HAI502I - TRAVAUX PRATIQUES 3
MANIPULATION DE LA STRUCTURE D'UNE BASE
MISE A JOUR DES DONNEES – INDEX – VUES

Objectifs du TP :
L'objet de cette séance de TP est multiple :
Dans un premier temps il s'agit de prendre en compte différentes évolutions de l'univers réel modélisé par la BD BIBLIO, en effectuant diverses manipulations sur la structure de la base exemple (*intension de la BD*). Il s'agit, également, d'utiliser les opérations de mises à jour des données (*extension de la BD*).
Dans un second temps, il s'agit de consulter les tables systèmes (méta-base).
Enfin, nous nous intéressons aux index et aux vues. L'objectif est d'aborder certains aspects de la gestion de la sécurité d'une base de données. En fait, nous nous limitons au contrôle de la cohérence des données (intégrité) et à la gestion de l'accès à ces données (confidentialité).

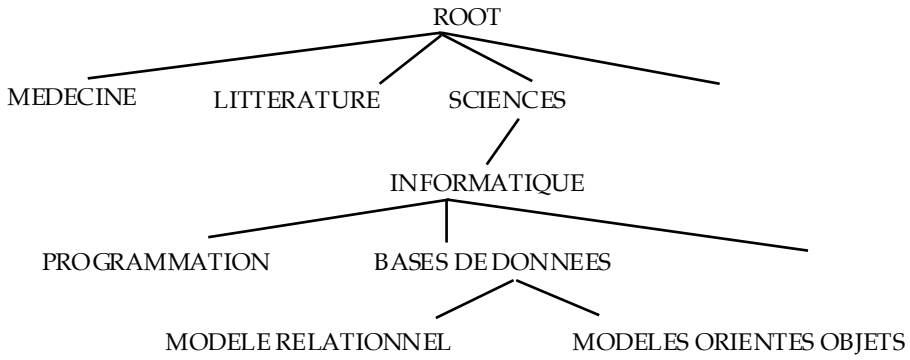
Récupérer, sous MOODLE, les fichiers suivants les sauvegarder dans un répertoire TP3 :

- `creationBIBLIO.sql`
- `remplissageBIBLIO.sql`
- `cleancreationBIBLIO.sql`
- `mesresultatsrequetesTP3.sql`
- `tuplesauteurs.sql`

Création et modification d'attributs

On désire pouvoir simuler un thesaurus et intégrer dans la BD BIBLIO de nouvelles données sur les abonnés.

Q1 *:* A partir de la relation MOT_CLEF, qui répertorie l'ensemble des termes pouvant indexer les différents ouvrages, on désire intégrer, dans la base, un mécanisme permettant de simuler un thesaurus. Un thesaurus est une hiérarchie de mots clefs décrivant les liens entre des termes génériques et des termes plus spécifiques. Par exemple, le thesaurus dans notre application pourrait être (partiellement) représenté par l'arborescence suivante :



Le principal intérêt de simuler un tel thesaurus et de permettre d'étendre ou au contraire de restreindre les recherches d'ouvrages par mots clefs. Par exemple, si un livre est seulement indexé par le mot clef "Modèle Relationnel", il devra pouvoir être sélectionné par un utilisateur recherchant tous les ouvrages traitant de "Bases de données" ou de "Sciences".

Modifier le schéma conceptuel de la base, donné en annexe, de manière à simuler ce thesaurus. Lire la question 2 avant de reporter les modifications sur le schéma relationnel.

Q2 : **ATTENTION, aucune contrainte d'intégrité n'a été spécifiée sur vos bases exemples !!!**
Dans le fichier `cleancreationBIBLIO.sql`, spécifier les différentes contraintes d'intégrité (clés primaires et étrangères) et prendre en compte les contraintes suivantes :

- Un livre doit forcément avoir un titre.
- Le siècle est entre 0 et 21.
- Un abonné a forcément un nom.
- L'âge d'un abonné est compris entre 0 et 120.
- L'état d'un livre ne peut prendre ses valeurs que dans ('BON', 'ABIME', 'EN_REPARATION').
- Le code prêt d'un livre est uniquement dans l'ensemble ('EXCLU','EMPRUNTABLE', 'CONSULTABLE').
- Le nombre de relances pour un emprunt est compris entre 1 et 3.

Remarques :

- Les données saisies dans la base actuellement vérifient les contraintes spécifiées.
- Penser à nommer ses contraintes.
- Attention, à la fin des modifications ou à chaque fois que vous allez modifier le fichier, il ne faut pas oublier de relancer les ordres de création de la base : `@ cleancreationBIBLIO.sql` ainsi que l'insertion des tuples : `@ remplissageBIBLIO.sql` pour que les modifications soient prises en compte.
- Bien entendu, il s'agit d'un TP et dans une situation réelle il n'est pas possible de relancer à chaque fois les créations de relations et les insertions de tuples.

Q3 : Reporter les modifications du schéma conceptuel de la question 1 sur votre schéma relationnel en utilisant le fichier `mesresultatsrequetesTP3.sql`. **Ne pas saisir de données pour l'instant cela se fera ultérieurement.**

Q4 : Le gestionnaire de la base de données vient de se rendre compte qu'il y avait eu une erreur lors de l'étape de conception et que l'attribut AGE a été utilisé plutôt que la date de naissance.

Outre leur date de naissance (ce qui permettra un calcul automatique de leur âge), il souhaite aussi ajouter leur type (parmi : {'ADULTE', 'ENFANT}) et leur catégorie dont les valeurs admissibles sont : {'REGULIER', 'OCCASIONNEL', 'A PROBLEME', 'EXCLU'}.
Après avoir choisi les types syntaxiques adaptés, dans le fichier `mesresultatsrequetesTP3.sql`, en utilisant le ALTER du Langage de Définition de Données de SQL, réaliser la création des nouveaux attributs DATE_NAI, TYPE_AB et CAT_AB dans la relation concernée en spécifiant une contrainte d'intégrité de domaine pour TYPE_AB et CAT_AB (donnez les valeurs admissibles en majuscules). Après avoir modifié et exécuté `@mesresultatsrequetesTP3.sql`, vérifier l'opération en consultant la structure de la relation par DESCRIBE (ou DESC). Contrôler la mise en œuvre des contraintes de domaine en tentant d'affecter, à l'aide de UPDATE, une valeur invalide aux attributs TYPE_AB et CAT_AB pour l'abonné (902043).
Remarques :

- Si vous vous êtes trompé il est conseillé de relancer `@cleancreationBIBLIO.sql`, `@remplissageBIBLIO.sql` et `@mesresultatsrequetesTP3.sql`.
- **Ne toujours pas saisir les données** (le UPDATE précédent a causé une erreur donc la donnée n'a pas été modifiée).

Q5 : On désire étendre la définition du type de l'attribut NOM dans la relation ABONNE, en permettant d'avoir des noms de 20 caractères. Consulter la définition de cet attribut puis effectuer dans `mesresultatsrequetesTP3.sql`, la modification correspondante et vérifier que celle-ci est bien prise en compte.

Création de relation

Q6 : Telle qu'elle est conçue, la BD exemple ne stocke aucune information sur les auteurs des ouvrages. Il faudrait en particulier connaître le nom et le prénom des auteurs ainsi que leur nationalité d'origine. Sachant qu'il existe des ouvrages collectifs mais aussi des écrits anonymes et des auteurs homonymes, effectuer les modifications nécessaires sur le schéma conceptuel donné en annexe, pour prendre en compte ces différentes informations.
Répercutez ces modifications sur le schéma relationnel (en annexe), en précisant, si besoin est, les clefs primaire et étrangère(s).

Modifier le fichier `mesresultatsrequetesTP3.sql` pour reporter les mises à jour du schéma relationnel et en spécifiant les différentes contraintes d'intégrité mises en jeu. Après avoir exécuté le fichier, vérifier que les modifications ont bien été reportées.
Remarque :

- en cas de gros problèmes d'incohérence, exécuter successivement `cleancreationBIBLIO.sql`, `@remplissageBIBLIO.sql` et `@mesresultatsrequetesTP3.sql`.

Attention : Après avoir exécuté dans l'ordre : `cleancreationBIBLIO.sql`, `@remplissageBIBLIO.sql` et `@mesresultatsrequetesTP3.sql`, la base est dans un

état incohérent car les tuples ne vérifient pas les modifications apportées dans le fichier `@mesresultatsrequetesTP3.sql` (changement de la taille de l'attribut nom, ajout de la date de naissance, etc.). **C'est tout à fait normal et l'objectif de la prochaine partie est justement de saisir ces données.**

Mise à jour des données

Sur la base re-structurée, on vous demande de réaliser les opérations de mise à jour des données suivantes.

Il vous est conseillé de paramétrer vos requêtes de mise à jour.
Rappel : Il est possible de faire saisir une valeur par l'utilisateur à l'aide du symbole &.
Par exemple :

SELECT * **FROM** PILOTE **WHERE** Plnum=&Plnum ;

Affichera :

Enter value for Plnum:

Pour une chaîne de caractères, mettre des ' ' :

SELECT * **FROM** PILOTE **WHERE** Plnom='&Plnom' ;

Affichera :

Enter value for Plnom:

Q7 : Dans le fichier `@mesresultatsrequetesTP3.sql` Effectuer les modifications nécessaires pour que les liens entre termes génériques et spécifiques du thesaurus donné en exemple précédemment soient pris en compte dans la base.

Q8 : Ne voulant pas demander aux abonnés leur date de naissance, le gestionnaire souhaite simplement calculer la date de naissance à partir de l'âge des personnes.
Pour connaître la date système :

SELECT SYSDATE FROM DUAL ;

retourne :

```
SYSDATE
----------
09-09-2019
```

Il est tout à fait possible de manipuler des dates en faisant des additions ou des soustractions. Par exemple :

SELECT SYSDATE, SYSDATE + 1, SYSDATE + 366, SYSDATE - 365 FROM DUAL;

retourne :

```
SYSDATE    SYSDATE+1  SYSDATE+36 SYSDATE-36
---------- ---------- ---------- ----------
09-09-2019 10-09-2019 09-09-2020 09-09-2018
```

Dans le fichier `@mesresultatsrequetesTP3.sql`, mettre à jour l'attribut DATE_NAI pour l'ensemble des abonnés par rapport à l'âge qu'ils ont dans la base. De plus, en tenant compte de la

date de naissance, mettre à jour l'attribut TYPE_AB pour tous les abonnés (un ADULTE a plus de 18 ans). Vérifier que les modifications ont bien été reportées.

Q9 : Ouvrir le fichier `tuplesauteurs.sql` et recopier l'ensemble des tuples dans le fichier `@mesresultatsrequetesTP3.sql`. Exécuter successivement `cleancreation BIBLIO.sql`, `@remplissageBIBLIO.sql` et `@mesresultatsrequetesTP3.sql`. La base est à présent dans un état cohérent.

Les tables systèmes

Effectuez les requêtes suivantes de consultation des tables systèmes (ALL_TABLES, USER_TABLES, DBA_TABLES, USER_CONSTRAINTS, USER_TAB_COLUMNS, USER_INDEXES…). Faire un DESC *nom_de_la_table* pour connaître ses attributs.

Q10 : Quels sont les différents propriétaires (OWNER) des tables ?

Q11 : Quels sont les noms de tous les attributs de la relation ABONNE ?

Q12 : Quel est le nombre d'attributs définis dans la BD BIBLIO ?

Q13 : Quelles sont toutes les contraintes d'intégrité de clé primaire de la base BIBLIO ?
Remarque : normalement les contraintes de clé primaire ont été nommées à l'aide de PK_. Vous voyez ici l'intérêt de nommer ses contraintes.

Prise en compte de l'intégrité de relation par création d'index

*Les index, créés dans un souci d'optimisation des requêtes, ont également un intérêt lorsqu'ils sont primaires, dans la prise en compte **effective** des contraintes d'intégrité de relation. C'est ce dernier aspect qui est illustré dans cette étape.*

Q14 : Comment, dans SQLPLUS, assurer la vérification automatique de l'intégrité d'une relation ? Créer la relation suivante : TESTINDEX(NOM) où NOM est de type VARCHAR(10). Ne pas définir la clé primaire de la relation. Créer un index unique CLE_TESTINDEX sur TESTINDEX. Vérifier la table système concernée pour contrôler la définition de l'index créé. Enfin, vérifier la prise en compte de l'index en tentant d'insérer une valeur de clef dupliquée.
Remarque : lors de la création d'une clé primaire il y a automatiquement création d'un index. Ici nous montrons comment cet index est créé.

Q15 : Existe-t-il, dans la base BIBLIO, une relation pour laquelle aucun index n'a été spécifié ?

Création et manipulation de vues

Les vues à créer ont un double objectif : assurer la prise en compte des contraintes d'intégrité de la base et également limiter la consultation des données pour certains utilisateurs.

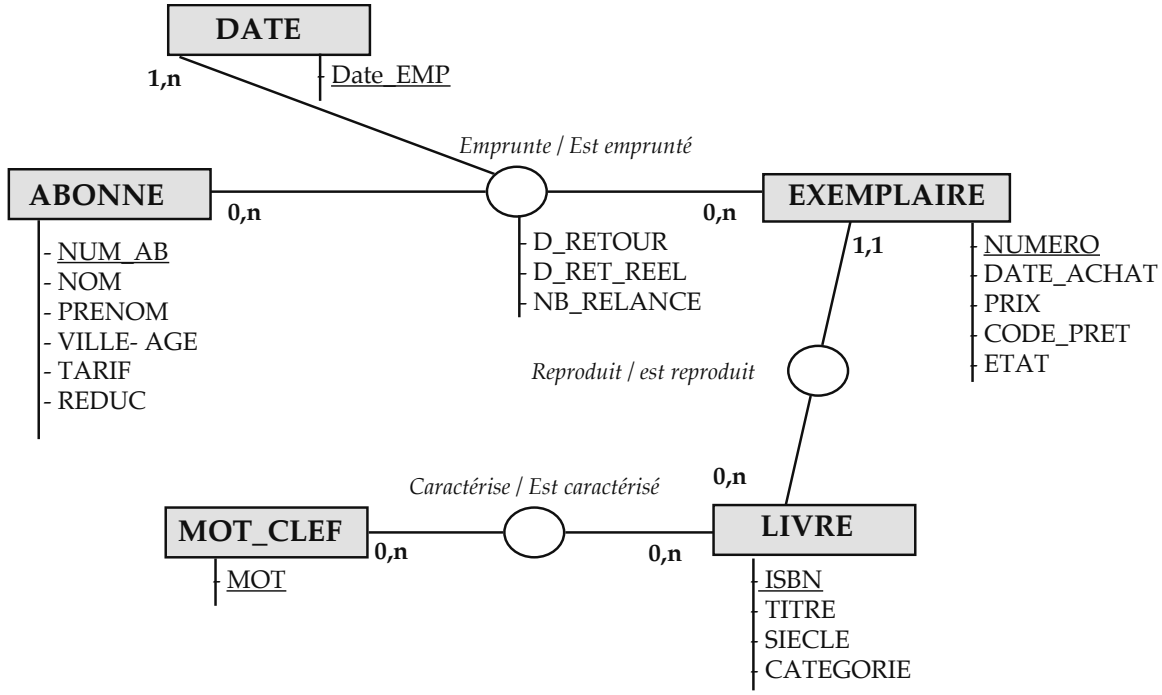
Q16 : Créer une vue ABONNE_MONTP donnant le numéro, le nom et le prénom des abonnés habitant Montpellier. Puis réaliser l'insertion, **à travers la vue créée**, d'un nouvel abonné montpelliérain. Vérifier ensuite la propagation de la mise à jour en consultant les tuples de la relation ABONNE.

Q17 : Vérifier que la vue a bien été créée dans les tables systèmes.

Q18 : Il s'agit de prendre en compte l'intégrité de domaine de l'attribut ETAT dans la relation EXEMPLAIRE. En effet, ces valeurs admissibles ne peuvent appartenir qu'à l'ensemble suivant : {'BON', 'PERDU', 'ABIME', 'EN PREPARATION'}. Créer une vue ETAT_EXEMPLAIRE, permettant d'assurer que toute insertion à travers cette vue vérifie la contrainte de domaine énoncée. Vérifier l'opération réalisée en tentant une insertion invalide.

Q19 : Définir la vue nécessaire pour prendre en compte toutes les contraintes d'intégrité de référence mises en jeu lors d'une insertion dans la relation EMPRUNT ?

ANNEXE :



ABONNE (NUM_AB, NOM, PRENOM, VILLE, AGE, TARIF, REDUC)
EXEMPLAIRE (NUMERO, DATE_ACHAT, PRIX, CODE_PRET, ETAT, #*ISBN*)
LIVRE (ISBN, TITRE, SIECLE, CATEGORIE)
MOT_CLEF (MOT)
EMPRUNT (#*NUM_AB*, #*NUM_EX*, D_EMPRUNT, D_RETOUR, D_RET_REEL, NB_RELANCE)
CARACTERISE (#*ISBN*, #*MOT*)