

Travail de conception et d'exploitation d'une base de données (INF3180 – ÉTÉ 2017) TP 2

Le travail consiste à faire la conception, la construction et la mise en œuvre partielle d'une application de base de données avec le SGBD Oracle. Le travail doit se faire en **équipe de deux à quatre (les groupes d'un étudiant seront acceptés exceptionnellement!)**. Les détails sur les environnements de travail Oracle ont été présentés dans vos séances de laboratoire.

À REMETTRE DANS UNE ENVELOPPE FERMÉE (identifiée avec le sigle du cours, le groupe, le nom du professeur et les noms des membres de l'équipe)

- **ET SUR PAPIER**

- 1) Le document INF3180tp_template.doc dûment rempli

À REMETTRE SUR MOODLE

- **UN UNIQUE FICHIER ZIP INCLUANT**

- 1) Le document INF3180tp_template.doc dûment rempli
- 2) Tous les fichiers Oracle dans les répertoires « script/ » et « script/resultats/ »

Le non-respect des spécifications (taille des équipes, identification de l'enveloppe et du travail, Template du travail fourni, fichiers à fournir, etc.) du TP vous coûtera 10 points en moins par violation dans votre note finale.

La non-remise d'un des deux livrables (Document Papier ou Fichier zip sur Moodle) vous coûtera 50% de votre note finale.

La qualité du français constitue un critère d'évaluation (pour un maximum de 10%)

Les travaux remis en retard ne seront pas considérés.

Échéancier

Le 18 Juillet 2017 avant 13h00 sur Moodle et à la chute de courriers du département d'informatique au 4e étage du PK.

Références

Godin, R. (2006). *Systèmes de gestion de bases de données par l'exemple*. 2e édition, Montréal, Canada

Voici un texte délimitant la portée du système et rappelant les règles de base du domaine :

Une entreprise de consultation a fourni le schéma relationnel ci-dessous permettant de gérer les consultations entre des docteurs identifiés par leurs matricules d'employés (répertoire partagé des employés du centre médical) et des patients identifiés par leur numéro de dossier et associés à un docteur traitant.

DOSSIERPATIENT (**numDos**, nomP, prenomP, sexe, numAS, dateNaiss, dateC, matricule, nbrConsultation)

DOCTEUR (**matricule**, nomM, prenomM, specialite, ville, adresse, niveau, nbrPatients, nbrMoyenMedicaments)

CONSULTATION (**CodeDocteur**, **numDos**, **dateC**, diagnostic, numOrd)

ORDONNANCE (**numOrd**, recommandations, type, dateC, nbrMedicaments)

ORDONNANCECHIRURGIE (**numOrd**, **idChir**, rang)

CHIRURGIE (**idChir**, idType, idSalle, dateChirurgie, HeureDebut, HeureFin)

SALLE (**idSalle**, nom)

SPECIALISATIONSALE (**IdType**, **idSalle**, dateC)

TYPECHIRURGIE (**IdType**, nom, Description)

ORDONNANCEMEDICAMENTS (**numOrd**, **idMed**, nbBoites)

MEDICAMENT (**idMed**, nomMed, prix, categorie)

SPECIALITE (**code**, titre, description)

CATEGORIES (**IdCategorie**, nom, Description)

Le patient est identifié par son numDos numéro de dossier et son *numAS* numéro d'assurance maladie est unique et son docteur traitant est *matricule*. Le docteur est identifié par *matricule* et il peut avoir une spécialité. Un patient consulte un docteur à une certaine date. Lors d'une consultation, le docteur effectue un diagnostic et lui prescrit une ordonnance dont l'identifiant est *numOrd*. Une ligne de l'ordonnance du docteur '*numOrd*' prévoit de donner un *nbBoites* du médicament *idMed*. Le médicament est identifié, il a un certain prix et correspond à une catégorie de médicament. La date *dateC* représente la date de création. HeureDebut et HeureFin sont des entiers positifs (Exemple, 900 pour 9h AM et 1300 pour 1h PM).

Attention à bien définir les clés primaires (**Champs en gras**) et étrangères (Champs soulignés).

Voici les autres contraintes portant sur ces tables.

- ✓ Les données *nomP*, *prenomP*, *nomM*, *prenomM*, *nomMed*, *nom*, *titre*, et *diagnostic* doivent toujours être connues.
- ✓ Il ne peut pas y avoir deux chirurgies pour une même salle qui se chevauche dans la plage horaire.
- ✓ Toutes les valeurs (*nbBoites*, *prix*, *etc.*) ont une valeur par défaut à 0 et prennent des valeurs positives. Le champ *sexe* peut avoir uniquement les valeurs 'F' et 'M'.
- ✓ Il ne peut pas y avoir deux médicaments avec le même nom et la même catégorie.
- ✓ Les niveaux autorisés sont : Étudiant, Interne, ou Docteur et les types autorisés sont : Chirurgie ou Médicaments.
- ✓ Il ne peut pas y avoir deux chirurgies pour une même ordonnance avec le même rang.
- ✓ Le type de la salle opératoire doit correspondre au type de la chirurgie.
- ✓ Le détail de l'ordonnance (ORDONNANCECHIRURGIE ou ORDONNANCEMEDICAMENTS) doit correspondre au type d'ordonnance.
- ✓ Les *nbrPatients* (nombre de patients d'un docteur à titre de médecin traitant), *nbrMoyenMedicaments* (nombre moyen de médicaments prescrits par un docteur par prescription), *nbrConsultation* (nombre total de consultations pour un patient), *nbrMédicaments* (nombre de médicaments différents – pas les boîtes - pour une unique ordonnance) doivent être toujours mis à jour automatiquement.
- ✓ La suppression d'un docteur doit entraîner la suppression de ses consultations. La suppression d'un patient doit entraîner la suppression de ses consultations. La suppression ou la modification d'une ordonnance ou d'un médicament, référencés respectivement dans CONSULTATION ou ORDONNANCE, ne sont pas autorisées. La suppression d'un docteur doit entraîner la modification de ses patients en donnant la valeur nulle à la matricule.

Des demandes typiques auxquelles la base de données devra pouvoir répondre sont (**exemples de cas d'utilisation – Choix d'un cas par point**) :

- ✓ Afficher le nombre de chirurgies par docteur, afficher le nombre de chirurgies par salle, ou afficher le nombre de chirurgies par type.
- ✓ Afficher le nombre de consultations par docteurs ou afficher le nombre de consultations par spécialité.
- ✓ Afficher le nombre moyen de consultations par mois, afficher le nombre de consultations par année.
- ✓ Afficher le nombre de médicaments prescrits par docteur, afficher le nombre moyen de médicaments prescrits par mois, afficher le nombre de médicaments prescrits par docteur par année.
- ✓ Ratio moyen des chirurgies versus traitements médicamenteux par mois ou ratio des chirurgies versus traitements médicamenteux par années.

Travail demandé :

1. Code du schéma SQL (40pts)

Dans cette partie, vous devez utiliser le schéma relationnel, les exigences décrites dans la description et les contraintes d'intégrité pour produire le schéma en SQL correspondant. Vos tables doivent respecter toutes les contraintes définies dans l'énoncé pour la création de tables.

- Écrire le script SQL*plus pour la création des tables avec les contraintes d'intégrité CHECK, clé primaire et référentielle. Ces contraintes d'intégrité statiques doivent être déclarées au niveau de la table (pas par des trigger). Comme indiqué dans l'exemple ci-dessous, toutes ces contraintes doivent être identifiées par un nom.

Exemple :

```
CREATE TABLE NomTable
(attr1 INTEGER NOT NULL,
CONSTRAINT nomTab_pk PRIMARY KEY(attr1),
CONSTRAINT nomTab_fk FOREIGN KEY(attr1) REFERENCES
Table2(attr1),
CONSTRAINT nomTab_chk CHECK( attr1 >= 10)
)
```

- Écrire le script SQL*plus pour la création de triggers pour répondre à 3 des contraintes définies dans l'énoncé. Une contraintes peut avoir un à plusieurs triggers. Ces 3 contraintes ne doivent pas être des contraintes statiques pouvant être simplement implémentées dans la création des tables. Ces dernières (contraintes statiques simples) ne seront pas considérées dans l'évaluation de la partie des triggers.
- Écrire un script SQL*plus pour l'insertion de données de test. Le jeu de données, organisé sous une forme tabulaire correspondante au modèle *UML*, doit être fourni (voir section 4). Ce dernier (jeu de données) doit être assez varié pour pouvoir vérifier toutes les contraintes implémentées. Vous devriez insérer ces données dans la base et puis les utiliser pour vérifier vos contraintes d'intégrité et les requêtes de la section 3.
- Écrivez un script pour la suppression complète de votre schéma. Ce fichier doit inclure les commandes DROP pour la suppression des :
 - ✓ Tables,
 - ✓ Séquences,
 - ✓ Triggers,
 - ✓ etc.
- Écrivez un script SQL*plus qui teste le bon fonctionnement de vos triggers. Vous devez faire des tests montrant que les triggers ne permettent pas certaines insertions, modifications ou suppressions, faire les modifications nécessaires, et qu'ils vont afficher un message d'erreur indiquant le problème.

2. Liste des index (10pt)

Pour que les requêtes de la section 3 soient performantes, vous devez décider sur quels champs d'une table concernée il est utile d'avoir un index secondaire.

Pour cette partie, vous devez donner la liste des champs indexés dans le format suivant :

- NomTable1
 - ✓ nomIndex(nomChamp)
 - ✓ nomIndex(nomChamp)
- NomTable2
 - ✓ nomIndex(nomChamp)

Exemple : Si vous jugez nécessaire d'avoir des index secondaires pour les champs de la table A, alors la liste des index est le rectangle droit.

<pre>CREATE TABLE A (id INTEGER NOT NULL, login INTEGER NOT NULL, nom VARCHAR2(20), CONSTRAINT A_PK PRIMARY KEY(id), CONSTRAINT A_UN UNIQUE(login)); CREATE INDEX ind_nom ON A(nom);</pre>	<pre>A - A_PK(id) - A_UN(login) - ind_nom(nom)</pre>
--	--

Note :

Oracle insère, automatiquement, des index pour tout champ qui est PRIMARY KEY ou UNIQUE. L'ordre des champs dans une clé composée ou unique composée est important.

3. Code SQL des cas d'utilisation pour la conception physique (40 pts)

Pour 4 cas d'utilisation de votre choix (les exemples de cas d'utilisation sont fournis dans la description – Page 3 – **choix d'un cas par point**), donnez le code SQL*plus des requêtes SQL nécessaires pour sa réalisation et le résultat de l'exécution. Vous serez évalué sur la consistance de votre travail et l'optimalité de vos requêtes.

4. Jeu de données (10 pts)

Le jeu de données, organisé sous une forme tabulaire correspondante au modèle *UML*, doit être fourni (voir document gabarit fourni). Ce dernier (jeu de données) doit être assez varié pour pouvoir vérifier toutes les contraintes. Vous devriez insérer ces données dans la base et puis les utiliser pour vérifier vos contraintes d'intégrité et réaliser la section 3.