

Data Management Design

.

Persistence

- Setiap sistem informasi memerlukan persistent data yaitu data yang selalu aktif walaupun sistem sudah tidak aktif.
- Persistent data dapat dimodelkan dengan menggunakan UML.
- Persistent data dapat disimpan dalam bentuk file atau DBMS atau juga dalam bentuk objek database sehingga mempengaruhi rancangan dari manajemen data

Mekanisme Persistence

- File menyimpan data, biasanya dalam media magnetic seperti disk atau tape
- Database management systems (DBMS) terdiri atas tabel data (relational DBMS) atau objek (object DBMS)
- DBMS menggunakan file untuk menyimpan data atau objek, namun mereka menyembunyikan proses fisik dan menyimpan data pada layer abstrak
- Objek bisa serial secara langsung dengan file

Arsitektur Persistence

- Pemilihan arsitektur untuk persistence adalah masalah desain sistem
- Desain penyimpanan untuk kelas khusus dan asosiasi dalam kerangka arsitektur itu adalah masalah desain kelas
- Desain keseluruhan dapat dibatasi oleh harus beroperasi dengan sistem yang ada atau menggunakan DBMS yang ada

Petanyaan yang berkaitan dengan Persistence

- File dapat digunakan untuk beberapa penyimpanan?
- Apakah itu benar-benar sistem OO atau hanya sebuah antarmuka untuk database relasional menggunakan Java atau C++?
- Apakah sistem menggunakan DBMS yang ada?
- Apakah akan menggunakan DBMS relasional?
- Apakah akan menggunakan obyek DBMS?

Petanyaan yang berkaitan dengan Persistence

- Apa layering logis dari sistem?
- Apa layering fisik dari sistem?
- Apakah sistem terdistribusi? Apakah ini termasuk penyimpanan data terdistribusi?
- Protokol apa yang akan digunakan untuk berkomunikasi dalam sistem?

Mapping Kelas ke Tabel

- Kelas dengan struktur yang sederhana maka kelas tersebut akan menjadi sebuah tabel.
- Identifier dari objek akan menjadi primary key.
- Kelas yang berisi kelas lain sebagai atribut dipisahkan menjadi kelas yang berbeda dan identifier dari kelas yang berada dalam kelas lain akan menjadi foreign key pada kelas yang memuat kelas tersebut.

Mapping Kelas ke Tabel

- Untuk koleksi kelas maka dibuat dalam dua tabel, satu untuk objek dalam koleksi, yang lain untuk memegang Obyek ID dari obyek yang berisi dan obyek yang terkandung.
- asosiasi Satu-ke-banyak dapat diperlakukan seperti koleksi
- asosiasi Banyak-ke-banyak menjadi dua tabel terpisah untuk objek dan tabel untuk menempatkan ID Obyek pasangannya

Mapping Kelas ke Tabel

- asosiasi Satu-ke-satu diimplementasikan sebagai atribut foreign key -masing-masing kelas memperoleh atribut ekstra untuk ID Obyek yang lain
- Untuk mengimplementasikan inheritance
 - hanya mengimplemetasikan superclass sebagai tabel termasuk semua atribut subclass
 - hanya mengimplementasikan subclasses sebagai tabel, ada duplikasi superclass pada setiap atribut
 - implementasi superclass dan subclasses sebagai tabel akan menshare primary keys

Object DBMS

- ODBMS have the advantage that objects can be stored directly
- Object Data Management Group (ODMG) standard
- Not all object databases conform to the standard
- Object databases are closely linked to programming languages with ways of navigating through the database

Contoh Operasi C++ Menyimpan Objek

```
IntCampaign * CreativeStaff::findIntCampaign
                                   ( string campaignCode )
{
    IntCampaign * intCampaignPointer;
    intCampaignPointer =
    staffIntCampaignList.getValue().query_pick(
        "IntCampaign*",
        "campaignCode == this->campaignCode",
        os_database::of(this));
    return intCampaignPointer;
}
```

Referensi

- Korth and Silberschatz (2001)
- Howe (2001)
- Roos (2003)
- Alur, Crupi and Malks (2003)

(For full bibliographic details, see Bennett, McRobb and Farmer)