

# Conception de BDD

• Démarche pour bien concevoir une BDD

=> Diagramme entité-association

=> On part d'une table universelle  
↳ tout est dans le même tableau  
↳ dangers d'incohérence

=> La modélisation va dépendre du cahier des charges

=> Faire en sorte que les données traduisent les contraintes de la réalité

=> 1<sup>er</sup> étape: faire un diagramme E/A

=> Il peut y avoir plusieurs choix "justes"

=> Dépendances fonctionnelles, elles vont servir uniquement

à traduire le modèle E/A

E/A  $\longleftrightarrow$  modèle  
langages relationnel

- Algorithme de normalisation
  - On prend le cahier des charges
  - On exprime les dépendances fonctionnelles
  - On applique l'algorithme

Diagramme E/A :

- on fait le travail de normalisation
- bien par un dessin (un peu intuitivement)

Les DF aident à la réalisation du diagramme E/A on le traduit en schéma relationnel

• Ex. :

• Si  $\text{numéro de sœur} \longrightarrow \text{nom}$   
alors on isole (numéro de sœur, nom)  
dans une seule table  
car si on écrit le numéro de sœur  
on aura le nom.

• Chaque entité est un ensemble d'objets

• Chaque association est un ensemble de couples  
↳ clés étrangères.

```
2
3 DROP TABLE IF EXISTS Herite ; --on détruit d'abord la table enfant (par la contrainte
4
5 DROP TABLE IF EXISTS Classe ; --puis la table maître
6
7
8
9 CREATE TABLE Classe(
10     id INTEGER PRIMARY KEY,
11     description TEXT
12 );
13
14 CREATE TABLE Herite(
15     enfant INT REFERENCES Classe(id),
16     parent INT REFERENCES Classe(id),
17     PRIMARY KEY (enfant, parent)
18 );
19
20 INSERT INTO Classe VALUES(0, 'chose');
21 INSERT INTO Classe VALUES(1, 'animal');
22 INSERT INTO Classe VALUES(2, 'humain');
23 INSERT INTO Classe VALUES(3, 'enseignant');
24 INSERT INTO Classe VALUES(4, 'chat');
25 INSERT INTO Classe VALUES(5, 'science');
26
27 INSERT INTO Herite VALUES(1, 0);
28 INSERT INTO Herite VALUES(2, 1);
29 INSERT INTO Herite VALUES(3, 2);
30 INSERT INTO Herite VALUES(4, 1);
31 INSERT INTO Herite VALUES(3, 5);
32
```

Une relation sans clef primaire est dénormalisée

→ on s'écarte du modèle relationnel

```
In [24]: 1 %%sql
2
3 INSERT INTO Classe VALUES(0, 'patate');
4
5 --IntegrityError: (sqlite3.IntegrityError) UNIQUE constraint failed: Classe.id
6 [SQL: INSERT INTO Classe VALUES(0, 'patate');]
```

violation de dépendance fonctionnelle

0 → chose  
~~0 → patate~~

mais INSERT INTO Classe VALUES(0, 'chose')  
échouerait aussi ⇒ pas de doublon  
dans le modèle relationnel  
⇒ on manipule des ensembles

