

# BDD1 : Modèle relationnel

## 1 Limites des structures de données plates

Dans un jeu multi-joueurs, chaque joueur possède un pseudonyme, appartient à un peuple et à une alliance. Il dirige par ailleurs un (ou plusieurs) villages, qui ont un nom qu'il choisit ; ces villages possèdent (entre autres choses) une certaine population et ont des coordonnées spatiales.

Ainsi, le joueur "Asterix" joue un Gaulois. Il appartient à l'alliance "Redoutable", dirige le village "Les Fous" situé en (34;-15) qui compte 130 habitants.

Le joueur "Obelix" joue également un Gaulois. Il appartient lui aussi à l'alliance "Redoutable", dirige le village "Baston" situé en (35;-20) qui compte 155 habitants.

Le joueur "Ocatarinettabellatchichix" joue un Corse de l'alliance "Redoutable". Il dirige le village "Corse" situé en (15;15) qui compte 72 habitants et un autre village "Les Fous" situé en (15;16) qui compte 17 habitants.

Enfin le joueur "Cesar" joue un Romain. Il appartient à l'alliance "Rome" et dirige le village "Roam" (-17;15) qui compte 425 habitants.

On souhaite modéliser ces données, afin de les stocker et de pouvoir les traiter depuis une application hébergée sur le web, depuis laquelle des joueurs pourront accéder aux différentes données (et envoyer des ordres pour faire grandir leurs villages).

La première idée, la plus simple en apparence, serait de stocker ces données dans une liste de listes. On peut par exemple créer une liste d'alliances contenant chacune une liste de villages.

Cela donnerait dans notre exemple :

```
Jeu=[
  ["Redoutables",[
    ["Les Fous","Asterix","Gaulois",(34,-15),130],
    ["Baston","Obelix","Gaulois",(35,20),155],
    ["Corse","Ocatarinettabellatchichix","Corse",(15,15),72],
    ["Les Fous","Ocatarinettabellatchichix","Corse",(15,16),17]],
  ["Rome",[
    ["Roam","Cesar","Romain",(-17,15),425]]]
```

*Exercice 1* : Écrire une fonction Python qui renvoie un tableau contenant les différents villages d'une alliance

*Exercice 2* : Écrire une fonction Python qui renvoie un tableau contenant les coordonnées des différents villages d'une alliance.

*Exercice 3* : Écrire une fonction Python qui renvoie un tableau contenant les coordonnées des différents villages contrôlés par des joueurs du peuple Gaulois

Ces deux dernières recherches s'avèrent beaucoup plus compliquées que la première. La complexité du parcours des boucles imbriquées est en  $O(n^2)$  Imaginez alors sa difficulté dans un jeu réel sur des serveurs comportant 100000 joueurs! On peut se demander si la représentation choisie est la bonne.

*Exercice 4* : Quels autres représentations sous forme de liste de listes sont possibles? Quelle recherche deviennent plus compliquées que précédemment?

Dans cette représentation de données, dite "plate", une appartenance est privilégiée par rapport aux autres (l'appartenance à une alliance, à un peuple, à un joueur etc..) Or on peut être amené à effectuer des recherches indépendamment de tout lien. Un nouveau modèle s'avère nécessaire.

## 2 Notion de relation

Dans l'exemple précédent, en faisant abstraction de tout lien d'appartenance, il est possible de représenter les alliances, les villages et les joueurs de la manière suivante :

Alliance : (Nom)

Joueur : (Pseudonyme,Alliance,Peuple)

Village : (Nom,Coordonnées, Joueur, Population)

On appellera une telle représentation une *relation* ou dans le langage des bases de données, une *table*. De manière plus précise :

**définition 1 (attribut, domaine) :**

On considère donné un ensemble (infini)  $\mathcal{A}$  dont les éléments sont appelés *attributs*, un ensemble  $D$  et une application  $\text{dom} : \mathcal{A} \rightarrow \mathcal{P}(D)$  où  $\mathcal{P}(D)$  désigne les sous-ensembles de  $D$ . Si  $A \in \mathcal{A}$ ,  $\text{dom}(A)$  est appelé le *domaine de A*.

*Exercice 5* : Pour la table "Village" quels sont les attributs ? Quels sont leurs domaines ?

*Exercice 6* : Proposer un domaine approprié pour des attributs représentant :

- une date
- un mot de passe
- un classement à un concours
- une note de musique

**définition 2 (schéma relationnel) :**

On appelle schéma relationnel, un n-uplet de la forme  $S=(A_1,\dots,A_n) \in \mathcal{A}^n$  où les  $A_i$  sont deux à deux distincts.  
On le notera parfois  $((A_1,\text{dom}(A_1)),\dots,(A_n,\text{dom}(A_n)))$

*Exercice 7* : Donner le schéma relationnel associé à la table Village avec la dernière notation de la définition.

On notera  $B \in S$  pour dire que  $B$  est un des attributs de  $S$ . Par extension si  $X = B_1, \dots, B_m$  où les  $B_i$  sont des attributs deux à deux distincts, on notera  $X \subset S$  pour dire que  $X$  est un sous-ensemble de  $S$ . En réarrangeant l'ordre des attributs,  $X$  pourra être vu comme un sous  $n$ -uplet de  $S$ .

**définition 3 (relation ou table) :**

Une relation, aussi appelée table, associée à un schéma relationnel  $S = (A_1, \dots, A_n)$  est un ensemble **fini** de  $n$ -uplets de  $\text{dom}(A_1) \times \dots \times \text{dom}(A_n)$   
On note  $R(S)$  la relation  $R$  pour dire qu'elle est associée au schéma relationnel  $S$ .

*Exercice 8 :* Détailler la relation Village.

**définition 4 (enregistrements, cardinal) :**

Les éléments de  $R$  sont appelés les valeurs ou les enregistrements de la relation.  
Leur nombre (fini!) est appelé le cardinal de  $R$  et est noté  $\# R$   
Si  $e \in R(S)$  et  $A \in S$ , on note  $e \cdot A$  la composante du  $n$ -uplet  $e$  associée à l'attribut  $A$

*Exercice 9 :* Soit  $R$  la relation Village. Que vaut  $\# R$  ?

$A$  désigne l'attribut Coordonnées et  $e$  l'enregistrement "Corse". Que vaut  $e \cdot A$  ?

*Exercice 10 :* Proposer un schéma relationnel permettant de représenter les différents éléments chimiques avec des informations équivalentes à celles fournies par la table de Mendeleïev. Donner un exemple significatif de relation associée au schéma proposé.