Re-Engineering Your Database Design with Oracle SQL Developer Data Modeler 3.1

Article Summary

Oracle SQL Developer Data Modeler 3.1 is a feature-rich product supporting logical, relational, physical, and multidimensional datatype and data flow models. Database design and generation can start with a blank page, and users start by building a new entity relationship diagram (ERD). In today's development environments, much development starts with an existing data model. This article involves reverse-engineering an existing data model, making changes to it, comparing and merging changes, and updating the database with those changes. Reverse engineering relational to logical model is useful when scheming a bottom-up design. This article discusses the approach offered by Oracle SQL Developer Data Modeler 3.1 for generating a relational model from a database design and validating the same.

Introduction to Modeling

Data modeling can be defined as a process used to define and analyze data requirements needed to support the business processes. The data requirements need to be within the scope of corresponding information systems in organizations. When modeling data, you can choose an approach that suits the nature of work that is done. The approaches to data modeling include the following:

- Top-Down Modeling
- Bottom-Up Modeling
- Targeted Modeling

Top-Down Modeling is the approach used for designing a new database. You gather information about business requirements and the internal environment, and proceeds to define processes, a logical model of the data, one or more relational models, and one or more physical models for each relational model. The process of gathering information can range from simple to elaborate, depending on your needs.

Bottom-Up Modeling is used for creating a database based on extracting metadata from an existing database or using the DDL code obtained from an implementation of an existing database.

Target Modeling is the suggested approach for adapting a database to new requirements.
Oracle SQL Developer Data Modeler 3.1

Oracle SQL Developer Data Modeler is a data modeling and design tool used to increase developer productivity by providing the means to develop logical, multi-dimensional, user defined data types, relational and physical data models. The tool is designed to enable forward and reverse engineering capabilities. The Data Modeler imports from and exports to a variety of sources and targets, provides a variety of formatting options and validates the models through a predefined set of design rules.

Features of Oracle SQL Developer Data Modeler 3.1

- Multi-level Data Modeling across platforms within one integrated system
- Multiple platform and database support
- Support for configurable forward and reverse engineering
- Integrated version control (Subversion) for collaborative development
- Integrated reports
- Incremental Oracle Database 11g features
- Support for multiple open designs
- Import and export packages, and functions
- Addition of custom Design Rules and transformations
- Import from CA ERwin Data Modeler Release 7
- Various additional enhancement requests and feature updates
Need for a Relational Model

The logical data model describes the data required for the business and is independent of the implementation details. This same logical data model could be used as the foundation for implementation of any type of database management system (DBMS) or even a file system.

A logical data model is a high-level representation that cannot be implemented as is. When creating this model you may not be aware of the physical and database constraints, but still they provide a conceptually “workable” solution. This is why it is important to have a validated and agreed upon logical data model before going into the physical database design. Forward engineering the logical data model creates the first view of a relational model or database design. This initial model is intended to serve as the basis for defining the physical implementation of the database.

Generating a Relational Model using Oracle SQL Developer Data Modeler 3.1

The relational model can be generated in either of the following ways using Oracle SQL Developer Data Modeler 3.1:

- Extract metadata directly from an existing database
- Import Data Definition Language (DDL) code that reflects an existing database implementation

Once the relational model is generated, you can modify the same as required. You can also check the design rules for the model. In Oracle SQL Developer Data Modeler 3.1, you can run the design rules to verify that all the objects in your model are defined completely. The Design Rules facility will not verify the business level correctness, but it may identify some issues such as entity naming discrepancies and so on.

The steps involved in generating a database design based on an existing database include:

- Create a relational model from the data dictionary by extracting the metadata directly from an existing database
• Modify the relational model depending on the business requirement. You can either
  o Update the relational model and create additional relational models or,
  o Denormalize the relational model
• Analyze the relational model by checking the design rules for the model
• Generate and Run Data Definition Language(DDL) script
• Reverse engineer the logical model from the relational model

Conclusion

Oracle SQL Developer Data Modeler 3.1 is designed for all database data modelers. The role of Oracle SQL Developer Data Modeler 3.1 is to make data modeling development tasks simpler and facilitate as a robust communication tool between developers and business users. Using SQL Developer Data Modeler 3.1 users can create, browse and edit the various types of data models such as logical, relational and multi-dimensional. The generation of Data Definition Language (DDL) scripts enhances productivity and promotes the use of standards.