

# Le dictionnaire de données

## Définition

Un dictionnaire de données va représenter l'ensemble des données manipulées par le système. Il permet à l'équipe de développement d'avoir accès à la même information. Pour une base de données c'est une collection de données qui sert à la conception d'une base de données relationnelle. Le dictionnaire de données est obtenu suite à l'étude des besoins du client.

Il est intéressant de dresser la liste des données que le système va utiliser dans un tableau. Pour chaque donnée il faudra la décrire selon :

- Le nom : le nom de la donnée
- La description, utile lorsque le nom de la donnée n'est pas significatif (identifiants)
- Le type de données : selon le SGBD ou général
- Contraintes

## Exemple de dictionnaire de données:

Données	Description	Types	Contraintes	Commentaire
idClient	Désigne le numéro du client	Integer	Unique et not null	Séquentiel (aut-increment)
nomClient	Désigne le nom du client	Varchar (40)	Obligatoire (not null)	
AdresseClient	Désigne l'adresse du client	Varchar (40)	Obligatoire	Données composées de no de rue, ville et le code postale
noComde	Désigne le numéro de commande	Integer	Unique et non null	Séquentiel (Auto-Increment)
DateCdme	Représente la date de la commande	Date	Obligatoire	
idArticle	Désigne le numéro d'un article	Integer	Obligatoire. Non null	
PrixUnitaire	Désigne le prix unitaire du produit	Monétaire		
Quantite	Quantité commandée pour chaque article dans une commande	Integer	Obligatoire	
Total	Désigne le total pour un produit acheté dans une commande	Monétaire	Donnée calculée. Voir la règle R1	
Montant	Désigne le montant total de la commande	Monétaire	Donnée calculée. Voir la règle R2	

## Règles de gestion et ou de calcul.

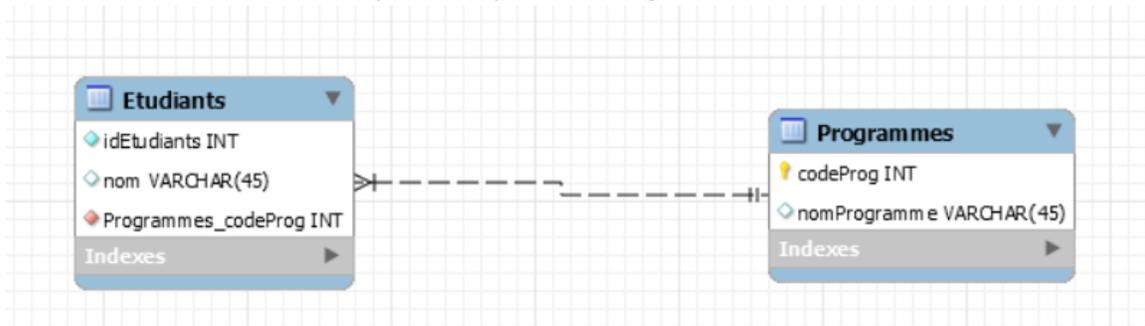
Les règles de gestion permettent de définir les liens entre les différentes tables de votre base de données relationnelle.

Exemple, les règles R10 et R11 donne un modèle de base de données différent du modèle qui applique les règles R10 et R12

R10 : Dans un programme nous avons plusieurs étudiants.

R11 : Un étudiant s'inscrit **à un et un seul programme.**

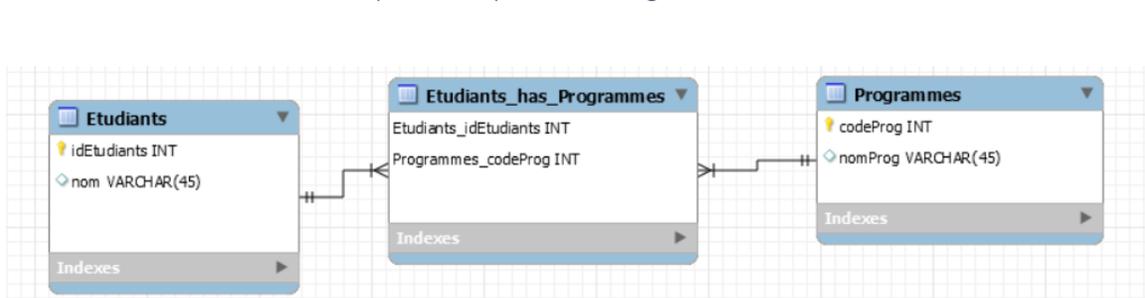
Voici le modèle de données qui correspond aux règles R10 et R11



R10 : Dans un programme nous avons plusieurs étudiants.

R12 : Un étudiant s'inscrit **à un ou plusieurs programmes.**

Voici le modèle de données qui correspond aux règles R10 et R12



## Exemples de règles de gestion :

R1 : un client peut faire plusieurs commandes

R2 : une commande concerne un et un seul client

R3 : Un article peut se retrouver sur plusieurs commandes

R4 : une commande peut contenir plusieurs commandes.

R5 : les articles sont répartis par catégories pour fin de recherches

R6 : le prix de l'article dépend uniquement de l'article.

R7 : le prix du billet dépend aussi de la section de la salle (pas uniquement du spectacle)

## Exemple de Règles de calcul :

R1 : sur une commande le total est calculé comme suit :  $\text{Total} = \text{Quantite} * \text{prixUnitaire}$ . (Pour chaque produit)

R2 : le montant de la commande est calculé comme suit :  $\text{Montant} = \text{somme}(\text{Total}) + \text{les taxes applicables}$ .

## Rappel : normalisation

Après la normalisation, il faudra vérifier :

- Tous les attributs **non calculés** sont présents dans des tables.
- Aucun attribut calculé n'est présent dans une table
- Aucun attribut n'est redondant, sauf la clé primaire qui devient clé étrangère dans d'autres tables.
- **Toutes les tables ont une clé primaire.**
- Tous les attributs sont élémentaires (non décomposable ou se traitent comme un tout, exemple l'attribut adresse peut-être décomposable).
- Toutes les tables sont au moins en 3FN.

La normalisation a pour buts de garantir :

- La non redondance des données → 1FN
- L'intégrité des données → 2FN, 3FN
- La facilité de mise à jour → 3FN

**Une base de données est normalisée si et seulement si elle est au moins en 3FN**

### Énoncé 1 : La 1FN : (La clé)

Les tables ne doivent pas contenir des groupes de données répétitives. Si tel est le cas, il faut sortir le groupe de données et créer une autre table qui va contenir ce groupe de données.

**Énoncé 2 : La 2FN (Toute la clé)**

Une table est en deuxième forme normale 2FN, si elle est :

- a. En 1FN
- b. Tous les attributs de la relation ou de l'entité dépendent de **toute la clé** (identifiant) et non d'une partie de la clé

**Énoncé 3 : La 3FN (Rien que la clé)**

Une table est en troisième forme normale (3FN) si

- a. Elle est déjà en 2FN
- b. Tous les attributs non clé dépendent uniquement de la clé. Il n'y a pas de dépendance entre deux attributs non clé.

## Validation d'une base de données

Une fois que votre modèle de données est produit, que vous êtes certain que votre base de données est au moins en 3 FN, il faudra s'assurer qu'elle répond bien aux besoins du système. Alors.. Comment valider une base de données ? Comment être certain que l'on

1. S'assurer que toutes les données non calculées du dictionnaire de données sont dans des tables ou de relations
2. S'assurer qu'il existe en tout temps un moyen de calculer les données calculées
3. S'assurer que les données à saisir, ou à mettre à jour soient dans des tables
4. S'assurer que les consultations (SELECT) se font bel et bien. Pour cette partie, on peut procéder de différentes façons. L'approche suggérée est de faire une validation pour les requêtes les plus complexes. Ce qui revient à dire, que nous allons écrire les requêtes les plus complexes (même si la BD n'est pas créée physiquement) pour vérifier que le modèle de données répond bien.

Attention : 

Une BD normalisée n'est pas nécessairement validée. Après la normalisation, il faudra vérifier que votre BD répond.

Exemple voici un modèle de BD normalisé.

Dans l'exemple suivant on peut avoir les informations suivantes :

1. La liste des cours enseignés par un prof donnée (Exemple Alexandre)
2. La liste des profs qui enseignent un cours donné.
3. La liste des étudiants ayant suivi ou qui suivent un cours donné (exemple KB6)
4. La liste des cours suivis par étudiant.

**MAIS** Avec le modèle suivant, vous ne pouvez pas obtenir ceci :  
 Liste des étudiants d'un enseignant donné dans un cours donné.

**Vous ne pouvez pas connaître la liste des étudiants d'Alexandre dans le cours de KBE. Le modèle ne répond pas à cette requête.**

