

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

Une école vous demande d'implémenter un système simplifié permettant la réservation de locaux pour leurs examens. Dans ce système, les examens ont lieu sur une journée complète : un local ne peut accueillir qu'un examen par jour. On dira qu'un examen est complètement réservé si la somme des places des locaux réservés pour cet examen est suffisante pour accueillir tous les étudiants inscrits. À sa création, un examen n'est bien sûr pas complètement réservé.

Tables

Il y aura trois tables : examens, locaux et reservations.

Pour un examen, on retient son code unique (le code d'un examen commence toujours par BINV et est suivi de 4 chiffres ; ex : BINV2040), son nom, son bloc (1,2 ou 3), sa date, le nombre d'étudiants inscrits et s'il est complètement réservé. On retient également si l'examen a lieu sur machine ou pas (booléen).

Pour un local, on retient son nom et son nombre de places. On retient également si des machines sont disponibles dans le local (booléen). Deux locaux ne peuvent avoir le même nom.

Pour une réservation, on retiendra son examen et son local. On ne peut bien sûr pas réserver deux fois le même local pour un examen.

Attention, dans le cadre de cet examen, vous ne pouvez pas rajouter de champs supplémentaires dans les tables examens, locaux et réservations hormis des clés primaires.

À développer :

- Il faut créer une procédure stockée pour insérer une réservation. Cette procédure prend 2 paramètres : le nom du local et le code de l'examen. La méthode renverra le nombre total d'examens complètement réservés se déroulant dans le local en paramètre.

Une exception sera lancée dans les cas suivants :

- si l'examen se déroule sur machine et qu'il n'y a pas de machine disponible dans le local. Notez qu'un examen n'ayant pas lieu sur machine peut se dérouler dans un local disposant de machines ;
 - si l'examen est complètement réservé ;
 - s'il existe déjà une réservation pour un autre examen dans le local le même jour.
- Il faut créer un mécanisme qui permet de mettre à jour le champ de la table examen qui retient si l'examen est complètement réservé. À chaque réservation, il faut vérifier si le nombre de locaux est suffisant pour accueillir tous les étudiants inscrits.
- Une classe Java qui se connecte au serveur : le programme demande à l'utilisateur le numéro d'un bloc (via un Scanner) et affichera ensuite tous les examens du bloc, triés par date, à la console. Pour chaque examen, il affichera son nom, sa date et le nombre de locaux réservés pour cet examen. Si un examen n'a pas encore de locaux réservés,

il doit quand même apparaître (avec 0 comme nombre de locaux réservés). 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 bloc = Integer.parseInt(s.nextLine());
```

Si vous voulez tester votre système, vous pouvez insérer des locaux et des examens (via de simples INSERT) et appeler la procédure stockée de réservation dans votre script.