

INF3180 - FICHIERS ET BASES DE DONNÉES
ÉNONCÉ DU TP2 – Automne 2013
Définition des données en SQL et PL/SQL d'Oracle avec tests (25%)

Date de remise : 4 décembre 2013 à 17h au plus tard

Lieu de remise : Chute à courrier du département d'informatique (PK-4150).

Autres précisions : Le rapport de TP doit être mis dans une enveloppe identifiée par le groupe-cours (sigle de cours, numéro de groupe et session), le nom du destinataire et le nom de l'expéditeur et le code permanent des membres de l'équipe.

Le projet donnera lieu à la remise d'un rapport de TP et à la création de deux fichiers (TP2.sql et TP2.out) qui seront envoyés par courriel **sur demande**. Le rapport contiendra une page d'identification (Identification du TP, groupe-cours, vos noms et codes permanents) et le contenu de TP2.out, résultat de l'exécution du script SQL TP2.sql.

Le fichier TP2.sql est un « script SQL*Plus » destiné à être exécuté en ligne avec l'outil SQL*Plus. NB Faites la commande SQL*Plus «SET ECHO ON» (au début du fichier TP2.sql ou avant son exécution) pour visualiser les énoncés SQL avec le résultat dans la sortie. Faites la commande SET SERVEROUTPUT ON pour voir le résultat des procédures du paquetage DBMS_OUTPUT. Vous pouvez diriger l'output du script TP2.sql dans un fichier texte à l'aide de la commande SQL*Plus « SPOOL *nomFichier* ». Faites la commande SPOOL OFF pour arrêter le processus de direction de l'output et fermer le fichier.

Vous devez respecter scrupuleusement ces directives. Le travail doit être réalisé en équipe de trois étudiants au plus.

Le script SQL*plus (fichier TP2.sql) doit contenir les éléments suivants :

1. Schéma DossierEtudiantModifié : (1,5 point par contrainte C1 à C6)

En prenant comme point de départ le script SQL du dossier étudiant UQAM ([ScriptDossierEtudiantUQAMPourTP2.sql](#)), implémentez les contraintes d'intégrité suivantes en modifiant le schéma.

C1 : Un *codeProfesseur* est formé de quatre lettres suivies de 1 chiffre (e.g. TREJ4)

(NB Vous pouvez employer la fonction PL/SQL REGEXP_LIKE permet de faire de l'appariement d'expressions régulières)

C2 : La note est un entier entre 0 et 100 si elle n'est pas nulle

C3 : Si elle est non nulle, la date d'abandon ne peut précéder la date d'inscription

C4 : Si la date d'abandon est non nulle, la note doit être nulle

C5 : Lorsqu'un groupe est supprimé, toutes ses inscriptions sont automatiquement supprimées

C6 : Il est interdit de faire diminuer la valeur de *maxInscriptions* de plus de 10% lors d'une mise à jour.

Documentez clairement comment chacune des contraintes est réalisée en faisant référence aux identificateurs de contrainte (C1, C2, ...).

2. Insertion des données et tests des contraintes C1 à C6

Après l'insertion des données du script initial, ajoutez les tests suivants pour chacune des contraintes C1 à C6.

NB Les tests sont obligatoires pour obtenir tous les points

NB Faites un ROLLBACK pour annuler la mise à jour après un test

SQL> PROMPT Test de violation de la contrainte C1

Après l'insertion des données du script initial, ajoutez les tests suivants pour valider les contraintes C1 à C6.

NB Les tests sont obligatoires pour obtenir tous les points

NB Faites un ROLLBACK pour annuler la mise à jour après un test

SQL> PROMPT Test de violation de la contrainte C1

Test de violation de la contrainte C1

SQL> INSERT INTO Professeur

2 VALUES('ULLJT','Ullman','Jeffrey')

3 /

INSERT INTO Professeur

*

ERREUR à la ligne 1 :

ORA-02290: violation de contraintes (GODIN.C1FORMATCODEPROF) de vérification

SQL> PROMPT Test de violation de la contrainte C2

Test de violation de la contrainte C2

SQL> UPDATE Inscription

2 SET note = 150

3 WHERE codePermanent = 'TREJ18088001' AND sigle = 'INF1110' AND noGroupe = 20 AND codeSession = 32003

4 /

UPDATE Inscription

*

ERREUR à la ligne 1 :

ORA-02290: violation de contraintes (GODIN.C2NOTEENTRE0ET100) de vérification

SQL> PROMPT Test de violation de la contrainte C3

Test de violation de la contrainte C3

SQL> UPDATE Inscription

2 SET dateAbandon = '15/08/2003'

3 WHERE codePermanent = 'VANV05127201' AND sigle = 'INF3180' AND noGroupe = 30 AND codeSession = 32003

4 /

UPDATE Inscription

*

ERREUR à la ligne 1 :

ORA-02290: violation de contraintes (GODIN.C3DATEABANDONAPRESINSCRIPTION) de vérification

SQL> PROMPT Test de violation de la contrainte C4

Test de violation de la contrainte C4

SQL> UPDATE Inscription

2 SET dateAbandon = '17/08/2003'

3 WHERE codePermanent = 'TREJ18088001' AND sigle = 'INF1110' AND noGroupe = 20 AND codeSession = 32003

4 /

UPDATE Inscription

*

ERREUR à la ligne 1 :

ORA-02290: violation de contraintes (GODIN.C4DATEABANDONNOTE) de vérification

SQL> PROMPT Test de la contrainte C5

Test de la contrainte C5

SQL> SELECT * FROM Inscription

2 WHERE sigle = 'INF5180' AND noGroupe = 40 AND codeSession = 12004

3 /

CODEPERMANEN	SIGLE	NOGROUPE	CODESESSION	DATEINSCRI	DATEABANDO	NOTE
--------------	-------	----------	-------------	------------	------------	------

EMEK10106501	INF5180	40	12004	19/12/2003		80
--------------	---------	----	-------	------------	--	----

SQL>DELETE FROM GroupeCours

```
SQL> DELETE FROM GroupeCours
  2  WHERE sigle = 'INF5180' AND noGroupe = 40 AND codeSession = 12004
  3  /
```

1 ligne supprimée.

```
SQL> SELECT * FROM Inscription
  2  WHERE sigle = 'INF5180' AND noGroupe = 40 AND codeSession = 12004
  3  /
```

aucune ligne sélectionnée

```
SQL> ROLLBACK
  2  /
```

Annulation (rollback) effectuée.

```
SQL> PROMPT Test de violation de la contrainte C6
Test de violation de la contrainte C6
SQL> UPDATE GroupeCours
  2  SET maxInscriptions = maxInscriptions-20
  3  WHERE sigle = 'INF1110' AND noGroupe = 20 AND codeSession = 32003
  4  /
SET maxInscriptions = maxInscriptions-20
*
ERREUR à la ligne 2 :
ORA-20102: Diminution de plus de 10% de maxInscriptions est interdite
ORA-06512: à "GODIN.BUINSCRIPTIONC6", ligne 2
ORA-04088: erreur lors d'exécution du déclencheur 'GODIN.BUINSCRIPTIONC6'
```

3. Ajout de la colonne *nbInscriptions* à la table *GroupeCours* (1 point pour l'ajout)

Ensuite dans votre script, ajoutez à la table *GroupeCours* une colonne *nbInscriptions* du type INTEGER (avec ALTER TABLE). Et implémentez une contrainte (C7) qui empêche que la valeur soit négative. (1 point pour C7)

Écrivez une fonction PL/SQL appelée *fNbInscriptions* qui prend en paramètres un *sigle*, un *noGroupe* et un *codeSession* et retourne le nombre d'étudiants actuellement inscrits (excluant les abandons) : (2 points pour la fonction)

Faites l'essai suivant :

```
SQL> select fNbInscriptions('INF1110',20,32003) from dual
  2  /
```

```
FNBINSCRIPTIONS('INF1110',20,32003)
```

```
-----
```

4

Dans un UPDATE mettez à jour la colonne *nbInscriptions* de la table *GroupeCours* en vous servant de la fonction *fNbInscriptions* (1 point pour la mise à jour)

Affichez la table *GroupeCours* ainsi mise à jour. Le résultat devrait être le suivant :

```
SQL> SELECT * FROM GroupeCours
  2  /
```

SIGLE	NOGROUPE	CODESESSION	MAXINSCRIPTIONS	CODEP	NBINSCRIPTIONS
INF1110	20	32003	100	TREJ4	4

INF1110	20	32003	100	TREJ4	4
INF1110	30	32003	100	PASB1	1
INF1130	10	32003	100	PASB1	2
INF1130	30	32003	100	GALE9	1
INF2110	10	32003	100	TREJ4	0
INF3123	20	32003	50	GOLA1	0
INF3123	30	32003	50	GOLA1	0
INF3180	30	32003	50	DEVL2	2
INF3180	40	32003	50	DEVL2	2
INF5180	10	32003	50	KNUD1	0
INF5180	40	32003	50	KNUD1	0

SIGLE	NOGROUPE	CODESESSION	MAXINSCRIPTIONS	CODEP	NBINSCRIPTIONS
INF1110	20	12004	100	TREJ4	0
INF1110	30	12004	100	TREJ4	0
INF2110	10	12004	100	PASB1	2
INF2110	40	12004	100	PASB1	2
INF3123	20	12004	50	GOLA1	0
INF3123	30	12004	50	GOLA1	0
INF3180	10	12004	50	DEVL2	1
INF3180	30	12004	50	DEVL2	0
INF5180	10	12004	50	DEVL2	2
INF5180	40	12004	50	GALE9	1

21 ligne(s) sélectionnée(s).

4. Création d'une procédure PL/SQL *pTacheEnseignement* (5 points pour la procédure)

Créez une procédure *pTacheEnseignement* qui affiche la tâche d'enseignement d'un professeur. L'affichage est effectué avec le paquetage DBMS_OUTPUT() selon le format de l'exemple suivant. La procédure prend en entrée un *codeProfesseur*. Faites l'essai suivant.

```
SQL> EXECUTE pTacheEnseignement('TREJ4')
code professeur :TREJ4
```

```
nom:Tremblay
```

```
prenom:Jean
```

```
sigle   noGroupe  session
```

```
INF1110 20        32003
```

sigle	noGroupe	session
INF1110	20	32003
INF2110	10	32003
INF1110	20	12004
INF1110	30	12004

Procédure PL/SQL terminée avec succès.

5. Mise à jour automatique de *nbInscriptions* (2 points pour le TRIGGER)

Créer un TRIGGER qui incrémente le *nbInscriptions* de 1 lors d'une insertion correspondante dans la table *Inscription*. Faites l'essai suivant suivi d'un ROLLBACK.

```
SQL> INSERT INTO Inscription
  2  VALUES('TREY09087501','INF5180',10,12004, '19/12/2003',null,80)
  3  /
```

1 ligne créée.

```
SQL> SELECT * FROM GroupeCours
  2  WHERE sigle = 'INF5180' AND noGroupe = 10 AND codeSession = 12004
  3  /
```

SIGLE	NOGROUPE	CODESESSION	MAXINSCRIPTIONS	CODEP	NBINSCRIPTIONS
INF5180	10	12004	50	DEVL2	3

```
SQL> ROLLBACK
  2  /
```

Annulation (rollback) effectuée.

6. Création d'une vue (2 points pour la vue et 2 points pour le TRIGGER)

Créer une vue *MoyenneParGroupe* qui affiche le sigle de cours, le numéro de groupe, le code de la session et la moyenne des notes du groupe correspondant.

Afficher la vue avec un SELECT * FROM MoyenneParGroupe.

Créer un TRIGGER INSTEAD OF UPDATE qui permet de modifier (UPDATE) la vue. Un changement de la moyenne du groupe doit mettre à jour chacune des notes du groupe selon la différence entre la moyenne actuelle et la nouvelle moyenne pour les cas de note non nulle. Illustrer par le test de mise à jour suivant de la moyenne du cours INF1130-10 de la session 32003 suivi de l'affichage de la vue *MoyenneParGroupe* pour le groupe modifié et des lignes de la table *Inscription* modifiées.

Exemple :

```
SQL> SELECT * FROM MoyenneParGroupe
  2  /
```

SIGLE	NOGROUPE	CODESESSION	MOYENNENOTE
INF1130	10	32003	10

SIGLE	NOGROUPE	CODESESSION	MOYENNENOTE
-------	----------	-------------	-------------

INF1130	10	32003	80
INF3180	30	32003	75
INF2110	40	12004	80
INF1110	30	32003	90
INF3180	10	12004	90
INF2110	10	12004	85
INF5180	40	12004	80
INF1130	30	32003	80
INF1110	20	32003	77,5
INF3180	40	32003	75
INF5180	10	12004	85

11 ligne(s) sélectionnée(s).

SQL>

SQL> CREATE OR REPLACE TRIGGER InsteadUpdateMoyenneParGroupe

...

14 /

Déclencheur créé.

SQL>

SQL> UPDATE MoyenneParGroupe

2 SET moyenneNote = 70

3 WHERE sigle = 'INF1130'AND noGroupe = 10 AND codeSession = 32003

4 /

1 ligne mise à jour.

SQL> SELECT * FROM MoyenneParGroupe

2 WHERE sigle = 'INF1130'AND noGroupe = 10 AND codeSession = 32003

3 /

SIGLE	NOGROUPE	CODESESSION	MOYENNENOTE
-------	----------	-------------	-------------

INF1130	10	32003	70
---------	----	-------	----

SQL> SELECT * FROM Inscription

2 WHERE sigle = 'INF1130'AND noGroupe = 10 AND codeSession = 32003

3 /

CODEPERMANEN	SIGLE	NOGROUPE	CODESESSION	DATEINSCRI	DATEABANDO	NOTE
--------------	-------	----------	-------------	------------	------------	------

TREJ18088001	INF1130	10	32003	16/08/2003		60
MARA25087501	INF1130	10	32003	22/08/2003		80

