

L2 INF245 — Examen — mai 2014,  
session 1



Durée : 2H

Document autorisé : une feuille A4, recto-verso, manuscrite

## 1 A propos de relations

Soient R et S deux relations dont la valeur est donnée dans les tableaux suivants :

R	A	B
	1	11
	2	20
	8	
	4	21
	3	10

S	B	C
	10	100
	8	110
	11	101
	10	101
	11	100
	12	100

### Question 1 (4 points) :

Donner, sous forme tabulaire, le schéma et la valeur de la relation construite par chacune des requêtes ci-dessous (il peut s'agir d'un message d'erreur Oracle).

1. `select C, B, D  
from S where D <> 3;`
2. `select A, B  
from R where B < 20;`
3. `select R.A, R.B, S.C  
from R left outer join S on (R.B=S.B);`
4. `select A, count(distinct B) as nbB,  
count(distinct C) as nbC  
from R natural join S group by A;`

### Question 2 (3 points) :

Reformuler en SQL chacune des requêtes listées ci-dessous. Suivre très rigoureusement les instructions fournies.

1. `select A, B from R where B in  
(select B from S where C > 100);`  
**Instruction** : la solution ne doit pas utiliser l'opérateur `in` ni sa négation `not in`.
2. `select B from S group by B having count(distinct C) = 1;`  
**Instruction** : la solution ne doit pas utiliser l'opérateur `group by`.

## 2 A propos d'une bibliothèque

On étudie une application de base de données utilisée par une bibliothèque.

### 2.1 Position du problème

Une association, constituée d'un ensemble d'*adhérents*, gère une bibliothèque de prêts : elle possède un certain nombre d'exemplaires (au moins un) de chaque *livre* de son *catalogue*. Une personne ne peut emprunter un livre que si toutes les règles suivantes sont respectées :

- La personne doit être adhérente de l'association.

- Le livre doit apparaître dans le catalogue et au moins un exemplaire doit exister en bibliothèque. Un exemplaire est disponible s'il n'est pas en cours de prêt.
- Le nombre de livres empruntés par une personne à un instant donné ne peut pas dépasser une limite fixée qui est la même pour tous les adhérents (on choisit arbitrairement la valeur 4).

La base de données que l'on considère ici gère le *catalogue* des livres et le *répertoire* des adhérents. Pour chaque livre, on dispose des informations suivantes : son titre, son auteur identifié par ses nom et prénom, et son année d'édition. Un livre est identifié par son titre. Le *fonds* de la bibliothèque est constitué d'un ensemble d'exemplaires de ces livres (au moins un exemplaire de chaque livre). Chaque exemplaire est identifié par sa cote. Pour chaque adhérent, on dispose des informations suivantes : le numéro qui l'identifie, ses nom, prénom, adresse, année de naissance et date d'adhésion. Chaque *emprunt* associe un exemplaire, un adhérent et une date de début de l'emprunt. Lorsqu'un exemplaire emprunté est rendu, le jour du retour est enregistré.

## 2.2 Schéma de relations

LeCatalogue (titre, nom, prénom, anEd)

*/\* <t, n, p, a> ∈ LeCatalogue ⇔ le livre de titre t a été écrit par l'auteur de nom n et de prénom p. Ce livre a été édité pour la première fois dans l'année a. \*/*

LeFonds (cote, titre)

*/\* <c, t> ∈ LeFonds ⇔ la bibliothèque possède un exemplaire identifié par c du livre de titre t. \*/*

LesEmprunts (cote, datEmp, datRet, noAdh) (cote, datRet) est une autre clef candidate.

*/\* <c, de, dr, a> ∈ LesEmprunts ⇔ l'exemplaire identifié par c est emprunté par l'adhérent a depuis la date de. Si dr a une valeur, alors l'exemplaire c a été rendu à la date dr, sinon l'emprunt est en cours. \*/*

LesAdhérents (noAdh, nom, prénom, adresse, anNais, datAdh)

*/\* <a, n, p, v, y, d> ∈ LesAdhérents ⇔ l'adhérent identifié par a, a pour nom n et prénom p, et habite dans la ville v. Son année de naissance est y. Il(elle) est adhérent(e) depuis la date d. \*/*

Les domaines :

domaine(titre) = domaine(nom) = domaine(prénom) = chaîne de caractères

domaine(cote) = domaine(noAdh) = entiers

domaine(anEd) = domaine(anNais) = date (année) (par exemple, 1999, 2014, etc.)

domaine(datEmp) = domaine(datRet) = domaine(datAdh) = date (jour) (par exemple, 21/02/2014, 1/3/2014, etc.)

Les contraintes d'intégrité référentielles :

LeFonds[titre] = LeCatalogue[titre]

LesEmprunts[cote] ⊆ LeFonds[cote]

LesEmprunts[noAdh] ⊆ LesAdhérents[noAdh]

Les autres contraintes :

- Le nombre d'exemplaires empruntés par un adhérent à un instant donné est inférieur ou égal à 4.

- A chaque instant, un exemplaire est emprunté 0 ou 1 fois.

- Dans LeFonds, titre a obligatoirement une valeur.

- Dans LesEmprunts, datEmp < datRet.

## 2.3 Expression de requêtes

**Indications :**

Les opérateurs + et - ont été étendus aux dates. Par exemple, d étant une date, d-15 est la date d décalée de 15 jours dans le passé. d1 et d2 étant deux dates, d1-d2 est le nombre de jours séparant d1 de d2 (si d1 > d2 alors d1-d2 > 0). La fonction sysdate retourne un instant (à la granularité de la seconde), dont la valeur est observée au moment de son exécution.

to\_char(sysdate, 'DD-MM-YYYY') = '23-05-2014'.

to\_char(sysdate, 'Day') = 'Vendredi'.

to\_char(to\_date('23/05/2014', 'DD/MM/YY'), 'DD-MM-YY HH:MI:SS') = '23-05-14 12:00:00'

**Question 3 (5 points) :**

Exprimer en SQL les requêtes ci-dessous.

**Les requêtes devront construire des résultats sans répétition de valeurs, la clause `distinct` ne sera utilisée que lorsque nécessaire. Les produits de relation seront exprimés dans la clause `from` des requêtes. Toute requête imbriquée sera soigneusement spécifiée.**

1. Donner les adhérents en dépassement (c'est-à-dire ceux qui ont au moins un emprunt en cours depuis plus de 15 jours).
2. Pour chaque adhérent, donner le nombre de livres empruntés (dans le passé et au présent). On dit qu'un livre est emprunté dès lors qu'un exemplaire de ce livre est emprunté. Le résultat est 0 pour les adhérents sans emprunt.
3. Pour chaque livre, donner son titre et la cote de l'exemplaire de ce livre emprunté le plus souvent.

**2.4 Implantation de l'application**

Nous considérons la base de données telle que spécifiée Section 2.1. Le code SQL permettant de définir les relations `LesAdhérents` et `LeCatalogue` est donné ci-dessous :

```
create table LesAdherents ( noAdh integer, nom varchar(20), prenom varchar(20),
                           adresse varchar(20), anNais date, datAdh date,
                           constraint Adh_c1 primary key (noAdh) );
create table LeCatalogue ( titre varchar(50), nom varchar(20), prenom varchar(20),
                           anEd date,
                           constraint Cat_c1 primary key (titre) );
```

**Question 4 (2 points) :**

Fournir le code SQL Oracle qui permet de définir les relations `LesEmprunts` et `LeFonds`.

**Indications :**

On suppose l'existence de la fonction dont la signature est `getConstraintName: string → string`. Par exemple, soit `e` une erreur SQL retournée par le système, `getConstraintName(e)` est le nom de la contrainte dont la tentative de validation a échoué (par exemple `Adh_c1`).

**Question 5 (2 points) :**

Le programme `AjoutAdherent-action.php` écrit en PHP et dont le squelette est donné page 4 permet d'enregistrer une nouvelle adhésion et de construire un message destiné à l'utilisateur qui l'informe de l'issue de cette adhésion. Compléter le programme `AjoutAdherent-action.php` (utiliser la page fournie avec le sujet).

**Question 6 (4 points) :**

Le programme `AjoutEmprunt-action.php` écrit en PHP et dont le squelette est donné page 5 permet d'enregistrer un nouvel emprunt et de construire un message destiné à l'utilisateur qui l'informe de l'issue de cette adhésion. Compléter le programme `AjoutEmprunt-action.php` (utiliser la page fournie avec le sujet).

Merci d'utiliser cette page pour répondre à la question 3.

Numéro d'anonymat :

```

/* AjoutAdherent-action.php enregistre l'adhésion d'un nouvel adhérent.
 * Les valeurs décrivant un adhérent (noAdh, nom, prenom, adresse et anNais) ont été
 * saisies par le biais d'un formulaire.
 * Génère un message indiquant si l'opération s'est bien
 * déroulée ou un message d'erreur.
 * Hypothèse : la base de données est cohérente.
 */
<?php
    $noAdh = $_POST['noAdh']; $nom = $_POST['nom'];
    $prenom = $_POST['prenom']; $adresse = $_POST['adresse'];
    $anNais = $_POST['anNais'];
    // construction de la requete
    $requete = "insert into LesAdherents values(:a,:n,:p,:ad,to_date(:an,'YYYY'), sysdate);
    $curseur = oci_parse ($lien, $requete);
    oci_bind_by_name($curseur,":a",$noAdh); oci_bind_by_name($curseur,":n",$nom);
    oci_bind_by_name($curseur,":p",$prenom);
    oci_bind_by_name($curseur,":ad",$adresse);
    oci_bind_by_name ($curseur, ":an", $anNais);
    $res = oci_execute ($curseur, OCI_NO_AUTO_COMMIT);
    if ($res) {
        .....
        $message = "Nouvel adhérent enregistré.";
    } else {
        .....
        $e = oci_error($curseur);
        $error = getConstraintName(e) ;
        if ($e = "ad_C0") { /* violation d'une contrainte de clef primaire */
            $message = "Erreur : le numéro d'adhérent (".$noAdh.") existe déjà.";
        } else { /* Autre erreur SQL */
            $message = "Erreur : ".$e['message'];
        }
    }
    echo $message;
    Deconnexion ($lien);
?>

```

Merci d'utiliser cette page pour répondre à la question 4.

Numéro d'anonymat :

```

/* AddEmprunt-action.php enregistre l'emprunt de l'exemplaire cote par l'adhérent noAdh.
 * La valeur de cote et celle de noAdh ont été saisies par le biais d'un formulaire.
 * Génère un message indiquant si l'opération s'est bien déroulée ou un message
 * d'erreur.
 * Hypothèse : la base de données est cohérente.
 */
$noAdh = $_POST['noAdh'];
$cote = $_POST['cote'];
$reqTestNb = ".....";
$curseurTestNb = oci_parse ($reqTestNb);
oci_bind_by_name ($curseurTestNb, ....., .....);
$resTestNb = oci_execute($curseurTestNb, OCI_NO_AUTO_COMMIT);

if (.....) {
    $message = "Nombre d'emprunts dépassé.";
} else {
    $requete = "insert into LesEmprunts values(.....)";
    $curseur = oci_parse($requete);
    oci_bind_by_name ($curseur, ":c", $cote);
    oci_bind_by_name ($curseur, ":a", $noAdh);
    $res = oci_execute ($curseur, OCI_NO_AUTO_COMMIT);
    if ($res) {
        .....
        .....
    } else {
        .....
        $e = oci_error($lien);
        $error = getConstraintName(e);
        if ($e = "Emp_c1") { /* violation d'une contrainte de clef primaire */
            $message = "Erreur : ..... ";
        }
        if (.....)
            $message = "Erreur : l'adhérent ".$noAdh." n'existe pas.";
        }
        if (.....) {
            $message = .....;
        }
        if (.....) {
            /* Autre erreur SQL */
            $message = "Erreur : ".$e['message'] ;
        }
    }
}
echo $message;
Deconnexion ($lien);

```