
What is Object Relational Mapping (ORM)?

In programming, Java developers used to take the JDBC approach for data persistence, which involves writing and executing SQL statements in retrieving/updating data in database. This approach was popular, but probably not a good one because SQL can be difficult to read and write, error prone and hard to debug.

To make programming easier and reduce the chance of making mistakes, many developers prefer not to execute SQL statements directly, but to build an object model that reflects the data structure. In runtime, data will be retrieved from database and filled into the object model. Developers can then work entirely with objects, without writing any SQL statements. The technique to convert data between object model and relational database is known as object-relational mapping (ORM, O/RM and O/R mapping).

Hibernate

A well-built object model can be very useful both in programming and debugging. However, to build an object model is not that easy. While programming language like Java is object-oriented, which represents data as interconnected graph of objects, relational database, on the contrary, represents data in tabular format (like spreadsheet). You need to tackle the mismatch between object model and relational model if you decide to write such a tier, not to mention the amount of time and work required in programming and debug the model, which is always propotional to the scale of your system. All these make data persistence difficult.

There are both free and commercial packages you can use to perform object-relational mapping. [Hibernate](#) is one of the most popular packages. Visual Paradigm supports the generation of Hibernate ORM from your database design ([ERD](#)). Visual Paradigm can also synchronize your database design to object model (Class Diagram). You can then generate Java objects to use in your program.