



POLYTECH°
NICE SOPHIA



UNIVERSITÉ
CÔTE D'AZUR

17/08/2020

Bases de données relationnelles

Claudine Peyrat

Définition(s) informelle(s) d'une base de données

- Définition large: Tout ensemble de données stocké numériquement et pouvant servir à un ou plusieurs programme.
 - inclus à peu près tous les types de fichiers
- Définition plus restreinte : Un ensemble de données **numériques** apparentées qui possède une **structure** ; c'est à dire dont l'organisation répond à une logique systématique. On parlera de modèle logique de données pour décrire cette structure.

Modèles logiques

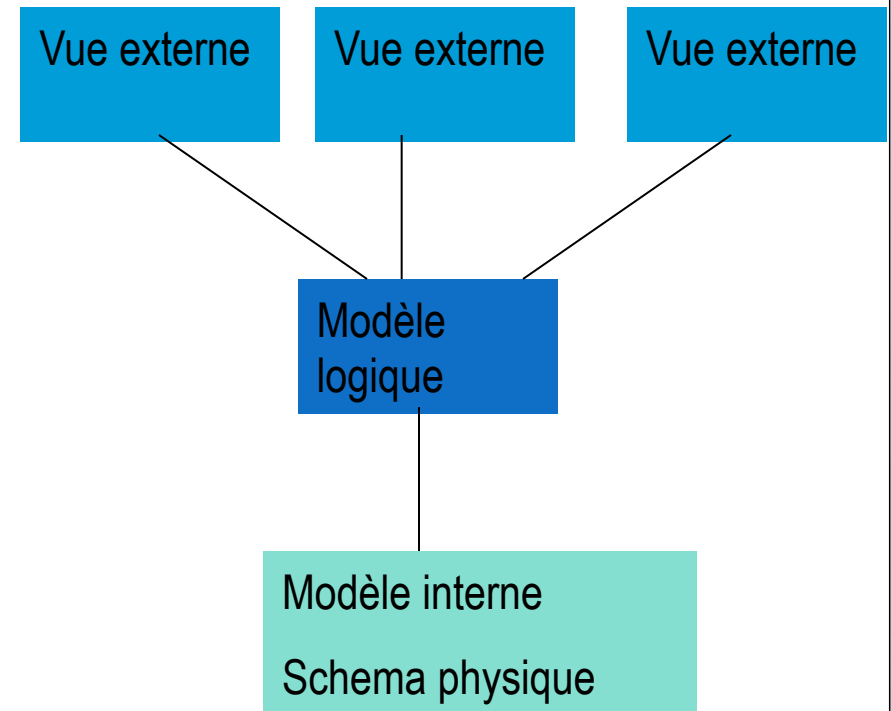
- 1960s: premières bases de données : hiérarchie ou réseau
- En 1970, Ted Codd (Médaille de Turing 1981), propose le modèle relationnel.
 - Dans ce modèle ensembliste, l'information est organisée en plusieurs tables ou relations homogènes qui peuvent être interrogées et combinées grâce à des opérateurs ensemblistes.
 - De la théorie à la pratique SQL
- Depuis 2000s : NoSQL. Différents modèles: clé-valeurs, document, graphe...

Fonctions d'un SGBD

- Décrire les données (DDL: data definition language)
- Modifier, Rechercher de l'information de manière fiable (DML data manipulation language)
 - Traiter de grands volumes de données
 - Traiter rapidement les données
- Contrôler les données (DCL data control language)
 - Sécuriser les accès aux données
 - Contrôler la qualité des données
- Partager les données (entre plusieurs applications)
- Gérer la concurrence des accès parallèles

Abstraction des données ANSI-SPARC

- Indépendance entre représentation physique et logique
- Offre différentes vues de la même structure





POLYTECH°
NICE SOPHIA



UNIVERSITÉ
CÔTE D'AZUR

17/08/2020

2. Le modèle relationnel

Où le schéma logique est représenté par des relations

Modèle relationnel : les fondamentaux

- Les contenants : Schéma d'une relation : un n-uplet d'attributs
 - Domaine
 - Attribut : Nom x Domaine
- Les contenus : Instance d'une relation : ensemble fini de n-uplets (ou tuple)

Domaines

- Un domaine est un ensemble de valeurs D non vide, que peut prendre un attribut. Il est caractérisé par un nom.
- Un domaine définit le contenu possible d'un attribut. Il définit donc des contraintes sur le contenu de chacun des tuples qui seront présents dans une instance de la relation.
- Ces domaines sont, dans SQL ANSI2, toujours de type scalaire (entiers, chaînes,...) et finis. On ne dispose d'aucun opérateur permettant de leur associer des types structurés.

EXEMPLES DE DOMAINES

- ENTIER
- REEL
- CHAINES DE CARACTERES
- COULEUR= {BLEU, BLANC, ROUGE}

Mais ceci n'est pas conforme au modèle relationnel...

- $POINT = \{(X:REEL, Y:REEL)\}$
- $TRIANGLE = \{(P1:POINT, P2:POINT, P3:POINT)\}$

Schéma relationnel (exemple)

- Schéma marque
- Attributs A : $A = (\text{IdM}, \text{NomM}, \text{Classe}, \text{IdProp})$
- Domaines :
 - $\text{dom}(\text{IdM}) = [1..99\ 999]$
 - $\text{dom}(\text{NomM}) = \text{tous les mots construits sur l'alphabet } \{A,...,Z, a,...,z, 0..9\}$
 - $\text{dom}(\text{Classe}) = [1..30]$
 - $\text{dom}(\text{IdProp}) = \text{tous les mots construits sur l'alphabet } \{A,...,Z, a,...,z, 0..9\} \text{ en se limitant à 100 caractères}$

Instance d'une relation , exemple

IdM	NomM	Classe	IdProp
122 233	renault21	24	renault
145 245	sun-sparc	27	sun
147 064	renegade	24	renault

- les lignes peuvent être permutées
- Les lignes ne peuvent être dupliquées.
- Le schéma de la relation définit les propriétés de chaque attribut (nom, type, **contraintes**, ...).

Les colonnes peuvent elles être permutées ?

- En principe OUI
- Mais ça complique l'écriture des tuples, imaginons une relation avec deux attributs A et B tous les deux du même domaine entier
- Si l'on ne met pas d'ordre sur les attributs, alors $\{12,13\}=\{13,12\}$
- En principe on note un tuple de cette relation sous la forme $\{12:A,13:B\}$
- $\{12:A,13:B\} \neq \{13:A,12:B\}$ et $\{12:A,13:B\} = \{13:B,12:A\}$
- Comme cette notation est lourde, on note plus souvent ces tuples (12,13) avec un ordre imposé sur les attributs par exemple A est le premier, B est le deuxième

Instances de relations

Soit R une relation, ayant comme ensemble d'attributs $A = \{A_1, \dots, A_n\}$.

On appelle **tuple** défini sur R , tout n -uplet de valeurs v_1, \dots, v_n associées respectivement aux attributs A_1, \dots, A_n , avec comme seule contrainte, $v_i \in \text{dom}(A_i)$.

On note la valeur v_i associée à l'attribut A_i du tuple t par : **$v_i = t.A_i$**

Exemple : tuple défini sur la relation MARQUE

$t = (122\ 233$: IdMarq,
 COCA : NomMarq,
 12 : Classe,
 CocaLtd : IdProp)

qui peut se représenter par la ligne :

IdMarq	NomMarq	Classe	IdProp
122 233	COCA	12	CocaLtd

On a alors dans cet exemple : **$t.Classe = 12$**

Formellement un tuple est ...

-un élément du produit cartésien des domaines

Instance de relation, exemple

$r(\text{MARQUE}) = \{t1, t2, t3\}$

$t1 =$ (1222 :IdM,
COCA :NomM,
12 :Classe,
CocaLtd:IdProp)
 $t2 =$ (1224 :IdM,
ORANGINA:NomM,
12 :Classe,
Perrier :IdProp)
 $t3 =$ (1226 :IdM,
PEPSI :NomM,
12 :Classe,
PepsiLtd :IdProp)

Contraintes

Contraintes vérifiées à tout moment par l'instance du schéma.

- facilitent la conception de la base
- aident au choix de représentations physiques (clés,).
- Le langage d'expression des contraintes peut être très varié

Contraintes - exemples

Exemple : Contraintes sur MARQUE :

Deux marques différentes ne peuvent pas avoir le même nom et la même classe

$\forall t1, t2 \in \text{marque}$

$(t1.\text{NomM} = t2.\text{NomM} \wedge t1.\text{Classe} = t2.\text{Classe})$

\Rightarrow

$(t1.\text{IdM} = t2.\text{IdM} \wedge t1.\text{IdProp} = t2.\text{IdProp}))$

Contrainte particulière : notion de clé candidate

- Ensemble minimal d'attributs dont la connaissance des valeurs permet d'identifier un tuple unique de la relation considérée
- Exemple précédent (NomM, classe) est peut être une clé candidate . IdM aussi
- Parmi toutes les clés candidates, une sera **distinguée** et deviendra **la clé de la relation**

Contrainte de domaine

Contrainte d'intégrité imposant qu'un attribut d'une relation ne puisse prendre que des valeurs vérifiant une assertion logique.

Schéma de bases de données : plusieurs relations

En algèbre relationnelle, le **même nom d'attribut** utilisé dans deux relations distinctes véhicule le **même type d'information , donc même domaine**.
SQL ne tient pas compte de cette présupposition.

Il est cependant fortement recommandé de la suivre si l'on veut faciliter la cohérence de la conception

Des contraintes globales pour toute la base (i.e., faisant intervenir plusieurs relations) peuvent être définies.

Schéma de bases de données(suite)

Un schéma de base de données relationnelle est un ensemble $S = \{R_1, \dots, R_n\}$ de relations

dans lequel **deux attributs de même nom** dans deux relations différentes, **ont le même domaine**.

Schéma de bases de données(suite)

Une instance d'un schéma $S = \{R_1, \dots, R_n\}$, est un ensemble d'instances de relations, $r = \{r_1, \dots, r_n\}$, où chaque r_i est une instance de R_i

On peut ajouter au schéma S des contraintes globales, portant sur l'ensemble des tables constituant une instance du schéma. Toute instance vérifiant ces contraintes est dite acceptable pour S .

Contrainte particulière : contrainte référentielle

Contrainte d'intégrité portant sur une relation R_1 , consistant à imposer que la valeur d'un groupe d'attributs apparaisse comme valeur de clé dans une autre relation R_2 .

Contrainte référentielle -exemple

Marque (IdM, NomM, Classe, IdProp)

Societe (IdProp, NomSoc, Pays)

Le propriétaire d'une marque doit être une société

$\forall t \in \text{Marque}, \exists v \in \text{Societe } t.\text{IdProp} = v.\text{IdProp}$

Pour en savoir plus....

- http://georges.gardarin.free.fr/Livre_BD_Contentu/XX-TotalBD.pdf

OU

- http://www.info.univ-angers.fr/~gh/Pluripass/Db/coursBD_Rigaux.pdf