

## Bases de données

### *Les limites des tables de données*

Comme nous l'avons vu l'année dernière, il est possible de stocker des données dans des tables afin de les exploiter par la suite. Voici un extrait d'une table avec quelques films.

| Titre                 | Réalisateur     | Année | Durée |
|-----------------------|-----------------|-------|-------|
| 2001: A Space Odyssey | Stanley Kubrick | 1968  | 149   |
| Alien                 | Ridley Scott    | 1979  | 117   |
| Blade Runner          | Ridley Scott    | 1982  | 117   |
| Akira                 | Katsuhiro Ôtomo | 1988  | 124   |

Ce modèle pose des problèmes si on souhaite enrichir la base de données. Tout d'abord, si on veut rajouter des informations sur le réalisateur, comme sa nationalité, il faut le faire pour chacun de ses films. Si un réalisateur change de nom, ou si on souhaite rajouter un deuxième prénom pour le distinguer d'un homonyme, il faut le faire également pour chacun de ses films. Comment faire s'il y a un deuxième réalisateur? Faut-il faire une nouvelle entrée? Ou faut-il rajouter son nom au premier dans la colonne "Réalisateur"? Mais alors, comment trouver facilement tous les films d'un même réalisateur?

Il faut également faire attention à ce que pour tous les films d'un même réalisateur, les données sont identiques. Si on se trompe dans son nom ou si on ne met pas la même nationalité, on a un risque d'inconsistance.

Tous ces problèmes sont démultipliés si on veut rajouter les scénaristes et les acteurs.

Pour apporter une solution, on sépare les données en plusieurs tables qui ont un lien les unes avec les autres. C'est ce qu'on appelle une **base de données**.

### *Modèle relationnel*

Le **modèle relationnel**, créé par E. F. Codd en 1970, consiste à voir chaque **table** comme une **relation** sur les données qui la constituent. On va donc définir une relation pour les films et une autre pour les réalisateurs.

| Relation Films |                       |      |       |       | Relation Réalisateurs |                 |
|----------------|-----------------------|------|-------|-------|-----------------------|-----------------|
| film           | titre                 | réal | année | durée | réal                  | nom             |
| 1              | 2001: A Space Odyssey | 1    | 1968  | 149   | 1                     | Stanley Kubrick |
| 2              | Alien                 | 2    | 1979  | 117   | 2                     | Ridley Scott    |
| 3              | Blade Runner          | 2    | 1982  | 117   | 3                     | Katsuhiro Ôtomo |
| 4              | Akira                 | 3    | 1988  | 124   |                       |                 |

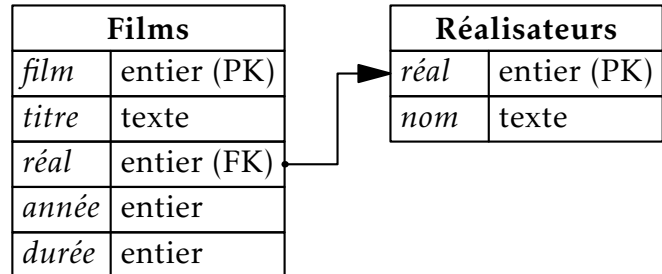
Afin d'identifier de façon unique chaque réalisateur, on lui ajoute un numéro unique, qu'on appelle **clef primaire**. Chaque ligne dans une relation s'appelle un **t-uplet** et elle doit être unique. C'est ce qu'on appelle une **contrainte d'entité**. Chaque colonne s'appelle un **attribut**. Même s'il semble peu probable que deux réalisateurs aient le même nom, la même année de naissance et la même nationalité, il est courant d'ajouter une clef primaire afin de ne jamais avoir de problème à ce niveau.

De la même manière, on rajoute une clef primaire pour les films. En effet, il est possible que deux films aient le même nom, comme dans le cas pour un remake. Pour indiquer le réalisateur d'un film, il suffit alors de rajouter une colonne avec la clef primaire correspondant. On parle alors de **clef étrangère**, puisque c'est la clef primaire d'une autre relation. C'est une **contrainte de référence**, qui permet de s'assurer que tous les réalisateurs dans la table des films se trouvent aussi dans la table des réalisateurs.

Le modèle relationnel permet également de définir le type des données, ce qui donne les **contraintes de domaine**. Dans notre cas, les données sont soit des entiers, soit des textes. On appelle **schéma** la définition des attributs et types de chaque relation. Voici les deux schémas de notre base :

- **Film**(film : entier, titre : texte, réal : entier, année : entier)
- **Réalisateur**(réal : entier, Nom : texte)

Les clefs primaires sont soulignées et les clefs étrangères sont soulignées en pointillés. On peut aussi représenter les schémas comme ci-contre. Les clefs primaires sont notées "PK" (Primary Key) et les clefs étrangères "FK" (Foreign Key).



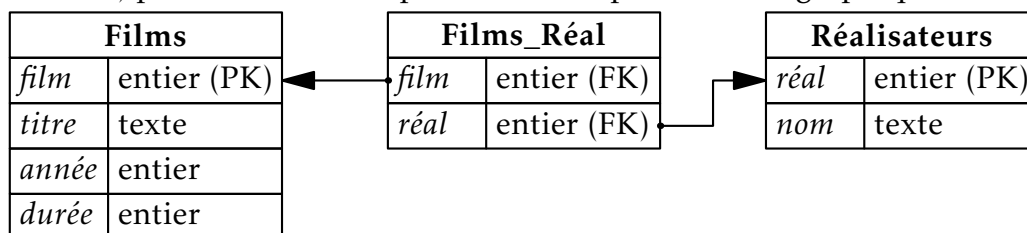
Les contraintes de domaines peuvent restreindre encore plus les valeurs possibles. Ainsi, le titre du film ne peut pas être vide. Les années aussi doivent être des entiers de supérieurs à 1800. Toutes ces contraintes forment les **contraintes d'intégrité** qui assurent qu'à chaque instant, les données de la base sont cohérentes.

### Vers une base plus complexe

Dans notre exemple, il n'est pas possible de rajouter un deuxième réalisateur pour un film, puisqu'il faudrait rajouter une nouvelle ligne pour le film. Afin de résoudre ce problème, on crée une nouvelle relation qui fera le lien entre les réalisateurs et les films. Les nouveaux schémas sont :

- **Film**(film : entier, titre : texte, année : entier)
- **Réalisateur**(réal : entier, Nom : texte)
- **Film\_Réal**(film : entier, réal : entier)

Puisqu'il n'est pas possible d'avoir deux fois le même réalisateur pour le même film, le couple (titre, réal) peut servir de clef primaire. La représentation graphique est la suivante :



Ce qui donne ces tables :

| film | titre                 | année | durée | titre | réal | réal | nom             |
|------|-----------------------|-------|-------|-------|------|------|-----------------|
| 1    | 2001: A Space Odyssey | 1968  | 149   | 1     | 1    | 1    | Stanley Kubrick |
| 2    | Alien                 | 1979  | 117   | 2     | 2    | 2    | Ridley Scott    |
| 3    | Blade Runner          | 1982  | 117   | 3     | 2    | 3    | Katsuhiro Ôtomo |
| 4    | Akira                 | 1988  | 124   | 4     | 3    |      |                 |

### Système de gestion de base de données

Dans la pratique, les bases de données sont souvent stockées sur un serveur. On y accède via un **système de gestion de base de données**. Le but de ce système est de définir les droits des différents utilisateurs (lecture, modification de certaines tables), de gérer les accès en parallèle, de sécuriser les données et d'assurer leur sauvegardes.