

NFA011 - Développement d'applications avec les bases de données

Examen février 2017. Durée : 2h00.

Consignes :

- Tous les documents sont autorisés.
 - Les téléphones mobiles et autres équipements communicants (exemple : PC, tablette, etc.) doivent être éteints et rangés dans les sacs pendant toute la durée de l'épreuve.
 - Pensez à reporter votre numéro d'anonymat sur toutes les feuilles.
 - Merci de soigner votre écriture et de ne pas rédiger au crayon papier sur votre copie.
- Ce sujet comporte 3 page d'énoncé.
-

Soit la base de données suivante permettant de gérer des tournoi de tennis :

```
JOUEUR(id, nom, prénom, age, pays)
TERRAIN(id, capacité, ville)
MATCH(idMatch, idJoueur1, idJoueur2, idTerrain, dateMatch, gagnant, nbSpectateurs, nomTournoi)
BILLET(idMatch, numéroSpectateur, nomSpectateur, prénomSpectateur, prixBillet)
```

x' L'attribut MATCH(gagnant) est un INTEGER qui indique le joueur gagnant : 1 pour le 1er joueur ou 2 pour le 2eme joueur. L'attribut TERRAIN(capacité) désigne le nombre de places du terrain.

1 Questions SQL (7 points)

- 1) Donner la liste (nom, prénom) des spectateurs ayant vu un match de Gaël Mafille le 12 janvier 2017. **(en algèbre)**.
- 2) Donner le nombre de matchs que Richard Casquette a gagné contre Andy Beurré. **(en SQL)**.
- 3) Donner le nom et prénom des joueurs n'ayant jamais gagné un match dans la ville de Saint-Pol-de-Léon. **(en SQL)**.
- 4) Donner le nom des spectateurs ayant vu tous les matchs de Rafael Ladal. **(en SQL)**.
- 5) Donner les noms des joueurs ayant joué deux matchs la même journée. **(en SQL)**.

2 Questions PL/SQL (7 points)

- 1) Écrivez en PL/SQL un déclencheur (*trigger*) qui, lors de l'insertion d'une ligne dans la table BILLET, met à jour l'attribut nbSpectateurs dans la table MATCH. Si le idMatch correspondant n'existe pas dans la table MATCH la nouvelle ligne ne doit pas être insérée et un message d'erreur doit être affiché. (3 points).

Correction :

```
CREATE OR REPLACE TRIGGER MisAJourMATCH
BEFORE INSERT ON BILLET FOR EACH ROW
BEGIN
    UPDATE MATCH SET nbSpectateurs = nbSpectateurs + 1
    WHERE idMatch = :new.idMatch;

    IF SQL%ROWCOUNT = 0 THEN
        RAISE_APPLICATION_ERROR(-20100, 'Le match n'existe pas.');
```

```
END IF;
END
```

2) Écrivez une fonction PL/SQL qui prend en paramètre le nom d'un pays et qui retourne le nombre de joueurs de cette nationalité qui remplissent les deux conditions suivantes :

- Ont l'âge plus grand que la moyenne d'âge (de ce pays) ;
- Ont gagné tous leurs matchs joués à Paris.

Utilisez un curseur avec paramètre. (4 points)

Correction :

```
CREATE OR REPLACE FUNCTION nbGagnantsParis(ppays IN JOUEUR.pays%TYPE)
RETURN INTEGER IS
  -- Curseur avec paramètre (liste des joueurs du pays p)
  CURSOR idJoueurs(p JOUEUR.pays%pays) IS SELECT id FROM Joueur WHERE pays = p;
  moyenne NUMBER;
  n1 NUMBER;
  n2 NUMBER;
  age_j JOUEUR.age%TYPE;
  resultat NUMBER := 0;
BEGIN
  -- Moyenne d'âge pour ppays
  SELECT AVG(age) INTO moyenne FROM Joueur
  WHERE pays = ppays;
  -- Parcourir la liste des joueurs et verifier qu'ils remplissent les deux conditions
  FOR jid in idJoueurs(ppays) LOOP
    --Le nombre de matchs joués à Paris par ce joueur
    SELECT COUNT(*) INTO n1
    FROM MATCH m, TERRAIN t
    WHERE m.idTerrain = t.id AND (jid = m.idJoueur1 OR jid = m.idJoueur2) AND t.ville = 'Paris';
    --Le nombre de matchs gagnés à Paris par ce joueur
    SELECT COUNT(*) INTO n2
    FROM MATCH m, TERRAIN t
    WHERE m.idTerrain = t.id AND ((jid = m.idJoueur1 AND gagnant = 1) OR
      (jid = m.idJoueur2 AND gagnant = 2)) AND t.ville = 'Paris';
    -- Trouve l'âge du joueur
    SELECT age INTO age_j FROM JOUEUR WHERE id = jid;
    IF(n1 = n2 AND age_j > moyenne) THEN
      resultat := resultat+1;
    END IF;
  END FOR
  RETURN resultat;
END
```

3 Questions JDBC (6 points)

Écrivez en Java une procédure stockée qui affiche pour chaque joueur qui a gagné le tournoi Roland-Garros, le nombre total d'entrées d'argent qu'il a fait (tous les tournois compris, pas seulement à Rolland-Garros). Utilisez une interface PreparedStatement paramétrée. (Écrire uniquement la partie du code java qui fait le travail demandé et non l'importation de paquets, l'enregistrement du pilote et l'établissement des connexions.)

Correction :

```

public class CorrectionExamen {
public static int verifieRolandGarros() {
try {
    Connection conn = DriverManager.getConnection("jdbc:default:connection:");

    // recuperer les ids des joueurs qui ont gagné le tournoi de Roland-Garros
    String query = "SELECT j.id
                    FROM JOUEUR j, MATCH m
                    WHERE ((j.id = m.idJoueur1 AND gagnant = 1) OR
                           (j.id = m.idJoueur2 AND gagnant = 2)) AND
                           m.nomTournoi = 'Roland-Garros'";
    Statement stmt = connexion.createStatement(ResultSet.TYPE_FORWARD_ONLY,
                                                ResultSet.CONCUR_READ_ONLY);

    ResultSet rs = stmt.executeQuery(query);

    // PreparedStatement paramétrée
    String instrSQL = "SELECT SUM(prixBillet)
                      FROM MATCH m, BILLET b
                      WHERE m.idMatch = b.idMatch AND (idJoueur1 = ? OR idJoueur2 = ?)";
    PreparedStatement pstmt conn.prepareStatement(instrSQL);

    while(rs.next()) {
        int jid = rs.getInt(1);
        pstmt.setInt(1, jid);
        pstmt.setInt(2, jid);
        ResultSet r = pstmt.executeQuery();
        r.next();
        int nb = r.getInt(1);
        System.out.println(nb);
        r.close();
    }
    pstmt.close();
    stmt.close();
} catch(SQLException e) {
System.err.println('SQLException: ' + ex.getMessage());
}}

```