

Laboratoire_2

Date de remise : le 28 septembre

Objectifs :

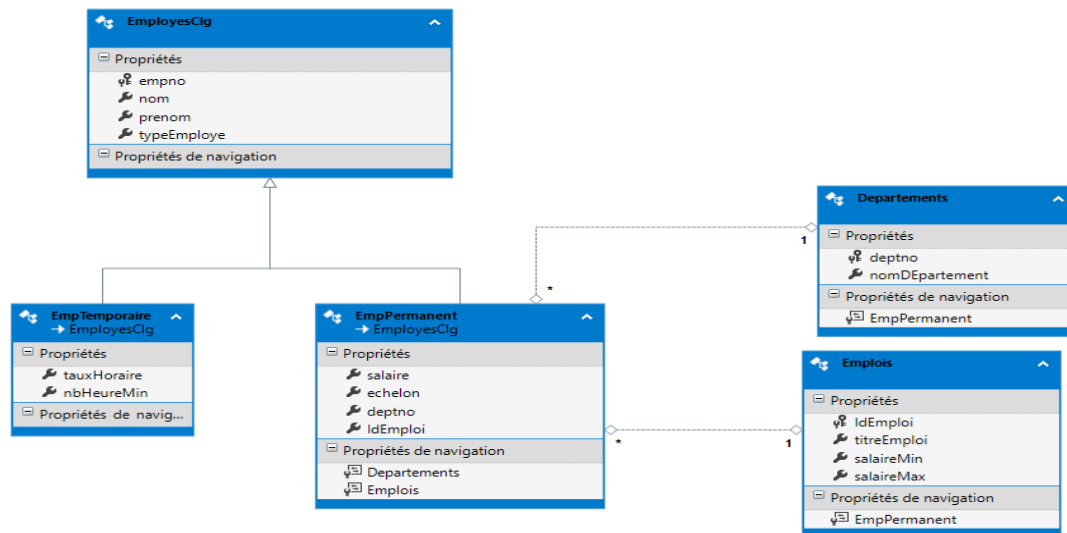
1. Introduction au langage Transact-SQL
2. Écrire et exécuter des procédures stockées

Consignes :

1. Vos requêtes doivent être enregistrées dans un fichier de nom Laboratoire2.sql. Toutes vos procédures doivent être parfaitement identifiées par leur numéro.
2. Vous devez également remettre un fichier d'exécution identifié par le nom Execution.sql. Ce fichier d'exécution doit contenir toutes vos requêtes EXECUTE et SELECT
3. Vous devez le déposer dans la boîte à la date et à l'heure indiquée.

Énoncé :

La base de données EmpClgDB contient les tables représentées par le diagramme suivant : (ce diagramme sera vu plus loin dans la session, mais peut-être, qu'il vous fait penser à des notions que vous avez déjà vues)



Attention ! ce qui suit est très important pour comprendre ce que vous devez faire

Dans cette base de données les informations des employés sont stockées dans 3 tables :

1. La table `EmployesClg` qui contient toutes les informations communes aux employés.
2. La table `EmpPermanent` qui contiendra les informations des employés permanent uniquement
3. La table `EmpTemporaire` qui va contenir les informations des employés temporaires uniquement.
4. Les trois tables ont la même clé primaire : `empno`.

Ce qui veut dire que pour connaître toutes les informations d'un employé permanent par exemple, nous avons besoin d'accéder à deux tables. `EmployesClg` et `EmpPermanent`.

Lorsque nous faisons des insertions, il faudra s'assurer :

1. Un numéro inséré dans la table `EmployesClg` doit être dans la table `EmpPermanent` **ou** `empTemporaire`. **Jamais dans les deux.**
2. Les numéros des `EmpPermanent` et la table `empTemporaire` se trouvent dans la table `EmployesClg`.

Lorsque nous faisons des suppressions, il faudra s'assurer :

1. Un numéro supprimé dans la table `EmployesClg` doit être supprimé dans la table `EmpPermanent` ou la table `empTemporaire`.

Question 1 : Création des tables et insertion

1. Lisez le script de création des tables contenu dans `laboratoire2.sql`. Le but est de voir les contraintes d'intégrités et les liens entre les tables.
2. Exécuter le script `laboratoire2.sql` pour créer les tables de la Base de données `EmpClgDB` et les tables.
3. Faire les insertions dans chacune des tables.

Question 2 : Procédures et fonctions

1. Écrire la procédure stockée ***ajouterEmployePermanent*** qui permet d'ajouter un employé permanent.
2. Écrire la procédure stockée ***ajouterEmployetemporaire*** qui permet d'ajouter un employé temporaire

Pour les questions 1 et 2 vous devez vous assurer que la transaction (les insertions) soit faite uniquement lorsque toutes les données sont valides.

3. Écrire la procédure stockée ***supprimerEmploye*** qui permet de faire une suppression en cascade d'un employé. La suppression se fait selon son numéro de l'employé
4. Écrire une procédure stockée ***listeEmployeselonTypeemploye*** qui affiche la liste des employés selon leur type d'employé (`typeEmploye`). Afficher le nom, le prénom. Puis pour les employés permanent afficher les salaires, le nom du département et le titre de

leur emploi. Pour les employés temporaires afficher le taux horaire et le nombre d'heure minimum. Ici il est recommandé d'utiliser des vues.

5. Écrire une fonction table **moyenneSalaireParDepartement** , qui retourne pour chaque département (nom du département) ainsi que la moyenne des salaires dans ce département.
6. Écrire une fonction scalaire **totalEmployeParDepartement** qui retourne le nombre d'employés de chaque département. Le nom du département est fourni en paramètre
7. Donner le script d'exécution. Celui qui contient l'exécution de vos procédures et fonction.

Questions optionnelles : Pour vous exercer un peu plus :

1. Écrire une procédure stockée **listeEmployeselonTitreEmploi** qui affiche le nom, le prénom des employés selon le titre de l'emploi (titreEmploi)
2. Écrire une procédure stockée **supprimerDepartement** qui fait la suppression d'un département selon le code département et qui met à NULL le code département des employés de ce département .
3. Écrire une fonction **TotalEmployeselonTypeEmploye** qui le nombre d'employé selon le typeEmploye.