|  |  |
| --- | --- |
| ***OFPPT*** |  |
| **Office de la Formation Professionnelle**  **et de la Promotion du Travail** |

**Contrôle Continue 1**

**Module :** Bases de données

|  |  |
| --- | --- |
| **Filière : DD 2ème année Année de formation : 2023/2024** | ***Epreuve : théorique*** |
| **Niveau : TS** | **Variante : 3** |
| **Durée : 2h00** | **Barème : 20 Pts** |

**NB : La mauvaise présentation de la copie peut être sanctionnée par -2**

## Exercice 1 (6 points)

Une entreprise désire développer une application permettant la création d’agendas électroniques partagés entre ses employés. Chaque employé peut utiliser son agenda pour saisir ses activités personnelles et aussi pour modifier ou supprimer ces activités. L’agenda d’un employé peut être consulté par tous les employés. Les chefs de services peuvent utiliser l’application pour inscrire des activités personnelles dans l’agenda des employés qui travaillent dans leur service. Le directeur des ressources humaines utilise l’application pour inscrire des activités à exécuter dans l’agenda des chefs de service. Un administrateur utilise l’application pour créer ou supprimer les agendas.

Un employé est identifié par un matricule, le nom, le prénom et adresse email. Chaque employé travaille dans un service unique qui est caractérisé par un code et un nom. Un service possède un chef de service. Chaque employé a un seul agenda qui est identifié par le code et la date de création. Une activité est caractérisée par un type d’activité (exemple « exposé », « réunion », « séminaire »…), par l’heure de début, l’heure de fin de réalisation de l’activité et par une description (exemple « exposé sur les nouveaux produits »).

Une activité peut être unique ou périodique (chaque semaine). Une activité unique possède en plus la date de réalisation. Une activité périodique est caractérisée par les jours de la semaine concernés par cette activité (exemple chaque ‘lundi’ et ‘jeudi’). Un employé peut inscrire une activité privée (non professionnelle comme « rendez-vous médecin » …) dans son agenda ; dans ce cas cette activité est marquée par l’information « non visible » (les autres employés ne peuvent pas la consulter).

1) Établir le Modèle conceptuel des données de cette application. (3 pts)

2) En déduire le Modèle Logique de données. (2 pts)

3) Tracez le dictionnaire des données de l’application (1 pt)

# Exercice 2 (14 points)

Pour gérer les élections des représentants des employés d’une entreprise, une application utilise une base de données composée des tables Electeurs, Candidats et Votes :

**Electeur** (**idElecteur** , nomElecteur, prenomElecteur, aVoté)

Le champ aVoté prend la valeur 0 quand l’électeur n’a pas encore voté et 1 quand il a voté.

**Candidat**(**idCandidat**,nomCandidat,prenomCandidat,dateNaissance, nombreVoix)

nombreVoix est le nombre de voix obtenus par le candidat.

**Votes** **(#idElecteur , #idCandidat**)

Cette table enregistre les votes des électeurs.

1) Ecrire une **fonction Q1** qui retourne le nombre des candidats de la base de données. **(2 pts)**

2) Ecrire une **fonction Q2** qui retourne le **nom complet (concatenation du nom et du prénom)** d’un électeur dont l’identifiant est passé comme paramètre. **(2 pts)**

3) Ecrire une **vue Q3** qui affiche pour chaque candidat la liste de ces électeurs qui ont voté pour lui triés par nom et par âge de candidat des plus vieux au plus jeunes **(idCandidat, nomCandidat, prenomCandidat, idElecteur nomElecteur, prenomElecteur) (2pts)**

4) Ecrire une **fonction** **Q4** qui accepte comme paramètres le nombre des votes et qui retourne le **nombre des candidats** qui ont obtenus plus que ce nombre. **(2 pts)**

5) Ecrire une **procédure stockée Q5** qui accepte le **id d’un candidat** et qui affiche la liste des candidats qui ont obtenus le même nombre de votes que lui (le candidat dont l’id est passé comme paramètre ne doit pas exister dans cette liste). **(2 pts)**

6) Ecrire un trigger **Q6** qui permet d’incrémenter de 1, le champ nombreVoix d’un candidat à chaque ajout d’une ligne à la table Votes qui concerne ce candidat. Le trigger doit également mettre le champ aVoté à 1 pour l’électeur qui vient de voter. **(2 pts)**

7) Ecrire une procédure stockée Q7 qui permet d’enregistrer le vote d’un électeur ;

Cette procédure sera appelée comme suit : **call Q7 (130,120,300,null)**

elle a les paramètres :

* **idElect** : identifiant de l’électeur. (130 dans notre exemple)
* **idCandidat1** (120), **idCandidat2**(300) et **idCandidat3**(null) : identifiants des 3 candidats choisis par l’électeur.

La procédure ajoute 1 à 3 lignes à la table Votes selon les valeurs non NULL des paramètres **idCandidat1**, **idCandidat2** et **idCandidat3**.

Si ces paramètres sont tous NULL, la procédure affiche un message d’erreur. **(2 pts)**