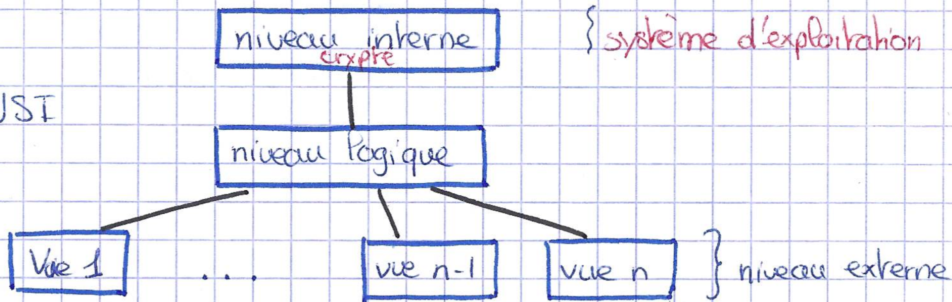
BD
01-1

Bases de Données (Databases) BD

Définition: Un ensemble de données persistantes : qui restent quand on ne s'en sert pas.
Réalité : ce qui existe toujours quand on y croit plus.

➡ une BD est une représentation de la réalité

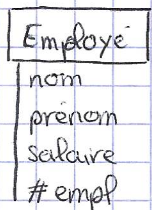
Niveaux ANSI



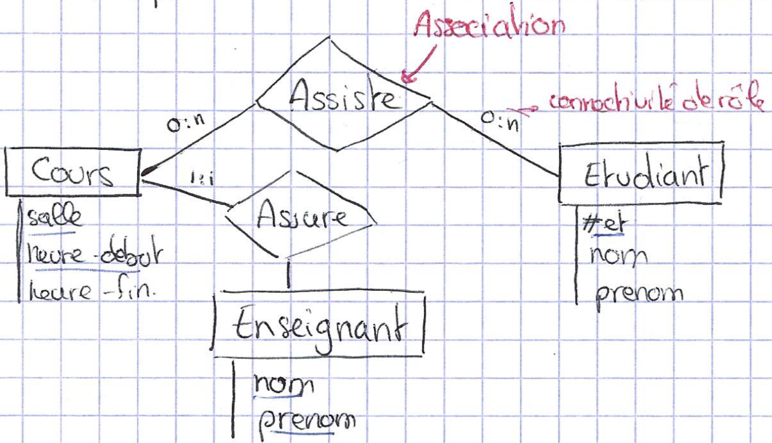
Modèle Entité Association (Entity-Relationship Model)

2 concepts : Entité quelquechose qui "est", qui existe.
On ne peut le connaître que par ses attributs

Exemple :

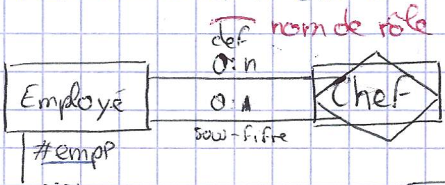


⚠ Parmi ces attributs, on doit avoir un identifiant.

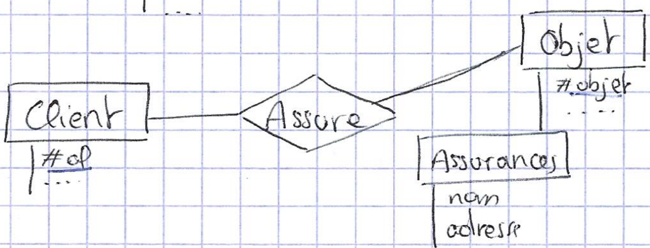


Une association permet de relier n entités → elle est identifiée par les entités qu'elle lie.

Exemples divers :



Seulement 4 connectivités : 0:1, 1:1, 0:n, 1:n



association ternaire

Modèle E-A

graphique ⊕ clair => idéal pour la conception d'une BD
=> employé dans MERISE

⚠ de ne pas ~~confondre~~ le compliquer : éviter
- entités "faibles"
- associations partiellement identifiées par autre chose

Garder les contraintes de P.P. Chen.
un attribut est atomique → chaîne de caractère
→ nombre
→ date
une entité, un identifiant, une association est identifiée par les entités qu'elle lie

les identifiants que l'on y met ne seront toujours ceux qu'on ~~a~~ implémentera
→ par une création d'identifiants artificiels inutiles
→ on ne choisit pas souvent d'identifier une association par les entités qu'elle relie

Modèle relationnel

La meilleure manière de crypter une BD c'est d'utiliser une table de hachage ie un tableau indexé via une fonction de hachage h. tab(h(donnée)) dans l'idéal injective, minimisant les collisions

Une Relation R(A_1:D_1, ..., A_n:D_n) + une clef "primaire" (=clef d'accès de la tab de hachage)

R : nom de relation ; D_1 : données ; A_n : nom d'attribut

Exemple : Etudiant (nom, prenom, date_naiss, #et)
Cours (titre, salle, heure, nom_ens, tel_ens)
Suit (#et, salle, heure, chargement)

schéma pas graphique du tout, peu lisible
c'est exactement ce qui est programmé
simple => normalisation comment obtenir un "bon" schéma ?

Ex de mauvais schéma : Cours_mauvais(salle, heure, #et, nom_et, p_et, @et). si je veux changer l'adresse d'un étu, il faut le faire ∀ cours où il est inscrit
→ au mieux inefficace, au pire incohérence ⚠ Redondances.

Langages BD

- Algèbre relationnelle

Opérateurs unaires :
- nom et prenom de la relation Emp.

Emp(#em, nom, pr, dept, salaire, prime)

→ Projection $\Pi_{nom,pr}$ Emp.

Dept(nom, budget, #chef)

→ Selection employés du dept Info. dept = σ_{info} Dept

BD
CH1.2

Axiome de sélection :

si t est un ensemble alors $\{x \in t \mid \quad (x)\}$ est un ensemble

Conditions de sélection :

- $A \,\theta$ valeur ou $A \,\theta\, B$ avec A, B attributs. θ comparateur $\{=, \neq, \geq, \leq\}$
(condition de sélection atomique)

- Si C et C' sont des conditions de sélection
alors $\neg C$, $C \wedge C'$, $C \vee C'$, $C \supset C'$ st des conditions de sélection.

Opérateurs essentiels : $\cup, \cap, -$

$R \cup S$, $R \cap S$, $R - S$ n'ont de sens que si R et S sont $\cup$ compat (m̂ arité)

par ex : dept sans employé $\quad \pi_{nom} Dept - \pi_{dep} Emp$.

fermeture : $\bowtie$

Déf : $R \underset{C}{\bowtie} S = \sigma_C (R \times S)$.

Cas particulier : équijointure = la condition de jointure est une cond. d'égalité

Dept avec les noms et prenom de leur def.

$\pi_{Dep.nom, \; emp.nom \; prenom} \left(Dept \underset{\#def = \#emp}{\bowtie} Emp \right)$ $\quad$ jointure "naturelle" = équijointure sur tous le attributs de m̂ nom.

Problème de renommage :

- nom et prénom des Emp ac les nom et prénoms de leurs def.

$\pi_{nom \; prenom, \; nom' \; prenom'} \left(Emp \underset{dept = Dept.nom}{\bowtie} Dept \underset{\#def = \#emp'}{\bowtie} Emp' \right.$

Le langage relationel est dit complet (au sens de Codd) ssi il permet d'exprimer l'algèbre relationelle.

Le seul opérateur coûteux est la jointure.

$R \underset{\theta}{\bowtie} S$

- au pire en O(taille R × taille S)
- quand C une équijointure et qu'il y a un index sur B (arbre).

$R \underset{A=B}{\bowtie} S$ est en $O(R) \times O(\log S)$.

Il faudrait faire le - possible de ⋈ res

? mettre tous les attributs dans la même relation ? FAUX.

Ex: Etudiant nouveau (#et, nom, ..., ~~titre cour, salle cours~~, h. cours)

Pb: le nom d'un étu est répété ∀ cours auquel il assiste, la salle de cours répét ∀ étu

Pb au niveau des mises à jour.

Il faut Etudiant (#et, nom, ...)
Cours (titre, salle, heure)
Suit (#et, titre cours)

perte info (on sait pas qui suit quoi)

Définition : R se décompose Sans perte d'Info (SPI)

en S et T ssi $R = S \bowtie T$

joint nat ou equijoint

Ex : Etudiant se décompose SPI en Etudiant × Cours, Suit

Un langage implanté SQL (Structural Query Language)

Π_{dept} Emp. (les depts qui ont au moins 1 employé

* SELECT DISTINCT dept FROM Emp.

* $\sigma_{dept=emp \wedge salaire > 10000}$ → SELECT * FROM Emp
WHERE dept = 'Info'
AND salaire > 10000

UNION INTERSECT MINUS

~~SELECT FROM MINUS~~
SELECT nom FROM Dept MINUS
SELECT dept FROM Emp

* jointure

* SELECT Dep.nom, Emp.nom
FROM Dept, Emp
WHERE #def = #emp

⚠ sans cond de jointure, c'est le produit cartésien

* SELECT Imbriqués (x) + burb

SELECT nom, budget FROM Dept
WHERE nom NOT IN
(SELECT dept FROM Emp)

Fn d'agrégation SELECT dept, COUNT (*)
FROM Emp
WHERE salaire > 10000
GROUP by dept

HAVING COUNT(*) > 4 :
↑
porte sur les groupes

← compte lignes (COUNT DISTINCT(*))

← WHERE avant GROUP BY
← avant calc le COUNT

BD
CH1.2

Tout cela c'est le langage de requêtes.

SQL comprend aussi un langage de mise à jour.

```
INSERT INTO
DELETE FROM
UPDATE
```

```
UPDATE Emp
WHERE dept = 'Info'
 AND salaire < 1000
 SET salaire = salaire * 1.1
```

Quand je fais une maj
- je bloque la relation de maj pour les autres users.
- je vois le nouvel état de la BD par les autres users

Les maj sont transmises à la BD quand j'applique COMMIT.

BDD

Ex : Etudiant (#et, cours, sport) en FNBC.

Ex de valeurs : (813, histoire, escrime)
si j'ai ces 2 val. (813, Français, savate)
(813, Français, escrime)

On dit qu'il y a une Dépendance Multivaluée (DMV)
#et $\twoheadrightarrow$ cours | sport.

Définition : la DMV $X \twoheadrightarrow Y | Z$ (X, Y, Z disjoints 2 à 2)
est vraie ds R ssi

$\forall s, t \in R \quad s.X = t.X \rightarrow \exists u \quad u.X = s.X \; [= t.X]$
$u.Y = s.Y$
$u.Z = t.Z$

Théorème de décomposition : si X, Y, Z partition de R

$X \twoheadrightarrow Y | Z \Leftrightarrow R$ se décompose SPI en $\pi_{X,Y}(R)$ et $\pi_{XZ}(R)$
$\Leftrightarrow R = \pi_{XY}(R) \bowtie \pi_{XZ}(R)$.

Exemples de DMV.

- $X \twoheadrightarrow Y | \phi$ et $X \twoheadrightarrow \phi | Z$ (DMV ~~triviales~~)
- $X \twoheadrightarrow Y | Z \Rightarrow X \twoheadrightarrow Z | Y$.
- si $X \rightarrow Y \Rightarrow X \twoheadrightarrow Y | Z$

Certains notent $X \twoheadrightarrow Y$ pour $X \twoheadrightarrow Y | \complement_R XY$.

Quatrième Forme Normale (4FN)

R est en 4FN ssi toute DMV $X \twoheadrightarrow Y | Z$ est telle que

* $Y = \phi$ ou $Z = \phi$ (DMV triviale)
  X est super clef ($X \rightarrow R$).

4°FN plus forte que FNBC

Toute relation peut se décomposer ~~SPI~~ SPI en des relations en 4°FN.

Cas particulier : $\phi \twoheadrightarrow Y | Z$ (Y, Z, partition de R).

$\Leftrightarrow R$ ~~=~~ $\pi_X(R) \times \pi_Z(R)$

Exemple Buveurs (#buv, biere, bar)

~~#buveur $\twoheadrightarrow$ biere | bar~~ NON
$\Leftrightarrow$ Buveurs se décompose en Aime (#buveur, biere)
$\bowtie$ Frequente (#buveur, bar).

On a perdu ge sr (bar bière)

On dit qu'il y a une dépendance de jointure (DJ)

5°FN toutes les DJs sont triviales ou se déduisent des clés.

On peut éviter les DJs (ainsi que certaines DMV de partie gauche vide).

en choisissant correctement les attributs

dans le cas des Buveurs

R (#buveur, bière_aimée, bar_fréquenté, bière servie)

(α) bar_fqté →→ bière_servie | #buveur, bière aimée

(β) #buveur →→ bar_fqté | bière aimée

(α) + pté de cent → Sert (bar, bière)
et un reste (bar_fqté, #buveur, bière aimée)

(β) th de comp → fréquente (bar, buveur) et ... (#buveur, bière)

Méthode de normalisation

→ on établit une liste des attributs → (!) . atomique pas de liste pas d'ensemble
. pas d'attribut calculables
. on veille à ce que chaque attribut ait une seule signification p/r au pb choisi.

→ on établit la liste des DFs et DMVs

→ on décompose à l'aide des thm de décomp

la + compacte possible DF réduites à gauche
dépendance pas de transitives
aucune qui puisse se déduire d'une autre (couverture minimale)

le but du jeu est de décomposer avec toutes les dépendances

=> commencer par les DFs (feuilles)

~~A → B et B → C~~

d'abord B → C

=> Second 2 DFs sont incluses

{nom, but, #salle → labo
labo → nom + but

d'abord la ① gde

(!) Appliquer rigoureusement le thm de décomp (DF) si

$X \to Y$ est vraie ds R
$X \cap Y = \emptyset$) =>

R se décomp SP l
$\Pi_{XY}(R)$ et $\Pi_{C_X}(R)$

Résumé normalisation

DFs + th décomposition

~~1FN, 2FN, obsolète~~

3FN idée de "dénormalisation"

FNBC

DMVs + th décomposition

4FN

cas particuliers gén évitable DJs (5FN)
+ DMV de partie gauche vide

| Exam : TD |