

La conception des bases de données avec UML

Troisième partie

Le modèle logique de données

Le M.C.D. permet de recenser et de représenter les interactions entre les diverses informations du domaine d'étude.

Jusqu'à maintenant, nous n'avons pas précisé quelles étaient les caractéristiques des divers champs des entités, et ce alors que le menu de création des champs nous le permettait.

Nous avons simplement défini l'identifiant de l'entité.

Pour définir le modèle logique de données (M.L.D.), il nous faudra préciser les caractéristiques des champs.

Nous traiterons plus en détail cette problématique.

Le modèle logique de données

Il existe plusieurs modèles logiques de données.

Nous allons utiliser le mode relationnel afin de créer un modèle logique de données relationnel (M.L.D.R).

Ce modèle se compose d'un ensemble de **tables** et de **clés**.

Le M.L.D.R. peut être représenté de façon graphique ou sous une forme textuelle, le ***schéma relationnel***.

En mode texte, chaque table est représentée comme suit:

Nom de la Table(colonne1,colonne2,#colonne3)

Les champs composant la clé primaire sont soulignés et les clés étrangères sont précédées par un caractère #.

Le modèle logique de données

Le marqueur NULL

Lors de l'exploitation d'une base de données, il peut arriver que l'on ne connaisse pas la valeur d'un champ.

Lorsqu'une société envoie une facture à un client, il ne connaît pas encore la date du règlement de cette facture.

On affecte alors ce marqueur au champ correspondant.

La caractéristique de ce marqueur est qu'il est considéré comme étant différent de toute valeur, ***même de lui-même***.

Autrement dit, NULL est différent de NULL

D'où le terme de *marqueur* et pas de *valeur*.

Le modèle logique de données

Vous pouvez remarquer que, lors de la définition d'un champ, lorsque la case identifiant est cochée, la case NOT NULL qui lui est associée est cochée automatiquement.

En effet, si une partie de l'identifiant était NULL, elle serait différente de toute autre valeur et on ne pourrait jamais atteindre une ligne de la table.

Rubrique

Nom: champ1

Nom logique:

Type

Texte: ☒ Caractères ASCII ☐ Caractères Unicode ☐ Binaire

☒ Variable Longueur: 50 Collation:

☐ Fixe

☐ Volumineux

Numérique

☐ Entier

☐ Décimal

☐ Réel

☐ Monétaire

☐ Compteur

Date / Heure

☐ Date ☐ Heure ☐ Date-Heure

Autre

☐ Booléen ☐ Xml ☐ Géométrique ☐ Géographique

☐ Libre

Propriétés

☒ Identifiant ☒ NOT NULL

Complément:

Commentaire

OK Annuler

Le modèle logique de données

Le M.L.D.R. est uniquement composé de **tables**.

La première opération à effectuer est de transformer chaque entité du M.C.D. en table.

A l'origine chaque table contient un certain nombre de **colonnes** qui seront équivalentes aux champs de la classe entités correspondante.

Chaque ligne de la table correspondra à une instance de la classe entité.

Le modèle logique de données relationnel

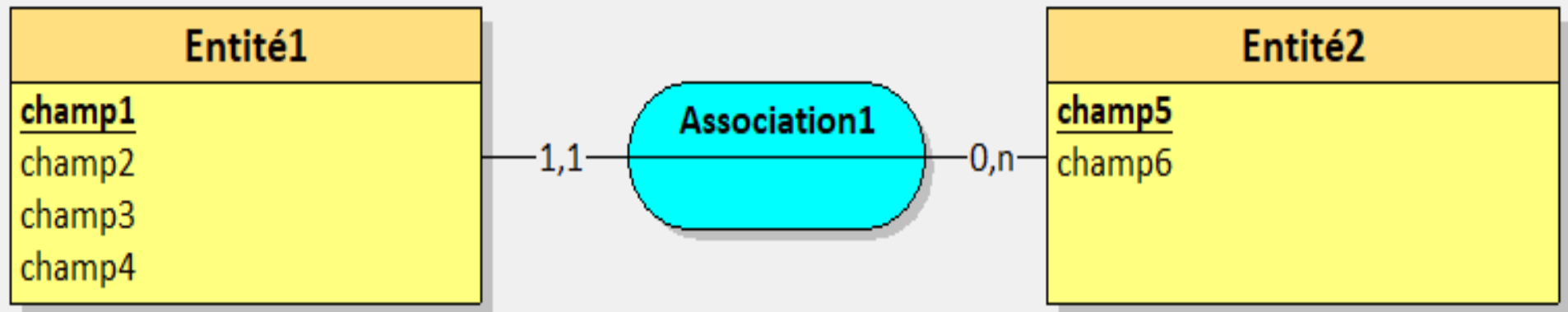
Les interactions entre les classes entités qui ont été symbolisées par des liens et des associations seront décrites par des **clés** que l'on appellera *clé étrangères* (**Foreign Key** ou **FK**) par opposition à l'identifiant de l'entité qui sera la *clé primaire* (**Primary Key** ou **PK**)

Le passage du M.C.D. au M.L.D.R. correspondant répond à un certain nombre de règles bien précise.

Il est réalisé automatiquement par Looping, mais il est nécessaire d'en connaître le mécanisme.

La compréhension de ces règles est plus simple en partant de la représentation Merise. Elles vont consister à éliminer les associations en créant les clés étrangères et certaines tables.

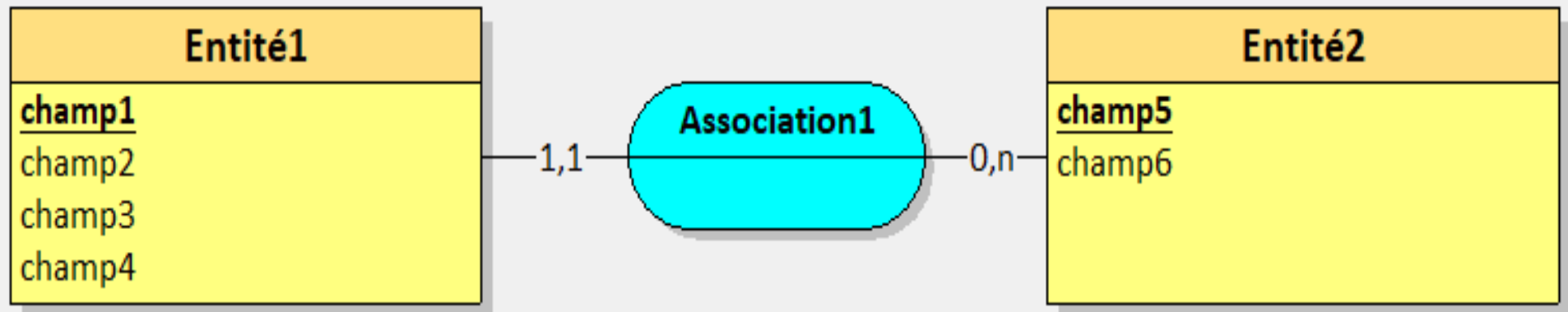
Le modèle logique de données relationnel



On définit le type d'une association par les deux cardinalités maximum (séparées par :) reliées à l'association: Il y a 3 possibilités:

- a) 1:1 association de un à un.
- b) 1:n association de 1 à plusieurs.
- c) n:m association de plusieurs à plusieurs.

Le modèle logique de données relationnel



Cette association est de type 1:n

Le modèle logique de données relationnel

On va traiter successivement les trois types d'association.

Dans le cas des association de 1 à plusieurs (1:n), on va créer une colonne supplémentaire dans la table qui est du côté 1;

Cette colonne sera une clé étrangère contenant la clé primaire de la table située du côté n.



Le modèle logique de données relationnel

On va traiter successivement les trois types d'association.

Dans le cas des association de 1 à plusieurs (1:n), on va créer une colonne supplémentaire dans la table qui est du côté 1;

Cette colonne sera une clé étrangère contenant la clé primaire de la table située du côté n.

Le schéma relationnel correspondant pourra être écrit comme suit:

Entité2 (champ5, champ6);

Entité1 (champ1, champ2, champ3, champ4 , #*champ5*);

Il n'est pas utile de préciser les caractéristiques des champs

Le modèle logique de données relationnel

On va traiter successivement les trois types d'association.

Dans le cas des association de 1 à plusieurs (1:n), on va créer une colonne supplémentaire dans la table qui est du côté 1;

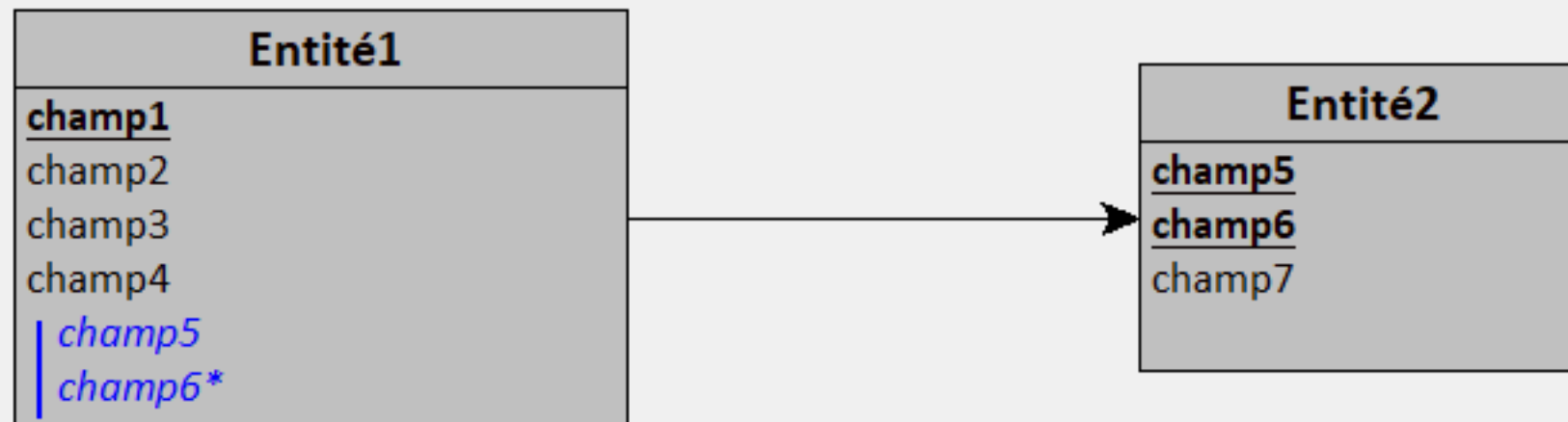
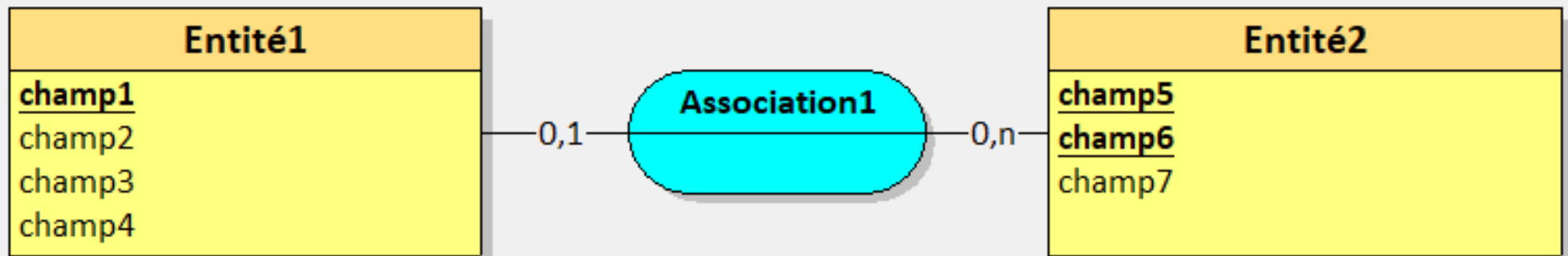


Le schéma relationnel s'écrit:

Entité2 (champ5, champ6, champ7);

Entité1 (champ1, champ2, champ3, champ4 , #(champ5,champ6));

Le modèle logique de données relationnel

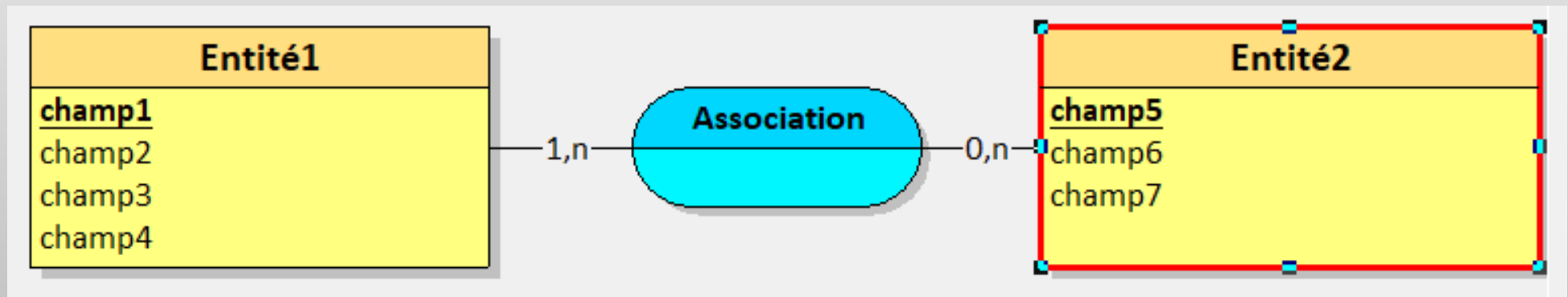


Le modèle logique de données relationnel

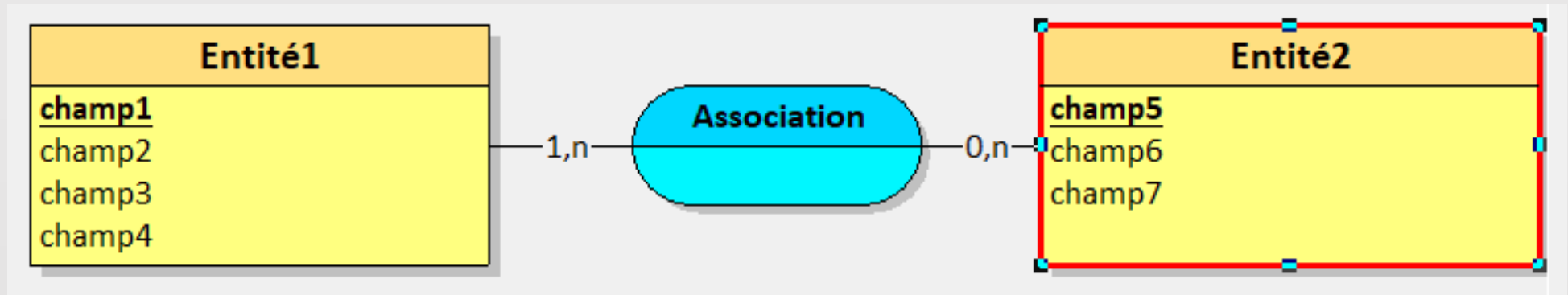
Dans le cas des association de plusieurs à plusieurs (n:m), on va créer une table supplémentaire (dite *table de jonction*)

Cette table va avoir pour clé primaire les clés primaires des deux tables concernées.

De plus, dans cette table, chacune des clés primaires sera une clé étrangère vers sa table d'origine.



Le modèle logique de données relationnel



Le modèle relationnel va s'écrire:

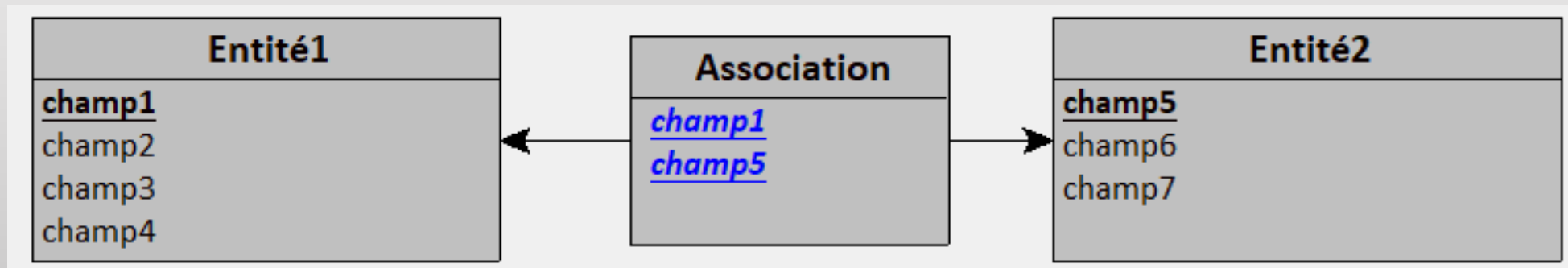
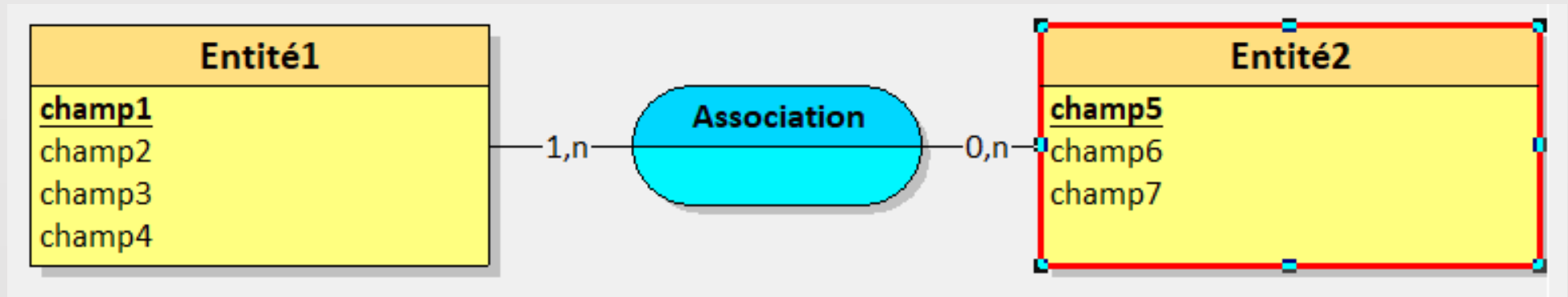
Entité1 (champ1, champ2, champ3, champ4);

Entité2 (champ5, champ6, champ7);

Association(#champ1, #champ5);

Le M.L.D.R. deviendra alors:

Le modèle logique de données relationnel

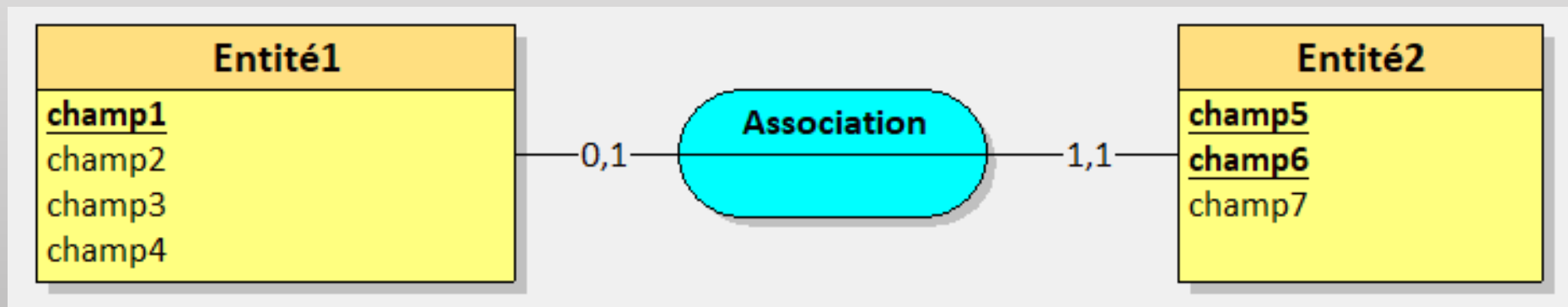


Le modèle logique de données relationnel

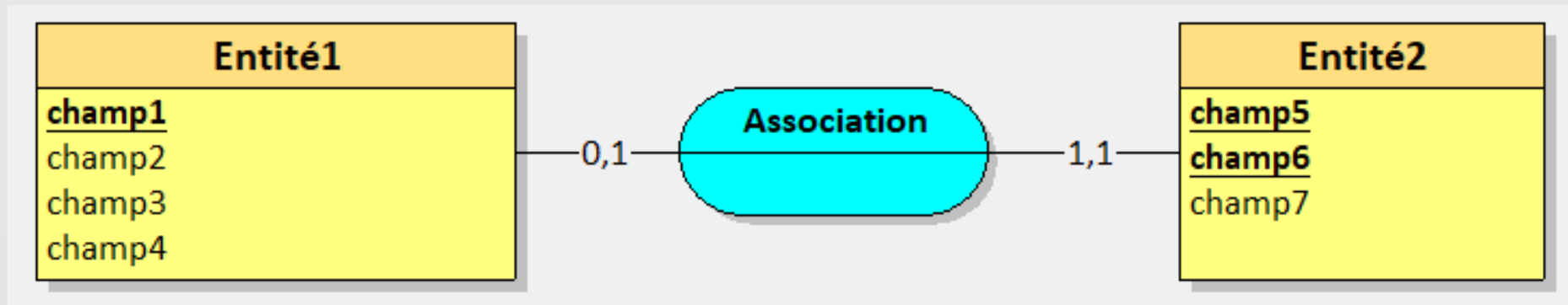
Dans le cas des association de un à un (1:1), on va effectuer le même traitement que dans le cas des associations 1:n à ceci près que les deux tables sont du côté 1.

On va rajouter la clé étrangère du côté opposé à la cardinalité 0:1; sinon de n'importe quel côté.

La clé étrangère sera affectée de la propriété unique en plus de la propriété NOT NULL.



Le modèle logique de données relationnel

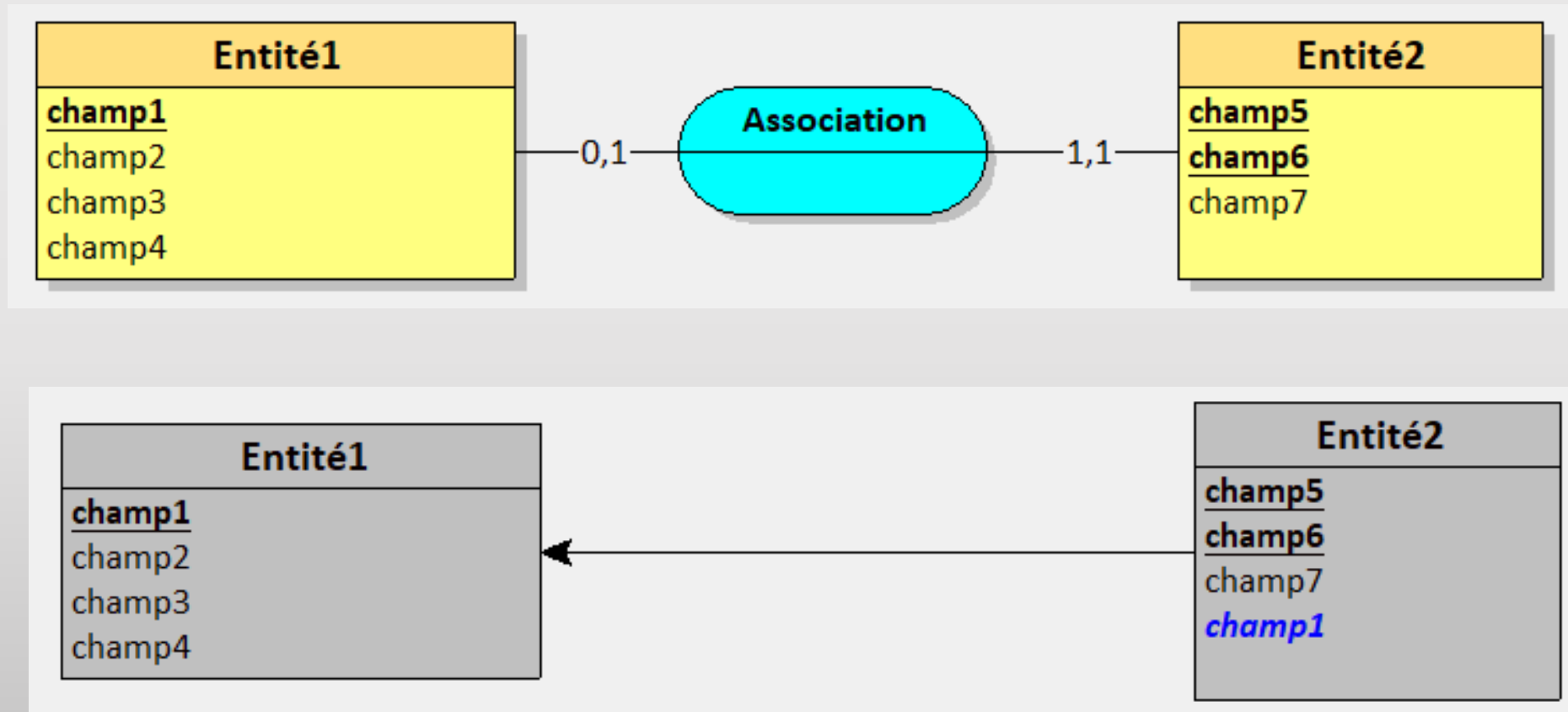


Le schéma relationnel sera :

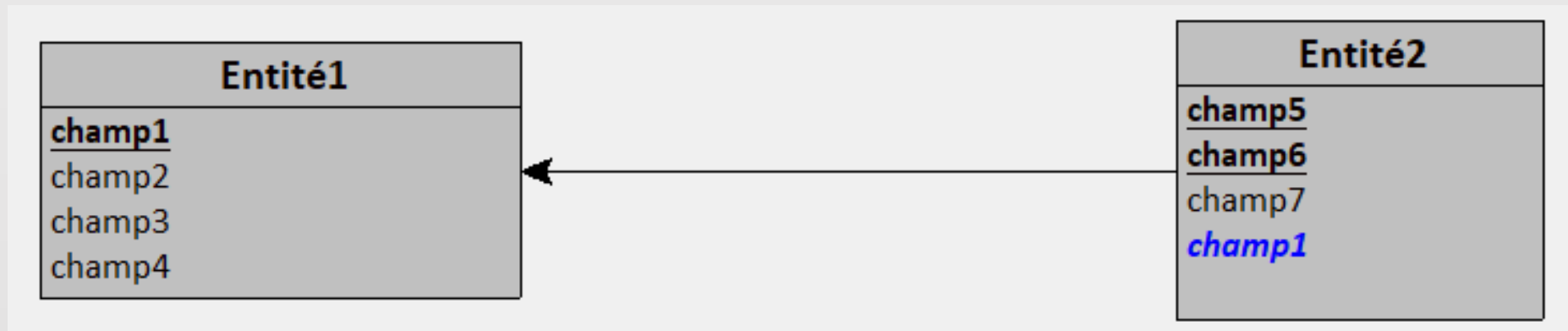
Entité1 (champ1, champ2, champ3, champ4);

Entité2 (champ5, champ6, champ7, #champ1 NOT NULL UNIQUE);

Le modèle logique de données relationnel



Le modèle logique de données relationnel



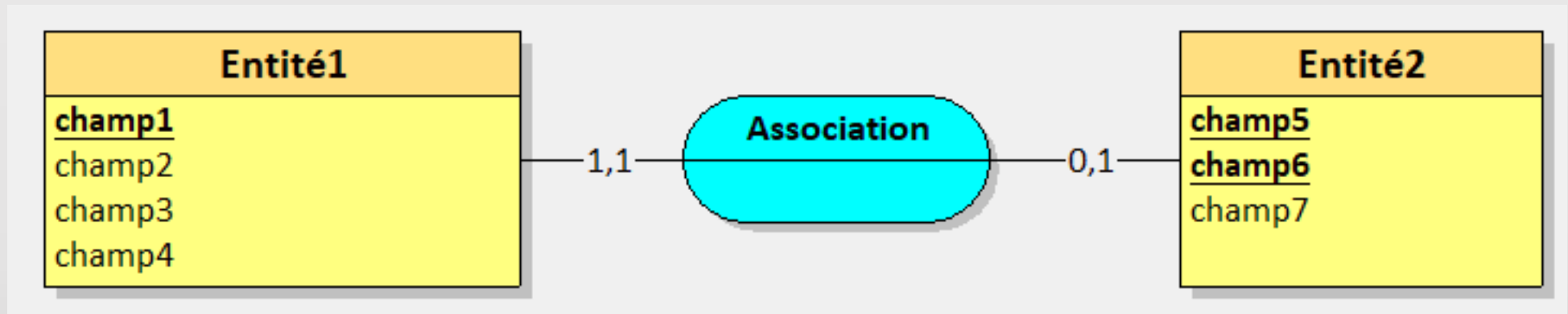
Le schéma relationnel sera :

Entité1 (champ1, champ2, champ3, champ4);

Entité2 (champ5, champ6, champ7, #champ1 NOT NULL UNIQUE);

Le modèle logique de données relationnel

Dans ce cas:

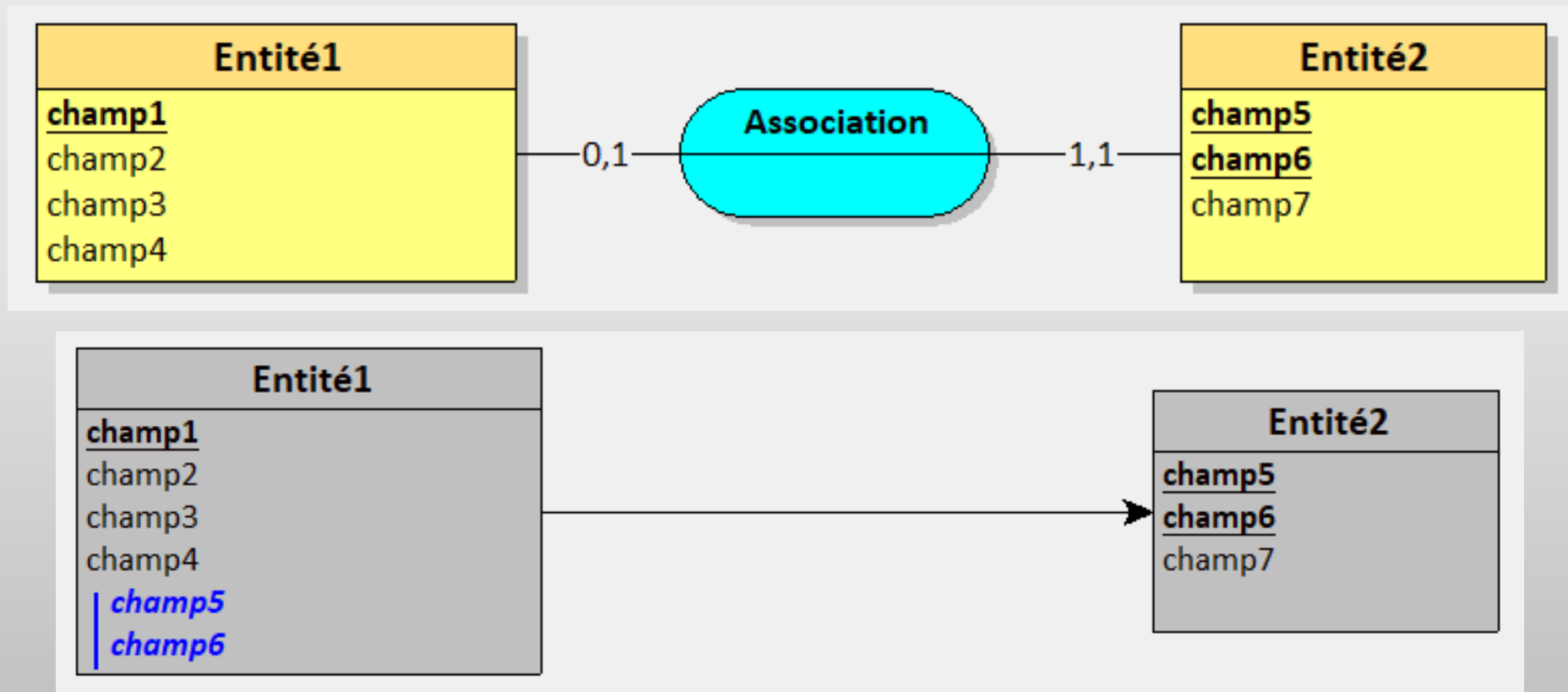


Le schéma relationnel serait:

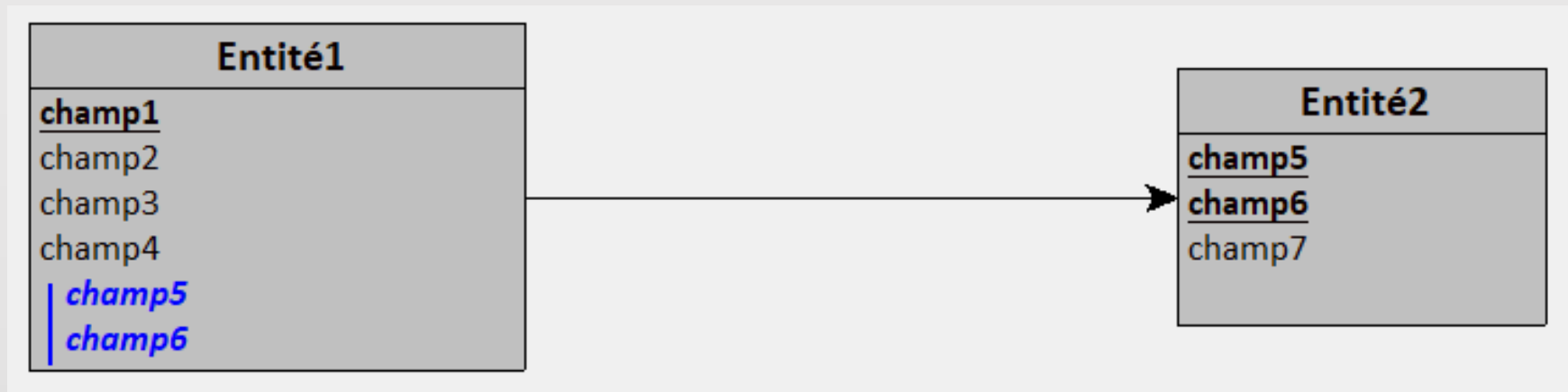
Entité1 (champ1, champ2, champ3, champ4,
#(champ5, champ6) UNIQUE NOT NULL);
Entité2 (champ5, champ6, champ7);

Le modèle logique de données relationnel

Dans ce cas, on aurait:



Le modèle logique de données relationnel

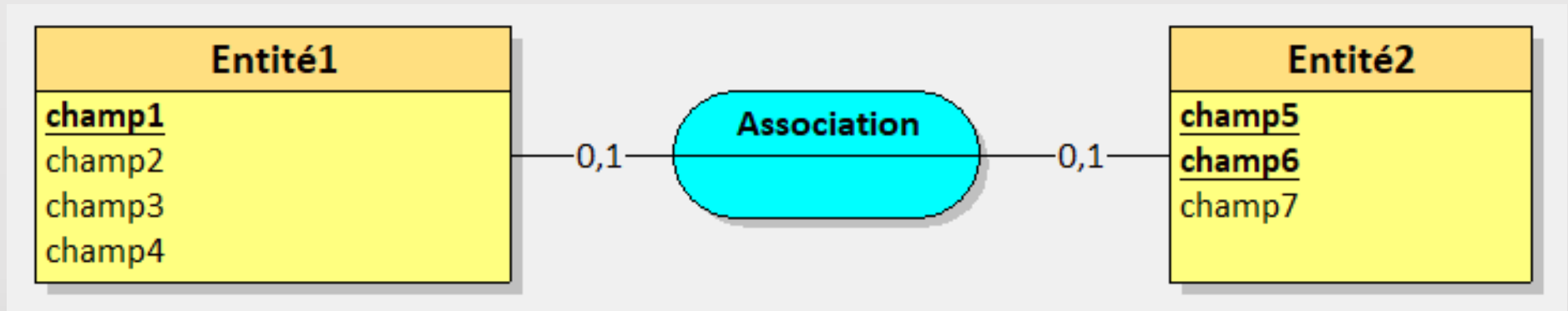


Le schéma relationnel serait:

Entité1 (champ1, champ2, champ3, champ4,
 #(champ5,champ6) UNIQUE NOT NULL);
Entité2 (champ5, champ6, champ7);

Le modèle logique de données relationnel

Dans ce cas:

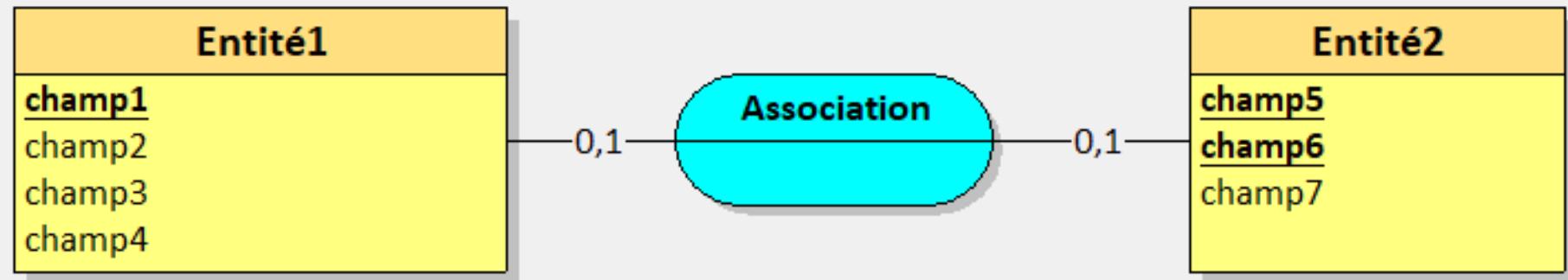


Les deux cardinalités sont semblables. On peut donc créer la clé étrangère de n'importe quel côté.

De façon automatique, pour de meilleures performances, il est préférable de créer une clé étrangère avec le minimum de champs.

Dans ce cas, on va créer la clé dans la table Entité2

Le modèle logique de données relationnel



Le modèle logique de données relationnel



Le schéma relationnel serait:

Entité1 (champ1, champ2, champ3, champ4);

Entité2 (champ5, champ6, champ7, #champ1 UNIQUE);

Le modèle logique de données relationnel

Il existe un dernier cas de figure que l'on ne peut pas traiter, car c'est une mauvaise pratique.

Il s'agit du cas où les deux cardinalisés sont de type 1:1.

Cela signifie que chacune des entités est associée à une seule autre entité.

C'est une mauvaise conception car, dans cette situation, on doit placer toutes les rubriques des deux tables dans la même entité.

En fait, dans certains cas de figure, une telle conception peut se justifier pour des questions de confidentialité et d'accès aux données.