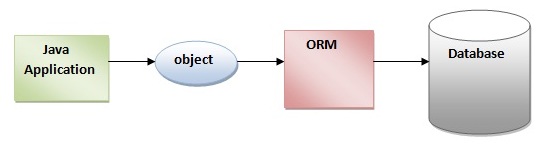
What is ORM



Java is an object oriented programming language which is represented as a graph of objects whereas relational database is represented in a tabular format using rows and columns. So, when it comes to save an object in a database, there exist some obvious mismatch and to handle this mismatch we have Object Relational Mapping(ORM). Hence, ORM can be defined as a framework used to map an object to a relational database.

As we know, in any relational database, the top level element is table and each table is divided into rows and columns. A column contains values of a particular type and a row contains one set of data for a paticular table. For example we can say, if the table name is Employees, then name, age, address can be its different columns whereas the complete info of one single employee is maintained in a single row.

In java, the top level element is an object of a class. A class has its name similar to table name, it has some attributes defined(private or public) similar to column names in a table.

We can conclude that one plain object is equivalent to a single row in a relational database table and these are the main parameters that maps an object in java to a table in a database.

**Why ORM is Required**

As we know, java provides an API called Java Database Connectivity (JDBC) to access database. It provides ways to query a relational database. You write some native sql queries and and ask JDBC to execute those queries and underlying driver class returns you the resultset after executing it. This resultset contains multiple rows and we extract each columns of the rows using methods available based on the type of data whether its a String or Integer. Doing so a developer has to keep in mind the database table relations to write queries, the exact column names and the datatypes.

Whereas while dealing with the same data in Java, the scenario completely changes. You will encounter abstraction, inheritance, composition, identity and many more and converting the table format data to an object often leads to mismatch. The solution to this mismatch is using ORM.

As we know, there are different types of database. Each database have some of its custom functions and datatypes defined and while using JDBC we also need to care of those functions and datatypes.

.