Domaine D : ensemble de valeurs

Exemples : chaînes de caractères de longueur 25, entiers

Attribut A (ou colonne) : nom de constituant (ou variable) prenant ses valeurs dans un domaine

Exemples : département de naissance d'un étudiant, N°INE d'un étudiant, couleur d'une voiture

Relation R (ou table) (définition en extension)

Partie (ou sous-ensemble) du produit cartésien $D_1 \times D_2 \times \cdots \times D_n$ sur les attributs $A_1, A_2, \ldots, A_n$ dont chaque élément $(d_1, d_2, \ldots, d_n)$ est un **tuple** (ou n-**uplet**) avec $d_i \in D_i$ le domaine de l'attribut A_i pour tout $1 \leq i \leq n$

R

A_1	A_2	$\cdots$	A_n
d_1	d_2	$\cdots$	d_n
d'_1	d'_2	$\cdots$	d'_n
$\vdots$	$\vdots$	$\ddots$	$\vdots$
d''_1	d''_2	$\cdots$	d''_n

avec $\{d_i, d'_i, \ldots, d''_i\} \subseteq D_i$ pour tout $1 \leq i \leq n$

Chaque cellule (intersection d'une ligne et d'une colonne) admet une et une seule valeur

Degré d'une relation : nombre d'attributs (ou de colonnes)

Cardinal d'une relation : nombre de tuples (ou de lignes)

Exemple :

Etudiants

N°INE	NomEtu	DepartNaissEtu
5	DURAND	33
2	LEROI	40
4	MARTIN	47
7	LEROI	33
3	DUPOND	17

Schéma de Relation $R(A_1, A_2, \ldots, A_n)$ (définition en compréhension)

Liste des attributs d'une relation

Exemple : Etudiants (N°INE , NomEtu , DepartNaissEtu)

Remarque : la définition d'une relation (en terme ensembliste) implique que tous les tuples sont différents deux à deux; cependant, les SGBDR ne vérifient pas cette propriété (doublons autorisés) et cela conduit à choisir systématiquement une clé (parmi les clés candidates) pour chaque relation

Clé (primaire ou principale)

Attribut ou groupe d'attributs minimal qui identifie de façon unique chaque tuple d'une relation

Exemples : numéro INE de la relation Etudiants, numéro d'immatriculation complet (chiffres, lettres et département) de la relation Voitures

Contre-exemple : nom des étudiants (homonymies)

Clé étrangère

Attribut(s) d'une relation qui correspond à la clé (primaire) d'une autre (ou de la même) relation

Exemple : le N°INE de la relation Voitures (indiquant quel étudiant possède cette voiture)

Remarque : la clé étrangère ne fait pas partie de la clé primaire de la même relation

Schéma [de la base de données] relationnel[le]

Collection de schémas de relations et de contraintes d'intégrité, qui représentent l'univers réel du système d'information

::= conditions à vérifier par toutes les relations : dépendance entre attributs, prédicat sur les valeurs d'un ou plusieurs attributs des tuples d'une relation, ...

Remarque : on distingue des contraintes statiques/dynamiques, sur les tuples/attributs/relations/inter-relations

Contrainte de clé (primaire) : une relation possède une clé primaire

> Exemple : l'attribut N°INE pour la relation Etudiants

Contrainte de clé étrangère : l(es) attribut(s) d'une relation ont la même sémantique que la clé primaire correspondante d'une autre (ou de la même) relation

> Exemple : le N°INE des relations Voitures et Etudiants

Contrainte d'intégrité référentielle : l'ensemble des valeurs d'une clé étrangère est inclu dans ou égal à l'ensemble des valeurs de la clé primaire correspondante

> Exemple : valeurs(Voitures.N°INE) $= \{5, 4\} \subseteq$ valeurs(Etudiants.N°INE) $= \{5, 2, 4, 7, 3\}$

Contrainte existentielle : les valeurs d'un attribut sont définies pour tous les tuples de la relation

> Exemple : valeur d'indétermination (NULL) interdite pour le nom de l'étudiant

Contrainte de domaine : précision sur le domaine des valeurs d'un attribut (liste ou intervalle de valeurs, formats, ...)

> Exemples : la couleur d'une voiture doit être jaune ou orange ou rouge, un département (de naissance d'un étudiant ou d'immatriculation de voiture) doit être compris entre 1 et 97

Contrainte de valeur par défaut : valeur à affecter à un attribut lors de l'insertion d'un tuple (si l'instruction correspondante n'affecte pas de valeur !)

> Exemples : année courante pour l'année d'obtention d'un diplôme, 33 pour le département

Contrainte sur un tuple : prédicat (ou formule) mettant en jeu plusieurs attributs

> Exemple : le prix de vente d'un produit doit être supérieur à son prix d'achat

Contrainte de dépendance fonctionnelle : vérification des dépendances fonctionnelles entre groupes d'attributs

Contrainte de normalité : vérification du respect d'une forme normale

Contrainte de cardinalité : borne pour le nombre de tuples d'une relation

> Exemple : un étudiant ne peut pas avoir obtenu plus de cinq diplômes

Contrainte de type pré-post condition : condition vérifiée par la BD lors de son évolution, sous forme d'une condition explicite avant, pendant ou après une opération de mise à jour

> Exemples : ne supprimer un diplôme (les intitulés) que si aucun étudiant ne l'a obtenu, n'insérer une voiture que si l'étudiant qui la possède est déjà enregistré

Règle active : action(s) à exécuter suite à un changement d'état de la BD

> Exemple : dès qu'un étudiant est supprimé, supprimer les voitures qu'il possède et les diplômes qu'il a obtenus

... et toute autre **contrainte spécifique**

Définition

Soient X et Y deux ensembles d'attributs. On dit que X détermine Y (ou Y dépend fonctionnellement de X) noté X $\longrightarrow$ Y lorsque deux tuples qui ont une même valeur sur les attributs de X implique qu'ils ont même valeur sur les attributs de Y (ou encore qu'à toute valeur de X n'est associée qu'une et une seule valeur de Y, dans toute extension de la relation)

Ex. : IntitAbrege $\longrightarrow$ IntitComplet
IntitComplet $\longrightarrow$ IntitAbrege
N^{o}INE $\longrightarrow$ NomEtu,DepartNaissEtu
N^{o}INE,IntitAbrege $\longrightarrow$ Annee
N^{o}INE,IntitComplet $\longrightarrow$ Annee
N^{o}ImmatChiffres,N^{o}ImmatLettres,N^{o}ImmatDepart $\longrightarrow$ Couleur,N^{o}INE,NomEtu,DepartNaissEtu

Propriétés des dépendances fonctionnelles

Réflexivité : X $\longrightarrow$ X' avec X' $\subseteq$ X

Projection : X $\longrightarrow$ Y,Z $\Longrightarrow$ X $\longrightarrow$ Y et X $\longrightarrow$ Z

Additivité : X $\longrightarrow$ Y et X $\longrightarrow$ Z $\Longrightarrow$ X $\longrightarrow$ Y,Z

Augmentation : X $\longrightarrow$ Y $\Longrightarrow$ X,Z $\longrightarrow$ Y

Transitivité : X $\longrightarrow$ Y et Y $\longrightarrow$ Z $\Longrightarrow$ X $\longrightarrow$ Z

Pseudo-transitivité : X $\longrightarrow$ Y et Y,W $\longrightarrow$ Z $\Longrightarrow$ X,W $\longrightarrow$ Z

Théorème : toutes les dépendances fonctionnelles peuvent être générées à partir d'un ensemble d'entre elles par les propriétés de réflexivité, augmentation et transitivité (règles d'Amstrong)

Typologie des dépendances fonctionnelles X $\longrightarrow$ Y

Canonique : Y est réduit à un seul attribut

Triviale : Y $\subseteq$ X

Élémentaire : il n'existe pas X' $\subset$ X tel que X' $\longrightarrow$ Y existe

Directe : il n'existe pas Z tel que X $\longrightarrow$ Z existe, Z $\longrightarrow$ Y existe, Z $\longrightarrow$ X n'existe pas

Hyper-graphe des dépendances fonctionnelles

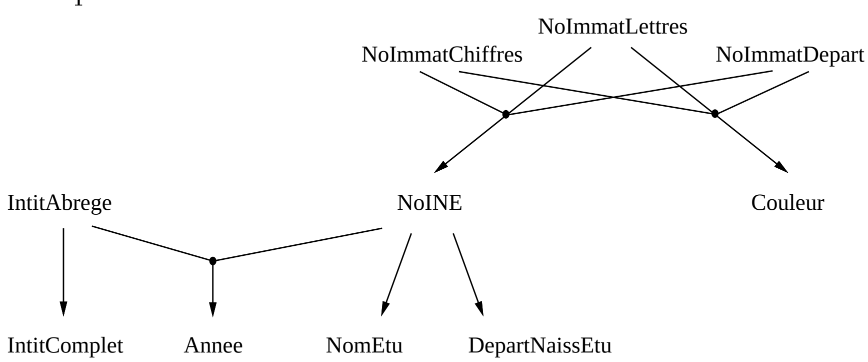
Sommets : attributs

Hyper-arcs : dépendances fonctionnelles

La fermeture transitive d'un ensemble de dépendances fonctionnelles consiste à appliquer systématiquement (tant que cela est possible) la propriété de transitivité

La couverture minimale d'un hyper-graphe de dépendances fonctionnelles est un ensemble de dépendances fonctionnelles élémentaires F vérifiant qu'aucune dépendance fonctionnelle n'est redondante ($\forall$ dépendance fonctionnelle f, F $\setminus$ {f} $\neq$ F) et que toute dépendance fonctionnelle est dans la fermeture transitive de F

Exemple :



Notation : R (C_1 , ... , C_k , A_1 , ... , A_l) avec (C_1 , ... , C_k) clé primaire de R

Première forme normale (1FN)

Les attributs d'une relation ne faisant pas partie de sa clé primaire sont fonctionnellement dépendant de la clé primaire de cette relation
(i.e. qu'aucun attribut hors clé n'est à valeur multiple pour une valeur de clé donnée)

$C_1,\ldots,C_k \longrightarrow A_1,\ldots,A_l$

Deuxième forme normale (2FN) — 1FN + élémentarité

Les attributs d'une relation ne faisant pas partie de sa clé primaire sont en dépendance fonctionnelle élémentaire avec la clé primaire de cette relation
(i.e. qu'il n'y a pas de dépendance fonctionnelle entre une partie de la clé primaire et un attribut hors clé)

$\nexists$ C $\subset \{C_1,\ldots,C_k\}$ et A_j tel que C $\longrightarrow A_j$

Troisième forme normale (3FN) — 2FN + non transitivité

Les attributs d'une relation ne faisant pas partie de sa clé primaire sont en dépendance fonctionnelle élémentaire directe avec la clé primaire de cette relation
(i.e. qu'il n'y a pas de dépendance fonctionnelle entre attributs hors clé)

$\nexists$ A',A" $\subset \{A_1,\ldots,A_l\}$ et A' $\cap$ A" $= \emptyset$ tel que A' $\longrightarrow$ A"

Théorème

Toute relation admet une décomposition sans perte ni redondance en troisième forme normale
(i.e. que chaque attribut d'une relation ne faisant pas partie de sa clé primaire ne dépend que de cette clé et d'aucune des parties de cette clé)

Autres formes normales

Troisième forme normale de Boyce-Codd-Kent

$\nexists$ A $\subset \{A_1,\ldots,A_l\}$, C $\subset \{C_1,\ldots,C_k\}$ tel que A $\longrightarrow$ C

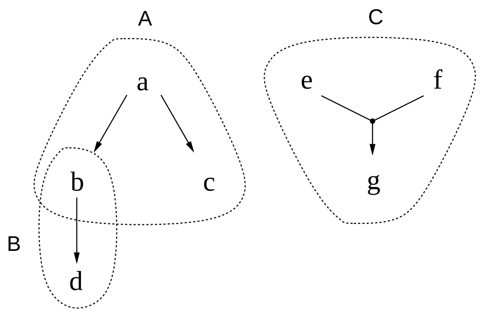
Quatrième forme normale

Cinquième forme normale

...

Remarque

Une fois l'ensemble des dépendances fonctionnelles établi, le schéma relationnel (schémas des relations et contraintes d'intégrité de clés primaires et référentielles) se déduit de l'hyper-graphe des dépendances fonctionnelles de couverture minimale. En effet, pour chaque sommet interne de l'hypergraphe on obtient une relation dont la clé primaire correspond au sommet interne et dont les attributs hors clé correspondent aux sommets fils du sommet interne; de plus, les attributs correspondant à des sommets fils qui sont eux-mêmes des sommets internes sont des clés étrangères



devient

$A\ (\ a\ ,\ b\ ,\ c\ )$
$B\ (\ b\ ,\ d\ )$
$C\ (\ e\ ,\ f\ ,\ g\ )$
a est la clé de la relation A
b est la clé de la relation B
$(\ e\ ,\ f\ )$ est la clé de la relation C
valeurs($A.b$) $\subseteq$ valeurs($B.b$)