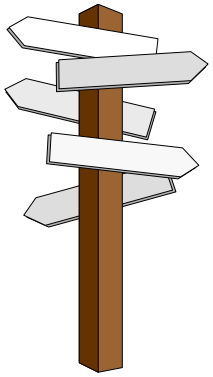
Cours n°1
Systèmes de Gestion de Données

Chantal Reynaud

Université Paris X - Nanterre UFR SEGMI

Licence MIAGE 2003-2004



Plan

I. Introduction aux systèmes de bases de données
II. L’architecture des bases de données
III. Une introduction aux bases de données relationnelles
IV. La conception d’un schéma relationnel

Partie I. Introduction aux systèmes de base de données

I. Le concept de Système de Base de Données

I.1. Les données
I.2. Le matériel
I.3. Le logiciel
I.4. Les utilisateurs

II. Le concept de base de données

III. Pourquoi utiliser une base de données ?

I. Un Système de Bases de Données

- Le logiciel qui permet à un utilisateur d ’exploiter une base de données (SGBD)

  - organise et conserve les données sur les supports périphériques
  - fournit les procédures de recherche et de sélection de ces mêmes données
  - cache les détails de l ’organisation et de la localisation des données sur disques

Les données

- Intégrées :
  - Une unification de fichiers distincts, toute redondance ayant été totalement ou partiellement éliminée
- Partagées :
  - Chaque BD peut être partagée par plusieurs utilisateurs. Chacun peut accéder à la même donnée en même temps.

Le matériel

- Composants :
  - Volumes de mémorisation secondaires
  - Le(s) processeur(s) et la mémoire principale

Le logiciel

- Couche entre la base de données physique et le système : le gestionnaire de BD (DB Manager), communément appelé SGBD.
- Le SGBD est *le composant le plus important* du système.
- Autres utilitaires : outils de développement d 'applications, d 'aide à la conception, rédacteurs de rapports, gestionnaires des transactions, etc.

Les utilisateurs

- Trois classes d'utilisateurs :
  - Les programmeurs d 'applications
  - Les utilisateurs finaux
  - L'administrateur de la base de données (DBA).

II. Le Concept de Base de Données

- Une collection de données *persistantes* utilisées par les systèmes d ’application d ’une entreprise
  - Une fois introduites dans la base, elles ne peuvent être enlevées que par une requête explicitement envoyée au SGBD et non par le seul effet de bord de l ’exécution d ’un programme.

III. Pourquoi utiliser une base de données ?

- Partage des données
- Redondance réduite
- Incohérence évitée
- Transactions gérées

- Intégrité assurée
- Sécurité appliquée
- Conflits de besoins équilibrés
- Normes appliquées

+ *Indépendance des données*

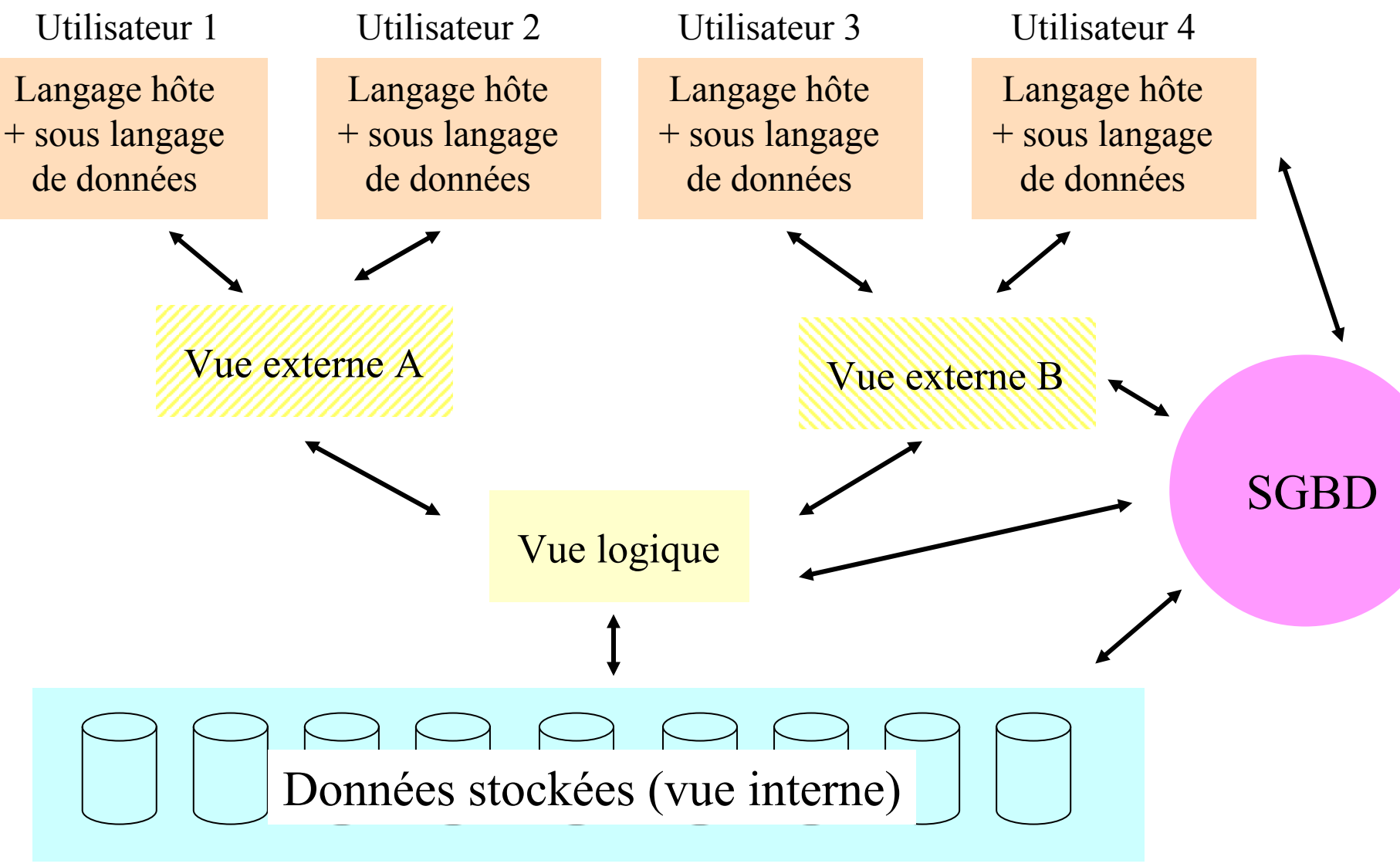
Partie II. L ’architecture des bases de données

I. L ’architecture ANSI/SPARC à trois niveaux

I.1. Le niveau externe
I.2. Le niveau conceptuel
I.3. Le niveau interne

II. L ’architecture client/serveur

I. Schéma de l 'architecture ANSI/SPARC



Illustration

- Niveau **externe** : (COBOL)

```
01 Empl.
        02 empno Pic X(6).
        02 deptno Pic X(4).
```

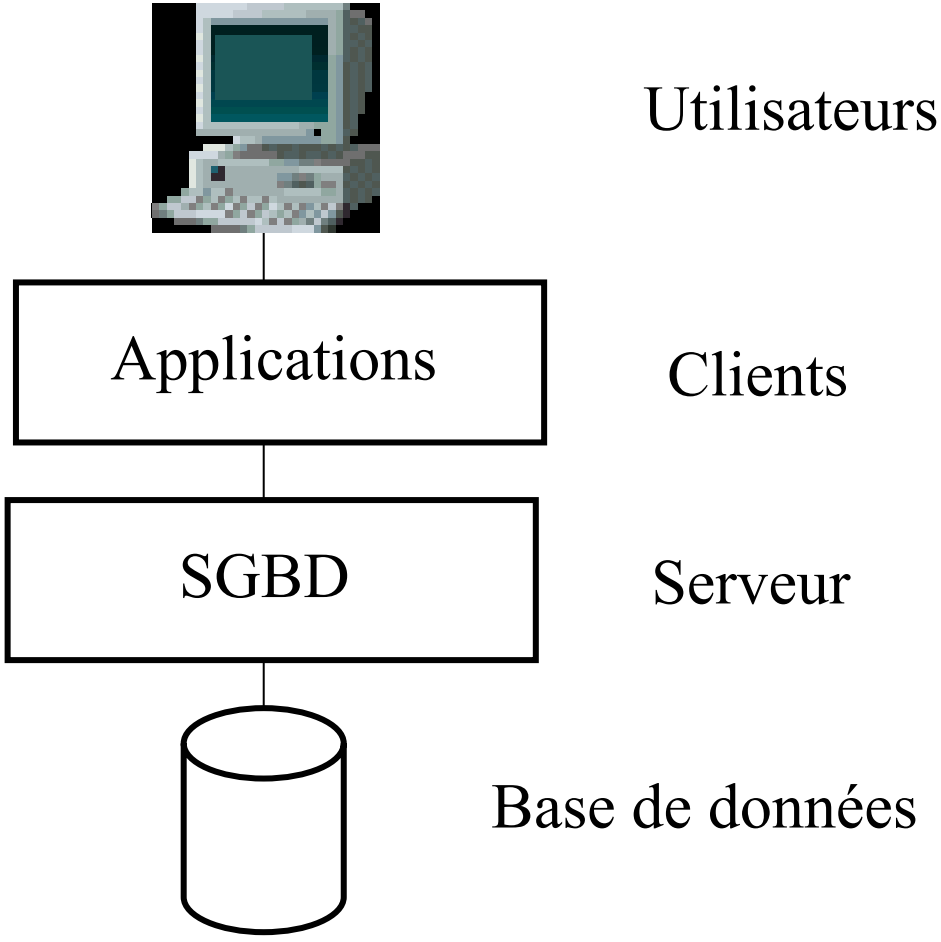
- Niveau **conceptuel** :

```
EMPLOYE
        NUMERO_EMPLOYE CHAINE(6)
        NUMERO_DEPARTEMENT CHAINE(4)
        SALAIRE ENTIER(5)
```

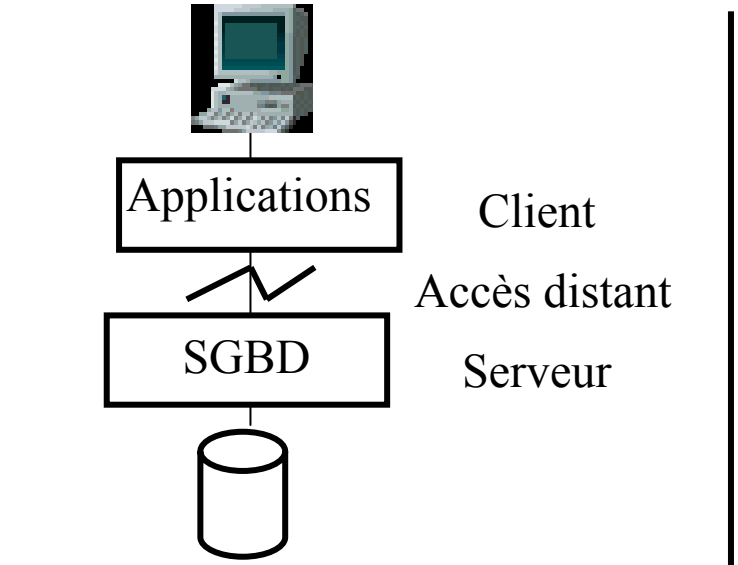
- Niveau **interne** :

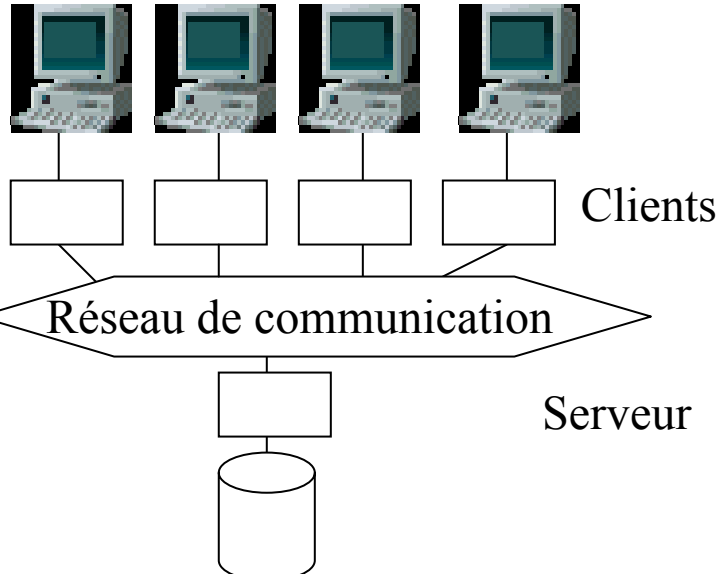
```
Stored_Emp Length = 20
        PREFIX TYPE=BYTE(6), OFFSET=0
        EMP# TYPE=BYTE(6), OFFSET=6, INDEX=EMPX
        DEPT# TYPE=BYTE(4), OFFSET=12
        PAY TYPE=FULLWORD, OFFSET=16
```

II. Architecture client/serveur

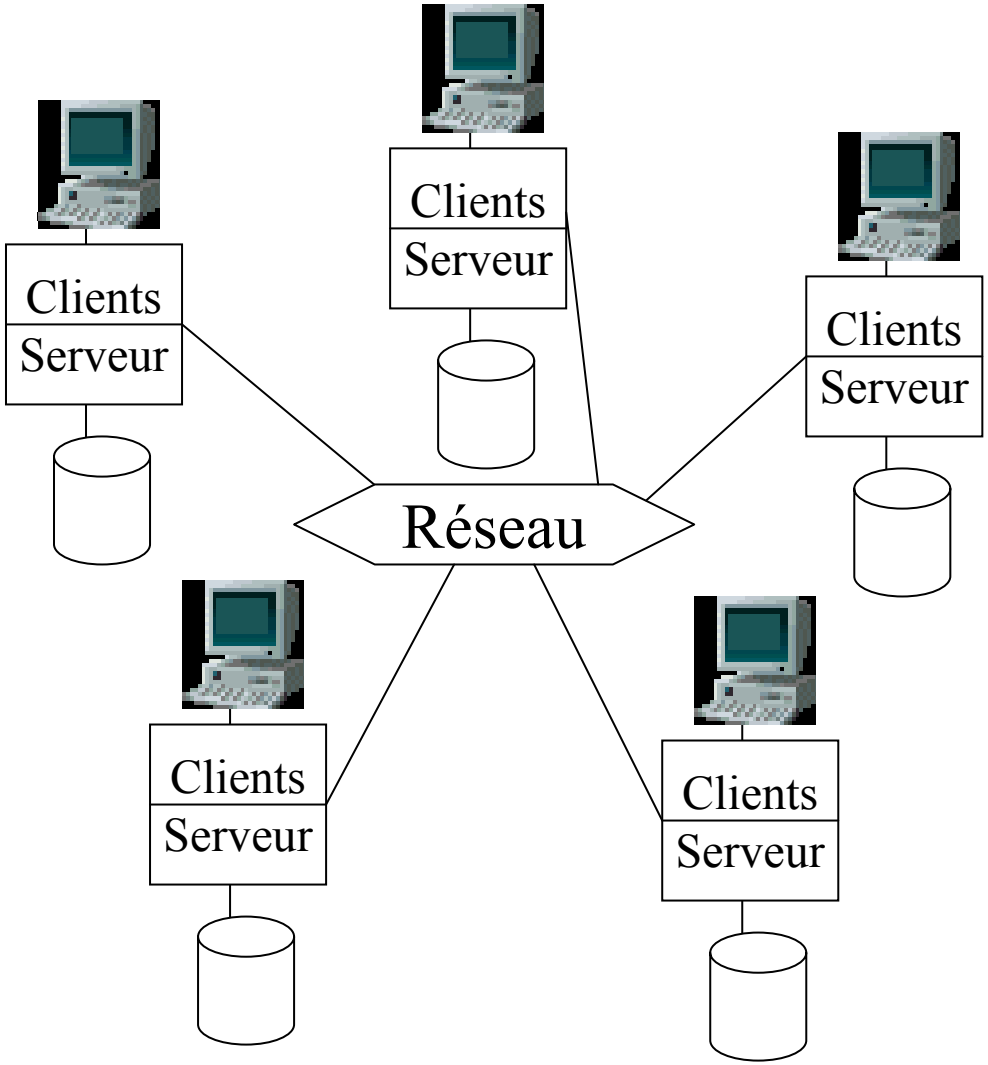


Architecture client/serveur





Traitement distribué



Partie III. Une introduction aux bases de données relationnelles

I. Le modèle relationnel

I.1. Les concepts de relation - attribut - domaine
I.2. Schéma d ’une relation - Schéma d ’une BDR

II. Le passage d ’un schéma entité-association à un schéma relationnel

II.1. Règles de passage
II.2. Cas particuliers

I. Le modèle relationnel

Les concepts de relations - attribut - domaine

Une **relation** sur les **attributs** A1, A2, ..., An de **domaines** respectifs D1, D2, ..., Dn est un sous-ensemble du produit cartésien des domaines D1, D2, ..., Dn, soit un ensemble de n-uplets de l'ensemble produit D1 x D2 ... x Dn.

Ex : la relation r ci-dessous sur les attributs Num, Nom, Adresse
r = {(2140, Dupont, Paris), (1128, Durant, Nanterre),(3456, Dubois, La Défense)}

Les **attributs** sont des variables prenant leurs valeurs dans des **domaines**

Ex : Num prend ses valeurs dans D1 avec D1 = {Entiers}

Les noms des attributs d'une relation doivent être tous différents.
Les noms des domaines peuvent être identiques.

Ex : Nom et Adresse prennent leurs valeurs dans D2 avec D2 = {Chaîne de caractères}

La représentation des relations sous forme de tableaux

Attributs de r

Num	**Nom**	**Adresse**
2140	Dupont	Paris
1128	Durant	Nanterre
3456	Dubois	La Défense

- Deux lignes ne sont jamais identiques
- L'ordre des lignes est indifférent
- Chaque case du tableau ne contient qu'une seule valeur d'un type déterminé
- L'ordre des colonnes est indifférent (cette condition est satisfaite à partir du moment où chaque colonne possède un nom).

Schéma d ’une relation - Schéma d ’une BDR

On appelle **schéma de la relation** r la liste des attributs de la relation r avec, pour chaque attribut, son domaine. On dit aussi que r est une relation de schéma R

Ex : soit R le schéma de la relation r présentée précédemment
R = (Num : D1, Nom : D2, Adresse : D2)
avec D1 = {entiers} et D2 = {chaîne de caractères}

Souvent, on sous-entend le domaine et on note :
R = (Num, Nom, Adresse)

Un schéma de base de données relationnelle B est un ensemble de schémas de relation R1, R2, ..., Rp

Une base de données b sur B est un ensemble de relations r1, r2, ..., rp de schémas respectifs R1, R2, ..., Rp. Une base de données relationnelle peut donc être vue comme un ensemble de tables.

II. Le passage d ’un schéma entité-association à un schéma relationnel

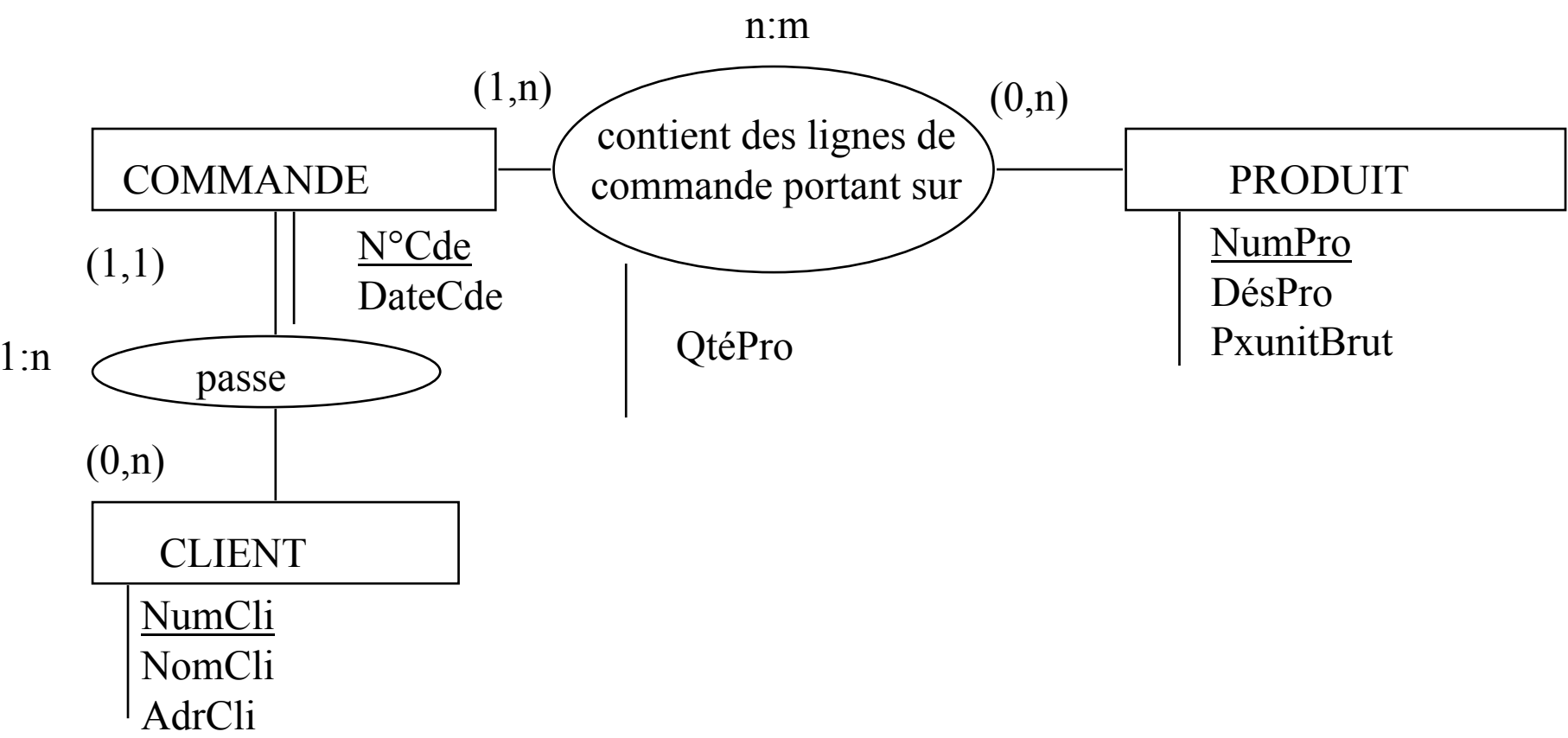
Règles de passage

1. A chaque **classe d’entités** est associé un schéma de relation contenant tous les attributs de l’entité

2. Pour traduire une classe d’**associations binaires** entre deux classes d’entités E et E’, **telle que la cardinalité maximale du couple (E,A) est égale à 1**, on ajoute dans le schéma de relation qui traduit E un identifiant de la classe d’entités E’ et les attributs de l’association A.

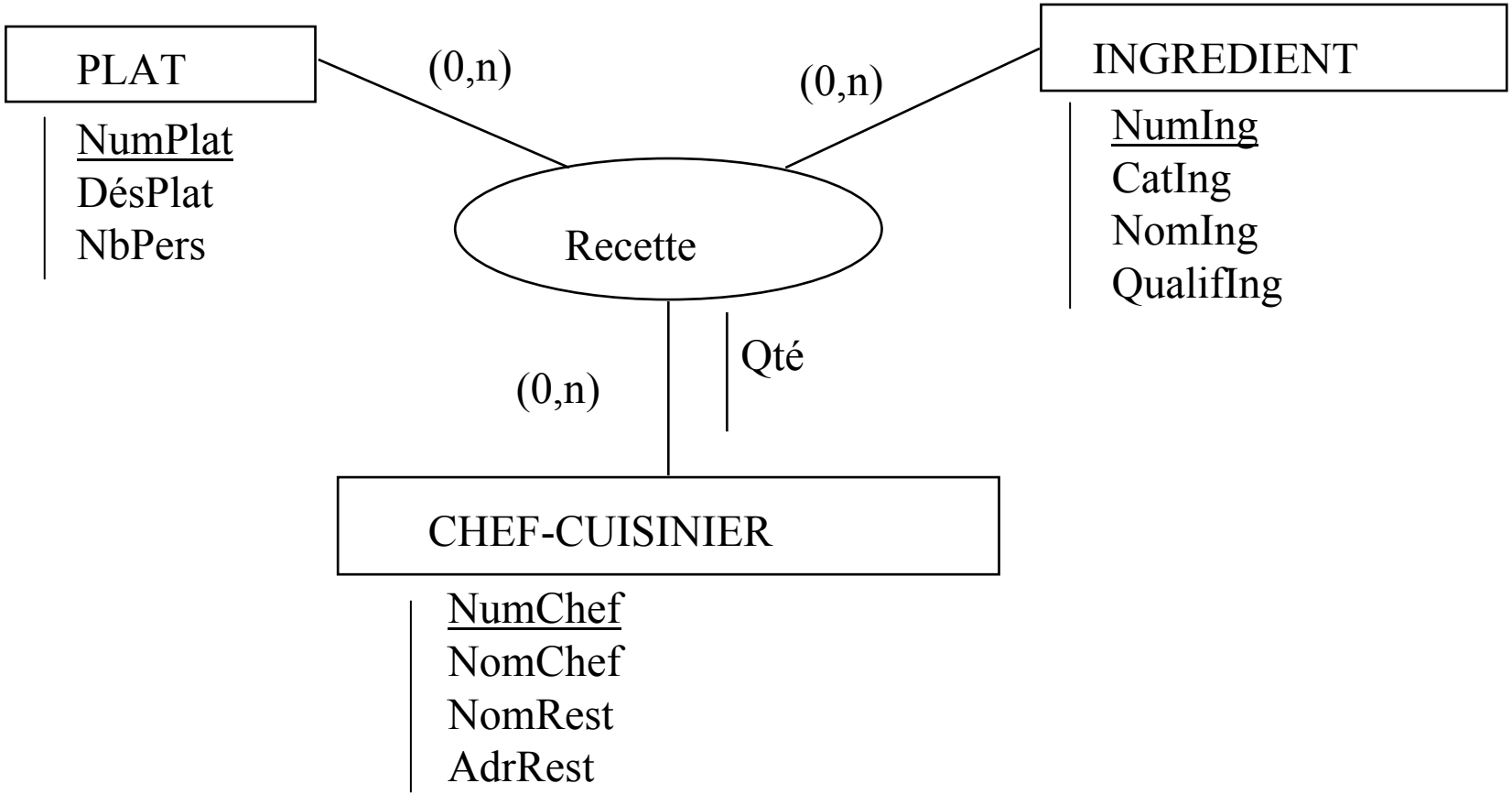
3. Pour traduire une classe d’**associations binaires de type n:m ou n-aires (n>2)**, on crée un nouveau schéma de relation contenant un identifiant de chacune des classes d’entités participant à l’association A et tous les attributs de l’association A.

Exemple



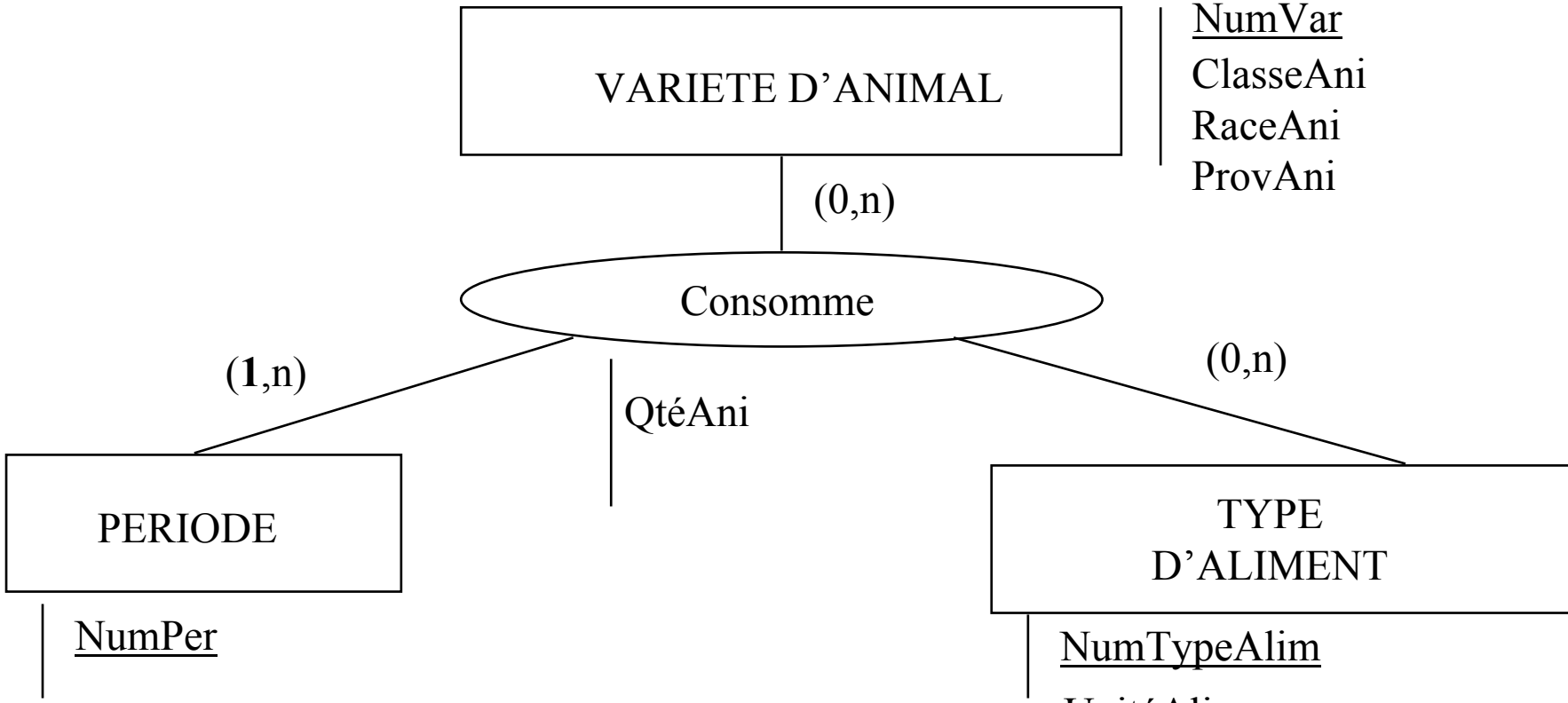
CLIENT = (NumCli, NomCli, AdrCli)
PRODUIT = (NumPro, DésPro, PxUnitBrut)
COMMANDE = (N°Cde, DateCde, NumCli)
LIGNECOMMANDE = (N°Cde, NumPro, QtéPro)

Exemple



PLAT = (NumPlat, DésPlat, NbPers)
INGREDIENT = (NumIng, CatIng, NomIng, QualifIng)
CHEF-CUISINIER = (NumChef, NomChef, NomRest, AdrRest)
RECETTE = (NumPlat, NumIng, NumChef, Qté)

Cas particulier (1)

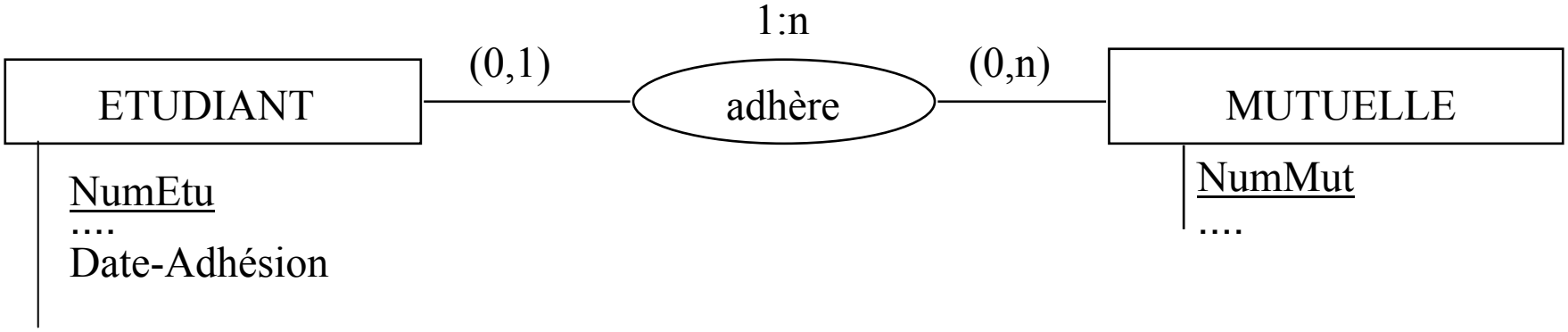


VARIETE = (NumVar, ClasseAni, RaceAni, ProvAni)
TYPEALIMENT = (NumTypeAlim, UnitéAlim)
CONSOMME = (NumVar, NumPér, NumTypeAlim, QtéAni)

Inutile de conserver la relation PERIODE = (NumPer) car les périodes s'obtiennent par la relation CONSOMME du fait de la cardinalité min. à 1

Cas particulier (2)

Dans le cas particulier de cardinalités (0,1) on pourra déroger à la règle 2 pour éviter d'avoir trop de valeurs "nulles" dans le schéma traduisant E.



S'il y a très peu d'étudiants qui adhèrent à une mutuelle , on pourra avoir la table suivante :

ADHERE = (<u>NumEtu, NumMut</u>, Date-Adhésion)

Cas des généralisations avec héritage

Solutions possibles

Une spécialisation d 'une classe C en plusieurs classes C1, C2, …, Cn peut être traduite par :

1) répétition de la clé de la table représentant C au niveau de chacune des tables représentant C1, C2, …, Cn.

2) répétition des attributs représentant C au niveau de chacune des tables représentant C1, …, Cn et par transformation de C en une vue dérivée de C1, …, Cn.

3) une table unique comportant la traduction de C complétée avec C1, …, Cn, les tables correspondant à C1, …, Cn étant des vues dérivées de C par projection sur les attributs pertinents.

Partie IV. La conception d ’un schéma relationnel

I. Les dépendances fonctionnelles (dfs)

I.1. Définitions
I.2. Propriétés
I.3. Dépendances triviales et non triviales
I.4. Fermeture d’un ensemble de dfs
I.5. Fermeture d’un ensemble d’attributs - notion de clé
I.6. Couverture minimale d’un ensemble de dfs

Définitions

- Soient A et B des attributs, A détermine fonctionnellement B ($A \rightarrow B$) si chaque valeur prise par A **détermine au plus une valeur** pour B.

- De façon plus générale, soient A1, …, Ap, B1, …, Bq des attributs, A1, …, Ap déterminent fonctionnellement B1, …, Bq si la donnée d 'une valeur pour chacun des attributs A1, …, Ap **détermine au plus une valeur** pour chacun des attributs B1, …, Bq.

- Dans le contexte du modèle relationnel : Soit R un schéma de relation et X et Y deux sous-ensembles quelconques de l 'ensemble des attributs de R. Nous dirons que Y est en dépendance fonctionnelle avec X ssi, pour n 'importe quelle valeur licite de R, à chaque valeur de X dans R **correspond exactement une seule valeur** de Y dans R.

Propriétés

- **Règles d 'Armstrong**

<u>Réflexivité :</u> $Y \subseteq X \Rightarrow X \rightarrow Y$

<u>Augmentation :</u> $X \rightarrow Y \Rightarrow XZ \rightarrow YZ$

<u>Transitivité :</u> $X \rightarrow Y$ et $Y \rightarrow Z \Rightarrow X \rightarrow Z$

Dfs triviales: impossible qu'elles ne soient pas satisfaites!

- **Autres propriétés**

<u>Auto-détermination :</u> $X \rightarrow X$

<u>Décomposition :</u> $X \rightarrow YZ \Rightarrow X \rightarrow Y$ et $X \rightarrow Z$

<u>Union :</u> $X \rightarrow Y$ et $X \rightarrow Z \Rightarrow X \rightarrow YZ$

<u>Pseudo-Transitivité :</u> $X \rightarrow Y$ et $YZ \rightarrow W \Rightarrow XZ \rightarrow W$

<u>Composition :</u> $X \rightarrow Y$ et $Z \rightarrow W \Rightarrow XZ \rightarrow YW$

Fermeture

- On appelle **fermeture d 'un ensemble de dfs** F (notée F+) l 'ensemble des dfs impliquées par F.
- La **fermeture d 'un ensemble d 'attributs** X relativement à un ensemble de dfs F est l 'ensemble des attributs A pour lesquels la df $X \rightarrow A$ est dans la fermeture de F. Nous noterons cet ensemble X+.

Etant donnés R = (A1, …, An), F un ensemble de dfs sur R, X un ensemble d 'attributs de R, on dit que **X est une clé de R** muni de F si une des conditions suivantes (équivalentes) est satisfaite : (1) $X \rightarrow R$ appartient à F+

(2) $F \models X \rightarrow R$

(3) Toute relation r sur R qui satisfait F vérifie $X \rightarrow R$

Clé - Clé minimale

- Vérifier que K est une **clé** du schéma de relation R muni de l 'ensemble des dfs F revient à montrer que tous les attributs de R appartiennent à K+.
- X est une **clé minimale** (de R muni de F) ssi X est une clé et si tout sous-ensemble de X, différent de X, n 'en est pas une.

Propriétés

- Tout attribut qui ne figure pas dans le membre droit d 'une df non triviale de F doit appartenir à toute clé de R.
- Si l'ensemble des attributs de R qui ne figure pas en membre droit d 'une df non triviale de F est une clé, alors F possède une clé minimale unique formée de l'ensemble de ces attributs.
- Un schéma de relation muni d 'une seule df possède une clé minimale unique.

Couverture minimale

- On appelle **couverture minimale** de F un ensemble G de df tel que :

  (1) G+ = F+
  (2) Tout membre droit d'une df est réduit à un seul attribut
  (3) Pour aucune df $X \rightarrow A$ de G, on n 'a $G - \{X \rightarrow A\} \models G$
  (4) Pour aucune df $X \rightarrow A$ de G, on n'a $G \models (G - \{X \rightarrow A\}) \cup Y \rightarrow A$ avec $Y \subset X$.

<u>Rq :</u> en général, il n 'y a pas unicité de la couverture minimale