

Université de Badji Mokhtar
Département Informatique

Base de Données Avancées BDA

Master Réseau RSI

Cours: Djellali Hayet.

Janvier 2021

1

Base de Données Avancées

Master RSI

Chapitre 1

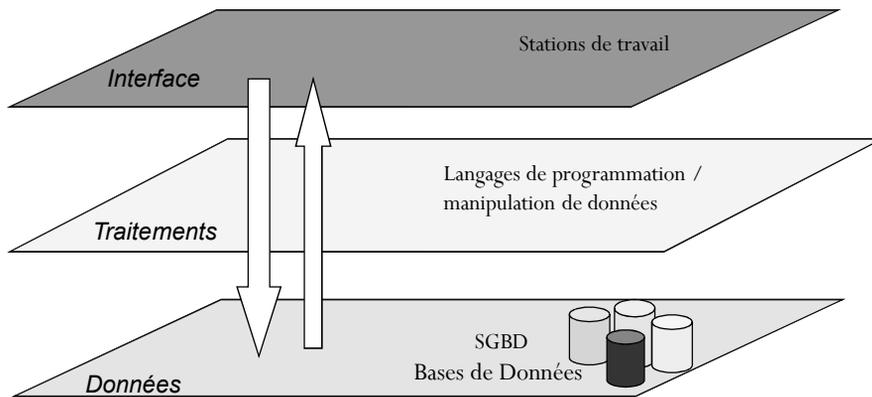
- Rappels de Base de Données 1..6
- Les Systèmes de Gestion de Bases de Données 7..29
- Représentation interne des systèmes de BD
 - structures et mécanisme d'accès
- Modèle de Données 30
- Modèle Entité Association 31
- Modèle Relationnel 48
 - Normalisation des Bases de Données 63
 - Algèbre Relationnels 84
 - Langage SQL
- Modèle de Données en Réseau et DBTG

2

Base de Données Avancées

Master RSI

Les trois dimensions d'une application



3

Qu'est-ce une BD ? Et un SGBD ?

- **Base de Données (BD)**

- Définition "classique":
un ensemble de données structurées et corrélées, enregistrées avec le minimum de redondance et pouvant être traitées par une ou plusieurs applications, de façon optimale. La BD contient également la description de ses données.
- Un utilisateur doit pouvoir les retrouver. Il faut pouvoir les interroger par des requêtes. Les données évoluent, il faut donc pouvoir les manipuler : ajouter, modifier, supprimer des données.

- **Système de Gestion de Bases de Données (SGBD)**

un ensemble de logiciels permettant aux utilisateurs de définir, créer, maintenir, contrôler et accéder à la BD

Le logiciel de base qui permet de manipuler ces données est appelé e un système de gestion de bases de données – SGBD (database management system – DBMS).
Tout système d'information est construit autour de bases de données

4

Depuis quand ?

- Depuis les années '60, avec :
 - les BD hiérarchiques (ex : IMS, Information Management Systems)
 - BD réseaux ou CODASYL (Committee on Data Systems and Languages; ex : IDS, Integrated Data Store)
- BD relationnelles (E.F. Codd, 1970)
- BD objets (ex : o2, Versant, 1990)
- BD hybrides objets-relationnel (ex : Oracle V8 en 1998)
- BD natives XML (ex: Tamino de Software AG, 2000)

Contexte

- Les Bases de Données sont nécessaires à tous les domaines d'activité: industrie, commerce, services, recherche scientifique, ...
- Leur succès est lié aux avancées scientifiques et technologiques en gestion de l'information et des communications
- Poids économique: par exemple le SGBD Oracle a un CA 14,4 Milliards de \$ et 65000 collaborateurs ⁽¹⁾, utilisé par 98 entreprises du classement Fortune 100

Fonctionnalités Principales des Bases de Données

- Mémoriser de grandes quantités de données essentielles à la vie de l'entreprise, les organiser,
- Faciliter les requêtes et permettre l'évolution,
- Donner l'accès à des applications diverses, à des utilisateurs différents avec des modes d'accès variés.

Exemples de grandes applications

- Systèmes de compagnies aériennes
- Systèmes bancaires, d'assurance, commerciaux
- Bases de données scientifiques, techniques
 - Biologie
 - Astronomie
 - Produits industriels
- Bases de données bibliographiques
- et, de plus en plus, interactions entre applications de divers domaines
santé, transports, tourisme, ...

Exemples (1)

Pour une BD pour une compagnie aérienne.

Pour supporter les réservations:

- quelles informations doivent être stockées ?
- quels types d'interrogations sont souhaités ?

Exemples

- Les données
 - les appareils
 - les vols
 - les aéroports
 - les réservations
 - les achats
- Les types d'interrogations
 - quels sont les vols au départ de X et arrivant à Y le 15 mars 2004 ?
 - quels sont les prix de ces vols ?
 - combien de passagers ont voyagé sur le vol 1234 du 15 mars 2004 ?

Exemples

SGBD relationnel: les données sont stockées dans des tables

Exemple :

Vols	<u>n°vol</u>	<u>compagnie</u>	<u>type avion</u>
	523	Air Algerie	Boeing 747
	234	Alitalia Boeing	737

Requête SQL: Donner le type d'avion du vol 123

```
SELECT t_type_avion
FROM Vols
WHERE n°vol = 523 ;
```

11

Exemple 2: Gestion de commandes

- BD gestion de commandes Contenu : les clients, les produits, les prix, les fournisseurs, la facturation, la livraison, ...
- Applications : toutes les applications de gestion de l'entreprise, Manipulation : par les services de l'entreprise mais aussi de l'extérieur (commandes par le site web, ...)
- Variabilité : forte, des ajouts et modifications fréquentes ;
- Consultation : par les services de l'entreprise mais aussi de l'extérieur (suivi de commande, ...)

12

Exemple 3

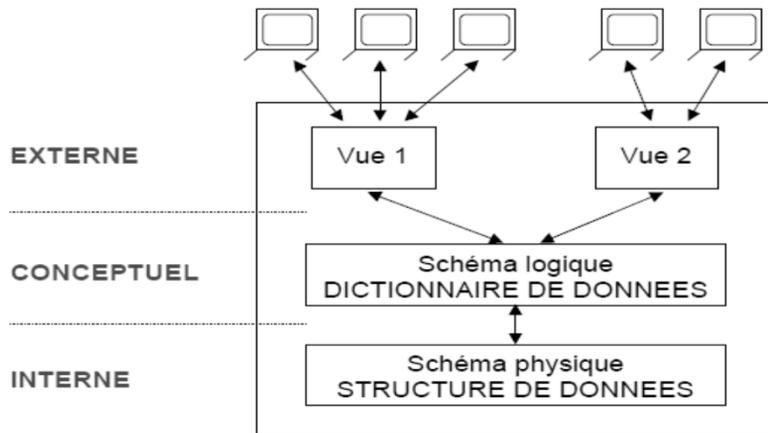
- BD et Web Un site dynamique est construit sur une base de données
- l'utilisateur sollicite une page, la demande est envoyée au site,
- le serveur exécute un programme, les informations utiles sont extraites de la base de données et renvoyées au serveur, une feuille de style est appliquée, un document html est construit et envoyé sur le réseau, le document est affiché dans le navigateur de l'utilisateur.
- l'intérêt est de distinguer les données des traitements et de l'affichage : pour modifier le contenu du site, il suffit de mettre à jour les informations dans la base de données ; pour modifier la présentation, il suffit de modifier le style du document Web.

Systèmes de Gestion de Base de Données

Les fonctions d'un SGBD:

A partir des exemples des systèmes précédents,
Quelles fonctions doit-on exiger pour un SGBD ?

SGBD

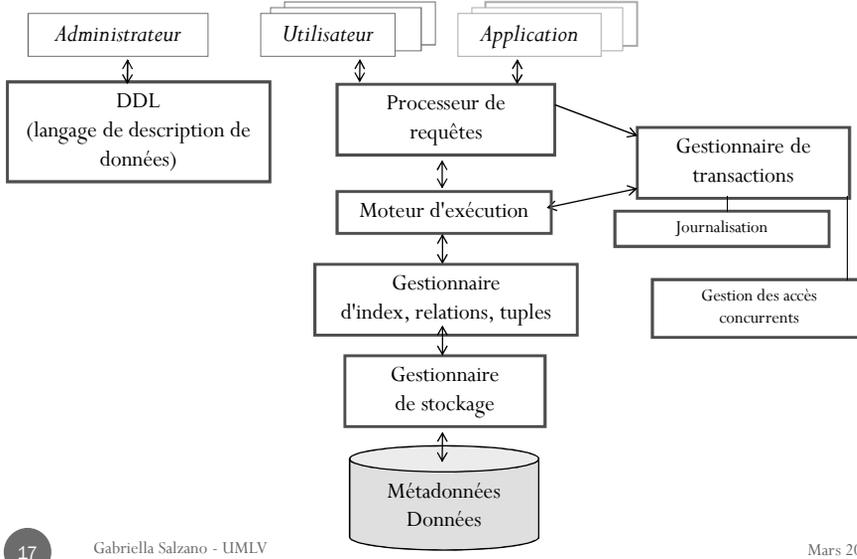


SGBD
Niveaux de représentation des données

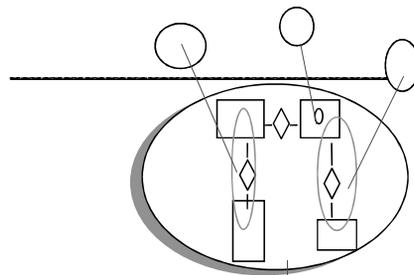
Les SGBDs: Fonctions

- Un SGBD assure la gestion du stockage sur des supports physiques éventuellement repartis et distants,
- Assurer la rapidité des accès, l'indépendance des données et des applications,
- **Langages** de définition de données, d'interrogation et de manipulation
- Gestion de l'**intégrité** des données
- Assurer le contrôle de la concurrence car les données sont partagées et réparties,
- Gestion des **autorisations**, des **transactions**
- Assurer la protection des données : reprise sur panne, vérifier les droits d'usage et d'accès

Les principales composantes d'un SGBD



Les niveaux des BD



Vues externes
chaque vue représente la partie de la bd relative à chaque utilisateur / application

Schéma conceptuel
vue globale de la base : les données stockées et les relations parmi les données

$R_1(A \text{ number}(10), B \text{ string}, \dots, \text{primary key } (A), \dots)$
.....
 $R_n(X \text{ number}(15), Y \text{ string}, \dots, \text{primary key } (X), \dots)$

Schéma logique
Représente un modèle de données conforme au modèle ciblé de SGBD (ex: **SGBDRelationnel**)

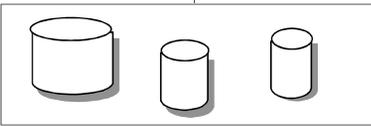
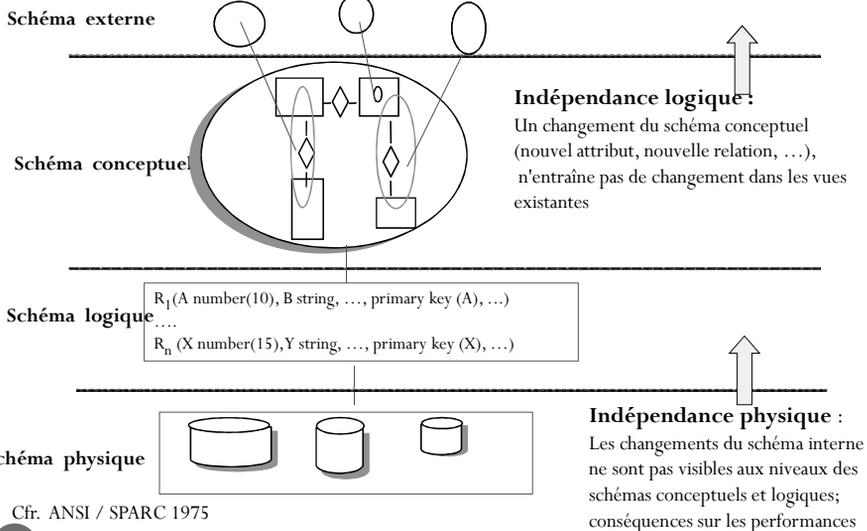


Schéma physique
décrit comment les données sont stockées dans la bd + les index

Niveaux d'abstractions

- **Schéma externe (*External Schema*)**
- Description d'une partie de la base de données extraite ou calculée à partir de la base physique, **correspondant à la vision d'un programme ou d'un utilisateur**, donc à un arrangement particulier de certaines données.
- **Schéma conceptuel (*Conceptual Schema*)**
- Description des données d'une entreprise ou d'une partie d'une entreprise en termes de types d'objets et de liens logiques indépendants de toute représentation en machine, correspondant à une vue canonique globale de la portion d'entreprise modélisée.
- **Schéma interne (*Internal Schema*)**
- Description des données d'une base en termes **de représentation physique en machine**, correspondant à une spécification des **structures de mémorisation et des méthodes de stockage et d'accès** utilisées pour ranger et retrouver les données sur disques.

Caractéristiques de l'Architecture



Le principal objectif d'un SGBD

- Assurer l'indépendance des programmes aux données, c'est-à-dire la possibilité de modifier les schémas conceptuel et interne des données sans modifier les programmes d'applications, et donc les schémas externes vus par ces programmes.
- **L'indépendance logique** qui permet de modifier les schémas conceptuels (par exemple, ajouter un type d'objet) sans changer les programmes d'applications.

les objectifs d'un SGBD :

- Un SGBD se doit donc **de faciliter l'administration** (c'est-à-dire la création et la modification de la description) **des données.**
 - **Indépendance physique des programmes aux données**
 - **Indépendance logique des programmes aux données**
 - **Manipulation des données par des langages non procéduraux**
- **Administration facilitée des données.**

Indépendance physique

- Un des objectifs essentiels des SGBD est donc de permettre de réaliser l'indépendance des structures de stockage aux structures de données du monde réel [Stonebraker74],
- Modifier le schéma interne sans avoir à modifier le schéma conceptuel, en tenant compte seulement des critères de performance et de flexibilité d'accès.

L'indépendance logique

- L'indépendance logique est donc la possibilité de modifier un schéma externe sans modifier le schéma conceptuel.
- Elle assure aussi l'indépendance entre les différents utilisateurs, chacun percevant une partie de la base via son schéma externe, selon une structuration voire un modèle particulier.
- Les avantages de l'indépendance logique [Date71] sont les suivants :
- Permettre à chaque groupe de travail de voir les données comme il le souhaite ;
- Permettre l'évolution de la vue d'un groupe de travail (d'un schéma externe) sans remettre en cause, au moins dans une certaine mesure, le schéma conceptuel de l'entreprise ;
- Permettre l'évolution d'un schéma externe sans remettre en cause les autres schémas externes.

Manipulation des données par des langages non procéduraux

- Les utilisateurs doivent pouvoir manipuler simplement les données, c'est-à-dire les interroger et les mettre à jour sans préciser les algorithmes d'accès.
- Plus généralement, si les objectifs d'indépendance sont atteints, les utilisateurs voient les données indépendamment de leur implantation en machine

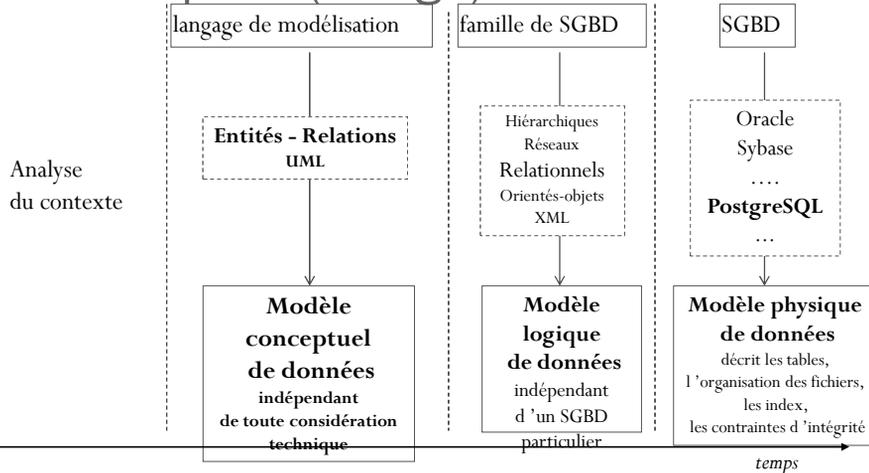
Administration facilitée des données

- Un SGBD doit fournir des outils pour décrire les données, à la fois leurs structures de stockage et leurs présentations externes.
- Les fonctions qui permettent de définir les données et de changer leur définition sont appelées outils d'administration des données.
- **Un dictionnaire de données** dynamique pourra aider les concepteurs de bases de données.

Objectifs additionnels des SGBD,

- Efficacité des accès aux données
- Partage des données
- Cohérence des données
- Redondance contrôlée des données
- Sécurité des données.

Conception (design) de la BD



Typologie des Modeles

- Le modèle Entité-association
- Le modèle Relationnel
- Le modele Réseaux

Le modèle Entité-Association

- Modèle le plus connu de la classe des modèles sémantiques.
- Modélisation de la sémantique = préoccupation de bon nombre de recherches fin 70-début 80.
- Motivation : les systèmes à base de données n'ont qu'une compréhension limitée de la signification des données.
- Si les systèmes en comprenaient un peu plus, ils pourraient répondre de manière plus intelligente aux interactions de l'utilisateur.

Présentation globale de l'approche

- Modélisation sémantique caractérisée par les 4 étapes suivantes :
 - on identifie des concepts sémantiques utiles pour décrire le système d'informations
 - on détermine l'ensemble d'objets symboliques qui sont utilisés pour représenter ces concepts sémantiques
 - on définit un ensemble de règles d'intégrité formelles pour donner une signification à ces objets formels.
 - on développe un ensemble d'opérateurs formels pour manipuler ces objets formels.

Présentation globale de l'approche

Concept	Définition formelle	Exemples
ENTITE	un objet que l'on peut distinguer	Fournisseur, pièce, cargaison, Personne Employé, département
PROPRIETE	une information qui décrit une entité	N° fournisseur, qté cargaison, département d'employé, poids de personne
ASSOCIATION	une entité qui permet de connecter 2 entités ou plus	Cargaison(fournisseur-pièce) Affectation(employé- département)

Un aperçu du modèle E-A : les objets sémantiques

- Modèle introduit par Chen en 1976, raffiné par Chen et d'autres.
- Entité [définition de Chen] :
 - chose qui peut être identifiée distinctement
- Propriété (ou Attribut) :
 - les entités (et les associations) sont décrites par des propriétés caractérisées par un nom et un type
- Association [définition de Chen] :
 - Lien entre entités
 - elle peut être binaire, ternaire, n-aire

Les diagrammes E-A

- Les entités
 - Représentées par un rectangle contenant le nom du type de l'entité
- Les propriétés (ou attributs)
 - le nom de la propriété écrit dans l'entité ou l'association correspondante
 - Les propriétés clés sont soulignées

Les diagrammes E-A

- Les associations
 - Représentées par une ellipse contenant le nom de l'association
 - Les entités participantes de chaque association sont rattachées à l'association au moyen de lignes continues.
 - Chacune de ces lignes est étiquetée par la cardinalité de l'association.

Entité

- ENTITE :
 - Concept :
 - pourvu d'une existence propre
 - conforme aux besoins de gestion de l'entreprise
 - Il peut représenter une notion concrète : CLIENT
ou une notion abstraite : PORTEFEUILLE D' ACTIONS
 - Synonymes : INDIVIDU, OBJET

Association

- ASSOCIATION :
 - Lien sémantique entre deux ou plusieurs entités.
 - Le lien n'est pas orienté : les commandes comportent des produits veut dire également que les produits peuvent être commandés.
 - Souvent nommé par un verbe ou un substantif
 - Synonyme : RELATION

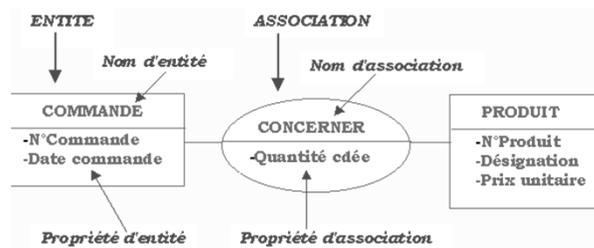
Entités vs Associations

- Il est difficile de faire une distinction entre les 2 concepts.
- Le même objet peut être vu comme entité par certains utilisateurs et comme association par d'autres.
- Exemple : le mariage
 - association entre deux personnes
 - une entité à part entière (si l'on veut connaître, par exemple, le nombre de mariages célébrés dans cette église depuis le début de l'année)

Propriété

- PROPRIETE :
 - Donnée élémentaire permettant de décrire une entité ou une association.
 - Cette donnée peut se mesurer par une valeur.
 - Synonyme : ATTRIBUT
- REGLES DE BASE :
 - Une propriété ne peut pas figurer sur deux objets différents.
 - Une entité possède au moins une propriété (son identifiant : par exemple le N° de commande).
 - Une association peut ne pas avoir de propriété

Un diagramme



OCCURRENCE

- OCCURRENCE :
 - Réalisation particulière d'une entité, propriété ou association.
 - Synonyme : INSTANCE
- Le mot "commande" est donc ambigu : ou bien le concept ou bien l'instance. Si l'on souhaite l'éviter, on peut employer pour les concepts les mots de : ENTITE-TYPE et ASSOCIATION-TYPE.
- En pratique, et en l'absence de précision, un mot sera relatif à un concept et lorsque l'on voudra parler d'un individu, on dira occurrence de ...

Les cardinalités

- La cardinalité est une notion OBLIGATOIRE du modèle qui permet de résoudre la question de l'anomalie d'une commande qui aurait pris la liberté de ne point comporter de produits.
- C'est donc l'expression d'une CONTRAINTE (une "loi") perçue sur le monde, et que l'on écrit dans le modèle. Par exemple, "il n'est pas possible qu'une commande ne concerne aucun produit".
- Comme il s'agit d'exprimer des lois, on ne peut pour ce faire qu'utiliser une autre loi :
 - Pour une occurrence de cette entité, combien y a-t-il d'occurrences de l'association auxquelles cette occurrence d'entité participe, au plus et au moins ?

Cardinalités

- Association 1,1
 - Un client donné ne commande qu'un seul produit.
 - Un produit donné n'est commandé que par un seul client.



Les identifiants

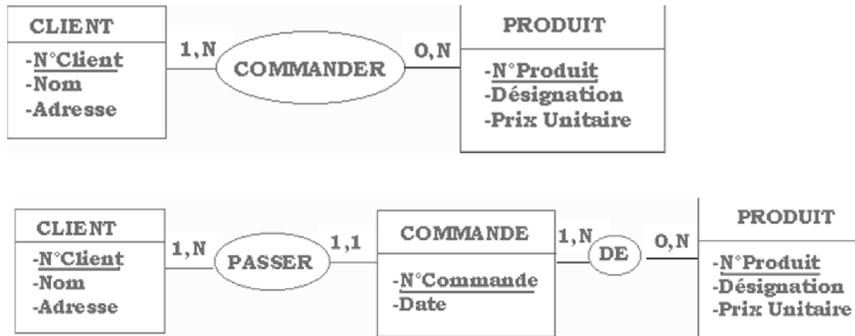
- IDENTIFIANT D'ENTITE :
 - Propriété PARTICULIERE de l'entité telle que pour chacune des valeurs de cette propriété, il existe une occurrence UNIQUE de l'entité.
- Remarque :

Si l'on ne sait pas trouver d'identifiant à une entité, c'est qu'elle n'a peut être pas d'existence propre. Il pourrait donc s'agir d'une association.
- Présentation :

L'identifiant est inscrit en tête de la liste des propriétés et souligné.

Dans les modèles très denses il peut suffire à résumer les autres propriétés, pour faciliter la lecture.

Les identifiants



Modèle relationnel

Modèle relationnel

Les concepts descriptifs

- Notion de domaine
- Produit cartésien
- Relation
- Attribut
- Clé
- Schéma de relation et de BD
- Clé étrangère
- Métabase

47

Modèle relationnel

Notion de domaine

- Définition
 - Ensemble de valeurs
- Exemples
 - Entier, réel, chaîne de caractères, booléen
 - Salaire = 1000...100000 (€)
 - Couleur = { 'rosé', 'blanc', 'rouge' }

48

Produit cartésien

- Définition
 - Le produit cartésien de D_1, \dots, D_n est l'ensemble des n-uplets (tuples) $\langle V_1, \dots, V_n \rangle$ tel que $V_i \in D_i$
- Notation
 - $D_1 \times \dots \times D_n$
- Exemple :
 - $D_1 = \{ 'BD', 'IO' \}$ (Code UV)
 - $D_2 = \{ 'Carpentier', 'Lalevée', 'Millot' \}$ (Prof)

$D_1 \times D_2$	D_1	D_2
	BD	Carpentier
	BD	Lalevée
	BD	Millot
	IO	Carpentier
	IO	Lalevée
	IO	Millot

Relation

- Définition
 - Sous-ensemble du produit cartésien d'une liste de domaines
 - Caractérisée par un nom
- Exemple
 - $D_1 = \text{Code UV}$
 - $D_2 = \text{Prof}$

UV	D_1	D_2
	BD	Carpentier
	IO	Lalevée

Relation (2)

- Plus simplement, une relation est un tableau à deux dimensions
- Une ligne est un n-uplet (tuple)
- On associe un nom à chaque colonne afin de la repérer indépendamment de l'ordre = attribut
 - Prend ses valeurs dans un domaine
 - Exemple : code

UV	code	coord
	BD	Carpentier
	IO	Lalevée

Exemples de relations

Elève	Num	Nom	Adresse	Age
	1	Bélaïd	Maisel	20
	2	Millot	CROUS	20
	3	Meunier	Maisel	21

Inscrit	NumElève	CodeUV	Note
	2	BD	10
	1	BD	20
	2	IO	17
	3	IO	18

Clé

- Définition
 - Une clé est un **groupe d'attributs minimum** qui détermine un n-uplet unique dans une relation (à tout instant)
- Exemple
 - Clé de Elève ?
 - Clé de UV ?
 - Clé de Inscrit ?
- Contrainte d'intégrité
 - Toute relation doit posséder une clé renseignée (sans valeur inconnue)

Schéma de relation

- Définition
 - Le schéma d'une relation décrit :
 - Son nom
 - La liste des attributs qu'elle comporte et des domaines associés
 - La liste des attributs composant la clé (la clé est soulignée)
- Exemple
 - Elève(num : entier, nom : chaîne, adresse : chaîne, age : entier de 18 à 35)
- Intention vs. Extension
 - Schéma de relation : intention de la relation
 - Table : extension
 - Schéma d'une BD relationnelle : ensemble des schémas des relations

Clé étrangère

- Définition
 - Une clé étrangère est un groupe d'attributs qui apparaît comme clé dans une autre relation

$$R1(\underline{A1}, A2, \dots, \mathbf{Ap}, \mathbf{Ap+1}, \dots, An)$$

$$R2(B1, B2, \dots, Bn)$$

- Rôle
 - Les clés étrangères définissent des contraintes d'intégrités référentielles entre relations

Clé étrangère (2)

- Mises à jour et clés étrangères
 - Insertion: la valeur des attributs doit exister dans la relation référencée.
 - Insertion de (4, 'BD', 15) dans Inscrit ?
 - Suppression dans la relation référencée; les n-uplets référençant doivent disparaître.
 - Suppression de l'élève 2 dans Elève ?
- Les **clés étrangères** sont la **traduction** des **associations** du modèle E/A

Clé étrangère

- Exemples

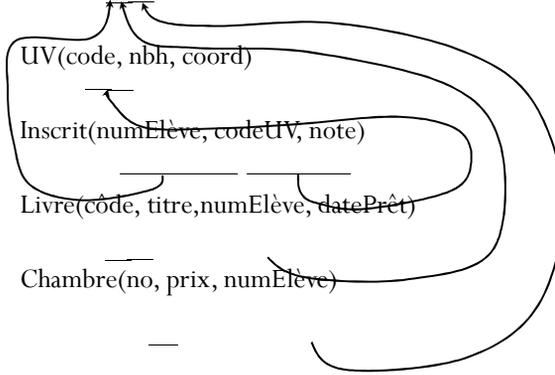
Élève(num, nom, adresse, age)

UV(code, nbh, coord)

Inscrit(numÉlève, codeUV, note)

Livre(code, titre, numÉlève, datePrêt)

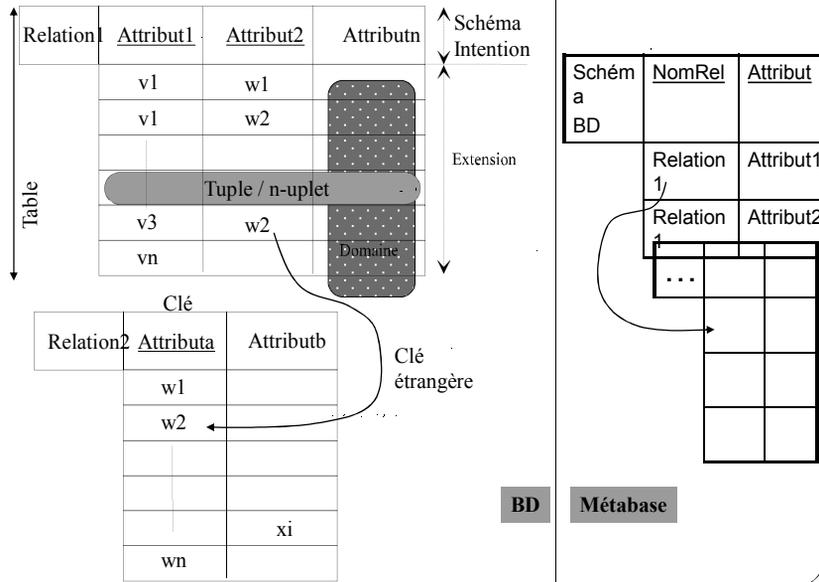
Chambre(no, prix, numÉlève)



Métabase

- Définition
 - base de données contenant l'ensemble des schémas et des règles de correspondances associées à une base de données
- Principe
 - Une base décrivant les autres bases, c'est-à-dire:
 - les relations
 - les attributs
 - les domaines
 - les clés
 - Notion de dictionnaire de données
 - Base particulière, système, gérée par l'administrateur de BD

Résumé des notions



59

Langages associés au modèle relationnel

- Langages de Définition de Données (LDD) :
 - Définition /mise à jour des schémas des relations
- Langages de manipulation de données (LMD) :
 - Interrogation : recherche de données
 - Mises à jour : insertion, suppression, modification
- 2 classes de langages :
 - Algébriques → SQL
 - Prédicatifs → QBE

60

Normalisation

RELATION NORMALE

- Une relation est dite normale si aucun des domaines qui la composent n'est lui-même une relation. En d'autres termes, une relation normalisée est un ensemble de n-uplets de la forme :

$$t_j = (d_{j,1}, d_{j,2}, \dots, d_{j,n}), d_{j,k} \in D_k, k \in (1, n)$$

Dépendance fonctionnelle (suite)

- on peut toujours décomposer une relation suivant une dépendance fonctionnelle
- on ne peut décomposer une relation s'il n'y a pas de dépendance fonctionnelle
- la décomposition suivant une dépendance fonctionnelle ne perd pas d'information

Dépendance fonctionnelle élémentaire

- Une dépendance fonctionnelle (DF) donnée1 \rightarrow donnée2 est élémentaire, s'il n'existe pas une donnée3, incluse dans donnée1, qui assure elle-même une DF donnée3 \rightarrow donnée2.
 - **Exemples :**
 - Ref_Article \rightarrow Nom_Article VRAI
 - Num_Facture, Ref_Article \rightarrow Qte_Article VRAI
 - Num_Facture, Ref_Article \rightarrow Nom_Article FAUX

Dépendance fonctionnelle directe

- Une dépendance fonctionnelle (DF) donnée1 -> donnée2 est directe, s'il n'existe pas une donnée3 (ou une collection de rubriques) qui engendrerait une DF transitive donnée1 -> donnée3 -> donnée2.
- **Exemples :**
 - Num_Facture -> Num_Représentant
 - Num_Représentant -> Nom_Représentant
 - Num_Facture -> Nom_Représentant DF non directe car on peut écrire :
 - Num_Facture -> Num_Représentant -> Nom_Représentant

Anomalies de stockage

- Les anomalies de stockage concernent les opérations d'insertion, de suppression et de modification de la base de données.

nom usine	Nom équipement	directeur usine	fournisseur équipement	adresse fournisseur
éthylène	refroidiss. Final	Jim Smith	ABC échangeurs	1247 Locust
éthylène	radiateur alimentation	Jim Smith	ABC échangeurs	1247 Locust
éthylène	radiateur alimentation	Bill Gunn	ABC échangeurs	1247 Locust

Si nous avons qu'un seul fournisseur pour un équipement, nous ne pouvons pas insérer un autre fournisseur pour le même équipement. Il faut obligatoirement insérer le même nom du directeur d'usine dans notre insertion. Si une erreur se produit à l'insertion du directeur d'usine, on aura pour la même usine deux directeurs différents.

Anomalie de suppression : Exemple

nom usine	Nom équipement	directeur usine	fournisseur équipement	adresse fournisseur
éthylène	refroidiss. Final	Jim Smith	ABC échangeurs	1247 Locust
éthylène	radiateur alimentation	Jim Smith	ABC échangeurs	1247 Locust
styrène	pompe alimentation	Bill Gunn	XYZ pompes	432 Broadway
styrène	radiateur alimentation	Bill Gunn	ABC échangeurs	1247 Locust

- Supposons qu'on supprime les deux derniers enregistrements de la table ci-dessus. Ça implique que nous n'avons plus d'information sur l'usine styrène. Alors que ce que nous voulions faire était seulement d'enlever deux équipements de la table.

Anomalie de modification : Exemple

nom usine	Nom équipement	directeur usine	fournisseur équipement	adresse fournisseur
éthylène	refroidiss. Final	Jim Smith	ABC échangeurs	1247 Locust
éthylène	radiateur alimentation	Jim Smith	ABC échangeurs	1247 Locust
styrène	pompe alimentation	Bill Gunn	XYZ pompes	432 Broadway
styrène	radiateur alimentation	Bill Gunn	ABC échangeurs	1247 Locust

- Supposons qu'on change de directeur d'usine à l'usine éthylène. On doit alors changer le directeur d'usine dans tous les enregistrements de la table. Si on en oublie un, on se retrouve avec une anomalie de stockage.

Normalisation des relations

- À partir des règles sémantiques qui traduisent les contraintes de l'entreprise modélisée, le concepteur doit définir les dépendances à introduire dans la définition du schéma relationnel.
 - Ces dépendances seront utilisées pour générer les différentes relations irréductibles du schéma.

Les différentes formes normales d'une relation

- La théorie sur la normalisation repose sur l'observation que certaines relations ont de meilleures propriétés dans un environnement de mise à jour, que d'autres relations équivalentes contenant les mêmes informations.
- Cette théorie fournit un cadre rigoureux pour la définition du schéma relationnel.

Les différentes formes normales d'une relation (suite)

- La théorie est basée sur une série de formes normales :
 - première forme normale (1FN)
 - deuxième forme normale (2FN)
 - troisième forme normale (3FN)
 - quatrième forme normale (4FN)
 - cinquième forme normale (5FN)

Première forme normale

- Une relation est dite normalisée ou en première forme normale si :
 - aucun attribut qui la compose n'est lui-même une relation, c'est-à-dire si tout attribut est atomique (non décomposable).
 - Cette forme n'utilise que les structures de base d'une relation, elle ne résout pas le problème de la redondance.

Première forme normale : Exemple

- Ce schéma viole la première forme normale :

No_Cours	Titre	No-Pré-requis	Type
GMC230	Energétique	GIN400 et GIN200	Antérieure
GEI210	Electronique I	GEI205	Antérieure
GEI215	Electronique II	GEI210	Préalable

- Ce schéma satisfait la première forme normale :

No_Cours	Titre	No_Pré-requis	Type
GMC230	Energétique	GIN400	Antérieure
GMC230	Energétique	GMC200	Antérieure
GEI210	Electronique I	GEI205	Antérieure
GEI215	Electronique II	GEI210	Préalable

Deuxième forme normale

- Une relation est dite en deuxième forme normale si et seulement si :
 - Elle est en première forme normale ;
 - Chaque attribut est totalement dépendant de la clé primaire.
 - Avec cette forme, les problèmes de redondance ne sont pas entièrement résolus.

Deuxième forme normale : Exemple

- Ce schéma viole la deuxième forme normale :

No_Cours	Titre	No_Pré-requis	No_Equivalent
GMC230	Energétique	GIN400	GIN666
GMC230	Energétique	GMC200	GMC300
GEI210	Electronique I	GEI205	
GEI215	Electronique II	GEI210	GEI207

No_Equivalent dépend de la clé primaire No_Cours et No_Pré-requis

mais

Titre ne dépend que d'une partie de la clé primaire

Deuxième forme normale : Exemple

- Ce schéma satisfait la deuxième forme normale :

No_Cours	Titre
GMC230	Energétique
GMC230	Energétique
GEI210	Electronique I
GEI215	Electronique II

No_Cours	No_Pré-requis	No_Equivalent
GMC230	GIN400	GIN666
GMC230	GMC200	GMC300
GEI210	GEI205	
GEI215	GEI210	GEI207

Troisième forme normale

- Une relation est en troisième forme normale si et seulement si :
 - elle est en 2FN;
 - et chaque attribut non-clé primaire dépend directement de la clé primaire.
 - La 3FN est adéquate pour la majorité des designs de BD mais elle n'élimine pas toutes les redondances et incohérences. Pour cela, Codd a pensé à BCNF qui est une forme plus stricte de 3NF.

Troisième forme normale : Exemple

- Ce schéma ne satisfait pas la 3FN :

No_Cours	Titre
GMC230	Energétique
GMC230	Energétique
GEI210	Electronique I
GEI215	Electronique II

No_Cours	No_Pré-requis	No_Equivalent	Titre_Equivalent
GMC230	GIN400	GIN666	Matériaux
GMC230	GMC200	GMC300	Mécanique
GEI210	GEI205		
GEI215	GEI210	GEI207	Electronique virtuel

- Il existe une dépendance entre Titre_Equivalent et No_Equivalent

Troisième forme normale : Exemple (suite)

- Ce schéma satisfait la 3FN :

No_Cours	Titre
GMC230	Energétique
GMC230	Energétique
GEI210	Electronique I
GEI215	Electronique II

No_Equivalent	Titre_Equivalent
GIN666	Matériaux
GMC300	Mécanique
GEI207	Electronique virtuelle

No_Cours	No_Pré-requis	No_Equivalent
GMC230	GIN400	GIN666
GMC230	GMC200	GMC300
GEI210	GEI205	
GEI215	GEI210	GEI207

Dépendance à valeur multiple (DVM)

- Les dépendances à valeur multiple sont une conséquence de la 1NF puisque les attributs doivent être atomiques.
 - Pour ce faire, il faut dédoubler certaines données de la table
 - puisque des valeurs redondantes sont alors stockées dans la BD, une relation comportant des DVM peut occasionner des anomalies de mise-à-jour.
 - Ce phénomène n'est toutefois pas résolu par les trois premières formes normales et c'est pourquoi la 4NF existe.

Quatrième forme normale

- Permet autant que possible de minimiser l'occurrence d'attributs indépendants à valeur multiple.
- Une relation est de la 4FN si :
 - elle satisfait la 3FN
 - les données composant chaque attribut ne comportent aucune répétition inutile -> dans une même colonne, il faut minimiser les répétitions.
- Une base qui est 4FN est des plus optimales quoiqu'il soit possible de généraliser cette dernière afin d'obtenir la 5FN.

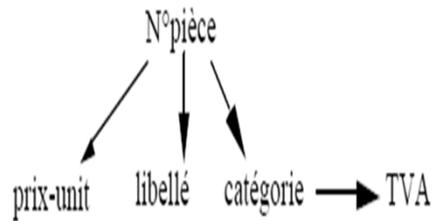
Exercice 1

Pour chaque relation ci-dessous:

- - identifier les redondances éventuelles dans sa population,
- - établir le (un) graphe minimum de ses dépendances,
- - définir son (ses) identifiant(s), - définir sa forme normale et la justifier,
- - si nécessaire, proposer une décomposition optimale.
- **a. Pièce: description des pièces employées dans un atelier de montage.**
- Pièce (N°pièce, prix-unit, TVA, libellé, catégorie) avec les dépendances fonctionnelles suivantes:
- N°pièce → prix-unit, TVA, libellé, catégorie ,
- catégorie → TVA

Solution Exercice 1

- **a. Pièce**
- a) Il y a redondance des valeurs de TVA, par rapport aux catégories.
- b) Le graphe minimum des dépendances fonctionnelles est:



Solution exo 1

- c) **L'identifiant est N°pièce.**
- d) La relation est **en seconde forme normale**, mais pas en troisième forme normale, parce qu'il y a **une dépendance transitive**:
 $N^{\circ}pièce \rightarrow TVA$ déduite des dépendances $N^{\circ}pièce \rightarrow catégorie$ et $catégorie \rightarrow TVA$.
- e) **La décomposition est la suivante:**
- Pièce ($N^{\circ}pièce, prix-unit, libellé, catégorie$)
- Catégorie ($catégorie, TVA$)
- Ces deux relations sont en forme normale de Boyce Codd (toutes les sources de dépendance fonctionnelle sont des identifiants entiers) et en quatrième forme normale (il n'y a pas de dépendance multivaluée).

Suite exo1

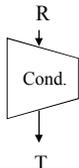
- **b. Prime: liste des primes attribuées au personnel technique en fonction des machines sur** lesquelles il travaille
- Prime (N°machine, atelier, N°techn, montant-prime, nom-techn)
- avec les dépendances fonctionnelles suivantes: N°machine → atelier et N°techn → nom-techn
- (N°machine, N°techn) → montant-prime
- **c. Employé1: description d'un employé travaillant sur un projet d'un laboratoire.**
- Employé (N°Emp, N°Lab, N°Proj, NomEmp, NomProj, adresse)
- avec les dépendances fonctionnelles suivantes: (N°Emp, N°Lab) → N°Proj, NomProj, NomEmp
- N°Emp → NomEmp et N°Emp → adresse et N°Proj → NomProj
- **d. Adresse**
- Adresse (rue, ville, NPA) avec les dépendances fonctionnelles suivantes: NPA → ville (rue, ville) → NPA

Les opérateurs de manipulation

- Tout résultat d'une opération est une relation; peut donc être réutilisée en entrée d'un nouvel opérateur.
- Les opérateurs peuvent être classifiés en :
 - opérateurs ensemblistes / opérateurs relationnels
 - opérateurs de base / opérateurs dérivés
 - opérateurs unaires / opérateurs binaires
 - Unaires : sélection (restriction), projection,
 - Binaires : union, intersection, différence, produit cartésien, jointure, division

Restriction

- But
 - Permet de "sélectionner" des tuples
 - La restriction réduit la taille de la relation verticalement
- Contraintes
 - Unaire
 - Spécifier une condition
- Notation
 - Notation textuelle: $T \leftarrow \sigma_{cond}(R)$
 - Notation graphique:



- Inscrits en BD : $\sigma_{codeUV='BD'}(Inscrit)$

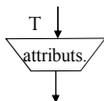
Resu	NumElève	CodeUV	Note
2		BD	10
1		BD	20

- Majors (note > 15) de BD : $\sigma_{codeUV='BD' \text{ et } note > 15}(Inscrit)$

Resu	NumElève	CodeUV	Note
1		BD	20

Projection

- But
 - Permet de "sélectionner" des attributs
 - La projection réduit la taille de la relation horizontalement
- Contraintes
 - Unaire
 - Spécifier une liste d'attributs
- Notation
 - Notation textuelle: $T \leftarrow \Pi_{attributs}(R)$
 - Notation graphique:



Adresses des élèves : $\Pi_{Adresse}(Elève)$

$\Pi_{code,nbh}(UV)$

Resu	Adresse
	Maisel
	CROUS

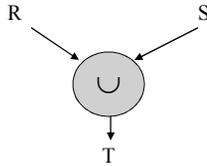
Pas de doublon

Resu	Code	nbh
	IO	45
	BD	15

Union



- But
 - Permet de fusionner 2 relations
- Contraintes
 - Binaire
 - **Même schéma**
- Notation
 - Notation textuelle: $T \leftarrow R \cup S$
 - Notation graphique:



- Nom des profs, des élèves

Prof	Nom	Elève2	Nom
	Lalevée		Bélaïd
	Carpentier		Millot
	Millot		Meunier

- Nom des personnes à l'INT : $\text{Prof} \cup \text{Elève2}$

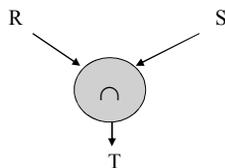
Resu	Nom
	Lalevée
	Carpentier
	Millot
	Bélaïd
	Meunier

Pas de doublon

Intersection



- But
 - Permet d'obtenir l'ensemble des tuples appartenant à deux relations
- Contraintes
 - Binaire
 - **Même schéma**
- Notation
 - Notation textuelle: $T \leftarrow R \cap S$
 - Notation graphique:



- Nom des profs, des élèves

Prof	Nom	Elève2	Nom
	Lalevée		Bélaïd
	Carpentier		Millot
	Millot		Meunier

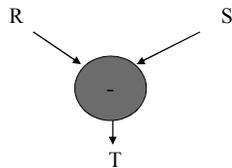
- Noms communs élèves-profs : $\text{Prof} \cap \text{Elève2}$

Resu	Nom
	Millot

Différence

But

- Obtenir l'ensemble des tuples d'une relation qui ne figurent pas dans une autre
- Contraintes
 - Binaire
 - **Même schéma**
 - Non commutatif
- Notation
 - Notation textuelle: $T \leftarrow R - S$
 - Notation graphique:



- Nom des profs, des élèves

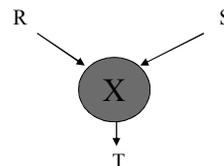
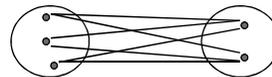
Prof	Nom	Elève2	Nom
	Lalevée		Bélaïd
	Carpentier		Millot
	Millot		Meunier

- Noms des élèves qui ne portent pas le nom d'un prof : Eleve2-Prof

Résu	Nom
	Bélaïd
	Meunier

Produit cartésien

- But
 - Ensemble de tous les tuples obtenus par concaténation de chaque tuple de R avec chaque tuple de S
- Contraintes
 - Binaire
 - Schéma du résultat:
 - $R(a_1, a_2, \dots, a_n), S(b_1, b_2, \dots, b_p)$
 - $T \leftarrow R \times S, T(a_1, a_2, \dots, a_n, b_1, b_2, \dots, b_p)$
 - $\text{Card}(R \times S) = \text{Card}(R) * \text{Card}(S)$
- Notation
 - Notation textuelle: $T \leftarrow R \times S$
 - Notation graphique:

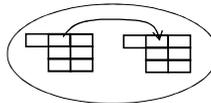


Produit cartésien (2)

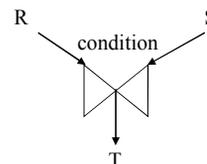
Elève	Num	Nom	Adresse	Age	UV	Code	Nbh	Coord
	1	Bélaïd	Maisel	20		IO	45	Lalevée
	2	Millot	CROUS	20		BD	15	Carpentier
	3	Meunier	Maisel	21				

Elève X UV	Num	Nom	Adresse	Age	Code	Nbh	Coord
	1	Bélaïd	Maisel	20	IO	45	Lalevée
	2	Millot	CROUS	20	IO	45	Lalevée
	3	Meunier	Maisel	21	IO	45	Lalevée
	1	Bélaïd	Maisel	20	BD	15	Carpentier
	2	Millot	CROUS	20	BD	15	Carpentier
	3	Meunier	Maisel	21	BD	15	Carpentier

Jointure



- But
 - Permet d'établir le lien sémantique entre les relations
- Contraintes
 - Binaire
 - Schéma du résultat:
 - $R(a_1, a_2, \dots, a_n), S(b_1, b_2, \dots, b_p)$
 - $T \leftarrow R \bowtie S(a_1, a_2, \dots, a_n, b_1, b_2, \dots, b_p)$
- Notation
 - Notation textuelle: $T \leftarrow R \bowtie S$
condition
 - Notation graphique:



1er exemple de jointure

Elève	<u>Num</u>	Nom	Adresse	Age
	1	Bélaïd	Maisel	20
	2	Millot	CROUS	20
	3	Meunier	Maisel	21

Elève.Num=Chambre.numElève

Chambre	<u>No</u>	Prix	<u>numElève</u>
	10	200	3
	21	150	2

Elève	<u>Num</u>	Nom	Adresse	Age	No	Prix	<u>numElève</u>
▷◁Chambre	2	Millot	CROUS	20	21	150	2
	3	Meunie	Maisel	21	10	200	3

- 1 tuple de Chambre → 1 tuple de résultat
- 1 tuple de Elève → 0 ou 1 tuple de résultat
 - On a perdu Bélaïd !

95

2ème exemple de jointure

Inscrit	<u>NumElève</u>	<u>CodeUV</u>	Note
	2	BD	10
	1	BD	20
	2	IO	17
	3	IO	18

Inscrit.NumElève=Elève.Num

Elève	<u>Num</u>	Nom	Adresse	Age
	1	Bélaïd	Maisel	20
	2	Millot	CROUS	20
	3	Meunier	Maisel	21

Inscrit	<u>Num</u>	Nom	Adresse	Age	<u>NumElève</u>	CodeUV	Note
▷◁Elève	1	Bélaïd	Maisel	20	1	BD	20
	2	Millot	CROUS	20	2	IO	17
	2	Millot	CROUS	20	2	BD	10
	3	Meunier	Maisel	21	3	IO	18

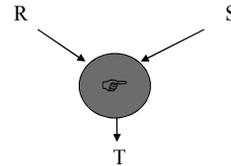
- 1 tuple de Inscrit → 1 tuple de résultat
- 1 tuple de Elève → 0 à n tuples de résultat
 - On a dupliqué Millot !

96

Division

But

- Répondre aux requêtes de type « tous les »
- Un tuple t est dans T si et seulement si pour tout tuple s de S , le tuple $\langle t, s \rangle$ est dans R
- Contraintes
 - Binaire
 - Schéma du résultat:
 - $R(a_1, a_2, \dots, a_n, b_1, b_2, \dots, b_p), S(b_1, b_2, \dots, b_p)$
 - $T \leftarrow R \div S, T(a_1, a_2, \dots, a_n)$
- Notation
 - Notation textuelle: $T = R \div S$
 - Notation graphique:
- Dérivation
 - Projection + Produit cartésien + Différence.
 - $R \div S \leftarrow T_1 - T_2$ avec:
 - $T_1 \leftarrow \Pi_{\text{schéma}(R) - \text{schéma}(S)}(R)$
 - $T_2 \leftarrow \Pi_{\text{schéma}(R) - \text{schéma}(S)}((\Pi_{\text{schéma}(R) - \text{schéma}(S)}(R) \times S) - R)$



Division (2)

- Exemple
 - Quels sont les élèves inscrits à toutes les UVs ?

Inscrit	NumElève	CodeUV	Note
	2	BD	10
	1	BD	20
	2	IO	17
	3	IO	18

Division (3)

- Exemple
 - Construire R : ensemble de toutes les informations dont on a besoin = attributs NumElève et CodeUV de Inscrit (R)
 - Construire S : ensemble correspondant à "tous les" (UV) = codeUV (S)
 - Résultat $\leftarrow R \div S$
 - Vérification :
 - Résultat $\times S \subseteq R$

$R \leftarrow \Pi_{\text{NumElève, CodeUV}}(\text{Inscrit})$

R	NumElève	CodeUV
	2	BD
	1	BD
	2	IO
	3	IO

$S \leftarrow \Pi_{\text{Code}}(\text{UV})$

S	CodeUV
	BD
	IO

• Résultat

Resu	NumElève
	2

Bilan : sémantique et notations des opérateurs

Opérateur	Sémantique	Notation textuelle	Notation graphique
Restriction	« Sélectionner » des tuples	$T \leftarrow \sigma_{\text{cond}}(R)$	
Projection	« Sélectionner » des attributs	$T \leftarrow \Pi_{\text{attributs}}(R)$	
Union	Fusionner les extensions de 2 relations	$T \leftarrow R \cup S$	
Intersection	Obtenir l'ensemble des tuples communs à deux relations	$T \leftarrow R \cap S$	
Différence	Tuples d'une relation qui ne figurent pas dans une autre	$T \leftarrow R - S$	
Produit cartésien	Concaténer chaque tuple de R avec chaque tuple de S	$T \leftarrow R \times S$	
Jointure	Etablir le lien sémantique entre les relations	$T \leftarrow R \bowtie_{\text{condition}} S$	
Division	Répondre aux requêtes de type « tous les »	$T \leftarrow R \div S$	

Bilan : contraintes des opérateurs

Opérateur	Unaire/ Binaire	Schémas	« Paramètres »
Restriction $T \leftarrow \sigma_{\text{cond}}(R)$	Unaire	Schéma(T) = Schéma(R)	Condition sur attributs de R
Projection $T \leftarrow \Pi_{\text{attributs}}(R)$	Unaire	Schéma(T) \subseteq Schéma(R)	Liste d'attributs de R
Union $T \leftarrow R \cup S$	Binaire	Schéma(R)=Schéma(S)=Schéma(T)	
Intersection $T \leftarrow R \cap S$	Binaire	Schéma(R)=Schéma(S)=Schéma(T)	
Différence $T \leftarrow R - S$	Binaire	Schéma(R)=Schéma(S)=Schéma(T)	
Produit cartésien $T \leftarrow R \times S$	Binaire	Schéma(T)=Schéma(S) \cup Schéma(R)	
Jointure $T \leftarrow R \bowtie_{\text{condition}} S$	Binaire	Schéma(T)=Schéma(S) \cup Schéma(R)	Condition de jointure sur attributs de R et S
Division $T \leftarrow R \oslash S$	Binaire	Schéma(R)=Schéma(S)+Schéma(T)	

101

Exemples de requêtes en algèbre relationnelle

Exemples

- Base de données exemple : les vins

Vins(num, cru, annee, degre)

Recoltes(nvin, nprod, quantite)

Producteurs(num, nom, prenom, region)

Buveurs(num, nom, prenom, ville)

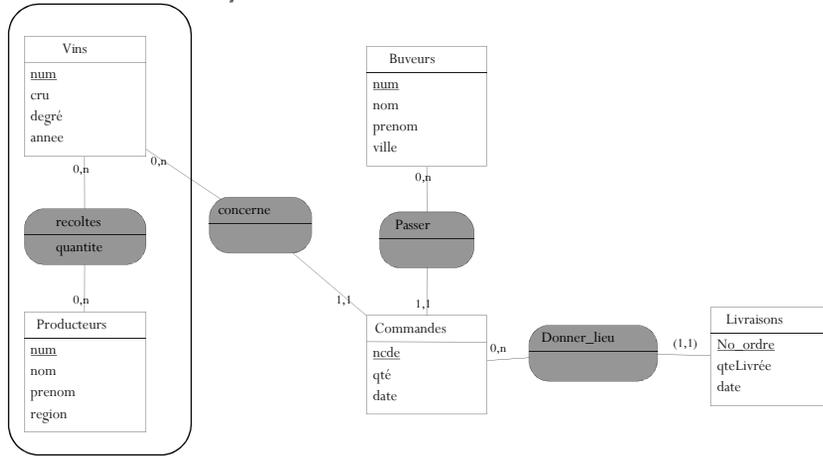
Commandes(ncde, date, nb, nvin, qte)

Livraisons(ncde, no_ordre, qteLivree)

V
R
P
B
C
L

102

Modèle E/A de la BD des Boissons



103

Composition des opérateurs

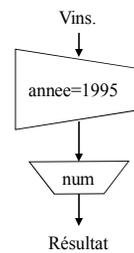
- Ecriture textuelle :

$$\text{Temp} \leftarrow \sigma_{\text{annee}=1995}(\text{Vins})$$

$$\text{Resultat} \leftarrow \Pi_{\text{num}}(\text{Temp})$$

$$\text{Resultat} \leftarrow \Pi_{\text{num}}(\sigma_{\text{annee}=1995}(\text{Vins}))$$

- Arbre algébrique :



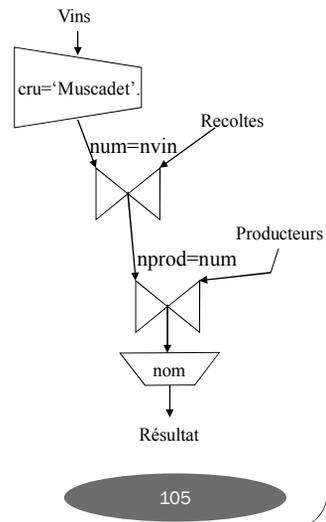
104

Noms des producteurs de Muscadet

$$\text{Mus} \leftarrow \sigma_{\text{cru}='Muscadet'}(V)$$

$$\text{RecMus} \leftarrow \text{Mus} \bowtie_{\text{Mus.num}=\text{R.nvin}} \text{R}$$

$$\text{ProdMus} \leftarrow \text{RecMus} \bowtie_{\text{Rec.nprod}=\text{P.num}} \text{P}$$

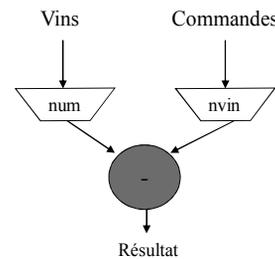
$$\text{Resultat} \leftarrow \Pi_{\text{nom}}(\text{ProdMus})$$


105

Numéros des vins ne faisant l'objet d'aucune commande

$$\text{NumVins} \leftarrow \Pi_{\text{num}}(V)$$

$$\text{VinsCdes} \leftarrow \Pi_{\text{nvin}}(C)$$

$$\text{Resultat} \leftarrow \text{NumVins} - \text{VinsCdes}$$


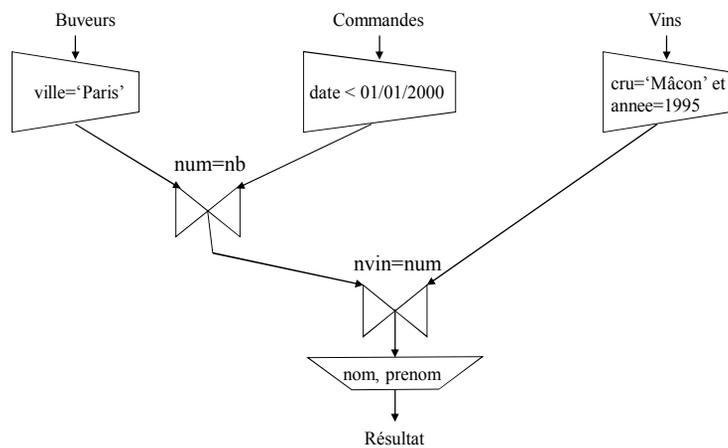
106

Optimisation

- Noms et prénoms des buveurs habitant Paris ayant commandé du Mâcon 1995 avant le 01 janvier 2000
 - Plusieurs réponses possibles
 - Optimiser = choisir la meilleure façon

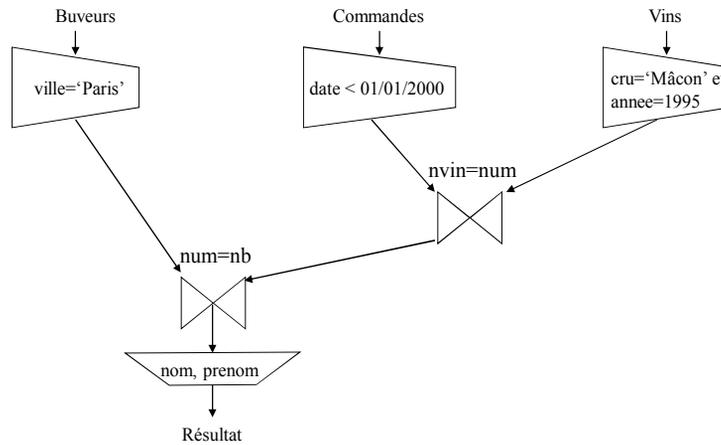
107

Réponse 1



108

Réponse 2



109

Optimisation par le SGBD

- Optimiser :
 - Le nombre d'E/S
 - Le temps UC
 - ...
- Facteurs déterminants :
 - Ordre d'exécution des opérations algébriques
 - Algorithme implantant les opérations algébriques
 - Placement des données sur le disque
 - Taille des relations intermédiaires

110

Modèle Réseau

Les points abordés dans ce chapitre sont :

- *Concept de base du modèle réseau*
- *Transformations E-R vers BDR (Base de données réseau)*
- *Le modèle CODASYL DBTG*
- *Langage de manipulation des données des BDR*

Le modèle réseau

- Le modèle réseau a été proposé par le groupe DBTG du comité CODASYL [Codasy171]. Des rapports plus récents [Codasy178, Codasy181] ont apporté des améliorations notables. Malheureusement, ces extensions ne sont que rarement intégrées dans les produits, pour la plupart construits au début des années 70.
- Le modèle réseau type CODASYL 1971 reste encore utilisé par plusieurs systèmes, le plus connu en France étant IDS.II de BULL.

Concepts de base du modèle réseau

- Les données du modèle réseau sont représentées par des
- **ensembles d'enregistrements associés par des relations.**
- *Modèle logique orienté enregistrement;*
- *Les enregistrements représentent les entités d'une structure entités-relations.*
- *Les enregistrements sont associés par des relations que l'on appelle liens.*

exemple

- **RECORD** NAME IS VINS;
- 02 NV TYPE IS SIGNED PACKED DECIMAL 5;
- 02 CRU TYPE IS CHARACTER 10;
- 02 MILLESIME OCCURS 5 TIMES;
- 03 ANNEE TYPE IS SIGNED UNPACKED DECIMAL 4;
- 03 DEGRE TYPE IS BINARY 15.

Exemple 2

- **RECORD NAME IS PRODUCTEURS;**
- **02 NP TYPE IS SIGNED PACKED DECIMAL 5;**
- **02 NOM TYPE IS CHARACTER 10;**
- **02 PRENOM TYPE IS CHARACTER 10;**
- **02 ADRESSE**
- **03 RUE TYPE IS CHARACTER 30;**
- **03 CODE TYPE IS SIGNED PACKED DECIMAL 5;**
- **03 VILLE TYPE IS CHARACTER 10;**
- *Figure IV.1 — Description des données de deux types d'articles*
- **2.3 La**

Merci de Votre Attention

- Ce cours est tiré de :
- Chapitre 2 Gardarin : **OBJECTIFS ET ARCHITECTURE DES SGBD**
- L'algèbre relationnel, IUT de Nice - Cours SGBD1 69