

## Du modèle conceptuel au modèle physique

Lors de la conception d'une base de données, la transformation du **Modèle Conceptuel de Données (MCD)** au **Modèle Physique de Données (MPD)** est une étape essentielle. Cette transformation permet de passer d'une représentation abstraite des données à une implémentation concrète dans MySQL. Cet article présente les étapes clés de cette transformation avec des exemples concrets.

 niveau

En réalité, le passage du modèle conceptuel au modèle physique nécessite le passage par un modèle intermédiaire, le **MODÈLE LOGIQUE DE DONNÉES**. Pour des raisons de simplification dans ce cours, nous n'en parlerons pas. Sache juste que ça existe..

## Définitions Clés

- **Table** : Une structure de stockage qui organise les données en lignes et colonnes.
- **Colonne** : Un champ d'une table contenant des données d'un type particulier.
- **Clé primaire (PK)** : Un identifiant unique pour chaque enregistrement d'une table, garantissant son unicité.
- **Clé étrangère (FK)** : Un attribut qui référence la clé primaire d'une autre table, établissant ainsi une relation entre les tables.

## 1. Transformation d'une Entité en Table

Une **entité** du MCD devient une **table** en base de données. Chaque **attribut** devient une **colonne**, et un identifiant unique est défini.

**Exemple** : Entité **Utilisateur**

**MCD**

```
Utilisateur (  
    idUtilisateur [PK],  
    nom,  
    email,  
    date_inscription  
)
```

**\*\*##### MPD en MySQL**

```

1 CREATE TABLE Utilisateur (
2     idUtilisateur INT AUTO_INCREMENT PRIMARY KEY,
3     nom VARCHAR(100) NOT NULL,
4     email VARCHAR(255) UNIQUE NOT NULL,
5     date_inscription DATE NOT NULL
6 );

```

## 2. Transformation d'une Association 1-N

Une relation **1-N** signifie qu'un enregistrement d'une table A peut avoir plusieurs enregistrements dans une table B, mais chaque enregistrement de B est lié à un seul enregistrement de A.

Dans ce type d'association, une **clé étrangère** est utilisée pour établir le lien entre les tables. Une clé étrangère est un attribut qui référence la clé primaire d'une autre table, garantissant ainsi l'intégrité des données et empêchant les incohérences.

Dans une relation **1-N**, la clé étrangère est toujours ajoutée dans la table du côté **N** de l'association, c'est-à-dire dans la table qui contient plusieurs références à l'autre. Cela permet d'établir la connexion entre les enregistrements tout en maintenant une organisation logique des données.

**Exemple** : Relation **Utilisateur** - **Commande** (1 Utilisateur passe plusieurs Commandes)

### MCD

Utilisateur (idUtilisateur, nom, email, date\_inscription)  
 Commande (idCommande, date, montant)  
 Association "Passe" (1,N)

### MPD en MySQL

```

1 CREATE TABLE Commande (
2     idCommande INT AUTO_INCREMENT PRIMARY KEY,
3     date DATE NOT NULL,
4     montant DECIMAL(10,2) NOT NULL,
5     idUtilisateur INT NOT NULL,
6     FOREIGN KEY (idUtilisateur) REFERENCES Utilisateur(idUtilisateur) ON DE
7 );

```

## 3. Transformation d'une Association N-N

Une relation **N-N** implique qu'un enregistrement de la table A peut être lié à plusieurs enregistrements de la table B, et inversement.

Dans ce type d'association, il est nécessaire d'ajouter une **table de jointure** pour stocker les relations entre les deux entités. Cette table de jointure contiendra **les clés étrangères** des deux tables associées, permettant ainsi d'établir les connexions entre les enregistrements des deux entités. Chaque ligne de la table de jointure

représente une association unique entre un enregistrement de la première table et un enregistrement de la seconde.

Dans la table de jointure, la définition `PRIMARY KEY (idEtudiant, idCours)` signifie que la combinaison des deux clés étrangères forme une **clé primaire composite**. Cela garantit que chaque paire `(idEtudiant, idCours)` est unique, c'est-à-dire qu'un étudiant ne peut s'inscrire qu'une seule fois à un même cours. Cette structure assure également une efficacité optimale pour les jointures et les requêtes de recherche sur ces relations.

**Exemple :** Relation `Etudiant` - `Cours` (Un étudiant suit plusieurs cours et inversement)

#### MCD

Etudiant (idEtudiant, nom)  
Cours (idCours, titre)  
Association "S'inscrit" (N,N)

#### MPD en MySQL

```
1 CREATE TABLE Inscription (  
2     idEtudiant INT NOT NULL,  
3     idCours INT NOT NULL,  
4     PRIMARY KEY (idEtudiant, idCours),  
5     FOREIGN KEY (idEtudiant) REFERENCES Etudiant(idEtudiant) ON DELETE CASCADE  
6     FOREIGN KEY (idCours) REFERENCES Cours(idCours) ON DELETE CASCADE  
7 );
```

## 4. Transformation d'une Association avec Attributs

Si une **association possède des attributs**, elle devient une **table propre**.

Dans ce cas, il est nécessaire de créer une table distincte pour l'association, qui contiendra **les clés étrangères** des deux entités associées ainsi que les attributs propres à l'association. Chaque enregistrement de cette table représentera une relation unique entre les entités tout en stockant les informations supplémentaires qui leur sont liées.

Dans cette table, la définition `PRIMARY KEY (idEmploye, idProjet)` signifie que la combinaison des deux clés étrangères constitue une **clé primaire composite**. Cela garantit que chaque association entre un employé et un projet est unique. En d'autres termes, un employé ne peut être assigné qu'une seule fois à un projet donné. Cette structure assure également une intégrité référentielle et optimise les performances des requêtes qui exploitent ces relations.

**Exemple :** Relation `Employé` - `Projet` avec un attribut `rôle`

#### MCD

Employe (idEmploye, nom)  
Projet (idProjet, titre)  
Association "Travaille\_sur" (N,N) avec attribut "rôle"

```

1 CREATE TABLE Travailleur_sur (
2     idEmploye INT NOT NULL,
3     idProjet INT NOT NULL,
4     role VARCHAR(50) NOT NULL,
5     PRIMARY KEY (idEmploye, idProjet),
6     FOREIGN KEY (idEmploye) REFERENCES Employe(idEmploye) ON DELETE CASCADE
7     FOREIGN KEY (idProjet) REFERENCES Projet(idProjet) ON DELETE CASCADE
8 );

```






## 5. Récapitulatif des Transformations

Concept MCD	Transformation en MySQL
Entité	Table avec colonnes et clé primaire
Association 1-N	Ajout d'une clé étrangère dans la table côté "N"
Association N-N	Table d'association avec deux clés étrangères
Association avec attributs	Table d'association avec colonnes supplémentaires

## Conclusion

La transformation du **MCD en MPD** est une étape essentielle pour structurer une base de données relationnelle. En suivant ces principes, on garantit une conception efficace, optimisée pour les requêtes et respectant les contraintes d'intégrité.

## Ce qu'il faut retenir

- Une **entité** devient une **table** avec des colonnes et une clé primaire .
- Une **relation 1-N** se traduit par l'ajout d'une **clé étrangère** dans la table du côté "N" .
- Une **relation N-N** nécessite une **table de jointure** avec deux clés étrangères .
- Une **association avec attributs** devient une **table propre** avec ses propres colonnes .
- Les **clés primaires composites** garantissent l'unicité des relations et optimisent les performances .

Si vous avez des questions ou souhaitez approfondir un point, n'hésitez pas à commenter ! .