

BASES DE DONNÉES

CNAM
Centre associé de Clermont-Ferrand
Cycle A – Année 1997-98

J. Darmont

I. INTRODUCTION

II. LES SYSTÈMES HIÉRARCHIQUES

III. LES SYSTÈMES RÉSEAU

IV. LES SYSTÈMES RELATIONNELS

V. LE LANGAGE SQL

VI. LE MODÈLE E/A

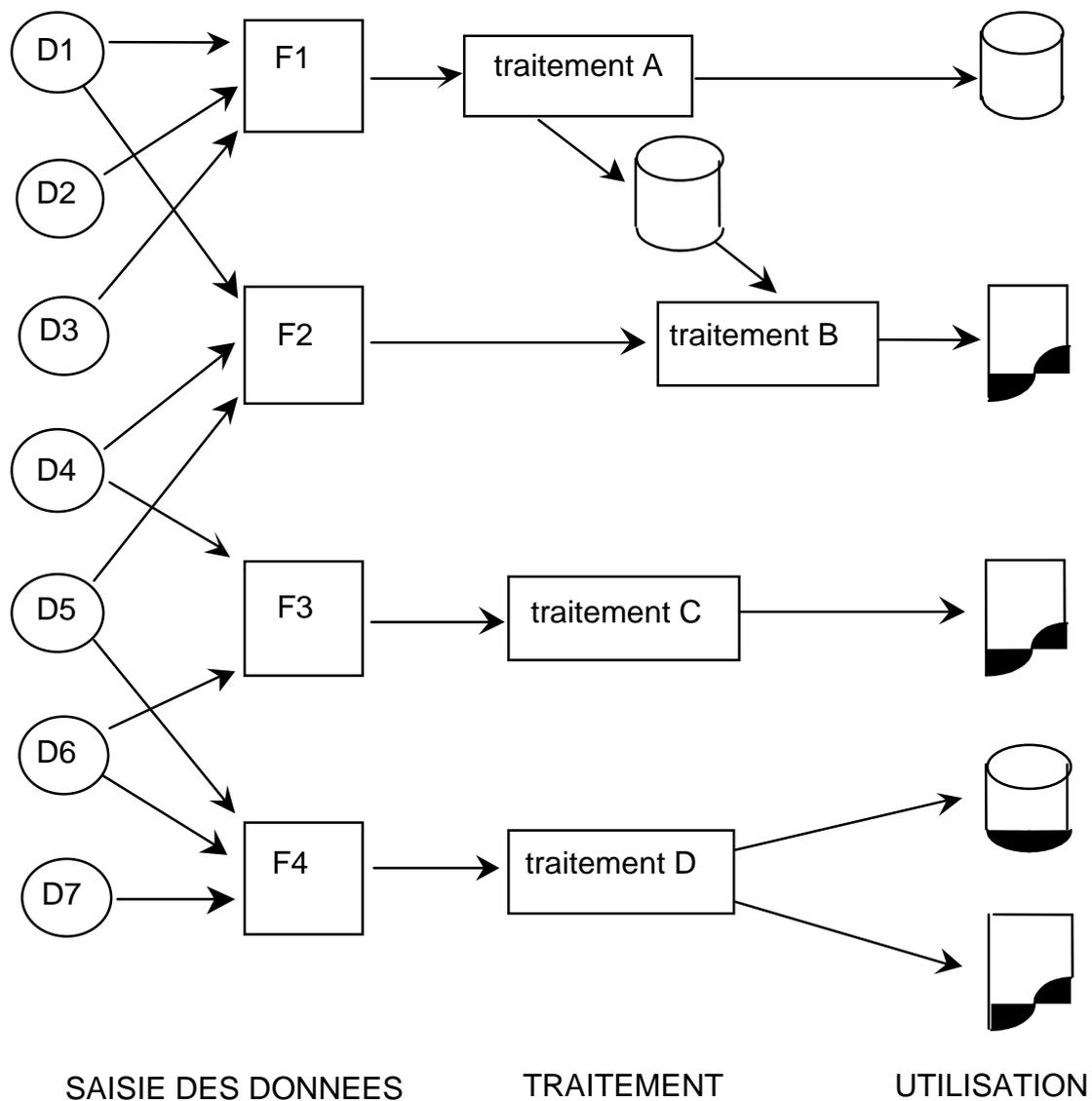
VII. CONCEPTION AVEC LE MODÈLE RELATIONNEL

VIII. ORGANISATIONS PHYSIQUES

I. INTRODUCTION

1. Historique des SGBD¹

Jusqu'aux années 60 : organisation classique en fichiers.

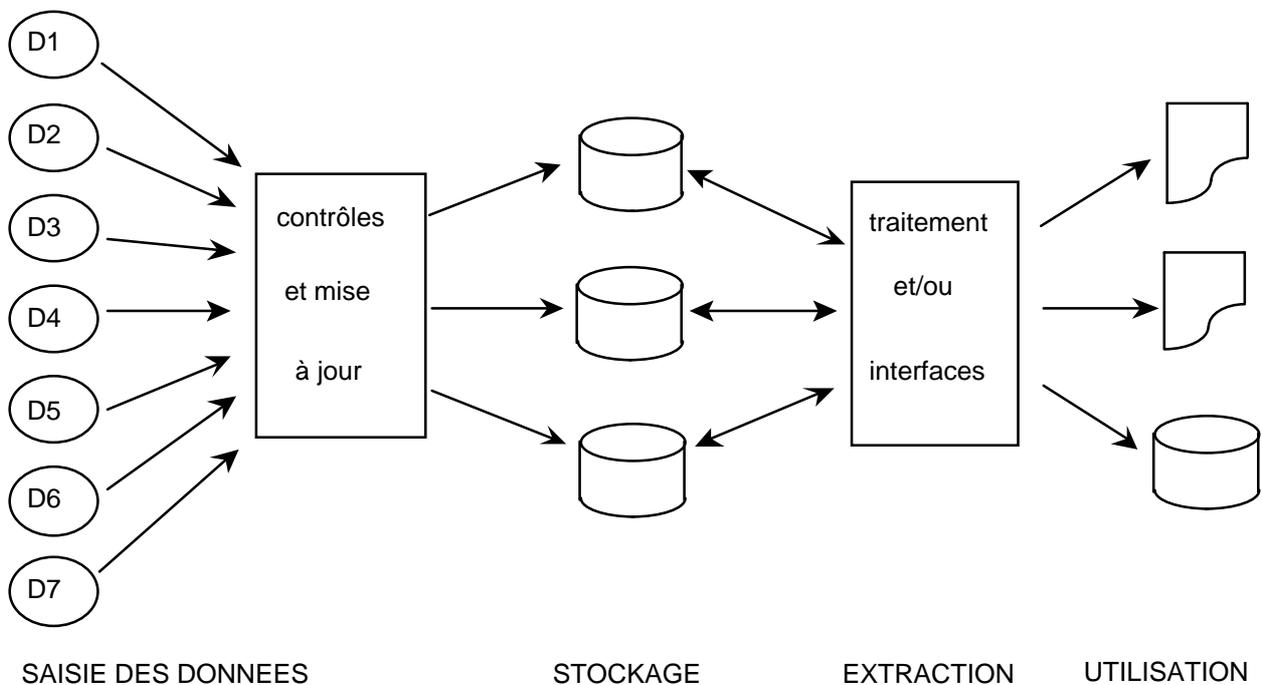


Organisation classique

¹ SGBD = Système de Gestion de Bases de Données

- particularisation de la saisie en fonction des fichiers (un programme de mise à jour pour chaque fichier)
- contrôle en différé des données \Rightarrow recyclages pour les erreurs + augmentation des délais et du risque d'erreur
- particularisation des fichiers en fonction des traitements
- grande redondance des données
- peu de standardisation dans les traitements
- utilisation de nombreux supports intermédiaires

Fin des années 60 : apparition des premiers SGBD. Séparation de la description des données de leur manipulation par les programmes d'application. *Modèles réseau et hiérarchique*. SOCRATE, TOTAL, IDMS, IDS2, IMS2.



Organisation base de données

- uniformisation de la saisie
- contrôle immédiat de la validité des données saisies
- partage d'une même structure de données entre plusieurs traitements
- limitation de la redondance des données
- standardisation des traitements généraux (consultation, restitution sous forme de listes ou tableaux)
- gestion rationnelle des supports

À partir de 1970 : deuxième génération de SGBD à partir du *modèle relationnel*. Enrichissement et simplification des SGBD afin de faciliter l'accès aux données pour les utilisateurs. Commercialisation depuis 1982. ORACLE, INGRES, SYBASE, INFORMIX, DB2, RDB.

Début des années 80 : troisième génération de SGBD basées sur le *modèle objet*. ONTOS, ObjectStore, VERSANT, ORION, O₂.

2. Définitions et propriétés

Base de données (BD) : Collection de données interreliées, stockées ensemble pour servir une ou plusieurs applications, en parallèle, de façon optimale. Stockage des données indépendant des programmes d'utilisation. Approche commune pour incorporer de nouvelles données, pour modifier et retrouver les données actuelles.

Système de Gestion de Bases de Données (SGBD) : Ensemble des programmes assurant structuration, stockage, maintenance, mise à jour et recherche des données d'une base + interfaces nécessaires aux différentes formes d'utilisation de la base.

Propriétés

- Usage multiple des données
- Accès facile, rapide, protégé, souple, puissant
- Coût réduit de stockage, de mise à jour et de saisie
- Disponibilité, exactitude, cohérence et protection des données ; non redondance
- Évolution aisée et protection de l'investissement de programmation
- Indépendance des données et des programmes
- Conception *a priori*

3. Objectifs des SGBD

a) *Indépendance physique*

Un remaniement de l'organisation physique des données n'entraîne pas de modification dans les programmes d'application.

b) *Indépendance logique*

Un remaniement de l'organisation logique des données (ajout d'une nouvelle rubrique, ajout d'une nouvelle liaison...) n'entraîne pas de modifications dans les programmes d'application dont la "vision" logique n'a pas évolué.

c) *Manipulation des données par des langages non procéduraux*

Des utilisateurs non informaticiens doivent pouvoir manipuler simplement les données, c'est-à-dire les interroger et les mettre à jour sans préciser d'algorithme d'accès.

d) *Administration facilitée des données*

Un SGBD doit fournir des outils pour décrire les données, permettre leur suivi de ces structures et autoriser leur évolution. C'est la tâche des *administrateurs de données* : conception, création, maintenance, « arbitrage ».

e) *Efficacité des accès aux données*

Nécessité de garantir un bon *débit* (nombre de transactions exécutées par seconde) et un bon *temps de réponse* (temps d'attente moyen pour une requête). Partage des ressources (CPU, disques...) entre les utilisateurs en optimisant l'utilisation globale afin d'éviter, par exemple, qu'une requête courte d'un utilisateur attende la fin d'une requête longue d'un autre utilisateur.

f) *Redondance contrôlée des données*

Si redondance, volume de stockage plus important, opérations de mise à jour multiples, incohérences momentanées ou permanentes.

g) *Cohérence des données*

Ex. L'âge d'une personne doit être un entier positif. Un SGBD doit veiller à ce que les applications respectent cette règle lors des modifications de données. Une telle règle est appelée *contrainte d'intégrité*.

h) *Partage des données*

Diverses applications doivent pouvoir partager les données de la base dans le temps et simultanément, comme si elles étaient seules à les utiliser.

i) *Sécurité des données*

Les données doivent être protégées contre les accès non autorisés ou mal intentionnés. La sécurité des données doit aussi être assurée en cas de panne d'un programme ou du système, voire de la machine.

4. Fonctions des SGBD

a) *Description des données*

Aux niveaux externe, internes, conceptuels
Par les administrateurs
Grâce à un *Langage de Définition de Données* (LDD)

b) *Recherche de données*

c) *Mise à jour des données*

Insertion, modification, suppression
Grâce à un *Langage de Manipulation de Données* (LMD)

d) *Transformation des données*

Ex. Changement de format : date sur 2 chiffres → date sur 4 chiffres

e) *Contrôle de l'intégrité des données*

Exactitude des données stockées dans la base (respect des contraintes d'intégrité)

Maintien de la cohérence entre les copies multiples d'une même donnée

f) *Gestion de transactions et sécurité*

Gestion des transactions = s'assurer qu'un groupe de mises à jour est totalement exécuté ou pas du tout (*atomicité* des transactions)

Sécurité = personnalisation des accès à la base, mots de passe

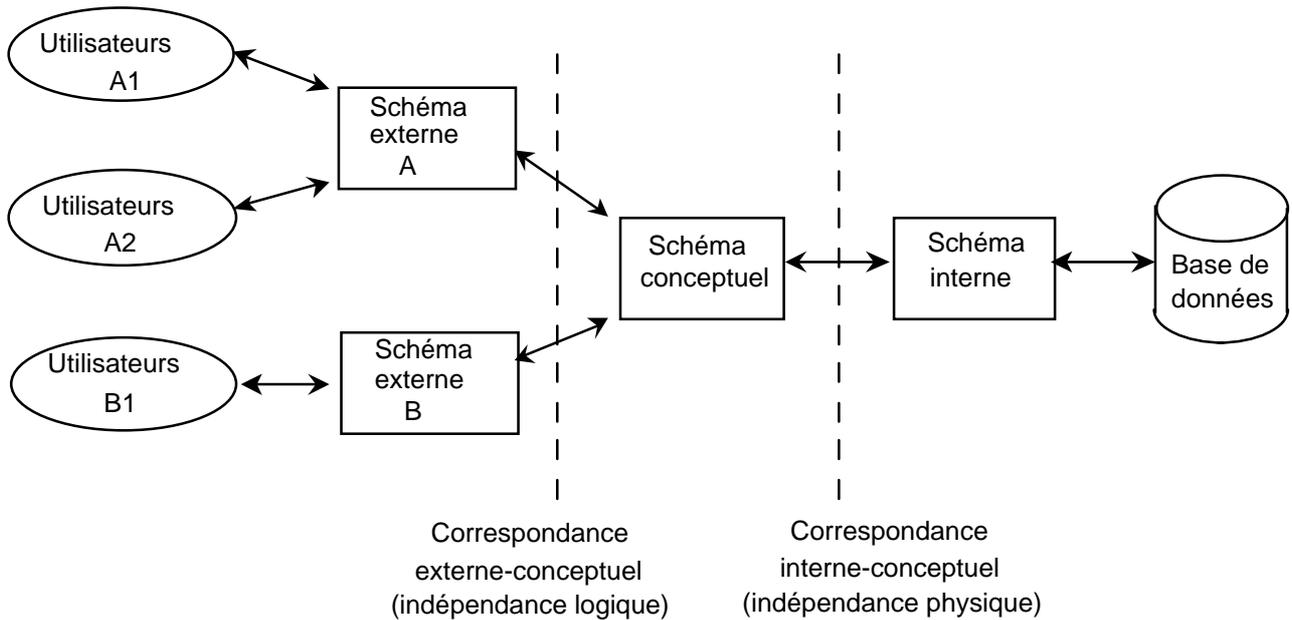
g) *Autres fonctions*

Ex. *Déclencheurs* (triggers)

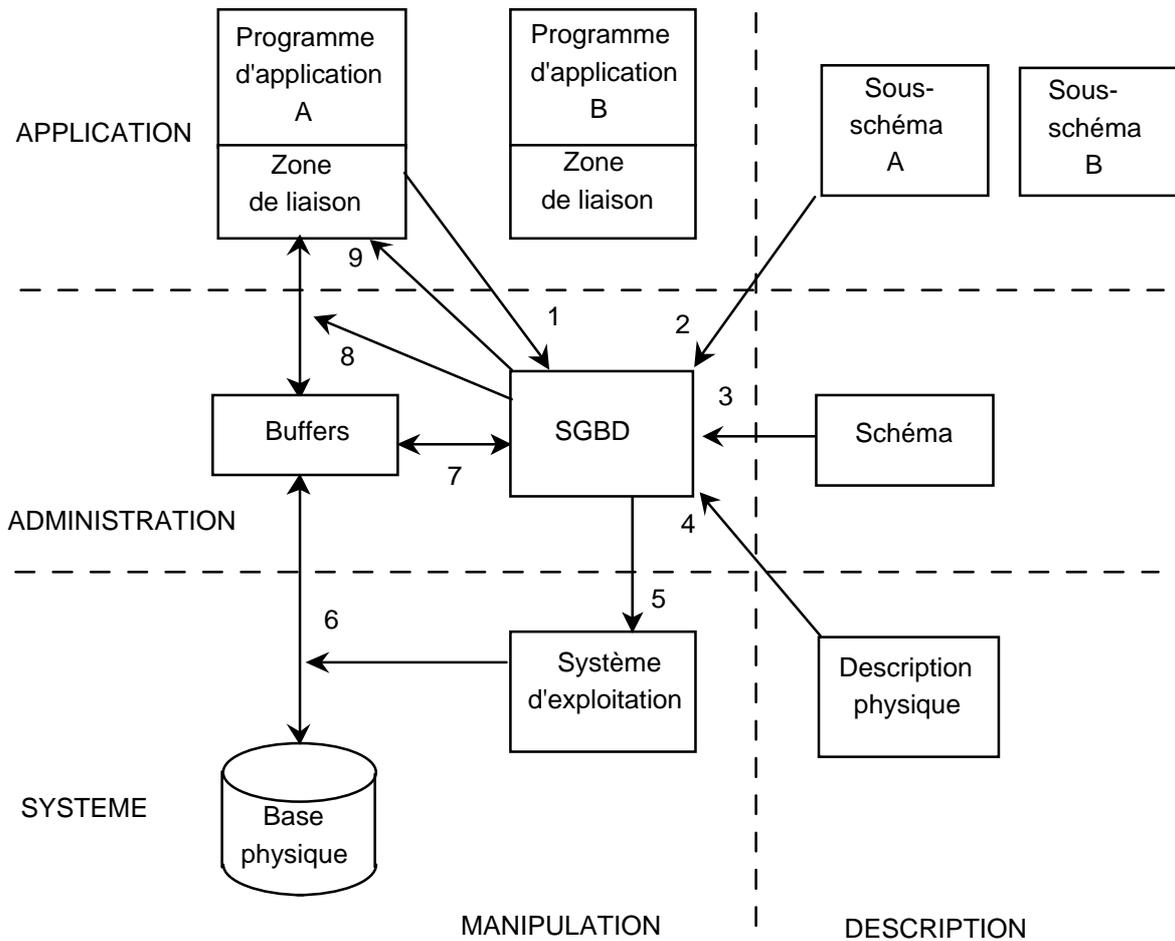
Conséquence Modification	Réécriture des autres programmes	Modification de la description globale	Modification de la structure physique
Ecriture d'un nouveau programme d'application avec types existants	NON	NON	NON
Modification d'un programme sans modification des types existants	NON	NON	NON
Ecriture d'un nouveau programme d'application avec nouveaux types	NON	OUI	OUI
Insertion ou suppression de nouvelles occurrences de types existants	NON	NON	NON
Création de nouvelles liaisons entre données	NON	OUI	OUI
Fusion avec une autre base de données	NON	OUI	OUI
Modification de la structure physique	NON	NON	OUI
Modification des techniques d'adressage	NON	NON	OUI
Changement du type de support physique	NON	NON	OUI
Changement du système d'exploitation	NON	NON	OUI
Changement de matériel	NON	NON	OUI

Conséquences de diverses modifications dans un système à base de BD

5. Fonctionnement d'un SGBD



Architecture ANSI/SPARC à trois niveaux d'une base de données



Fonctionnement d'un SGBD

- 1) Le programme d'application A émet une demande de lecture à l'intention du SGBD.
- 2) Le SGBD consulte le sous-schéma relatif à A pour obtenir la description logique de ses données.
- 3) Le SGBD consulte le schéma et détermine la structure logique des données à extraire.
- 4) Le SGBD examine la description physique de la base et détermine les enregistrements physiques à lire.
- 5) Le SGBD lance une commande au système d'exploitation pour provoquer la lecture de l'enregistrement physique.
- 6) Le système d'exploitation provoque le transfert de l'enregistrement entre la base physique et les buffers du SGBD.
- 7) Le SGBD, à partir du sous-schéma A, extrait les données à communiquer au programme d'application A.
- 8) Le SGBD provoque le transfert des données dans la zone de liaison de A.
- 9) Le SGBD retourne au programme d'application les informations d'état relatives à l'échange (en particulier les codes des erreurs éventuelles).

6. Conception d'une base de données

a) Analyse

Analyse du monde réel

Le monde réel est perçu comme un système abstrait.

Ce système abstrait se traduit par :

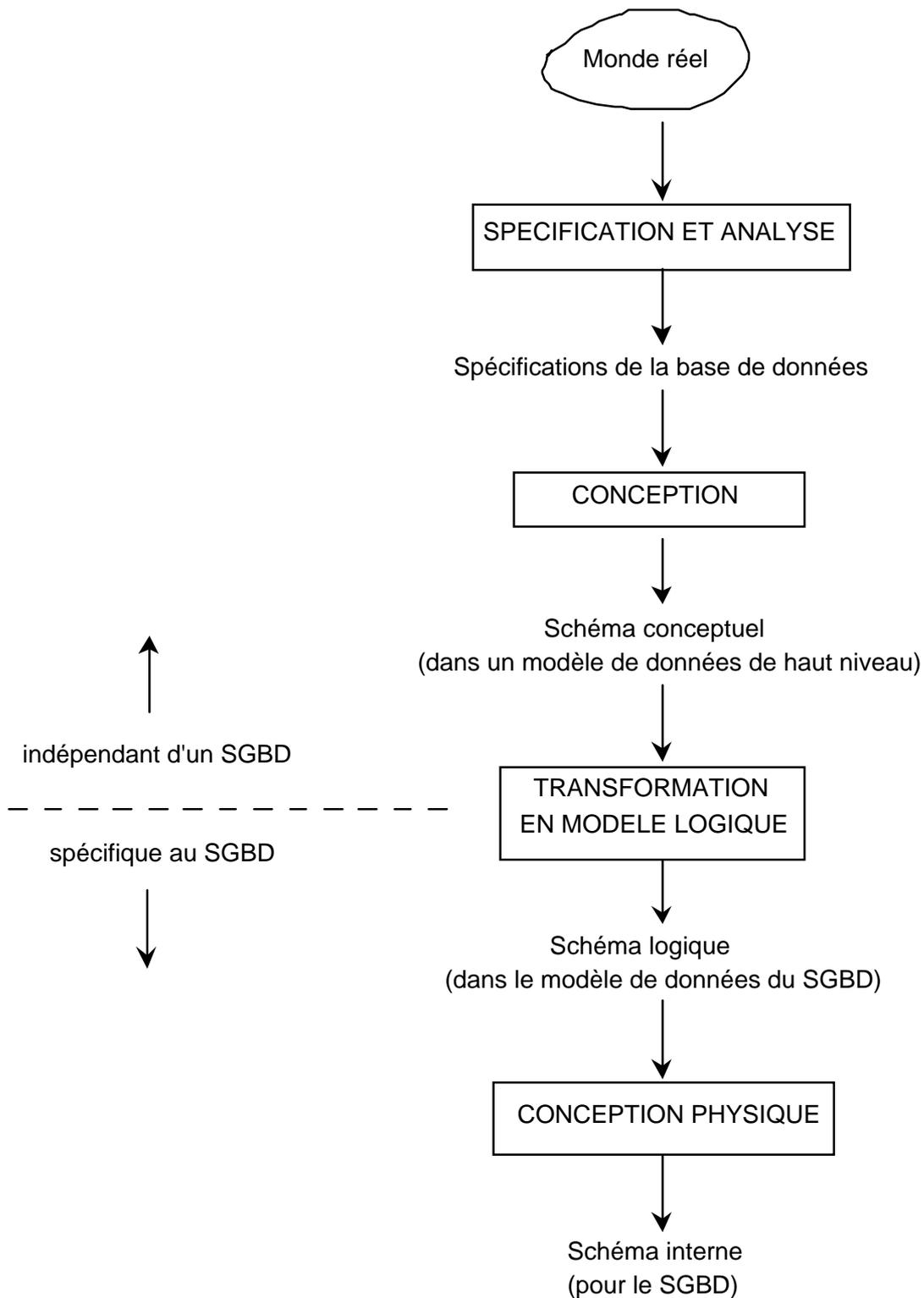
- des classes d'entités,
- des propriétés sur ces classes,
- des liaisons entre ces classes.

Le système abstrait est décrit par un schéma conceptuel.

Modélisation conceptuelle

Principes généraux à respecter :

- le schéma conceptuel doit être libre de toute considération non significative du système abstrait (organisation physique des données, aspects particuliers à un usager tels que des formats de messages...)



Processus de conception d'une base de données

- tous les aspects du système abstrait doivent être décrits dans le schéma conceptuel ; aucun d'eux ne doit intervenir ailleurs en particulier dans des programmes d'application indépendants du schéma conceptuel.

Caractéristiques :

- prend en compte les aspects statiques et dynamiques du système abstrait ;
- fournit un langage pour communiquer avec un système informatique et avec diverses catégories d'utilisateurs ;
- permet de prendre en compte des évolutions ;
- est indépendante de tout SGBD.

Modélisation logique

Traduit le modèle conceptuel dans le modèle du SGBD.

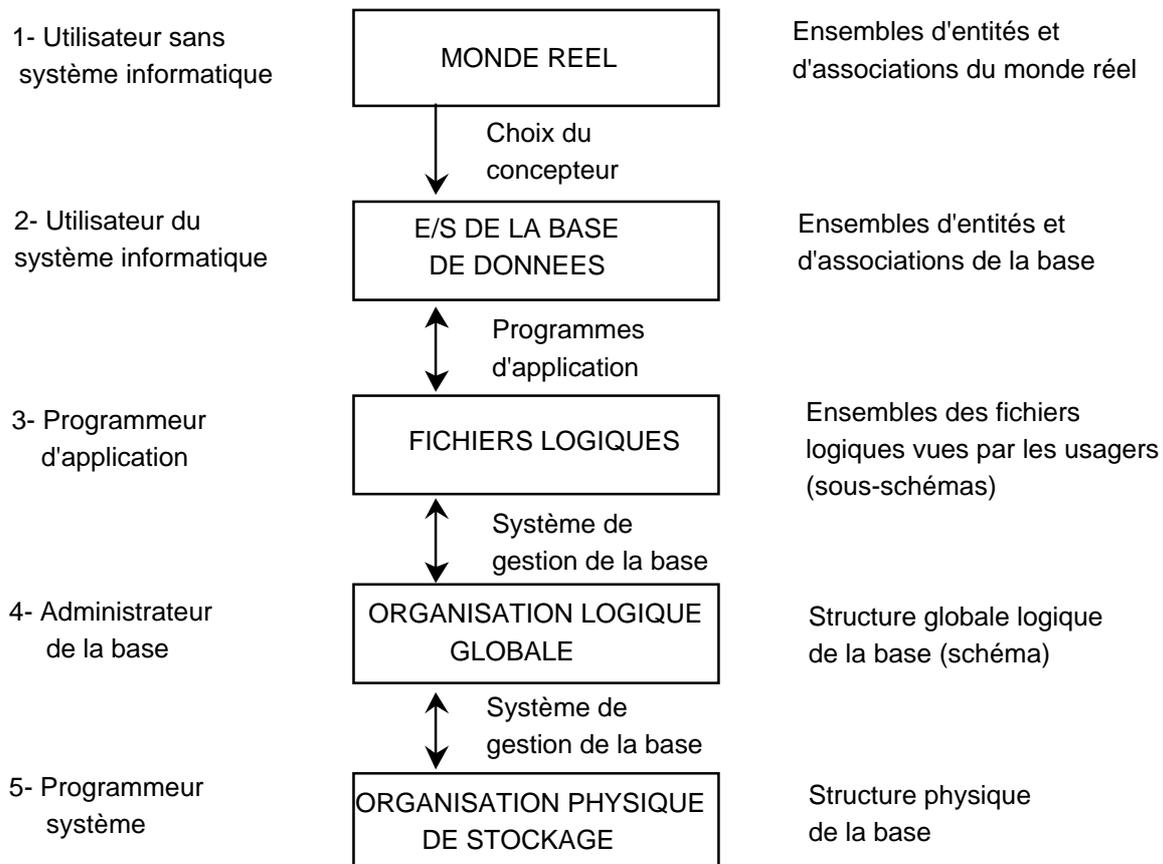
Il existe différents types de modèles logiques de SGBD : hiérarchiques, réseau, relationnels, orientés objet.

Certains modèles peuvent être spécifiques à un SGBD.

Modélisation physique

Structures de stockage internes

Organisation des fichiers



Vision des différents usagers d'une base de données

b) Modèles conceptuels de données

Concepts de base

Entité (ex. Voiture)

Attribut : propriété de l'entité (ex. Numéro d'immatriculation, Couleur)

Valeur de l'attribut (ex. 5255VX63, Blanche)

Association : liaison perçue entre entités

Type (ex. Couleur = chaîne de caractères)

Occurrence (deux voitures différentes sont deux occurrences de l'entité Voiture)

Caractéristiques des associations

Association *un à plusieurs* (en abrégé **1-N**) entre les deux types d'entités A et B :

- Chaque occurrence du type A est associée à zéro, une ou plusieurs occurrences du type B.
- Chaque occurrence du type B est associée à une occurrence du type A.

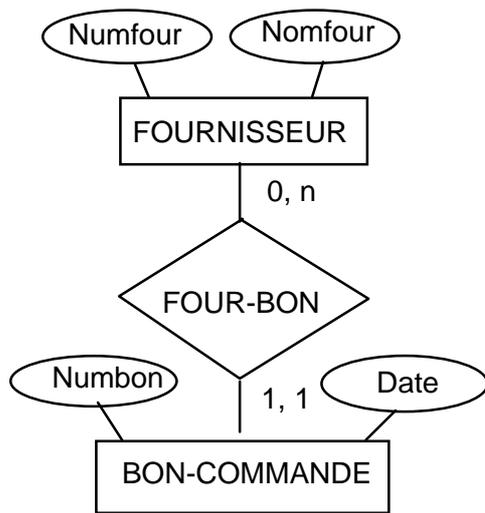
Exemple : association 1-N entre le type d'entités FOURNISSEUR et le type d'entités BON DE COMMANDE.

Association **1-1** : cas particulier de l'association 1-N.

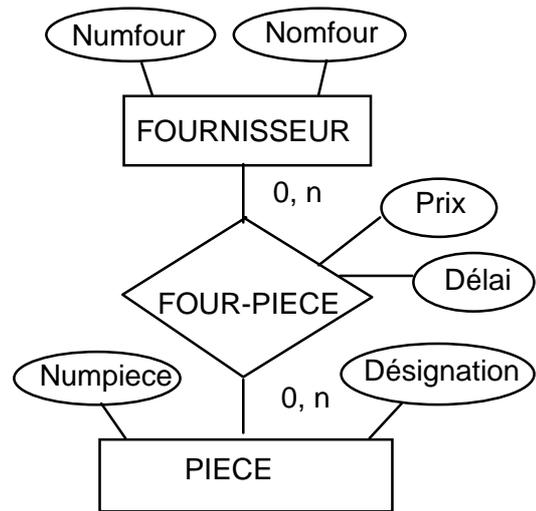
Association *plusieurs à plusieurs* (en abrégé **M-N**) entre les deux types d'entités A et B :

- Chaque occurrence du type A est associée à zéro, une ou plusieurs occurrences du type B.
- Chaque occurrence du type B est associée à zéro, une ou plusieurs occurrences du type A.

Exemple : association M-N entre le type d'entités FOURNISSEUR et le type d'entités PIÈCE.

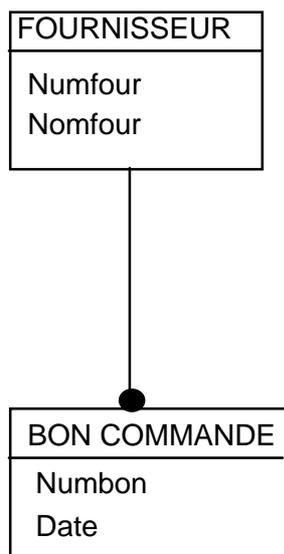


association 1-n

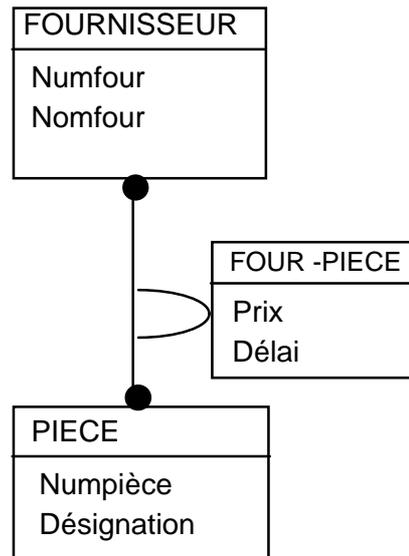


association m-n

Représentation graphique des associations 1-N et M-N (formalisme E/A)



association 1-n

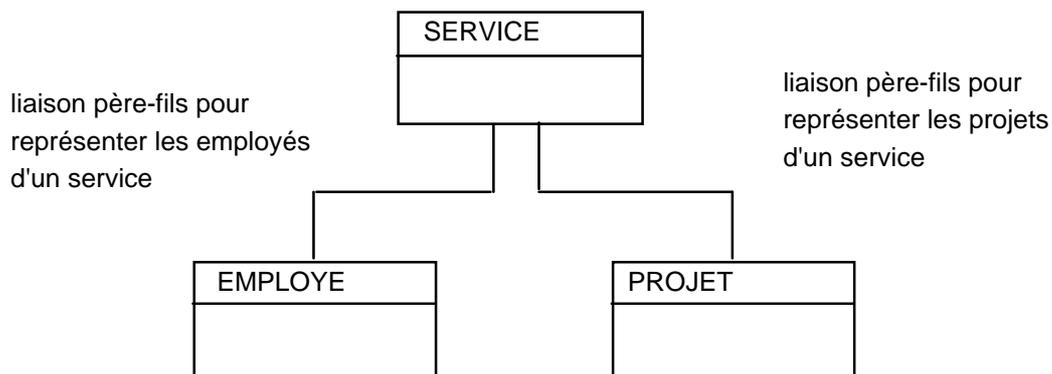


association m-n

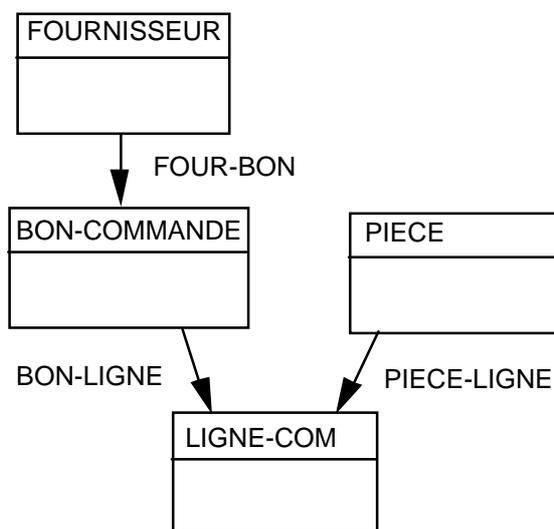
Représentation graphique des associations 1-N et M-N (formalisme OMT)

c) Modèles logiques de données

Ils varient selon le SGBD utilisé.



SGBD hiérarchiques



SGBD réseaux

FOURNISSEUR

<u>NUFOUR</u>	NOM	ADRESSE

BON-COMMANDE

<u>NUBON</u>	NUFOUR	DATE

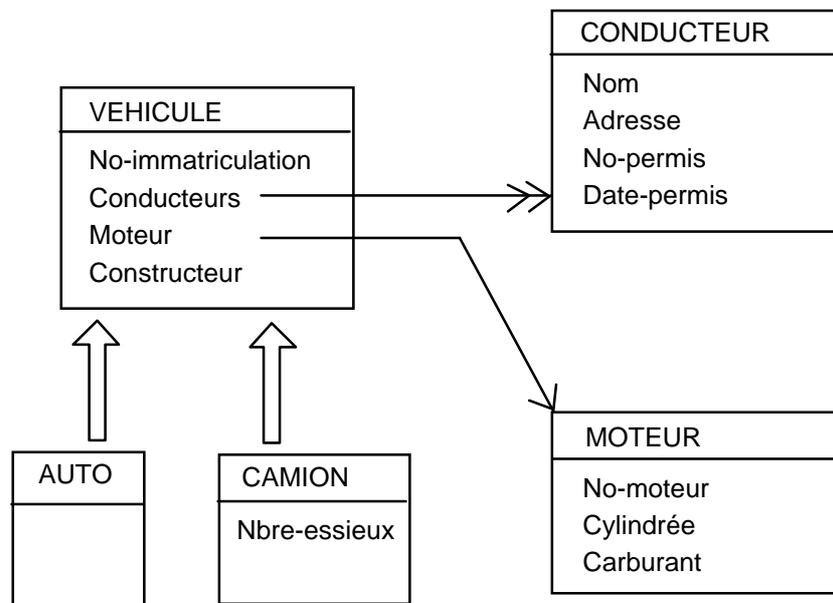
LIGNE-COMMANDE

<u>NUBON</u>	<u>NUPIECE</u>	DESIGNATION	QUANTITE

SGBD relationnels (représentation tabulaire)

FOURNISSEUR (NUFOUR, NOM, ADRESSE)
 BON-COMMANDE (NUBON, NUFOUR, DATE)
 LIGNE-COMMANDE (NUBON, NUPIECE, DESIGNATION, QUANTITE)

SGBD relationnels (représentation textuelle)



SGBD orientés objet

Exemple de base de données : Fabrication des pièces d'une entreprise

Deux types d'entités de base :

PIECE (NOP, DESIGNATION, COULEUR, POIDS)
 SERVICE (NOS, INTITULE, LOCALISATION)

Deux autres types d'entités :

ORDRE (QUANTITE)
 NOMENCLATURE (QUANTITE)

Associations entre :

SERVICE et ORDRE
 PIECE et ORDRE
 PIECE et NOMENCLATURE
 PIECE et NOMENCLATURE

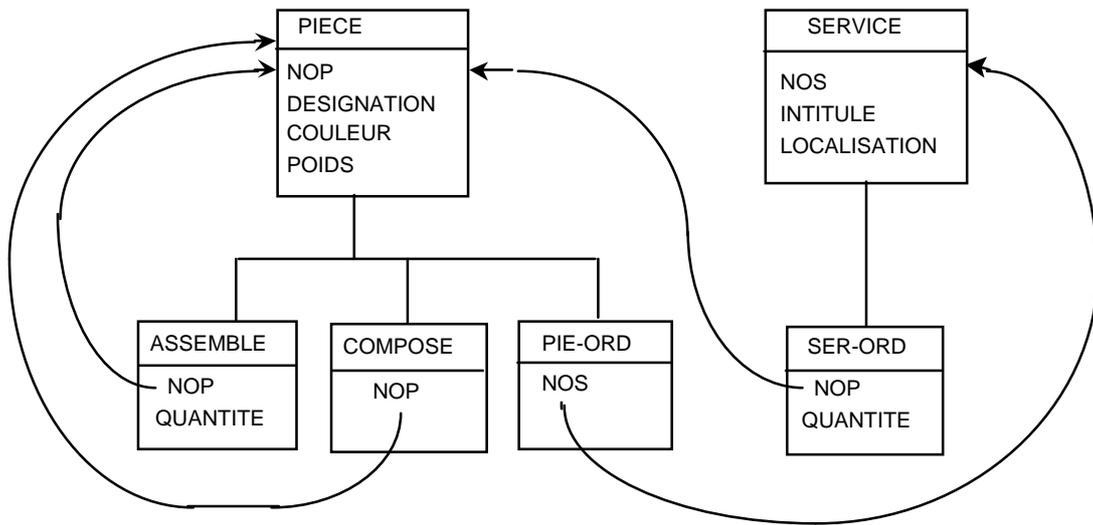


Schéma hiérarchique

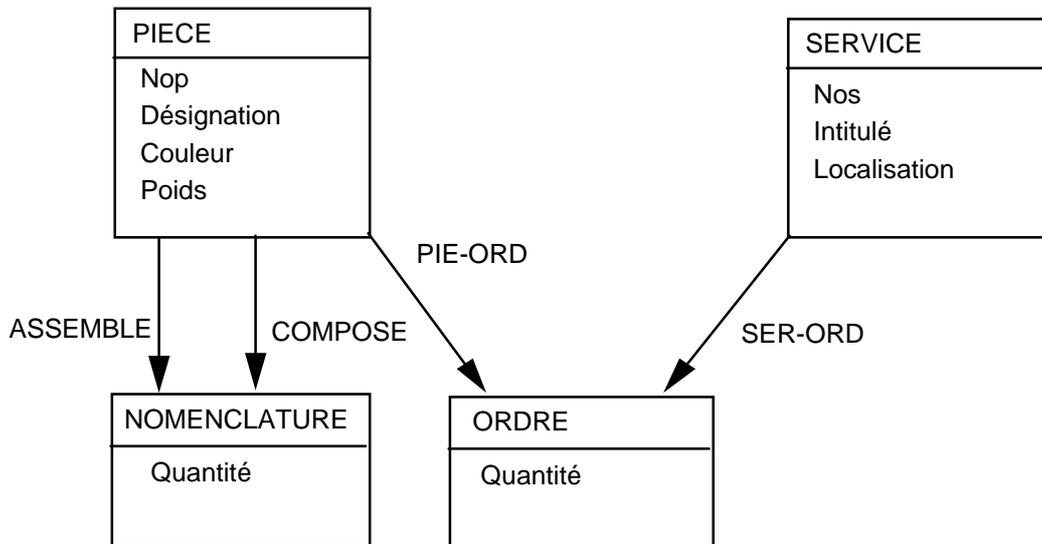


Schéma réseau

PIECE (NOP, DESIGNATION, COULEUR, POIDS)
 SERVICE (NOS, INTITULE, LOCALISATION)
 ORDRE (NOP, NOS, QUANTITE)
 NOMENCLATURE (NOPA, NOPC, QUANTITE)

Schéma relationnel

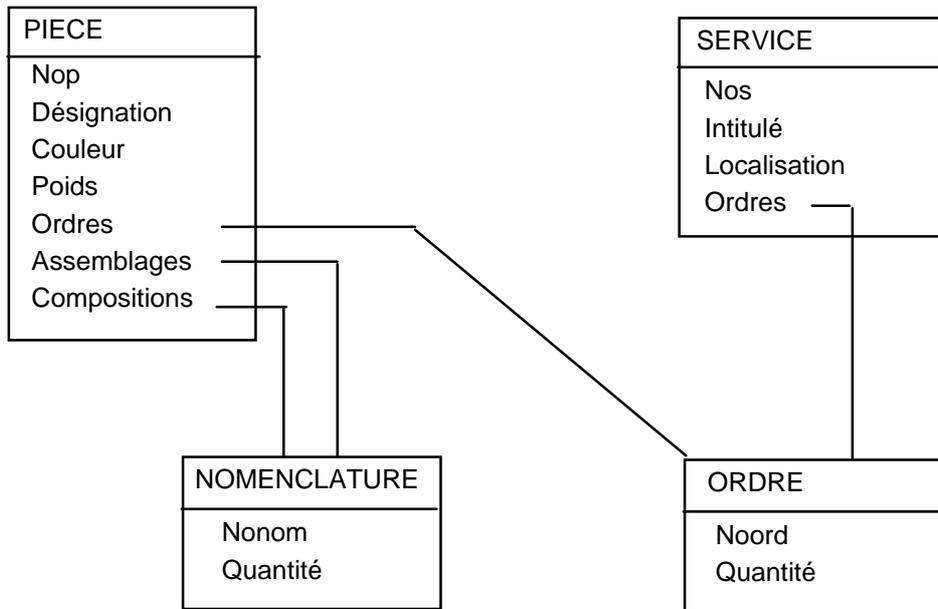


Schéma orienté objet