Base de données

- 1. BD et SGBD
- 2. Modèle conceptuel de données (E/A)
- 3. Modèle relationnel de données
- 4. Algèbre relationnel et CRT
- 5. Le langage SQL
- 6. PL/SQL

1

1. BD et SGBD
a) Limite des systèmes de gestion de fichiers

Pour une application

- Données ⇒ risque de **redondance**
- Programmes ⇒ **complexité / cohérence / organisation physique des données**
- Gestion des **accès multiples**

Notion de base de données (début 1960)

2

1. BD et SGBD
b) Définitions

➢ **Base de données** (BD) : représente une collection de données

- **Cohérentes** et **structurées**
- **Indépendantes** de l'application
- De **redondances** minimales
- **Accessibles** (multi-utilisateurs)

➢ **Système de Gestion de Base de données** (SGBD)

Logiciel assurant **structuration, stockage, mise à jour** et **consultation** d'une base de données

➢ **Nombreux domaines d'applications**

Gestion, transactionnel, librairies électroniques, documentations…

3

1. BD et SGBD
c) Objectifs

➢ **Indépendance** données / programmes (**physique** et **logique**)

➢ **Manipulation** « aisée » pour non informaticien (**interrogation** et **mise à jour**)

➢ **Administration** « aisée » des données (**outils d'interrogation**)

➢ **Partage** des données (**plusieurs applications**)

➢ **Efficacité** des accès aux données (**débits** et **temps de réponse**)

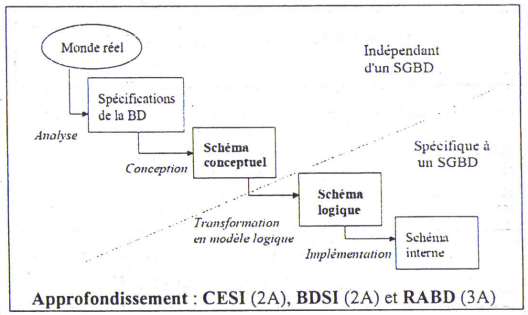
4

1. BD et SGBD
d) Fonctions

- Langage de Définition des Données (LDD) (**objets, attributs, liens** et **contraintes**) ⇒ **Schéma d'une BD**
- Langage de Manipulation des Données (~~LMD~~ LMD) (**créer, modifier, consulter** et **mise à jour**) ⇒ Langage **SQL + interfaces**
- Contrôle de **l'intégrité** des données ⇒ **Schéma** ou **programme**
- Sécurité de fonctionnement ⇒ **Transaction** et « **journalisation** »
  - accès concurrents
  - confidentialité

5

1. BD et SGBD
e) Conception



Approfondissement : **CESI** (2A), **BDSI** (2A) et **RABD** (3A)

6

2. Modèle conceptuel de données (E/A)
a) Principe

Proposé par Shen en 1976

Représentation graphique **Entité/Association** très utilisée pour décrire les données d'un **système d'information** (SI)

Permet ensuite facilement de modéliser les tables de la BD (SQL)

7

2. Modèle conceptuel de données (E/A)
b) Entités

Entité : objet concret ou abstrait à propos duquel on souhaite conserver des informations

Exemple : étudiant, client, département…

Type d'entité : classe de toutes les entités de même nature et jouant le même rôle

Exemple : employé

Occurrences d'un type entité : tout individu particulier faisant partie du type entité

Exemple : les employés

Térieur Alex
Auchon Paul
Pasdanslavabo Onorine
…

8

2. Modèle conceptuel de données (E/A)
c) Attributs

Attributs : **propriétés rattachées à une entité (ou une association)**

Exemple : *nom, adresse, âge...*

« **Type** » **d'attribut** : **nom** et **domaine** (ensemble des valeurs possibles)

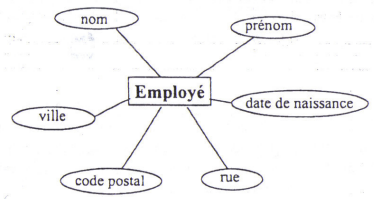
Exemple : *Prix des produits : nombre réel positif*
Nom : chaînes de caractères alphabétiques
Module : {IB1, IB2, IB3, IA1 ...}

Occurrence d'attribut : **valeur particulière**

Exemple : *rouge, vert, bleu : occurrences de l'attribut Couleur*

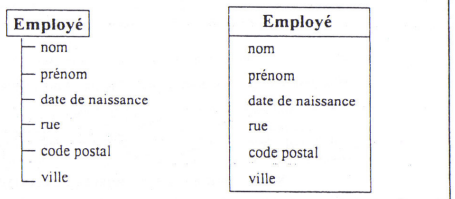
9

2. Modèle conceptuel de données (E/A)
Représentation graphiques Entité et Attributs



10

2. Modèle conceptuel de données (E/A)
Représentation graphiques Entité et Attributs

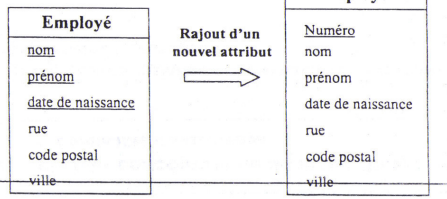


11

2. Modèle conceptuel de données (E/A)
d) Identifiant d'entité

Identifiant : ensemble **minimal** d'attributs caractérisant de façon **unique** les occurrences d'un type entité.

Ex : {nom, prénom, date de naissance}



12

2. Modèle conceptuel de données (E/A)
e) Associations

Association : **liaison** existant entre entités

Une association peut concerner **plusieurs entités** : binaire, ternaires…

⇒ Si une seule : préciser le **rôle** de l'association (*ex : marié à*)

Une association peut posséder des **attributs**

Exemple : date de la commande et quantité de produits commandés

Occurrence association ⇒ entités (ref) + associations

13

2. Modèle conceptuel de données (E/A)
f) Cardinalité d'association

Cardinalité : nombres **minimum** et **maximum** de « participations » de chaque entité à l'association.

Possibilités : min ⇒ 0,1 et max ⇒ 1, N ou M

En fonction des cardinalités maximales :
associations **1-1, 1-N** et **N-M**

14

2. Modèle conceptuel de données (E/A)
Association 1-1

Association 1-1 : une association A entre deux entités E et F est de cardinalité 1-1 si une occurrence de E ne peut être associée qu'à une occurrence de F et réciproquement

Exemple :

Un client est titulaire d'au plus un compte

Un compte n'est attribué qu'à un seul client;
un compte ne peut exister que si un client lui est associé

15

2. Modèle conceptuel de données (E/A)
Association 1-N

Association 1-N : une association A entre deux entités E et F est de cardinalité 1-N si une occurrence de F peut être associée à plusieurs occurrences de E et qu'une occurrence de E ne peut être associée qu'à une occurrence de F.

Exemple :

Un produit est fourni par un seul fournisseur et un seul.

Un fournisseur peut fournir un nombre quelconque de produits.

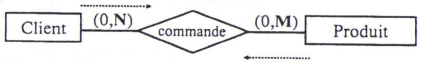
16

2. Modèle conceptuel de données (E/A)
Association N-M

Association N-M : une association A entre deux entités E et F est de cardinalité N-M si une occurrence de F peut être associée à plusieurs occurrences de E et réciproquement.

Exemple :

Un client peut commander plusieurs produits.



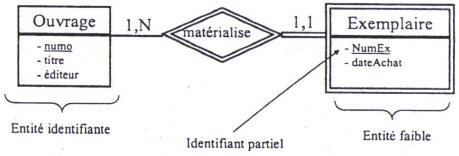
Un produit peut être commandé par plusieurs clients.

2. Modèle conceptuel de données (E/A)
h) Entités faibles

Entité faible : entité mais **pas d'identifiant** propre.

⇒ N'existe qu'en référence à une **entité identifiante**.

L'association qui les unit est dite **association identifiante** avec une contrainte de **cardinalité 1,1**.



2. Modèle conceptuel de données (E/A)
i) Contraintes d'intégrité

But : spécifier des contraintes de la réalité perçue qui ne **sont pas exprimables** dans le modèle E/A.

Contraintes d'intégrités (CI) : propriétés que doivent satisfaire les données.

CI statiques : doivent être vérifiées à tout moment

Date naissance < date mariage

Fax : attributs facultatifs

Nom : obligatoire

CI dynamiques

Exemple: salaire : croissant

Contraintes « programmées » : trigger, interface…

2. Modèle conceptuel de données (E/A)
j) Règles de complétude

Pour qu'un schéma soit **complet** chaque classe d'objets doit posséder **toutes les propriétés requises** obtenues par le modèle E/A

✓ ***Entité***

Nom	Définition (contexte)	Liste d'attributs	identifiant
auteur	livre référencé par l'éditeur	{nom, prénom, adr}	{nom,prénom}

✓ ***Association***

Nom	Définition	Entités	Rôles / cardinalité	attributs
Ecriture	L'écriture associe les livres à l'auteur qui les a écrits	auteur livre	auteur écrit 1-N livre livre est écrit par 1-1 auteur	∅

Remarque : descriptions entités, associations et cardinalités ⇒ **univers du discours** (résumé synthétique de l'application)

2. Modèle conceptuel de données (E/A)
Règles de complétude

✓ *attributs*

Nom	Définition	structure	Rôles / cardinalité	Obligatoire?
ville-aut	Nom de la ville où réside un auteur	atomique (mono-valué)	Chaîne de caractères alphabétiques	Non

✓ *Contraintes d'intégrité*

Nom	Éléments	Expression
Existence d'un mariage	Association Mariage Attributs : âge, sexe de l'entité Personne	Une occurrence de l'association Mariage n'est valide que si âge(P1)≥18 et âge(P2)≥18 et sexe(P1) ≠ sexe(P2)

⇒ **Dictionnaire des données**

21

Exemple 1 : cas d'étude

Spécifications du système d'informations

- Les clients sont caractérisés par un numéro de client, nom, prénom, date de naissance, rue, code postal et ville.
- Les clients commandent des produits à une date donnée et dans une quantité donnée.
- Les produits sont décrits par un numéro de produit, libellé et prix unitaire.
- Chaque produit est fourni par un fournisseur unique
- Un fournisseur peut fournir plusieurs produits
- les fournisseurs sont décrits par un numéro de fournisseur et leur raison sociale.

22

Exemple 1 : 1. Recherche des (types d') entités

Spécifications du système d'informations

Les clients sont caractérisés par un numéro de client, nom, prénom, date de naissance, rue, code postal et ville.

Les clients commandent des produits à une date donnée et dans une quantité donnée.

Les produits sont décrits par un numéro de produit, libellé et prix unitaire.

Chaque produit est fourni par un fournisseur unique

Un fournisseur peut fournir plusieurs produits

les fournisseurs sont décrits par un numéro de fournisseur et leur raison sociale.

23

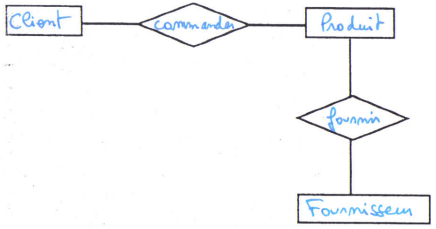
Exemple 1 : 2. Recherche des (types d') associations

Spécifications du système d'informations

- Les clients sont caractérisés par un numéro de client, nom, prénom, date de naissance, rue, code postal et ville.
- Les clients commandent des produits à une date donnée et dans une quantité donnée.
- Les produits sont décrits par un numéro de produit, libellé et prix unitaire.
- Chaque produit est fourni par un fournisseur unique
- Un fournisseur peut fournir plusieurs produits
- les fournisseurs sont décrits par un numéro de fournisseur et leur raison sociale.

24

Exemple 1 :
Représentation graphique



25

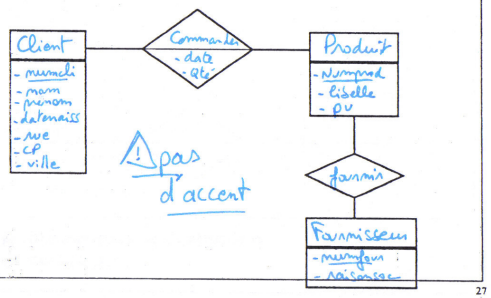
Exemple 1 :
3. Identifications(*) des attributs

Spécifications du système d'informations

- Les clients sont caractérisés par un numéro de client, nom, prénom, date de naissance, rue, code postal et ville.
- Les clients commandent des produits à une date donnée et dans une quantité donnée.
  Les produits sont décrits par un numéro de produit, libellé et prix unitaire.
  Chaque produit est fourni par un fournisseur unique
- Un fournisseur peut fournir plusieurs produits
  les fournisseurs sont décrits par un numéro de fournisseur et leur raison sociale.

26

Exemple 1 :
Représentation graphique



27

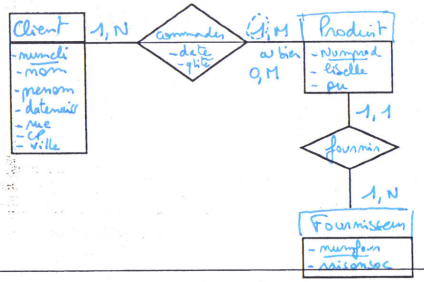
Exemple 1 :
4. Cardinalités

Spécifications du système d'informations

- Les clients sont caractérisés par un numéro de client, nom, prénom, date de naissance, rue, code postal et ville.
- Les clients commandent des produits à une date donnée et dans une quantité donnée.
- Les produits sont décrits par un numéro de produit, libellé et prix unitaire.
  Chaque produit est fourni par un fournisseur unique
- Un fournisseur peut fournir plusieurs produits
- les fournisseurs sont décrits par un numéro de fournisseur et leur raison sociale.

28

Exemple 1 : Représentation graphique



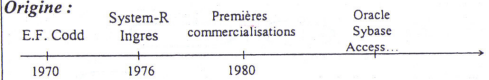
3. Passage au Modèle Relationnel
a) Historique

Principe :

Base de données vue par l'utilisateur : **ensemble de tables**

Modèle logique associé aux SGBD relationnels

Origine :



Modèle relationnel
- **Concepts** (tables, attributs, domaine…)
- **Langage de manipulation des données**
  - algèbre relationnel
  - langage prédicatif : CRT

3. Passage au Modèle Relationnel
b) notion de relation

Relation R avec n (arité) attributs ⇒ $R(A_1, A_2, \dots A_n)$

Exemple : Produit (NumProd, libellé, pu)

Base de données relationnelles ⇒ ensemble de relations reliées entre elles

De manière similaire (MCD)

Attributs ⇒ Nom, domaine + possibilité de valeur Nulle

Un **tuple** d'une relation $R(A_1, A_2, \dots A_n)$ est un ensemble de valeurs $<v_1, v_2, \dots, v_n>$ tq. $v_i \in Dom(A_i)$ ou soit Null

Exemple : <10, écrou, 2> est un tuple de la relation Produit

3. Passage au Modèle Relationnel
Extension de relation

Extension de relation ⇒ ensemble des tuples (tableau 2D)
Extension définie par nom de la relation, liste d'attributs + domaine et CI.

~ 1925

Dans (rouge, blanc, rosé, gris,vert)

Exemple : Vin(Cru, Millésime, Région, couleur)

Clé(CI) ?

Cru, millésime, Couleur

Cru	Millésime	Région	Couleur
Chenas	1983	Beaujolais	Rouge
Tokay	1980	Alsace	Blanc
Tavel	1986	Rhône	Rosé
Chablis	1986	Bourgogne	Blanc
St-émilion	1987	Bordelais	Rouge

attribut

tuple

3. Passage au Modèle Relationnel
c) Notion de clés

Clé d'une relation : groupe **minimum** d'attributs qui identifie de manière **unique** un tuple dans la relation

Toute relation doit avoir **au moins une** clé documentée (et jamais Nulle!), on parle **de clé primaire**. – soulignée dans schéma –

Clé étrangère d'une relation : attribut(s) constituant la clé primaire d'une autre relation. – *italique* ou # dans schéma –

Les clés étrangères définissent les CI référentielles.

33

3. Passage au Modèle Relationnel
Notion de clés (exemple)

Soit la base de données ESIAL1A suivante :

Buveur (nb, nom, prénom)

Vin (nv, cru, millésime, degré)

Abus (*nb*, *nv*, date, quantité)

Abus.nv est une clé étrangère référençant Vin.nv
Abus.nb est une clé étrangère référençant Buveur.nb

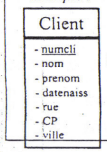
34

3. Passage au modèle relationnel
Étape 1 : entité

Chaque entité donne une relation du même nom

Attributs de entité ⇒ **Attributs de la relation**
Seul les **attributs simples** des attributs composés sont inclus
Identifiant de l'entité ⇒ **Clé primaire de la relation**

Exemple :



⇒ **Client** (numcli, nom, prenom, datenaiss, rue, CP ,ville)

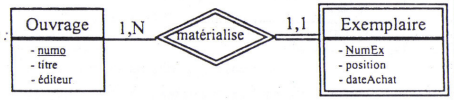
35

3. Passage au modèle relationnel
Étape 2 : entité faible

Chaque **entité faible I** donne une **relation R** qui contient **tous les éléments** de **I** et la **clef primaire** de **l'entité identifiante**

⇒ La **clé primaire de I** est la concaténation de la clé partielle de I et de la **clé primaire** de l'entité identifiante

Exemple :



Ouvrage (numo, titre, éditeur)
Exemplaire (numo, Numex, position, dateAchat)

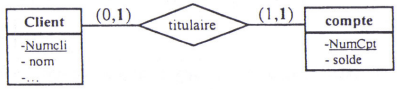
36

3. Passage au modèle relationnel

Étape 3 : association de type 1-1

Chaque **association de type 1-1** est prise en compte en incluant la **clé primaire** d'une relation comme **clé étrangère** dans l'autre.

Exemple :



1. { Compte(NumCpt,solde)
   Client(Numcli, nom, prénom, rue, datenaiss, CP, ville, ***Numcpt***)
2. { Compte(NumCpt, solde, ***Numcli***)
   Client(Numcli, nom, prénom, rue, CP, ville)

Si **cardinalité minimale = 1** : on peut fusionner les relations + attributs de l'association et choisir une des deux clefs comme clé primaire

Client(Numcli, nom, prénom, rue, CP, ville, ***Numcpt*, solde**)

37

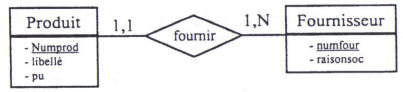
3. Passage au modèle relationnel

Étape 3 : association de type 1-N

Chaque **association de type 1-N** est prise en compte en incluant la **clé primaire** de la relation de cardinalité comme **clé étrangère** dans l'autre relation.

⇒ Idem pour les attributs de l' association

Exemple :



Entités Produit(numProd, libellé, pu, ***Numfour***) *association (lien 1-N)*
Fournisseur(numFour, raisonSoc)

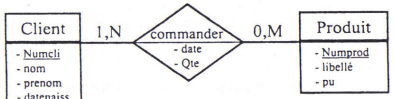
38

3. Passage au modèle relationnel

Étape 3 : association de type N-M

Chaque **association de type N-M** donne lieu à une **nouvelle relation** incluant :
- Les clefs primaires des entités (relations participantes)
- les attributs de l'association

⇒ Choix d'une clé minimale parmi ces attributs



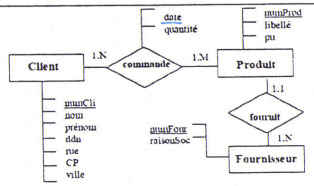
Client(Numcli, nom, prenom, datenaiss, rue, CP, ville)
Produit(NumProd, libellé, pu)

Commander(***numcli***, ***Numprod***, date, Qte)

39

3. Passage au modèle Relationnel

Retour sur l'exemple 1



Entités
Client(numcli, nom, prénom, ddn, rue, CP, ville)
Produit(numProd, libellé, pu, ***Numfour***) *association (lien 1-N)*
Fournisseur(numFour, raisonSoc)

Association (lien N-M) Commande(***numCli***, ***numProd***, date, quantité)

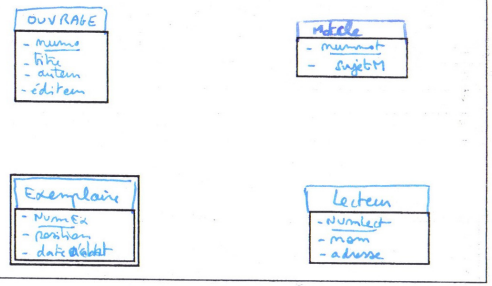
40

Récapitulatif
Exemple 2

- Un *ouvrage* est caractérisé par un numéro identifiant, un titre, un auteur et un éditeur.
- Un ouvrage est aussi décrit par une liste de *mots-clés* numérotés précisant les sujets qui y sont traités.
- La biblio. dispose d'un ou plusieurs *exemplaires* de chaque ouvrage. L'exemplaire est identifié par un numéro et est caractérisé par sa position dans les rayons et sa date d'achat.
- Un exemplaire peut être emprunté par un *lecteur* pouvant en emprunter plusieurs.
- Un lecteur est décrit par un numéro, son nom et son adresse.

41

Exemple 2
entités + attributs



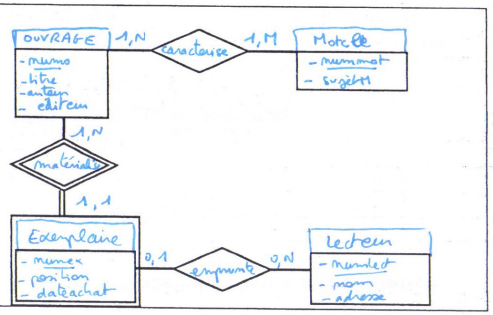
42

Exemple 2
Associations + cardinalité

- Un *ouvrage* est caractérisé par un numéro identifiant, un titre, un auteur et un éditeur.
- Un ouvrage est aussi décrit par une liste de *mots-clés* numérotés précisant les sujets qui y sont traités.
- La biblio. dispose d'un ou plusieurs *exemplaires* de chaque ouvrage. L'exemplaire est identifié par un numéro et est caractérisé par sa position dans les rayons et sa date d'achat.
- Un exemplaire peut être emprunté par un *lecteur* pouvant en emprunter plusieurs.
- Un lecteur est décrit par un numéro, son nom et son adresse.

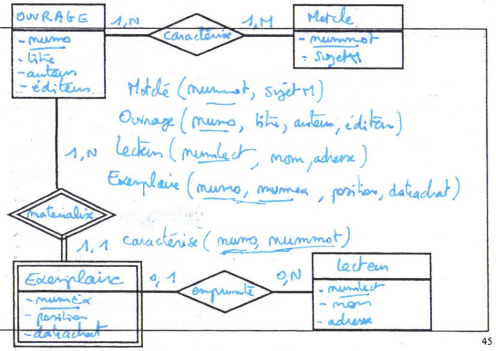
43

Exemple 2
Entités + Attributs et cardinalités



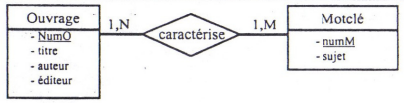
44

Modèle relationnel



45

Exemple « Matérialisation » d'une relation



Ouvrage(NumO, Titre, Auteur, Editeur)

NumO	Titre	Auteur	Editeur
1	La bête humaine	Hugo	Livre de Poche
2	Gaston Lagaffe – 3	Franquin	Dupuis

MotClé(numM, sujet)

numM	sujet
1	Drame
2	Réalisme
3	BD

Caractérise(NumO,NumM)

numO	numM
1	1
1	2
2	3

46

Au « menu » du prochain cours

- Formes normales (redondance)
- Algèbre relationnel
- Calcul relationnel de tuples
- « Suivant le temps » : un peu de SQL

47

3. Passage au modèle relationnel
d) Redondances

Exemple : table « employé »

Nom	Prénom	Dep	Cdep	Grade	Salaire	Sit.	Adresse
Tron	Paul	Info	C1	E1	1500	∅	20 rue… 54000 Nancy
Mene	Jean	Sec	C2	E2	1750	Marié	23 rue… 75000 Paris
Oré	Eric	Info	C1	E1	1500	∅	14 rue… 31000 Toulouse
Tron	Paul	Sec	C2	E1	1500	Divo.	

Redondance dans les relations

- augmentation de la **quantité** de données,
- **temps de calcul**,
- problème de **mise à jour**,
- perte **d'informations**,
- **incohérences…**

⇒ **Normalisation des données**

1

3. Passage au modèle relationnel
e) Dépendances Fonctionnelles (DF)

Soit R(X,Y,Z) une relation où X, Y et Z sont des ensembles d'attributs avec Z pouvant être vide.

Définition : Y dépend **fonctionnellement** de X si la valeur de X détermine la valeur de Y dans **toute extension** de R.

Notation : X → Y

X est une **clé** ssi : X → (Y,Z) et X est **minimal**

Exemple : Produits(numProd, libellé, pu)

numProd → libellé
numProd → pu

2

3. Passage au modèle relationnel
e) Propriétés des DF

Dans ce qui suit X, Y, Z… sont des ensembles d'attributs

Réflexivité :
Si Y ⊆ X alors X→ Y
Augmentation :
Si X → Y et Z ⊆ W alors X, Z → Y,W
Union :
Si X → Y et X → Z alors X → Y,Z
Transitivité :
Si X → Y et Y→ Z alors X → Z
Pseudo transitivité :
Si X → Y et Y,Z → T alors X,Z → T
Décomposition :
Si X → Y et Z⊆ Y alors X → Z

⇒ Module TC CESI 2A

3

3. Passage au modèle relationnel
f) Forme normale - 1ère

Définition : une relation est en première forme normale si chacun de ses attributs a un domaine **atomique** (non décomposable) et **monovalué**.

Exemple 1 : la relation Personne(CodeP, nom, prénom, adresse) n'est pas en 1NF…

car adresse → num, rue, ville, CP.

L'attribut « Adresse » sera considéré comme un attribut simple en TD.

Exemple 2 : la relation Personne(CodeP, nom, prénoms, âge) n'est pas en 1NF…

car prenoms → prenom1, prenom2, …

4

3. Passage au modèle relationnel
f) Forme normale - 2ème

Définition : une relation est en deuxième forme normale si elle est en **1NF** et si **tout attribut n'appartenant pas à la clé ne dépend pas d'une partie de la clé**.

Exemple 1 : la relation Personne(CodeP, nom, prénom, rue, CP, ville) est en 2NF…

Exemple 2 : la relation Stock(NumProd, NumpDép, libelléProd, QtéStockée) n'est pas en 2NF.

car NumProd → libelléProd

la relation « stock » peut être décomposée en 2 relations

Produits (NumProd, libelléProd)

Stock2 (NumProd, NumDep, QtéStockée)

5

3. Passage au modèle relationnel
f) Forme normale - 3ème

Définition : une relation est en troisième forme normale si elle est en 2NF et si il n'existe pas de DF entre attributs non clés.

R (A, B, C, D, E, F) Non 3 NF

Exemple : la relation Avion(no-avion, constructeur, type, capacité) n'est pas en 3NF…

car type → constructeur et type → capacité

Avion peut être décomposé en 2 relations

Avion 2 (no-avion, type)

Modèle (type, constructeur, capacité)

6

3. Passage au modèle relationnel

f) Forme normale – Normalisation (théorème)

Toute relation a au moins une décomposition en 3ème forme normale qui est sans perte d'information et qui préserve les dépendances fonctionnelles.

La relation initiale est obtenue par jointure naturelle des relations issues de sa décomposition.

Si dans une relation R on peut trouver trois ensembles d'attributs A, B et C tels que A → B alors R peut être décomposée en deux relations R1(A,B) et R2(A,C) sans perte d'information tout en préservant la dépendance.

3. Passage au modèle relationnel

f) Forme normale – Normalisation (exemple)

Soit la relation **FILM**(no-exploitation, titre, réalisateur)
avec les DF : no-exploitation → titre et titre → réalisateur

Sous quelle forme normale est cette relation?

1NF? **Oui** car tous les attributs sont simples

2NF? **Oui si on choisit** no-exploitation **comme clé primaire**

On a : no-exploitation → titre
titre → réalisateur

et par transitivité : no-exploitation → réalisateur

3NF? **Non** car il existe une DF entre deux attributs non clés

3. Passage au modèle relationnel

f) Forme normale – Normalisation (exemple)

Si dans une relation R on peut trouver trois ensembles d'attributs A, B et C tels que A → B alors R peut être décomposée en deux relations R1(A,B) et R2(A,C) sans perte d'information tout en préservant la dépendance.

FILM(no-exploitation, titre, réalisateur) A, B et C?

Décomposition :

FILM1(no-exploitation, titre)

FILM2(titre, réalisateur)

- Ces relations sont en 3NF et toutes les DF sont conservées

- Il y a une « lourdeur » de données au niveau du titre (un code est préférable) mais elle est « contrôlée » et permet d'éviter les problèmes de mise à jour

3. Passage au modèle relationnel

f) Forme normale – Remarques

Algorithme de partitionnement des DF sous forme d'un graphe

3NF ne garantit pas une absence de redondance!

Modèle(type, constructeur, capacité)

Contrat(Codepersonne, numassurance, numsociété, date)

Autres formes normales :

- **3ème forme normale BOYCE-CODD (3 BCNF);**
- **4ème forme normale;**
- **5ème forme normale**

⇒ CES1 2A

4. Algèbre relationnelle

a) Opérateurs - catégories

Opérateurs ensemblistes ⇒
- Union : ∪
- Intersection : ∩
- Produit cartésien : ×
- Différence : −

Opérateurs spécifiques ⇒
- Projection : Π
- Restriction : σ
- Jointure : ∞
- Division : %
- Agrégation...

4. Algèbre relationnelle

b) Opérateurs ensemblistes - Union

Soit : R1(A1, A2, ..., An) **compatible** avec R2(B1, B2, ..., Bn)
⇒ **même domaine syntaxique et sémantique**

L'union est une relation R3, de schéma R3(A1, A2, ..., An), qui contient les tuples (n-uplets) de R1 et ceux de R2

Représentation symbolique : R3 ← R1 ∪ R2

Représentation graphique :

Représentation par langage : R1 union R2 giving R3
R3 ← union(R1,R2)

4. Algèbre relationnelle
b) Opérateurs ensemblistes - Union (exemple)

Chauffeur (code_C, Nom)

145	Antoine
568	Maurice

Secrétaire (Id_S, Nom)

5609	Lucette
7534	Noëlle
568	Maurice

R3 ← **Chauffeur** ∪ **Secrétaire** (code_C, Nom)

145	Antoine
568	Maurice
5609	Lucette
7534	Noëlle

4. Algèbre relationnelle
b) Opérateurs ensemblistes - Intersection

Soit : R1(A1, A2, ..., An) **compatible** avec R2(B1, B2, ..., Bn)
L'intersection entre R1 et R2 est une relation R3, de schéma R3(A1, ..., An), qui contient les tuples de R1 qui sont aussi dans R2.

R.S. : R3 ← R1 ∩ R2

R.G. :

R.L. : R1 intersect R2 giving R3
R3 ← intersect(R1,R2)

4. Algèbre relationnelle
b) Opérateurs ensemblistes - Intersection (exemple)

Chauffeur (code_C, Nom)

145	Antoine
568	Maurice

Secrétaire (Id_S, Nom)

5609	Lucette
7534	Noëlle
568	Maurice

R3 ← **Chauffeur** ∩ **Secrétaire**

R3 (code_C, Nom)

568	Maurice

4. Algèbre relationnelle
b) Opérateurs ensemblistes - Différence

Soit : R1(A1, A2, ..., An) **compatible** avec R2(B1, B2, ..., Bn)
Le **différence** est une relation R3, de schéma R3(A1, ..., An), qui contient les tuples de R1 qui n'apparaissent pas dans R2

R.S. : R3 ← R1 – R2

R.G. :

R.L. : R1 minus R2 giving R3
R3 ← minus(R1,R2)

4. Algèbre relationnelle
b) Opérateurs ensemblistes - Différence (exemple)

Chauffeur (code_C, Nom)

145	Antoine
568	Maurice

Secrétaire (Id_S, Nom)

5609	Lucette
7534	Noëlle
568	Maurice

R3 ← **Chauffeur** – **Secrétaire**

R3 (code_C, Nom)

145	Antoine

4. Algèbre relationnelle
b) Opérateurs ensemblistes - Produit cartésien

Soit : R1(A1, A2, ..., An) **compatible** avec R2(B1, B2, ..., Bk)
Le **produit cartésien** entre R1 et R2 est une relation R3 de schéma R3(A1, ..., An, B1, ..., Bk) contenant toutes les concaténations des tuples de R1 avec ceux de R2.

R.S. : R3 ← R1 × R2

R.G. :

R.L. : multiply R1 by R2 giving R3
R3 ← product(R1,R2)

4. Algèbre relationnelle
b) Opérateurs ensemblistes - Produit cartésien (exemple)

Chauffeur (code_C, Nom)

145	Antoine
568	Maurice

Secrétaire (Id_S, Nom)

5609	Lucette
7534	Noëlle
568	Maurice

R3 ← Chauffeur × Secrétaire

R3 ← (code_C, C.Nom, Td_S, S.Nom)

145	Antoine	5609	Lucette
145	Antoine	7534	Noëlle
145	Antoine	568	Maurice
568	Maurice	5609	Lucette
568	Maurice	7534	Noëlle
568	Maurice	568	Maurice

19

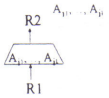
4. Algèbre relationnelle
c) Opérateurs spécifiques - Projection

La projection d'une relation R1 de schéma R1(A_1, A_2, …, A_n) sur les attributs (A_{i1}, A_{i2}, …, A_{ik}) (avec n ≥ k) est une relation R2 dont le schéma comporte :

- les **attributs** $A_{ij=1,k}$,
- les **tuples de R1 restreints aux valeurs des** $A_{ij=1,k}$ (sans doublon).

R.S. : R2 ← $\Pi_{A_{i1}, \ldots, A_{ik}}$ (R1)

R.G. :



R.L. : project R1 over $A_{i1}, \ldots, A_{ik}$ giving R2
R2 ← proj(R1; $A_{i1}, \ldots, A_{ik}$)

20

4. Algèbre relationnelle
c) Opérateurs spécifiques - Projection (exemple)

Exemple : soit la table **Personne**

Code_P	Nom	Prénom	Ville
145	Dupont	Antoine	Toulouse
568	Viadieu	Maurice	Toulouse
5609	Dupont	Alain	Toulouse
1616	Durand	Nathalie	Nancy

R2 ← $\Pi_{\text{Nom, ville}}$ (**Personne**)

R2 (Nom, Ville)

Dupont	Toulouse
Viadieu	Toulouse
Durand	Nancy

En algèbre relationnelle,
une relation n'admet **pas de tuples en double**.

21

4. Algèbre relationnelle
c) Opérateurs spécifiques - Restriction / sélection

S'applique à une relation R1 de schéma R1(A1, A2, …, An).
La **restriction** de la relation R1 selon le **prédicat** P est une relation R2 ayant pour attribut (A1, A2, …, An) et contenant les n-uplets de R1 qui vérifient le prédicat P.

R.S. : R2 ← σ_P (R1)

R.G. :

R.L. : select R1 where P giving R2
R2 ← select(R1; P)

22

4. Algèbre relationnelle
c) Opérateurs spécifiques - Restriction (exemple)

Exemple : soit la table **Personne**

Code_P	Nom	Prénom	Ville
145	Dupont	Antoine	Toulouse
568	Viadieu	Maurice	Toulouse
5609	Dupont	Alain	Toulouse
1616	Durand	Nathalie	Nancy

R2 ← $\sigma_{\text{Ville="Nancy"}}$ (**Personne**)

R2 (Code_P, Nom, Prénom, Ville)

1616	Durand	Nathalie	Nancy

23

4. Algèbre relationnelle
c) Opérateurs spécifiques - Jointure

S'applique sur deux relations R1(A1, …, An) et R2 (B1, …, Bk) qui possèdent **au moins un attribut défini sur le même domaine**.
La **jointure** de R1 et R2 selon une **condition C** (prédicat) est une relation R3(A1, …,An,B1,…Bk) qui contient les n-uplets de R1 associés à ceux de R2 qui vérifient C.

R.S. : R3 ← R1 $\downarrow_C$ R2 (ou R3 ← R1 $\bowtie_C$ R2)
(ou R3 ← R1 $\Rightarrow_C$ R2)

R.G. :

R.L. : join R1, R2 where C giving R3
R3 ← join(R1,R2/C)

Remarque : R1 $\downarrow_C$ R2 = σ_C (R1 × R2)

24

4. Algèbre relationnelle
c) Opérateurs spécifiques - Jointure (exemple)

Personne

Code_P	Nom	Prénom	Ville
145	Dupont	Antoine	Toulouse
568	Viadieu	Maurice	Toulouse
5609	Dupont	Alain	Toulouse
1616	Durand	Nathalie	Nancy

Ville_Imp (Nom_Ville, Nb_hab)

Toulouse	650 000
Lyon	2 000 000

JOIN Personne, ville_Imp **WHERE** ville=Nom_ville **GIVING** R

R (Code_P, Nom, Prénom, Ville, Nom_Ville, Nb_hab)

145	Dupont	Antoine	Toulouse	Toulouse	650 000
568	Viadieu	Maurice	Toulouse	Toulouse	650 000
5609	Dupont	Nathalie	Toulouse	Toulouse	650 000

4. Algèbre relationnelle
c) Opérateurs spécifiques - Type de jointure

La **jointure naturelle** de deux relations R et S est l'équi-jointure (opérateur d'égalité) de R et S sur tous les attributs communs.

Notation : R3 ← R1 ↓ R2

Personne

Code_P	Nom	Prénom	Ville
145	Dupont	Antoine	Toulouse
568	Viadieu	Maurice	Toulouse
5609	Dupont	Alain	Toulouse
1616	Durand	Nathalie	Nancy

Ville_Imp (Ville, Nb_hab)

Toulouse	650 000
Lyon	2 000 000

R ← Personne ↓ Ville_Imp

R (Code_P, Nom, Prénom, Ville, Nb_hab)

145	Dupont	Antoine	Toulouse	650 000
568	Viadieu	Maurice	Toulouse	650 000
5609	Dupont	Nathalie	Toulouse	650 000

4. Algèbre relationnelle
c) Opérateurs spécifiques - Type de jointure

La **jointure externe** permet d'inclure les tuples « célibataires » (non pris en compte) dans la jointure naturelle (*jointure interne*).

.) **jointure externe** à gauche R1 = ↓ R2 (+ célibataires de R1)

.) **jointure externe** à droite R1 ↓ = R2 (+ célibataires de R2)

.) **jointure externe** pleine R1 = ↓ = R2 (+ célibataires de R1 et R2)

Produit (CodeP, libellé)

1	Marteau
2	Perceuse
3	Ponceuse

Commande (CodeP, codeCli, qte)

1	2	2
3	13	12
1	4	13

R ← Produit = ↓ Commande

R (CodeP, libellé, codeCli, qte)

1	Marteau	2	2
3	Ponceuse	13	12
1	Marteau	4	13
2	Perceuse	Null	Null

4. Algèbre relationnelle
c) Opérateurs spécifiques - Division

La **division** de la relation $R1(A_1, \ldots, A_n)$ par la relation $R2(A_{p+1}, \ldots, A_n)$ est la relation R3 $(A_1, \ldots, A_p)$ formée de tous les tuples $(A_1, \ldots A_p)$, qui concaténés à chaque tuple de R2, donnent toujours un tuple de R1.

R.S. : R3 ← R1 % R2

R.G. :

R.L. : Divide R1 by R2 giving R3
R3 ← Div(R1,R2)

Division $R(X,Y) \div S(Y) = T(X) = \{ <x> \mid \forall\, y, <y> \in S \Rightarrow <x,y> \in R \}$

4. Algèbre relationnelle
c) Opérateurs spécifiques - Division (exemple)

Commander (Code_C, Lib_Prod)

7534	Vis
7534	Ecrous
5643	Ecrous
5632	Vis
5632	Ecrous
5632	Marteau

Produit (Lib_Prod)

Vis
Ecrous

R ← Commander % Produit

R (Code_C)

7534
5632

4. Algèbre relationnelle
d) Exercice (requête algébrique)

Client(numcli, nom, prénom, ddn, rue, CP, ville)
Produit(numProd, libellé, pu, ***numFour***)
Fournisseur(numFour, raisonSoc)
Commande(***numCli***, ***numProd***, date, qté)

Question : donner la liste des numéros de produits avec leur libellé, leur prix unitaire et la date de commande des produits commandés en quantité supérieure à 100 et dont le prix est inférieur à 1000.

Requête complexe : composition d'opérateurs

R1 ← $\sigma_{pu \leq 1000}$ (Produit)

R2 ← $\sigma_{qte > 100}$ (Commande)

R3 ← R1 ↓ R2

Res ← $\Pi_{numProd,libelle,pu,date}$ (R3)

Version compacte :

Res ← $\Pi_{numProd,libelle,pu,date}$ ($\sigma_{pu \leq 1000}$ (Produit) ↓ $\sigma_{qte > 100}$ (Com

Remarque : si date de commande non spécifiée ⇒ ambiguïté (idem pour unitaire?)

4. Algèbre relationnelle
d) Exercice (à vous de jouer)

Client(numCli, nom, prénom, ddn, rue, CP, ville)
Produit(numProd, libellé, pu, ***numFour***)
Fournisseur(numFour, raisonSoc)
Commande(***numCli***, ***numProd***, date, qté)

Question 1 : libellé et prix unitaire de tous les produits.

$Res \leftarrow \pi_{libellé, pu}(produit)$.

31

4. Algèbre relationnelle
d) Exercice (à vous de jouer)

Client(numCli, nom, prénom, ddn, rue, CP, ville)
Produit(numProd, libellé, pu, ***numFour***)
Fournisseur(numFour, raisonSoc)
Commande(***numCli***, ***numProd***, date, qté)

Question 2 : libellé des produits de prix inférieur à 50.

$R1 \leftarrow \sigma_{pu \leq 50}(produit)$

$Res \leftarrow \pi_{libellé}(R1)$.

32

4. Algèbre relationnelle
d) Exercice (à vous de jouer)

Client(numCli, nom, prénom, ddn, rue, CP, ville)
Produit(numProd, libellé, pu, ***numFour***)
Fournisseur(numFour, raisonSoc)
Commande(***numCli***, ***numProd***, date, qté)

Question 3 : nom et prénom des clients ayant commandé des produits numéro 56

$R1 \leftarrow \sigma_{numProd = 56}(commande)$

$R2 \leftarrow R1 \downarrow Client$

$Res \leftarrow \pi_{nom, prénom}(R2)$

33

4. Algèbre relationnelle
d) Exercice (à vous de jouer)

Client(numCli, nom, prénom, ddn, rue, CP, ville)
Produit(numProd, libellé, pu, ***numFour***)
Fournisseur(numFour, raisonSoc)
Commande(***numCli***, ***numProd***, date, qté)

Question 4 : nom des clients ayant commandé au moins un produit de prix supérieur à 500.

$R1 \leftarrow \sigma_{pu \geq 500}(produit)$

$R2 \leftarrow R1 \downarrow commande$

$R3 \leftarrow R2 \downarrow client$

$Res \leftarrow \pi_{nom}(R3)$

34

4. Algèbre relationnelle
d) Exercice (à vous de jouer)

Client(numCli, nom, prénom, ddn, rue, CP, ville)
Produit(numProd, libellé, pu, ***numFour***)
Fournisseur(numFour, raisonSoc)
Commande(***numCli***, ***numProd***, date, qté)

Question 5 : nom des clients n'ayant pas commandé le produit 56

$R1 \leftarrow \sigma_{numProd = 56}(Commande)$

$R2 \leftarrow \pi_{numCli}(R1)$

$R3 \leftarrow \pi_{numCli}(Client)$

$R4 \leftarrow R3 - R2$

$R5 \leftarrow R4 \downarrow Client$

35

$Res \leftarrow \pi_{nom}(R5)$

4. Algèbre relationnelle
d) Exercice (à vous de jouer)

Client(numCli, nom, prénom, ddn, rue, CP, ville)
Produit(numProd, libellé, pu, ***numFour***)
Fournisseur(numFour, raisonSoc)
Commande(***numCli***, ***numProd***, date, qté)

Question 6 : nom et prénom des clients ayant commandé tous les produits.

$R1 \leftarrow \pi_{numProd}(Produit)$

$R2 \leftarrow \pi_{numProd, numCli}(Commande)$

$R3 \leftarrow R2 \div R1$

$R4 \leftarrow R3 \downarrow Client$

$Res \leftarrow \pi_{nom, prénom}(R4)$

36

4. Algèbre relationnelle
e) Agrégation

Permet d'effectuer des calculs sur les attributs d'une relation

R.S. : R2 ← AGG (R1; X; B)

R.G. : R2 ↑ [AGG : X, B)] ↑ R1

R.L. : AGG (R1;X;B) giving R2

Où : - AGG ∈ {SUM, COUNT, MIN,MAX, AVG,...}
- X : regroupement
- B : l'attribut agrégé

4. Algèbre relationnelle
e) Agrégation (exemple)

Personne (Code_P, Nom, Prénom, Ville, Age)

Code_P	Nom	Prénom	Ville	Age
1616	Dupont	Antoine	Toulouse	30
568	Valade	Julien	Toulouse	20
568	Dupont	Lucette	Toulouse	25
568	Durand	Joël	Nancy	30

R ← COUNT (Personne; Ville; Code-C) **R** (Ville, count(Code_c))

Ville	count(Code_c)
Toulouse	3
Nancy	1

R ← Min (Personne; Ville; Age) **R** (Ville, Min(Age))

Ville	Min(Age)
Toulouse	20
Nancy	30

4. Algèbre relationnelle
e) Agrégation (remarques)

L'agrégation peut ne pas intégrer de critères de regroupement

R ← COUNT (Personne;; Code-C) R (count(Code_c))
4

L'agrégation peut intégrer des prédicats de restriction à réaliser avant ou après l'agrégation.

⇒ Plus clair en SQL ☺

4. Algèbre relationnelle
f) Ensemble minimal d'opérateurs

.) Les requêtes SQL sont **transformées** en **expressions algébriques**

.) **5 opérateurs sont nécessaires** : ∪, −, Π, ×, σ

.) les autres opérateurs sont déductibles des précédents

.) **Règles d'équivalences**

- $\sigma_{c1 \wedge c2}(R) = \sigma_{c1}(\sigma_{c2}(R))$
- $\sigma_{c1}(\sigma_{c2}(R)) = \sigma_{c2}(\sigma_{c1}(R))$
- $\Pi_{L_{n+1}}(\Pi_{L_{n+2}}(R)) = \Pi_{L_{n+1}}(R)$
- $R \downarrow_C S = S \downarrow_C R$
- $R \downarrow_C (S \downarrow_{C'} T) = (S \downarrow_C R) \downarrow_{C'} T$

etc

Propriétés utilisées pour **l'optimisation des requêtes** dans les SGBD.

4. AR et Calcul relationnel de tuples (CRT)
g) Définition

.) **CRT** : autre approche pour manipuler les données relationnelles

.) Langage fondé sur le calcul des prédicats du 1er ordre (CP1)

- Les **variables** sont associées à des **tuples**
- Les **prédicats** comportent
  - Les comparateurs logiques (=, ¬=, >, ≥, <, ≤)
  - Un prédicat unaire associé à chaque relation prédicat et relation : même nom

(CRT très utile pour certaines **requêtes complexes**, écriture « proche » du SQL)

⇒ à partir du 5ème TD

4. AR et Calcul relationnel de tuples (CRT)
h) Application

Soit la BD de schéma :
- Produit(numProd,libellé,pu)
- Depot(numDep,capacité,adresse)
- Stock(numProd, numDep, Qté)

.) le prédicat **Produit(p)** signifie que le tuple **p** appartient à la relation Produit

.) **p.numProd** désigne le numéro du produit **p**

Exemples de requêtes en CRT

a. Nom et prix unitaire de tous les produits

{ p.libellé, p.pu | Produit(p)}

b. Numéro des produits stockés dans le dépôt D2

{ p.numProd | Produit(p) ∧ ∃ s t.q. (Stock(s) ∧ s.numProd = p.numProd ∧ s.numDep = "D2") }

c. Adresse et numéro des dépôts stockant tous les produits

{ d.adr, d.numDep | Depot(d) ∧ ∀ p Produit(p) ⇒ ∃ s t.q. (Stock(s) ∧ s.numProd = p.numProd ∧ s.numDep = d.numDep) }

Division $R(X,Y) \div S(Y) = T(X) = \{ <x> \mid \forall y, <y> \in S \Rightarrow <x,y> \in R \}$

(autre écriture avec deux ¬∃)