

ODMG

Le manifeste Bancilhon dis : « on oublie le relationnel »

Il y a 2 autres approches :

Stonebreaker : réponse à Bancilhon, les bdd du futur auront sql : évolution et intégration en douceur

L'approche la plus propre : pas relié au sql est celle du manifeste de Date : faire des domaines aux classes d'objets

A l'époque sql3 avançait très lentement (versioning rare)

Objectif : servir très vite service bd aux développeurs objet

Dualité dans l'ODMG : tout n'est pas valeur, on a des structures de données à gérer

Valeur : monde bd

Objet : monde objet

Adaptation de ODMG; on veut rajouter la persistance du monde des bd

Odl : object definition language : langage définition schéma objet

Oml : object manipulation language : équivalence sql

Oif : object interaction format : pouvoir importer des bd ou fichier venant d'un autre univers dans l'univers objet

Les autres sont des interfaces vers ces langages de programmation

Parler d'un SGBDOO doit offrir les interfaces ci-dessus

Le système objet de l'ODMG : modèle dualité objet – valeur :

Classe d'objet dans l'ODMG :

OID : chaque objet a un identifiant

Value : correspond aux tuples, la table de valeurs : ensemble des instances

de l'objet qui peuvent être nommée et stockée

Attributes / Methodes

Liens bidirectionnels binaires entre classes (ref, inverse) : correspond aux jointures : fusion des tables entre elle : rentre explicite les liens entre 2 classes

On en peut faire de jointure que si on a un suivi de pointeur

Propriété RICE dans ODMG

R : Héritage multiple

I : Oid

C : Structures

Collections

set (élément unique) ,bag (élément dupliqué, sans trie), list (trie), array, dictionary

Pointeurs bidirectionnels binaires

E : Type avec méthodes

Hiérarchie de types :

Paradigme objet – valeur

Valeur : littéral

Atomic : type de base long short ect

Collection : operateur d'ensemble : set, bag, list ect

Structured : vecteur de valeur : structured, date, time ect

Object :

Atomic : que l'utilisateur va construire

Collection : ensemble d'objet avec niveau de complexité voulu

Structured : structure temporelle date time ect

Meta-modèle ODMG

Classe : interpretation :

Key list : liste des attributs qui vont constituer la clé primaire

Extent name : nom donné à l'ensemble des n-upplet (instance de la classe)

Super classe : abréviation de l'héritage

Object : une classe est composée de un ou plusieurs objets

OID : identifiant

Has_name : on peut le nommer

Class : class auquel elle appartient

Create : crée une instance

Delete : supprimer une instance

Exist : vérifier si une instance existe

Same_has : comparer instances

Opération : une classe importe plusieurs opérations

Signature : l'ensemble des operateurs possible chacun défini avec leur syntaxe (paramètre d'entrée et de sortie)

Invoke : appeler

Return : code de retour

Return_abnormally : code de retour d'erreur

Property: une classe peut avoir plusieurs propriété

Attribute

Attr_name : nom

Attr_type : type

Set_value ; changer la valeur

Get-valeur : récupérer la valeur

Transversal path : chemin de traverse : concaténation de plusieurs liens d'une classe vers n'importe quelle autre

Path_name : nom

To_cardinality : cardinalité

To_type : type

Traverse : chemin du lien

Creator_iterator : création itérateur

OQL : prendre comme modèle sql + looksql

Attrait du relationnel : pouvoir poser des questions par un utilisateur final vers une base de donnée ODMG

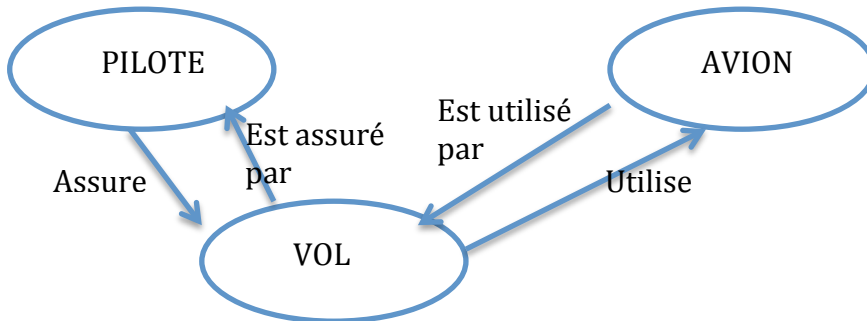
Chemin de traverse : concaténation de pointeur d'une classe A vers une classe B

Lien select / where : chemin de traverse

Mono-valué : clause select/where

Multi-valué : clause from

Faire un tableau des liens entre classes



ex OQL : Quels sont les numéros des vols assurés par un pilote de nom 'Serge' ?

Ma réponse :

```
SELECT V.VOL#  
FROM V IN VOL  
WHERE V.ESTASSUREPAR.NOM = 'Serge'
```

Réponse exacte :

```
Select v.v#  
From  
    v IN vol <IN collection>  
    p In v.est_assure_par  
Where p.Nom = 'Serge'
```

Réponse optimisée si c'est mono-valué:

```
Select v.v#  
From v IN vol  
Where v.est_assure_par.Nom = 'Serge'
```

Quels sont les noms des avions conduits par un pilote Niçois ?

<Parcours d'associations multi-valuées en utilisant des collections dépendantes>

```
Select A.nom  
From A in Avion  
    V in v.est_utilisé_dans  
    P in p.est_assuré_apar  
Where P.ville = 'Nice'
```

Q1. Nom des pilotes niçois qui assurent un vol au départ de Nice avec un avion localisé à Nice ?

Select P.nom

FROM P in Pilote

V in P.assure

A in V.utilise

Where P.ville = 'Nice' AND V.depart='Nice' AND A.ville = 'Nice'

Q2. Noms des pilotes qui habitent dans la ville de localisation d'un airbus ?

Ma réponse

Select P.nom

FROM P in Pilote

V in P.assure

A in V.utilise

WHERE A.type = 'AIRBUS'

AND A.ville = P.ville

Réponse : Impossible à faire en ODMG avec les liens existants : il faut créer les liens multi-valué habite inverse est-localisé-dans

Quantificateur universel : utilisation de l'opération algébrique 'for all'

Noms des pilotes qui conduisent tous les airbus localisés à Nice

Select p.plnom

From p in pilote

V in p.assure

For all a in v.utilise : a.avnom = 'airbus' and a.loc = 'Nice'

//!\ \ Sans for all : quantificateur existentiel,

Avec for all : quantificateur universel