

BDW1 - Contrôle Terminal

UCBL - Département Informatique de Lyon 1 - 2017

Aucun document n'est autorisé. Durée : 1h30

Contexte

Soit le schéma de la base de données d'une bibliothèque suivant :

Oeuvre (*idO*, *titre*, *anneeC*, *categorie*, *#concours*, *anneeP*)

EstEnLien (*#idO1*, *#idO2*, *typeLien*)

Auteur (*idA*, *prenom*, *nom*, *dateN*, *dateD*)

Ecrit (*#idO*, *#idA*, *rang*)

Exemplaire (*#idO*, *numE*, *format*, *nbPages*, *langue*, *prix*, *#editeur*)

Emprunteur (*idP*, *prenom*, *nom*, *dateN*, *anPremInsc*)

Emprunt (*#idO*, *#numE*, *#idP*, *dateE*, *dRetour*, *dRendu*)

Editeur (*idE*, *nom*, *dir*, *pays*)

ConcoursLit (*nom*, *anneeC*, *resp*)

La relation *Oeuvre* stocke les informations relatives à une œuvre littéraire (i.e., le contenu d'un livre). Une œuvre est identifiée par un entier (*idO*). On stocke son titre, son année de création (*anneeC*), la catégorie de l'œuvre ('roman', 'essai...'), le concours dans lequel l'œuvre a été potentiellement primée (*concours*) et l'année d'obtention de ce prix (*anneeP*). L'attribut *concours* référence donc le nom d'un concours se trouvant dans la relation *ConcoursLit*.

Un tuple possible de *Oeuvre* est (1245, 'Chanson dure', 2016, 'roman', 'Congourt', 2016)

La relation *EstEnLien* stocke les informations relatives au lien éventuel entre deux œuvres représentées respectivement par leur identifiants (*idO1* et *idO2*) et le type lien entre les deux œuvres (*typeLien*). Ainsi, pour stocker l'information que l'œuvre identifiée par 109 et la suite de l'œuvre identifiée par 15, le tuple (15, 109, 'suite') sera stocké.

La relation *Auteur* stocke les informations sur les auteurs d'œuvres littéraires. Un auteur est identifié par un entier (*idA*). On stocke son nom, son prénom, sa date de naissance (*dateN*) et sa date de décès, si l'auteur est décédé (*dateD*). L'attribut *dateD* est sans valeur si l'auteur est vivant.

Un tuple possible de *Auteur* est (567, 'John Ronald Reuel', 'Tolkien', '1892-01-03' , '1973-09-02')

La relation *Ecrit* stocke l'information indiquant qu'un auteur (*idA*) a écrit l'œuvre (*idO*). Dans le cas d'œuvres écrites par plusieurs auteurs l'attribut (*rang*) indique le rang d'apparition de l'auteur. Ainsi pour stocker l'information que l'auteur identifiée par 123 est le premier auteur de l'œuvre 15, le tuple (15, 123, 1) sera stocké.

La relation *Exemplaire* stocke les informations relatives aux exemplaires de livre associés à une œuvre (i.e., le livre en papier). Un exemplaire est identifié relativement à une œuvre (*idO*) par un numéro (*numE*). On stocke le format de l'exemplaire ('classique', 'poche', 'illustrée...'), le nombre de pages (*nbPages*), la langue dans laquelle est écrite l'œuvre dans cet exemplaire et le prix d'achat de l'exemplaire par la bibliothèque.

Deux exemplaires en français de l'œuvre 145 édité par la maison d'édition 102 au format 'classique' dont on ne connaît ni le nombre de pages ni le prix d'achat seront stockés par les tuples (145, 1 , 'classique'

, NULL , 'français' , NULL , 102) et (145, 2 , 'classique' , NULL , 'français' , NULL , 102).

La relation *Emprunteur* stocke les informations sur les personnes qui ont adhéré à la bibliothèque. Un emprunteur est identifié par un entier (*idP*). On stocke son nom, son prénom, sa date de naissance (*dateN*) et l'année de sa première inscription à la bibliothèque (*anPremInsc*).

Un tuple possible de *Emprunteur* est (1245, 'Patrice' , 'STEULIVRE', '1977-12-06', 2015)

La relation *Emprunt* stocke l'information indiquant qu'un emprunteur (*idP*) a emprunté un exemplaire (*idO, numE*) à une date donnée (*dateE*). On stocke également la date de retour à laquelle l'exemplaire doit être rendu (*dRetour*) et la date à laquelle l'exemplaire a été rendu (*dRendu*). L'attribut *dRendu* est sans valeur tant que l'exemplaire n'est pas rendu.

Un emprunt réalisé le 10 mai 2017 d'un exemplaire de l'œuvre 678 par la personne 345 pour un retour prévu le 10 juin 2017 sera stocké par le tuple (678, 1, 345, '2017-05-10', '2017-06-10', NULL).

La relation *Editeur* stocke l'information sur les maisons d'édition qui éditent les exemplaires. Un éditeur est identifié par un entier (*idE*). On stocke également le nom de l'éditeur, le prénom et le nom du directeur de la maison d'édition (*dir*) et le pays où se trouve la maison d'édition (*pays*).

L'éditeur français 'Ligamard' dirigé par Eddy TAIX sera stocké par la tuple (12, 'Ligamard', 'Eddy TAIX', 'FR').

La relation *ConcoursLit* stocke les informations relatives aux concours littéraires qui chaque année prime des auteurs pour une œuvre. Un concours littéraire est identifié par son nom et on stocke l'année de création du concours (*anneeC*) et le prénom et le nom du responsable actuel du concours (*resp*)

Un tuple possible de *ConcoursLit* est ('Congourt', 1903, 'Bernard Vipot')

Partie A : Conception

Exercice 1 : Rétro-ingénierie (4 points)

Proposer un schéma Entité/Association dont la transformation permettrait d'obtenir le modèle relationnel donné précédemment dans la section Contexte.

Partie B : Interrogation

Exercice 2 : Requête SQL (10 points)

Écrire les requêtes suivantes en SQL.

1. Donner le titre des œuvres écrites par un auteur vivant et publiées par un éditeur français. Le résultat sera trié par ordre lexicographique sur le titre des œuvres puis selon l'ordre décroissant sur l'année de création en cas de titre identique.
2. Donner le prénom et le nom des auteurs qui ont écrit une œuvre à la fois en 2015 et en 2017.
3. Donner le prénom et le nom des auteurs qui n'ont jamais obtenu de prix littéraires.
4. Donner le prénom et le nom des emprunteurs qui ont du retard dans leur rendu. Pour rappel, la date courante est récupérée par NOW() ou SYSDATE.

5. Donner le titre des œuvres constituées d'au moins trois parties (i.e., une œuvre ayant une suite et dont la suite a également une suite) dont les 3 œuvres ont été empruntées par madame 'Ella PATOULU'.
6. Donner le nombre moyen d'exemplaires (achetés par la bibliothèque) pour chaque œuvre de l'auteur Yvan DUREVE . Le résultat retourné sera renommé 'NbMoy'.
7. Donner le prénom et le nom des auteurs qui ont obtenu au moins un prix dans tous les concours littéraires.
8. Donner le nom et le prénom des auteurs dont le nom contient un V en troisième place et qui ont écrit plus de 3 œuvres la même année.

Exercice 3 : Requête en Algèbre Relationnelle (3 points)

Écrire les requêtes suivantes en Algèbre Relationnelle.

1. Donner le titre des œuvres de catégorie 'roman' primées en 2016 et éditées par un éditeur américain.
2. Donner le nom des maisons d'édition qui n'ont jamais eu d'œuvre primée au concours littéraire nommé 'Congourt'.

Partie C : Web

Pour cette partie, nous supposons que la base de données de la bibliothèque est stockée dans la base "maBD" d'un SGBD Mysql stocké sur le serveur Web. L'utilisateur dispose du compte "TOTO" avec le mot de passe "titi".

Exercice 4 : Chercher les erreurs (3 points)

Soit le code PHP suivant :

```
<?php

$c = mysqli_connect("localhost", "TOTO", "titi", "maBD");

function retourneInfos(){
    $res = "<table>";
    $q = "SELECT O.titre, E.numE AS nb ";
    $q .= "FROM OEUVRE O NATURAL JOIN EXEMPLAIRE E";
    $q .= "GROUP BY O.idO";
    $t = mysqli_query($c, $q);

    if($r = mysqli_fetch_array($t) ){
        $res .= "<tr>";
        $res .= "<td>" . $r['titre'] . "</td>";
        $res .= "<td>" . $r['nb'] . "</td>";
        $res .= "</tr>";
    }
    $res = "</table>";

    return echo $res;
}
?>
```

L'appel '`<?php retourneInfos();?>`' est sensé retourner une table HTML dans laquelle chaque ligne correspond à un titre d'une œuvre et au nombre d'exemplaires associées à cette œuvre.

1. Le code fourni ne retourne pas le résultat attendu. Proposer une correction de ce code. NB : Vous pouvez soit lister les erreurs trouvées et proposer une correction, soit réécrire entièrement la fonction.
2. On souhaite afficher les titres en gras dans le tableau HTML, et un fond rouge pour le nombre d'exemplaires associés. Quelle solution proposez-vous ?