heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

BDR

Laboratoire N°1

11 mars 2010

Réalisation par :

Arnaud Burkhalter

Steve Lienhard

Destiné à :

Nastaran Fatemi: Professeur BDR

Sébastien Noir: Ingénieur assistant

Table des matières

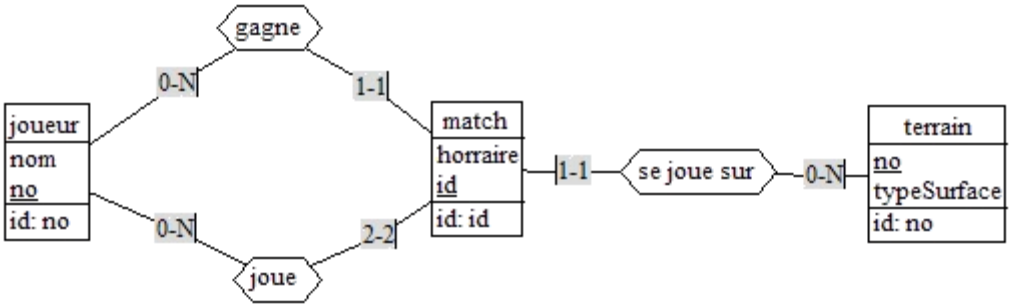
1. Intro

Le but de ce premier laboratoire du cours sur les Bases de Données Relationnelles , est de bien comprendre le concept de modélisation de base de données, et de prendre en main le programme DB-MAIN.
Afin de d'atteindre cet objectif, il nous est demandé d'effectuer différents exercices, tous simulant une situation différente et nécessitant la réalisation d'un schéma Entité-Association.

2. Réalisation

Les explications et justifications des schémas concernant les différents exercices ne sont pas exhaustives et ne prennent en compte que les parties pouvant être ambigües ou pouvant varier en fonction de l'interprétation que l'on s'est fait de l'énoncé du problème.

2.1 Exercice 1



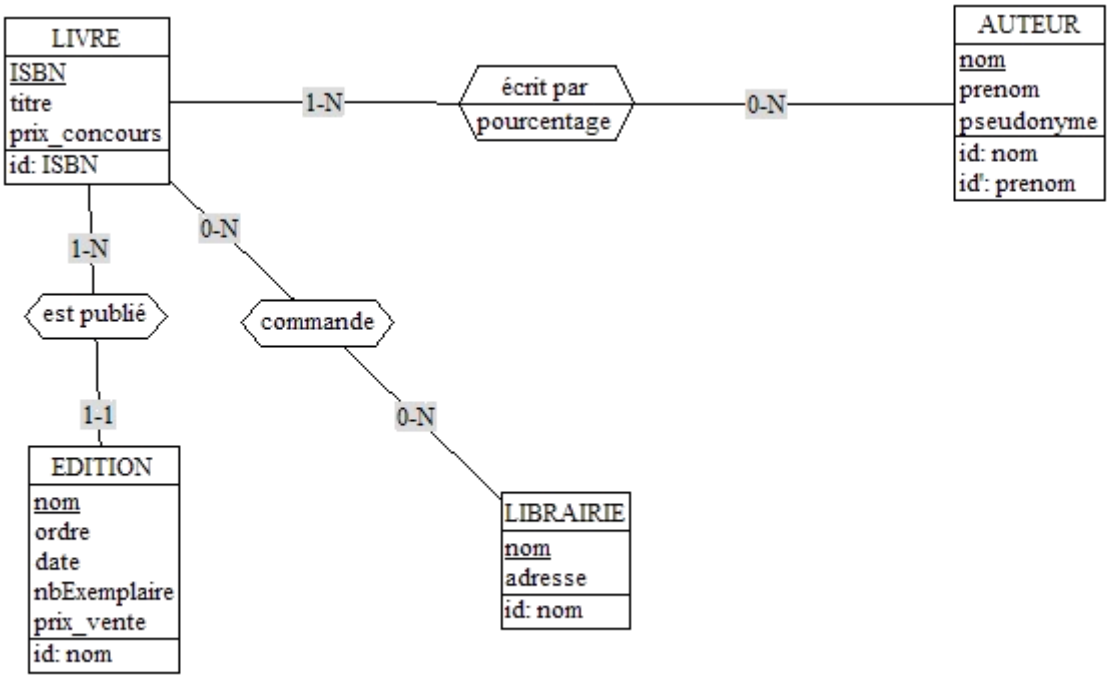
Peut-on jouer des matchs en double ?
Non, car la relation entre l'entité 'match' et l'entité 'joueur' nous indique qu'un match se joue entre 2 et uniquement 2 joueurs. A comprendre par là, un de chaque côté du terrain.

Un joueur peut-il gagner un match sans même y avoir participé ?
Oui, selon la base de données cela est possible. Si l'on veut éviter cela, il faudra ajouter des contraintes d'intégrités lors de la réalisation de celle-ci.

Peut-il y avoir deux matchs sur le même terrain à la même heure ?
Oui. La logique voudrait que cela ne soit pas possible. Cependant c'est possible selon la base de données car il n'y a aucune contrainte entre les entités terrain et match qui nous indique qu'il ne peut pas y avoir deux matchs joués en même temps sur un terrain.

Connaissant un joueur, peut-on savoir sur quels terrains il a joué ?
Oui, de manière indirecte, le joueur connait les matchs qu'il a joué, et à travers ceux ci, nous pouvons donc retrouver sur quels terrains ont été joué les matchs en question.

2.2 Exercice 2

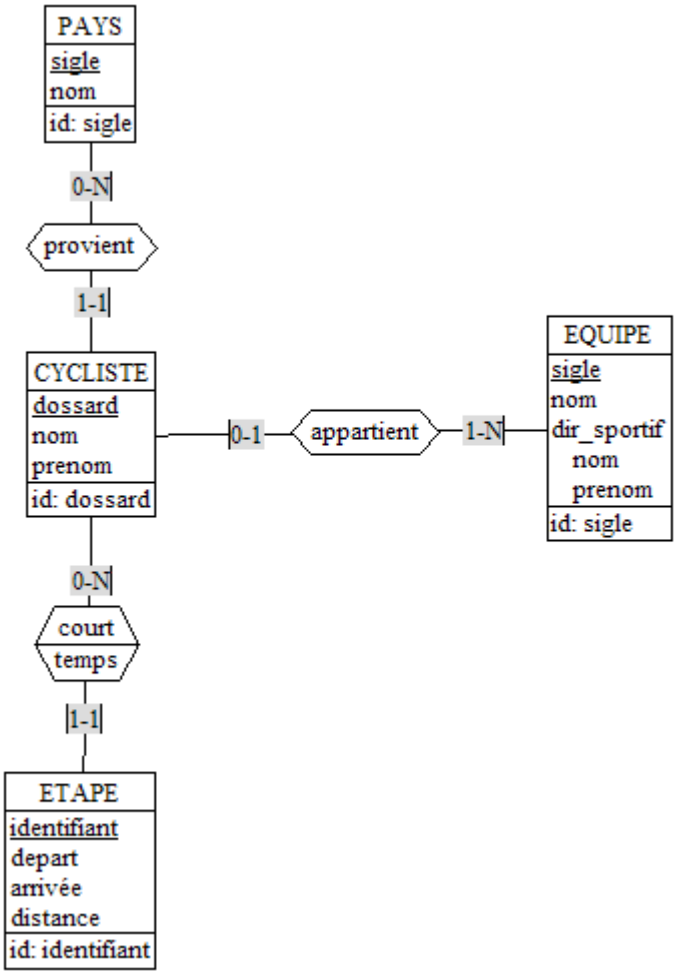


Le schéma ci-dessus représente la modélisation que nous avons mise en place pour cet exercice de gestion de livres.

Concernant l'entité EDITION, nous avons pris la décision d'utiliser l'attribut *nom* comme clé, car les éditions portant toutes des noms différents, nous n'avons pas jugé nécessaire d'ajouter une clé à valeur numérique.

Afin de gérer les droits d'auteurs que touche chaque auteur, nous avons ajouté un champ *pourcentage* dans l'association *écrit par*, reliant les entités LIVRE et AUTEUR, car un auteur peut avoir plusieurs pourcentages, qui peuvent varier selon différents critères, tel que le nombre de personnes ayant travaillé sur l'écriture du livre. De plus il n'aurait pas été judicieux d'inclure ce champ dans la table LIVRE car pour un livre, les pourcentages ne seront pas forcément identiques entre les différents auteurs ayant travaillé sur celui-ci.
Le fait d'avoir ajouté cette information dans l'association, nous contraindra à ajouter une table supplémentaire dans notre base de données.

2.3 Exercice 3

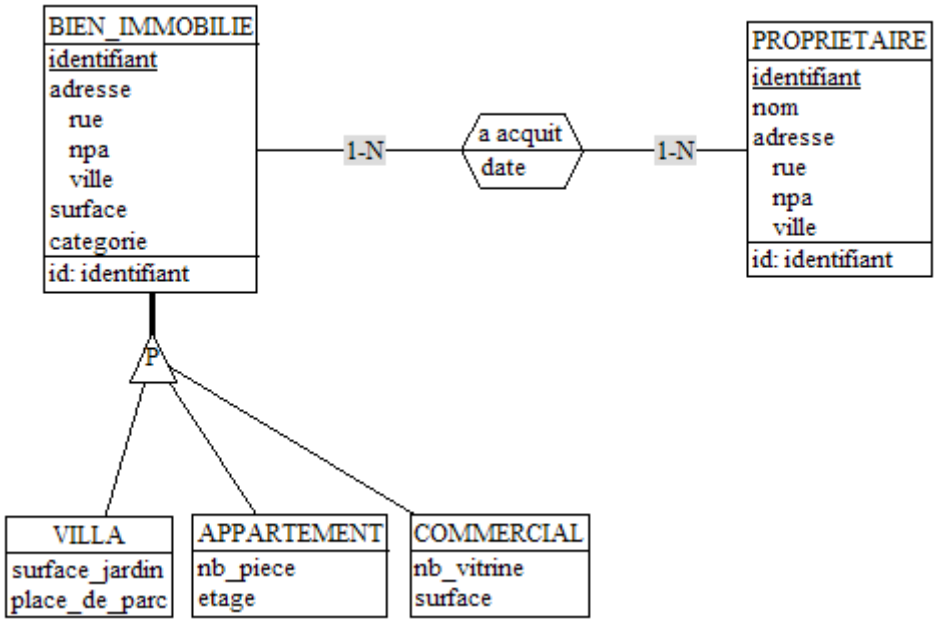


Le type des attributs *sigle* présent dans les entités PAYS et EQUIPE sont de type varchar, du fait que nous avons imaginé que ces champs peuvent être comparés à des plaque numérologiques et contenir des valeurs numériques autant que des lettres.
Toujours concernant les types utilisés, nous avons pris la décision de représenter la distance de chaque étape au format float, du fait que cette valeur n'est pas forcément une valeur entière.

Afin de mettre en place l'hypothèse que le pays de provenance des coureurs doit obligatoirement être renseigné, nous avons contraint cela en indiquant une cardinalité 1-1 entre CYCLISTE et PAYS, se qui contraint un cycliste à être en relation avec une instance de la classe PAYS.

L'association *court* contient l'attribut *temps*, qui représente le temps de parcours du cycliste X pour arriver à bout de l'étape Y.

2.4 Exercice 4



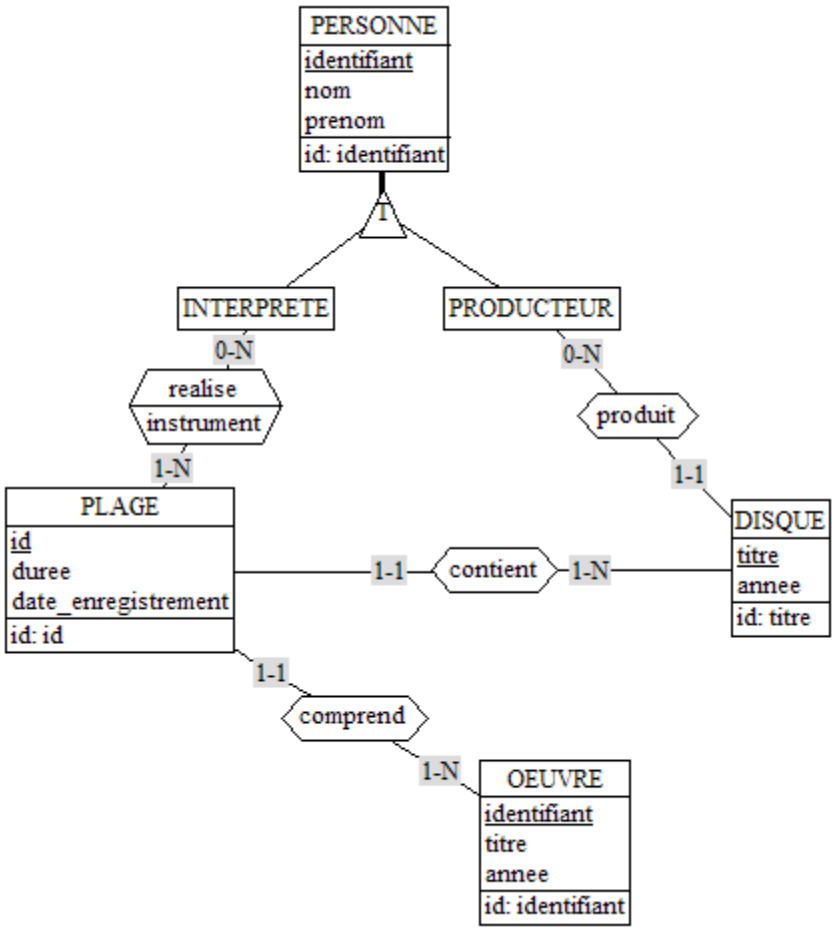
La principale problématique de cette base de données, vient du fait que nous du faire des sous-classes héritant de la classe BIEN_IMMOBILIE. Celles-ci sont disjointes, car un bien immobilier ne peut faire parti que de l'une des trois catégories, ainsi que totales, étant donné que tout bien immobilier doit faire partie de l'une des trois sous-classes. Ces deux contraintes sont visibles à l'aide de la lettre P, se trouvant au centre de la relation reliant la superclasse à ses différentes sous-classes.

L'attribut *categorie* présent dans l'entité BIEN_IMMOBILIER, permet de définir une catégorie pour chaque bien immobilier (V = villa, C=local commercial, A = appartement), et a été défini à l'aide d'un char de longueur 1 du fait qu'une seul lettre sera nécessaire pour représenter ce champ.

De plus, concernant l'attribut *adresse* nous avons pris la décision que le champ *rue* contiendrait le nom de la rue, ainsi que le numéro de rue, raison pour laquelle nous n'avons pas utilisé d'attribut pour représenter celui-ci.

La contrainte qui veut que tout propriétaire se doit de posséder au moins un bien immobilier a été implémenté grâce aux cardinalités présentes entre les entités BIEN_IMMOBILIER et PROPRIETAIRE. Pour ce faire, nous avons indiqué qu'un propriétaire se devait d'avoir 1 à N bien, se qui lui empêche de n'avoir aucun bien en sa possession.

2.5 Exercice 5

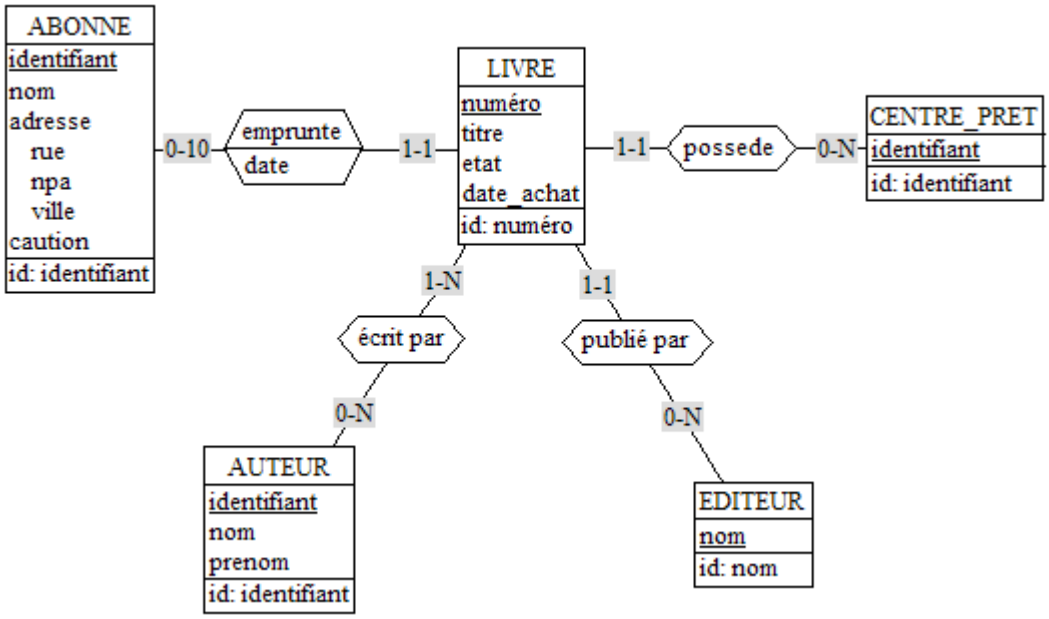


Notre schéma contient deux sous-entités, héritant de PERSONNE. Cela a été réalisé de cette manière, car INTERPRETE et PRODUCTEUR ont tous deux les même attributs et informations à stocker. De plus, la spécialisation est totale, car toutes les entités de la superclasse doivent être des producteurs ou interprètes.

Concernant l'entité PLAGE, nous avons ajouté l'attribut *id* qui fait office de clé unique, car aucun des autres champs présents ne pouvais jouer ce rôle.
De plus, nous avons supposé qu'une plage pouvait être réalisée par plusieurs interprètes, et qu'un interprète peut n'avoir encore réalisé aucune plage, d'où la relation 1-N et 0-N, entre les entités PLAGE et INTERPRETE.

Pour répondre à la question, posée en fin d'exercice 5, concernant l'attribut *Instrument*, nous avons ajouté celui-ci dans la relation entre INTERPRETE et PLAGE, se qui permet à un interprète d'avoir un instrument différent pour chaque plage réalisée.

2.6 Exercice 6

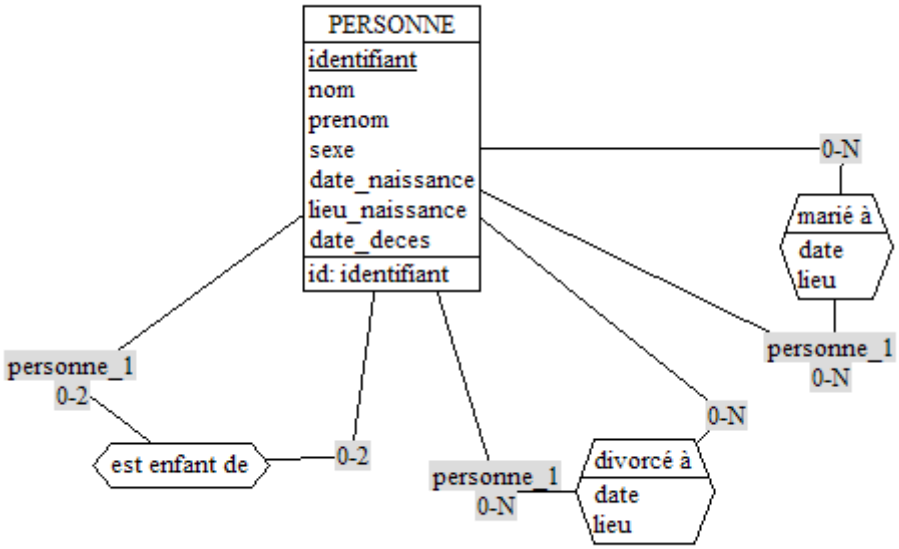


Pour ce schéma, nous avons fait l'hypothèse qu'un centre de prêt peut ne posséder aucun livre, dans le cas où tous les livres seraient empruntés en même temps.

Pour cet exercice, passablement de contraintes d'intégrité sémantique seront à implémenter par la suite. Premièrement nous avons limité l'emprunt de livre à 10 par abonné, cependant nous n'avons pas les moyens de décider combien il pourra en louer en fonction de la caution qu'il a versé. Cela sera donc à faire par la suite. La liste des emprunteurs en retard sera aussi à réaliser par la suite.

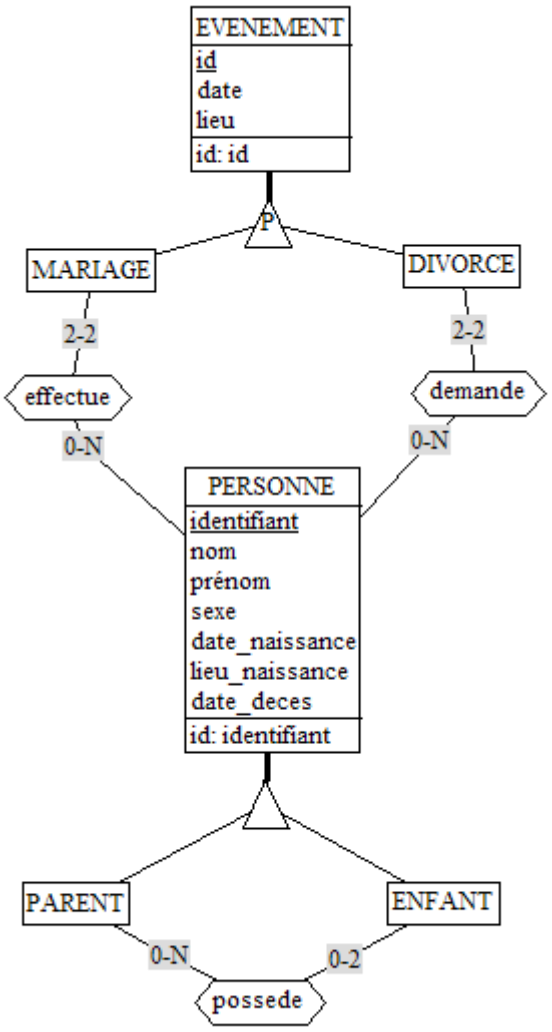
2.7 Exercice 7

Sans lien de généralisation



Pour ce premier schéma, qui a été réalisé sans avoir recours au lien de généralisation, nous avons mis en place trois relations.
La première permet de mettre une personne en relation avec ses parents. Pour se faire, nous avons fais l'hypothèse que toute personne peut avoir de 0 à 2 parents (0 dans le cas d'un enfant abandonné, 1 si l'enfant est adopté ou de père inconnu et 2 pour toute autre situation).
Les deux relations suivantes permettent de représenter les mariages et divorces d'une personne. Pour chacun de ces deux événements, nous avons les cardinalités 0-N car toute personne peut être mariée ou divorcé de zéro à plusieurs fois.
Toutefois, une fois la base de données implémentée, il faudra ajouter des contraintes d'intégrité sémantique pour contrôler qu'une personne ne peut pas être mariée à plusieurs personnes simultanément, de même qu'une personne ne peut être divorcé que de personnes avec lesquelles elle a été mariée au préalable.

Avec lien de généralisation



Pour cette deuxième partie de l'exercice 7, nous avons mis en place le principe de généralisation. Pour commencer nous avons divisé les personnes en deux sous-entités, PARENT et ENFANT. La relation que ces deux entités ont avec la super-entité PERSONNE n'est pas disjointe, du fait que chaque personne peut être parent d'une autre personne, tout en étant l'enfant de quelqu'un d'autre. Cette relation n'est pas non plus totale, nous avons fait l'hypothèse qu'une personne peut n'avoir aucun parent, donc n'être l'enfant de personne et n'avoir aucun enfant.

Dans une seconde phase, nous avons décidé de regrouper les mariages et divorces au sein d'une super-entité mariage, ce qui nous permet d'éviter les attributs à doublon. Cependant, nous avons défini les relations comme étant totales et disjointes, étant donné que tout événement se doit d'être un mariage et un divorce et qu'un événement ne peut pas être mariage et divorce à la fois.

3. Conclusion

Arrivé en fin de ce premier laboratoire, nous pouvons faire ressortir plusieurs points qui nous semblent passablement important lors de la modélisation d'une base de données.
Premièrement, lire attentivement la donnée et se tenir aux informations que celle-ci contient, car nous avons parfois trop tendance à vouloir ajouter des champs qui nous semblent importants pour la base de données, malgré que ceux-ci ne soient pas explicitement mentionnés dans la consigne. Deuxièmement, il est indispensable de bien faire attention aux cardinalités entre les entités, car celles-ci seront particulièrement importantes dans le produit final. Finalement, penser à factoriser les entités au maximum. On entend par là, regrouper les entités au sein de classes et sous-classes lorsque cela est possible, afin d'éviter les informations en doublon.