

# LE PROGRAMME COMPREND CINQ CHAPITRES

Chapitre I: **Préliminaires**

Chapitre II: **Concepts du modèle objet et principaux apports**

Chapitre III: **Le modèle ODMG et et le langage ODL**

Chapitre IV: **Relationnel-Objet**



# Bases de données orientées objets

## Chapitre I: Préliminaires



# RAPPELS: DÉFINITIONS

- ❖ **BD**: par définition, c'est un ensemble de données permanentes, intégrées, partagées, en accès simultané
- ❖ **SGBD**: acronyme de système de gestion de bases de données. Il doit assurer:
  - ❖ L'intégrité des données: assurer les règles d'intégrité générales et spécifiques en affichant des messages d'erreurs en cas de transgression (clé primaire toujours renseignée, clé étrangère doit d'abord exister en tant que clé primaire, respect de format spécifique d'un champ...)
  - ❖ Le partage des données: architecture client/serveur
  - ❖ L'accès simultané: mise en place d'algorithmes pour maintenir la cohérence de la BD en cas d'opérations d'écriture simultanée: verrou mortel...



# RAPPELS: DÉFINITIONS

- ❖ Il doit assurer:
  - ❖ La sécurité de la base de données : création des profils avec mots de passe et droits d'utilisation
  - ❖ L'atomicité des transactions: une transaction est exécutée dans sa totalité ou annulée ( insérer une carte bancaire pour faire un retrait d'argent est une transaction qui lira l'enregistrement du compte utilisateur, vérifiera si le solde est suffisant, puis mettra à jour l'enregistrement en fonction du montant du retrait).
  - ❖ La Fiabilité de la base de données : protection contre les pannes
- ❖ La communication avec les SGBDs se fait par des langages de requêtes et de mises à jour déclaratifs
- ❖ Ils se différencient entre eux suivant:
  - ❖ Les techniques de stockage, l'optimisation des requêtes...





# SGBD ET BD PLUS EN DÉTAIL

On doit distinguer entre la description de la base de données et la base de données elle-même:

- ❖ La description d'une base de données s'appelle le **schéma** de base de données, qui est spécifié lors de la conception de la base de données et qui n'est pas censé changer fréquemment (description des propriétés des champs, des contraintes...).
- ❖ Les états de la base de données : un **état** (ou instance ou occurrence) correspond aux données d'une BD à un moment donné. Il change souvent chaque fois que l'on fait une manipulation sur la BD (suppression, mise à jour...)



## SGBD ET BD PLUS EN DÉTAIL

- ❖ Lorsque nous définissons une nouvelle base de données, nous spécifions son schéma de base de données uniquement au SGBD (il s'agit des **méta données** stockées dans le **catalogue** par le SGBD)
- ❖ À ce stade, l'état correspondant de la base de données est **l'état vide**, sans données.
- ❖ **L'état initial** de la base de données est lorsque celle-ci est alimentée ou chargée pour la première fois avec les données initiales.
- ❖ Chaque fois qu'une opération de mise à jour est appliquée à la base de données, nous obtenons un autre état de la base de données.
- ❖ Le SGBD est en partie responsable de s'assurer que chaque état de la base de données est **cohérent**, c'est-à-dire un état qui satisfait à la structure et aux contraintes spécifiées dans le schéma.



# SGBD ET BD PLUS EN DÉTAIL

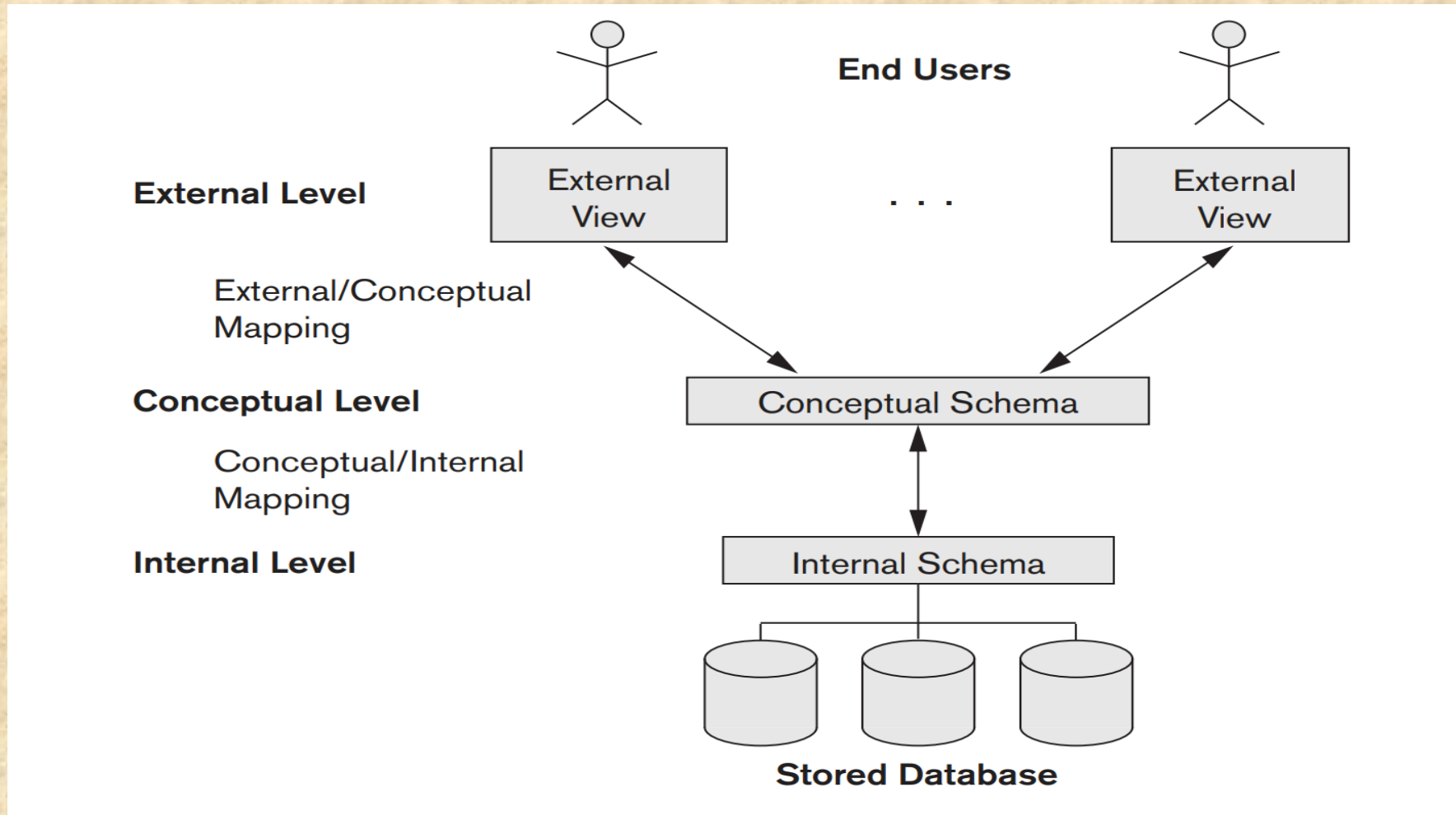
Pour assurer ses fonctions, un SGBD présente en général une architecture à trois niveaux:

- ❖ **Niveau interne:** qui possède un schéma interne qui décrit la structure de stockage physique de la base de données. Il utilise un modèle de données physiques et décrit les détails complets du stockage des données et des chemins d'accès à la la base de données.
- ❖ **Niveau conceptuel:** qui possède un schéma conceptuel qui est une description de haut niveau de l'ensemble de la base de données pour l'ensemble des utilisateurs. Il cache les détails des structures de stockage physiques et se concentre sur la description des entités, des types de données, des relations, des opérations utilisateur et des contraintes
- ❖ **Niveau externe:** qui possède des schémas externes qui décrivent les vues des différents groupes d'utilisateurs.





# SGBD ET BD PLUS EN DÉTAIL



Architecture à trois niveau d'un SGBD

Source: Fundamentals-of-Database-Systems-Pearson-2015-  
Ramez-Elmasri-Shamkant-B.-Navathe





# SGBD ET BD PLUS EN DÉTAIL

- ❖ Notez que les trois schémas ne sont que des descriptions des données ; les données réelles ne sont stockées qu'au niveau physique.
- ❖ Chaque groupe d'utilisateurs fait référence à son propre schéma externe. Le SGBD doit donc transformer une requête spécifiée sur un schéma externe en une requête sur le schéma conceptuel, puis en une requête sur le schéma interne pour la traiter sur la base de données stockée.
- ❖ Si la demande est une extraction de base de données, les données extraites de la base de données stockée doivent être reformatées pour correspondre à la vue externe de l'utilisateur. Les processus de transformation des demandes et les résultats entre les niveaux sont appelés **mappings**.
- ❖ La plupart des SGBD ne séparent pas complètement ces trois niveaux



# SGBD ET BD PLUS EN DÉTAIL

L'architecture à trois niveaux peut être utilisée pour mieux expliquer le concept **d'indépendance des données**, qui peut être défini comme la capacité à modifier le schéma à un niveau d'un système de base de données sans avoir à modifier le schéma du niveau supérieur suivant. On peut définir deux types **d'indépendance des données**:

- ❖ **Indépendance logique**: la capacité de modifier le schéma conceptuel sans avoir à modifier les schémas externes ou les programmes d'application (en ajoutant ou en supprimant un type d'enregistrement ou un élément de données)
- ❖ **Indépendance physique**: la capacité de modifier le schéma interne sans avoir à modifier le schéma conceptuel. Par conséquent, les schémas externes ne doivent pas être modifiés non plus (certains fichiers physiques sont réorganisés, par exemple en créant des structures d'accès supplémentaires, afin d'améliorer les performances d'extraction ou de mise à jour).



# CLASSIFICATION DES SGBDs

Plusieurs critères peuvent être utilisés pour classer les SGBDs:

1. le modèle de données sur lequel le SGBD est basé
2. le nombre d'utilisateurs supportés par le système.
3. le nombre de sites sur lesquels la base de données est répartie
4. le coût
5. les types d'options de chemin d'accès pour le stockage des fichiers
6. l'usage: à usage général ou à usage spécifique





# CLASSIFICATION DES SGBDS: MODÈLES DE DONNÉES

En se focalisant plus particulièrement sur les modèles de données, on distingue:

- ❖ Les modèles de données de haut niveau ou conceptuels : fournissent des concepts qui sont proches de la façon dont de nombreux utilisateurs perçoivent les données (**modèle entité association et ses extensions, modèle de données objet**)
- ❖ Les modèles de données de bas niveau ou physiques: fournissent des concepts qui décrivent comment les données sont stockées sous forme de fichiers dans l'ordinateur en représentant des informations telles que les formats d'enregistrement, l'ordre des enregistrements et les chemins d'accès. **Un chemin d'accès est une structure de recherche qui rend efficace la recherche d'enregistrements de base de données particuliers, comme l'indexation ou le hachage**(**destinés aux informaticiens, et non aux utilisateurs finaux**)



# CLASSIFICATION DES SGBDS: MODÈLES DE DONNÉES

- ❖ Les modèles de données représentationnels (ou d'implémentation): fournissent des concepts qui peuvent être facilement compris par les utilisateurs finaux mais qui ne sont pas trop éloignés de la façon dont les données sont organisées dans le stockage informatique( modèles hiérarchique, réseau, relationnel, objet).
- ❖ Les modèles de données autodescriptifs: ils combinent la description des données avec les valeurs des données elles-mêmes. Dans les SGBD traditionnels, la description (schéma) est séparée des données. Ces modèles comprennent le XML ainsi que de nombreux systèmes de stockage de type clé-valeur et NOSQL récemment créés pour la gestion des données volumineuses.

**Extensible markup language: langage informatique permettant de mettre en forme un document avec des balises**



# ÉVOLUTION DES SGBDs: MODÈLES DE DONNÉES REPRÉSENTATIONNELS

## SGBD hiérarchique:

Les années 60

- ❖ Données structurées en utilisant des **pointeurs unidirectionnels** (arborescences hiérarchiques : un élément mène uniquement à un autre)
- ❖ Requêtes plutôt **procédurales** que déclaratives (i.e comment aller chercher les données plutôt que quoi aller chercher )
- ❖ Modèle **défini par des produits** et non sur un modèle abstrait (d'où une grande dépendance aux technologies)
- ❖ Produit le plus connu : Information management system (IMS) de IBM (**1968**)





# ÉVOLUTION DES SGBDs: **MODÈLES DE DONNÉES REPRÉSENTATIONNELS**

## **SGBD réseau:**

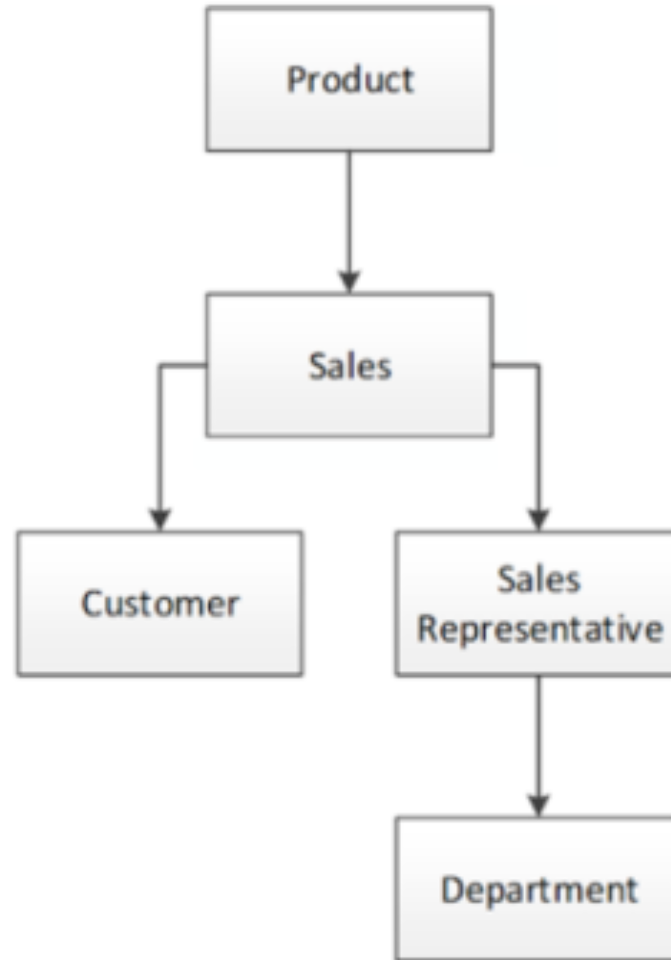
**Les années 60-70**

- ❖ Données structurées en utilisation des **pointeurs bidirectionnels** (un élément peut pointer et être pointé par plus d'un élément)
- ❖ Modèle défini dans un rapport du Data Base Task Group de CODASYL (**1971**) (consortium formé en **1959** pour créer un langage de prog commun qui devint la base de COBOL)
- ❖ Basé sur le langage de **programmation COBOL**
- ❖ Produit le plus connu : Integrated Database Management System (IDMS) de Computer Associates



# ÉVOLUTION DES SGBDs: MODÈLES DE DONNÉES REPRÉSENTATIONNELS

Modèle hiérarchique



Modèle réseau



**Figure 1-3.** Hierarchical and network database models

Source : HARRISON, Guy. *Next generation databases : NoSQL, NewSQL, and Big Data*. [New York] : aPress, 2015, p. 6.



# ÉVOLUTION DES SGBDs: MODÈLES DE DONNÉES REPRÉSENTATIONNELS

**SGBD relationnel: Atouts**

**Les années 70-98 (SQL2)**

- ❖ Défini **formellement** (mathématiquement) avant toute implantation par l'informaticien britannique Edgar Frank Codd en **1970**, ce qui en fait un "vrai" modèle.
- ❖ Les premières implémentations commerciales du modèle relationnel sont apparues au début des années 1980, comme le système SQL/DS sur le système d'exploitation MVS (Multiple Virtual Storage) d'IBM et le SGBD Oracle.
- ❖ Depuis, le modèle a été implémenté dans plusieurs SGDBs commerciaux: DB2 (IBM), Sybase (SAP), Oracle (Oracle), SQL Server et Access (Microsoft), MySQL, PostgreSQL
- ❖ Structure physique cachée aux utilisateurs
- ❖ Langages déclaratifs (SQL) avec des API (Application program interface) pour la plupart des langages de programmation





# EVOLUTION DES SGBDs: MODÈLES DE DONNÉES REPRÉSENTATIONNELS

Les années 70-98 (SQL2)

## SGBD relationnel: Inconvénients

- 1) Les types de données disponibles sont limités et non extensibles:
  - ❖ SQL ne prend en charge qu'un nombre restreint de types intégrés qui traitent des nombres et des chaînes de caractères. Au départ, le seul objet complexe était BLOB (Binary Large Object), mais les fournisseurs incluent désormais d'autres objets tels que CLOB (Character Large Object) et XML\_Type.
  - ❖ Malgré ceci, on ne peut pas considérer des objets complexes tels que des graphiques, des vidéos, des fichiers audio et des documents complets, qui peuvent tous se présenter sous différents formats.



# EVOLUTION DES SGBDs: MODÈLES DE DONNÉES REPRÉSENTATIONNELS

## SGBD relationnel: Inconvénients

### Les années 70-98 (SQL2)

2) Les structures de données telles que les tableaux et les ensembles ne sont pas prises en charge:

- ❖ Les tables relationnelles sont essentiellement des fichiers plats reliés par des jointures.
- ❖ La théorie des ensembles signifie que vous pouvez avoir des données regroupées selon certains critères (par exemple, les personnes âgées de 20 à 29 ans) qui peuvent être visualisées en termes de relations avec d'autres groupes de données (par exemple, les propriétaires de smartphones). L'intersection des deux ensembles serait les données communes aux deux ensembles (les personnes âgées de 20 à 29 ans qui possèdent un téléphone intelligent), tandis que l'union serait toutes les données des deux ensembles (tous les propriétaires de téléphones intelligents et toutes les personnes âgées de 20 à 29 ans).
- ❖ Un tableau permettrait de stocker différentes structures avec différentes opérations. Ce problème nous ramène au premier inconvénient.



# EVOLUTION DES SGBDs: MODÈLES DE DONNÉES REPRÉSENTATIONNELS

Les années 70-98 (SQL2)

## SGBD relationnel: Inconvénients

- 3) Des relations du monde réel ne peuvent être représentées facilement avec le relationnel:
- ❖ Dans le monde réel, les choses sont souvent organisées en hiérarchies.
  - ❖ Une hiérarchie introduit une superclasse à laquelle sont associées une ou plusieurs sous-classes. Les attributs et les méthodes de la superclasse sont hérités par les sous-classes.
  - ❖ La traduction de cette hiérarchie en relationnel entraîne toujours une certaine inefficacité, sous la forme de champs NULL ou de jointures de tables excessives, ce qui a un impact sur les performances de la base de données.





# EVOLUTION DES SGBDs: MODÈLES DE DONNÉES REPRÉSENTATIONNELS

**SGBD relationnel: Inconvénients**

**Les années 70-98 (SQL2)**

- 4) Décalage entre le langage d'accès aux données (SQL) et le langage hôte (par exemple java):
- ❖ dans la programmation orientée objet, l'un des principaux concepts est l'encapsulation. Certains objets encapsulés ont leur représentation cachée. Ils sont appelés objets privés et sont en désaccord avec les représentations des bases de données relationnelles où l'accès est relatif au besoin plutôt qu'à une caractéristique absolue des données.
  - ❖ Les problèmes de ce type sont souvent résolus par une solution de programmation.
  - ❖ Certains fournisseurs ont résolu au moins certains de ces problèmes, mais d'une manière propriétaire plutôt que dans le cadre d'une norme industrielle. Oracle, par exemple, a une implémentation pour les ensembles et les opérateurs pour les manipuler.



## EVOLUTION DES SGBDs: MODÈLES DE DONNÉES REPRÉSENTATIONNELS

**En conséquence:** les SGBDs relationnels **insuffisants**  
face aux nouvelles applications telles que:

- ❖ Conception assistée par ordinateur: *rassemble des outils informatiques (logiciels et matériels) qui permettent de réaliser une modélisation géométrique d'un objet afin de pouvoir simuler des tests en vue d'une fabrication*
- ❖ Production assistée par ordinateur : *programmes de gestion de production permettant de gérer l'ensemble des activités, liées à la production, d'une entreprise industrielle*
- ❖ Gestion des stocks et des achats
- ❖ Création et gestion du planning de fabrication.
- ❖ ...



## EVOLUTION DES SGBDs: MODÈLES DE DONNÉES REPRÉSENTATIONNELS

**En conséquence:** les SGBDs relationnels **insuffisants** face aux nouvelles applications telles que:

- ❖ Systèmes d'informations géographiques: *peuvent stocker et analyser des cartes, des données météorologiques et des images satellites*
- ❖ Les entrepôts de données et les systèmes de traitement analytique en ligne (OLAP): *utilisés dans de nombreuses entreprises pour extraire et analyser des informations commerciales utiles à partir de très grandes bases de données afin de faciliter la prise de décision.*
- ❖ génie logiciel et systèmes multimédia ...





## EVOLUTION DES SGBDs: **MODÈLES DE DONNÉES REPRÉSENTATIONNELS**

**Ces applications ont entraîné de nouveaux besoins en terme de:**

- ❖ Structures de données plus complexes sont nécessaires pour modéliser les nouvelles applications
- ❖ Nouveaux types de données sont nécessaires en plus des types de base numériques et de chaînes de caractères.
- ❖ Nouvelles opérations et constructions de langage de requête sont nécessaires pour manipuler les nouveaux types de données.
- ❖ Nouvelles structures de stockage et d'indexation sont nécessaires pour effectuer des recherches efficaces sur les nouveaux types de données.



## EVOLUTION DES SGBDs: MODÈLES DE DONNÉES REPRÉSENTATIONNELS

Les années 90-...

### SGBD orienté objet: Réponses à ces besoins

- ❖ Les bases de données orientées objet ont adopté de nombreux concepts développés à l'origine pour les langages de programmation orientés objet.
- ❖ L'Object Data Management Group (ODMG), qui s'est formé en 1991 et s'est dissous dix ans plus tard en 2001, a développé un ensemble de normes, dont la dernière était ODMG 3.0 en 2000.
- ❖ Cela a formé la base d'une norme industrielle donnant des directives pour un langage de type SQL pour manipuler les objets, Object Query Language (OQL).



## EVOLUTION DES SGBDs: MODÈLES DE DONNÉES REPRÉSENTATIONNELS

Les années 90-...

### SGBD orienté objet: Réponses à ces besoins

- ❖ À peu près à la même époque, Atkinson et al. (1992) ont produit le Manifeste sur les systèmes de base de données orientés objet. Ce manifeste proposait treize caractéristiques obligatoires:
  - ❖ 1) Objets complexes 2) Identité des objets 3) Encapsulation 4) Types et classes 5) Hiérarchies de types et de classes 6) Surcharge, surcharge et liaison tardive 7) Complétude du calcul 8) Extensibilité 9) Persistance 10) Efficacité 11) Concurrence 12) Fiabilité 13) Langage d'interrogation déclaratif.





## EVOLUTION DES SGBDs: MODÈLES DE DONNÉES REPRÉSENTATIONNELS Les années 90-...

### SGBD orienté objet: Réponses à ces besoins

- ❖ Il s'agissait bien entendu d'un manifeste qui, avec l'ODMG, a servi de guide pour le développement des bases de données orientées objet depuis le début des années 1990.
- ❖ Le langage de requête résultant, Object Query Language (OQL), répondait aux exigences du manifeste et de l'ODMG.
- ❖ Il ressemble beaucoup au SQL normal, mais au lieu de nommer les tables dans la clause SELECT, on nomme les classes d'objets.
- ❖ Dans les années 1990, un certain nombre d'implémentations de bases de données orientées objet par des vendeurs sont apparues telles que: O2 (aujourd'hui propriété d'IBM), JADE et des produits open source plus récents tels que db4o.



## EVOLUTION DES SGBDs: MODÈLES DE DONNÉES REPRÉSENTATIONNELS

### SGBD relationnel-objet:

- ❖ Une grande partie de l'expertise et des systèmes avait été développée autour des bases de données relationnelles et SQL était devenu le langage de requête universel. Donc la reconversion vers le concept objet était coûteuse.
- ❖ La seconde approche de l'orientation objet a donc été d'étendre les bases de données relationnelles pour y incorporer des fonctionnalités orientées objet.
- ❖ Ces bases sont connues sous le nom de bases de données relationnelles objet.
- ❖ Elles sont manipulées par des commandes SQL qui ont été étendues pour traiter les structures d'objets.