Table des matières

MODÈLE RELATIONNEL DE DONNÉES

Le modèle relationnel de données a été défini en 1970 par Codd et les premiers systèmes commerciaux sont apparus au début des années 80. Le modèle relationnel est simple (3 concepts), facile à appréhender, même pour un non spécialiste et repose sur de solides bases théoriques qui permettent notamment de définir de façon formelle les langages de manipulation associés.

Le modèle relationnel représente l'information dans une collection de *relations*. Intuitivement, on peut voir une relation comme une table à double entrée, voire même comme un fichier. Chaque ligne de la table (appelée nuplet ou tuple) peut être vue comme un fait décrivant une entité du monde. Une colonne de la table est appelée un *attribut*.

I. Définition formelle

- un **domaine** D est un ensemble de valeurs atomiques. Le terme atomique signifie que ces valeurs sont considérées comme insécables au niveau du modèle.

- un **schéma de relation** `R`, dénoté par `R(A1, A2, ..., An)`, est un ensemble d'attributs `R = {A1, A2, ..., An}`. A chaque attribut `Ai` est associé un domaine `Di`, `Ai` indiquant le rôle joué par le domaine `Di` dans la relation `R`. Un schéma de relation décrit une relation et représente l'intension de celle-ci. Le degré de la relation est le nombre d'attributs de celle-ci.

  Exemple : `ETUDIANT(Nom, No-ss, adresse, age, diplôme)`

- une **relation** `r` sur le schéma de relation `R(A1, A2, ..., An)` est un ensemble de nuplets `r= {t1, t2, ..., tn}`. `r` est souvent appelée*extension* du schéma `R`.

  On peut également définir une relation à partir du produit cartésien des domaines de son schéma `R` :

  `r(R) inclus-ou-egal dom(A1) X dom( A2) X ...X dom(An)`

Il est important de noter qu'un schéma de relation est quasi invariant dans le temps, alors que l'extension représente les données présentes à un instant donné dans la base.

II. Caractéristiques des relations

- une relation est un ensemble de nuplets, il n'y a donc pas de notion d'ordre sur les nuplets,

•par contre un nuplet est un séquence ordonnée d'attributs,

•une valeur d'attribut est atomique mais peut être éventuellement nulle (valeur particulière qui indique que la valeur est manquante).

III. Contraintes d'intégrité

Un *schéma de base de données* est un ensemble de schémas de relation `S = {R1, R2, ..., Rn}` et un ensemble de contraintes d'intégrité `CI`. Une contrainte d'intégrité est une propriété du schéma, invariante dans le temps. On peut distinguer plusieurs catégories de contraintes. Les contraintes structurelles, définissent plus précisément la structure des associations entre les données (le modèle de données les supporte en partie) et les contraintes sur les valeurs donnent des relations entre les données (un chef gagne plus que ses subordonnés). La plupart des contraintes ne sont pas supportées pas le modèle de données et doivent donc être codées par les programmeurs dans des programmes d'application.

III.1 Contrainte d'intégrité sur les clés.

Par définition, tous les nuplets d'une relation sont distincts deux à deux (puisqu'il s'agit d'un ensemble). Il est également intéressant de définir des sous-ensembles du schéma qui permettent d'identifier de manière unique un nuplet (par exemple dans l'exemple de l'étudiant, l'attribut `No-ss` permet d'identifier un étudiant). Ces sous-ensembles du schéma s'appellent des *clés*. Le système doit donc garantir l'unicité des valeurs de clé (un seul nuplet étudiant peut avoir une valeur de `No-ss` donnée).

Il peut y avoir plusieurs clés pour une même relation (elles sont alors appelées clés candidates) et on choisit le plus souvent une clé particulière dite *clé primaire* qui servira d'identification privilégiée des nuplets. Cette clé sera indiquée de manière graphique en soulignant par exemple les attributs la composant.

Par exemple : `ETUDIANT(Nom, No-ss, adresse, age, diplôme)`

III.2 Contrainte d'intégrité sur les entités.

Une clé primaire ne peut contenir de valeur nulle.

III.3 Contrainte d'intégrité référentielle.

C'est une contrainte exprimée entre deux relations. Intuitivement cela consiste à vérifier que l'information utilisée dans un nuplet pour désigner un autre nuplet est valide, notament si le nuplet désigné existe bien. Par exemple, si on considère une relation `EMPLOYE(`**`no-ss`**`, nom, adresse, rôle, no-dept)` et une relation `DEPARTEMENT(`**`no-dept`**`, nom, no-ss-chef)`, un nuplet de`EMPLOYE` référence un nuplet de `DEPARTEMENT` via l'attribut `no-dept` (numéro de département) et un nuplet de `DEPARTEMENT` référence un nuplet de `EMPLOYE` via l'attribut `no-ss-chef` (numéro sécurité sociale du chef de département). Il est important de s'assurer que les nuplets référencés existent bien. On peut noter qu'un nuplet peut référencer un autre nuplet de la même relation. Dans ce cas, `no-dept` et `no-ss-chef` sont appelés des clés étrangères, puisque se sont des attributs clés d'une autre relation.

Cette contrainte implique un ordre dans la création ou la destruction des entités, puiqu'on ne peut créer un département si le nuplet correspondant au chef n'a pas été créé et on ne peut détruire un chef de département si le département existe toujours (ou bien si on n'a pas mis un autre chef).

Ces contraintes d'intégrité sont exprimées en SQL au niveau du langage de définition.

Les contraintes d'intégrité sont détaillées dans le chapitre concernant la protection des informations.