

Dr. Rim Samia Kaabi

Bases de Données Avancées
Le modèle objet

1

BD à objets???

BD objet: Langage programmation OO +
persistance

2

SGBD OO???



- Persistance des objets: objet persistant vs objet transient
- Concurrence d'accès: transactions multiples, nombre de blocage minimal
- Fiabilité des objets: transaction totalement exécutée ou pas du tout.
- Facilité d'interrogation
- Modèles de transaction évolués: transaction imbriquée

3

SGBD OO???



- Versions d'objets: objet dont l'historique des instances créées (successivement ou simultanément) est gardé dans la base sous forme de versions consultables et modifiables
- Support d'objets atomiques et complexes
- Identité d'objets
- Héritage simple
- Polymorphisme

4

Gestion de la persistance

→ **oid = Persist (<Nom>,<Ref>) ;**

L'objet persistant est retrouvé à partir de son nom, puis monté en mémoire à partir de son identifiant d'objet

//retrouver l'oid d'un objet persistant à partir d'un nom

Oid= Lookup (<Nom>) ;

//Activer un objet persistant désigné par son oid

Ref= activate (<oid>)

5

Gestion de la persistance

→ **oid = Persist (<Nom>,<Ref>) ;**

L'objet persistant est retrouvé à partir de son nom, puis monté en mémoire à partir de son identifiant d'objet

//retrouver l'oid d'un objet persistant à partir d'un nom

Oid= Lookup (<Nom>) ;

//Activer un objet persistant désigné par son oid

Ref= activate (<oid>)

//Désactiver un objet persistant actif

oid= DesActivate (<Ref>)

//Supprimer un objet persistant désigné par son nom

Void UnPersist (<Nom>) ;

//Supprimer un objet persistant désigné par son identifiant

Void UnPersist (<oid>,)

6

Persistence des objets

Objet persistant: objet stocké dans la base dont la durée de vie est supérieure au programme qui l'a créé ou modifié

Un objet créé est rendu persistant n'a pas besoin d'être écrit explicitement sur disque

Objet transitoire: objet du mémoire dont la durée de vie ne dépasse pas celle du programme qui l'a créé

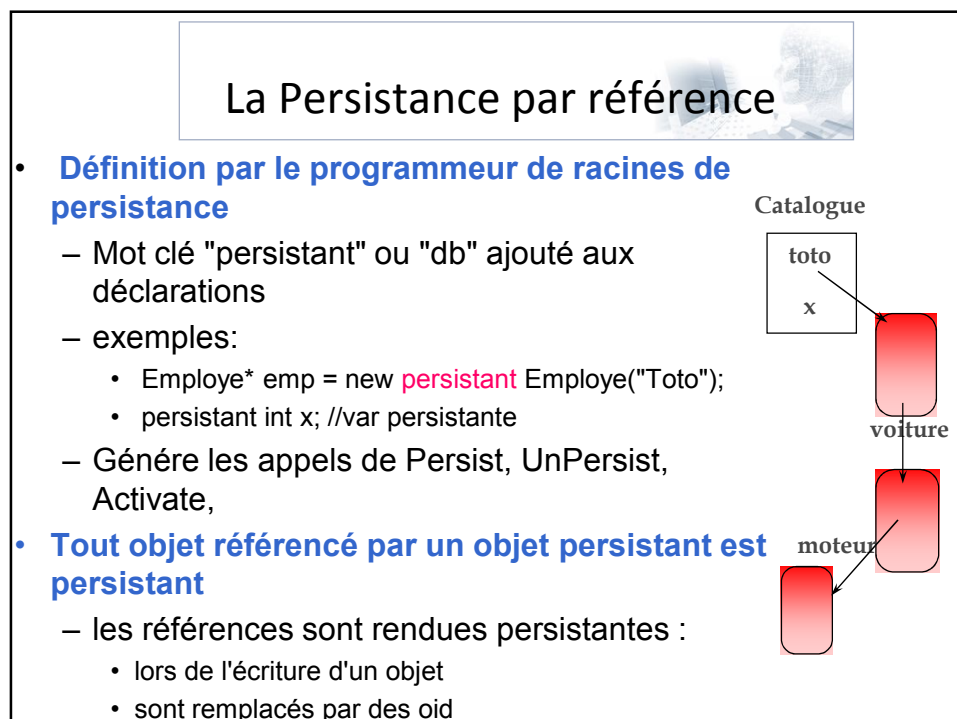
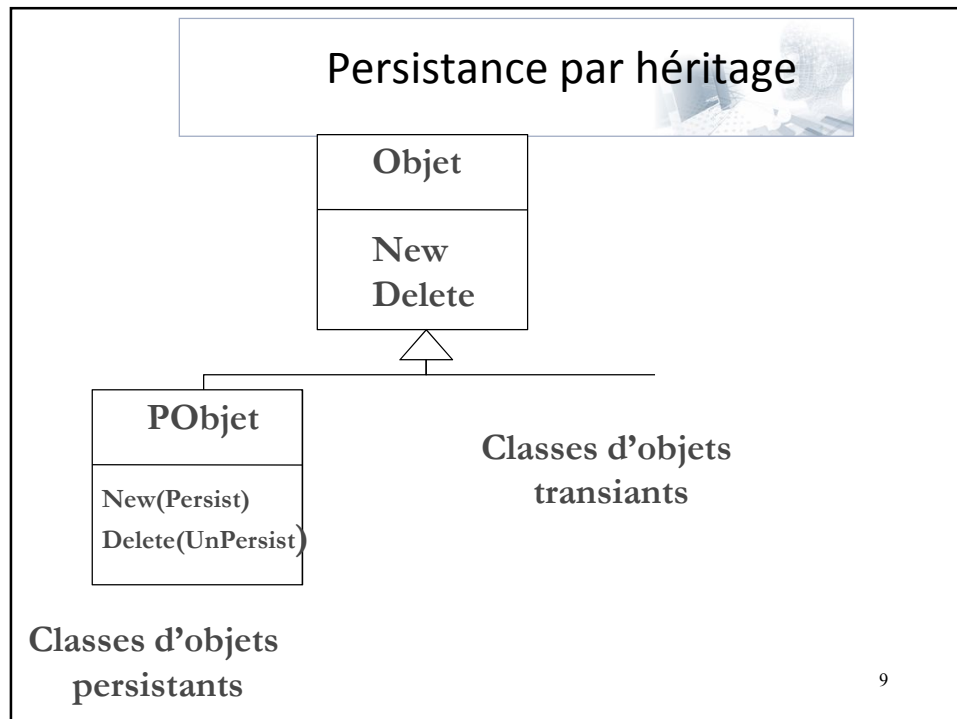
7

Persistence par héritage

Technique permettant de définir la qualité d'un objet à être persistant par héritage d'une classe racine de persistance, rendant invisible l'activation et la désactivation des objets

→ La persistance n'est pas orthogonale au type

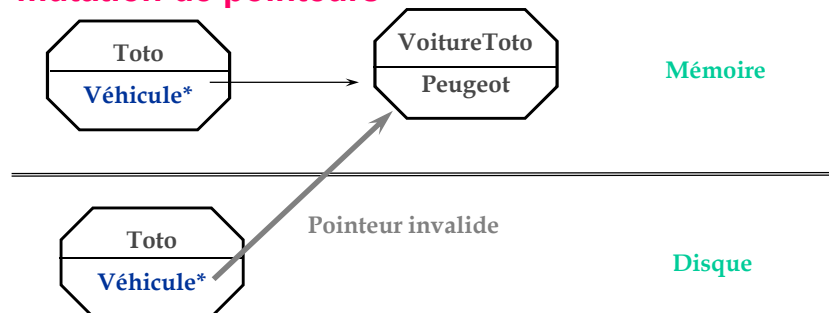
8



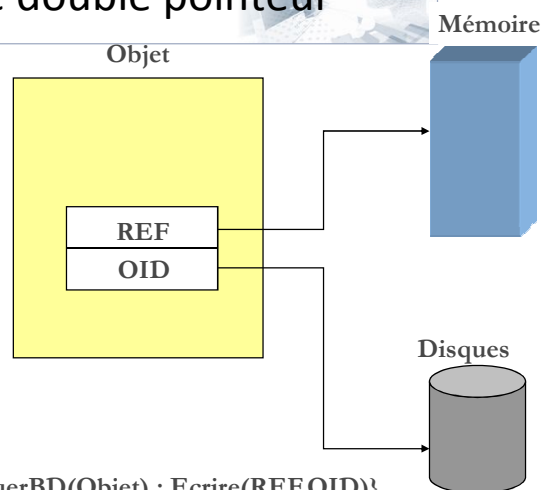
Navigation dans une BD OO

- Parcours dans une BD OO par suivi des pointeurs entre objets
- Les adresses mémoires doivent être traduites en adresses BD (oid) lors des écritures et vice versa

→ **mutation de pointeurs**



Approche double pointeur



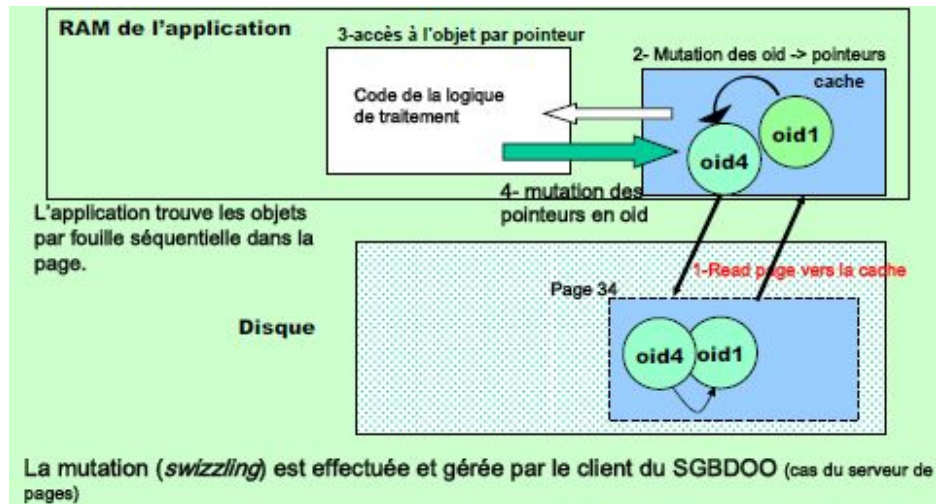
En écriture

Si $OID = NUL$ Alors { $OID = AllouerBD(Objet)$; $Ecrire(REF,OID)$ }

En lecture

Si $REF = NUL$ Alors { $REF = AllouerMC(Objet)$; $Lire (OID,REF)$ }

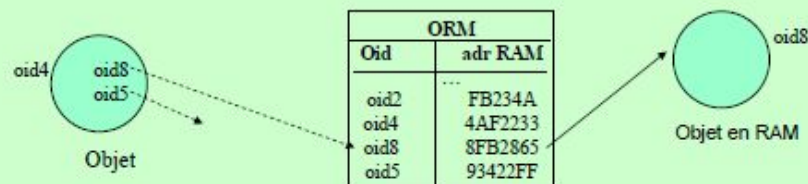
Serveur de pages: accès à un objet: swizzling



13

Implémentation de la localisation des objets

- Lors d'un référencement d'objet par oid:
- 1- Si objet est dans la RAM: la table ORM est consultée pour trouver l'adresse RAM de l'objet.

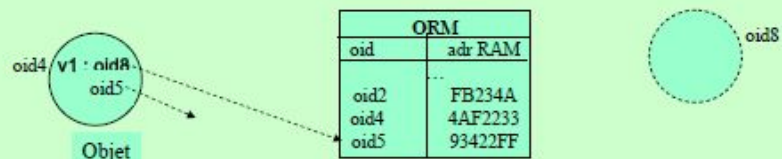


Évaluation de la variable : x.objet.v1 avec x de type objet

14

Implémentation de la localisation des objets

2- Objet est absent de la RAM: la table ORM est consultée pour trouver que oid8 est absent

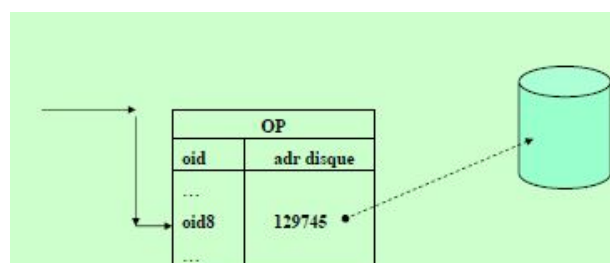


Évaluation de la variable : x.objet.v1 avec x de type objet

15

La table OP d'accès aux objets

- Chaque objet persistant de la BD est inscrit dans une autre table OP
- La table OP est partiellement en RAM et swappée par l'initiative du noyau selon les besoins

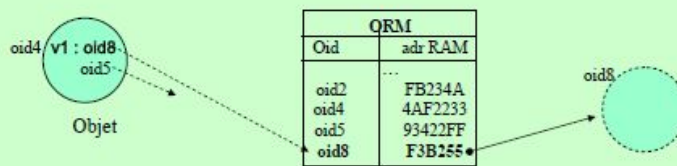


16

Transfert de l'objet oid8

- La table OP est consultée pour trouver l'adresse de oid8 sur disque
- L'objet oid8 est transféré et inscrit dans la table ORM

- 1- La table OP (Table des Objets Persistants) est consultée pour trouver l'adresse de oid8 sur disque.
- 2- L'objet oid8 est transféré et inscrit dans la table ORM



Évaluation de la variable : x.objet.v1 avec x de type objet

17

But de la mutation de pointeur

- Eviter la consultation de tables en matérialisant l'adresse en RAM
- Techniques de matérialisation des adresses:
 - 1 en place/copie
 - 2 Tous/sur demande
 - 3 direct et indirect

18

Versions d'objets

- Le versionnage conserve plusieurs générations modifiées du même objet
- Ce mécanisme permet de revenir en arrière et d'interroger les anciens états des objets à des dates précises (BD temporelles)
- Les versions sont conservées en totalité ou en image différentielle: seuls alors les ajouts ou les modifications à la version précédente sont stockés dans un espace spécialisé. Au besoin la version est reconstituée par le SGBD

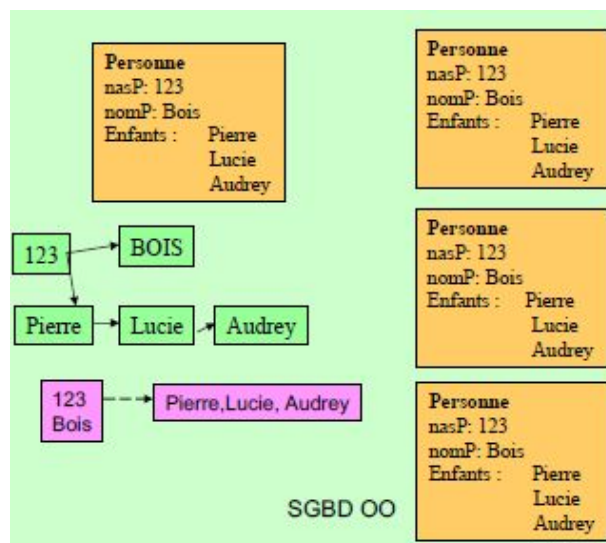
19

Une même structure de l'objet

À l'écran

En RAM

Sur disque



Une même structure de l'objet