

Pemrograman Framework Java

Pertemuan 6



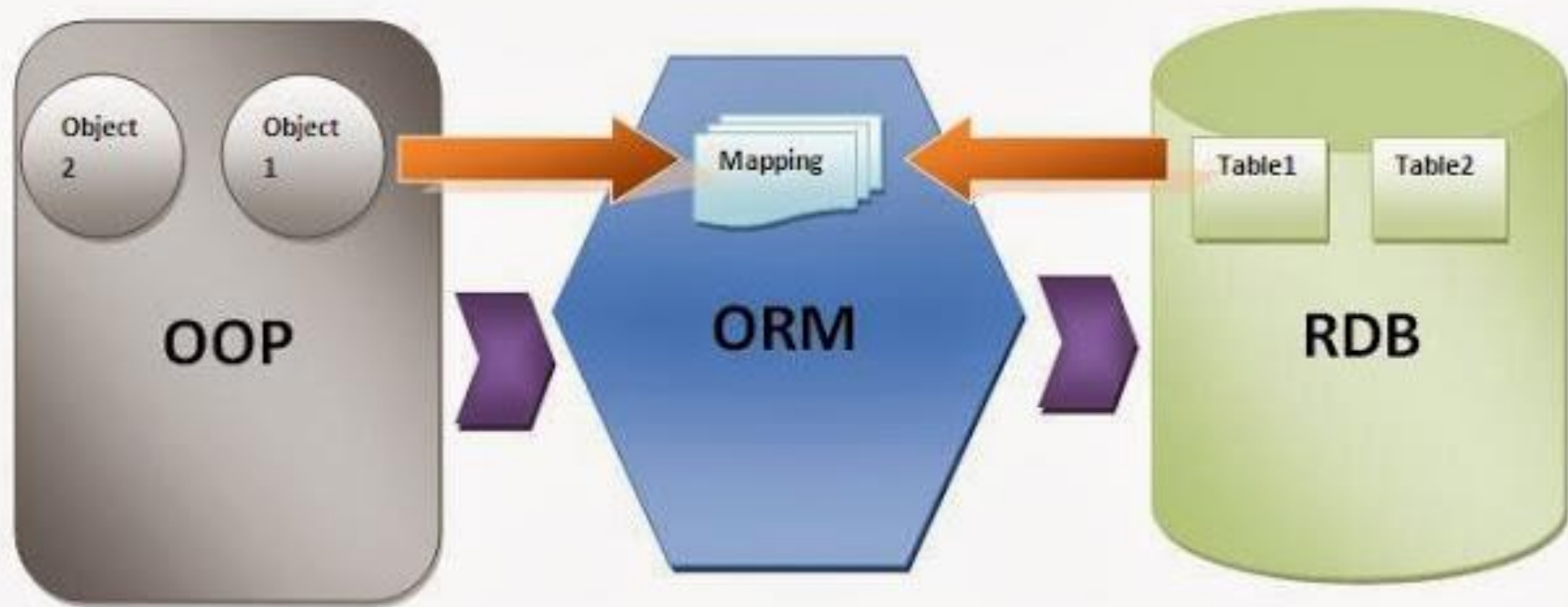
Object-Relational Mapping

- Pemetaan objek-relasional (ORM, O / RM, dan alat pemetaan O / R) dalam ilmu komputer adalah teknik pemrograman untuk mengkonversi data antara sistem tipe yang tidak kompatibel menggunakan bahasa pemrograman berorientasi objek.



Cont'd

- Dibandingkan dengan teknik pertukaran tradisional antara bahasa berorientasi objek dan database relasional, ORM mengurangi jumlah kode yang perlu ditulis.
- Kerugian dari alat ORM umumnya berasal dari tingkat abstraksi yang tinggi mengaburkan apa yang sebenarnya terjadi dalam kode implementasi.



Object Relational Mapping



Mengenai Hibernate

- Hibernate ORM adalah alat pemetaan objek-relasional untuk bahasa pemrograman Java.
- Kerangka kerja untuk memetakan model domain berorientasi objek ke database relasional.
- Hibernate menangani masalah ketidakcocokan impedansi objek-relasional dengan mengganti akses database langsung dan persisten dengan fungsi penanganan objek tingkat tinggi.



Cont'd

- Fitur utama Hibernate adalah pemetaan dari kelas Java ke tabel database, dan pemetaan dari tipe data Java ke tipe data SQL.
- Hibernate juga menyediakan permintaan data dan fasilitas pengambilan.
- Panggilan SQL ini membebaskan pengembang dari penanganan manual dan konversi objek dari set hasil.



Cont'd

- Menggantikan tipe Default SQL Ketika pemetaan kolom ke sebuah properti
- Pemetaan Enums Java ke Kolom seolah property biasa
- Pemetaan single property ke berbagai kolom



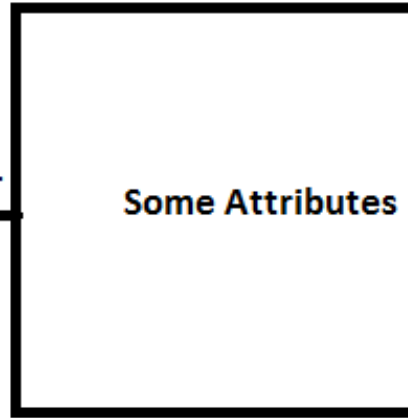
Pemetaan

- Pemetaan kelas Java ke tabel database diimplementasikan dengan konfigurasi file XML atau dengan menggunakan Java Annotations.
- Hibernate dapat menggunakan file XML atau anotasi Java untuk mempertahankan skema database.
- Ada fasilitas yang disediakan untuk mengatur hubungan satu-ke-banyak dan banyak-ke-banyak antar kelas. Selain mengelola asosiasi antara objek, Hibernate juga dapat mengelola asosiasi refleksif

Table1



Table 2



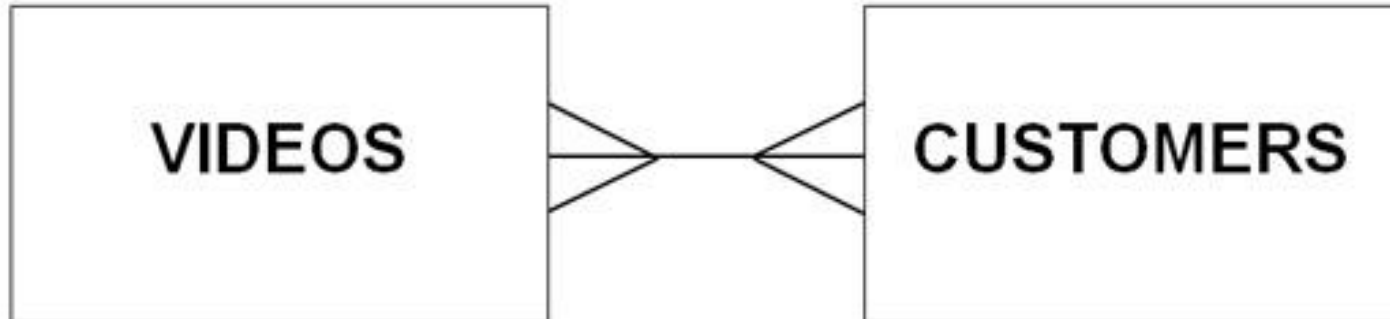
1

1



Some Attributes

Many-to-many relationships





Keuntungan Hibernate Framework

- 1) Sumber Terbuka dan Ringan
- 2) Kinerja Cepat
- 3) Query Independen Basis Data
- 4) Pembuatan Tabel Otomatis
- 5) Menyederhanakan Bergabung dengan Kompleks
- 6) Menyediakan Statistik Kueri dan Status Basis Data



Hibernate Query Language

- Hibernate menyediakan bahasa yang terinspirasi SQL yang disebut Hibernate Query Language (HQL) untuk menulis query seperti SQL terhadap objek data Hibernate.
- Kriteria Query disediakan sebagai alternatif berorientasi objek untuk HQL.



Cont'd

- Metode yang sering digunakan
 - `public int executeUpdate()` digunakan untuk menjalankan pembaruan atau menghapus kueri.
 - `public List list()` mengembalikan hasil relasi sebagai daftar.
 - `public Query setFirstResult(int rowno)` menentukan nomor baris dari mana catatan akan diambil.
 - `public Query setMaxResult(int rowno)` menetapkan no. catatan yang akan diambil dari relasi (tabel).
 - `public Query setParameter(int position, Object value)` itu menetapkan nilai ke parameter kueri gaya JDBC.
 - `public Query setParameter(String name, Object value)` itu menetapkan nilai ke parameter kueri bernama.

```
try {  
    //编写hql语句 在hql中 如果使用select *, 可以省略。  
    String hql="from User";  
    /* Query query = session.createQuery(hql);  
    List<User> users = query.list();*/  
    List<User> users=session.createQuery(hql).list();  
    for(User u:users){  
        System.out.println(u.getName()+"\t"+u.getAge());  
    }  
  
} catch (Exception e) {  
    e.printStackTrace();
```

https://blog.csdn.net/Miracle_Gaara1



Caching di Hibernate

- Caching hibernate meningkatkan kinerja aplikasi dengan menyatukan objek dalam cache.
- Berguna ketika kita harus mengambil data yang sama beberapa kali.
- Terutama ada dua jenis caching:
 - Cache Tingkat Pertama, dan
 - Cache Tingkat Kedua



First Level Cache

- Objek sesi menyimpan data cache tingkat pertama.
- Diaktifkan secara default.
- Data cache tingkat pertama tidak akan tersedia untuk seluruh aplikasi. Aplikasi dapat menggunakan banyak objek sesi.



Second Level Cache

- Objek SessionFactory menyimpan data cache level kedua.
- Data yang disimpan dalam cache level kedua akan tersedia untuk seluruh aplikasi.
- Tetapi kita perlu mengaktifkannya secara eksplisit.
 - EH (Easy Hibernate) Cache
 - Swarm Cache
 - OS Cache
 - JBoss Cache

Cont'd

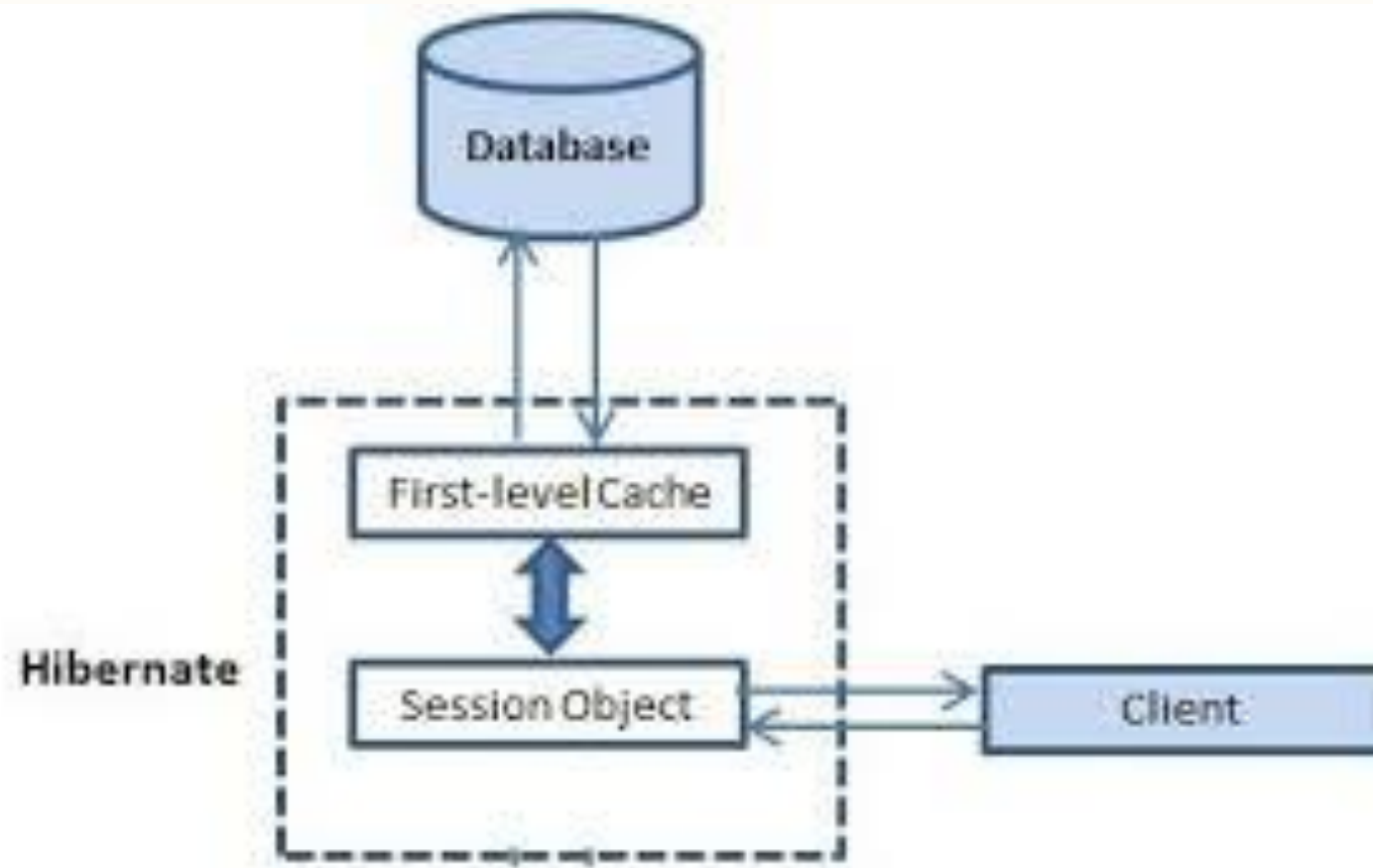


- Hibernate cache level kedua menggunakan cache umum untuk semua objek. Berguna jika Anda memiliki beberapa objek sesi
- SessionFactory menyimpan data cache level kedua. Bersifat global untuk semua objek sesi dan tidak diaktifkan secara default.
- Vendor yang berbeda telah menyediakan implementasi Cache Tingkat Kedua.

Cont'd

Implementation	read-only	nonstrict-read-write	read-write	transactional
EH Cache	Yes	Yes	Yes	No
OS Cache	Yes	Yes	Yes	No
Swarm Cache	Yes	Yes	No	No
JBoss Cache	No	No	No	Yes

Ilustrasi



Ilustrasi

