

**TP 1.**Évaluation : **30 points**

Date de distribution : 16/06/2020

Date de remise : 14/07/2020

Remettre le rapport et les programmes.

Le rapport doit être obligatoirement rédigé avec un traitement de texte WORD. Il doit contenir :

- Le code de tous les programmes;
- Les jeux de tests;
- Toute information pertinente que vous jugez nécessaire.

La page de couverture doit contenir le nom du collège et du département, la session, le nom de l'étudiant, le numéro de groupe, le nom et le numéro du module, le numéro du travail pratique et le nom de l'enseignant.

Mettez vos programmes et votre rapport dans le répertoire <nom\_famille\_équipier1\_nom\_famille\_équipier2> et envoyez-le sur Mio.

**GESTION DES EMPLOYES.**

Soit la base de données *GestionEmployes* pour la gestion des employés avec les tables suivantes :

**Departements** (**NumDepartement**, NomDepartement, LocalDepartement, *ChefDepartement*, DateCreationDepartement)

**Employes** (**NumeroEmploye**, NomEmploye, PrenomEmploye, AdresseEmploye, VilleEmploye, ProvinceEmploye, CodePostalEmploye, TelephoneEmploye, PosteEmploye, SalaireEmploye, CommissionEmploye, *NumDepartement*, *SuperieurEmploye*, DateEmbaucheEmploye, DateNaissanceEmploye)

La BD est créée par le script suivant :

```
CREATE TABLE Departements (
    NumeroDepartement VARCHAR2 (6)
        CONSTRAINT Departements_PK_NumDepartement PRIMARY KEY,
    NomDepartement VARCHAR2 (30)
        CONSTRAINT Departements_NN_NomDepartement NOT NULL
        CONSTRAINT Departements_UQ_NomDepartement UNIQUE,
    LocalDepartement VARCHAR2 (30),
    ChefDepartement NUMBER (9)
        CONSTRAINT Departements_UQ_ChefDepartement UNIQUE,
    DateCreationDepartement DATE DEFAULT SYSDATE
);
CREATE TABLE Employes (
    NumeroEmploye NUMBER (9)
        CONSTRAINT Employes_PK_NumEmploye PRIMARY KEY
        CONSTRAINT Employes_CK_NumEmploye CHECK (NumeroEmploye > 0),
    NomEmploye VARCHAR2 (20)
        CONSTRAINT Employes_NN_NomEmploye NOT NULL,
    PrenomEmploye VARCHAR2 (20)
        CONSTRAINT Employes_NN_PrenomEmploye NOT NULL,
    AdresseEmploye VARCHAR2 (50),
    VilleEmploye VARCHAR2 (20) DEFAULT 'Montréal',
    ProvinceEmploye CHAR (2) DEFAULT 'QC',
```

```

CodePostalEmploye CHAR (6),
TelephoneEmploye NUMBER (10),
PosteEmploye VARCHAR2 (20),
SalaireEmploye NUMBER (9,2),
CommissionEmploye NUMBER (2,2),
NumeroDepartement VARCHAR2 (6)
CONSTRAINT Employes_FK_NumDepartement
REFERENCES Departements(NumeroDepartement),
SuperieurEmploye NUMBER (9)
CONSTRAINT Employe_FK_SuperieurEmploye
REFERENCES Employes(NumeroEmploye),
DateEmbaucheEmploye DATE DEFAULT SYSDATE,
DateNaissanceEmploye DATE,
CONSTRAINT Employe_CK_DateNaiss CHECK ((dateEmbaucheEmploye -
dateNaissanceEmploye) >= 6574.32)
);
ALTER TABLE Departements ADD CONSTRAINT Departements_FK_Chef FOREIGN KEY
(ChefDepartement) REFERENCES Employes(NumeroEmploye);

```

Les scripts de création et de remplissage de BD GestionEmployes se trouvent sur le Léa sous les noms CreateEmpDept et InsertEmpDept.

### Travail à faire.

1. **(6 points)** Écrire le programme PL/SQL non stocké **DeleteDept.sql** qui supprime tous les départements qui n'ont aucun employé. Pour effectuer les tests, insérer dans la table **Departements** quelques nouveaux départements.
2. **(6 points)** Écrire le programme PL/SQL non stocké **Affiche.sql** qui affiche les numéros, les noms, les dates d'embauche et les salaires de tous les employés d'un département dont le nom est entré au clavier. A la fin, le programme doit afficher le nombre total des employés trouvés.
3. **(6 points)** Créer la procédure PL/SQL stockée **AugmenteSalaire** (fichier **AugmenteSalaire\_Proc.sql**) qui augmente les salaires de tous les employés d'un département donné en paramètre à l'exception du chef à un taux donné en paramètre, si le total des salaires de tous les employés de ce département ne dépasse pas 150000. Tester cette procédure dans un bloc anonyme (fichier **Proc.sql**).
4. **(6 points)** Créer une fonction PL/SQL stockée **MaxTotalSalaires** (fichier **MaxTotalSalaires\_Fonc.sql**) qui retourne la valeur maximale des totaux des salaires des anciens employés par département. Les dates limites pour définir un ancien employé sont données en paramètres. Tester cette fonction dans un bloc anonyme (fichier **Fonc.sql**).
5. **(6 points)** Créer le déclencheur **DateEmbauche** de BD qui vérifie si la date d'embauche d'un employé est ultérieure à la date de création du département dans lequel l'employé est embauché (au moment d'insertion ou de modification).