

## Examen Base de Données

Considérez les tables suivantes :

- Employee (employee\_id :varchar2, employee\_name :varchar2, hire\_date : date, department : varchar2, salary : number(5,2))
- Project (project\_id :varchar2, project\_name :varchar2, start\_date : date, end\_date : date)
- Employee\_Project (#employee\_id, #project\_id, hours\_worked :number)

**Exercice 1** : Ecrire les requêtes SQL suivantes, permettant de :

- Créer l'utilisateur **guest11** identifié par le mot de passe "**Gt11**" et lui attribuer tous les droits d'accès pour se connecter et créer des ressources dans la base de données
- Se connecter avec cet utilisateur et créer les tables : **Employee**, **Project** et **Employee\_Project** en tenant compte que des contraintes suivantes :
  - les champs "hire\_date", "departement" doivent être non nuls.
  - Le champ "hours\_worked" doit avoir des valeurs positives.

**Exercice 2** : Ecrire les requêtes SQL suivantes, permettant de :

- Créer un point de sauvegarde nommée « UpdateEmpProjet »
- Augmenter le salaire pour les employés E165, E254 et E380 de 15%.
- Supprimer les projets P523 et P750 de la table **Project**.
- Annuler le point de sauvegarde créé dans la première question
- Se déconnecter de l'utilisateur **guest11**, se connecter à l'administrateur du SGBD et créer un nouvel utilisateur identifié par "**UserX2**" et ayant le mot de passe "**UX789**".
- Attribuer à l'utilisateur "**UserX2**" les droits de sélection, de modification et de suppression de données sur la table **Project**
- Se connecter avec "**UserX2**" et Insérer les tuples ci-dessous dans la table **Project** de l'utilisateur **guest11** :

Table	Tuples
Project	('P821', 'Centre Médical', '01/02/2024', '25/08/2025') ( 'P822', 'Super marché', '15/02/2024', '25/10/2025')

- Valider toutes ces insertions.

**Exercice 3 :** Ecrire les commandes SQL permettant de rechercher :

1. Afficher les projets qui sont encore en cours (CURRENT\_DATE).
2. Afficher les heures travaillées par employé sur chaque projet.
3. Afficher les employés qui ont travaillé sur au moins deux projets différents.
4. Afficher les employés ayant le salaire le plus élevé dans chaque département.

**Exercice 4 :**

1. Créez un curseur explicite qui affiche le nom et le salaire de tous les employés dont le salaire est supérieur à 3700.

Écrire le bloc ci-dessus avec deux versions : LOOP et FOR.

2. Créez une fonction stockée en PL/SQL qui prend en paramètre le nom d'un département et retourne le nombre total d'employés dans ce département. Gérer l'exception si aucun enregistrement n'est trouvé (NO\_DATA\_FOUND), et renvoyer le résultat.
  3. Créer une procédure stockée qui prend en paramètre le nom d'un projet et met à jour la table "Project" en définissant la date de fin du projet sur la date actuelle (SYSDATE).
  4. Soient les tables suivantes :
    - Employee (employee\_id :varchar2, employee\_name :varchar2, hire\_date : date, department :varchar2, salary :number(5,2))
    - Employee\_Project (#employee\_id, #project\_id, hours\_worked :number)
    - Formation (id Formation : varchar2, #employee\_id, durée : date)
- i. Ecrire un déclencheur (Trigger) PL/SQL qui vérifie qu'un employé de la société doit suivre une formation avant d'effectuer un projet. Si cette contrainte n'est pas respectée une exception RAISE\_APPLICATION\_ERROR sera déclenchée ayant comme code -20121, et le message : 'cet employé doit être formé avant d'effectuer ce projet !!!!'

*Bon travail*