



HOW TO MODERNIZE YOUR DATABASES WITH DATA MODELING

As the volume of data within organizations continues to grow, database modernization must be considered. Database modernization isn't about adding capacity. It's about deriving greater benefits from the data being created.



Historically, legacy databases were often utilized to capture business transactions. While it's possible to perform basic reporting and analytics on the data contained within such databases, modern databases offer far greater capabilities. Modern databases can, for example, support real-time analytics from new data as it is being created. The idea of real-time analytics simply did not exist when many legacy databases were deployed.

DATA MODELING AS A TOOL FOR DEPLOYING MODERN DATABASES

One of the best tools that can help you through the database deployment process is data modeling. Danny Sandwell, solutions marketing manager at Quest, defines data modeling as “a graphical depiction of an organization's data requirements. It's about what data you need as a business, how it fits together, and the business rules behind that data.”

Data modeling is based on entity relationships, which essentially refer to the way humans think about data. Consider a simple e-commerce application on a website. There are objects such as customers and products, and each object contains attributes. A customer would have a name, address, and possibly a customer number. In addition to these objects and attributes, there are rules governing how objects interact with one another. When a customer places an order, there might be a rule limiting the total number of a particular item they can purchase in a single order. Conversely, there might be a rule offering the customer a discount if they buy in bulk.

Data modeling usually begins with the creation of conceptual or logical models. This process is

designed to establish the relationships between the various data objects, attributes, and business rules. These models initially tend to be very simplistic but are gradually built out as stakeholders further refine the project. A data model will often take into account variables like business requirements, industry best practices, design standards, and technical requirements.

Sandwell explained that “this sort of abstraction from technology allows everyone to be on an even playing field. Typically one might be intimidated because they don't know anything about Oracle, but with this type of modeling, they don't need to.”

Another type of data modeling is physical modeling, where the completed logical or conceptual model is adapted to a very specific technology. A physical model might define how the modeled database would run on Microsoft SQL Server, for example. This means taking the entity attributes and relationships defined in the previous model and turning those objects and relationships into tables, columns, and constraints, which are very natural things in the physical database world. From that design, you can generate the database.

One of the biggest advantages of using data modeling is lowering costs and reducing the potential for human error. During the modeling process, you're not coding anything; you're just creating a model and working with it, testing all the various “what-if” scenarios. This helps the organization get everything right before investing in technology writing code. Sandwell explained, “If you make mistakes with the actual database, it's very expensive to fix those mistakes. Whereas in a model, it's easy to make fixes,

and you can verify you have everything right before you begin creating the database.”

Another advantage to modeling your data before building a database is the modeling process allows you to classify your data. In doing so, you can define the data that should be considered “sensitive.” You can also assess how relevant regulatory mandates will apply to your new database.

Perhaps the greatest advantage of data modeling is that it turns the data development process into a collaborative effort. Sandwell explained that the model “becomes a place where everyone gets together and gets much smarter about the data in the organization and how to best use it for the benefit of the business.”

DATA MODELING FOR DATABASE MODERNIZATION

Given its advantages, it’s easy to see why data modeling is essential when deploying a new database workload. The modeling process can help organizations produce a better overall application while avoiding costly mistakes. It’s worth noting, however, that database modeling is not used solely for deploying new databases. It can also be an extremely useful tool for database modernization.

Database modeling can be used to reverse engineer an existing database. A modeling tool can analyze an existing database and automatically build a comprehensive model containing all of the same details you would expect to see for a newly created database.

For an organization that wants to make some changes to a database, you can reverse engineer the database to create a model, make changes to the model, and then forward engineer those changes back to the production environment. This means changes are made more quickly and efficiently than might have been possible in the past and with less risk.

As useful as database modeling might be for making changes to an existing database, it can also be used by organizations who want to switch to a completely different database platform as a part of the database modernization process.

Suppose, for example, an organization is using a relational database as a backend for an e-commerce site. As a part of its database modernization process, it decides to adopt a NoSQL database. NoSQL databases tend to be much faster than relational databases, thereby allowing for real-time data analytics. In the case of an e-commerce site, this could mean making product suggestions in real time as a customer places an item into their shopping cart.

Restructuring a relational database for use on a NoSQL platform does not have to be a tedious, manual process. It is possible to reverse engineer the existing relational database, denormalize the relational data, and then restructure the data for a NoSQL platform such as MongoDB or Couchbase. The modeling tool should be able to automatically create all of the code that’s required to generate the database in a MongoDB environment. The same basic process can also be used to modernize data warehouses and data lakes.

Simply put, data modeling is an accelerator for a tedious and time-consuming job that involves manual processes and a huge level of risk. This often stems from the fact that the person asked to modernize and redeploy the database most likely was not involved in creating it in the first place. Add to that a lack of documentation, and the job becomes even more difficult and expensive.

Of course, the biggest problem with manual database modernization is the greater possibility of making mistakes that would cause a mission-critical application to break. Such mistakes, even if short-lived, can be extremely costly to the business.

Manual database modernization can also be risky from a compliance standpoint. If sensitive data

The erwin product is a powerful tool for database modeling that offers several benefits, including:

Integration with a wide range of data systems: erwin can connect with traditional relational databases, NoSQL and NewSQL databases, modern cloud databases, and more.

Ability to publish designs in various formats: erwin can publish its designs in formats such as XML for other tools to ingest and publish its models to Git environments for developers.

Reverse engineering and comparison of data structures: erwin can reverse engineer databases and compare data structures, allowing organizations to quickly develop a project plan for standardizing and merging their data.

Ubiquity: The erwin model is ubiquitous and has been for many years, making it a popular choice for technologies that need to understand data structures.

Standardization: erwin can help organizations standardize their data, making it easier to manage while reducing waste.

is not properly identified, the data may be mishandled, resulting in a regulatory violation.

Conversely, database modeling is a managed, controlled, and relatively low-cost alternative to risky and labor-intensive manual modernization processes.

HOW QUEST CAN HELP

The erwin Data Modeler by Quest tool is a powerful tool that can help organizations with all of their database modeling and modernization needs. The erwin tool integrates with a wide range of data structures and systems, including traditional relational databases, NoSQL and NewSQL databases, modern cloud databases, and more. erwin Data Modeler can also publish its designs in various formats, including XML and messaging, for other tools to ingest. It can also publish models to Git environments for developers, such as GitHub.

The erwin tool can build models used to conceptualize and later automate the deployment of new databases. However, erwin can also reverse engineer and model existing databases, thereby simplifying the process and reducing the risk associated with making database changes.

While it's easy to think of erwin as a tool for use with popular database platforms such as Oracle or Microsoft SQL Server, erwin can also help organizations unlock data stored within popular applications, such as Salesforce or SAP. For example, Erwin makes it possible to extract Salesforce data and bring it into an organization's larger data landscape.

The erwin tool is particularly useful in scenarios such as mergers and acquisitions. In these situations, companies need to quickly understand and figure out how best to integrate unfamiliar and disparate data systems. erwin can reverse engineer these database platforms, creating models that reveal the data structures within. This allows companies to more easily develop a project plan for standardizing and merging their newly acquired data.

Learn more about erwin Data Modeler by Quest Software at www.erwin.com.

Quest