



SGBD orientés objet

Concepts Généraux

1

Généralités

Modèles de données et SGBD orientés objet

- Définitions

MDOO : Un modèle de données qui capture la sémantique des objets supportée en programmation objet.

BDOO : Une collection persistante et partageable d'objets définis par un MDOO.

SGBDOO : Le gestionnaire d'une BDOO

2

Généralités Modèles de données et SGBD orientés objet

- Fonctionnalités d'un SGBDOO les plus générales

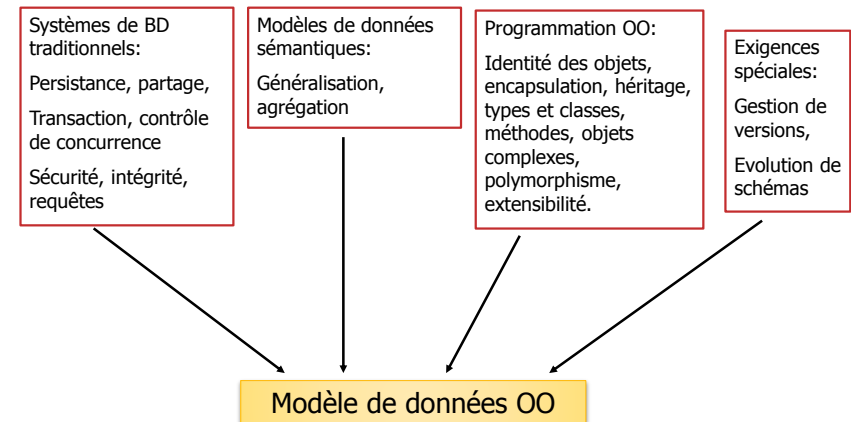
A partir de « **orientation objet = types de données abstraits + identité d'objet + héritage** » alors:

- Fonctionnalités d'un SGBD classique **plus**,
- Il supporte la notion d'identité des objets,
- Il assure l'encapsulation,
- Il gère les objets complexes

Même si l'héritage est très utile, il n'est pas essentiel à la définition d'un SGBDOO

3

Généralités : origine du modèle OO



4

Manifeste du système de BD OO
13 règles [Atkinson 1989]

1. Il gère des objets complexes: objets de base + constructeurs:
 - Ensemble minimal de constructeurs:
 - Ensemble (SET)
 - L'occurrence (TUPLE)
 - La liste (LIST ou ARRAY)
2. Il gère l'identité d'objet:
 - Tout objet doit posséder une identité unique, indépendante des valeurs d'un attribut.

5

Manifeste du système de BD OO

3. Il permet l'encapsulation :
 - Accès aux méthodes qu'à travers l'interface de l'objet.
 - Données et implémentation des méthodes sont masqués dans l'objet.
4. Il gère les types ou les classes :
 - Le schéma de BD comporte un ensemble de classes ou de types.

6

Manifeste du système de BD OO

5. Les types ou les classes doivent pouvoir hériter de leurs ancêtres:

- Un sous-type ou une sous-classe doit hériter des attributs et des méthodes, respectivement, de son super type ou de sa superclasse.

6. Il doit assurer les liens dynamiques :

- Les méthodes s'appliquent à des objets de types différents. A chaque type un code différent, donc la liaison s'effectue à l'exécution: liaison dynamique

7

Manifeste du système de BD OO

7. Le LMD est complet au niveau calcul :

- Le LMD du SGBDOO doit être un langage de programmation à vocation générale et non pas seulement un langage de requêtes.

8. L'ensemble des types de données est extensible :

- L'utilisateur doit pouvoir construire de nouveaux types à partir des types prédéfinis par le système.

8

Manifeste du système de BD OO: règles des SGBD traditionnels

9. Le système permet la persistance des données :

- Comme dans un SGBD traditionnel, les données doivent continuer d'exister donc persister après la fin de l'application qui les a créées.

10. Le SGBD doit être capable de gérer de très vastes bases de données:

- Les données sont stockées en mémoire secondaire, moyennant l'utilisation de méthodes d'accès et de stockage sophistiquées. Le SGBDOO doit disposer de mécanismes semblables.

11. Le SGBD accepte des utilisateurs simultanés et concurrents :

- Existence de mécanismes de contrôle de concurrence semblables à ceux des SGBD classiques.

9

Manifeste du système de BD OO

12. Le SGBD assure la reprise en cas de défaillances matérielles et/ou logicielles:

- Existence de mécanismes de récupération semblables à ceux des SGBD traditionnels.

13. Le SGBD fournit un moyen simple pour interroger les données :

- Existence d'un utilitaire de requêtes de haut niveau (déclaratif), efficace, indépendant de l'application ; (langage ou navigateur graphique)

10

Approches de développement de SGBDOO [Khoshafian 1990]

1. **Étendre un langage de programmation OO existant avec des fonctionnalités de BD** : par exemple, Smalltalk, C++, Java; (GemStone):
2. **Proposer des bibliothèques de SGBD Orientées Objet**: Ajouter des fonctionnalités à un langage de POO existant mais à travers des bibliothèques de classes acceptant la persistance, l'agrégation, les transactions, la sécurité etc. par exemple Ontos, ObjectStore.
3. **Intégrer des constructions objet du langage de BD dans un langage hôte conventionnel**. Par exemple extensions intégrées au langage C: système O2

12

Approches de développement de SGBDOO [Khoshafian 1990]

1. **Étendre le langage de BD existant avec des caractéristiques OO. Étendre SQL par de constructions OO** (norme SQL 1999). L'object Data Management Group (ODMG) spécifie une norme Object SQL. Nombre d'éditeurs de SGBDOO se mettent en conformité avec cette norme.
2. **Développer un nouveau modèle et un nouveau langage avec des caractéristiques totalement orientées objet**. Approche suivie par Semantic Information Manager (SIM) : modèle sémantique plus un LDD et un LMD nouveaux. [Jagannathan 1988].

13

Caractéristiques des SGBDOO

- Les SGBD de manière générale sont concernés par la création et l'entretien de vastes collections de données: ils se caractérisent par :
 - Le modèle de données
 - La persistance de données
 - Le partage des données
 - La fiabilité
 - L'évolutivité
 - La sécurité et l'intégrité
 - La distribution

Les langages de programmation traditionnels ne permettent pas la prise en compte de ces caractéristiques. Aujourd'hui les applications exigent à la fois des fonctionnalités de langage de programmation classique et de SGBD; des efforts sont consentis pour développer des systèmes tenant compte de ces deux domaines.

14

Caractéristiques des SGBDOO

- Les SGBD classiques utilisent un modèle de stockage à deux niveaux :
 - Le modèle de stockage de l'application en mémoire principale
 - Le modèle de stockage des données en mémoire secondaire.
- Les SGBDOO utilise un modèle de stockage à un seul niveau:

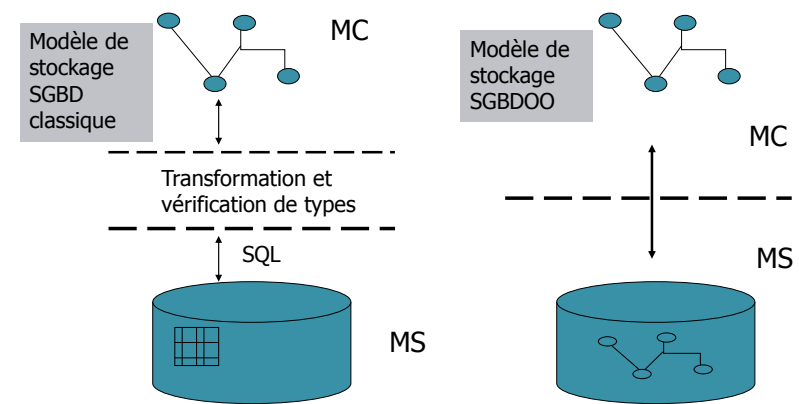
15

Quelques SGBDOO

- Versant Object Database pour C++ et Java ;
- Fast Objects .NET
- ObjectStore
- Objectivity/DB
- O2 (par universitaires français, n'existe plus)
- db4o (acheté par Versant fin 2008)

16

Caractéristiques des SGBDOO



17

Caractéristiques des SGBDOO

- **1-Gestion des OID : 2 types**
 - OID logique : indépendants de l'emplacement physique de l'objet sur MS: donc indirection nécessaire pour connaître l'adresse physique.
 - OID physique qui codifie l'emplacement.
 - Dans les deux cas, pour atteindre des performances raisonnables et optimiser les accès aux objets, le SGBDOO doit convertir les OID en pointeurs en mémoire et vice-versa.

18

Techniques de conversion de pointeurs

- Action qui consiste à convertir les OID en pointeurs en mémoire principale et vice-versa
- Utilisation d'une table de correspondance appelée « table d'objets résidents »

19

Table d'objets résidents référençant 4 objets présents en mémoire centrale

Agence :OID1

Numfi : F003
 Adressefi :
 Employé : (OID4, OID5)
 Propriété : (OID2, OID3)
 Gérant : OID6

Employé : OID4

Numemp : EMP14
 Prénom : Mohamed
 Nom : Rami
 ;;
 Agence : OID1

Table d'objets résidents

OID	Adresse en mémoire
1	
2	
3	
4	
..	

PropriétéAlouer

Numprop : LM1
 Adresse
 Type: appartement
 Employe : OID4
 Agence: OID1

PropriétéAlouer

Numprop : LM3
 Adresse
 Type: appartement
 Employe : OID4
 Agence: OID1

Si nous voulons accéder depuis agence (OID1) à employé OID5, une recherche dans la table indique qu'il n'est pas résident en MC. Il faut donc accéder à l'objet en MS et transférer son adresse dans la table. Par contre si nous voulons accéder à employé OID4 à partir de Agence, une recherche dans la table indique que l'objet est en MC.

20

Caractéristiques des SGBDOO

- 2- Persistance : 3 schémas de réalisation

- Etablissement de points de contrôle
- Basée sur l'accessibilité
- Basée sur l'allocation

Rappel

Objets persistants sont des objets qui survivent après la fin du programme qui les a créés.

21

Caractéristiques des SGBDOO : persistance par
établissement de points de contrôle

- Copie de la totalité ou d'une partie de l'espace d'adressage d'un programme d'application dans la MS. En cas d'interruption, le programme peut redémarrer à partir de « checkpoint ».

22

Caractéristiques des SGBDOO
Persistance basée sur l'accessibilité

- Un objet est persistant s'il est accessible par un objet racine persistant. Le système gère alors une arborescence d'accessibilité.

23

Caractéristiques des SGBDOO : persistance basée sur l'allocation

- Un objet ne devient persistant que s'il a été déclaré comme tel au sein du programme d'application.

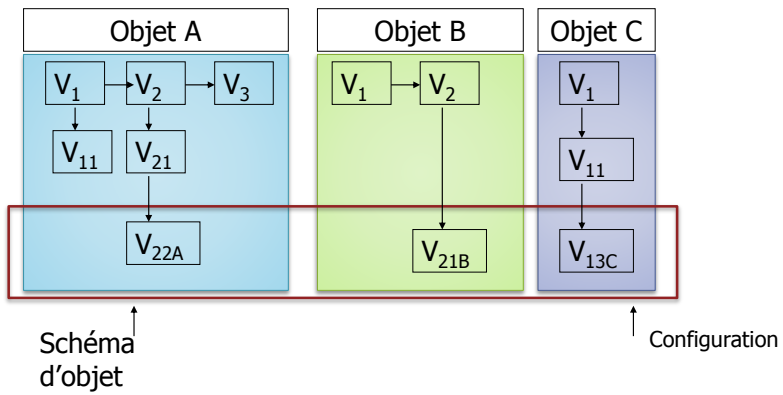
24

Notions spécifiques aux SGBDOO : la gestion de versions, l'évolution de schémas

- Versions :
 - Le processus de conservation de l'évolution des objets s'appelle la gestion des versions.
 - Une version d'un objet représente un état identifiable de l'objet.
 - Un historique des versions représente l'évolution de l'objet

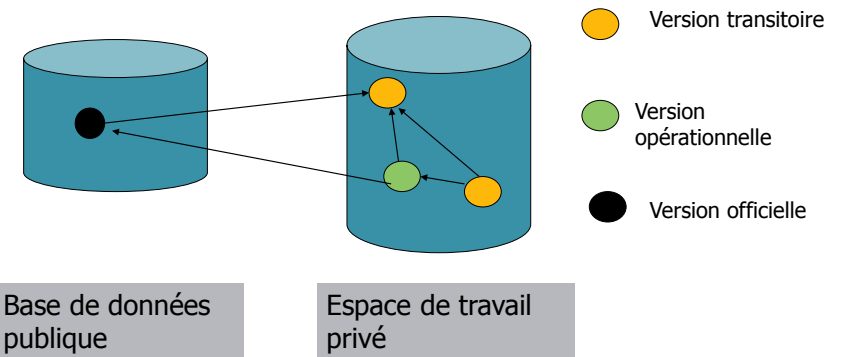
25

Exemple



26

Gestion de versions



27

Notions spécifiques aux SGBDOO

■ Evolution de schéma

■ Modification de la définition de classes

- Modifier les attributs
- Modifier les méthodes

■ Modification de la hiérarchie d'héritage

- Rajouter une superclasse
- Supprimer une classe de la liste des superclasses
- Modifier l'ordre des superclasses

■ Modification sur l'ensemble des classes : création, suppression de classes, modification des noms des classes.

28

Notions spécifiques aux SGBDOO

- Résolution des conflits dus à l'héritage multiple, à la redéfinition d'attributs, et de méthodes dans une sous-classe.

- Règle de précedence des sous-classes sur les super-classes:

Quand un attribut (ou une méthode) est défini dans une classe avec le même nom qu'un attribut (ou une méthode) d'une superclasse, la définition spécifiée dans la sous-classe prend le pas sur la définition de la superclasse

- Règle de précedence entre superclasses d'origines différentes:

Quand plusieurs superclasses ont des attributs (ou des méthodes) de mêmes noms mais d'origines différentes, l'attribut ou la méthode de la première superclasse est hérité dans la sous classe.

- Règle de précedence entre superclasses de même origine:

Quand plusieurs superclasses possèdent des attributs ou des méthodes de même nom et de même origine, l'attribut ou la méthode est hérité(e) une seule fois.

29

Notions spécifiques aux SGBDOO

- Résolution des conflits dus à l'héritage multiple, à la redéfinition d'attributs, et de méthodes dans une sous-classe.
 - Règle de précedence des sous-classes sur les super-classes:
 Quand un attribut (ou une méthode) est défini dans une classe avec le même nom qu'un attribut (ou une méthode) d'une superclasse, la définition spécifiée dans la sous-classe prend le pas sur la définition de la superclasse

30

Notions spécifiques aux SGBDOO

- La propagation des modifications aux sous classes:
 - Règle de propagation dans l'éventualité de conflits
 L'introduction d'un nouvel attribut (ou méthode), ou la modification d'un nom d'attribut (ou d'une méthode) se propage seulement aux sous classes où il n'y a aucun conflit de nom résultant.

31

Notions spécifiques aux SGBDOO

- Règles sur la mise à jour de classes

- Règle d'insertion de superclasses:

Quand une classe C vient s'ajouter à la liste de superclasses d'une classe, cette classe devient la dernière des superclasses de la liste. Tout conflit d'héritage résultant est résolu par les règles précédentes

- Règle de suppression de superclasses:

Quand une classe C est supprimée de la liste des superclasses alors ces sous classes deviennent des sous classes directes de la super classe de C.

32

Notions spécifiques aux SGBDOO

- Règles sur la mise à jour de classes

- Règle d'insertion d'une classe dans un schéma:

Si C n'a aucune superclasse spécifiée alors elle devient une sous classe de la racine du schéma.

- Règle de suppression d'une classe dans un schéma:

Pour supprimer une classe C d'un schéma, la règle de suppression précédente s'applique. La classe racine ne peut être supprimée.

33

SGBD relationnels objet

- Les SGBDOO n'ont pas eu dans le commerce l'engouement qu'avaient eu avant eux les SGBD relationnels.
- SGBDRO : système hybride entre SGBDR et SGBDOO.
- SGBDRO préserve le capital de connaissances des SGBDR
- SGBDRO offre certaines fonctionnalités des SGBDOO pour les applications évoluées.
- Extension de la norme SQL : SQL 99 et SQL 2003.

34

Avantages des SGBDRO

- Résolution des faiblesses des SGBDR (données simples,)
- Prise en compte de la réutilisation et du partage par l'offre de fonctionnalités standard de manière centralisée au niveau du serveur de SGBD.
- Préservation du noyau de connaissances et d'expérience investi dans le développement d'applications relationnelles.

35

Inconvénients des SGBDRO

- Accroissement des coûts et de la complexité
- Simplicité et normalité du modèle relationnel disparaît
- La complexité excessive pour étendre le modèle relationnel ne satisfait pas les puristes du modèle objet.