

Projet Gestion De Données 2020

Une école désire développer un système de réservations de locaux pour ses examens. Le système a deux objectifs principaux :

- Permettre d'éviter de créer des mauvais horaires d'examen. Premièrement, le système permettra d'éviter les conflits horaires. Deux examens ne pourront pas se chevaucher s'il existe un étudiant inscrit à ces deux examens. Deux examens se déroulant dans le même local ne pourront pas non plus se chevaucher. De plus, le système évitera de placer deux examens du même bloc le même jour.
- Permettre d'éviter de réserver trop de locaux. On dira qu'un examen est complètement réservé si la somme des places des locaux réservés pour cet examen est suffisant pour accueillir tous les étudiants inscrits. Lorsqu'un examen est complètement réservé, le système empêchera la réservation de locaux supplémentaires pour cet examen.

L'utilisation du système se déroule en quatre phases :

- D'abord, l'administrateur encode les examens. Pour chaque examen, il encodera son code unique (le code d'un examen commence toujours par IPL et est suivi de 3 chiffre ; ex : IPL204), son nom, son bloc, sa durée et si l'examen se déroule sur machine.
- Puis, les étudiants s'inscrivent aux examens qu'ils désirent présenter. Ils peuvent s'inscrire à n'importe quel examen. On ne peut bien sûr pas s'inscrire deux fois au même examen.
- Ensuite, l'administrateur du système va encoder l'heure de début de chaque examen (date, heure et minutes). Le système empêchera l'encodage d'une heure de début qui provoquerait un conflit horaire pour un étudiant inscrit à l'examen. De plus, deux examens du même bloc ne peuvent pas se dérouler le même jour. Lorsque l'heure de début d'un examen a été fixée, il ne sera plus possible à un étudiant de s'inscrire à cet examen.
- Finalement, l'administrateur va réserver des locaux pour chaque examen.

Notez que votre système ne doit être utilisé que pour une seule session. Pour organiser une nouvelle session, on vide simplement la DB.

Le système sera découpé en deux applications java : une pour les utilisateurs, l'autre pour les administrateurs.

Avant d'utiliser le système, on rajoutera manuellement (via de simple insert) les formations et les blocs. Pour une formation, on retient son nom (par exemple « Bachelier en Informatique de Gestion ») et son école. Pour un bloc, on retient son code unique (par exemple « 2BIN ») et sa formation.

Application centrale

Cette application offrira un menu permettant d'accéder à huit points :

1. Ajouter un local. Pour chaque local, on retiendra son nom unique et son nombre de places. On retiendra également si des machines sont disponibles dans le local.
2. Ajouter un examen.
3. Encoder l'heure de début d'un examen. Cet encodage échouera s'il n'y a pas d'étudiants inscrits, s'il provoque des conflits horaires ou si un examen du même bloc est déjà placé le même jour. On peut utiliser cette fonction pour modifier l'heure de début d'un examen uniquement si aucune réservation de local n'a encore été effectuée pour cet examen.

4. Réserver un local pour un examen. On ne peut bien sûr pas réserver deux fois le même local pour un examen. La réservation échouera dans les cas suivants :
 - Si l'examen se déroule sur machine et qu'il n'y a pas de machines disponibles 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 déjà complètement réservé.
 - S'il existe déjà une réservation pour un autre examen dans le local à ce moment-là.
 - Si l'heure de début de l'examen n'a pas encore été fixée.
5. Visualiser l'horaire d'examen pour un bloc particulier. Les examens devront être triés par ordre chronologique. Pour chaque examen, on affiche son heure de début, son code, son nom et le nombre de locaux réservés. Si un examen n'a pas encore d'heure de début ou n'a pas encore de locaux réservés, il doit quand même apparaître dans la liste.
6. Visualiser toutes les réservations d'examen d'un local particulier. Les examens devront être triés par ordre chronologique. Pour chaque examen, on affiche son heure de début, son code et son nom.
7. Visualiser tous les examens qui ne sont pas encore complètement réservés (triés par code).
8. Visualiser le nombre d'examens pas encore complètement réservés pour chaque bloc. Le nombre d'examens non encore complètement réservés par bloc doit être précalculé.

Application utilisateur

Cette application offrira un premier menu permettant d'inscrire et de connecter un utilisateur :

1. Inscription d'un utilisateur : l'utilisateur doit entrer son email, ainsi qu'un nom d'utilisateur et un mot de passe. Il doit également encoder son bloc.
2. Connexion d'un utilisateur via son nom d'utilisateur et son mot de passe.

Une fois connecté, l'utilisateur accèdera à un menu qui proposera quatre fonctionnalités :

1. Visualiser les examens. Pour chaque examen, on affichera son code, son nom, son bloc et sa durée.
2. S'inscrire à un examen. L'inscription échouera si la date de l'examen a déjà été fixée.
3. S'inscrire à tous les examens du bloc de l'étudiant. Si une des inscriptions échoue, alors aucune inscription ne sera enregistrée.
4. Voir son horaire d'examen. Les examens devront être triés par ordre chronologique. Pour chaque examen, on affiche son code, son nom, son bloc, son heure de début, son heure de fin, et ses locaux séparés par des + (ex : A025+A026). Si un examen n'a pas d'heure de début ou n'a pas encore de locaux réservés, il doit quand même apparaître dans la liste.

Contraintes

1. Utilisez en priorité les capacités de la DB là où elles sont prévues pour être utilisées. Ainsi il est préférable d'utiliser une contrainte d'intégrité plutôt qu'un test dans une procédure stockée. Il est préférable de placer le business au niveau des procédures stockées plutôt qu'au niveau du Java quand cela est possible, etc.
2. La partie Java doit être aussi simple que possible.
 - a. Par besoin d'une IHM, une simple console suffira : `scanner.nextXXX()` et `System.out.println()` .
 - b. 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.
3. Vous travaillez par paire et vous avez donc accès à deux bases de données différentes sur le même serveur (`dbu2binXXX` et `dbu2binYYY`). Pour ce projet, vous n'utiliserez qu'une seule de ces deux bases de données, à vous de choisir laquelle. Le login/password de la personne qui possède cette base de données sera utilisée par l'interface administrateur. Le login/password de l'autre personne (celui qui ne possède donc pas la base de données) sera utilisé par l'application utilisateur. Il faudra donc donner les droits nécessaires (**GRANT**) à ce login pour que cette seconde application puisse fonctionner correctement.
4. Attention à la sécurité.
 - a. **GRANT** adéquats.
 - b. Injection de SQL.
 - c. Stockage du mot de passe adéquat.

Délivrables

1. Pour le vendredi 13 novembre à 23h59 : Remettre sur moodle sous format PDF le DSD correspondant à l'énoncé. *A la lecture seule du DSD*, les propriétés et associations doivent être compréhensibles. Donnez un nom aux associations : vous pouvez soit le placer sur celles-ci soit le donner via le nom de la clé étrangère. Respectez les conventions du cours de Conception.
2. Pour le vendredi 4 décembre à 23h59 : Remettre sur moodle un fichier archive au format ZIP contenant :
 - a. un fichier `projet.sql` qui contient l'ensemble des commandes SQL à envoyer au serveur PostgreSQL ;
 - b. la copie de vos projets Java ;
 - c. si nécessaire, un pdf avec une explication des points/choix que vous estimez devoir expliquer dans votre solution ;
3. Le vendredi 11 décembre, une démonstration sera organisée sur les machines de l'institut pour chaque groupe. Des informations sur cette démonstration vous seront communiquées plus tard dans le semestre.

Le plagiat sera lourdement sanctionné (au minimum 0 à l'UE). Tous les projets seront automatiquement testés par un logiciel de détection de plagiat.