

CARDINALITES

Comprendre les relations et les
cardinalités

1. Définition d'une relation
2. Stockage d'une relation
 1. Stockage des relations n-n
 2. Stockage des relations 1-n

Définition d'une relation

Dans un enregistrement, il n'est pas possible de mettre une liste.

Chaque valeur dans un enregistrement doit être une valeur unitaire, soit un string, soit un int, soit un float, mais pas un array !

Cas impossible

Titre : Gateau chocolat

Prix : 18

Ingrédients : farine, œufs,
chocolat

Cas impossible

Titre : Gâteau chocolat

Prix : 18

Ingrédients : farine, œufs,
chocolat

Même si on est bien tenté d'introduire une liste dans une valeur d'un enregistrement, en informatique on ne sera pas capable de stocker cette information

Pas de liste (array) dans une valeur
d'un enregistrement !

Comment peut-on faire pour
stocker les ingrédients liés à ce
gâteau ?



On crée des relations !

Pour stocker les ingrédients d'un gâteau, il faut d'abord stocker :

- les gâteaux
- les ingrédients
- la relation entre les gâteaux et les ingrédients.

Ainsi, combien de choses est-il
nécessaire de stocker ?



Il faut en stocker 3 ! Nous devons stocker les 2 entités qui sont en relation, c'est-à-dire le gâteau et le ingrédients mais également la relation elle-même qui permet l'association des 2

On stocke le Gateau

id	...
Titre	...
Prix	...

Gateau

On stocke le Ingrédient

id	...
Titre	...
Prix	...

Gateau

id	...
Nom	...

Ingrédient

Comment stocke-t-on la relation
entre les deux ?



1. Nommer la relation à stocker
2. Déterminer la cardinalité
3. Déterminer l'implémentation

Stockage d'une relation

Pour stocker la relation, il faut d'abord la **nommer**.

Habituellement un **verbe** est utiliser pour nommer une relation.

Example #1

Dans le cas d'ingrédients et de Gateau, la relation à stocker est « avoir », car un gateau peut avoir des ingrédients

Nommez une relation entre
« Hotel » et « Chambre » ?



Nommez une relation entre
« Vendeur » et « Boutique » ?



Nommez une relation entre
« Compte bancaire » et
« Client » ?



Nommez une relation entre
« Touriste » et « Monument » ?



Nommez une relation entre
« Artiste » et « Musique » ?



Nommez une relation entre
« Producteur » et « Film » ?



Nommez une relation entre
« Message » et « Expéditeur » ?



1. Nommer la relation à stocker
2. Déterminer la cardinalité
3. Déterminer l'implémentation

La **forme du stockage** de la relation entre 2 structures comme la relation « avoir » pour Gateau et Ingrédient dépend de la **cardinalité** :

- **1-1** : création de 2 nouvelles colonnes
- **1-n** : création de 1 nouvelle colonne
- **n-n** : création de 1 nouvelle structure

Qu'est-ce que cela veut dire ?
Une **cardinalité**, c'est quoi ?

Dans la modélisation de données, une **cardinalité** ou **multiplicité** sert à définir le **nombre d'associations possibles** entre 2 entités

Pour déterminer la cardinalité d'une relation R entre $T1$ et $T2$, il est nécessaire de se poser 2 questions :

- Un $T1$ peut R combien de $T2$?
- Un $T2$ peut être R par combien $T1$?

Pour déterminer la cardinalité d'une relation « avoir » entre « Gateau » et « Ingrédient », il est nécessaire de se poser 2 questions :

- Un « Gateau » peut « avoir » combien de « Ingrédient » ? n
- Un « Ingrédient » peut être « eu » par combien de « Gateau » ? h

Répondons à ces questions :
choix possibles sont 1 ou
plusieurs



Pour déterminer la cardinalité d'une relation « avoir » entre « Gateau » et « Ingrédient », il est nécessaire de se poser 2 questions :

- Un « Gateau » peut « avoir » combien de « Ingrédient » ? **Plusieurs ingrédients**
- Un « Ingrédient » peut être « eu » par combien de « Gateau » ? **Plusieurs gateaux**

Comme on a répondu Plusieurs à chaque question, on dit que la relation est **n-n** :

- Stocker la relation « avoir » entre « Gateau » et « Ingrédient » nécessitera de **créer une nouvelle structure**

1. Nommer la relation à stocker ✓
2. Déterminer la cardinalité ✓
3. Déterminer l'implémentation

A quoi va ressembler cette
nouvelle structure ?



On stocke la relation
« avoir »

id	...
Titre	...
Prix	...

Gateau

gateau_id	135
ingrédient_id	42

Avoir

id	...
Nom	...

Ingrédient

Convention de nommage

Habituellement, il est recommandé d'écrire tout en minuscule et en snake-case.
L'utilisation d'accent n'est pas recommandé.
Le nom des 2 entités apparaissent dans le nom de la table de liens

Avec les conventions

id	...
titre	...
prix	...

gateau

gateau_id	...
ingredient_id	...

avoir_gateau_ingredient

id	...
nom	...

Ingredient

Exemple #2

Nommez une relation entre
« Vendeur » et « Boutique » ?



Stockons la relation « **travailler** », comme dans
« un vendeur travaille dans une boutique ».

Nous aimerions par exemple stocker cette
information pour savoir quel vendeur travaille
dans quelle boutique.

Veillez écrire les 2 questions
qu'il faut se poser



- Pour déterminer la cardinalité d'une relation « travailler » entre « Vendeur » et « Boutique », il est nécessaire de se poser 2 questions :
- Un « Vendeur » peut « travailler » dans combien de « Boutique » ?
 - Une « Boutique » peut être « travaillé/le lieu de travail » par combien de « Vendeur » ?

Par exemple, une bonne réponse peut être :

- Un « **Vendeur** » peut « **travailler** » dans combien de « **Boutique** » ? **Une seule boutique**
- Une « **Boutique** » peut être « **le lieu de travail** » pour combien de « **Vendeur** » ? **Plusieurs vendeurs**

Nous reprenons les réponses pour déterminer la cardinalité de « travailler » :

- Une seule boutique
- Plusieurs vendeurs

Nous avons donc une relation de cardinalité :
1 boutique – n vendeurs,
dit aussi 1 - n

Pour stocker les relations 1-n, il faut créer une nouvelle colonne.

Pour stocker la relation **1 boutique – n vendeurs**,
on va **créer une nouvelle colonne à la patte n**,
donc vendeur :

id	... 135 PK
matricule	...
prenom	...
travail_boutique_id	... 46 FK

vendeur

id	...
nom	...

boutique



Cette nouvelle colonne utilisée pour stocker une relation s'appelle une **FOREIGN KEY** (clé étrangère). Elle permet d'identifier l'autre partie de la relation.

Les relations 1 – 1 ne sont que très peu utilisées (notamment pour des raisons de performances).

Si vous tombez Sur une relation 1 - 1 posez vous des questions, car ce n'est peut être pas logique

Vous avez donc 2 cardinalités majoritaires :

- 1 – n, appelé aussi 1 - *, appelé aussi OneToMany
- n – n, appelé aussi * - *, appelé aussi ManyToMany



