

Concepts Fondamentaux des Bases de Données

<http://deptinfo.unice.fr/~grin/mescours/linfo/bd>

L3 Sciences & Technologies
Mentions Informatique & Miage

Richard Grin
Philippe Lahire
Pierre Crescenzo

12 séances de Cours
lundi 10h15/12h15

12 séances de Tps

Contrôle des connaissances : Epreuves de 2h
Voir le site (en cours de mise à jour)

Bases de Données

Introduction

Philippe Lahire

à partir du cours d'Isabelle Mirbel

L3-Informatique

Année 07/08

■ Base de Données

- ❑ Ensemble **structuré** de données enregistrées en mémoire secondaire et accessibles de façon sélective par plusieurs utilisateurs simultanément
- ❑ Stocke de **grands volumes** de données / longue période
- ❑ **Contraintes** : sécurité, rapidité et facilité

■ Système de Gestion de Bases de Données

- ❑ Logiciel permettant d'**interagir efficacement** avec la base de données qui stocke de grands volumes de données

Quelques définitions (2)

■ Requête (Query)

- **Question** sur les données exprimée dans un langage proche du langage naturel

■ Transaction

- **Série d'opérations** d'accès aux données d'un utilisateur en bloquant l'accès à ces données si besoin
- Effets non visibles tant que la transaction n'est pas validée

Quelques définitions (3)

■ Principaux composants

- Gestionnaire de fichier
- Gestionnaire de transaction
- Gestionnaire de requêtes

Quelques définitions (4)

■ Les langages de requêtes

- Algèbre relationnelle
- Calcul relationnel à variable n-uplet
QUEL
- Calcul relationnel à variable domaine
QBE
- Structured Query Language

■ Par capacité

- Mainframes ou clusters de mainframe
DB2, ...
- Machine base de données
Teradata, ...
- Serveurs Unix & Windows
Oracle, Informix, DB2, Sybase, SQL server, ...
- Personnels
MS Access, ...

■ Créer des BD

- ☐ Décrire les données
- ☐ Définir leur schéma (structure logique)

➡ *Langage de définition des données*

■ Manipuler efficacement les données

➡ *Langage de manipulation des données*

■ Consulter efficacement les données

➡ *Langage d'interrogation des données*

■ Assurer la sécurité des données

- ☐ Confidentialité (autorisations, types d'accès, ...)
- ☐ Fiabilité des données (sauvegardes, reprise sur panne, ...)

■ Gérer les accès concurrents

- ☐ Problèmes de lectures/écritures simultanées
- ☐ Verrouillage des données (inter blocage, délai d'attente, ...)

- Centralisation des données
 - Non-redondance
 - Intégrité
 - Sécurité accrue
- Langages / outils puissants de manipulations
 - Développement rapide d'applications
- Langages non-procéduraux simples (ex: SQL, QBE)
 - Décrit le quoi et non le comment
 - Interrogation directe & réponse rapide pour les requêtes non prévues
- Indépendance des traitements par rapport aux données
 - Facilité de maintenance
 - Facilité d'évolution des traitements

■ Indépendance des données

- Applications isolées des changements de structure & du mode de stockage des données
- **Indépendance logique des données**: protection des changements de structure des données au niveau logique
- **Indépendance physique des données**: protection des changements de structure des données au niveau physique

■ Contrôle de concurrence

- Exécution concurrente de programmes essentielle pour SGBD

Accès disque lents & fréquents

➔ Nécessité pour l'UC d'exécuter des programmes en parallèle

- Exécution partielle des actions des différents programmes peut entraîner des incohérences

➔ Les SGBD assurent que la concurrence soit réalisée sans problème

- Concurrence transparente pour l'utilisateur

- Exécution d'un programme au-dessus d'un SGBD
 - Notion de **transaction**: séquence atomique d'actions sur une base de données
 - Chaque transaction est censée laisser la base de données dans un **état cohérent** après l'avoir prise dans un **état cohérent**
 - ➔ Importance des **contraintes d'intégrité** (règles inviolables) qui définissent la cohérence
 - ➔ **Préservation** de la cohérence de la base de données lors d'une transaction est **aussi** de la responsabilité de l'**utilisateur** !

■ Principe de l'ordonnancement des transactions

- Les SGBD assurent une exécution **concurrente** qui soit équivalente à une exécution **en série** des mêmes transactions
- **Moyens**: Verrouillage ou marquage des ensembles d'information manipulés par les transactions

■ Verrouillage strict

- Demande de verrou par la transaction au SGBD avant chaque action de lecture ou écriture.
 - Attente du verrou pour réaliser la transaction
 - Verrou relâché en fin de transaction
- ➡ Problème de deadlock

- Moyens (suite)

- Marquage

- Marquage affecté à chaque transaction
- Distinction des actions d'écriture & de lecture
- *Plus complexe mais évite les deadlock*

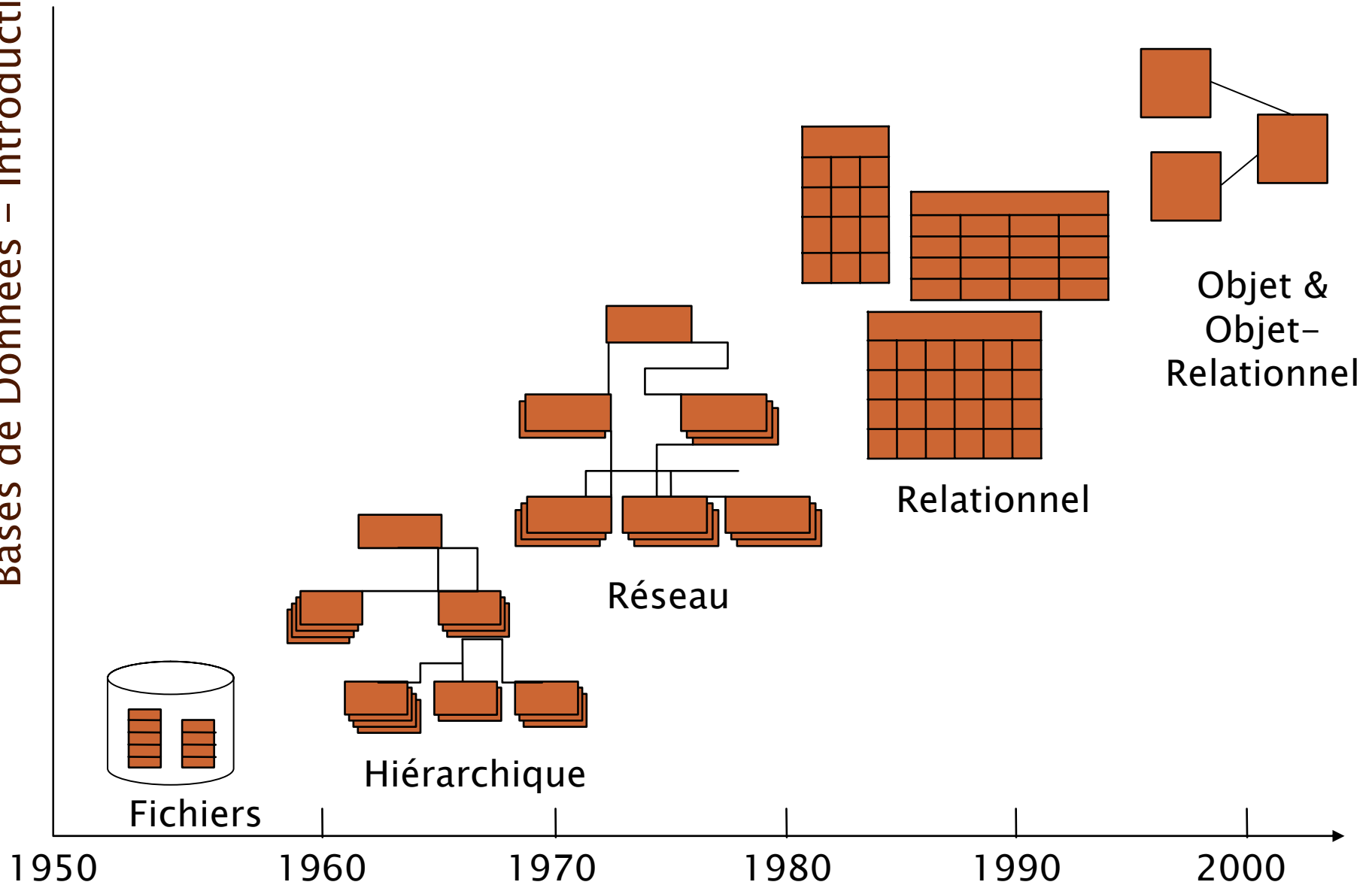
Pourquoi les SGBD (6) ?

■ Sûreté des traitements

- Les SGBD assurent la cohérence des données même en cas de crash
- Utilisation d'un journal de log de toutes les actions élémentaires de mise-à-jour réalisées par le SGBD
 - **Avant** la réalisation d'un changement, l'action est tracée dans le fichier de log
 - **Après** un crash, l'effet des transactions non abouties est annulé à l'aide du fichier de log

Évolution des bases de données

Bases de Données – Introduction



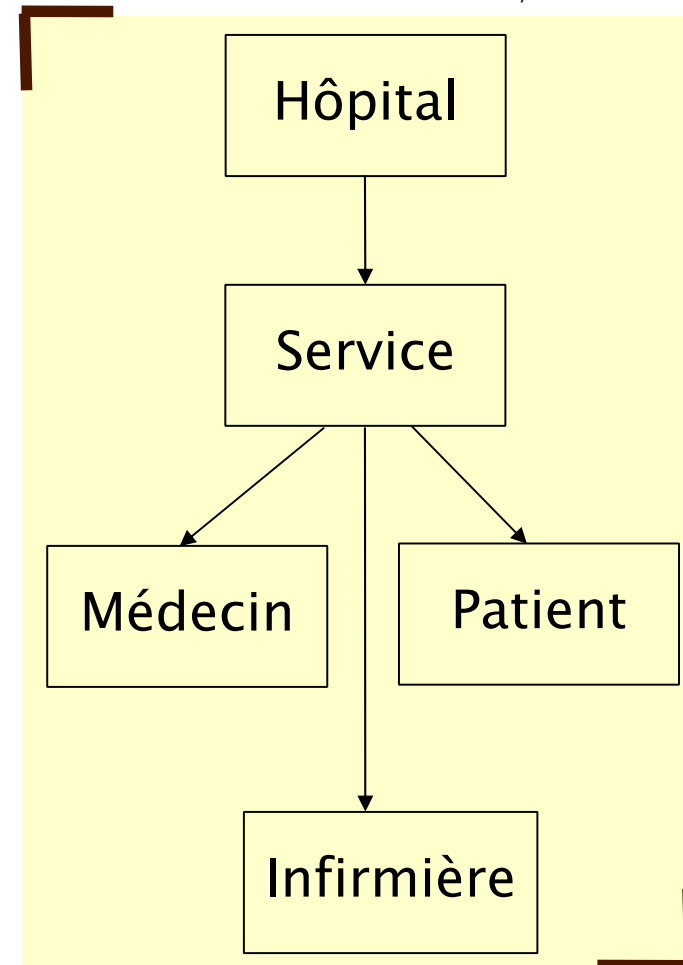
SGBD hiérarchiques & réseaux

■ Structure en graphe

- Articles (nœuds) reliés par des pointeurs (arcs)
- Hiérarchiques :
structure arborescente
- Réseaux :
structure en graphe quelconque

■ **Pointeurs** nécessaires pour déterminer le chemin d'accès aux données

- A indiquer dans les programmes
- Non indépendant de la structure logique de la BD
- Langage de manipulation complexe



SGBD relationnels

- Basés sur la **théorie mathématique** des relations
- Représentation des données sous forme de **tables** (relations)
- Presque tous les SGBD actuels

Patient

Nom	Adresse	Téléphone
Dupont Jean	Rue Barbe, 69000 Lyon	04 91 60 55 49
Durand Paul	Rue Minant, 06010 Nice	04 93 63 80 81
...

Occupation

Code	Nom	Arrivée	Départ
PE	Durand Paul	02/03/03	04/03/03
PE	Schmidt Pierre	17/02/03	
...

Service

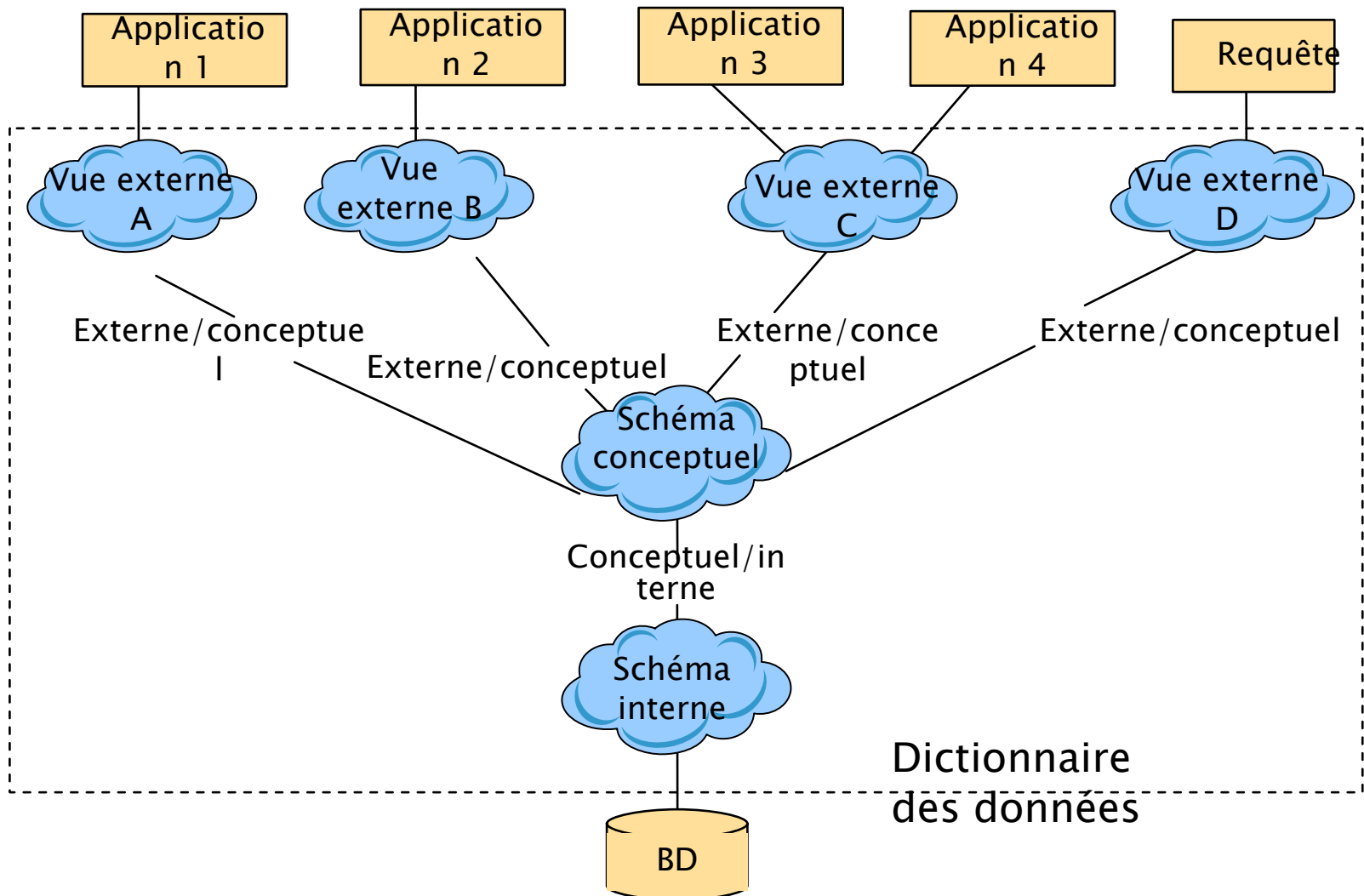
Code	Libellé	Bâtiment
PE	Pédiatrie	A
REA	Réanimation	D
...

SGBD objet & objet-relationnels (1)

- Notion de classe pour définir la structure des tuples
 - **Données** : instances de classe
 - **Fonctions** de manipulation spécifiques aux classes (enregistrées)
- Notion d'**héritage** entre les classes
- Permet de stocker et manipuler simplement des **types complexes** de données (listes, images, sons, vidéos, etc.)

■ SGBD objet & objet–relationnels (2)

- N'ont pas détrôné les SGBD relationnels
 - Évolution vers les **SGBD objet–relationnels** (SGBD–OR)
 - Norme **SQL3** (SGL–OR)



Niveaux de description d'une BD (1)

■ Schéma/vue **externe**

- Vision à partir d'une requête vers la BD
- Peut être partagé par plusieurs programmes / utilisateurs

■ Schéma **conceptuel**

- Vision exhaustive des données du domaine métier
- Description des contraintes sur les données
- Description de l'organisation logique des données

■ Schéma **interne**

- Structures et méthodes de stockage des données
- Méthode d'accès aux données

Niveaux de description d'une BD (2)

■ Schéma **conceptuel**

Client(Id, Nom, Prénom, Profession)
Compte(Idc, Type, solde)
Opération(Ido, Type, Date, Montant, Idc)

■ Schéma/vue **externe**

Cumul-mensuel(Idc, Nom, Prénom, Mois,
Total-crédit, Total-débit)

■ Schéma **interne**

Relations stockées dans des fichiers non
séquentiels

Niveaux de description d'une BD (3)

■ Changements au niveau externe

- Créer une nouvelle vue externe
- Modifier une vue externe existante
- ➔ Modification des programmes utilisant la/les vues

■ Changements au niveau conceptuel

- Ajout/Modification/Suppression d'attributs
- Ajout/Modification/Suppression de tables
- ➔ Modification de certaines vues
Pas de modification des programmes

Niveaux de description d'une BD (4)

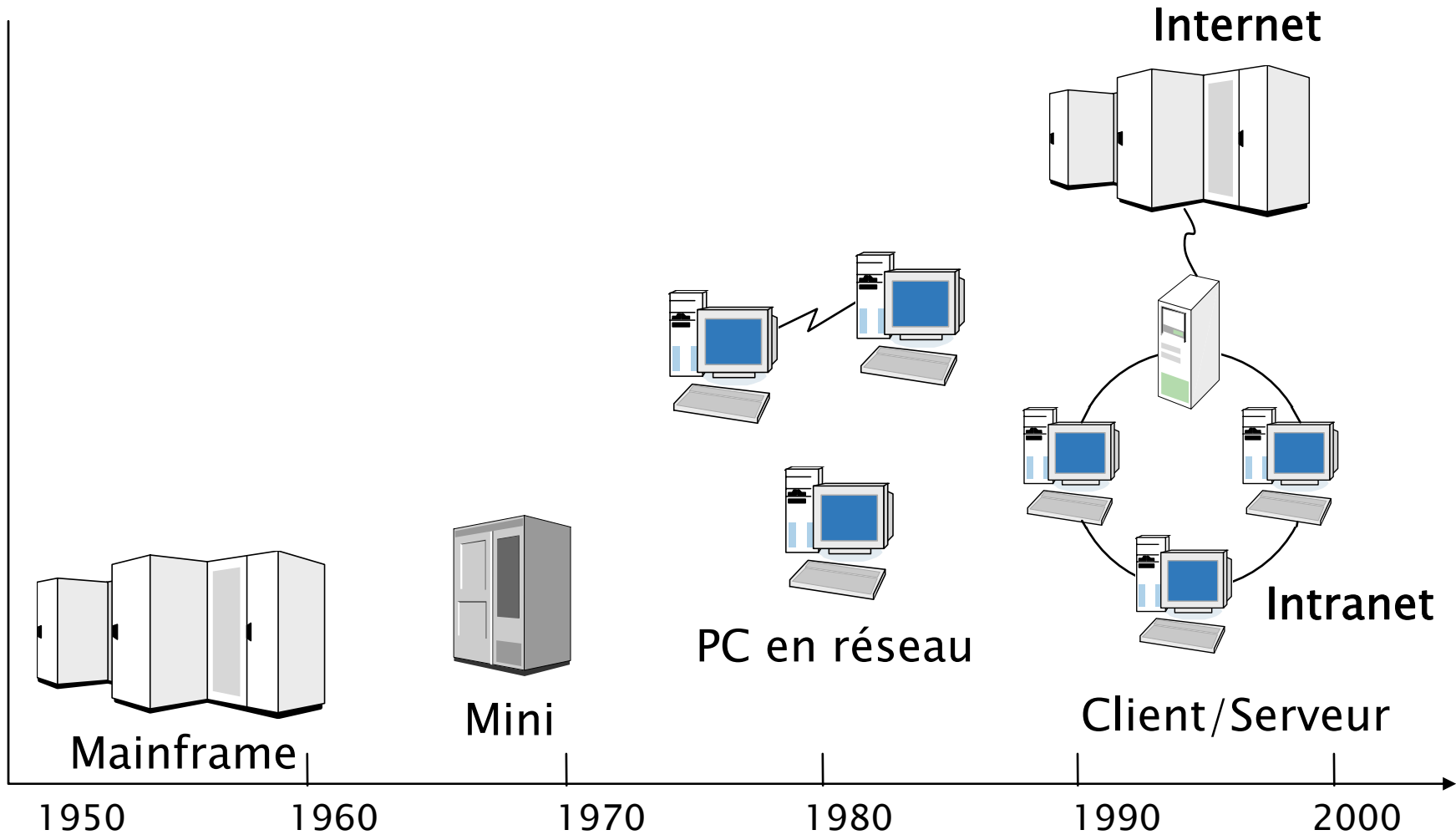
■ Changements au niveau interne

- Pour des raisons de performance
- Création/suppression d'index
- Organisation & structure des fichiers

➔ Pas de modification des vues
Pas de modification des programmes

Bases de Données – Introduction

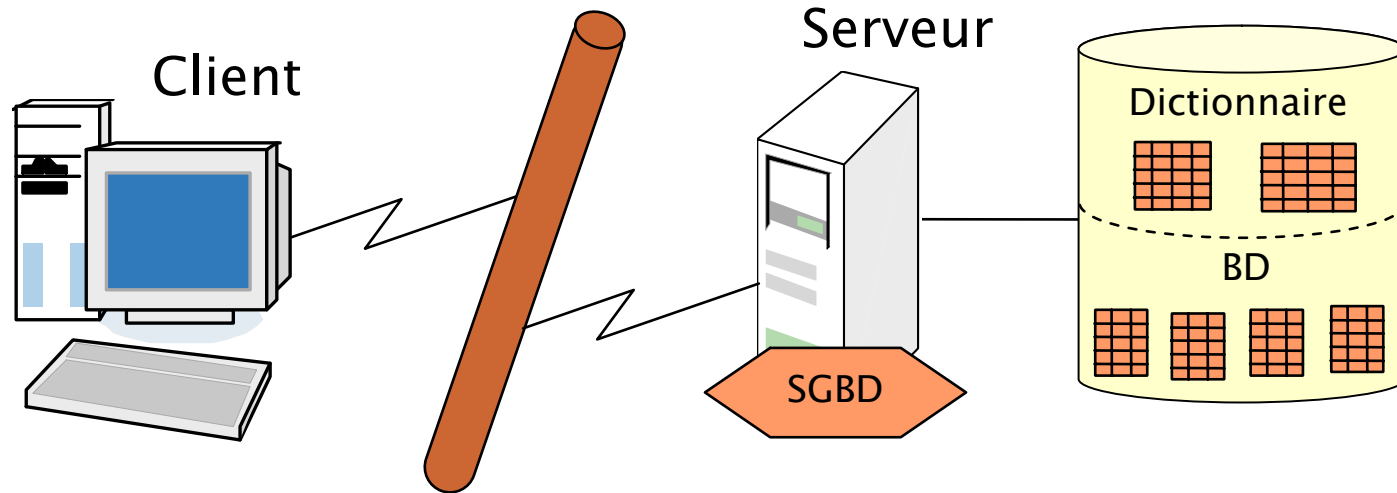
L'évolution technologique



L'architecture client/serveur

- Le moteur et les outils sont répartis sur le réseau
 - Le moteur du SGBD tourne sur un serveur
 - Les outils et applicatifs tournent sur des stations clients
- L'architecture Client/Serveur permet les systèmes ouverts
 - Le moteur peut s'interfacer avec des outils multiples
 - Tend à augmenter l'importance des standards (ex : SQL)
- L'architecture C/S est possible grâce aux technologies actuelles
 - Réseaux
 - Standardisation des réseaux et de SQL (en principe)
 - Multiplicité des logiciels clients et serveurs
- L'architecture Internet est comme un C/S sur plusieurs étages
 - L'interface client se fait à l'aide d'un browser (navigateur)

Traitement distribué en Client/Serveur



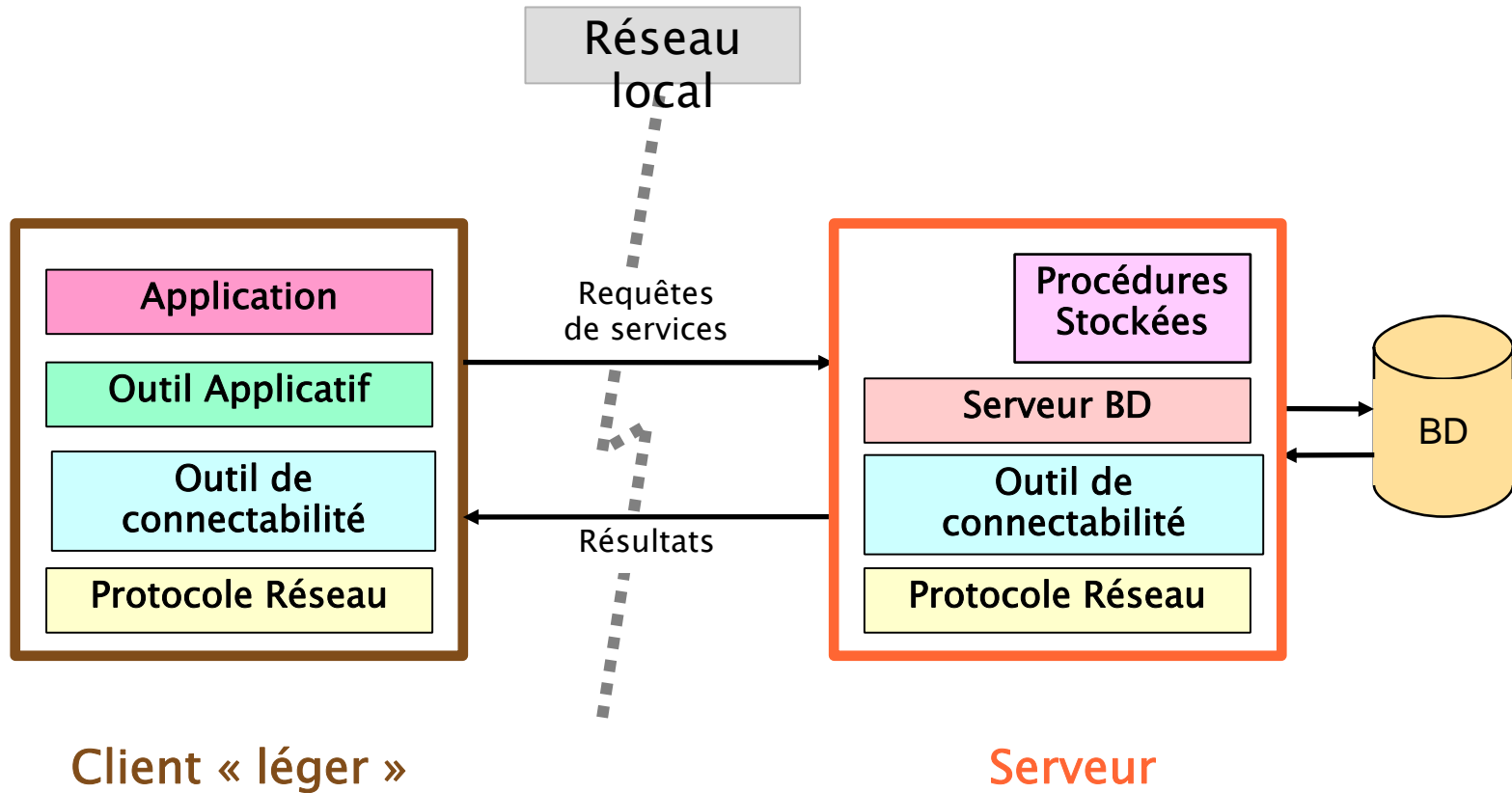
L'application

- ☐ Demande de données
- ☐ Traitement de données
- ☐ Présentation des données

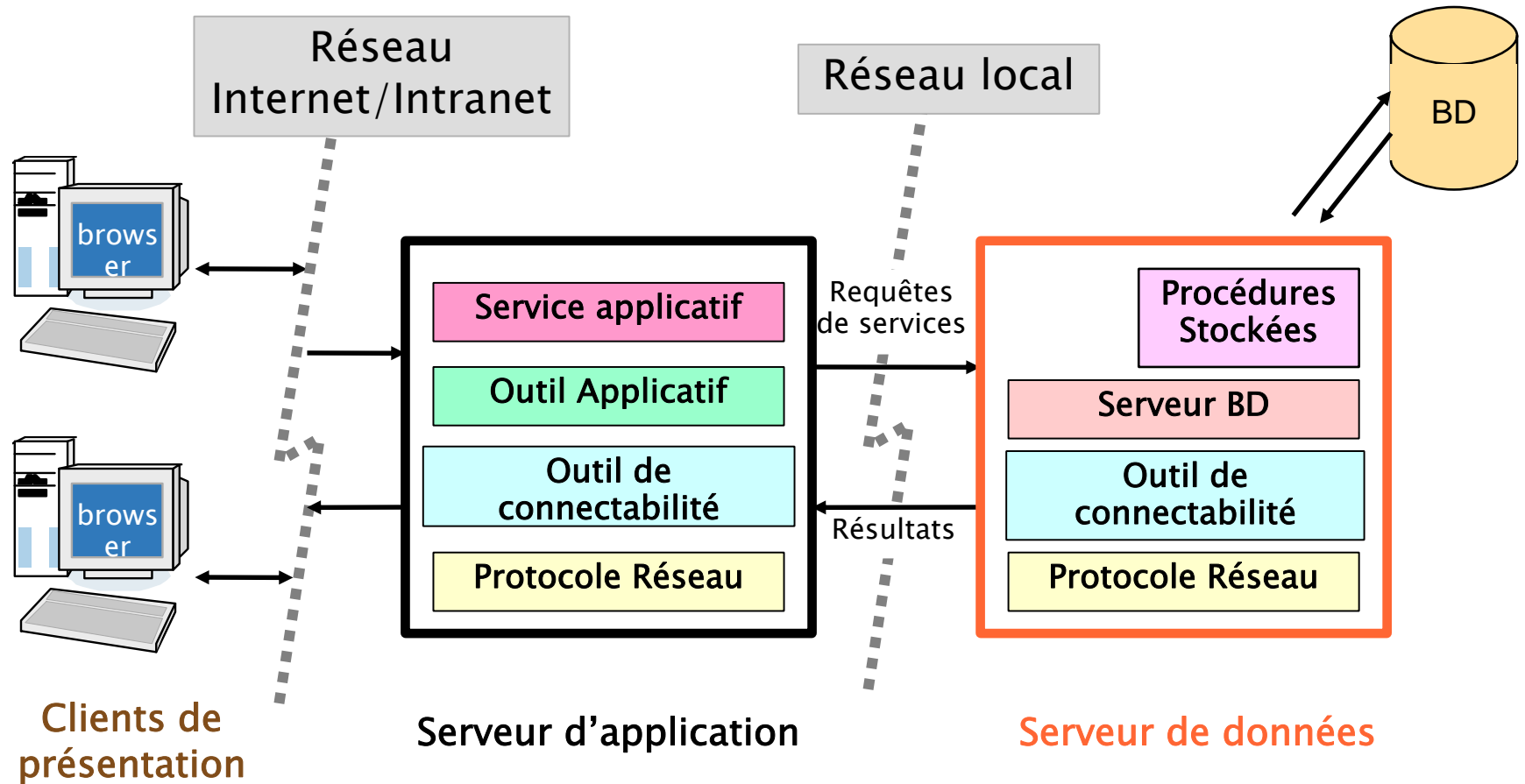
Gestionnaire de base

- ☐ Reçoit & traite les requêtes
- ☐ Gère le partage de données
- ☐ Gère sécurité, intégrité et reprise sur panne

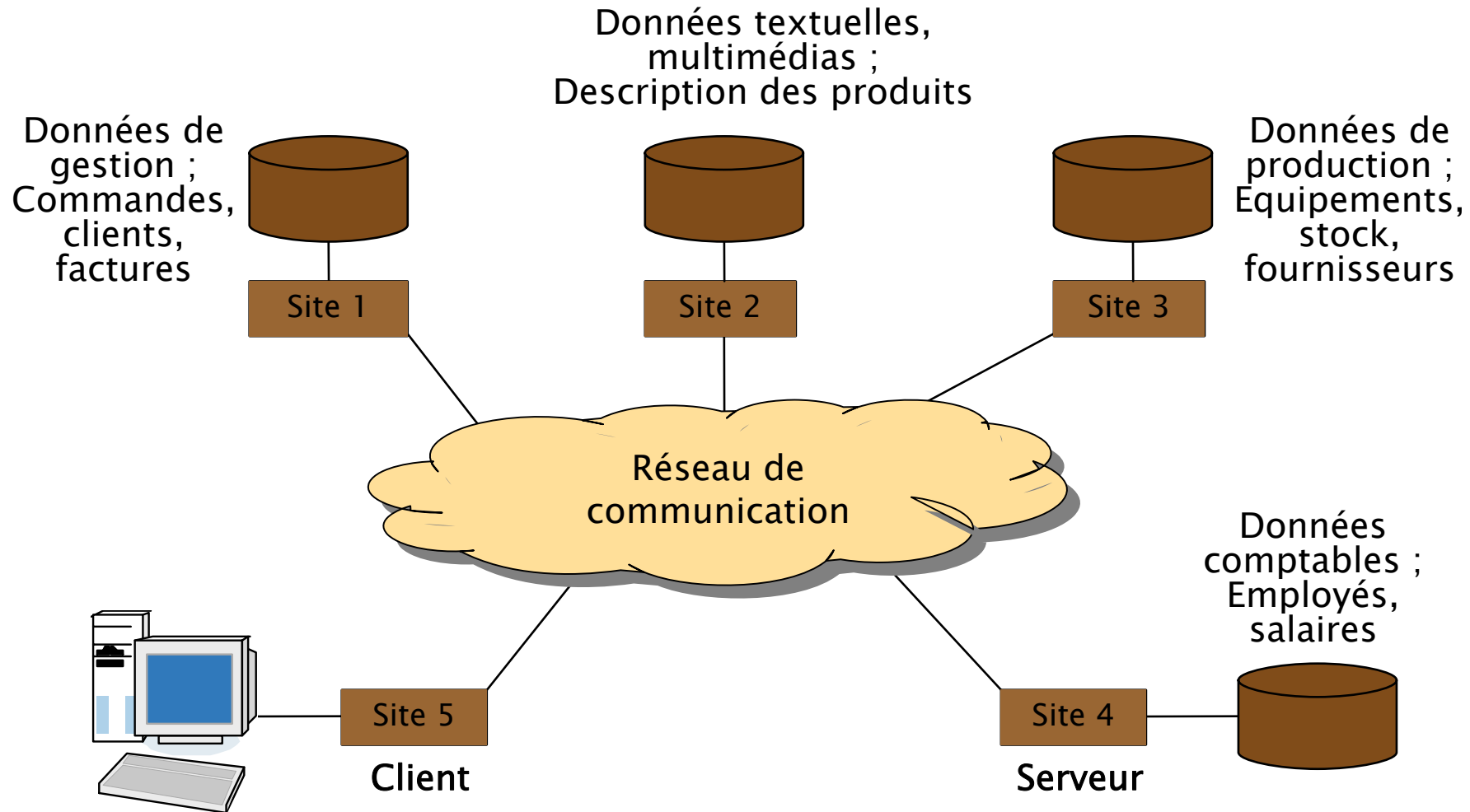
Architecture Client/Serveur à deux strates



Architecture Client/Serveur à trois strates



Architectures réparties



Bases de Données

Le modèle relationnel

Philippe Lahire

à partir du cours d'Isabelle Mirbel

L3-Informatique

Année 07/08

PLAN

- ▶ ■ Introduction
- Concepts de base
 - Attribut, Domaine, Produit cartésien, T-uple, Relation, Schéma, Contraintes d'intégrité*
- Dépendances fonctionnelles
- Décomposition
- Formes Normales
 - 1NF, 2NF, 3NF, BCNF, 4NF, 5NF*
- Couverture minimale
 - Définition*
 - Construction*
 - De la couverture au modèle relationnel*

■ Introduction

- Codd 1970
- Indépendant de l'implémentation & des applications
 - Cohérence des données
 - Limitation des redondances
- Structure de données
- Contraintes d'intégrité
- Algèbre relationnelle

PLAN

■ Introduction

▶ ■ Concepts de base

Attribut, Domaine, Produit cartésien, T-uple, Relation, Schéma, Contraintes d'intégrité

■ Dépendances fonctionnelles

■ Décomposition

■ Formes Normales

1NF, 2NF, 3NF, BCNF, 4NF, 5NF

■ Couverture minimale

Définition

Construction

De la couverture au modèle relationnel

■ Attribut & Domaine

■ Attribut

Le plus petit lot d'information utilisable & ayant une signification, indépendamment des autres lots.

Exemple

NomPilote, VilleDépart, VilleArrivée

■ Domaine

Ensemble des valeurs possibles pour un attribut

Exemple

D1 = Chaîne de caractères

D2 = {monsieur, madame, mademoiselle}

■ Produit cartésien

■ Produit cartésien de n domaines

Ensemble des tuples « V_1, V_2, \dots, V_n »
Tels que $V_i \in D_i$

➡ **Tuple**

Exemple

$D1 = \{A, B, C, D\}$

$D2 = \{1, 2, 3\}$

Produit cartésien:

(A,1)	(B,1)	(C,1)	(D,1)
(A,2)	(B,2)	(C,2)	(D,2)
(A,3)	(B,3)	(C,3)	(D,3)

■ Tuple

■ Tuple

Élément d'un produit cartésien

Un élément de l'ensemble figuré par une relation

Exemple

$D1 = \{A,B,C,D\}$

$D2 = \{1,2,3\}$

Exemples de tuples :

(A,1) (B,3) (C,1)

■ Relation

■ Relation

Sous-ensemble du produit cartésien des domaines d'un ensemble d'attributs

Exemple

D1 = chaîne de caractères

D2 = chaîne de caractères

Relation:

Paul	Tokyo
Pierre	Tokyo
Paul	Londres
Jacques	Paris
Pierre	New-York

■ Dépendance Fonctionnelle

■ Dépendance Fonctionnelle

On dit que X est en **DF** avec Y si à chaque valeur de X correspond **au plus 1** valeur de Y

Exemple

$N^{\circ}BonCommande \rightarrow TotalCommande$

$N^{\circ}Etudiant \nrightarrow Diplôme$

Schéma

■ Définition d'une relation en **extension**

➡ Tous les tuples

Relation PILOTE

Nom	Adresse
Paul	Tokyo
Pierre	Tokyo
Paul	Londres

■ Définition d'une relation en **intention**

➡ Attributs & domaines

PILOTE (Nom, Adresse)
D1=chaîne de caractère
D2=chaîne de caractère

■ Schéma relationnel

Ensemble de relations définies en intention

PILOTE (Nom, Prénom, Adresse)
AVION (Numéro, Capacité)

■ Contraintes d'intégrité (1)

■ Contrainte d'intégrité

Assertions qui doivent être vérifiées par les données à des instants déterminés

■ Clé

Clé candidate

Clé primaire

Clé étrangère

■ Intégrité référentielle

■ Contrainte d'intégrité sur les entités (tuples)

■ Contrainte d'intégrité sur les domaines

■ Contrainte d'unicité

■ Contraintes d'intégrité (2)

■ Clé candidate

Attribut ou groupe d'attributs **minimal** qui peut **identifier** de façon **unique** chaque tuple d'une relation

- ➔ La valeur d'une clé candidate est donc **distincte** pour chaque tuple
- ➔ Toute relation a **au moins** une clé candidate et peut en avoir plusieurs
- ➔ Les clés candidates d'une relation n'ont pas forcément le même nombre d'attributs
- ➔ Attribut ou ensemble d'attributs minimal **en DF élémentaire** avec les autres attributs de la relation (*de préférence mais pas obligatoire*)

■ Contraintes d'intégrité (3)

■ Clé primaire

Attribut ou groupe d'attributs sélectionné parmi les clés candidates de la relation

➔ peut être choisie de façon arbitraire mais le contexte aide souvent à la déterminer

■ Clé étrangère

Attribut ou groupe d'attributs correspondant à une **clé primaire** dans une autre relation du schéma

■ Contraintes d'intégrité (4)

Etudiant(N°SS, N°Etudiant, Nom, Prénom, date-naissance, adresse)

■ Clés candidates
N°SS
N°Etudiant

■ Clé primaire
N°Etudiant

➡ Etudiant(N°Etudiant, N°SS, Nom, Prénom, date-naissance, adresse)

■ Contraintes d'intégrité (4)

Etudiant(N°Etudiant, N°SS, Nom, Prénom, date-naissance, adresse)

Diplôme(Nom-dip, domaine, niveau, année-créditation)

■ Clé candidate
Nom-dip

■ Clé primaire
Nom-dip

➡ Diplôme(Nom-dip, domaine, niveau, année-créditation)

■ Clé étrangère

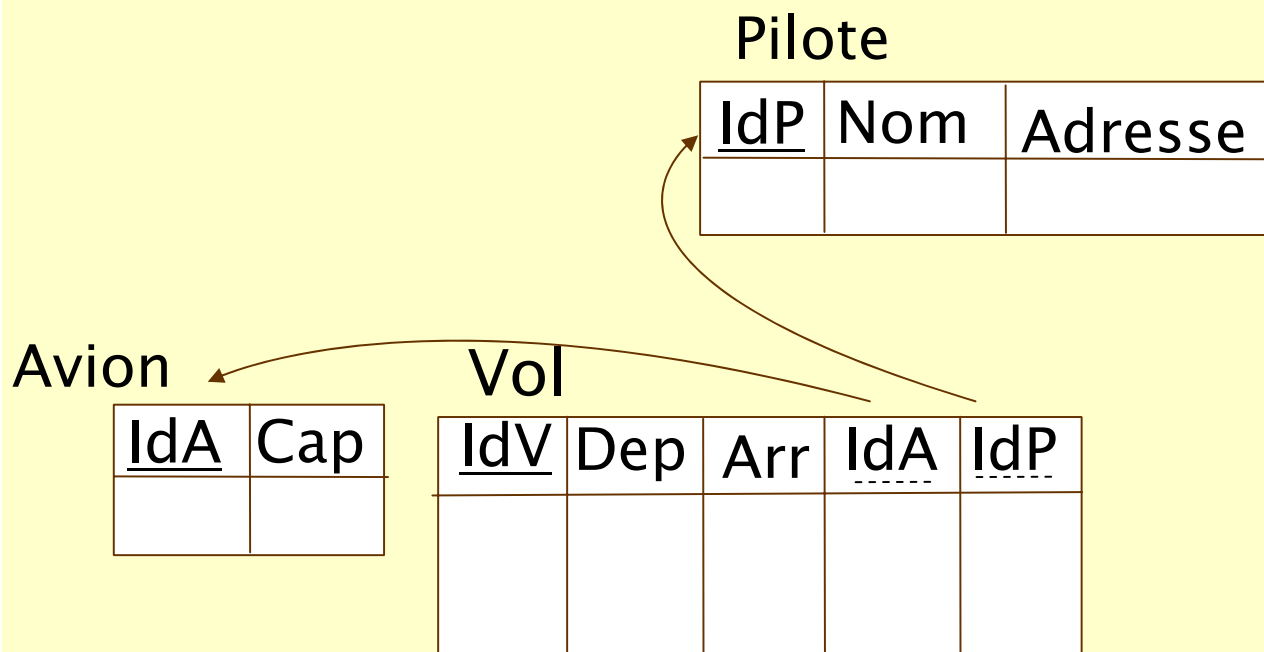
Etudiant(N°Etudiant, N°SS, Nom, Prénom, date-naissance, adresse, Nom-dip)

Contraintes d'intégrité (5)

Intégrité référentielle

Contraindre les valeurs d'un groupe d'attributs aux valeurs de la clé primaire d'une autre relation

Exemple



■ Contraintes d'intégrité (6)

■ Intégrité d'entité

La valeur `<null>` n'est pas autorisée pour les attributs constituant une clé.

■ Intégrité de domaine

Un domaine est associé à chaque attribut pour contraindre l'ensemble de ses valeurs possibles

■ Unicité

Une même valeur d'attribut clé ne peut être affectée à deux tuples différents

PLAN

■ Introduction

■ Concepts de base

Attribut, Domaine, Produit cartésien, T-uple, Relation, Schéma, Contraintes d'intégrité

▶ ■ Dépendances fonctionnelles

■ Décomposition

■ Formes Normales

1NF, 2NF, 3NF, BCNF, 4NF, 5NF

■ Couverture minimale

Définition

Construction

De la couverture au modèle relationnel

Dépendances Fonctionnelles (1)

Dépendance Fonctionnelle

$X \rightarrow Y$ X, Y : attributs

On dit que X est en **DF** avec Y si à chaque valeur de X correspond **au plus 1** valeur de Y

Exemple

NuméroImmatriculation \rightarrow Modèle
NuméroCommande \rightarrow IdClient

■ Dépendances Fonctionnelles (2)

■ Dépendance Fonctionnelle (suite)

Soient X et Y deux attributs de la relation R, on dit que l'on a une **DF** $X \rightarrow Y$ si et seulement si:
si E1 et E2, 2 n-uplets de R, ont les mêmes valeurs pour l'attribut X alors E1 et E2 ont nécessairement les mêmes valeurs pour l'attribut Y.

■ Utilité

- ❑ Contraintes sur les données que contient une relation
- ❑ Outil de contrôle de l'intégrité des données (au niveau du schéma)
- ❑ Outil mathématique permettant d'expliquer le processus de normalisation
- ❑ Indispensable pour re-modeler un schéma de BD qui pose problème

Dépendances Fonctionnelles (3)

Dépendance Fonctionnelle Directe

On dit qu'une DF $A \rightarrow B$ est **directe** s'il n'existe aucun attribut C tel que l'on puisse avoir $A \rightarrow C$ et $C \rightarrow B$.

- $A \rightarrow B$ ne peut être obtenue par transitivité

Dépendances Fonctionnelles (4)

■ Quelques propriétés des DFs (Axiomes d'Armstrong)

□ **Réflexivité** **Immatriculation** \rightarrow **Immatriculation**

$X \rightarrow X$

□ **Augmentation** **Immatriculation** \rightarrow **Modèle**
 Immatriculation, âge \rightarrow **Modèle, âge**

Si $X \rightarrow Y$ alors $XZ \rightarrow YZ$

□ **Transitivité** **Immatriculation** \rightarrow **Modèle**
 Modèle \rightarrow **Capacité**

Si $X \rightarrow Y$ et $Y \rightarrow Z$ alors $X \rightarrow Z$

□ **Pseudo-transitivité**

Si $X \rightarrow Y$ et $WY \rightarrow Z$ alors $WX \rightarrow Z$

Immatriculation \rightarrow **Modèle**
Modèle, nbPortes \rightarrow **Capacité**

Dépendances Fonctionnelles (5)

Dépendance Fonctionnelle Élémentaire

$$X \rightarrow A$$

Si A est un attribut unique et non-inclus dans X,
alors X', inclus dans X tel que $X' \rightarrow A$

n'existe pas

Non élémentaire

Exemple

~~NuméroImmatriculation, IdClient~~ → Modèle
NuméroImmatriculation → Modèle

Dépendance Fonctionnelle canonique

➡ Un seul attribut en partie droite

Rappel sur les Contraintes d'Intégrité

Clé

Une fois normalisé...

Attribut ou ensemble d'attributs minimal **en DF élémentaire** avec les autres attributs de la relation

Exemple

$R(A1, A2, A3, A4)$

$A1, A2$

Clés candidates



$A1, A2 \rightarrow A3$

$A1, A2 \rightarrow A4$

$A1 \not\rightarrow A3, A1 \not\rightarrow A4$

$A2 \not\rightarrow A3, A2 \not\rightarrow A4$

P L A N

■ Introduction

■ Concepts de base

Attribut, Domaine, Produit cartésien, T-uple, Relation, Schéma, Contraintes d'intégrité

■ Dépendances fonctionnelles

▶ ■ Décomposition

■ Formes Normales

1NF, 2NF, 3NF, BCNF, 4NF, 5NF

■ Couverture minimale

Définition

Construction

De la couverture au modèle relationnel

■ Décomposition (1)

■ Décomposition

La décomposition d'un schéma de relation $R(A_1, \dots, A_n)$ consiste en son remplacement par une collection de relations (R_1, \dots, R_m) tel que

$$\text{SCHEM}(R) = \text{SCHEM}(R_1) \cup \dots \cup \text{SCHEM}(R_m)$$

■ But de la décomposition

Casser R en de plus petites relations afin d'éviter

- les redondances
- les anomalies de mise-à-jour

■ Décomposition (1')

■ Décomposition de $R(X, Y, Z)$ tel que $X \rightarrow Y$

Si la DF $X \rightarrow Y$ est élémentaire :

- En créant une relation R' de clé X et d'attributs Y
- En supprimant dans R les attributs Y

Si la DF $X \rightarrow Y$ est non élémentaire (*pas minimale*) :
 X' est la partie de X en DF élémentaire avec Y

- En créant une relation R' de clé X' et d'attributs Y
- En supprimant dans R les attributs Y

Décomposition (2)

Exemple de redondances & anomalies de mise-à-jour

IdCde	IdPdt	Libellé	Prix U	Qté Cdée
100	A01	Crayon	3	2
100	A03	Stylo bleu	2.4	2
100	A02	Gomme	1.2	1
106	A03	Feutre bleu	5	4
110	A01	Crayon	3	1

Hypothèse: les prix des produits sont fixes et indépendants des commandes

Décomposition (3)

La jointure naturelle

IdCde	IdPdt	Qté Cdée
100	A01	2
100	A03	2
100	A02	1
106	A01	4
110	A03	1

IdPdt	Libellé	Prix U
A01	Crayon	3
A03	Stylo bleu	2.4
A02	Gomme	1.2

IdCde	IdPdt	Libellé	Prix U	Qté Cdée
100	A01	Crayon	3	2
100	A03	Stylo bleu	2.4	2
100	A02	Gomme	1.2	1
106	A01	Crayon	3	4
110	A03	Stylo bleu	2.4	1

Cde(IdCde,IdPdt,QtéCdée) \bowtie {IdPdt} Produit(IdPdt, Libellé, PrixU)

Décomposition (4)

■ Décomposition sans perte de données

Théorème de Heath: Soit une relation

$R(A,B,C)$

où A, B et C sont des ensembles d'attributs disjoints, avec

$B \rightarrow C$

alors

$R(A,B,C) = R[A,B] \Join_{\{B\}} R[B,C]$

EMPLOYE(Mat, NomE, NoDept, NomDept)



EMPLOYE(Mat, NomE, NoDept)

DEPARTEMENT(NoDept, NomDept)

Décomposition (5)

■ Décomposition sans perte de dépendance

La décomposition de R en $R1$ et $R2$ est **sans perte de dépendance** si on peut retrouver (par transitivité) **toutes les DF** de R à partir des projections sur $R1$ et $R2$


La **projection** de l'ensemble des DF de R sur $R1$ est l'ensemble des DF de R qui ont leurs attributs de départ et d'arrivée dans $R1$.

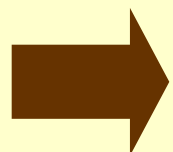
Décomposition (6)

Décomposition sans perte de dépendance (suite)

PARTICIPATION(NoDept, CodeProjet, Matricule, FonctionP)


 NoDept, CodeProjet → Matricule
 NoDept, CodeProjet → FonctionP


 PARTICIPATION(NoDept, CodeProjet, FonctionP)
 EMPLOYE(Matricule, NoDept)


 Perte de **dépendance** : un seul employé par département sur un projet donné

P L A N

■ Introduction

■ Concepts de base

Attribut, Domaine, Produit cartésien, T-uple, Relation, Schéma, Contraintes d'intégrité

■ Dépendances fonctionnelles

■ Décomposition

▶ ■ Formes Normales

1NF, 2NF, 3NF, BCNF, 4NF, 5NF

■ Couverture minimale

Définition

Construction

De la couverture au modèle relationnel

■ Les formes normales (1)

■ Normalisation

- ➔ Minimiser les redondances pour améliorer les performances
- Requêtes d'interrogation & de mise-à-jour plus rapides
- Moins de valeurs <null>
- Gestion de l'intégrité des données simplifiée

■ Les formes normales (2)

■ Normalisation (suite)

Technique de consolidation d'un schéma conceptuel

- Il existe plusieurs formes normales:
1NF, 2NF, 3NF, BCNF (4NF 5NF)

*Normal
Form*

*Boyce/Codd
Normal Form*

Les formes normales (3)

Première Forme Normale (1NF)

Une relation est en 1NF ssi tous ses attributs ont des valeurs simples (non multiples, non composées)

R1=(Matr,Nom, Adresse)

R2=(No, Rue, Code_postal, Ville)

R3=(Matr,Nom, Prénoms)

R4=(Matr,Nom, Prénom1, Prénom2, Prénom3)

*Non
normalisée*

1NF

Les formes normales (4)

■ Deuxième Forme Normale (2NF)

Une relation est en 2NF ssi elle est en 1NF et ssi tout attribut qui n'appartient pas à une clé *candidate* ne dépend pas d'un sous-ensemble strict des attributs d'une clé candidate



Une relation peut être en 2NF par rapport à une de ses clés candidates et ne pas l'être par rapport à une autre



Une relation avec une clé candidate choisie réduite à **un seul attribut** est, par définition, forcément en 2NF

■ Les formes normales (5)

■ Technique de recherche de la 2NF

- Recherche de toutes les **DFs**
- Recherche de toutes les **clés candidates**
- Vérification de la **forme normale** pour chaque clé candidate
- **Transformation** éventuelle de la relation

R1=(NoCommande, NoProduit, LibelléProduit, QtéCommandée)

■ DFs

NoCommande+NoProduit → QtéCommandée, LibelléProduit

NoProduit → LibelléProduit

■ Clés candidates ➡ NoCommande+NoProduit

■ Pas 2NF ➡ NoCommande+NoProduit

➡ R1=(NoCommande, NoProduit, QtéCommandée)
R2=(NoProduit, LibelléProduit)

Les formes normales (6)

Problèmes liés au non respect de la 2FN

R1 = (NoCommande, NoProduit, LibelléProduit, QtéCommandée)

Cmde	Pdt	Libellé	Qté
100	A01	Stylo bleu	2
100	A02	Gomme	1
106	A01	Stylo bleu	4
110	A03	Crayon	1

Stylo bleu
Stylo bille bleu

Redondances
Risque d'erreur
Problèmes de mise-à-jour
...

■ Les formes normales (7)

■ Troisième Forme Normale (3NF)

Une relation est en **3NF** ssi elle est en 2NF et ssi chaque attribut *qui n'appartient pas à une clé candidate* ne dépend pas fonctionnellement d'un attribut *qui n'appartient pas à une clé candidate*



Clés

candidates

■ Technique de recherche de la 3FN

- ☐ Recherche de toutes les **DFs**
- ☐ Recherche de toutes les **clés candidates**
- ☐ Vérification de la **forme normale** pour chaque clé candidate
- ☐ **Transformation** éventuelle de la relation

Les formes normales (8)

$R1 = (\text{Matricule}, \text{Nom}, \text{Service}, \text{Bâtiment}, \text{Grade})$

DFs

$\text{Matricule} \rightarrow \text{Nom}, \text{Service}, \text{Bâtiment}, \text{Grade}$

$\text{Service} \rightarrow \text{Bâtiment}$

Clés candidates \rightarrow Matricule

Pas 3FN \rightarrow $\text{Service} \rightarrow \text{Bâtiment}$

\rightarrow $R1 = (\text{Matricule}, \text{Nom}, \text{Service}, \text{Grade})$
 $R2 = (\text{Service}, \text{Bâtiment})$

Les formes normales (9)

Problèmes liés au non respect de la 3FN

R1 = (Matricule, Nom, Service, Bâtiment, Grade)

Nom	Service	Bâtiment
Dupont	S1	Bât A
Durand	S2	Bât A
Schmitt	S1	Bât A

S1 déménage au bâtiment B...

Redondances

Risque d'incohérences

Problèmes de mise-à-jour

...

→ Une relation à clé atomique et en 3NF est en BCNF et en 4 et 5 NF

■ Les formes normales (10)

■ Un problème...

■ Une relation est en 3NF si

- Tous les attributs sont en DF avec la clé
- Ils sont en DF avec toute la clé
- Pas de DF entre les éléments non clé

$R(\underline{A}, \underline{B}, C, D)$

$A, B \rightarrow C$

$A, B \rightarrow D$

$C \not\rightarrow D, D \not\rightarrow C$

$C \rightarrow B ?$

➔ Forme Normale de Boyce-Codd (BCNF)

■ Les formes normales (11)

■ Forme Normale de Boyce/Codd (BCNF)

Une relation est **BCNF** ssi elle est en 3NF et s'il **n'y a pas d'autres DF** élémentaires que celles issues des clés candidates



Les seules sources de DF sont les clés candidates

■ Technique de recherche de la forme BCNF

- ☐ Recherche de toutes les **DFs**
- ☐ Recherche de toutes les **clés candidates**
- ☐ Vérification de la **forme normale** pour chaque clé candidate
- ☐ **Transformation** éventuelle de la relation

Les formes normales (12)

PLANNING(Jour, Compagnie, Avion, NbPassagers)

1 vol par jour par compagnie

3 NF | Jour, Compagnie → Avion
Jour, Compagnie → NbPassagers
Avion ↗ NbPassagers et NbPassagers ↗ Avion

Dépendances attributs non clés vers attributs clés ?

Les formes normales (13)

PLANNING(Jour, Compagnie, Avion, NbPassagers)

1 vol par jour par compagnie

3 NF

Jour, Compagnie \rightarrow Avion

Jour, Compagnie \rightarrow NbPassagers

Avion \nrightarrow NbPassagers et NbPassagers \nrightarrow Avion

Dépendances attributs non clés vers attributs clés ?

Avion \rightarrow Compagnie

Les formes normales (14)

PLANNING(Jour, Compagnie, Avion, NbPassagers)

1 vol par jour par compagnie

Jour, Compagnie \rightarrow Avion

Jour, Compagnie \rightarrow NbPassager

Avion \nrightarrow NbPassagers et NbPassagers \nrightarrow Avion

Avion \rightarrow Compagnie ?



APPAREIL(Avion, Compagnie)

PLANNING(Jour, Compagnie, NbPassagers)

Perte de données ?

Perte de dépendance ?

Les formes normales (15)

PLANNING(Jour, Compagnie, Avion, NbPassagers)

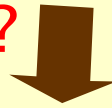
1 vol par jour par compagnie

Jour, Compagnie \rightarrow Avion

Jour, Compagnie \rightarrow NbPassagers

Avion \nrightarrow NbPassagers et NbPassagers \nrightarrow Avion

Avion \rightarrow Compagnie ?



APPAREIL(Avion, Compagnie)

PLANNING(Jour, Compagnie, NbPassagers)

Perte de données ?

Perte de dépendance ?

Jour , Compagnie \rightarrow Avion

■ Les formes normales (16)

PLANNING(Jour, Compagnie, Avion, NbPassagers)

1 vol par jour par compagnie

Jour, Compagnie → Avion

Jour, Compagnie → NbPassagers

Avion \nrightarrow NbPassagers et NbPassagers \nrightarrow Avion

Avion → Compagnie ?



PLANNING(Jour, Avion, NbPassagers)

APPAREIL(Avion, Compagnie)

Et si plusieurs avions de la même compagnie sont utilisés le même jour ?

■ Les formes normales (17)

- Perte de données / Perte de dépendances
 - Jusqu'en 3NF, décomposition
 - Sans perte de dépendance
 - Sans perte de données
 - En BCNF, décomposition
 - Sans perte de données
 - Éventuellement avec perte de dépendance

■ Un exemple de normalisation

R (A, B, C, D)

$A, B \rightarrow C, D$

$C \rightarrow A, B, D$

$A \rightarrow D$

■ Clés candidates ?

■ Un exemple de normalisation (2)

R(A, B, C, D)

$A, B \rightarrow C, D$

$C \rightarrow A, B, D$

$A \rightarrow D$

■ Clés candidates ?

A, B

C

■ 1NF ?

Oui

■ 2NF ?

Un exemple de normalisation (3)

$R(A, B, C, D)$

$A, B \rightarrow C, D$

$C \rightarrow A, B, D$

$A \rightarrow D$

■ 2NF ?

Non



~~Clés candidates~~

~~C → 2NF~~

~~A, B → pas 2NF~~



$R1(A, D)$

$R2(A, B, C)$

Clés candidates: A

Clés candidates: A, B et C

■ 3NF ?

Un exemple de normalisation (4)

$R(A, B, C, D)$
 $\left\{ \begin{array}{l} R1(A, D) \\ R2(A, B, C) \end{array} \right.$
 $A, B \rightarrow C, D$
 $C \rightarrow A, B, D$
 $A \rightarrow D$

■ 3NF ?

Oui

■ BCNF ?

Un exemple de normalisation (5)

$R(A, B, C, D)$

$A, B \rightarrow C, D$

$C \rightarrow A, B, D$

$A \rightarrow D$

$R1(A, D)$

$R2(A, B, C)$

■ BCNF ?

Oui



$R1(\underline{A}, D)$

$R2(\underline{A}, \underline{B}, C)$ ou $R2(\underline{C}, A, B)$

Formes normales – suite (1)

- BCNF insuffisante pour éliminer redondances & anomalies de mise-à-jour

R (N°Emp, Compétence, Langue)

- ➔ Deux relations :
employé/compétences & employé/langues
relations indépendantes
 $N^{\circ}Emp \rightarrow Compétence \mid Langue$

Aucune **dépendance fonctionnelle** ne permet une telle décomposition.

- ➔ **dépendance multivaluée** va permettre la décomposition

Formes normales – suite (1)

Dépendance multi-valuée

Soit $R(A_1, A_2, \dots, A_n)$ un schéma de relation et X et Y des sous ensembles de $\{A_1, A_2, \dots, A_n\}$.

On dit que X **multi-détermine** Y ($X \twoheadrightarrow Y$) si :
étant données des valeurs de X :

- Il y a un **ensemble** de valeurs de Y associées
- Cet ensemble est **indépendant**¹ des autres attributs $Z = R - X - Y$ de la relation R .

1: Indépendant dans le sens “*qui n'a rien à voir avec*”

Formes normales – suite (2)

Dépendance multi-valuée élémentaire

Une dépendance multi-valuée est élémentaire si pour une relation R , $X \twoheadrightarrow Y$

- Y n'est pas vide et est disjoint de X
- R ne contient pas une autre DM de type $X' \twoheadrightarrow Y'$ telle que $X' \subset X$ et $Y' \subset Y$

Formes normales – suite (3)

Dépendance multi-valuée (suite)

- Soit $R(A, B, C)$, on a une dépendance multi-valuée $A \twoheadrightarrow B$ dans R si et seulement si,
 - chaque fois que $\langle a, b, c \rangle$ et $\langle a, b', c' \rangle$ apparaissent dans R ,
 - alors $\langle a, b', c \rangle$ et $\langle a, b, c' \rangle$ apparaissent dans R .



Il ne faut pas considérer que si entre deux constituants d'une relation il n'y a pas de dépendance fonctionnelle, alors il y a une dépendance multi-valuée !

Formes normales – suite (4)

Quatrième forme normale (4NF)

Une relation est en 4NF si et seulement si les seules dépendances multi-valuées élémentaires sont celles dans lesquelles une clé détermine un attribut.

R (N°Emp, Compétence, Langue)

N° Emp	Compétence	Langue
4	cuisine	français
4	cuisine	allemand
4	cuisine	grec
4	dactylo	français
4	dactylo	allemand
4	dactylo	grec

R1 (N°Emp, Compétence)

R2 (N°Emp, Langue)



Pas de perte de données
car multi-valuées

Formes normales – suite (5)

Quatrième forme normale (4NF)

Contre-exemple

R (N°aiguille, Modèle pull, N° laine)

N° aiguille	Modèle pull	N° laine
A1	M1	L1
A1	M1	L2
A1	M1	L4
A1	M2	L3
A2	M1	L3
A3	M2	L1

$A1 \rightarrow \{L1, L2, L4, L3\}$

$A1, M1 \rightarrow \{L1, L2, L4\}$

Les laines associées à un N° aiguille ne dépendent pas uniquement de ce N° aiguille.

Formes normales – suite (6)

■ Quatrième forme normale (4NF)

Un autre exemple

Cours	Prof	Description
maths	Durand Dupont Blanc	bases analyse algèbre
physique	Dupont Noir	bases optique



R (NomCours, NomProf, DescCours)

Formes normales – suite (7)

Un autre exemple (suite)

R (NomCours, NomProf, DescCours)

NomCours	NomProf	DescCours
Maths	Durand	bases
Maths	Dupont	bases
Maths	Blanc	bases
Maths	Durand	algèbre
Maths	Dupont	algèbre
Maths	Blanc	algèbre
Maths	Durand	analyse
Maths	Dupont	analyse
Maths	Blanc	analyse
Physique	Noir	bases
Physique	Dupont	bases
Physique	Noir	optique
Physique	Dupont	optique

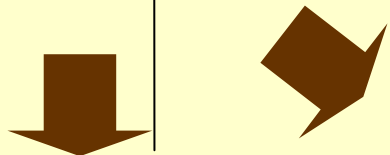
Anomalie...

... d'insertion
... de suppression

Formes normales – suite (8)

Un autre exemple (suite)

R (NomCours, NomProf, DescCours)

NomCours	NomProf	DescCours
		

R2 (NomCours, NomProf)

NomCours	NomProf
Maths	Durand
Maths	Dupont
Maths	Blanc
Physique	Noir
Physique	Dupont

R1 (NomCours, DescCours)

NomCours	DescCours
Maths	bases
Maths	algèbre
Maths	analyse
Physique	bases
Physique	optique

Formes normales – suite (9)

4NF & 2NF

R (NomCours, NomProf, DescCours)

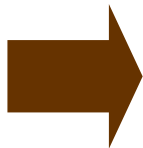
NomCours \rightarrow NomProf | DescCours

NomProf \rightarrow NomCours

➡ Une décomposition BCNF...

R1 (NomCours, NomProf)

R2 (NomProf, DescCours) ← Pas satisfaisant...



Analyser les DMs avant les DFs
Décomposer selon les DMs avant les DFs

Formes normales – suite (10)

Cinquième forme normale (5NF)

R (N°Représentant, N°Produit, NomSociété)

Le représentant *N°Représentant* représente le produit *N°Produit* pour le compte de la société *NomSociété*

➡ Il n'y a pas de DFs entre les attributs

R est en BCNF

➡ **MAIS** il n'y a pas de DMs entre les attributs

NomSociété → N°Produit | N°Représentant

Risques d'anomalies ➡ Décomposition

R1(NomSociété, N°Représentant)

R2(NomSociété, N°Produit)

Formes normales – suite (11)

Cinquième forme normale (suite)

R (N°Représentant, N°Produit, NomSociété)

Le représentant *N°Représentant* représente le produit *N°Produit* pour le compte de la société *NomSociété*

Un représentant ne peut travailler pour deux sociétés qui commercialisent certains produits identiques

Décomposition insuffisante...

R1(NomSociété, N°Représentant)

R2(NomSociété, N°Produit)

... on ne sait plus quels produits sont vendus par qui



Formes normales – suite (12)

Cinquième forme normale (suite)

R (N°Représentant, N°Produit, NomSociété)

- Anomalies habituelles au regard de la manipulation des données

➔ Le fait qu'une certaine société commercialise un certain produit ne peut être introduit sans connaître au moins un représentant de cette société qui représente ce produit.

Aucune **dépendance fonctionnelle**, aucune **dépendance multivaluée** ne permet une telle décomposition.

➔ **dépendance mutuelle** va permettre la décomposition

Formes normales – suite (13)

Cinquième forme normale un autre exemple...

R (N°Etudiant, Département, Projet)

- Anomalies habituelles au regard de la manipulation des données

➔ On ne peut pas décomposer en 2 relations sans perdre des informations
Quels département proposent quels sujets
Quels étudiants appartiennent à quels départements
Quels sujets ont été choisis par quels étudiants

Aucune **dépendance fonctionnelle**, aucune **dépendance multivaluée** ne permet une telle décomposition.

➔ **dépendance mutuelle** va permettre la décomposition

Formes normales – suite (14)

- Etant donnée une relation $R(X, Y, Z)$, on dira qu'il y a une dépendance mutuelle de X vers Y , notée $X \lt \sim \gt Y$ si dans la relation, étant donné :

(x, y, z) et (x, y', z') deux tuples de R alors :

$(y, z') \Rightarrow (x, y, z')$ et $(y', z) \Rightarrow (x, y', z)$

- Le symbole symétrique $(\lt \sim \gt)$ est utilisé parce que $X \lt \sim \gt Y$ est vérifiée si et seulement si $Y \lt \sim \gt X$ est vérifiée.

- $X \lt \sim \gt Y$ est vérifiée, si et seulement si $X \lt \sim \gt Z$ l'est aussi. On note ces deux dépendances mutuelles ainsi :
$$X \lt \sim \gt Y|Z.$$

Formes normales – suite (15)

Cinquième forme normale un autre exemple...

R (N°Etudiant, Département, Code-Projet)

■ Etant donnée une relation

R (N°Etudiant, Département, Code-Projet), on dira qu'il y a une dépendance mutuelle de N°Etudiant vers Département, notée $N^{\circ}Etudiant \sim Département$ si dans la relation, étant donné :

(E1, D1, P1) et (E2, D2, P2) deux tuples de R alors :

$(D1, P2) \Rightarrow (E1, D1, P2)$ et $(D2, P1) \Rightarrow (E1, D2, P1)$

$N^{\circ}Etudiant \sim Département \mid Code-Projet$



R1 (N°Etudiant, Département)

R2 (Département, Code-Projet)

R3 (N°Etudiant, Code-Projet)

■ Formes normales – suite (16)

■ Cinquième forme normale (5NF)

Une relation est en 5NF si et seulement si toute dépendance de jointure est impliquée par les clés candidates de la relation.

Formes normales – suite (17)

Cinquième forme normale (5NF)

Soit $R(A_1, A_2, \dots, A_n)$ une relation et X_1, X_2, \dots, X_m des sous-ensembles de (A_1, A_2, \dots, A_n) . On dit qu'il existe une dépendance de jointure $*(X_1, X_2, \dots, X_m)$ si R est la jointure de ses projections sur X_1, X_2, \dots, X_m

$R(\text{N}^\circ\text{Etudiant}, \text{Département}, \text{Projet})$
doit obéir à la dépendance de jointure
 $*(\text{Département Projet}, \text{Département étudiant}, \text{Code-Projet N}^\circ\text{Etudiant})$

Une dépendance multivaluée est un cas particulier de dépendance de jointure

Formes normales – suite (18)

Cinquième forme normale (suite)

R (N°Représentant, N°Produit, NomSociété)

Un représentant ne peut travailler pour deux sociétés qui commercialisent certains produits identiques

=

Lorsqu'un représentant représente un produit pour une société alors il représente ce produit pour toute société pour laquelle il travaille et qui le commercialise



R1 (N°Représentant, N°Produit)

R2 (N°Représentant, NomSociété)

R3 (N°Produit, NomSociété)

On peut recomposer la relation initiale sans perte d'information.

Formes normales – suite (19)

Cinquième forme normale (suite)

R (Id-Plongeur, NomCrustacé, NomVille)

Tout plongeur ayant consommé un type de crustacés et ayant séjourné dans une ville les cultivant a consommé de ce type de crustacés dans cette ville



R1 (Id-Plongeur, NomCrustacé)

R2 (Id-Plongeur, NomVille)

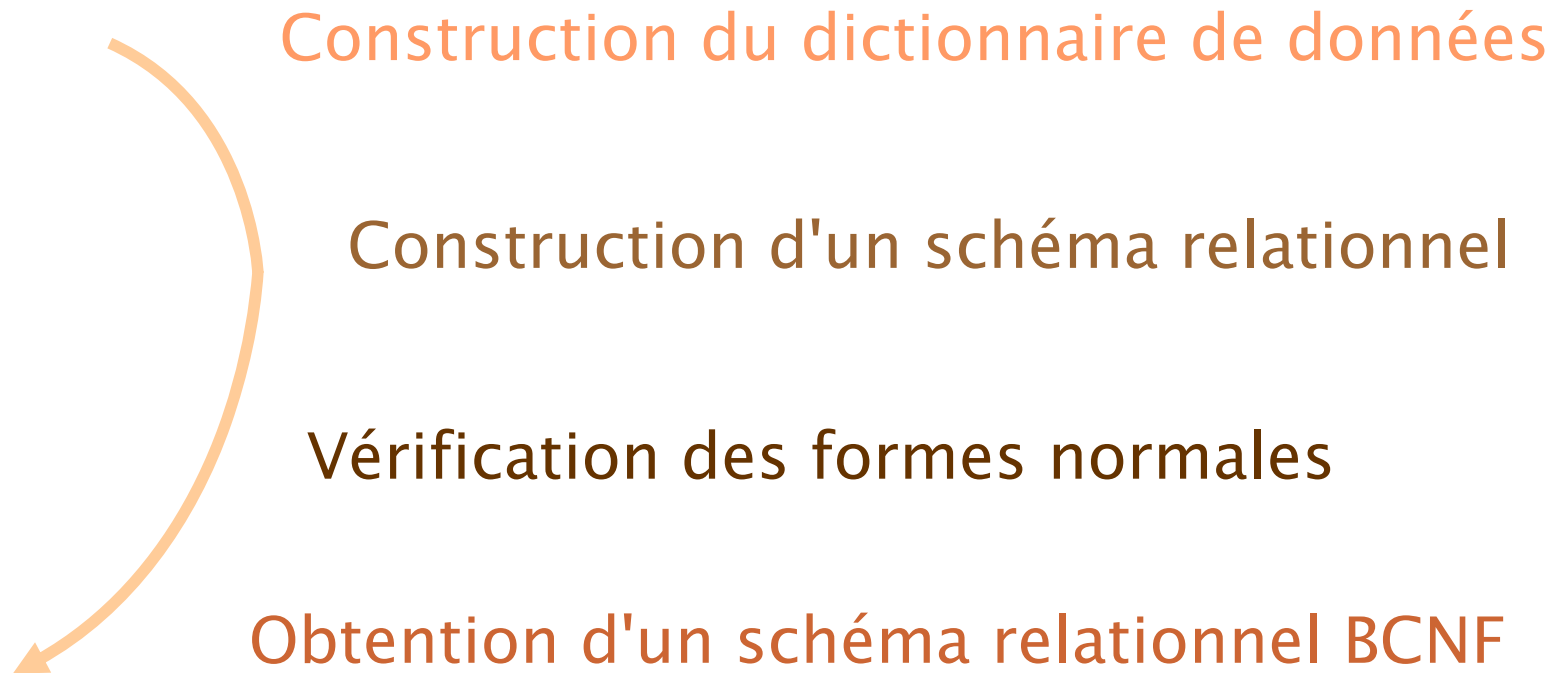
R3 (NomCrustacé, NomVille)

Formes normales – suite (20)

Les formes normales

- L'intuition souvent suffit pour 3NF
- Les anomalies peuvent peser peu comparées aux effets de plus de jointure...

■ Construction d'un schéma relationnel



PLAN

■ Introduction

■ Concepts de base

Attribut, Domaine, Produit cartésien, T-uple, Relation, Schéma, Contraintes d'intégrité

■ Dépendances fonctionnelles

■ Décomposition

■ Formes Normales

1NF, 2NF, 3NF, BCNF, 4NF, 5NF

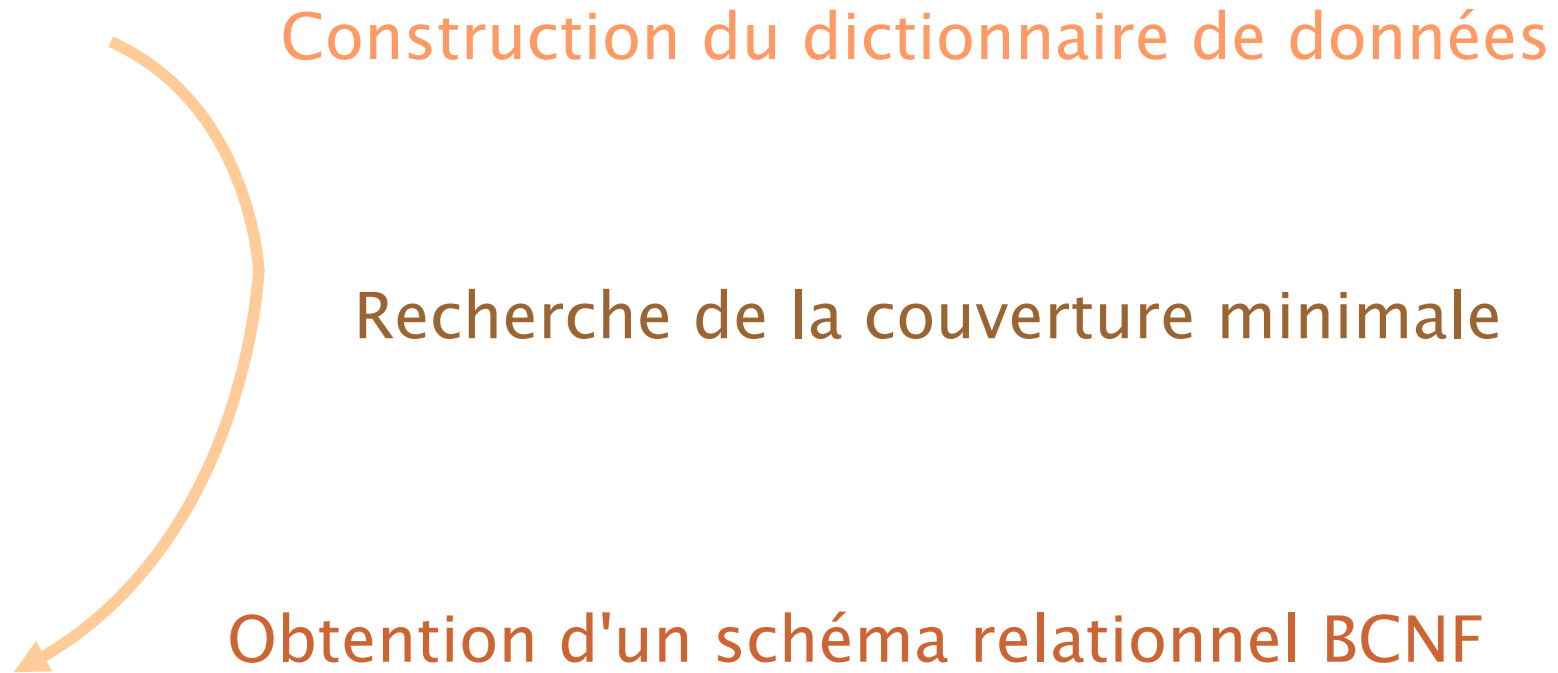
► ■ Couverture minimale

Définition

Construction

De la couverture au modèle relationnel

■ Construction d'un schéma relationnel



Couverture minimale – Définitions (1)

Fermeture transitive

Ensemble des DF élémentaires enrichi de toutes les DF élémentaires obtenues par transitivité pour un ensemble d'attributs A.

Ensemble des DF élémentaires:

$D = \{ \text{No Produit} \rightarrow \text{Nom Produit}; \text{No Fournisseur} \rightarrow \text{Nom Fournisseur}; \text{No Produit} \rightarrow \text{No Fournisseur} \}$

Fermeture transitive:

$D^+ = D \cup \{ \text{No Produit} \rightarrow \text{Nom Fournisseur} \}$

Conséquence

2 ensembles de DF élémentaires sont équivalents s'ils ont la même fermeture transitive

Couverture minimale – Définitions (2)

Couverture minimale

Un ensemble D de DF élémentaires pour un ensemble d'attributs A est une couverture minimale si

- toute DF f n'est pas redondante ($D - f$ n'est pas équivalent à D)
- toute DF élémentaire de A est dans la fermeture transitive

Recherche de la couverture minimale

Pour organiser les données sous forme d'un schéma relationnel BCNF

PLAN

■ Introduction

■ Concepts de base

Attribut, Domaine, Produit cartésien, T-uple, Relation, Schéma, Contraintes d'intégrité

■ Dépendances fonctionnelles

■ Décomposition

■ Formes Normales

1NF, 2NF, 3NF, BCNF, 4NF, 5NF

■ Couverture minimale

Définition

Construction

De la couverture au modèle relationnel



Couverture minimale – Construction (1)



Il n'existe pas d'algorithme simple et général pour trouver une couverture minimale

■ Technique de normalisation

- Ecrire les dépendances fonctionnelles
- Rendre canonique, élémentaires et directes les DFs qui ne le sont pas
- Représenter les nouvelles Dfs sous forme d'un graphe dont les noeuds sont les attributs impliqués dans les Dfs et les arcs les Dfs elles-mêmes

Couverture minimale – Construction (2)

□ Ecrire les dépendances fonctionnelles

IdCours \rightarrow IdProfesseur, LibelléCours

IdCours, N°Etudiant \rightarrow Note

Heure, N°Salle \rightarrow IdCours

N°Etudiant, Heure \rightarrow IdCours, N°Salle

N°Etudiant \rightarrow NomEtudiant, AdrEtudiant

Couverture minimale – Construction (3)

- Rendre canoniques & élémentaires les DFs qui ne le sont pas



Cette étape est la plus délicate dans la construction de la couverture minimale

$\text{IdCours} \rightarrow \text{LibelléCours}$

$\text{IdCours} \rightarrow \text{IdProfesseur}$

$\text{IdCours}, \text{N}^\circ\text{Etudiant} \rightarrow \text{Note}$

$\text{Heure}, \text{N}^\circ\text{Salle} \rightarrow \text{IdCours}$

$\text{N}^\circ\text{Etudiant}, \text{Heure} \rightarrow \text{IdCours}$

$\text{N}^\circ\text{Etudiant}, \text{Heure} \rightarrow \text{N}^\circ\text{Salle}$

$\text{N}^\circ\text{Etudiant} \rightarrow \text{NomEtudiant}$

$\text{N}^\circ\text{Etudiant} \rightarrow \text{AdrEtudiant}$

$$\begin{array}{l} \text{EH} \rightarrow \text{S} \Rightarrow \text{EHH} \rightarrow \text{SH} \\ \text{EHH} \rightarrow \text{SH} \\ \text{SH} \rightarrow \text{C} \end{array} \} \Rightarrow \text{EH} \rightarrow \text{C}$$

Couverture minimale – Construction (4)

- Représenter les nouvelles Dfs sous forme d'un graphe dont les noeuds sont les attributs impliqués dans les Dfs et les arcs les Dfs elles-mêmes
 - Construction de l'ensemble des **DFs** composées d'**un seul** attribut **source de DF**
 - Lister les DFs **non encore intégrées** (qui n'apparaissent pas dans le graphe représentant l'ensemble des DFs en 1).
 - Placer les DFs avec comme **source** un **sous-ensemble de l'ensemble des attributs** faisant partie de sources de DF et déjà placés dans le graphe.
 - **Intégrer** les DFs où l'un des attributs non affectés apparaît comme **source**
- ➔ Il ne doit pas y avoir de DFs non directes

Couverture minimale – Construction (5)

- Représenter les nouvelles DFs sous forme d'un graphe dont les noeuds sont les attributs impliqués dans les DFs et les arcs les DFs elles-mêmes
- Construction de l'ensemble des DFs composées d'un seul attribut **source de DF**

IdCours → LibelléCours

IdCours → IdProfesseur

N°Etudiant → NomEtudiant

N°Etudiant → AdrEtudiant



Couverture minimale – Construction (6)

- Lister les DFs **non encore intégrées** (qui n'apparaissent pas dans le graphe représentant l'ensemble des DFs en 1).

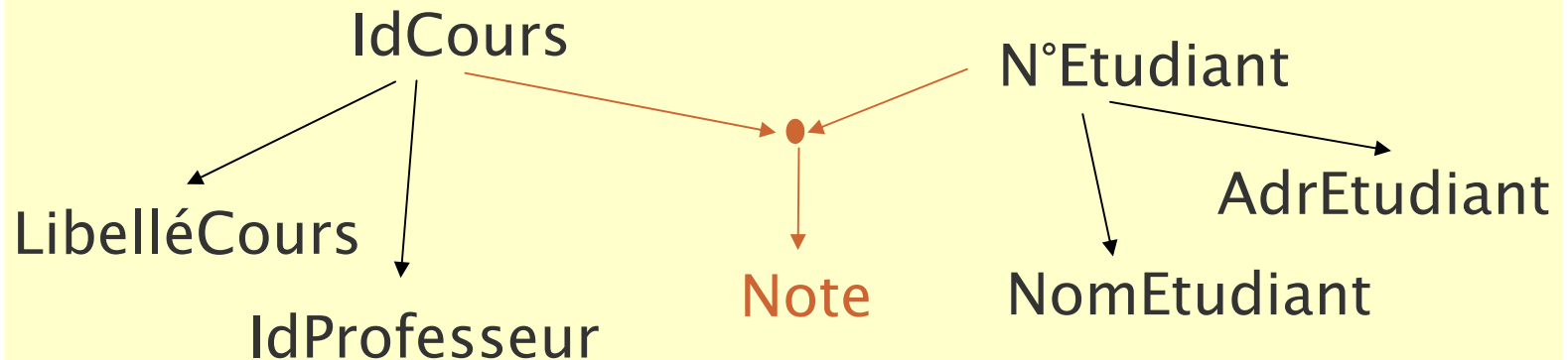
IdCours, N°Etudiant → Note

Heure, N°Salle → IdCours

N°Etudiant, Heure → N°Salle

- Placer les DFs avec comme **source** un **sous-ensemble** d'attributs déjà source de DF

IdCours, N°Etudiant → Note

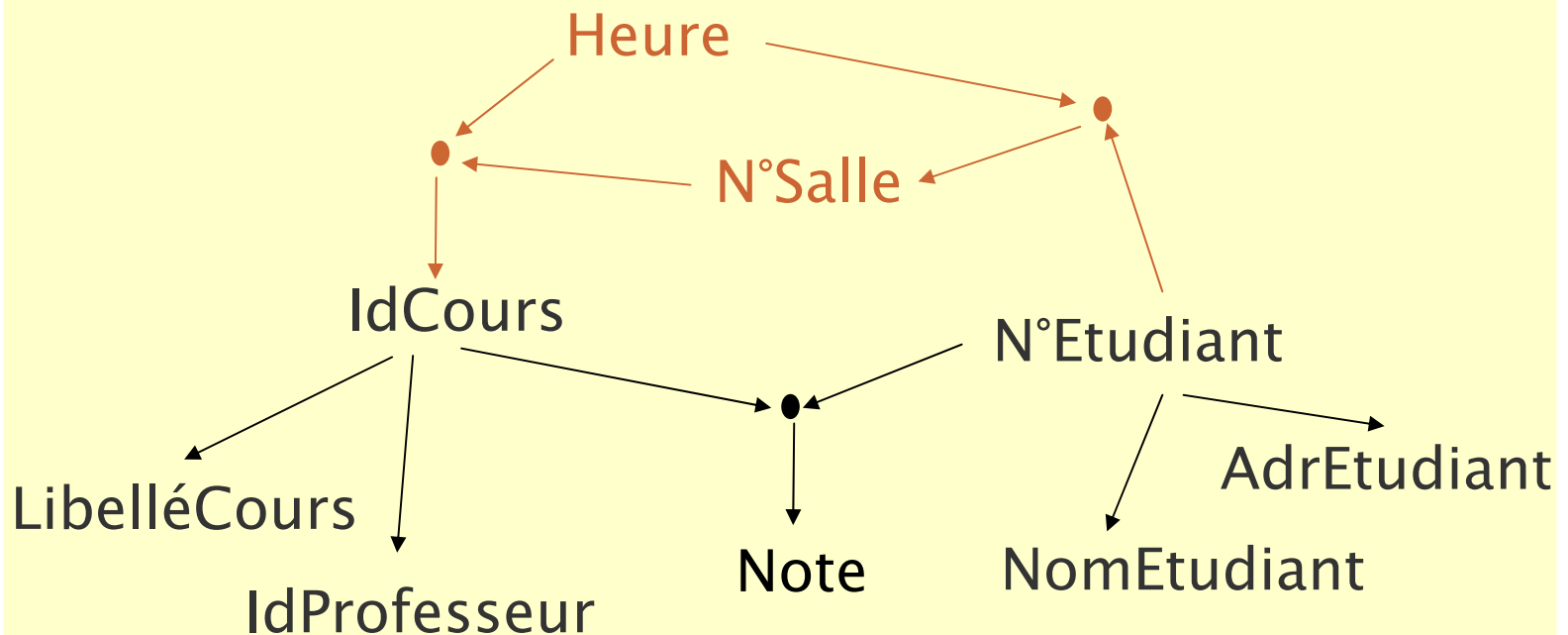


Couverture minimale – Construction (7)

- Intégrer les DFs où l'un des attributs non affectés apparaît comme **source**

Heure, N°Salle \rightarrow IdCours

N°Etudiant, Heure \rightarrow N°Salle



PLAN

■ Introduction

■ Concepts de base

Attribut, Domaine, Produit cartésien, T-uple, Relation, Schéma, Contraintes d'intégrité

■ Dépendances fonctionnelles

■ Décomposition

■ Formes Normales

1NF, 2NF, 3NF, BCNF, 4NF, 5NF

■ Couverture minimale

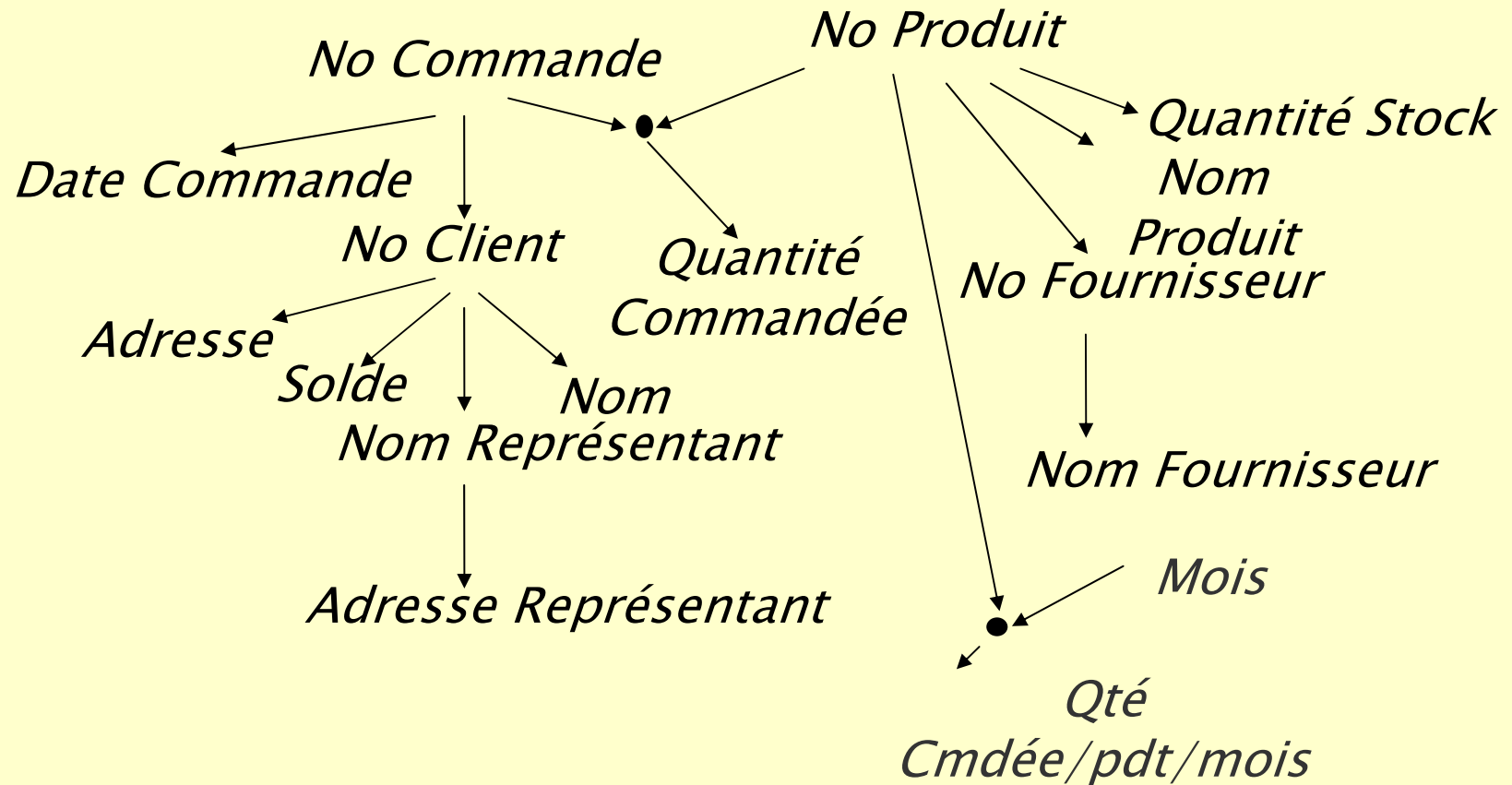
Définition

Construction



De la couverture au modèle relationnel

De la couverture minimale au schéma relationnel (1)



De la couverture minimale au schéma relationnel (2)

Etape 1

➔ Chaque **rubrique source** de DF est placée dans **une table relationnelle** dont la rubrique est **clé**

Commande(N°Commande,
Client(N°Client,
Représentant(NomReprésentant,
Produit(N°Produit,
Fournisseur(N°Fournisseur,

De la couverture minimale au schéma relationnel (3)

Etape 2

➔ Chaque **groupe de rubrique source** de DF est traduit en **une table relationnelle** dont la **clé** est composée des rubriques sources de la DF.

Commande(N°Commande,
Client(N°Client,
Représentant(NomReprésentant,
Produit(N°Produit,
Fournisseur(N°Fournisseur,
LigneCommande(N°Commande,N°Produit,
QtéCmdée(N°Produit, Mois,

De la couverture minimale au schéma relationnel (4)

Etape 3

➔ Chaque rubrique **but** de DF (directe) est placée dans la table créée pour la **source** de la DF

Commande(N°Commande, DateCommande
Client(N°Client, Nom, Adresse, Solde
Représentant(NomReprésentant, AdresseReprésentant
Produit(N°Produit, NomProduit, QuantitéStock
Fournisseur(N°Fournisseur, NomFournisseur
LigneCommande(N°Commande, N°Produit, QuantitéCommandée)
QtéCmdée(N°Produit, Mois, QtéCmdée/pdt/mois)

De la couverture minimale au schéma relationnel (5)

Etape 4

➔ Chaque **rubrique source et but** de DF est notée comme **clé étrangère** de la table relationnelle de la rubrique dont elle est but.

Commande(N°Commande, DateCommande, N°Client)
 Client(N°Client, Nom, Adresse, Solde, NomReprésentant)
 Représentant(NomReprésentant, AdresseReprésentant)
 Produit(N°Produit, NomProduit, QuantitéStock, N°Fournisseur)
 Fournisseur(N°Fournisseur, NomFournisseur)
 LigneCommande(N°Commande, N°Produit,
 QuantitéCommandée)
 QtéCmdée(N°Produit, Mois, QtéCmdée/pdt/mois)