

Laboratoire 2

Date à laquelle le laboratoire doit être terminé : le 19 septembre

Objectifs :

1. Introduction au langage Transact-SQL
2. Écrire et exécuter des procédures stockées
3. Expérimenter le concept de transactions

Important !

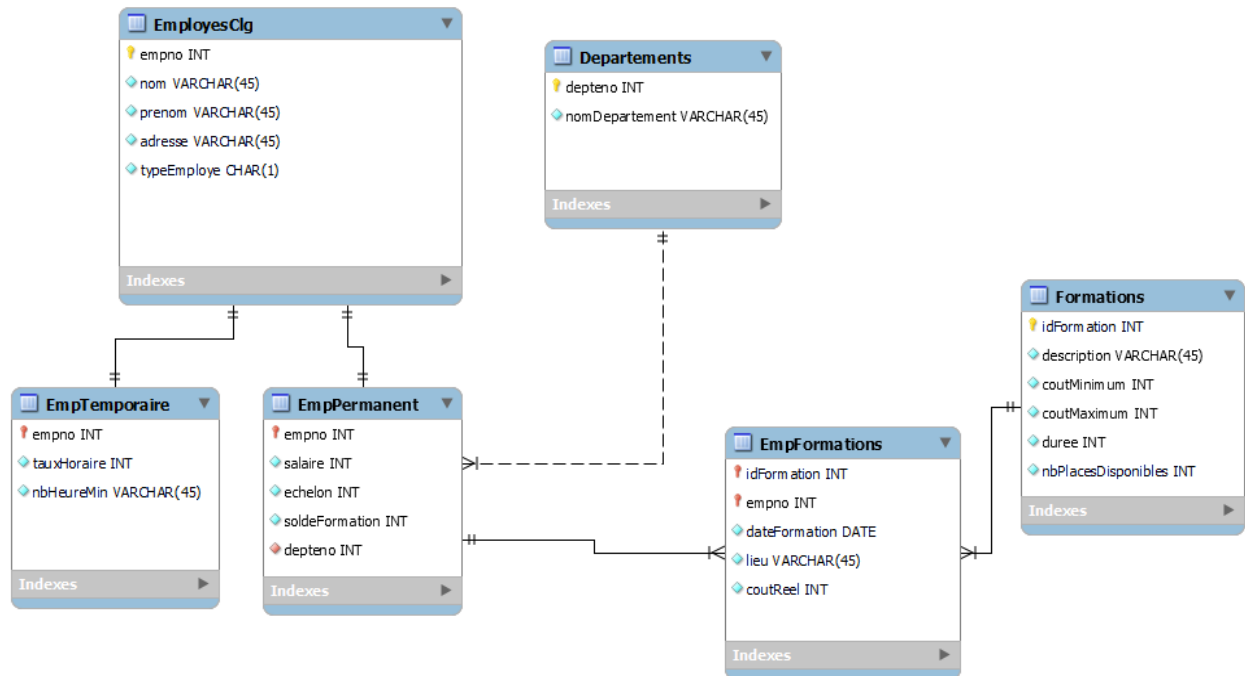
- Ce laboratoire est formatif. (Pas de points) ce qui vous enlève du stress et vous permet de le faire par vous-même. Vous avez le droit de vous entraider.
- Un quiz suivra ce laboratoire pour garantir votre compréhension de celui-ci
- La réalisation de ce laboratoire est requise pour réaliser le TP1, qui vaut 20%

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 avoir 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.
3. Le salaire des employés du département « Informatique » est toujours plus élevé que le salaire des employés du département « Achats »

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

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

À chaque fois qu'un employé suit une formation, le nombre de places disponibles doit diminuer de 1, et le solde disponible pour suivre une formation (soldeFormation) diminue du coût de la formation (coutReel. Un employé suit une formation, cela veut dire qu'il est inséré dans la table EmployesFormations.

Un trigger devra garantir que le salaire des employés en informatique est plus élevé que le salaire des employés du département achat. **(Laboratoire 3)**

Pour vérifier que le cout réel de la formation est entre le coutMinimum et le coutMaximum un mécanisme supplémentaire (**trigger**) sera implémenté **au laboratoire 3**.

Question 1 : Création des tables et insertion

1. Lisez le script de création des tables contenu dans ScriptLabo2.sql. Le but est de voir les contraintes d'intégrités et les liens entre les tables.
2. Exécuter le script ScriptLabo2.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. Ici il faut faire les insertions dans les deux tables EmployesClg et EmpPermanent
2. Écrire la procédure stockée **ajouterEmployetemporaire** qui permet d'ajouter un employé temporaire : Ici il faut faire les insertions dans les deux tables EmployesClg et EmpTemporaire

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

3. Écrire la procédure **insertEmpFormation** qui reçoit comme paramètres : @idformation, @empno, @dateFormation, @lieu, @cout qui permet d'attribuer une formation à un employé. On fournit tous les paramètres à la procédure. À la fin de cette transaction votre base de données doit rester dans un état COHERENT
4. 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.
5. 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. Pour les employés temporaires afficher le taux horaire et le nombre d'heure minimum. Ici il est recommandé d'utiliser des vues.

Attention : Les vues doivent être créées en dehors de la procédure stockée.

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