

# BASES DE DONNÉES

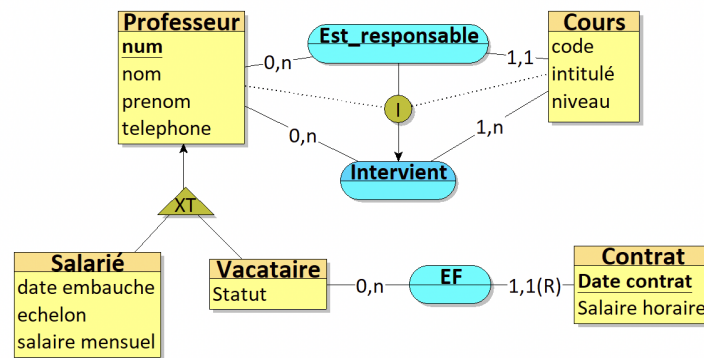
## Gestion des contraintes

Travaux Pratiques

13 octobre 2022

### Exercice 1 :

Un club de danse souhaite se doter d'une base de donnée pour gérer ses intervenants dans les cours. Après analyse, on propose le schéma Entité/Association ci-dessous.



1. Traduisez ce schéma en relationnel ; en ne tenant pas compte pour l'instant des contraintes XT et I. Créez les relations obtenues, ainsi que les clés et clés étrangères qui en découlent.  
**Indication 1** : Puisque qu'un cours a exactement un responsable, cette association se traduira par une clé étrangère dans la relation "Cours", ne pouvant prendre la valeur NULL.  
**Indication 2** : Les spécialisations et entités faibles induisent une contrainte d'existence ; par exemple, supprimer un professeur de la base induit de le supprimer également en tant que salarié ou vacataire. De même supprimer un vacataire, induit de supprimer les contrats qui s'y rapportent. Faites en sorte que toutes ces suppressions en cascade soient automatiques. (Option : ON DELETE CASCADE des clés étrangères.)
2. La contrainte I sur le schéma traduit le fait que si un professeur est responsable d'un cours, alors il intervient dans ce cours. C'est une implication/inclusion : tout couple (professeur, cours) de l'association "Est-Responsable" doit exister dans l'association "Intervient".
  1. Cette contrainte peut-elle être implantée par une clé étrangère dans la relation 'cours' ?
  2. Créez cette contrainte dans la base de données.
  3. Créez un professeur "Jean Wiejacka" (sans se soucier encore qu'il soit vacataire ou salarié). Créez un cours de salsa débutant, pour lequel Jean Wiejacka est le responsable. Quel problème rencontrez vous ?
3. Une transaction<sup>1</sup> est un ensemble de mises à jour considérées comme une seule opération, sans limite sur le nombre d'opérations dans une même transaction. Les SGBD relationnels garantissent que les contraintes sont satisfaites AVANT et APRES la transaction. Ce qui se passe PENDANT la transaction dépend des capacités des SGBD et des choix de l'opérateur. La vérification de certaines contraintes, comme les clé étrangères sous PostgreSQL, peut être différée à l'issu de la transaction.

1. Une transaction commence par "BEGIN" et se termine par "COMMIT"

1. Modifiez, lorsque cela est nécessaire, les déclarations de clés étrangère afin de différer leur vérification en fin de transaction.
2. Procédez maintenant à la création du cours de SALSA à l'aide d'une transaction contenant plusieurs insertions de tuples.
4. Le cahier des charges spécifie qu'un cours a obligatoirement (au moins) un intervenant (participation obligatoire à l'association "Intervient"). Peut-on considérer que cette contrainte est bien garantie dans notre base de données ?

5. Implanter la contrainte  $X$  indiquée dans la spécialisation des professeurs, indiquant qu'un professeur ne peut pas être à la fois un salarié et un vacataire.

**Indication :** Cette contrainte peut se programmer à l'aide de deux commandes "CHECK" dans la déclaration des relations "salaries" et "vacataires", par exemple sous la forme "Check (est\_vacataire(num\_prof) = false)" - où "est\_vacataire(num\_prof)" est une fonction qui retourne vrai si "num\_prof" existe déjà dans vacataires, faux sinon.

6. Implanter la contrainte  $T$  de la spécialisation, qui impose que tous les professeurs ont la position de salarié ou de vacataire. Lors de la création d'un professeur, vous pourrez utiliser une technique similaire à la précédente. Mais vous devez contrôler la suppression d'un vacataire ou salarié, en utilisant des "déclencheurs" (exemple : <https://docs.postgresql.fr/current/plpgsql-trigger.html>).

**Important :** Faites des tests de toutes vos contraintes ! Par ailleurs, assurez d'être finalement toujours en capacité de :

- Créer un professeur (qui devra donc être vacataire ou salarié)
- Passer un professeur de salarié à vacataire et inversement
- Supprimer un professeur

- 
- documentation officielle de PostgreSQL : <https://docs.postgresql.fr/current/>
  - Lors de la définition d'une clé étrangère, l'option "DEFERRABLE" permet de différer la vérification.
    - Pour une vérification différée, on commence la transaction concernée par "SET CONSTRAINTS nom-contrainte DEFERRED".
  - Suivant les options, les déclencheurs peuvent se déclencher :
    - Pour les insertions (INSERT), les suppressions (DELETE), les mises à jour (UPDATE)
    - Avant (BEFORE) ou après (AFTER) l'évènement déclenchant
    - A chaque ligne affectée par la commande (FOR EACH ROW) ou globalement sur l'instruction (FOR EACH STATEMENT)
  - Il est possible de déclarer un déclencheur comme une contrainte (CREATE CONSTRAINT TRIGGER)
    - Dans ce cas, il peut lui-même avoir la caractéristique "DEFERRABLE"
    - A condition d'être du type "FOR EACH ROW".