

LICENCE D'INFORMATIQUE

Examen de BASES DE DONNÉES

9 janvier 2024

Durée : 2 heures

Documents autorisés : aucun

Le barème est donné à titre indicatif

Exercice 1 : Match de foot (2 points)

Considérez la relation FOOT (IdMatch, IdEquipe, IdJoueur, Division, Lieu, DateM, NbButs) contenant de l'information sur le nombre de buts marqués par les joueurs d'une équipe de foot appartenant à une division, lors d'un match disputé à une certaine date et un certain lieu.

Questions :

1. représenter le graphe minimum des dépendances fonctionnelles,
2. définir le (les) identifiants(s),
3. définir sa forme normale en la justifiant (en une phrase),
4. proposer si la relation n'est pas normalisée, une décomposition sans perte et préciser la forme normale des nouvelles relations.

Exercice 2 : Naissance (8 points)

Le service maternité d'un hôpital utilise une base de données relationnelle pour gérer les informations sur les naissances qui y ont lieu.

PATIENTE (NumPatiente, Nom, Prénom, Commune)

NAISSANCE (DateN, Rang, idMère, Nom, Prénom, Poids, Taille, Sexe, idAcc, idMédecin)

TYPEACCOUCHEMENT (idAcc, LibelleAcc)

OBSTETRICIEN (idObs, Nom, Prénom)

Un obstétricien est un médecin qui supervise l'accouchement. Un accouchement peut être de deux types ayant respectivement pour libellé césarienne ou voie naturelle.

Dans la relation NAISSANCE,

- l'attribut DateN contient le mois et l'année de naissance du bébé, par exemple 'décembre 2023' ;
- l'attribut Rang indique le rang de naissance du bébé dans le mois. Il recommence donc à 1 au début de chaque nouveau mois ;
- la clé étrangère idMère fait référence à la clé primaire NumPatiente de la relation PATIENTE ;
- l'attribut Poids renferme le poids en grammes du bébé à la naissance et l'attribut Taille contient sa taille en centimètres à la naissance ;
- l'attribut Sexe a pour valeur 'F' ou 'M' ;
- la clé étrangère idAcc fait référence à la clé primaire idAcc de la relation TYPEACCOUCHEMENT. Elle permet de savoir si la patiente a accouché par voie naturelle ou par césarienne ;
- la clé étrangère idMédecin fait référence à la clé primaire idObs de la relation OBSTETRICIEN.

Questions :

1. Indiquer, pour chacune des propositions suivantes, si elles peuvent être ou non clé primaire de la relation NAISSANCE. Justifier chaque réponse.

- a) idMere
- b) (Date + Rang)
- c) (Poids + Taille)

2. Écrire les requêtes SQL suivantes :

- a) Afficher la liste des patientes (attributs nom et prénom) dont la commune de résidence est Dijon. On fait l'hypothèse qu'il n'y a pas d'homonymes.
- b) Afficher la liste des bébés (attributs nom, prénom et poids en kilogrammes) nés durant le mois de décembre 2023 par ordre décroissant de leur poids.
- c) Afficher le poids moyen en grammes des bébés nés par césarienne.
- d) Afficher la liste des patientes (attributs nom de la patiente et nom de son obstétricien) qui ont été accouchées par le même obstétricien pour tous leurs bébés.
- e) Afficher la liste des patientes (attributs nom et prénom) qui ont subi une césarienne pour tous leurs accouchements.
- f) Afficher le top 3 des prénoms féminins, avec leur nombre de naissances, donnés en 2023.

Pour limiter à N lignes retournées, il faut utiliser la clause FETCH : `FETCH FIRST N ROWS ONLY`.

Exercice 3 : Prime de pouvoir d'achat exceptionnelle simplifiée (4 points)

Un décret portant sur la création d'une prime de pouvoir d'achat exceptionnelle pour certains agents publics a été publié au Journal Officiel le 1^{er} août 2023. La prime de pouvoir d'achat est soumise à plusieurs conditions cumulatives :

- avoir été recruté avant le 1^{er} janvier 2023 ;
- être toujours en poste au 30 juin 2023 ;
- avoir perçu, entre le 1^{er} juillet 2022 et le 30 juin 2023, une rémunération inférieure ou égale à 39 000 € brut, soit 3 250 € brut par mois maximum.

Son montant forfaitaire est calculé en fonction de la rémunération brute perçue au titre de la période allant du 1^{er} juillet 2022 au 30 juin 2023 soit 12 mois :

- rémunération inférieure ou égale à 23 700 € : prime de 800 € ;
- rémunération supérieure à 23 700 € et inférieure ou égale à 27 300 € : prime de 700 € ;
- rémunération supérieure à 27 300 € et inférieure ou égale à 29 160 € : prime de 600 € ;
- rémunération supérieure à 29 160 € : prime de 500 €.

Question :

1. Écrire un programme PL/SQL qui met à jour l'attribut Prime dans la relation EMPLOYE (NE, DateEmbauche, DateDépart, SalaireMensuelBrut, Prime).

Remarques :

- l'attribut NE représente le numéro de l'employé ;
- les attributs DateEmbauche et DateDépart sont de type DATE ;
- si l'agent est toujours en poste l'attribut DateDépart est à NULL ;
- on fait l'hypothèse d'un SalaireMensuelBrut toujours le même ;
- l'attribut Prime est à NULL.

Exercice 4 : Analyse d'un programme PL/SQL (3 points)

Soit le schéma relationnel qui décrit des zones géographiques dont certaines, appelées zone de défense, présentent des caractéristiques demandant une surveillance accrue. Une zone de défense a un responsable.

Zone (CodeZone, Libellé)

ZoneDefense (CodeZone, Responsable)

On vous donne le programme PL/SQL suivant :

```
1. CREATE OR REPLACE TRIGGER T1
2. BEFORE UPDATE ON Zone
3. FOR EACH ROW
4. DECLARE
5.     INTEGRITY_ERROR EXCEPTION;
6.     ERRNO INTEGER;
7.     ERRMSG CHAR(200);
8.     DUMMY INTEGER;
9.     FOUND BOOLEAN;
10.    CURSOR T(p_CodeZone Zone.CodeZone%TYPE) IS
11.        SELECT 1 FROM ZoneDefense
12.        WHERE CodeZone = p_CodeZone;
13. BEGIN
14.     IF (:OLD.CodeZone != :NEW.CodeZone) THEN
15.         OPEN T (:OLD.CodeZone);
16.         FETCH T INTO DUMMY;
17.         FOUND := T%FOUND;
18.         IF FOUND THEN
19.             ERRNO := -20005;
20.             ERRMSG := 'Modification code zone ' || :OLD.CodeZone || ' interdite.';
21.             RAISE INTEGRITY_ERROR;
22.         END IF;
23.         INSERT INTO LogZone(CodeZone, DateMJ) VALUES (:OLD.CodeZone, SYSDATE);
24.         CLOSE T;
25.     END IF;
26. EXCEPTION
27.     WHEN INTEGRITY_ERROR THEN
28.         RAISE_APPLICATION_ERROR (ERRNO, ERRMSG); CLOSE T;
29. END;
30. /
```

On suppose que la table LogZone (CodeZone, DateMJ) a été préalablement créée.

Questions :

1. Quel est le but de ce code ?
2. Donner la signification des instructions suivantes :
 - a) « FOR EACH ROW » en ligne 3.
 - b) « CURSOR T(p_CodeZone Zone.CodeZone%TYPE) » en ligne 10.
 - c) « SELECT 1 FROM ... » en ligne 11.
 - d) « RAISE INTEGRITY_ERROR » en ligne 21.

Exercice 5 : Déroulement d'un programme PL/SQL (3 points)

Toutes les entreprises, petites ou grandes, sont obligées de réaliser un inventaire au moins une fois par an. Il consiste à établir la liste complète des éléments composant le stock en effectuant un comptage précis. Une jardinerie soucieuse de sa gestion et suite à une recrudescence des vols voudrait pouvoir être alertée des différences entre le stock théorique et le stock réel. Pour l'aider dans cette tâche, un programme PL/SQL a été mis au point. Il a pour objectif de détecter les anomalies dans l'inventaire de cette jardinerie.

Un jeu d'essai a été constitué pour tester le programme. La table ANOMALIE comporte deux colonnes : noProd et texte. Elle a été préalablement créée. Le contenu des tables STOCK et INVENTAIRE, pour réaliser le test, est fourni ci-dessous :

Table STOCK

noProd	Désignation	qte
A010	Arrosoir 10 litres	5
A015	Arrosoir 15 litres	1
B100	Bèche verte	2
P500	Plantoir	0
S050	Sac déchets 50 l	4

Table INVENTAIRE

noProd	qteReelle
A010	5
A300	1
B100	1
S050	5

Questions : pour cet exercice vous devez rédiger les réponses directement sur le sujet et glisser la page dans votre copie double après avoir renseigné votre numéro d'anonymat :

Numéro anonymat

1. Il vous est demandé de compléter le contenu de la table ANOMALIE tel qu'il sera après l'exécution du code PL/SQL.

Table ANOMALIE

noProd	texte

2. Que se passe-t-il si les deux tables sont vides ? Justifier votre réponse.

Voici le code PL/SQL qui a été proposé :

```
DECLARE
  CURSOR cursS IS SELECT * FROM Stock ORDER BY noProd;
  enregS Stock%ROWTYPE;
  CURSOR cursInv IS SELECT * FROM Inventaire ORDER BY noProd;
  enregInv Inventaire %ROWTYPE;
  CAS_FIN_S EXCEPTION;
  CAS_FIN_INV EXCEPTION;
BEGIN
  DELETE FROM Anomalie;
  OPEN cursS;   OPEN cursInv;
  FETCH cursS INTO enregS;  FETCH cursInv INTO enregInv;
  WHILE cursS%FOUND OR cursInv%FOUND LOOP
  BEGIN
    IF enregS.noProd = enregInv.noProd THEN
      IF enregS.qte < enregInv.qteReelle THEN
        INSERT INTO Anomalie VALUES (enregS.noProd, 'qteRéelle > qteStock');
      ELSIF enregS.qte > enregInv.QteReelle THEN
        INSERT INTO Anomalie VALUES (enregS.noProd, 'qteRéelle < qteStock');
      END IF;
      FETCH cursS INTO enregS;
      FETCH cursInv INTO enregInv;
      IF cursS%NOTFOUND THEN RAISE CAS_FIN_S; END IF;
      IF cursInv%NOTFOUND THEN RAISE CAS_FIN_INV; END IF;
    ELSIF enregS.noProd < enregInv.noProd THEN
      IF enregS.qte!=0 THEN
        INSERT INTO Anomalie VALUES (enregS.noProd, 'Article non inventorié ');
      END IF;
      FETCH cursS INTO enregS;
      IF cursS%NOTFOUND THEN RAISE CAS_FIN_S; END IF;
    ELSE
      INSERT INTO Anomalie VALUES (enregInv.noProd, 'Article inconnu');
      FETCH cursInv INTO enregInv;
      IF cursInv%NOTFOUND THEN RAISE CAS_FIN_INV; END IF;
    END IF;
  EXCEPTION WHEN CAS_FIN_INV THEN WHILE cursS%FOUND LOOP
    INSERT INTO Anomalie VALUES (enregS.noProd, 'Article non inventorié ');
    FETCH cursS INTO enregS;
  END LOOP;
  WHEN CAS_FIN_S THEN WHILE cursInv%FOUND LOOP
    INSERT INTO Anomalie VALUES (enregInv.noProd, 'Article inconnu ');
    FETCH cursInv INTO enregInv;
  END LOOP;
  END;
  END LOOP;
  CLOSE cursS;  CLOSE cursInv;
EXCEPTION WHEN NO_DATA_FOUND THEN DBMS_OUTPUT.PUT_LINE('Aucun enregistrement');
                                CLOSE cursS;   CLOSE cursInv;
END;
/
```