

1.2 Hibernate

Trabalhar com softwares orientados a objetos e banco de dados relacionais pode ser enfadonho e consumir muito tempo de desenvolvimento. Quando a linguagem de programação Java começou a tomar força no mercado, várias soluções começaram a surgir para resolver este problema. A vencedora dessas soluções, que desbancou até mesmo o padrão EJB 2.0, foi o Hibernate, uma ferramenta para mapeamento objeto/relacional para ambientes Java.

O Hibernate hoje é o framework Java para mapeamento objeto relacional mais conhecido no mercado. Sua principal função é abstrair o mapeamento, economizando esforço e preocupações concernentes a tal tarefa. Com uma arquitetura simples, de fácil configuração, e com funções de fácil entendimento, o Hibernate simplifica bastante a tarefa do desenvolvedor.

O Hibernate implementa a especificação JPA (Java Persistence API) através do conceito de anotações (implementada a partir do JDK5), o que facilita ainda mais o mapeamento objeto-relacional, que pode agora ser feito diretamente na classe.

O termo mapeamento objeto/relacional (ORM) refere-se à técnica de mapeamento de uma representação de dados em um modelo de objetos para um modelo de dados relacional baseado em um esquema E/R. O Hibernate não cuida somente do mapeamento das classes Java para tabelas do banco de dados (e dos tipos de dados Java para os tipos de dados SQL), mas também provê facilidades para consultar e retornar os dados da consulta, e pode reduzir significativamente o tempo de desenvolvimento em contrapartida ao alto tempo gasto pelas operações manuais dos dados feitas com SQL e JDBC.

O Hibernate é uma ferramenta de consulta e persistência objeto/relacional de alta performance. Uma das soluções ORM mais flexíveis e poderosas no mercado, ele faz o mapeamento de classes Java para tabelas de banco de dados e de tipos de dados Java para tipos de dados SQL. Ele fornece consultas e facilidades para retorno dos dados que reduzem significativamente o tempo de desenvolvimento.

1.2.1 Hibernate Annotations

O Hibernate, como toda ferramenta para mapeamento objeto/relacional, requer metadados que governem as transformações dos dados de uma representação para outra (e vice-versa).

Annotations podem ser divididas em duas categorias:

- **anotações de mapeamento lógico** – permitem descrever o modelo de objetos, a associação entre classes, etc.;
- **anotações de mapeamento físico** – descreve o esquema físico, tabelas, colunas, índices, etc.

1.2.2 Mapeando Classes e Atributos com Anotações

Toda classe Java que será persistida em um banco de dados através do Hibernate é considerada uma entidade e é declarada utilizando a anotação `@Entity` (acima da declaração da classe).

A anotação `@Id` permite que o usuário defina qual propriedade é a chave primária da sua entidade. A propriedade pode ser preenchida pela própria aplicação ou ser gerada pelo Hibernate (recomendado). É possível definir a estratégia para geração graças à anotação `@GeneratedValue`.

A anotação `@Table` é definida em nível de classe e permite que você descreva o nome da tabela que será utilizada para o mapeamento da entidade. Se nenhum `@Table` for declarado, os valores padrões serão utilizados: no caso, o próprio nome da classe.

É possível mapear as colunas de uma tabela com a anotação `@Column`. Utilize isso para sobrescrever o valor padrão, no caso o nome do próprio atributo. O código abaixo apresenta um exemplo.

Todas essas anotações e muitas outras veremos em nossos práticos que veremos em sala de aula.