

IFT3030

Base de données

Chapitre 10

Modèle et bases de données à objets

Cours tiré du livre Gardarin99

- ☒ Introduction
- ☒ Architecture
- ☒ Modèles de données
- ☒ Modèle relationnel
- ☒ Algèbre relationnelle
- ☒ SQL
- ☒ Conception
- ☒ Fonctions avancées
- ☒ Concepts avancés
- ☒ **Modèle des objets**
- ☒ **BD à objets**

Plan du cours

- Introduction
- Architecture
- Modèles de données
- Modèle relationnel
- Algèbre relationnelle
- SQL
- Conception
- Fonctions avancées
- Concepts avancés
- **Modèle des objets**
- **BD à objets**

2

TECHNIQUES DES SGBDO

- 1. Qu'est-ce qu'un SGBDO ?
- 2. Architectures des SGBDO
- 3. Persistance des objets
- 4. Langages de requêtes

Rappel modèle objet

- **Objet**
 - Identifiant
 - Classe
 - Attributs
 - Opérations
 - Héritage
 - Objets complexes (collections)
 - Polymorphisme

1. Qu'est-ce-qu'un SGBDO ?

- The Object-Oriented Database System Manifesto
 - Atkinson, Bancilhon, Dewitt, Ditrich, Maier, Zdonick (DOOD'89)
- Fonctionnalités BD obligatoires :
 - la persistance
 - la concurrence
 - la fiabilité
 - la facilité d'interrogation
- Fonctionnalités BD optionnelles :
 - la distribution
 - les modèles de transaction évolués
 - les versions

5

Fonctionnalités objets

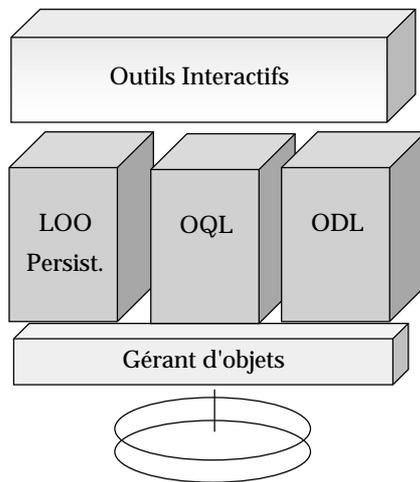
- Fonctionnalités objets obligatoires :
 - les objets atomiques et complexes
 - l'identité d'objets
 - l'héritage simple
 - le polymorphisme (surcharge)
- Fonctionnalités objets optionnelles :
 - l'héritage multiple
 - les messages d'exception

6

- ☒ Introduction
- ☒ Architecture
- ☒ Modèles de données
- ☒ Modèle relationnel
- ☒ Algèbre relationnelle
- ☒ SQL
- ☒ Conception
- ☒ Fonctions avancées
- ☒ Concepts avancés
- ☒ **Modèle des objets**
- ☒ BD à objets

2. ARCHITECTURES

■ Architecture fonctionnelle type :



- Editeur de classes
- Manipulateur d'objets
- Bibliothèques graphiques
- Débogueur, éditeur

OQL = Object Query Language

ODL = Object Definition Language

LOO = Langage Orienté Objet

- | | |
|------------------|---------------|
| • Persistance | • Concurrence |
| • Identification | • Fiabilité |
| • Accès | • Sécurité |

7

- ☒ Introduction
- ☒ Architecture
- ☒ Modèles de données
- ☒ Modèle relationnel
- ☒ Algèbre relationnelle
- ☒ SQL
- ☒ Conception
- ☒ Fonctions avancées
- ☒ Concepts avancés
- ☒ **Modèle des objets**
- ☒ BD à objets

Architecture opérationnelle

■ Problèmes :

- comment distribuer les fonctions du SGBDO ?
- quel protocole de communication utiliser ?
- quelle organisation en processus et tâches répartis ?

■ Constatation

- environnement de stations en réseau très fréquent
- puissance croissante des postes de travail
- le débit réseau n'est plus un goulot (100 Mbit/sec.)

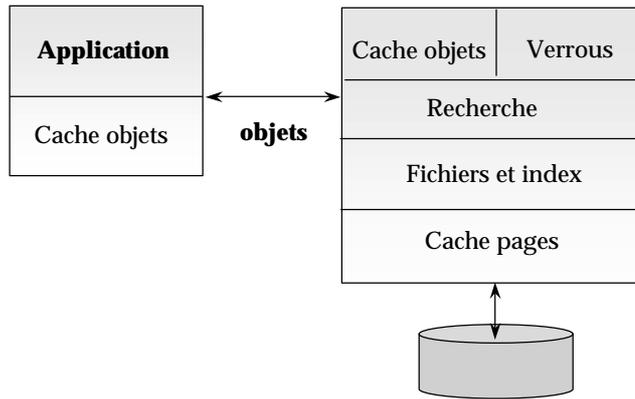
■ Conclusion

- puissance de traitement importante au niveau des clients
- une grande partie des fonctions peut résider à ce niveau

8

- ☒ Introduction
- ☒ Architecture
- ☒ Modèles de données
- ☒ Modèle relationnel
- ☒ Algèbre relationnelle
- ☒ SQL
- ☒ Conception
- ☒ Fonctions avancées
- ☒ Concepts avancés
- ☒ **Modèle des objets**
- ☒ BD à objets

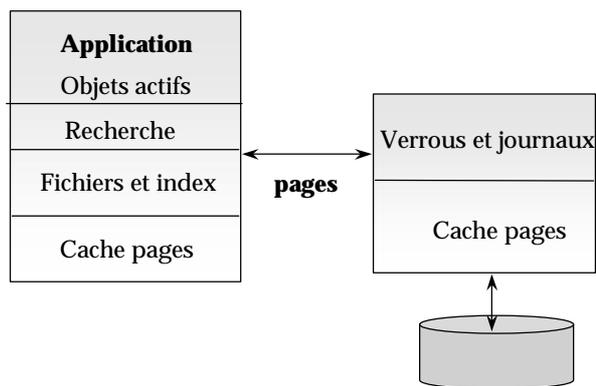
Serveur d'objets



9

- ☒ Introduction
- ☒ Architecture
- ☒ Modèles de données
- ☒ Modèle relationnel
- ☒ Algèbre relationnelle
- ☒ SQL
- ☒ Conception
- ☒ Fonctions avancées
- ☒ Concepts avancés
- ☒ **Modèle des objets**
- ☒ BD à objets

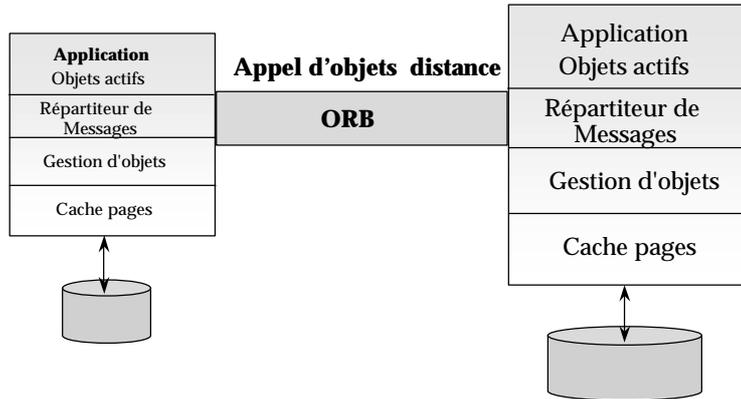
Serveur de pages



10

- ☒ Introduction
- ☒ Architecture
- ☒ Modèles de données
- ☒ Modèle relationnel
- ☒ Algèbre relationnelle
- ☒ SQL
- ☒ Conception
- ☒ Fonctions avancées
- ☒ Concepts avancés
- ☒ **Modèle des objets**
- ☒ BD à objets

Serveur de méthodes



11

- ☒ Introduction
- ☒ Architecture
- ☒ Modèles de données
- ☒ Modèle relationnel
- ☒ Algèbre relationnelle
- ☒ SQL
- ☒ Conception
- ☒ Fonctions avancées
- ☒ Concepts avancés
- ☒ **Modèle des objets**
- ☒ BD à objets

Comparaisons

	Points forts	Points faibles
Serveur d'objets	<ul style="list-style-type: none"> • méthodes sur client ou serveur • concurrence niveau objet 	<ul style="list-style-type: none"> • serveur centralisé • duplication de fonctions
Serveur de pages	<ul style="list-style-type: none"> • distribution de pages possible • serveur plus simple • hétérogénéité difficile 	<ul style="list-style-type: none"> • méthodes sur le serveur imposs. • concurrence niv. objet difficile
Serveur de méthodes	<ul style="list-style-type: none"> • méthodes sur client ou serveur • transfert de messages standards • système uniforme et extensible 	<ul style="list-style-type: none"> • performances pour petits objets • peu bases de données • Corba et service BD ?

12

- ☒ Introduction
- ☒ Architecture
- ☒ Modèles de données
- ☒ Modèle relationnel
- ☒ Algèbre relationnelle
- ☒ SQL
- ☒ Conception
- ☒ Fonctions avancées
- ☒ Concepts avancés
- ☒ **Modèle des objets**
- ☒ BD à objets

3. PERSISTANCE DES OBJETS

■ Objectifs

- faire persister les objets du LOO sans les démonter
- assurer le plus possible la transparence pour le programmeur
- garder des performances proches du travail en mémoire
- récupérer les emplacements des objets détruits

■ Moyens

- gérant d'objets persistants assurant concurrence et fiabilité
- identifiant d'objets permettant de retrouver les objets en 1 ou 2 accès
- ramasse-miettes périodique

13

- ☒ Introduction
- ☒ Architecture
- ☒ Modèles de données
- ☒ Modèle relationnel
- ☒ Algèbre relationnelle
- ☒ SQL
- ☒ Conception
- ☒ Fonctions avancées
- ☒ Concepts avancés
- ☒ **Modèle des objets**
- ☒ BD à objets

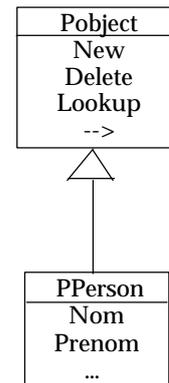
La Persistance par héritage

■ Classe racine de persistance PObject

- tout objet d'une sous-classe hérite des propriétés de persistance
- New, Delete et Envoi de message sont surchargés
- une primitive Lookup est ajoutée pour les recherches

■ Non orthogonale au type

- seuls les types héritants de PObject persistent
- nécessité de dupliquer les classes persistantes et transiantes
- possibilité de marquer la persistance pour éviter la duplication (attribut booléen Persistant)



14

La Persistance par référence

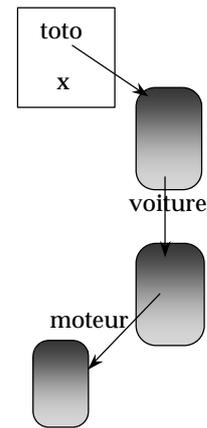
■ Définition par le programmeur de racines de persistance

- Mot clé "persistant" ou "db" ajouté aux déclarations
- exemples:
 - `Employe* emp = new persistant Employe("Toto");`
 - `persistant int x;`
- un objet racine de persistance est catalogué
 - accessible par lookup

■ Tout objet référencé par un objet persistant est persistant

- les références sont rendues persistantes :
 - lors de l'écriture d'un objet les références
 - sont remplacés par des oid

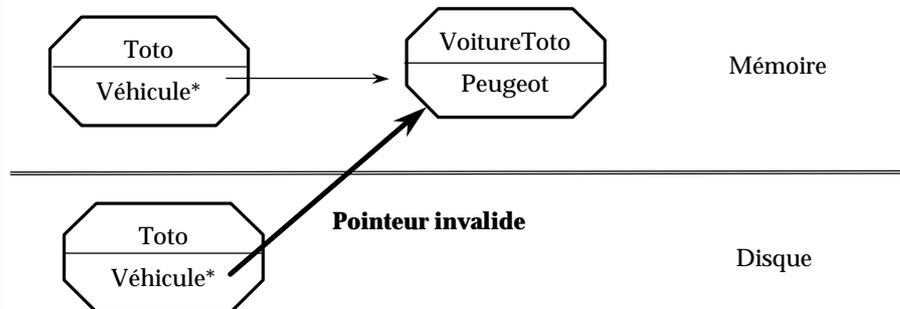
Catalogue



15

Translation des pointeurs

■ Les adresses mémoires doivent être translitées en adresses BD (oid) lors des écritures et vice versa



16

Approche double pointeur

- Tout pointeur sur un objet persistant est remplacé par un double pointeur

Ad MC

Ad BD

- En écriture
 - si Ad BD inconnue ALORS { Ad BD = AllouerBD(Objet) ; Ecrire (Ad BD, Objet) } ;
- En lecture
 - si Ad MC inconnue ALORS {Ad MC = AllouerMC(Ad BD); Lire(Ad MC, Ad BD) } ;
- Inconvénients
 - objets persistants versus objets transients (doubles pointeurs)
 - lecture des objets pointés
 - faiblesse des performances

17

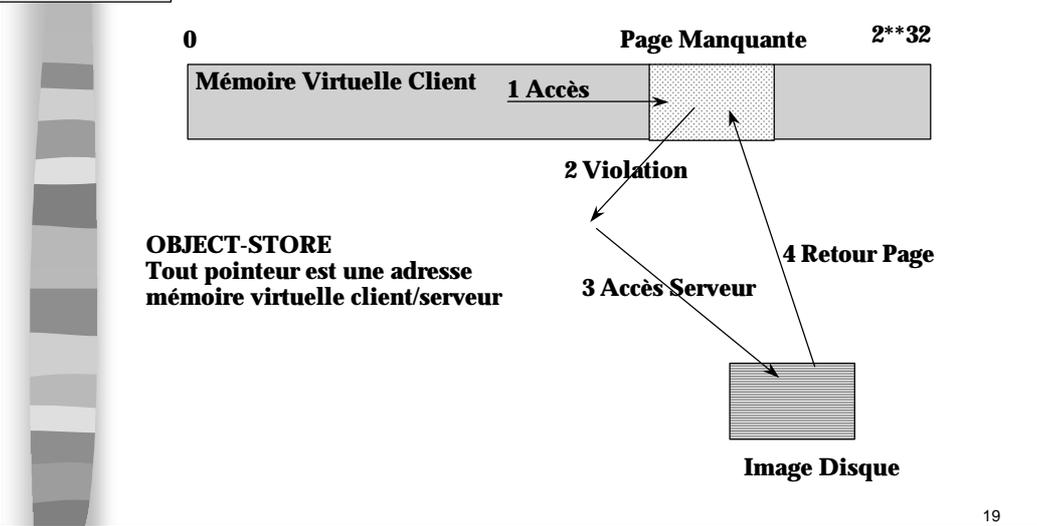
Approche mémoire virtuelle (1)

- Ecriture des objets sans modification
 - Pointeurs disques = adresse mémoire virtuelle
 - Sauvegarde des pages dans des partitions BD images de la mémoire
- Pré-allocation des objets en mémoire virtuelle
 - Lorsqu'un objet est retrouvé, des pages inaccessibles sont allouées pour tous les objets référencés.
 - L'objet est chargé lors du premier accès par récupération de la violation mémoire virtuelle.
- Cette technique a été brevetée par ODI

18

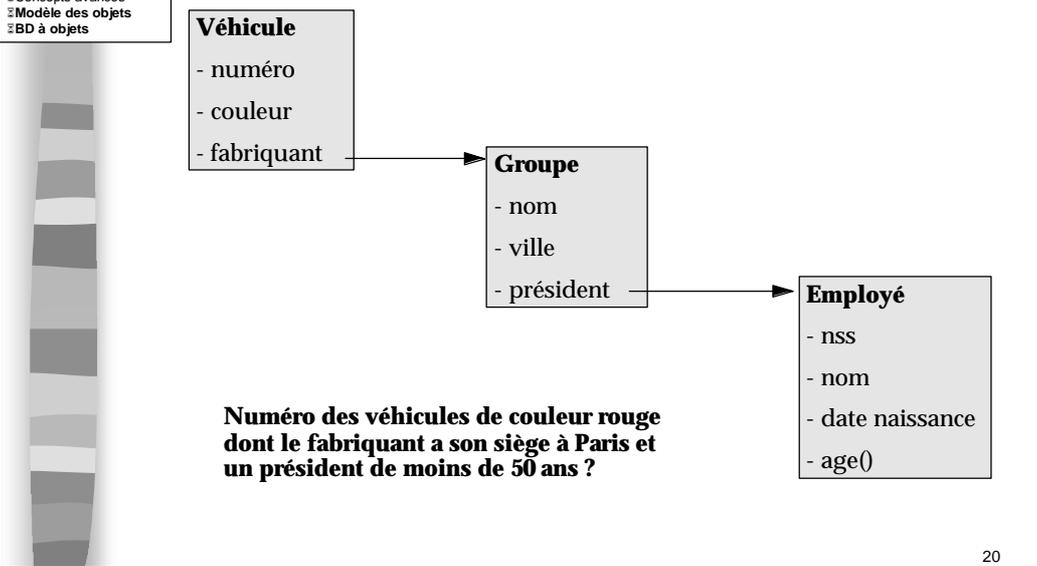
- ☒ Introduction
- ☒ Architecture
- ☒ Modèles de données
- ☒ Modèle relationnel
- ☒ Algèbre relationnelle
- ☒ SQL
- ☒ Conception
- ☒ Fonctions avancées
- ☒ Concepts avancés
- ☒ **Modèle des objets**
- ☒ BD à objets

Approche mémoire virtuelle (2)



- ☒ Introduction
- ☒ Architecture
- ☒ Modèles de données
- ☒ Modèle relationnel
- ☒ Algèbre relationnelle
- ☒ SQL
- ☒ Conception
- ☒ Fonctions avancées
- ☒ Concepts avancés
- ☒ **Modèle des objets**
- ☒ BD à objets

4. LANGAGES DE REQUETES



Langage navigationnel

- Parcours un objet à la fois via les pointeurs

```

véhicule* v, groupe* g, personne* p;
  for each v in vehicule
    { if v->couleur == "Rouge" then
      { g = v->fabriquant;
        if g->ville == "Paris" then
          { p = g->president;
            if p->age() < 50 then cout <<
              v.numéro; } ; } ; } ;
  
```

21

Les SQL Objets

- Extension ou Variation de SQL (adapté aux objets)

- parcours de chemin : Voiture.fabriquant.président => Personne
- exécution de méthodes : voiture.démarrer()
- prise en compte de l'héritage
- présentation des résultats : ensemble ou classe
- parcours des collections

- Problème de compatibilité avec la norme SQL

- travaux de l'ODMG (ODL et OQL 93/96)
- normalisation de SQL3 prévue pour 1999

22

- ☒ Introduction
- ☒ Architecture
- ☒ Modèles de données
- ☒ Modèle relationnel
- ☒ Algèbre relationnelle
- ☒ SQL
- ☒ Conception
- ☒ Fonctions avancées
- ☒ Concepts avancés
- ☒ **Modèle des objets**
- ☒ BD à objets

Principaux SQL objets

FSQL	HP	IRIS OPEN-DB
OSQL	MCC	ORION
XSQL	UNISQL	UNISQL
SQL3	ISO	Norme ?
ISQL	C.A.	INGRES
O-SQL	ORACLE	ORACLE
VSQL	VERSANT	VERSANT
O2SQL	O2	O2
ESQL	INRIA-BULL	DBS3
OQL	ODMG	Norme objet ?