École Polytechnique de Montréal
Département Génie Informatique et Génie Logiciel
INF3710 – Fichiers et Bases de données

Travail Pratique N° 4
Contraintes d'intégrité, déclencheurs et utilisation de l'API JDBC

1. Informations générales

Session	Automne 2016
Public cible	Étudiants de 1er cycle
Date de remise	Groupe 2 : samedi 12 novembre 2016 Groupe 1 : samedi 19 novembre 2016
Équipe de	2 étudiants
Pondération	8%
Directives particulières	1. Tout retard dans la remise du compte-rendu entraîne automatiquement l'attribution de la note de zéro sur vingt (0/20) aux étudiants concernés. 2. Aucun compte-rendu ne sera corrigé, s'il est soumis par une équipe dont la taille est différente de deux (2) étudiants sans l'approbation préalable du chargé de laboratoire. La note de zéro sur vingt (0/20) sera attribuée aux étudiants concernés. 3. Soumission du compte rendu (au format PDF ou word) par **moodle** uniquement (https://moodle.polymtl.ca). 4. Aucune soumission "hors **moodle**" ne sera corrigée. La note de zéro sur vingt (0/20) sera attribuée aux étudiants concernés.

2. Objectifs du laboratoire

Le but de ce travail est de permettre aux étudiants :

- de se familiariser avec les extensions SQL d'un langage de programmation (JDBC) ainsi qu'avec leur mise en œuvre et leur utilisation, et
- de réaliser une application en tenant compte de la gestion de transactions, des contraintes d'intégrité et de l'utilisation des mécanismes déclencheurs (*triggers* en anglais).

3. Travail à faire

Pour ce TP, vous utiliserez la BD **Inscriptions** décrite dans le TP 3. Vous devrez analyser des contraintes d'intégrité et écrire des triggers. Vous aurez aussi à développer en utilisant JDBC des méthodes permettant de réaliser divers traitements.

3.1. Première partie

Dans cette partie, il vous est demandé de mettre en place les déclencheurs permettant d'implémenter des contraintes. On considère les propriétés ci-dessous (contraintes d´intégrité).

A. Le responsable d'un cours doit toujours être un professeur.
B. Pour une personne, l'attribut «typPers» doit correspondre au type de la personne (P pour professeur, E pour étudiant, X pour employé).
    1. Un professeur (idem pour un employé) doit posséder un «pmatricule». Celui-ci est spécifié selon le format «p[[:digit:]]{3,6}». (lettre «p» suivie de 3 à 6 chiffres).
    2. Un étudiant doit posséder un « matricule » spécifié selon le format «[[:digit:]]{3,7}»
C. L'effectif d'une section de cours ne pas doit dépasser la capacité prévue.
D. Pour être inscrit à un cours, un étudiant doit posséder les pré-requis de ce cours.

Tâche A. Indiquez quelles actions risquent de compromettre la contrainte A (une *action* est ici une insertion, une mise à jour ou une suppression réalisée sur une table). Implantez un ou plusieurs triggers permettant de garantir le respect de la contrainte. Testez les différents cas possibles, montrant en particulier que vos triggers remplissent effectivement leur rôle.

Tâche B. Implantez un ou plusieurs triggers permettant de garantir le respect de la contrainte B.

Tâche C. Implantez un ou plusieurs triggers permettant de garantir le respect de la contrainte C.

Tâche D. Implantez un ou plusieurs triggers permettant de garantir le respect de la contrainte D.

Tâche E. Implantez un trigger qui se déclenche lorsque la moyenne d'un étudiant est inscrite ou modifiée (colonne «cumulatif» dans la table Inscription). Le trigger commence par vérifier que les pondérations des devoirs sont correctes (que leur somme vaut 100%). Puis, il calcule la moyenne pondérée des notes et refuse l'inscription de la moyenne si celle-ci est incorrecte.

Notez qu'un trigger peut être activé ou désactivé. Au besoin, vous pouvez par ailleurs supprimer les données précédemment enregistrées dans les différentes tables et les rajouter après la mise en place des déclencheurs.

3.2 Deuxième partie

On vous demande d'écrire, en exploitant l'API JDBC, une application permettant à un étudiant de réaliser son choix de cours et de le consulter ultérieurement. Lorsqu'on lance l'application, le programme invite l'utilisateur (un étudiant) à rentrer son matricule (aucun mot de passe n'est demandé), puis affiche un menu permettant de choisir parmi les 4 options ci-dessous.

1. *Affichage du choix de cours courant*

Le programme affiche le sigle, le titre, le numéro de section, ainsi que le responsable du cours, ceci pour chacun des cours du choix de cours.

2. *Suppression d'un cours*

L'utilisateur saisit le sigle d'un cours. Ce cours est alors supprimé du choix de cours.

3. *Ajout d'un cours*

Le programme affiche les cours auxquels l'étudiant peut s'inscrire. L'utilisateur est invité à choisir chacun des cours auxquels il veut s'inscrire en spécifiant le sigle du cours et le numéro de la section qu'il choisit.

4. *Validation*

Le programme vérifie si l'étudiant a bien choisi entre 9 et 15 crédits. Dans le cas contraire, un message s'affiche, la validation n'est pas effectuée et les choix réalisés ne seront donc pas pris en compte.

L'utilisateur doit toujours être informé du résultat de chacune de ses actions : si elle a été acceptée et, si ce n'est pas le cas, pourquoi. Vous pouvez vous contenter, si vous le souhaitez, d'une interface utilisateur sommaire en mode texte. Dans votre rapport, vous indiquerez (et justifierez au besoin) les différents contrôles effectués par le programme.

4. Livrable

Le livrable à soumettre est une archive .zip ou .rar dont le nom est formé des numéros de matricules des deux membres de l'équipe, séparés par un trait de soulignement (_). Il doit comporter les éléments suivants :

a) Les scripts des déclencheurs, pour la première partie : fichier au format .txt ou .sql.
b) Un fichier contenant les explications et la description des tests, pour la tâche A : fichier .pdf.
c) Le code java, pour la deuxième partie : fichier .java.

Les fichiers (a) et (c) doivent également être documentés en vue de faciliter la lecture et la compréhension des scripts et du code. Vous devez en outre indiquer dans votre fichier votre nom d'utilisateur (sans le mot de passe) afin que votre travail puisse être testé.

5. Évaluation

Rubrique	**Points**
Première partie	8
Deuxième partie	12

Le professeur : Philippe Galinier