Instructions générales

1. Les instructions générales du projet de SQL restent d'application pour l'examen.
    - Utilisez un schéma nommé **examen**.
    - Pour toute fonctionnalité demandée :
        - Si c'est possible, implémentez-la sur base de contraintes d'intégrité.
        - Sinon, si c'est possible, implémentez-la sur base d'un trigger.
        - Sinon, si c'est possible, implémentez-la sur base d'une vue.
        - Sinon, si c'est possible, implémentez-la sur base d'une procédure stockée.
    - Utilisez le modèle transactionnel et évitez d'ignorer silencieusement les erreurs.
    - Attention aux injections de SQL.
    - Chaque PreparedStatement ne doit être préparé qu'une seule fois.
    - Il ne faut, en Java, faire des SELECT **que** sur une seule chose à la fois. Cette chose peut être une table, une vue ou une procédure stockée.
2. Vous n'avez pas accès à Internet, mais vous trouverez dans le répertoire U: une copie du contenu du cours qui se trouve sur moodle.
3. Vous devez créer :
    - un fichier script.sql contenant toutes les commandes SQL à exécuter au serveur ;
    - un projet Java contenant la ou les classes Java.
4. 15 minutes avant la fin de l'examen :
    a. Vous recopiez le fichier script.sql à la racine de votre projet Java.
    b. Vous créez un fichier NOM_PRENOM.zip (par exemple DAMAS_CHRISTOPHE.zip) contenant l'intégralité de votre projet Java.
    c. Vous copiez ce fichier à la racine du disque U:.
    d. Finalement vous vérifiez que ce fichier contient bien le projet Java contenant le fichier script.sql et le répertoire src avec vos sources Java.

ATTENTION DATAGRIP PLANTE PARFOIS : SAUVEZ SOUVENT !

Enoncé

Vous avez la responsabilité de développer un logiciel simplifié pour la gestion des séries au sein de notre école. Lors de la création d'un étudiant dans ce logiciel, il lui est assigné une série. Certains étudiants peuvent demander à changer de série, sous certaines conditions :

- C'est la première fois que l'étudiant demande de changer de série
- La nouvelle série appartient au bon bloc (bloc de l'étudiant).
- La série initiale de l'étudiant était parmi les séries les plus peuplées de son bloc. Ex : s'il y a 3 séries dans un bloc particulier avec 14 étudiants pour la première, 12 pour la deuxième et 14 pour la troisième, il ne sera pas permis de quitter la deuxième série. Par contre, il sera permis de quitter la première et la troisième série.

L'objectif de l'examen est d'implémenter la procédure qui fera le changement de séries. Nous ne vous demandons pas d'implémenter la procédure d'insertion d'un étudiant qui vérifiera si sa série est bien du bon bloc.

Tables

Il y aura trois tables : `blocs`, `series` et `etudiants`.

Pour un bloc, on retient uniquement son numéro qui doit être compris entre 1 et 3. Ce numéro est la clé primaire de la table

La clé primaire de la table `series` est un entier auto-incrémenté. On retient également pour une série son bloc et son code unique. Son code commencera par le numéro du bloc suivi de 'BIN' suivi d'un chiffre (entre 1 et 9). Ex : « 2BIN3 ».

La clé primaire de la table étudiants est aussi un entier auto-incrémenté. Pour un étudiant, on retient également son nom, son prénom, son bloc et sa série. On retient également si l'étudiant a déjà changé de série (à sa création, il n'a pas changé de série). Tous les champs de cette table sont obligatoires.

Attention, dans le cadre de cet examen, vous ne pouvez pas rajouter de champs supplémentaires dans les tables.

À développer :

- Il faut créer une procédure stockée pour changer un étudiant de série. Cette procédure prend 2 paramètres : la clé primaire de l'étudiant et le code de la série que l'étudiant veut rejoindre. La procédure renverra le nombre de séries vides dans le bloc de l'étudiant (séries qui ne contiennent aucun étudiant).

  Procédure

  Une exception sera lancée dans les cas suivants :
  - Si la série en paramètre est la série actuelle de l'étudiant
  - Si l'étudiant a déjà changé de série
  - Si la série en paramètre n'appartient pas au même bloc que celui de l'étudiant.
  - Si la série initiale de l'étudiant n'est pas parmi les séries les plus peuplées du bloc.

Trigger

- Il faut créer un mécanisme pour mettre à jour le champ qui retient si l'étudiant a déjà changé de série

VIEW

- Il faut créer une classe Java qui se connecte au serveur. Le programme demandera à l'utilisateur de fournir un numéro de bloc (via un `Scanner`) et affichera ensuite tous les séries de ce bloc. Pour chaque série, on affichera son code, ainsi que le nombre d'étudiants qu'elle contient. Si une série est vide, elle devra quand même apparaitre avec 0 comme nombre d'étudiants. Les résultats seront triés par ordre décroissant du nombre d'étudiants.

  Il n'est pas nécessaire de gérer les cas d'erreurs et il ne faut pas perdre de temps à afficher les données de manière jolie.

  Pour rappel, voici deux instructions utiles pour les Scanner :

```
Scanner s = new Scanner(System.in);
int numero  = Integer.parseInt(s.nextLine());
```

Si vous voulez tester votre système, vous pouvez insérer des blocs, des séries et des étudiants (via de simples INSERT) et appeler la procédure stockée d'insertion d'une enchère dans votre script.