Bases de données

Souhila KACI

Partie 1

Déroulement

- 8 CM, 11 TD, 11 TP
- Contrôle Continu Intégral : 3 CC
- Note UE $= 40\%CC1 + 20\%CC2 + 40\%CC3$
- Rattrapage de CC (toute absence doit être justifiée – avec envoi de justificatif – au plus tard 7 jours après le CC). Absence non-justifiée (ou délai passé) $- > 0$
- Seconde chance (SC) - une nouvelle épreuve :
  Note UE $= 50\%SC + 50\%CC$ (CC est la note de l'UE en session 1)

Plan du cours

- Introduction aux bases de données
- Modèle Relationnel de données
- Comment obtenir le modèle relationnel de données ?
  Modélisation des données : Modèle Entité-Association
- Langages de manipulation des données

- Collection d'informations ou de données existantes sur une longue durée et décrivant les activités d'une ou plusieurs organisations.
- Ensemble de données modélisant les objets d'une partie du monde réel et servant de support à une application informatique.
- Un ensemble d'informations structurées et mémorisées sur un support permanent.
- Cet ensemble peut être interrogé et mis à jour par des utilisateurs.

Exemples de cadres d'utilisation de BD

- Gestion d'entreprises : stocks, personnel, clients
- Gestion bancaire : comptes, emprunts
- Systèmes de réservation : avions, trains, spectacles
- Bibliothèques : ouvrages, emprunteurs, prêts

Utilisation d'une base de données pour toute application nécessitant la structuration, le stockage, la manipulation et l'interrogation d'un ensemble conséquent d'informations.

Système de Gestion de Bases de Données (SGBD) (1)

Un SGBD est un logiciel intermédiaire entre la base de données et l'utilisateur (un humain ou un programme).
Un système de gestion de bases de données met à la disposition de l'utilisateur un outil pour

- la création et l'administration d'une BD,
- la sauvegarde (stockage) des données,
- la manipulation (insertion, modification, suppression, interrogation) des données d'une manière efficace.

Système de Gestion de Bases de Données (SGBD) (2)

Un SGBD permet de satisfaire les propriétés suivantes :

- Indépendance.
- Non redondance des données.
- Maintien de la cohérence des données (contraintes d'intégrité).
- Concurrence d'accès.

Modèle Relationnel de Données

- **Ensemble de valeurs caractérisé par un nom.**
- Ce sont les ensembles dans lesquels les données prennent leur valeur.
- Sa définition peut être :
  - en extension, en donnant la liste des valeurs composantes,
  - en intention, en définissant une propriété caractéristique des valeurs du domaine.
- Il peut être fini (chaîne de moins de 20 caractères) ou infini (les entiers naturels).

Exemple

- les entiers naturels (intention)
- La monnaie : réel avec deux chiffres derrière la virgule (intention)
- `CouleurVin = {'Blanc','Rouge','Rosé'}` (extension)
- `Crus = {'Aix','Sancerre'}` (extension)

Le produit cartésien d'un ensemble de domaines $D_1, D_2, \ldots, D_n$ est l'ensemble des vecteurs $\langle v_1, v_2, \ldots, v_n \rangle$, où $v_i \in D_i$.

Exemple

Le produit cartésien de
`CouleurVin = {'Rouge','Blanc','Rosé'}` et de
`Crus = {'Aix','Sancerre'}`

'Rouge'	'Aix'
'Rouge'	'Sancerre'
'Blanc'	'Aix'
'Blanc'	'Sancerre'
'Rosé'	'Aix'
'Rosé'	'Sancerre'

`CouleurVin` $\times$ `Crus`

Relation

- Concept central du modèle.
- Sous-ensemble d'un produit cartésien.
- **Une relation est donc un ensemble de n-uplets (ou tuples) de la forme $\langle a_1, a_2, \ldots, a_n \rangle$, avec $a_i \in A_i$.**
- Représentation commode : table à deux dimensions.
- Chaque ligne correspond à un vecteur (un tuple).
- Chaque colonne correspond à un domaine.

Exemple

CouleurCrus	Couleur	Cru
	'Rouge'	'Aix'
	'Blanc'	'Sancerre'
(Tuples) →	'Rosé'	'Aix'
	'Rosé'	'Sancerre'

↑

(Domaine)

Attribut

- **Colonne d'une relation caractérisée par un nom.**
- Porteur de sens : Souvent différent du domaine qui supporte l'attribut.

Exemple

- La relation précédente comportait deux attributs Couleur et Cru de même nom que leur domaine associé.
- Le premier attribut peut avantageusement être remplacé par le nom Type.
- Il faut pouvoir distinguer deux colonnes qui prendraient leurs valeurs dans le même domaine.
- Le choix du nom d'un attribut est aussi important que le choix du nom d'une fonction lorsque l'on programme.

Schéma d'une relation

- **Nom de la relation suivi de la liste des attributs et de la définition de leur domaine.**
- La structure d'une relation caractérisée par les concepts de domaines et d'attributs est un invariant pour une relation (elle ne change pas en fonction du temps).

Exemple

La relation précédente avait comme schéma de relation
R(Type : {'Rouge','Rosé','Blanc'},Cru : {'Aix','Sancerre'})

Définitions complémentaires

- ARITE d'une relation: Nombre fixe de ses attributs (colonnes).
- CARDINALITE d'une relation: Nombre de n-uplets (lignes).
- SCHEMA de base de données relationnelles : ensemble de schéma de relations.

Contraintes d'intégrité (1) : Clé primaire

Des contraintes sur le schéma de la BD (ensemble des schémas des relations) permettant en partie de garantir la cohérence de la BD.

- Toute relation possède une **clé primaire** : un ensemble minimal d'attributs dont les valeurs permettent de distinguer deux tuples de la relation.
- Par convention, on souligne les attributs participant aux clés primaires.
  - **Voiture**(<u>Immatriculation</u>,Couleur,NomModele)
  - **Modele**(<u>NomModele</u>,Marque)
  - **Commande_Produit**(<u>NumCmd,RefProduit</u>,QtéCom)

Clé primaire

- Les valeurs de la clé primaire permettent d'identifier de manière unique un tuple de la table.
- Impossible d'avoir deux tuples avec les mêmes valeurs pour les attributs de la clé primaire.

Voiture(Immatriculation,Couleur,NomModele)

'123XY34'	'Jaune'	'106'
'34UV62'	'Verte'	'106'
'345RT62'	'Verte'	'Megane'
~~'123XY34'~~	~~'Verte'~~	~~'106'~~
'234XU45'	'Bleue'	'Clio'

Contraintes d'intégrité (2) : Clé étrangère

- Clé étrangère : un attribut dont les valeurs appartiennent à l'ensemble des valeurs d'une clé primaire d'une autre relation.
- Une relation peut posséder une ou plusieurs clés étrangères.
- Par convention, on souligne en pointillés les clés étrangères.

Modele(<u>NomModele</u>,Marque)

Voiture(<u>Immatriculation</u>,Couleur,NomModele)

Ici, nous avons une clé étrangère entre les tables Modele et Voiture : l'attribut NomModele de la table Voiture **référence** l'attribut NomModele de la table Modele.

On parle de contrainte d'intégrité référentielle.

Clé étrangère

Modele

NomModele	Marque
'106'	'Peugeot'
'206'	'Peugeot'
'306'	'Peugeot'
'Clio'	'Renault'
'Espace'	'Renault'

Voiture

Immatriculation	Couleur	NomModele
'123XY34'	'Jaune'	'106'
'34UV62'	'Verte'	'106'
~~'345RT62'~~	~~'Verte'~~	~~'Megane'~~
'123XY36'	'Verte'	'106'
'234XU45'	'Bleue'	'Clio'

Comment obtenir le schéma d'une base de données ?

Le schéma d'une base de données (ou l'ensemble des relations ou le modèle relationnel de données) est dérivé du modèle conceptuel de données: modèle entité-association.

Modèle entité-association (EA)

- S'appuie sur les propriétés des relations entre ensembles.
- Toute conception peut être validée par un utilisateur non informaticien (modèle lisible).
- Une bonne conception du modèle entité-association permet d'avoir une base de donnée de bonne qualité.

Modèle entité-association (EA) : les entités

Entité

Ensemble d'individus ou d'objets intéressants pour le système d'information, et qui est décrit par les propriétés que partagent tous ces individus.

Propriété

- Elles peuvent être identifiantes : matricule, numéro, etc
- ou descriptives : nom, prénom, désignation, date de naissance, etc

Exemple

- Etudiant(num_étudiant,nom,prénom)
- Livre(côte,titre,année)

Modèle entité-association (EA) : les entités

Règles de constitution d'une entité

- Il faut qu'il y ait plus d'un individu dans l'entité.
- Toute entité est décrite par au moins une propriété identifiante.
- Toute entité reçoit un nom qui permet de la désigner.

Quelques informations

- Le nombre d'individus qui peuplent une entité s'appelle sa **population**.
- Dans le modèle EA, tous les individus ont des valeurs associées à toutes les propriétés.

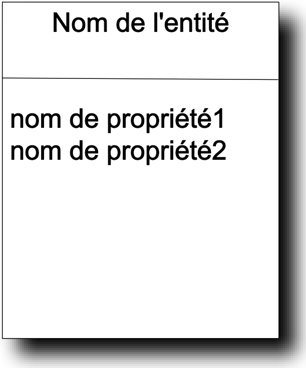
Règle de non redondance

Si une propriété sert à décrire une entité, alors elle ne peut figurer nulle part ailleurs que cette entité.

Remarque

- Des propriétés telles que “nom” et “prénom” peuvent apparaître plusieurs fois. Mais il ne s’agit pas de la même propriété.
- Par exemple, supposons que l’on ait une entité “ETUDIANT” et une entité “ENSEIGNANT”. Le nom, chez l’étudiant, sera désigné par “nom_étudiant”, et celui chez l’enseignant par “nom_enseignant”

Représentation graphique des entités



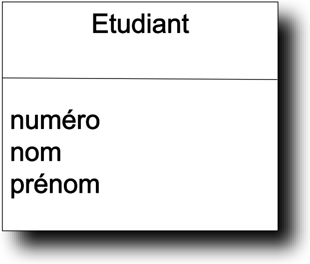
Identifier les entités

Identifiant

- Il peut être composé d'une seule propriété (ex: numéro).
- ou de plusieurs (ex: concaténation de plusieurs identifiants).
- Il est soit choisi par le concepteur, soit "calculé" (choisi par une méthode formelle).
- Il permet de reconnaître chaque individu (par sa valeur).
- Il dépend de la nature de la population.

Exemple: l'entité étudiant

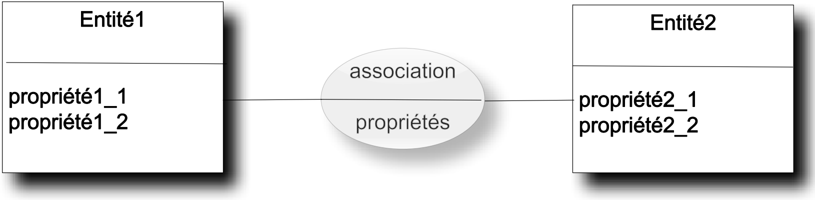
Définit le format de représentation de tous les étudiants.



Modèle entité-association (EA) : les associations

- Toutes les entités sont liées entre elles par des associations. L'existence d'une association est conditionnée par celle des entités qui la composent.
- Ces dernières sont des relations d'ensembles.
- Elles indiquent que tout ou partie de la population respective des entités est en relation.
- Les associations sont nommées.
- Elles peuvent avoir des propriétés propres.
- Elles sont de dimension (arité) et de type différent.

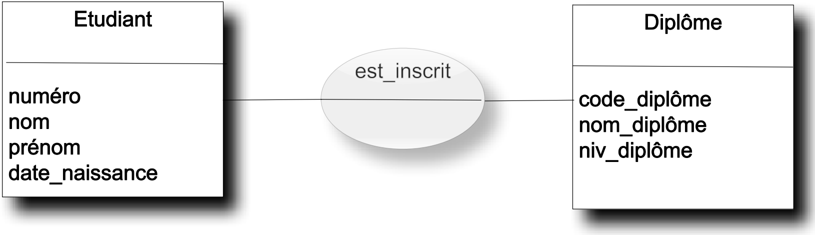
Représentation graphique d'une association



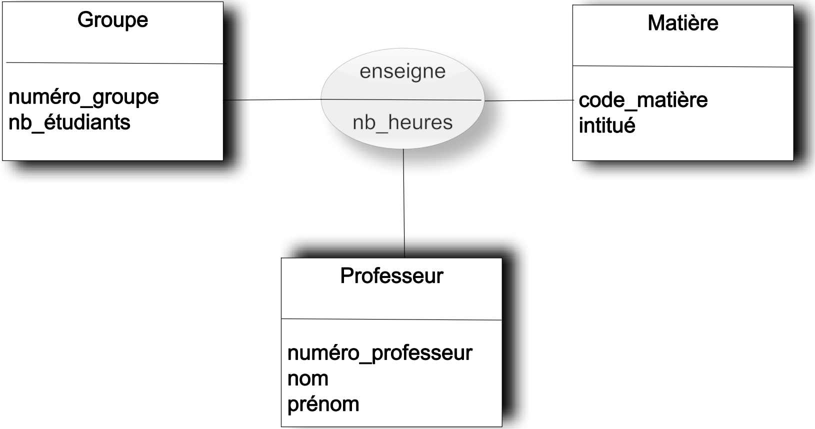
Arité des associations

- C'est le nombre d'entités qu'elle relie.
- Association unaire: l'entité est reliée avec elle-même.
- Association binaire: l'association relie deux entités.
- Ternaire, quaternaire, etc.
- En général, on dépasse rarement les associations quaternaires.

Arité des associations: Exemple d'une association binaire



Arité des associations: Exemple d'une association ternaire

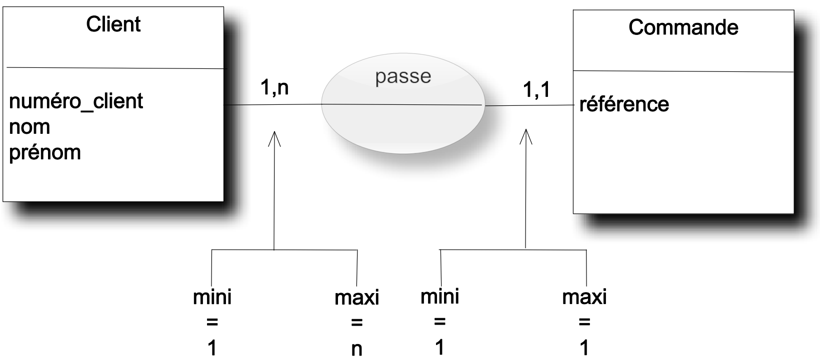


Cardinalité des associations

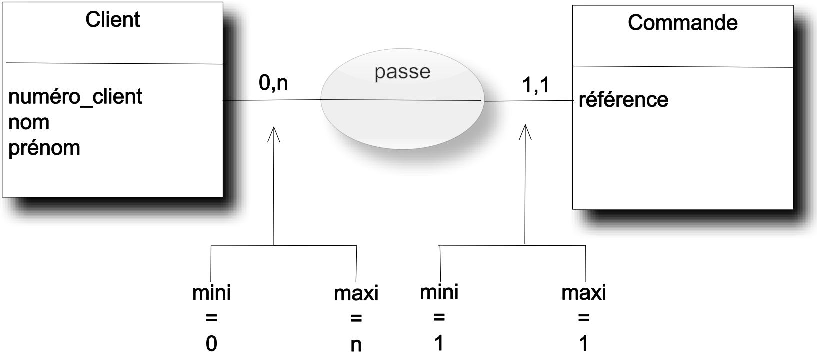
La cardinalité d'une association représente le nombre minimal ou maximal d'occurrences d'une entité pouvant prendre part à une association.

- Cardinalité minimale : nombre minimum de fois où une occurrence de l'entité participe aux occurrences de l'association. Cette cardinalité est en général égale à 0 ou 1.
- Cardinalité maximale : nombre maximum de fois où une occurrence de l'entité participe aux occurrences de l'association. Cette cardinalité est en général égale à 1 ou n (n signifie une cardinalité maximale indéfinie).

Cardinalité – Exemple 1

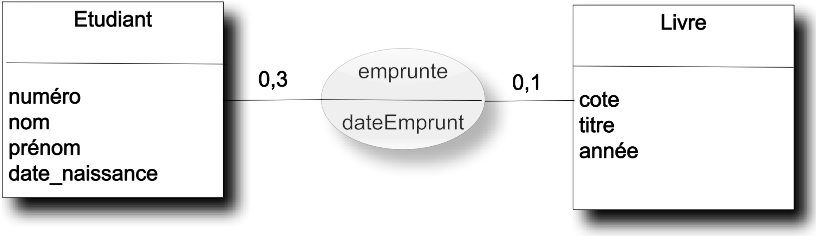


Cardinalité – Exemple 2



Cas particulier

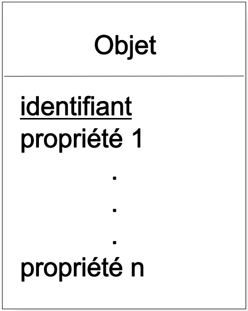
Il peut arriver que la cardinalité maximale soit connue et supérieure à 1.
Par exemple, considérons qu'un étudiant peut emprunter au maximum trois livres :



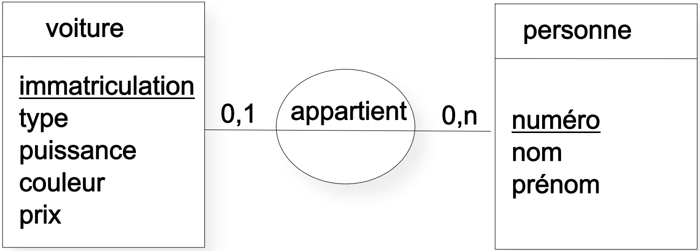
Transformation du modèle EA en modèle relationnel

Prise en compte des objets du modèle EA

- Un objet se transforme en relation (table).
- L'identifiant d'un objet devient la clé primaire de la relation.
- Les propriétés de l'objet deviennent les attributs de la relation.



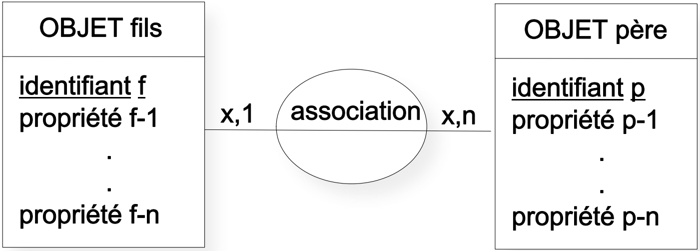
Exemple


voiture
immatriculation
type
puissance
couleur
prix
0,1
appartient
0,n
personne
numéro
nom
prénom

Prise en compte des associations du modèle EA

association PERE-FILS

C'est une association de type (0,1) - (0,n) ou (0,1) - (1,n) ou (1,1) - (0,n) ou (1,1) - (1,n)



Prise en compte des associations du modèle EA

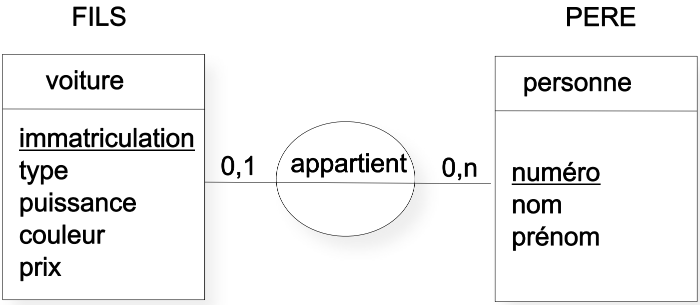
Idée générale

- Un fils n'a qu'un seul père.
- Un père peut avoir plusieurs fils.

Règles de traduction

- L'objet père devient la relation père.
- L'objet fils devient la relation fils.
- L'identifiant de la relation père devient également attribut de la relation fils. Cet attribut particulier est appelé clé étrangère.

Exemple


FILS
PERE
voiture
immatriculation
type
puissance
couleur
prix
0,1
appartient
0,n
personne
numéro
nom
prénom

Prise en compte des associations du modèle EA

Cas particulier

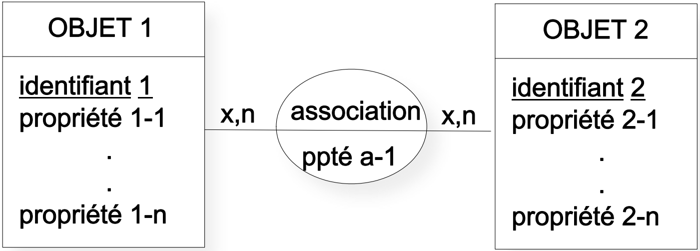
les relations père-monofils: n est ici égal à 1

On trouve une association de type (0,1) - (0,1) ou (0,1) - (1,1) ou (1,1) - (0,1) ou (1,1) - (1,1)

- L'association est symétrique.
- Il faut se fixer l'un des objets comme père, et l'autre comme fils.
- On applique alors les règles de transformation relatives à une association père-fils usuelle.

Association MAILLEE

C'est une association de type (0,n) - (0,n) ou (0,n) - (1,n) ou (1,n) - (0,n) ou (1,n) - (1,n) avec n $\neq$ 1

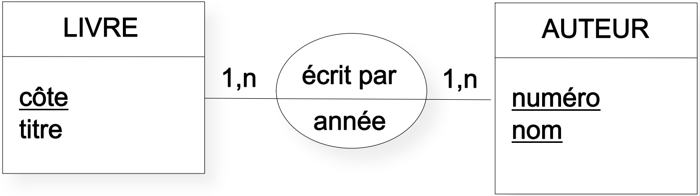


Prise en compte des associations du modèle EA

- L'association devient une relation pivot.
- L'identifiant de l'association devient la clé primaire de la relation pivot.
- Les propriétés de l'association deviennent les attributs de la relation pivot.

- OBJET1(<u>identifiant1</u>, propriété 1-1, $\cdots$, propriété 1-n)
- OBJET2(<u>identifiant2</u>, propriété 2-1, $\cdots$, propriété 2-n)
- ASSOCIATION(<u>identifiant1, identifiant2</u>, propriété a-1)

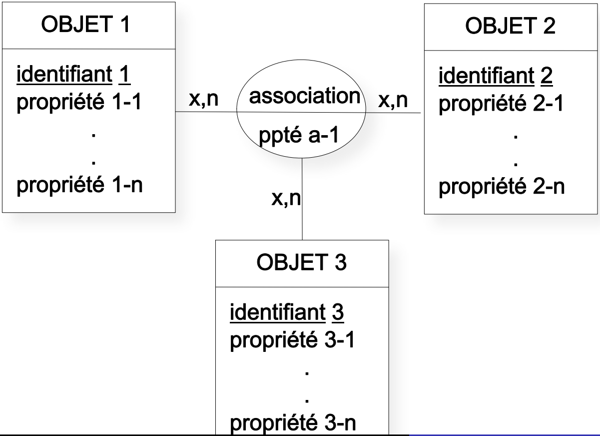
Exemple



Prise en compte des associations du modèle EA

association N-AIRE

On peut trouver des associations reliant plus de deux objets, que l'on appellera association n-aire.



- Chaque occurrence de l'objet 1 peut être reliée à un nombre quelconque de couples formés d'une occurrence de l'objet 2 et d'une occurrence de l'objet 3.
- Chaque occurrence de l'objet 2 peut être reliée à un nombre quelconque de couples formés d'une occurrence de l'objet 1 et d'une occurrence de l'objet 3.
- Chaque occurrence de l'objet 3 peut être reliée à un nombre quelconque de couples formés d'une occurrence de l'objet 1 et d'une occurrence de l'objet 2.

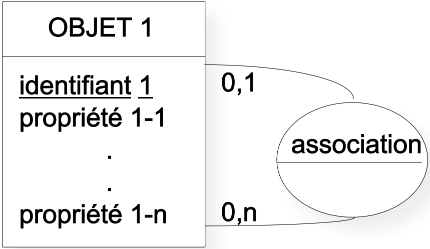
- OBJET1(<u>identifiant1</u>, propriété 1-1,$\cdots$, propriété 1-n)
- OBJET2(<u>identifiant2</u>, propriété 2-1,$\cdots$, propriété 2-n)
- OBJET3(<u>identifiant3</u>, propriété 3-1,$\cdots$, propriété 3-n)
- ASSOCIATION(<u>identifiant1, identifiant2, identifiant3</u>, propriété a-1)

Règle générale

Dans le cas d'une association liant plus de deux objets, on crée une relation par objet, plus une relation pivot dont la clé primaire est composée de l'ensemble des identifiants des objets reliés.

Association REFLEXIVE

C'est une association qui lie un objet à lui même.



Prise en compte des associations du modèle EA

- Dans le cas d'une association père/fils, on fera apparaître dans la relation une clé étrangère qui sera l'identifiant de l'objet lui-même. On prendra soin de le renommer pour le différencier de la clé primaire.
- Dans le cas d'une association maillée, on créera une relation pivot dont la clé primaire sera formée de deux identifiants de l'objet mis bout à bout. On prendra également soin de renommer l'un des deux afin de les distinguer.

Algèbre Relationnelle

Algèbre relationnelle

- L'algèbre relationnelle est une théorie mathématique dans laquelle sont définies les différentes opérations permettant de manipuler les relations.
- Une expression algébrique permet d'exprimer une requête.
- La valeur d'une expression algébrique correspond au résultat de l'évaluation de la requête.

Les opérateurs ensemblistes traditionnels

- Opérateurs binaires: à partir de 2 relations, ils en génèrent une nouvelle.

➜ Union, Différence, Produit Cartésien

Les opérateurs relationnels

- Répondent à certains besoins spécifiques de l'algèbre relationnelle.
- Ils génèrent également une nouvelle relation.

➜ Projection, Restriction, Jointure

Opérateurs. . .

Opérateurs dérivés

- Opérateurs pouvant étre obtenus par combinaison des opérateurs précédents.
- Utilisés pour plus de commodité.

➜ Intersection, division. . .

Union

Définition

Opération portant sur deux relations de même schéma R et S construisant une relation de même schéma ayant pour tuples ceux appartenant à R ou à S.

Notations

$R \cup S$, $Union(R, S)$, $Append(R, S)$

Lister les n-uplets qui appartiennent à cette relation **OU** à cette autre relation.

Exemple

Deux relations `Vins1` et `Vins2` de même schéma
`(Crus,Mill,Région,Type)`

Vins1

'Chenas'	1983	'Beaujolais'	'Rouge'
'Tokay'	1980	'Alsace'	'Blanc'
'Aix'	1995	'Provence'	'Rouge'

Vins2

'Bandol'	1988	'Provence'	'Rouge'
'Tokay'	1980	'Alsace'	'Blanc'
'Cassis'	2000	'Provence'	'Blanc'
'Aix'	1996	'Provence'	'Rouge'

Vins = Vins1 $\cup$ Vins2

'Chenas'	1983	'Beaujolais'	'Rouge'
'Tokay'	1980	'Alsace'	'Blanc'
'Aix'	1995	'Provence'	'Rouge'
'Bandol'	1988	'Provence'	'Rouge'
'Cassis'	2000	'Provence'	'Blanc'
'Aix'	1996	'Provence'	'Rouge'

Propriétés

- $R \cup S = S \cup R$
- $Card(R \cup S) \leq Card(R) + Card(S)$
- $Card(R \cup S) = Card(R) + Card(S) - Card(R \cap S)$
- $Arite(R \cup S) = Arite(R) = Arite(S)$

Différence

Définition

Opération portant sur deux relations de même schéma R et S construisant une relation de même schéma ayant pour tuples ceux appartenant à R mais pas à S.

Notations

$R - S$, $Diff(R, S)$, $Minus(R, S)$

Lister les n-uplets qui appartiennent à cette relation **MAIS PAS** à cette autre relation.

Exemple

Vins1

'Chenas'	1983	'Beaujolais'	'Rouge'
'Tokay'	1980	'Alsace'	'Blanc'
'Aix'	1995	'Provence'	'Rouge'

Vins2

'Bandol'	1988	'Provence'	'Rouge'
'Tokay'	1980	'Alsace'	'Blanc'
'Cassis'	2000	'Provence'	'Blanc'
'Aix'	1996	'Provence'	'Rouge'

Vins = Vins1 - Vins2

'Chenas'	1983	'Beaujolais'	'Rouge'
'Aix'	1995	'Provence'	'Rouge'

Propriétés

- $R - S \neq S - R$
- $Card(R - S) \leq Card(R)$
- $Card(R - S) = Card(R) - Card(R \cap S)$
- $Arite(R - S) = Arite(R) = Arite(S)$

Produit cartésien

Définition

Opération portant sur deux relations de schéma R et S $(Attr(R) \cap Attr(S) = \emptyset)$ construisant une relation ayant pour schéma la concaténation de R et de S et pour tuples toutes les combinaisons de tuples possibles.

Notations

$R \times S$, $Prod(R, S)$

Exemple

Vins1

'Chenas'	1983	'Beaujolais'	'Rouge'
'Tokay'	1980	'Alsace'	'Blanc'
'Aix'	1995	'Provence'	'Rouge'

Année

1988
2000

Vins = Vins1 $\times$ Année

'Chenas'	1983	'Beaujolais'	'Rouge'	1988
'Tokay'	1980	'Alsace'	'Blanc'	1988
'Aix'	1995	'Provence'	'Rouge'	1988
'Chenas'	1983	'Beaujolais'	'Rouge'	2000
'Tokay'	1980	'Alsace'	'Blanc'	2000
'Aix'	1995	'Provence'	'Rouge'	2000

Propriétés

- $R \times S = S \times R$
- $Card(R \times S) = Card(R) \times Card(S)$
- $Arite(R \times S) = Arite(R) + Arite(S)$

Projection

- Opération spécifique aux relations.
- Permet de supprimer des attributs d'une relation.

Définition

Opération portant sur une relation R construisant une relation contenant uniquement les attributs mentionnés en opérande et éliminant les tuples en double.

Notations

- $\Pi_{Att_i, Att_j \ldots}(R)$
- $Project(R, Att_i, Att_j, \ldots)$
- $R[Att_i, Att_j, \ldots]$

Lister **UNIQUEMENT** certaines colonnes de cette relation.

Exemple

Vins1

'Chenas'	1983	'Beaujolais'	'Rouge'
'Tokay'	1980	'Alsace'	'Blanc'
'Aix'	1995	'Provence'	'Rouge'
'Aix'	2000	'Provence'	'Rouge'
'Cassis'	2000	'Provence'	'Blanc'

$$Vins = \Pi_{Cru, Region}(Vins1)$$

'Chenas'	'Beaujolais'
'Tokay'	'Alsace'
'Aix'	'Provence'
'Cassis'	'Provence'

Propriétés

- $Card(Proj(R, A_1, A_2, \ldots)) \leq Card(R)$
- $Arite(Proj(R, A_1, A_2, \ldots)) \leq Arite(R)$

Restriction

- C'est également une opération spécifique. Elle sélectionne les tuples suivant un critère.

Définition

Opération portant sur une relation R construisant une relation de même schéma mais comportant uniquement les tuples qui vérifient la condition précisée en argument.

Notations

- $\sigma_{cond}(R)$
- $Select(R, cond)$
- $Restrict(R, cond)$

Lister **UNIQUEMENT** les nuplets **VERIFIANT** cette condition.

Exemple

Vins1

'Chenas'	1983	'Beaujolais'	'Rouge'
'Tokay'	1980	'Alsace'	'Blanc'
'Aix'	1995	'Provence'	'Rouge'
'Aix'	2000	'Provence'	'Rouge'
'Cassis'	2000	'Provence'	'Blanc'

$$\text{Vins} = \sigma_{Cru=Aix}(\text{Vins1})$$

'Aix'	1995	'Provence'	'Rouge'
'Aix'	2000	'Provence'	'Rouge'

Propriétés

- $Card(\sigma_{cond}(R)) \leq Card(R)$
- $Arite(\sigma_{cond}(R)) = Arite(R)$

Composition

- On dit que l'algèbre relationnelle est un langage fermé car chaque opération retourne une relation.
- On peut donc appliquer un opérateur de l'algèbre au résultat d'une autre opération.

Exemple

$\text{TouteRégion} = \Pi(\text{Vins1} \cup \text{Vins2}, \text{Région})$

'Provence'
'Beaujolais'
'Alsace'

- Opérateur essentiel à l'algèbre relationnelle.
- Compose deux relations à l'aide d'un critère de jointure.
- Peut être vue comme une extension du produit cartésien avec une condition permettant de comparer les attributs.

Définition

Opération portant sur deux relations R et S et construisant une relation qui contient la concaténation des tuples de R et de S vérifiant la condition de rapprochement.

Notations

$Join(R, S, cond)$, $R \bowtie_{cond} S$

Exemple

Vins1

'Chenas'	'Beaujolais'	'Rouge'
'Tokay'	'Alsace'	'Blanc'
'Aix'	'Provence'	'Rouge'

Année

1988
2000

V = Vins1 $\times$ Année

'Chenas'	'Beaujolais'	'Rouge'	1988
'Tokay'	'Alsace'	'Blanc'	1988
'Aix'	'Provence'	'Rouge'	1988
'Chenas'	'Beaujolais'	'Rouge'	2000
'Tokay'	'Alsace'	'Blanc'	2000
'Aix'	'Provence'	'Rouge'	2000

Vins $= \sigma_{Type='Rouge'}$ (V)
Vins = Vins1 $\bowtie_{Type='Rouge'}$ Année

'Chenas'	'Beaujolais'	'Rouge'	1988
'Aix'	'Provence'	'Rouge'	1988
'Chenas'	'Beaujolais'	'Rouge'	2000
'Aix'	'Provence'	'Rouge'	2000

Propriétés

- $Join(R, S, cond) = Join(S, R, cond)$
- $Card(Join(R, S, cond)) \leq Card(R) \times Card(S)$
- $Arite(Join(R, S, cond)) = Arite(R) + Arite(S)$
- $Join(R, S, cond) = Select(Prod(R, S), cond)$

Renommage

Définition

Opération permettant de renommer un attribut de la relation R. Le résultat est la même relation avec un schéma de relation différent. On peut renommer aussi une relation.

Intersection

Définition

Opération portant sur deux relations de même schéma R et S construisant une relation de même schéma ayant pour tuples ceux appartenant à R et à S.

Notations

$R \cap S$, $Inter(R, S)$

Propriétés

$R \cap S = R - (R - S) = S - (S - R)$

Lister les n-uplets qui appartiennent à cette relation **ET** à cette autre relation.

Exemple

Vins1

'Chenas'	1983	'Beaujolais'	'Rouge'
'Tokay'	1980	'Alsace'	'Blanc'
'Aix'	1995	'Provence'	'Rouge'

Vins2

'Bandol'	1988	'Provence'	'Rouge'
'Tokay'	1980	'Alsace'	'Blanc'
'Cassis'	2000	'Provence'	'Blanc'

Vins = Vins1 $\cap$ Vins2

'Tokay'	1980	'Alsace'	'Blanc'

Un petit exercice (1)

Voiture	Immatriculation	Marque	Annee	Prix	IdProprio
	'1111AA01'	'Toyota'	1997	16 000	'Id01'
	'2222BB02'	'Peugeot'	2000	31 200	'Id01'
	'3333CC03'	'Fiat'	1997	2 000	'Id03'
	'4444DD13'	'Fiat'	1995	30 300	'Id02'
	'5555EE62'	'Renault'	1997	21 000	'Id02'
	'6666FF59'	'Opel'	1999	2 900	'Id01'
	'7777ZZ75'	'Ford'	1998	22 222	'Id03'

Personnes	IdProprio	Nom	Prenom	Naissance
	'Id01'	'Martin'	'Paul'	01/02/1967
	'Id02'	'Duval'	'Jean'	03/09/1980
	'Id03'	'Dupond'	'Laurence'	01/01/1945
	'Id04'	'Durand'	'Julie'	03/03/1985

Un petit exercice (2)

1. Liste des immatriculations.
2. Liste des voitures de 1996.
3. Voitures qui appartiennent au proprio Id01.
4. Liste des voitures entre 10000 et 20000 euros.
5. Différentes marques de voitures.
6. Id des propriétaires ayant des voitures de plus de 20000 euros.