Enterprise Development Glossary

1. Database Layer: A software abstract layer that is responsible for all the physical operations related to storing and retrieving data onto a file data system. Some implementation examples of this layer are SQL Server, MySQL, Oracle database.... In other word, this layer is a database management system that support a set of API that is normally available to build program on top to talk to database SQL is the main language that is used in this layer.

2. Data Access Layer (DAL): This is a collection of Data Access Objects that are working for the purpose of ORM. In 3-tier architecture software, this layer will encapsulate the data access behaviors from the Business Logic Layer so that this layer do not need to know which exactly what kind of data store it is working with. Most often, this layer is built on top of Database Layer to simplify data access operations to a higher abstraction level. This promote maintainability and adaptability quality of software.

3. Data Access Object (DAO): Live inside a data access layer, each Data Access Object provides functions and methods to interact with a particular data store or table such as CRUD operation. It could also working with other data storage system such as file system or data comes from network interface.

4. Data Transfer Object (DTO): Basically, this is POCO (Plain Old Clr Object) which is using to transfer data between software layers. For example, data records after retrieving from a DBMS in Database Layer will be transform into a collection of DTOs and send back to Business Logic Layer. A similar concept is used across different programming language. For example in Java, they call this POJO (Plain Old Java Object) or POPO in Php.

5. ADO.NET. Implementation of database layers support in .Net. This technology is provided as a set of API and data structure that developers can use to build a DAO. It provides API to connect to a data source such as ODBC, MSSQL, MySql, Excel Spreadsheet, Access… and return DataTable or DataSet objects that can be used in OOP software. In Java, an equivalent to ADO.NET is JDBC.

6. Object Relational Mapping (ORM). The data structure of Database Layer is table whereas programming language can only understand objects. Therefore ORM comes into play as a mapping between table and object. This will abstract away the complexity of addressing table column directly and instead, using object properties to interact with data in the database. The ORM also takes care of generating SQL query to access or manipulate data, as well as database connection and access transaction. Some ORMs have more advanced features such as: lazy loading, caching, automatic generating entity from database schema and so on… Some of the most famous ORM are Hibernate (for Java)/NHibernate (for .Net), Entity Framework (.NET), doctrine (PHP).

7. Entity Object and Entity Framework (EF). And ORM implementation in .NET, developed and supported by Microsoft. From EF version 6.0, Entity Framework is available as an open source project and licensed using Apache License v2. Entity Object is the mapped object of a data table in EF. This object contain almost only properties and no behavior, and it can be retrieved from a DbSet property of a DbContext object. In other word, Entity Object can be equivalent to a row in data table, DbSet is a representation of a table in a database, and DbContext could be considered as a set of tables.