



I - Principe des jointures

Le principe est de créer de nouveaux n-uplets en récupérant les attributs de plusieurs autres n-uplets, issus la plupart du temps de tables différentes. On fait la jonction entre 2 n-uplets en faisant correspondre **une valeur de clé étrangère et une valeur de clé primaire**. Exemple :

table jeu			
<u>id</u>	jeu	type	#id_support
9	Pitfall	Plate-forme	2

table console			
<u>id</u>	nom	RAM	date sorite
2	Atari 2600	4 ko	1977-10-14

On peut créer un nouvel n-uplet en joignant ces deux n-uplets puisque **jeu.id = console.id** :

jeu.id	jeu	type	id_support	console.id	nom	RAM	date sorite
9	Pitfall	Plate-forme	2	2	Atari 2600	4 ko	1977-10-14

II - Ecriture SQL d'une jointure

2.1 Exemple avec des attributs ayant des noms différents : notation pointée optionnelle

Imaginons :

- une relation **livre** ayant une clé étrangère **id_auteur** et un attribut **titre**. **livre (id1, titre, #id_auteur)**
- une relation **auteur** ayant une clé primaire **id** et un attribut **nom**. **auteur (id2, nom)**

Ici, aucun nom en commun, on pourra écrire ceci (l'ordre primaire-étrangère ou étrangère-primaire n'est pas important :

```
SELECT titre, nom
FROM livre
JOIN auteur ON id2 = id_auteur
```

```
SELECT titre, nom
FROM livre
JOIN auteur ON id_auteur = id2
```

2.2 Exemple avec des attributs ayant des noms similaires : notation pointée obligatoire

Imaginons maintenant que les attributs **id** et **nom** soit un attribut dans les deux tables :

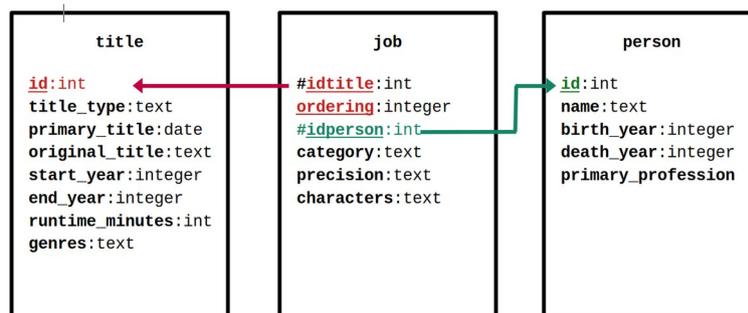
- une relation **livre** ayant une clé étrangère **id_auteur** et un attribut **nom**. **livre (id, nom, #id_auteur)**
- une relation **auteur** ayant une clé primaire **id** et un attribut **nom**. **auteur (id, nom)**

Ici, puisque id et nom sont ambiguës, il faut nécessairement utiliser la notation pointée :

```
SELECT livre.nom, auteur.nom
FROM livre
JOIN auteur ON auteur.id = livre.id_auteur
```

- sans cette notation, difficile de savoir quel nom

2.3 Double jonction (ou triple ou plus encore...)



```
SELECT title.primary_title, person.name, job.category
FROM job
JOIN person ON job.idperson = person.id
JOIN title ON job.idtitle = title.id
```

REMARQUE : L'ordre job puis jointure avec person puis title n'est pas obligatoire. On peut commencer par title puis joindre job puis finalement title.

CONCLUSION : Dès que vous avez une jointure, utilisez automatiquement la notation pointée, cela vous évitera l'erreur d'étourderie typique suivante : ne pas remarquer qu'un même nom d'attribut est présent dans plusieurs tables.