



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

(ATTENTION CE TP EST SUR DEUX SEANCES)

Objectifs du TP :

L'objet de cette deuxième 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).

Rappels :

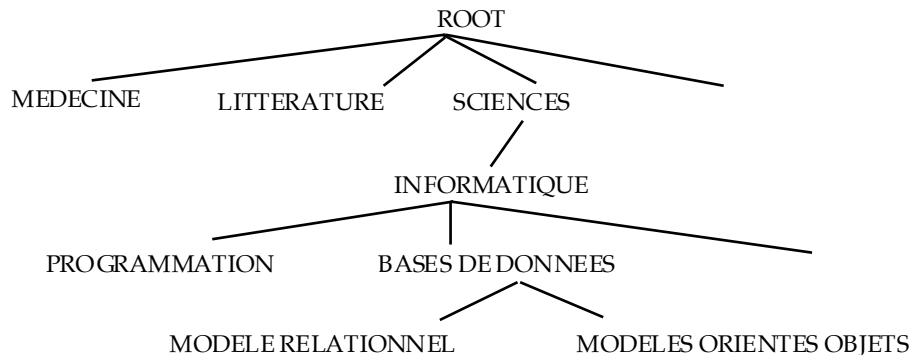
La création de la base et les différents tuples sont disponibles sous Moodle dans TP2 :

- `creationBIBLIO.sql`
- `remplissageBIBLIO.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".

Modifiez 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 !!!

Faire une copie du fichier `creationBIBLIO.sql` en `cleancreationBIBLIO.sql`. Spécifier dans ce dernier les différentes contraintes d'intégrité indispensables. Il faut au préalable consulter la définition des attributs compatibles. 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.

Remarque : 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 : Reportez les modifications du schéma conceptuel de la question 1 sur votre schéma relationnel. Vous pouvez pour cela utiliser le fichier `cleancreationBIBLIO.sql`. Ne pas saisir de données.

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, effectuez, **sans modifier le fichier `cleancreationBIBLIO.sql`**, 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). Vérifiez l'opération en consultant la structure de cette relation par DESCRIBE (ou DESC). Puis contrôlez la mise en œuvre des contraintes de domaine en tentant d'affecter une valeur invalide aux attributs TYPE_AB et CAT_AB pour un abonné existant.



Q5 : On désire étendre la définition du type de l'attribut TARIF dans la relation ABONNE, en augmentant la taille des valeurs possibles de 2 positions. Consultez la définition de cet attribut puis effectuez, **sans modifier le fichier cleancreationBIBLIO.sql**, sur la base la modification correspondante et vérifiez-la.

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, effectuez 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).

Réalisez ces mises à jour structurelles sur la BD exemple, en spécifiant les différentes contraintes d'intégrité mises en jeu. Vérifiez vos manipulations.

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 : Effectuez 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+366	SYSDATE-36
-----	-----	-----	-----
09-09-2019	10-09-2019	09-09-2020	09-09-2018

Mettez 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, mettez à jour l'attribut TYPE_AB pour tous les abonnés (un ADULTE a plus de 18 ans).

Q9 : Effectuez quelques insertions de tuples :

```
INSERT INTO AUTEUR VALUES (101, 'DUMAS', 'ALEXANDRE', 'FRANCAISE');
INSERT INTO AUTEUR VALUES (102, 'SARTRE', 'JEAN-PAUL', 'FRANCAISE');
INSERT INTO AUTEUR VALUES (103, 'GENEY', 'JEAN', 'FRANCAISE');
INSERT INTO AUTEUR VALUES (104, 'VALLES', 'JULES', 'FRANCAISE');
INSERT INTO AUTEUR VALUES (105, 'VILLON', 'FRANCOIS', 'FRANCAISE');
INSERT INTO AUTEUR VALUES (106, 'ECO', 'UMBERTO', 'ITALIENNE');
INSERT INTO AUTEUR VALUES (107, 'GARY', 'ROMAIN', 'FRANCAISE');
INSERT INTO AUTEUR VALUES (108, 'ROCHEFORT', 'CHRISTIANE', 'FRANCAISE');
INSERT INTO AUTEUR VALUES (109, 'STEINBECK', 'JOHN', 'AMERICAIN');
INSERT INTO AUTEUR VALUES (110, 'HOFSTADTER', 'DOUGLAS', 'ALLEMAND');
INSERT INTO AUTEUR VALUES (111, 'BOUZEGHOUB', 'MOKRANE', 'TUNISIENNE');
INSERT INTO AUTEUR VALUES (112, 'GARDARIN', 'GEORGES', 'FRANCAISE');
INSERT INTO AUTEUR VALUES (113, 'VALDURIEZ', 'PATRICK', 'FRANCAISE');
INSERT INTO AUTEUR VALUES (114, 'ULLMAN', 'JEFFREY', 'AMERICAINE');
INSERT INTO AUTEUR VALUES (115, 'DELOBEL', 'CLAUDE', 'FRANCAISE');
INSERT INTO AUTEUR VALUES (116, 'DATE', 'JC', 'AMERICAINE');
INSERT INTO AUTEUR VALUES (117, 'GELENBE', 'EROL', 'INDIENNE');
INSERT INTO AUTEUR VALUES (118, 'FLORY', 'ANDRE', 'FRANCAISE');
```

```
INSERT INTO ECRIT VALUES (102, '1_104_1050_2');
INSERT INTO ECRIT VALUES (103, '0_15_270500_3');
INSERT INTO ECRIT VALUES (104, '0_85_4107_3');
INSERT INTO ECRIT VALUES (105, '0_112_3785_5');
INSERT INTO ECRIT VALUES (116, '0_201_14439_5');
INSERT INTO ECRIT VALUES (112, '0_12_27550_2');
INSERT INTO ECRIT VALUES (117, '0_12_27550_2');
INSERT INTO ECRIT VALUES (111, '0_8_7707_2');
INSERT INTO ECRIT VALUES (118, '0_8_7707_2');
INSERT INTO ECRIT VALUES (106, '1_22_1721_3');
INSERT INTO ECRIT VALUES (107, '3_505_13700_5');
INSERT INTO ECRIT VALUES (108, '0_18_47892_2');
INSERT INTO ECRIT VALUES (109, '9_782070_36');
INSERT INTO ECRIT VALUES (110, '2_7296_0040');
```



```
INSERT INTO ECRIT VALUES (111, '0_26_28079_6');  
INSERT INTO ECRIT VALUES (112, '0_26_28079_6');  
INSERT INTO ECRIT VALUES (113, '0_26_28079_6');
```

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 noms de toutes les relations définies dans la BD BIBLIO (i.e. sans les tables systèmes) ?

Q11 : Afficher par ordre croissant, le nombre de tuples de chaque relation pour la BD BIBLIO.

Q12 : Quels sont les noms de tous les attributs de la relation ABONNE ?

Q13 : Quel est le nombre d'attributs définis dans la BD BIBLIO ?

Q14 : Quelles sont toutes les contraintes d'intégrité définies sur les relations de données de la base BIBLIO ?

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.*

Q15 : Comment, dans SQLPLUS, assurer la vérification automatique de l'intégrité de relation pour la relation EMPRUNT ?

Réalisez, si cela est nécessaire, une des manipulations correspondantes sur la base BIBLIO, puis vérifiez sa prise en compte en tentant d'insérer une valeur de clef dupliquée. Enfin, consultez la table système concernée pour contrôler la définition de l'index créé.

Q16 : 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.

Q17 : Créez une vue ABONNE_MONTP donnant le numéro, le nom et le prénom des abonnés habitant Montpellier. Puis réalisez l'insertion, **à travers la vue créée**, d'un nouvel abonné



montpellierain. Vérifiez ensuite la propagation de la mise à jour en consultant les tuples de la relation ABONNE.

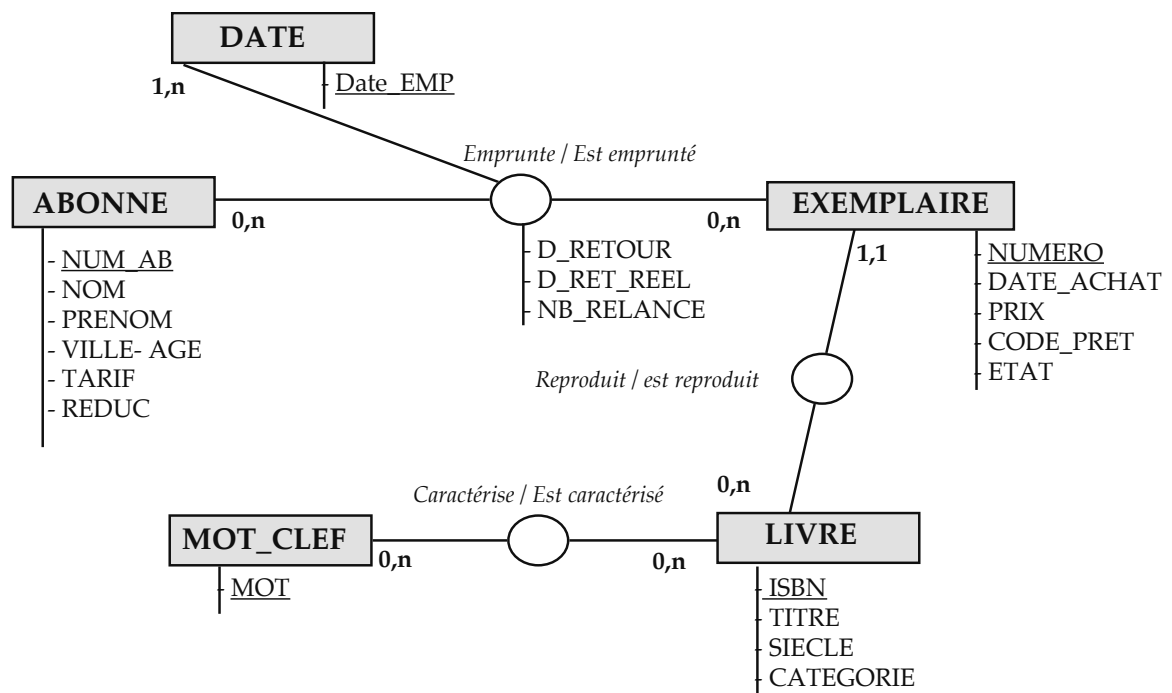
Q18 : Vérifiez que votre vue a bien été créée dans les tables systèmes.

Q19 : 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'}. Créez une vue ETAT_EXEMPLAIRE, permettant d'assurer que toute insertion à travers cette vue vérifie la contrainte de domaine énoncée. Vérifiez l'opération réalisée en tentant une insertion invalide.

Q20 : Définissez 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)