ORM - Object Relational Mapping
(MapeoObjetoRelacional)

AUTORES:
Alvial Cid Raquel (ralvial@mtt.cl)
Saavedra Quevedo Bernardita (bsaavedra@asrm.cl)
Valenzuela Parada Victor (vitoco67@gmail.com)

RESUMEN

El ORM (Mapeo Objeto relacional) es la técnica de programación que nos permite transformar los datos entre el sistema de tipos utilizados en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional, utilizando un motor de persistencia. Lo que da como resultado una base de datos orientada a objetos virtual sobre la base de datos relacional. Esta transformación permite el uso de las bondades de la programación orientada a objetos.

Los ORM nacieron a mediados de la década del 90, se hicieron masivos a partir de la masificación de Java. Por esta razón, los frameworksmás populares hoy en día en .NET son adaptaciones del modelo pensado en Java.

Uno de los componentes de OMR más utilizado, por no decir el más utilizado es HIBERNATE, surgido del ambiente JAVA y llevado al uso del framework.NET con la versión NHIBERNATE.

INTRODUCCION

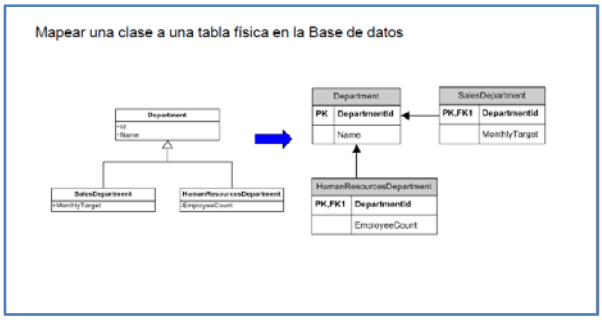
El mapeo objeto-relacional es una técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional. En la práctica esto crea una base de datos orientada a objetos virtual, por sobre la base de datos relacional. Esto posibilita el uso de las características propias de la orientación a objetos (básicamente herencia y polimorfismo).

DESARROLLO DEL TEMA

El ORM (ObjectRelationalmapping) es una técnica de programación para relacionar datos entre un lenguaje de programación orientado a objetos y el sistema de bases de datos relacional utilizado en el desarrollo de una aplicación.

El principal problema que surge hoy en día, prácticamente que todas las aplicaciones están diseñadas para usar la programación orientada a objeto (POO) mientras que las bases de datos más usadas son del tipo relacional. Las bases de datos relacionales solo permiten guardar tipos de datos primitivos (enteros, cadenas de texto, etc) por lo que no se pueden guardar de forma directa los objetos de la aplicación en las tablas, si no, que estos se deben de convertir antes en registros, (por lo general afectan a varias tablas) En el momento de volver a recuperar los datos hay que hacer el proceso contrario, se debe convertir los registros en objetos.

Es entonces cuanto ORM cobra importancia ya que se encarga de convertir en forma automática los objetos en registros y viceversa simulando así tener una base de datos orientada a objetos.



Persistencia:

Persistencia, es la habilidad que tiene un objeto de sobrevivir al ciclo de vida del proceso en el que reside. Los objetos que mueren al final de un proceso se llaman transitorios.
Si queremos cargar un objeto persistido en memoria, los pasos a seguir serían:

1. Abrir la conexión a la base de datos, crear una sentencia SQL parametrizada
2. Llenar los parámetros (por ejemplo la primary Key)
3. Ejecutarlo como una transacción
4. Cerrar la conexión a la base de datos.

Con un framework, la tarea se reduce a:

1. Abrir una sesión de la base de datos
2. Especificar el tipo de objetos que queremos (primarykay)
3. Cerrar sesión

No obstante,el framework debería resolver los siguientes puntos:

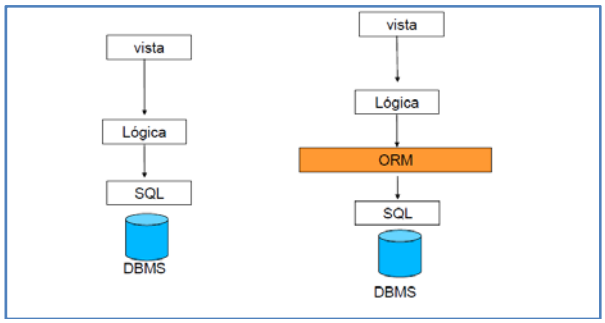
- ✓ Transacciones. ¿Se hacen todos los cambios sin excepción, o, se hace nada?
- ✓ Caché. ¿Se almacenan en memoria los objetos usados? ¿Se almacenan las consultas SQL hechas?
- ✓ Carga retardada. Cuando se carga el objeto en memoria, ¿se cargan en memoria todas sus relaciones? ¿se cargan en memoria sus campos menos usados (por ejemplo los BLOGS de gran tamaño)?
- ✓ Referencia circular. Si el objeto está relacionado 2 veces con el mismo objeto, ¿se carga 2 veces al objeto relacionado?
- ✓ OID. ¿Asignamos manualmente o automáticamente las claves primarias?

Estudios muestran que aproximadamente el 35% de desarrollo de software está dedicado al mapeo entre el modelo de objetos y su correspondiente modelo relacional. Ahora imaginemos, que necesitamos actualizar nuestro software. Como sabemos al ser objetos, lo que proveemos al cliente son las interfaces de acceso.

Ventajas y desventajas de utilizar ORM:

- ✓ Rapidez en el desarrollo. La mayoría de las herramientas actuales permiten la creación del modelo por medio del esquema de la base de datos, leyendo el esquema, nos crea el modelo adecuado.Esto quita mucho tiempo de “coding” repetitivo y que, seguro, el sistema hace mejor que nosotros, ya que los humanos siempre cometemos fallos tontos.
- ✓ Abstracción de la base de datos. Al utilizar un sistema ORM, lo que conseguidos es separarnos totalmente del sistema de Base de datos que utilicemos, y así si en un futuro debemos de cambiar de motor de bases de datos, tendremos la seguridad de que este cambio no nos afectará a nuestro sistema, siendo el cambio más sencillo.

- ✓ Reutilización. Nos permite utilizar los métodos de un objeto de datos desde distintas zonas de la aplicación, incluso desde aplicaciones distintas.
- ✓ Seguridad. Los ORM suelen implementar sistemas para evitar tipos de ataques como pueden ser los SQL injections.
- ✓ Mantenimiento del código. Nos facilita el mantenimiento del código debido a la correcta ordenación de la capa de datos, haciendo que el mantenimiento del código sea mucho mas sencillo.
- ✓ Lenguaje propio para realizar las consultas. Estos sistemas de mapeo traen su propio lenguaje para hacer las consultas, lo que hace que los usuarios dejen de utilizar la sentencias SQL para que pasen a utilizar el lenguaje propio de cada herramienta.



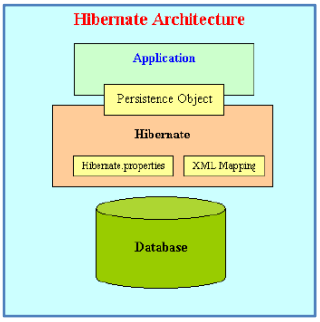
Desventajas

- ✓ Tiempo utilizado en el aprendizaje. Este tipo de herramientas suelen ser complejas por lo que su correcta utilización lleva un tiempo que hay que emplear en ver el funcionamiento correcto y ver todo el partido que se le puede sacar.
- ✓ Aplicaciones algo más lentas. Esto es debido a que todas las consultas que se hagan sobre la base de datos, el sistema primero deberá de transformarlas al lenguaje propio de la herramienta, luego leer los registros y por último crear los objetos.

NHibernate:

En la actualidad hay muchos tipos de framework que nos devuelven el mapeo objeto-relacional, según el lenguaje que estemos utilizando (Doctrine: para PHP 5.2 y posterior, Propel: para PHP 5 y superior, Linq: desarrollado por Microsoft).Entre estos frameworksORM el más utilizado es NHIbernate. Esto se debe principalmente al poderoso lenguaje de consulta que posee: HQL (HibernateQueryLanguage). Además de se un software de código abierto, posee las siguientes características:

- ✓ Herramienta de Mapeo objeto-relacional para la plataforma Java
- ✓ También disponible para.net (Nhibernate).
- ✓ Funciona mediante archivos declarativos (XML) que permiten establecer estas relaciones.
- ✓ Facilita el almacenamiento y consulta de objetos en una base de datos relacional.



Al momento de elegir un framework ORM, se deben tener presente los siguientes aspectos:

Referencia circular: Se refiere a si el framework es capaz de detectar cual es el objeto que se está solicitando, sin hacer el roundtrip a la base de datos.

Información oculta: (Shadow information). Así como el OID hay muchas columnas de la tabla que no necesitan ser mapeadas a una propiedad del objeto. Estas columnas contienen información oculta para el modelo de onbejos pero necesaria para el modelo relacional. En esta categoría entran el mecanismo de concurrencia: Estampa de tiempo y versión de objeto. Al leer el objeto, se lee esta información que es ocultada al objeto pero mantenida por el framework.

Lenguaje de consulta (OQL- ObjectQueryLanguage) La obtención de varios objetos a través de un lenguaje especial es una de las características más apreciadas. Por ejemplo, obtener todos los Juan Perez de una base de datos PERSONAS.

“SELECT Persona FROM Personas WHERE Nombre=”Juan”.

NHibernate con HQL es uno de los mejores.

Actualización en cascada:
La posibilidad de que modificaciones hechas a un objeto repliquen en los objetos relacionados.

Operaciones en masa: Habrá veces que por razones de perfomance, se querrá hacer una operación en masa. Por ejemplo, actualizar todos los objetos con nombre Juan a J. De la manera tradicional, deberíamos leer cada objeto, modificarlos en memoria y recién allí guardarlos de nuevo.

CONCLUSIONES Y BIBLIOGRAFIA

El uso de la herramienta ORM, permite la correspondencia lógica y natural entre modelo relacional a modelo de objetos.
Optimizando recursos de tiempo y rapidez, adicionalmente minimizando los errores debido a que es un proceso automático donde la intervención del programador es menor.

http://www.programacion.com/articulo/persistencia_de_objetos_java:_el_camino_hacia_hibernate_251
http://es.wikipedia.org/wiki/Mapeo_objeto-relacional
http://www.programacion.com/articulo/conceptos_basicos_de_orm_object_relational_mapping_349