
Database design & management with Visual Paradigm

Software applications are most likely to be developed with a database such that all data working with the application system can be retained, resulting in information and knowledge. Hence, database application is widely adopted by businesses of all sizes. In order to access and manipulate the relational database, a standard computer language, Structured Query Language (SQL) has to be used. The use of SQL statements was nearly the only choice when developing database application.

Taking a trading system as an example, if the end-user wants to update a Sales Order record, the system has to retrieve the corresponding record from the Sales Order table and display to the end-user. After the end-user confirms the modification of data, the system has to update the record accordingly. It is noticeable that a database application requires a lot of coding for handling SQL statements so as to access and manipulate the database.

Hence, it is inevitable that developers spend almost 50% of development time for implementing the code with SQL statement. Moreover, mapping between the persistent code and database table is maintained throughout the development life cycle. Once there is a change in the structure of a database table, SQL statements which related to the modified table have to be re-written. Developers have to keep an eye on every change in the database schema.

Visual Paradigm provides a solution to develop database application. Visual Paradigm features Object-Relational Mapping tool which provides an ease-to-use environment bridging between object model, data model and relational database. Visual Paradigm not only provides you a visual modeling for both logical data design and physical data design with ERD, but also automates the mapping between object model and data model.

Visual Paradigm generates not only Java and .NET persistent code, but also a cost-effective, reliable, scalable and high-performance object to relational mapping layer. The generated mapping layer includes the support of transaction, cache and other optimized feature. Visual Paradigm increases the productivity and significantly reduces the risk of developing the mapping layer manually.

Benefits of designing database with Visual Paradigm

Visual Paradigm provides the following key features so as to help you simplify your application development:

Persistence Made Easy

Traditionally developers spend a lot of effort in saving and loading objects between memory and database which makes the program complicated and difficult to maintain. Visual Paradigm simplifies these tasks by generating a persistence layer between object and data models.

Sophisticated Object-Relational Mapping Generator

Visual Paradigm generates object-relational mapping layer which incorporates prime features such as transaction support, pluggable cache layer, connection pool and customizable SQL

statement. With this mapping layer, developers can keep away from mundane implementation work and focus on the business requirements.

Model Driven Development

Visual Paradigm provides a true model driven platform for application development. Visual Paradigm allows developer not only to start from creating the models by using [UML Class Diagram](#) or Entity Relationship Diagram tools ([ERD tools](#)) to generating the executable persistence layer from the models, but also to modify the entity-relational model which comes from reverse engineering an existing database, transform into object model and generate persistence layer. With the sophisticated model-code generator, the persistent model will be updated automatically according to any modification.

Extensive Database Coverage

Visual Paradigm supports a wide range of database, including Oracle, DB2, Cloudscape/Derby, Sybase Adaptive Server Enterprise, Sybase SQL Anywhere, Microsoft SQL Server, PostgreSQL, MySQL and more. Visual Paradigm also promotes an easy migration between databases by enabling the same set of ORM Java objects to work with different databases and transforms the proprietary data type that suit the default database specified.

Database Reverse Engineering

Visual Paradigm allows you to reverse engineering an existing database through JDBC into the entity-relational model. Developers can transform the entity-relational model to object model and redesign the database for further development

Class Reverse Engineering

Visual Paradigm allows you to reverse engineering Java classes and Hibernate models into the persistent object model. Developers can transform the persistent object model to data model and redesign the models for further development.

IDE Integration

Visual Paradigm is not only a standalone application, but also integrated to the major Integrated Development Environments (IDEs), including Eclipse, Visual Studio, NetBeans, IntelliJ IDEA and Android Studio, which results in streamlining the entire model code- deploy software development process.