

Exercices Relations

Exercice 1 : Relation "Un-à-Un" (Facile)

Objectif :

Comprendre une relation **un-à-un** entre deux entités.

Contexte :

Imaginons une base de données pour une bibliothèque. Il existe une table `Membre` et une table `Abonnement`.

- Un **membre** peut avoir un **seul abonnement**.
- Un **abonnement** peut être attribué à un **seul membre**.

Tâches :

1. Crée une table `Membre` avec les colonnes `id`, `nom`, `email`.
2. Crée une table `Abonnement` avec les colonnes `id`, `date_debut`, `date_fin`.
3. Ajoute une relation **un-à-un** entre `Membre` et `Abonnement` (chaque membre a un abonnement, et chaque abonnement est attribué à un seul membre).

Question :

- Quelle est la cardinalité de la relation entre `Membre` et `Abonnement` ?

Exercice 2 : Relation "Un-à-Plusieurs" (Intermédiaire)

Objectif :

Comprendre une relation **un-à-plusieurs** entre deux entités.

Contexte :

Dans une base de données pour une école, chaque **professeur** peut enseigner à plusieurs **étudiants**, mais chaque **étudiant** a un seul **professeur principal**.

Tâches :

1. Crée une table `Professeur` avec les colonnes `id`, `nom`, `email`.
2. Crée une table `Etudiant` avec les colonnes `id`, `nom`, `professeur_id` (clé étrangère référencée à la table `Professeur`).
3. Assure-toi que la relation entre `Professeur` et `Etudiant` est **un-à-plusieurs**.

Question :

- Quelle est la cardinalité de la relation entre `Professeur` et `Etudiant` ?

Exercice 3 : Relation "Plusieurs-à-Plusieurs" (Intermédiaire)

Objectif :

Comprendre une relation **plusieurs-à-plusieurs** entre deux entités.

Contexte :

Imaginons une base de données pour une bibliothèque avec des **livres** et des **auteurs**. Un **livre** peut avoir plusieurs **auteurs**, et un **auteur** peut avoir écrit plusieurs **livres**.

Tâches :

1. Crée une table `Livre` avec les colonnes `id`, `titre`, `date_publication`.
2. Crée une table `Auteur` avec les colonnes `id`, `nom`, `email`.
3. Crée une table de jointure `Livre_Auteur` avec les colonnes `livre_id` et `auteur_id` pour gérer la relation **plusieurs-à-plusieurs** entre `Livre` et `Auteur`.

Question :

- Quelle est la cardinalité de la relation entre `Livre` et `Auteur` ?

Exercice 4 : Cardinalité et Intégrité Référentielle (Avancé)

Objectif :

Comprendre l'intégrité référentielle avec des contraintes de clé étrangère et des relations.

Contexte :

Dans une base de données pour une entreprise, chaque **département** peut avoir plusieurs **employés**, mais chaque **employé** appartient à un **seul département**.

Tâches :

1. Crée une table `Departement` avec les colonnes `id`, `nom`, `budget`.
2. Crée une table `Employe` avec les colonnes `id`, `nom`, `departement_id` (clé étrangère référencée à `Departement`).
3. Ajoute des contraintes d'intégrité référentielle pour que chaque `Employe` ait une référence valide à un `Departement`.

Question :

- Explique l'intégrité référentielle dans ce contexte et comment elle est assurée par la clé étrangère.

Exercice 5 : Relations Complexes avec des Contraintes de Cardinalité (Difficile)

Objectif :

Gérer des relations complexes avec des contraintes de cardinalité et des contraintes supplémentaires.

Contexte :

Imaginons une base de données pour une plateforme de **cours en ligne**. Un **cours** peut avoir plusieurs **étudiants** et plusieurs **professeurs**, mais chaque **étudiant** peut suivre plusieurs **cours**, et chaque **professeur** peut enseigner plusieurs **cours**. De plus, un étudiant peut être inscrit à un cours en tant qu'**auditeur** ou comme **étudiant actif** (ce qui affecte les résultats).

Tâches :

1. Crée une table `Cours` avec les colonnes `id`, `titre`, `description`.
2. Crée une table `Etudiant` avec les colonnes `id`, `nom`, `email`.
3. Crée une table `Professeur` avec les colonnes `id`, `nom`, `email`.
4. Crée une table de jointure `Cours_Etudiant` avec les colonnes `cours_id`, `etudiant_id`, `role` (indiquant si l'étudiant est auditeur ou actif).
5. Crée une table de jointure `Cours_Professeur` avec les colonnes `cours_id`, `professeur_id`.
6. Explique la cardinalité de chaque relation et comment les jointures permettent de gérer cette situation complexe.

Question :

- Quelle est la cardinalité de la relation entre **Etudiant** et **Cours** ? Quelle est la cardinalité de la relation entre **Professeur** et **Cours** ? Comment les rôles dans la table `Cours_Etudiant` affectent cette relation ?

Solution Diagramme

Solution SQL