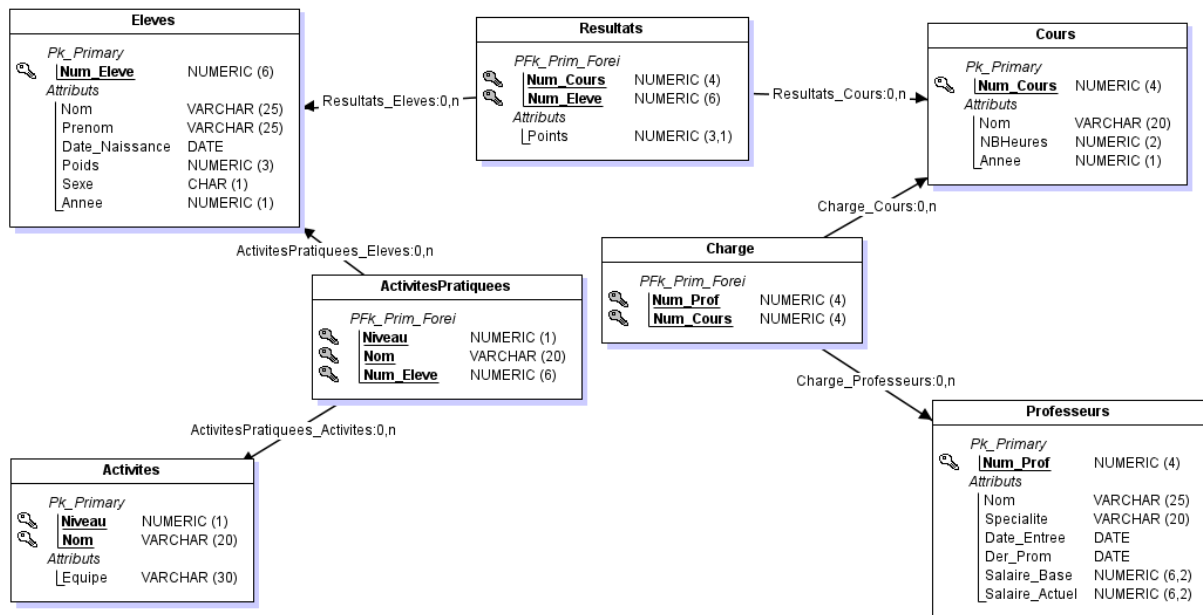


TP4 – Interopérabilité BDD

Cas Ecole

I. Description de la base de données

Soit la base de données ecole dont le schéma relationnel est le suivant :



II. Les transactions

Pour cette partie, vous pouvez utiliser votre serveur MySQL local et MySQL Workbench comme client. Afin d'avoir un contrôle total sur vos transactions, vous devez désactiver la validation automatique à chaque fois que vous ouvrez une

nouvelle connexion¹ à votre base de données. Pour ce faire, cliquez sur l'icône *«Activer/désactiver le mode autocommit»*.

À partir de ce moment, les transactions seront gérées par `START TRANSACTION`, `COMMIT` ou `ROLLBACK` que vous émettez ou en cliquant sur les icônes *«COMMIT»/«ROLLBACK»*.

Notez que pour que les transactions fonctionnent correctement sous MySQL, vos tables doivent utiliser le moteur de stockage InnoDB: assurez-vous que toutes vos tables de la base de données Ecole sont en InnoDB.

Pour ouvrir deux connexions distinctes à votre base de données, (que les connexions soient ouvertes pour le compte d'utilisateurs distincts ou pour le compte du même utilisateur n'a pas d'importance.) ouvrez deux connexions sous la même instance MySQL Workbench: cela créera deux onglets principaux dans la fenêtre du workbench, un pour chaque connexion. Notez que tous les onglets de requête de n'importe quel onglet principal se rapportent à la même connexion et partagent donc le même environnement de transaction.

1. Commit et Rollback

Vous ne devez utiliser qu'une seule connexion dans cet exercice. Après avoir désactivé l'autocommit, démarrez une nouvelle transaction. Insérez ensuite un nouveau département dans la table Eleves. Vérifiez que le nouvel élève est visible par la transaction en cours en émettant une instruction `Select`. Ensuite, exécuter une commande **Rollback**, puis une nouvelle instruction `Select`.

Qu'observez-vous ?

Répétez cette expérience en exécutant la commande **Commit** à la place de la commande **Rollback**.

¹ **autocommit** est un paramètre de connexion, pas un paramètre utilisateur

Qu'observez-vous ?

2. Transaction et remise sur panne

Comme dans la question précédente, démarrez une nouvelle transaction et insérez un nouvel élève dans la table Eleves. Fermez ensuite l'onglet de connexion. Puis, lancez à nouveau MySQL Workbench et vérifiez si l'élève que vous venez d'insérer est dans la table Eleves.

Répétez cette expérience, mais cette fois, arrêtez brusquement le programme client en arrêtant par exemple le processus avec le gestionnaire de tâches.

Qu'observez-vous ?

3. Niveau d'isolation des transactions

Déterminez le niveau d'isolation des transactions par défaut de votre base de données en exécutant la commande suivante :

```
show variables like '%isolation%'
```

Ensuite, concevez l'expérience la plus simple possible qui démontre que par défaut, les modifications effectuées dans une transaction sur la base de données ne sont visibles que par cette transaction. Puis, considérons par exemple l'insertion d'un nouvel élève dans la table Eleves. Répétez l'expérience en supprimant l'élève que vous venez d'insérer.

4. Niveau d'isolation "*read uncommitted*"

Déterminez le niveau d'isolation des transactions par défaut de votre base de données. Répétez ensuite l'expérience précédente et définissez le niveau d'isolation des transactions sur **read uncommitted** en émettant l'instruction correspondante. Cela définira le niveau d'isolation de toutes les transactions à venir.

Quelle transaction devez-vous modifier pour que le comportement soit différent de l'exercice précédent ?

Vérifiez l'effet sur les deux transactions.

5. Niveau d'isolation "*serializable*"

Répétez cette fois l'expérience précédente en utilisant le niveau d'isolation sérialisable. Qu'observez-vous sur la deuxième transaction ? Expliquer.