

INTRODUCTION AUX BASES DE DONNÉES

ALGÈBRE RELATIONNEL

Introduction

Le but de ce chapitre est d'assimiler les fondements des bases de données relationnelles et les systèmes de gestion de base de données relationnelles (SGBDR).

1. Limites des fichiers

L'utilisation des fichiers texte pour la sauvegarde et la manipulation des données d'un système d'informations est une solution simple à mettre en œuvre mais considérée comme limitée à cause de plusieurs inconvénients quelle présente, parmi ses inconvénients on peut citer :

1. La redondance d'informations : données dupliquées dans plusieurs fichiers.
2. Les problèmes d'intégrité de données. Pas de contrôle des données à enregistrer/modifier/supprimer dans les fichiers :
 - Type de données (entier, chaîne de caractère, ...).
 - Taille de données (nombres de caractères, chiffres, ...).
 - La cohérence entre les données.
3. La maintenance coûteuse des programmes :
 - Dépendance forte entre programmes (code source) et structures des fichiers (la manière dont les informations sont organisées).
4. Les problèmes de sécurité. Les données sont accessibles et modifiables par tous les utilisateurs :
 - Sans contrôle des droits d'accès aux données.
 - Sans fixer les responsabilités de chaque utilisateur.

2. Notion de base de données

Une base de données (database en anglais) est une grande quantité de données stockées (Une BD est composée d'une collection de fichiers), organisées et structurées pour être interrogées par un logiciel (et non directement par votre code source Python ou autres langages de programmation).

Elle désignait alors un ensemble d'informations numérisées, localisées dans un même lieu et sur un même support, et échangées entre plusieurs utilisateurs d'un système informatique.

Les traitements possibles sur une BD sont typiquement une combinaison d'opérations de recherches, de sélection selon des critères, de tri, de regroupement, et de concaténation.

L'utilisation des BD permet de palier toutes les limites d'utilisation des fichiers :

1. L'information est stockée dans un seul fichier. Pas de duplication de la même information (données dans un seul endroit), ce qui rend la mise à jour facile, sûr et rapide.
2. Pas de problème d'intégrité de données (un ensemble de règles relatives à l'accès et l'utilisation des informations) : Toutes les contraintes sont vérifiées avant le stockage/modification des données.
3. Maintenance des programmes indépendante du contenu des fichiers. Le développeur peut écrire son code et utiliser les données dont il a besoin sans s'intéresser de la manière dont ces informations sont organisées dans les fichiers de la BD.

4. La sécurité est assurée avec :

- Création des profils utilisateurs : noms utilisateurs, mots de passes, affectation des responsabilités : ajout, modification, lecture seule,...
- Contrôle des droits d'accès aux données : A chaque fois qu'un utilisateur demande d'effectuer une action sur des données, on vérifie si cet utilisateur est autorisé de faire ce type d'action sur ces données.
- Possibilité d'avoir une trace sur toutes les actions de chaque utilisateur (Enregistrer dans un fichier journal toutes les opérations effectuées par tous les utilisateurs).

Afin de pouvoir contrôler les données ainsi que les utilisateurs, la gestion de la base de données se fait grâce à un système appelé SGBD (système de gestion de bases de données) ou en anglais DBMS (Database management system).

3. Les systèmes de gestion de bases de données

Un système de gestion de bases de données (SGBD) est un ensemble de logiciels permettant de gérer la structure et le contenu des bases de données. Les SGBD sont des logiciels intermédiaires entre les utilisateurs et les bases de données. Le SGBD reçoit des demandes de manipulation de données et effectue les opérations sur les fichiers telles que rechercher, ajouter ou supprimer des enregistrements (et d'autres opérations plus avancées en cachant leur complexité).

Objectifs :

Les mécanismes du système de gestion de base de données visent à assurer

1. La cohérence :

- Il refusera qu'un usager ajoute une information si celle-ci existe dans la base de données et fait l'objet d'une règle d'unicité.
- Il refusera également de stocker une information qui n'est pas conforme aux règles de cohérence telles que les règles d'intégrité référentielle dans les bases de données relationnelles.

2. la confidentialité :

- Le logiciel refusera qu'un usager modifie ou supprime une information s'il n'y a pas été préalablement autorisé.

Exemples :

Les principaux SGBD : IBM DB2, Microsoft SQL server, Microsoft Access, Microsoft FoxPro, Oracle, Sybase, MySQL, PostgreSQL, **SQLite**,...

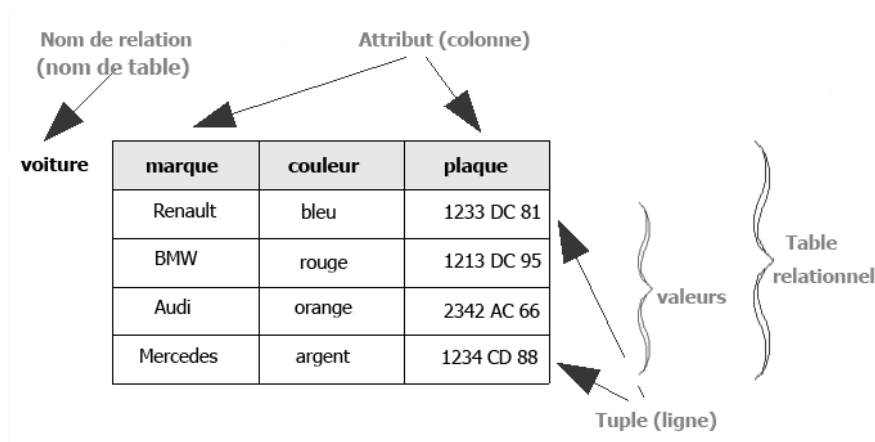
4. Modèle de données

Le schéma ou le modèle de données, est la description de l'organisation des données. Il se trouve à l'intérieur de la base de données et renseigne sur les caractéristiques de chaque type de donnée et les relations entre les différentes données qui se trouvent dans la base de données. Il existe plusieurs types de modèles de données : **relationnel**, objet, hiérarchique, réseau, ...

4.1. Le modèle relationnel

C'est le type de modèle de données le plus couramment utilisé pour la réalisation d'une base de données. Selon ce modèle, la base de données est composée d'un ensemble de **tables (relations)** avec des lignes et des colonnes dans lesquelles sont placées les données ainsi que les liens.

Selon le modèle relationnel, une relation est une table ou un tableau (matrice formée de n lignes et p colonnes) contenant un ensemble de n-uplets (tuples ou enregistrements).



Une base de données organisée selon un modèle de données de type relationnel, est dite **base de données relationnelle (BDR)** et gérée à l'aide d'un **SGBD Relationnel (SGBDR)**.

Le modèle relationnel est destiné à assurer l'indépendance des données et à offrir les moyens de contrôler la cohérence et d'éviter la redondance. Les règles de cohérence qui s'appliquent aux bases de données relationnelles sont l'absence de redondance ou de nul des clés primaires et l'intégrité référentielle. Avec ce modèle, n'importe quelle donnée contenue dans la base de données peut être retrouvée à l'aide du nom de la table, du nom de la colonne et de la clé primaire. Le modèle relationnel permet de manipuler les données comme des ensembles en effectuant des opérations de la théorie des ensembles.

5. Définitions

5.1. Attribut :

Un attribut d'une relation, noté **A**, est le nom d'une colonne de cette relation (tableau). Le domaine d'un attribut est l'ensemble des valeurs possibles qu'un attribut peut prendre (équivalent au type d'attribut en programmation).

Exemples :

- Le domaine de l'attribut filière est l'ensemble des filières en IPEIN (c'est l'ensemble des valeurs possibles que peut prendre l'attribut filière)
Domaine (filière) = {"PC", "PT", "MP", "SP", "ST", "SM"}
- Le domaine de numéro d'une classe est constitué des entiers strictement positifs entre 1 et 12.
Domaine (Numéro) = {1,2,3,4,5,6,7,8,9,10,11,12}
- Le domaine de l'attribut prénom est l'ensemble des chaînes de caractères possibles.
Domaine (Prénom) = {"Ali", "Ahmed", "Amel", ..., "Jihen"} = Text (chaîne de caractères)

5.2. Schéma relationnel :

Le schéma relationnel, noté « **S** », précise le format ou structure de la table : le nom de la relation ainsi que la liste des attributs avec leurs domaines.

Un schéma relationnel **S** est un n -uplet de la forme $\{A_1, \dots, A_n\} \in A^n$ où les attributs A_i sont **distincts deux à deux**.

Pour rappeler les domaines de ses attributs, On pourra noter : $S = ((A_1, \text{domaine}(A_1)), \dots, (A_n, \text{domaine}(A_n)))$

Exemples :

Schémas relationnels des relations Voiture et Classe :

Voiture = (marque, couleur, plaque)

Classe = ((filière, {"PC", "PT", "MP", "SP", "ST", "SM"}), (Numéro, N))

5.3. Une relation

Une relation **R** associée à un schéma relationnel $S = (A_1, \dots, A_n)$ est notée **R_S** ou **R(S)**, appelée aussi **table**.

$R(S)$ est formée d'un **ensemble** fini de n -uplets : $\text{domaine}(A_1) \times \dots \times \text{domaine}(A_n)$

$R(S) = D_1 \times \dots \times D_n$, où les D_i sont les domaines des attributs A_i du schéma **S**.

Une relation est un sous ensemble du produit cartésien des domaines des attributs.

Exemple :

Etudiant = (prénom, ville, âge)

La relation **R** associée au schéma relationnel Etudiant est un sous ensemble de toutes les combinaisons possibles entre les valeurs de domaines des attributs $A_i \in$ Etudiant

$$R(\text{Etudiant}) = \text{Domaine (prénom)} \times \text{Domaine (ville)} \times \text{Domaine (âge)}$$

Exemple :

$$R(\text{Etudiant}) = \{ ("Ahmed", "Tunis", 21), \\ ("Ala", "Nabeul", 20), \\ ("Ala", "Nabeul", 22), \dots \}$$

prénom	ville	âge
Ahmed	Tunis	21
Ala	Nabeul	20
Ala	Nabeul	22

5.4. n-uplet

On note par **e**, chaque n -uplet (tuple, élément, valeur ou enregistrement) du $R(S)$.

Si $e \in R(S)$ et $A \in S$, on note **e.A** la composante du n -uplet **e** associé à l'attribut **A**.

Propriétés des n -uplets :

- L'ordre des tuples n'est pas significatif.
- Si **R** est un ensemble, deux n -uplets **e** et **e'** diffèrent forcément au moins sur un attribut:

$$\forall e, e' \in R, \text{ si } e \neq e' \text{ alors } \exists A \in S, e.A \neq e'.A$$
- Chaque n -uplet est identifié par un attribut (un identifiant unique appelé "clef primaire") ou une combinaison de plusieurs attributs qui forme la clef.

Exemple :

Table : Etudiant

prénom	ville	âge
Ahmed	Tunis	21
Ala	Nabeul	20
Ala	Nabeul	22

5.5. nul

Dans les modèles de données relationnels, un attribut peut avoir une valeur nulle, indiquant que la donnée est absente, non disponible ou inapplicable.

Exemple :

Table : Etudiant

prénom	ville	âge
Ahmed	Tunis	21
Ala	Nabeul	20
Ala	Nabeul	null

5.6. Clé primaire

Dans les modèles de données relationnels, la clé primaire est un attribut (ou une combinaison de plusieurs attributs) dont le contenu est différent pour chaque enregistrement de la table, ce qui permet de retrouver un et un seul enregistrement.

Exemple :

Table : Etudiant

identifiant	prénom	ville	âge
1	Ahmed	Tunis	21
2	Ala	Nabeul	20
3	Ala	Nabeul	20

En effet, si on ne définit pas une clé primaire, il est possible d'insérer plusieurs lignes identiques dans une table, ce qui d'un point de vue ensembliste n'a pas de sens : un élément fait partie ou ne fait pas partie d'un ensemble. Si l'on veut appliquer strictement les règles des ensembles, il faut vérifier à chaque ajout dans une table que les lignes introduites ne sont pas déjà présentes.

5.7. Clé étrangère

Dans les modèles de données relationnels, une clé étrangère est un attribut qui contient une référence à une donnée connexe (généralement une valeur de la clé primaire de la donnée connexe).

Exemple :

Table : Etudiant

identifiant	prénom	ville	âge
1	Ahmed	Tunis	21
2	Ala	Nabeul	20
3	Ala	Nabeul	20

Table : Groupe

Nom_groupe	Id_Etudiant
SP2	1
PC1	2
PC1	3

L'attribut « id_etudiant » de la table « groupe » est une clé étrangère. Elle fait référence à l'attribut « identifiant » (clé primaire) de la table Etudiant.

5.8. Intégrité référentielle

Dans les modèles de données relationnels, il y a une situation d'intégrité référentielle lorsque toutes les données référencées par les clés étrangères sont présentes dans la base de données.

Exemple :

Dans l'exemple précédent, toutes les données référencées par la clé étrangère « id_etudiant » de la table Groupe sont présentes dans la colonne « identifiant » de la table Etudiant.

Dans le cas inverse, on dit qu'il y a une « violation d'intégrité référentielle ».

6. L'algèbre relationnelle :

L'algèbre relationnelle (AR) est une théorie mathématique proche de la théorie des ensembles. Cette algèbre est constituée d'un ensemble d'opérations formelles sur les relations.

Les opérations relationnelles (par exemple l'union, l'intersection, ou encore la différence) permettent de créer une nouvelle relation (nouvelle table) à partir d'opérations élémentaires sur d'autres tables .

Les Opérandes de ces opérations sont les relations du modèle relationnel et le résultat est une nouvelle relation qui peut à son tour être manipulée. Les principes de l'algèbre relationnelle sont beaucoup utilisés de nos jours par les SGBD pour la gestion des bases de données informatiques.

Les deux principaux types d'opérations relationnelles couramment utilisées sont :

1. Opérations unaires : Projection, Sélection et Renommage.
2. Opérations binaires : Intersection, Union, Jointure et Différence .

A. Opérations unaires

1) Projection

Soit $R(S)$ une relation de schéma S et $X \subset S$. On appelle projection de R selon X la relation :

$$\pi_X(R) = \{e(X) \mid e \in R\}$$

Le schéma de la nouvelle relation $\pi_X(R)$ est donc X .

Exemple :

Table : Etudiant

identifiant	prénom	ville	âge
1	Ahmed	Tunis	21
2	Ala	Nabeul	20
3	Ala	Nabeul	20

$\pi_{(\text{prénom}, \text{ville})}(\text{Etudiant})$

prénom	ville
Ahmed	Tunis
Ala	Nabeul

Remarque : Le résultat de la projection ne contient pas des tuples identiques.

2) Sélection

Si $R(S)$ est une relation de schéma S , $A \in S$ et $a \in \text{domaine}(A)$,

On appelle sélection de R selon $A=a$ la relation obtenue en sélectionnant dans R uniquement les valeurs e telles que $e.A=a$

On la note $\sigma_{(A=a)}(R)$, on a donc : $\sigma_{(A=a)}(R) = \{e \in R \mid e.A=a\}$

Exemple :

On considère par exemple la relation **Etudiant**, Sélectionner tous les étudiants qui habitent à « Nabeul ».

Table : Etudiant

identifiant	prénom	ville	âge
1	Ahmed	Tunis	21
2	Ala	Nabeul	20
3	Ala	Nabeul	20

$R1 = \sigma_{(\text{ville}=\text{"Nabeul"})}(\text{Etudiant})$

identifiant	prénom	ville	âge
2	Ala	Nabeul	20
3	Ala	Nabeul	20

3) Renommage

Le renommage est un opérateur unaire qui produit une relation identique à la relation de départ, ayant les mêmes enregistrements et le même schéma sauf l'attribut qui a été renommé.

Soit $S = (A_1, \dots, A_n)$ et B un attribut tel que $\text{dom}(B) = \text{dom}(A_i)$

Pour renommer A_i par B , on note :

$$\alpha_{(A_i \leftarrow B)}(S) = (A_1, \dots, (i-1), B, (i+1), \dots)$$

Exemple :

Renommer l'attribut « identifiant » par « numero_inscri » dans le schéma de la relation Etudiant.

Table : Etudiant

identifiant	prénom	ville	âge
1	Ahmed	Tunis	21
2	Ala	Nabeul	20
3	Ala	Nabeul	20

$R2 = \alpha_{(\text{identifiant} \leftarrow \text{numero_inscri})}(\text{Etudiant})$

numero_inscri	prénom	ville	âge
1	Ahmed	Tunis	21
2	Ala	Nabeul	20
3	Ala	Nabeul	20

B. Opérations binaires (Opérateurs ensemblistes)

Si deux relations ont **le même schéma (ou schémas compatibles)** alors il est possible de leur appliquer des opérateurs ensemblistes. Deux relations ont des schémas compatibles si :

- Les deux relations ont le même nombre d'attributs.
- Les attributs de deux relations de même rang ont des domaines compatibles.

Exemple :

$R1 = \{(A : \text{Entier}), (B : \text{réel}), (C : \text{Texte})\}$

$R2 = \{(A : \text{réel}), (B : \text{réel}), (D : \text{Texte})\}$

Parmi ces opérateurs, on utilisera en particulier l'union ($\cup$), l'intersection ($\cap$) et la différence ($-$).

On considère, deux relations *livre1* et *livre2* sur lesquelles on peut appliquer les opérateurs ensemblistes.

Livre 1		
Titre	Auteur	Année
Python3	Ali	2002
scilab	Sami	2013
scilab	Sami	2016

Livre 2		
Titre	Auteur	Année
Python3	Ali	2002
Java 7	Amir	2013

1) Union

L'union de deux relations $R1(S)$ et $R2(S)$ est l'ensemble des valeurs (tuples) comprises dans $R1$ ou dans $R2$.

Notation : $R3 = R1 \cup R2$ de schéma S

La relation $R2$ résultat de l'union de $R1$ et $R2$ n'aura pas des n-uplets doublons.

Exemple : $\text{Livre1} \cup \text{Livre2} = ?$

Livre 3		
Titre	Auteur	Année
Python3	Ali	2002
scilab	Sami	2013
scilab	Sami	2016
Java 7	Amir	2013

2) Intersection

L'intersection de deux relations $R1(S)$ et $R2(S)$ est l'ensemble des valeurs (tuples) comprises dans $R1$ et dans $R2$.

Notation : $R3 = R1 \cap R2$ de schéma S

Exemple : $\text{Livre1} \cap \text{Livre2} = ?$

Livre 3		
Titre	Auteur	Année
Python3	Ali	2002

3) Différence

La différence entre deux relations $R1(S)$ et $R2(S)$ est l'ensemble des valeurs (tuples) comprises dans $R1$ **mais pas** dans $R2$.

Notation : $R3 = R1 - R2$ de schéma S

Remarque : la différence est non commutative : $[R1 - R2] \neq [R2 - R1]$

Exemple : Livre1 - Livre2 = ?

Livre 3		
Titre	Auteur	Année
scilab	Sami	2013
scilab	Sami	2016

4) Jointure

La jointure consiste à fusionner deux tables avec des valeurs d'attributs communes.

Soient :

- S et S' deux schémas,
- $A \in S$ et $A' \in S'$ tels que $D(A) = D(A')$,
- et R et R' deux relations de schémas respectives S et S' .

On définit la jointure de R et R' selon A et A' par :

$$R \bowtie_{A=A'} R' = R[A=A']R' = \{e \in R \times R' \mid e.A = e.A'\} \\ = \sigma_{A=A'}(R \times R')$$

Remarque :

- Si $S \cap S' \neq \emptyset$, on renommera auparavant les attributs de S' .
- les deux relations doivent avoir au moins un attribut de même domaine.

Exemple : Document [numéro=NumLivre] Emprunt = ?

Document			
Numéro	Titre	Auteur	Année
1	Python3	Ali	2002
2	scilab	Sami	2013
3	scilab	Sami	2016

Emprunt			
NumLivre	NumEtudiant	Date_emprunt	Date_retour
1	1	11-10-2018	17-10-2018
1	2	07-10-2018	13-11-2018
3	5	01-09-2018	null

Résultat							
Numéro	Titre	Auteur	Année	NumLivre	NumEtudiant	Date_emprunt	Date_retour
1	Python3	Ali	2002	1	1	11-10-2018	17-10-2018
1	Python3	Ali	2002	1	2	07-10-2018	13-11-2018
3	scilab	Sami	2016	3	5	01-09-2018	null