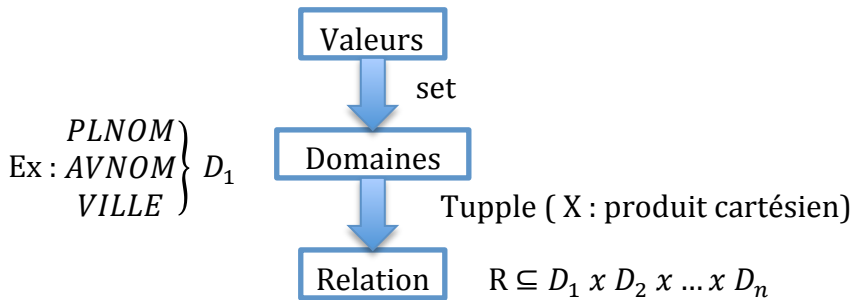


Base de donnée

Cours 1

Structuration « Modèle de Codd », paradigme « Valeur »

Modèle relationnel de Codd :



Exemple : $AVION \subseteq AVNOM \times AVNO \times CAP \times LOC$
 \rightarrow 4 domaines (Integer, String, ...)

Le domaine : Codd appel ça un type de données sémantique (sémantique datatype)

Sémantique : une partie de l'univers réel

Schéma à la sql :

```

PILOTE (PL#, PLNOM, ADR)
AVION(AV#, AVNOM, CAP, LOC)
VOL(V#, PL#, AV#, VD, VA, HD, HA)

```

Mais dans ce schéma il manque les domaines.

Toutes les valeurs appartiennent a un type de données sémantique : domaines.

Quels sont les adresses des pilotes qui conduisent un avion qui a le même nom qu'eux

```

SELECT ADR FROM PILOTE, AVION, VOL
PILOTE.PLNOM = AVION.AVNOM

```

Idiot car on a pas le même domaine de définition : 2 ensembles sémantiquement disjoint même si ils ont le même format syntaxique.

Ex : Un sceau dans un seau porté par un sot, les 3 sots ? tombent \rightarrow Impossible sémantiquement

En appliquant le constructeur tuple on obtient les relations

```

CREATE DOMAIN VILLE X(10)
CREATE DOMAIN AVNO INT
CREATE DOMAIN AVN X(12)

```

...

```
CREATE RELATION AVION (
    AV#    definie sur AVNO
    AVNOM definie sur AVN
    CAP    definie sur CAP
    LOC    definie sur VILLE
)
```

Dans le modèle de Codd : 2 structure de données
VALEURS → Ensemble de valeurs = DOMAINES

Une relation est un predicat (table, pas les domaines)
Relation = ensemble + predicat

Predicat : notion de proposition énoncé à valeur booléenne
Un predicat est une forme d'énoncé qui contient des variables et qui devient une proposition (n-uplet tuples) si on remplace les variables par des données (valeur)

X est mortel -> Socrate est mortel → predicat
PILOTE est un predicat a 3 variables
Une relation a un instant t est vrai

Quand les données sont simple (bureautique), pas de SQL.

Statut du
marché

SGBD-OO : Dans les BD Orienté Objet utilise l'ODMG
→ gestion des données scientifique et techniques (hierarchie et graphe)
SGBDR : (SQL2) simple et transactionnel (gestion, profuction, decision)
SGBD-OR : objet relationel → mobiquité

Marché SGBD-R < Marcgé SGBD-OO < SGBD-OR

TIPS :

- Transaction -> ACID
- Interface non procedurale (SQL)
- Persistance
- Structuration (schema)

Atomiticité : tout ou rien ne s'exécute

Cohérence : passé d'un etat cohérent à un autre

Isolation : verrou lecture avant de lire, ecriture avant d'ecrire _> coder bien

Durabilité : quelque soit la panne et le nombre de transaction, aucune perte

Un objet est une chose identifiable qui joue un rôle

BD <-> TIPS (avec T comprenant ACID, et S pour structuration, le S de SQL)

Objet : $\left\{ \begin{array}{l} \text{Reutilisabilité} \left\{ \begin{array}{l} \text{heritage} \\ \text{polymorphisme} \end{array} \right. \\ \text{Identifiable} \rightarrow (\text{OiD}, \text{valeur}) \\ \text{Complexité} \\ \text{Encapsulation (pendant du S)} \end{array} \right.$

Un objet est une structure de donnée qui vérifie les propriétés RICE

Double vision du monde :

- en programmation objet : encapsulé
- en base de donnée : structure

2 objets sont égaux si ils ont la même valeur mais pas le même OiD

2 objets sont identiques si ils ont le même OiD mais pas la même valeur

Mettre de l'objet dans le modèle relationnel de Codd

