card companies charge different fee amounts for using their service. The fees even differ under different circumstances. You realize that in order to understand what providing this service to your customers is going to cost you, you need to collect the service fee for each transaction separately to be able to analyze profitability better. This recognition of the need for a data element is step one.

Almost no data element is captured just because it exists. It needs to be of value to someone, or there need to be belief in a future value, to bring it to life. As you find out what the impact of the new data element is on the organization, you may discover other data elements also need to be available.

The 'Need It' phase is the recognition of a new fact, which needs to be available in order to accomplish a task. You now know you need a Credit Card Transaction Chargeback Amount in order to assess profitability in your business, determine Return on Investment (ROI) by looking at service usage and cost, and even plan next year's budget. So what happens after you know you need it?

Plan It

This is the stage where the exploration and analysis of the data element happens. Questions are asked about frequency, size, and business rules. Methods of capturing and storing it are explored. The security, reliability, and quality issues are looked into.

How much can a Credit Card Transaction Chargeback Amount be? Is it determined by the amount of the sale and so presented as a percent? Is it a flat fee per transaction? Do we know when we charge the customer what the service charge to the company will be, or are we charged at a later time? If so, how do we tie it back to the individual transaction? Is it credited back to us so the amount could be a + or a – value?

When we plan for it, we look at all sorts of different things about it and document as much as there is to know about this new data element, including where we think it fits with existing data elements. This may be in the form of a formal specification or a lively brainstorming session. What happens after you make a plan for it?

Collect It

This is the part of the lifecycle where you check to see if you were right and implement the plan. It's a time of testing and finally deployment of the chosen method of data creation. The data must be sufficiently understood to at least draft a solution to the data requirements, in order to be able to verify that the plan is going to work.

We start a collection of Credit Card Transaction Chargeback Amounts and see if we learned enough to manage them properly. Sometimes this means that we end up discovering we need more data elements or that we didn't plan for this one properly. This testing happens every time a new value (a real instance of the data element) comes into our hands to manage. We may find that things run smoothly for a period of time and then changes in the billing style of the Credit Card companies impact our understanding, use, and definition of this data element. When that happens the plan for the data element changes too. None of these stages in the life cycle have to be sequential. There are innumerable times when loopbacks might happen.

So now we feel comfortable that we have the process of collection of Credit Card Transaction Chargeback Amounts right. What next?

Store It

After collecting a certain amount of data, you must decide how to store it. Often the collection and storage requirements of the data differ, as we may want a narrow, fast database to collect real-time data, but a larger, less responsive one will be used for the storage. It may be collected in dozens of little individual credit card scanners, which need to be polled, have the data element applied correctly to the sales data from the cash registers, and finally aggregated together and sent on to a different environment. There may be security or accessibility issues that require parallel storage (in case of power or equipment failure).

This is also another point at which general monitoring of the process we created to collect this data element is critical. Time and quality performance are watched in order to increase both, and so satisfy the client's data needs. Credit Card Transaction Chargeback Amount may be important to us, but not important enough to cause a 20% increase in the complexity of completing a sale (causing incorrect billing and increased voids), or the negative impact on the speed of service time for completion of a sale. Many considerations happen when we look at how we are going to save and store our new data element.

But once we have collected and stored it to our satisfaction, what happens next?

Combine It

We start to use it to our advantage. This is a maturing of the data element itself. After it has been around a while, new ways to use it should start to spring up, especially if it is timely and reliable. Equally, any problems with the original analysis will have become apparent, and need revising. This is a time of integration with other data collections and highly complicated report writing. It means that the data element was well created and valuable to the data clients.

The Credit Card Transaction Chargeback Amount becomes a new variable, additive fact, or decision point in the Sales, Marketing, Finance, and Accounting departments. The data element becomes a popular new addition to the data element resource pool for the enterprise. It is referred to in the formula for Gross Sales and Net Sales. The data warehousing team is asked to provide it in the Monthly Sales Fact table. It becomes a part of the never-ending thirst for better understanding of the health of the enterprise.

So after it is aggregated, averaged, sliced, diced, racked, and stacked in reports and on screens all over the company, what can it do?

Act On It

It can become the foundation for action. It can even become the foundation for a new series of data elements that are used like a dashboard to effect decisions. This stage is only achieved if the data is understood and becomes a cornerstone of the data collection for the client. This is the crowning achievement for a piece of data. It has been proven to be a quality fact and can be relied on to support a decision. This data should have an auditable pedigree, in order to stand the scrutiny of doubt if the action doesn't get the expected results. In other words, you'd better know where it came from, and who thought it was a true fact.

This is, of course, the targeted goal of data, namely to be useful.

Credit Card Transaction Chargeback Amount can become the basis for a large decision, like a new program to provide a bank debit function for customers, which could be less expensive to the company than the charges from other credit cards. It can become the basis for daily small decisions, like offering discounts for cash sales of less than a certain value. It may even become the basis for a decision to cancel the credit card program and become the reason for its own removal from the data element inventory.

However, once the data element has established itself as a useful addition to the company data resource what do we do with it?

Archive It

We begin the process of archiving backup and historic copies for security and the ability to purge production data sets. Once data values have depreciated in relevance and value, they're moved to a nearby storage area. Only the data clients can determine where that boundary is. Setting it aside is fraught with its own challenges. How should the data be archived? How accessible should it be? How safe should the archive be? Should it be in the same format, structure, and rules as the 'live' data? The Credit Card Transaction Chargeback Amounts from the current week are critical to have available, the ones from last month are less so. The individual values from two years ago have almost lost their value. They may only have value now in aggregated form for trend reporting.

We probably begin to take slices of data out of the production systems and archive them in several ways. The data warehouse may keep a larger set of older data than any other application in the company, but even they may need to resort to removing transaction-level detail into near-line storage to keep performance at an appropriate level.

This is the golden retirement stage of data. It can still be useful but it is no longer of immediate relevance. What happens when the data values or data elements finally become obsolete?

Remove It

We delete them from all live environments. They fall off everyone's list of responsibilities. They may not be deleted the way we think of a transaction deleting a record, but they are gone for all intents and purposes from anyone's ability to use them. Over time their very existence fades away. Data values and data elements get tossed out when they are no longer of value; tapes are erased, backups deleted, whole collection tools like old applications, measuring devices, and storage devices are scrapped. Even if the data still exists (say on 8-inch floppy) the means to retrieve it may no longer be available.

This is the 'Rest in Peace' stage of data. The amount of data in our landfills is probably staggering although our perception of the value has changed over time. It is hard now to conceive of any data being worthless. But I do know that I still come across bone piles of tapes, disks, and clunky old machines that contain data that exists nowhere else, so it seems appropriate to note that this stage still does exist in the life of a data element.

When credit cards no longer exist, our Credit Card Transaction Chargeback Amount will be as useful as a collection of payroll information from 1912. Only historians will be interested, and they will only be interested in samples, not all of it. But that leads us into the next step that can happen to a data element.

Need It Again

This is where the data archeologists come in. Sometimes a need shows up for old data elements, or at least for the ability to pull a restricted selection of data elements out of data no one thought anyone would need, and that was deleted.

The data miners want to get a view of data over as much as ten or twenty years time to look for relationships between outlying factors and the behavior of company data. It seems that as soon as we decide no one needs it anymore and we toss it into a deep ocean trench, someone wants it back again. It was this need that really brought data warehousing to the importance it has today. Unfortunately even a data warehouse team has to pick and choose what it keeps based on the *known* understanding of need. We still lose data from lack of having an identified customer for it.

So you could say this is a rediscovery stage of data. It is usually extremely painful, and often expensive, to try to satisfy because nobody planned for it. It is very seldom that you can find a map of where data treasure is buried. But once you do, you need to restore it.

Restore It

This is the hard part if you haven't planned for an eventuality of this nature. This is the point where someone has to figure out how to get data out of files or media where the access tools don't exist anymore. Here we try to bring it back to life. We usually have to deal with the extra burden of lost understanding about the nature and rules of the data (in other words it may have to be analyzed to be understood again) as well as media corruptions and lack of usable tools.

How a Data Model Helps

We have talked about the lifecycle of data, because helping to manage and understand the company use of data elements is what data modeling is all about. Data models can be used to help in every stage of the lifecycle of data, bearing in mind that they need to be built differently according to the support requirement of the task in mind. Throughout iterations of these steps, the model can be used to help explain our understanding of the data to the client. This works both as a reality check (do I understand your business properly), and as a prompt to show what is missing in your understanding. This is particularly important during the first few stages of the lifecycle.

- In the 'Need it' stage, when the requirements are being gathered, a high-level concept model can be built to target whereabouts in the process that specific data element is created. It captures the definitions, descriptions of 'business world' vocabulary, and interview notes about the exceptions to the basic rules.
- In the 'Plan it' stage, all the little details can be discovered and documented in the data model. You can gather enough information to build actual data structures for the design and test stages of a project. The model in this stage is an evolving blueprint of the design and definition of many aspects of a data element.
- In the 'Collect it' stage, the model can be used to scope the test and verification scenarios for data gathering trials. You can actually build small subsets of the model that ease the ability to trace 'What if?' examples through the boxes and lines, discovering timing and security issues before trying it with an application.
- In the 'Store it' stage, many data models of the various local data structures can be used as a part of the requirements document to create a separate global repository. If data needs to be relocated, the model will confirm the basic rules required for the storage system.
- In the 'Combine it' stage, a model can act like a menu noting all the ingredient options and how they can be combined. Without a model as a map of what to choose, you are totally dependent on what the person assigned to help remembers. You can create a submodel with all the necessary elements to visually document the combination.
- In the 'Act on it' stage a model can be one half of the pedigree map. It can document what application, table, and column the data was pulled from. The original creation process can be documented in the form of an External (screen field) to Internal (table column) map. The data definitions and rules can document the quality and integrity goals of the structure. Stewardship and Security measures may also be a part of the model documentation. The model can therefore be added as part of the backup documentation for audit purposes.

- In the 'Archive it' stage, the data model can show the original and archived resting spots of the data element. It can show the structure with its rules and definitions of the data elements at the time of archival for later review in the case of restoration.
- In the 'Delete it' stage, the data model can be used to document the targeted full database, selected tables or columns, or possibly locations of data that is being proposed for surgical removal. It can be used as a map or catalog of the chosen deletions in order to be able to understand what they were when they were used.
- In the 'Need it again' stage, models, which were archived along with the data, can be used to refresh everyone's memory of what was actually archived. It can assist in targeting one specific area of data need rather than having to simply bring everything back from the grave.
- In the 'Restore it' stage, the data model can be used to build new data structures for the data to return to. They may not be named exactly the way they used to be due to uniqueness conflicts. You may need to plan a whole new structure for the return engagement.

We have covered what data is, and how models can be used to manage it. Now let's put a face on the Data Modeler role.

Who Are the Data Modelers?

This can be a tough question to answer. Finding the people in your organization with data modeling skills in their toolbox can be a challenge. Different companies use different titles for this position, and frequently append this skill to the job description of different job titles.

> **Data modeling is often an additional skill set rather than a full-time job (not because there isn't enough work to keep a modeler busy, more due to the way department responsibilities have evolved).**

Defining the Role

Data modeling as a profession doesn't seem to have established its own persona like database administration or programming. I have completely given up putting 'Data Modeler' down in occupation blanks on forms. Yet it is a profession unto itself, with its own internationally recognized professional organization DAMA (Data Management Association www.dama.org). Plus the skill of data modeling is brought up over and over in database development books as the discipline of choice towards building a quality database. Data modelers can be found doing the following, with or without a title that highlights the modeling skill set:

- Documenting, designing, and improving
- Guiding and advising
- Auditing / Enforcing
- Analyzing and researching
- Communicating

Let's now use a few metaphors to understand the data modeler's role.

Photographer

Think of the data modeler as a photographer for the National Geographic magazine. Depending on the assignment, they are either taking pictures of deep space Nebulas or inner space neutrons. They take pictures documenting reality at any level of detail and give the viewer the ability to find answers. Those beautiful, colorful, simple images provide easily recognized documentation. What would we do without pictures of our As-Is (current state) life?

A photographer does a similar job to someone *documenting* data structures already in existence (so called 'As-Is' data structures). The art of photography doesn't simply reflect true life but adds a little something, helping our understanding of the subject. Photographers use their skills to highlight and emphasize aspects that show the subject from a particular viewpoint, and focus our attention on something they have found to be important.

The data modeler working to document the current state of the Enterprise data is constantly updating a library of information relating to data elements, what they are, where they are used, how they are built, what they mean, and all of the details referring to their physical nature (data type, size, precision, default values, and so on.) The level of detail varies depending on the audience and purpose of the model.

Architect

Now think of the data modeler as an architect. Architects work to understand the owner's requirements, intended use, budget, resources, and future plans for a building. An architect would develop the blueprints for a museum and research facility, a design that would differ from a bakery or auto factory, since each design has to take into account multiple parameters. Then they oversee construction from a consultant and validation point of view. They stay a part of the build team and iteratively change the design if problems come up.

A Data Structure Architect describes someone *designing* a new data model, as we would in custom development projects. These are 'To-Be' (future state) designs. Quite frequently, several are offered to the clients and reviewed to find the appropriate final choice.

The data modeler working to build for the future is trying to design the best solution to a business problem. The solution may need to focus on flexibility, security, process improvement, data quality, or all of the above. They use their models of the 'As-Is' state of the data elements to discover the shortcomings and new data elements that are now needed. They may create a brand new solution or create a delta design that is an upgrade.

Upgrade Engineer

Upgrade Engineers (say in the Auto or Aerospace industry) frequently take existing machines, tools, or systems and try to make them better. They find ways to streamline steps and lower the energy requirements. They are constantly looking for ways to reduce moving parts to simplify construction and increase maintainability. They want to increase safety and reliability and satisfy new standards. And they want to reduce failure points to the lowest possible level while supporting critical parts with the highest amount of protection. Their prototypes are usually mini 'To-Be' proposals of solutions sent to a committee for review and approval.

This metaphor describes the Enterprise focus of an *improvement* style of data modeling. Data is an asset of the company. Ask any of the Finance department if they wouldn't prefer all assets to be maintained as inexpensively for as long a life as possible. A data modeler acting as an Upgrade Engineer will generally work only with physical data models. They will probably start with an 'As-Is' issue of some kind, followed up with one or more proposed 'To-Be' solutions.

Reference Librarian

Data modelers are people who it would be appropriate to approach if you were looking for a specific type of data. They can be the Reference Librarians of corporate data, especially if there is no Meta Data Management team. At the very least they know all of their designs inside and out.

Librarians ask you all sorts of questions about what you need. They head off to the various indexes and files and pull up all sorts of interesting references and descriptions of what you are looking for along with the locators of those references. They help point you in the right direction. The Data Reference Librarian can be analytical, and frequently creates several custom sub models incorporating data elements from multiple systems and sources, to be used for *guidance*.

Business Consultant

Given a complete and detailed enough set of requirements, a data business consultant can look at the specs of a project and provide feedback as to the company's ability to meet them, especially from the relationship rules built into a database design. The data business consultant can help solve a 'Make or Buy' decision. They provide *advice*, or perhaps a complete functional review of a software tool from a content and data restriction viewpoint. Frequently this has to be done with the help of a vendor if the database design or logical model of the tool is unavailable or private.

Buildings Inspector

Another metaphor of a data modeler can be that of an Inspector. Inspectors enforce rules issued by a higher authority in regard to a process, structure, or environment. Data modelers can do the same thing in an organization. They are frequently part of a review team enforcing naming, structure, use, integration, or data element definition standards for an enterprise.

Data Buildings Inspector is one of my least favorite roles to perform. It doesn't help you make friends, but it can be as vitally important as any of the other policing and monitoring functions that are needed to keep everyone honest. However, it is a part of the world of modeling and you might find them in an *audit* function in your corporation, attempting to enforce IT development rules.

A Data Modeler Charter

- ❑ To take responsibility for the Enterprise data element inventory and structure of the past, present, and future
- ❑ To document, design, research, guide, audit, or inspect everything there is to know about Enterprise data elements
- ❑ To enable and empower efficient, safe, maintainable, reliable, quality data structure development for the creation, maintenance, and accessibility of Enterprise data assets

Mostly a data modeler just wants to make life easier for people who need to use data as a second language, so they can leave at 5pm everyday to go play baseball with their kids. If they can depend on the documentation and designs that the data modeler can provide, it cuts their research and development time down considerably.

Job Titles

It is always fun to find out what the world calls people with your skill set these days, and find out what they have bundled it up with. I did a simple search for data modeling on www.Monster.com for the US, and here is what I found for one day of listings. It has gotten so that I don't know what to call myself anymore.

- Application Developer
- *Data Administrator*
- *Data Architect*
- Data Engineer
- *Data Modeler/Analyst*
- Database Administrator
- *Information Analyst*
- Programmer/Analyst
- Software Developer
- Systems Analyst

The ones in italics are the ones I would consider the better titles for data modelers. The key point to note is that most modelers are not just one thing. They are combinations of things, and data modelers are found all over an Enterprise.

Data modeling is often combined with other skills to create composite job titles and descriptions. This can make figuring out if you should apply for a data modeling job difficult to assess.

As-Is Support

Data modeling tasks generally fall into one of two areas. They deal with what is now, or what could be in the future. We call these areas **As-Is** and **To-Be**.

As-Is tasks require discipline and orderliness. They are often repetitive, like auditing the current state of a database, and systematic, like profiling all the data values in a production system.

As-Is tasks tend to be included in several areas such as the support of Configuration Management, providing impact analysis, promoting IT Standards, providing data integrity analysis, and researching existing techniques and tools.

Support Configuration Management

Configuration management is the process of controlling what an environment is. It may be the hardware/software compliment, or the data elements in the databases loading on a piece of hardware. There are some companies who have a whole department chartered to manage the configuration of at least the production data environment, although they may be managing the tools and content of several test environments as well. Data modelers participate in controlling data environment change, often through very detailed processes called **change control**. They maintain many different kinds of documentation tailored to fit the needs of the audience. These are the models, definitions, and mappings. They build and maintain:

- Enterprise-level conceptual data documentation
- Project logical data documentation
- Maps of data (logical/physical) objects into an Enterprise data library
- Physical data object documentation
- Data element relationship rule documentation
- Risk assessments
- Documentation of data element security needs

Sometimes this documentation is kept under the same controls as production code. It may be assessed, but not updated without going through the correct procedures of Check out, Change, Test, QA Review, Approve, and Check In.

Provide Impact Analysis

The data model library and meta data repository should be the first source for noting the ripple effects to an organization when there is a change to the Enterprise data. Y2K would have been much easier for most of us if there had been a central repository to start from. The modeler can look for impacts to and from:

- Alterations to data element definition and physical details
- Alterations to existing functions changing data deliverables
- Retiring applications, procedures, interfaces, concepts
- Incorporation off new applications, procedures, interfaces, concepts

Promote IT Standards

Data modelers are card-carrying members of the data quality team. They constantly promote data and data design standards, and are frequently called upon to participate in code, procedure, and document reviews. They frequently provide their expertise in the following:

- ❑ Assist in Test plan creation to test for Quality
- ❑ Planned integrity checks for denormalizations
- ❑ Promote proper relational integrity
- ❑ Promote standardized project documentation
- ❑ Promote Names based on RIS standards and platform requirements
- ❑ Build Definitions provided by the data element owner
- ❑ Promote standard sizes and data types
- ❑ Promote standard abbreviations
- ❑ Promote security for documented sensitive and critical data objects
- ❑ Promote standard nullability options
- ❑ Promote the creation and maintenance of documentation

Provide Data Integrity Assessments

They may assess the suitability of data design to highlight necessary concern over the existing data management environment, looking for areas of improvement in:

- ❑ Supportability – is this structure appropriate for future needs?
- ❑ Quality – does the existing structure promote quality data creation and updates?
- ❑ Integrity – are there business rule checks at the database level?
- ❑ Enterprise appropriateness – is this structure suitable for our company (are we trying to force a manufacturing application to manage contract documentation?)

Research Existing Techniques and Tools

Sometimes data modelers are tasked to be way out in front of the pack in early analysis and finding hidden or currently unknown data sources. This might mean that they have to be proactive and nosy instead of just working by request. This might also mean they need access to people, applications, machines, and documentation from all levels in order to accomplish the task. They may need to be able to interview both the CEO and a local machinist, or use the HR system in a test environment, or exercise a department's PC that they have set up to behave 'just so'. The data modeler needs an entrée to quite a bit to get their job done right.

- ❑ Documenting Business Process/Scope
- ❑ Document User requirements
- ❑ Mapping Business Processes to Data objects
- ❑ Discovering Data Structures not in Enterprise controlled environment, in other words, Desktop development

To-Be Support

To-Be support is different. This looks at Enterprise data with a view to the bright future of data management. To-Be support means designing, advising, and suggesting alternatives to the As-Is environment, which hopefully is already fully documented.

To-Be tasks involve creativity and innovative problem solving. They are less serial and more iterative, requiring a level of self motivation and management.

Design New Data Structures

- ❑ Create the planned Enterprise conceptual/logical data view
- ❑ Design data element structure fixes for change requests
- ❑ Design new database architecture for custom development projects

Provide Expert Advice

We will be required to provide advice about proposed changes, and may be asked to be responsible for preventing redundancy in data, and duplication of effort when integrating new plans into the company's data environment. Any advice we offer will need to be backed up with explanations and examples of what could happen if the choice goes one way or the other. We:

- ❑ Research data element scope and provide options
- ❑ Provide recommendations about designs
- ❑ Suggest a Make or Buy decision
- ❑ Research and provide a recommendation on data manipulation or software design tools
- ❑ Suggest Risk mitigation options, especially for denormalization decisions

Provide Alternatives

We may have to build and maintain a summary of choices, recognized deficiencies, and unresolved needs.

- ❑ Plan Integration – Identify and suggest candidate data elements for integration
- ❑ Plan of Data organization – provide options in a 1/5/10 year Enterprise data management plan

Provide Expectation Assessments

We may assess the suitability of new data designs (Purchase software decision support tasks) and check database designs for:

- Risk Analysis
- Supportability
- Quality
- Integrity
- Enterprise appropriateness

Research New Techniques and Tools

The world of data management is broad and full of growth. Data modelers need to keep up with current developments to know if they are still making the right decisions and providing the best advice possible to their Enterprise. They need to attend conferences, read journals, and develop a good network of others who can share experiences. They need to pay attention to new and evolving:

- Tools – Hardware and Software
- Techniques – Modeling, Programming, Designing
- Directions of the Enterprise market
- Client Goals
- Enterprise Goals
- And whatever else might come in handy

Summary

Data modeling can be used for much more than just building a new database. It can be an integral step in the management of the lifecycle of data. The model can be there at the conception, birth, life, death, and restoration of data. It is certainly a great design technique, but it is also a great cataloging technique. It has become more prominent in recent years, thanks to the increasing volume and complexity of data stored by individuals and businesses.

We have discussed the nature of data modelers, and shown how they can be just about anyone on a development or data management team, since data modelers are very involved in all aspects of data design, build, maintenance, and rediscovery.

We also took a look at how modeling can be used in an 'As-Is' environment for:

- Configuration Management – every change is drafted, reviewed, approved, and acted on
- Impact Analysis – every change is analyzed for ripples and given a cost
- Development Standards – every structure is designed and reviewed for corporate consistency
- Quality Review – every existing data element can be assigned a steward for quality, security, criticality, sensitivity, and integrity issues
- Research of the current enterprise data environment

And how it can also be used in 'To-Be' conceptualization in order to:

- Design New Data Structures – providing new solutions to data challenges
- Provide Data design 'Make or Buy' advice – with a data model of the business rules, you can compare the data model of a proposed vendor solution to verify a good match
- Provide alternatives – helping to map directions for the future
- Assessments of what to expect – creating risk analysis, supportability, quality, and integrity due to proposals
- Research of new directions for data management