



Département Informatique

Filières SMI5/LPII

**Module  
Bases de données**

**Travaux pratiques**

**Pr. Mustapha Machkour**

Année universitaire 2019/2020

## Table des matières

TP 1 : Langage de Définition des Données.....	3
TP 2 : Langage de Manipulation des Données .....	5
TP 3 : Langage de Manipulation des Données (2) .....	7
TP 4 : Langage de Manipulation des Données (3) .....	9
TP 5 : Intégrité et sécurité .....	10
TP 6 : Les triggers.....	11

### TP 1 : Langage de Définition des Données

On considère les relations suivantes d'une base de données :

Auteur ( codeAuteur, nom, prénom, nationalité )

Livre (codeLivre, titre, nbPages, année, prix)

Membre ( codeMembre, nom, prénom, dateInscription )

Possède (codeLivre, codeAuteur)

Prêt (codePrêt,codeMembre, codeLivre, datePrêt, dateRetour).

[Q1] Lancer l'interface SQL\*plus, en utilisant l'une des deux manières suivantes :

- a. "SQL command" du groupe de programmes d'Oracle ;
- b. A l'interface ligne de commande, tapez la commande : sqlplus /nolog.

Les commandes correctement exécutées doivent être enregistrées dans un fichier.

[Q2] Se connecter au serveur de la base Oracle en utilisant la commande :  
connect system/motdepasse.

[Q3] Créer un utilisateur qui s'appelle TP1votreNOM. Utiliser la commande :  
create user tp1 identified by motdepasse quota 10 M on system;

[Q4] Donner à cet utilisateur les droits : "create session" et "create table".

[Q5] Se déconnecter du serveur de la base Oracle avec la commande : disconnect.

[Q6] Se connecter de nouveau à oracle en utilisant le login de l'utilisateur créé.

[Q7] Préciser les clés primaires et les clés étrangères des relations ci-dessus.

[Q8] Créer les tables de cette base de données. Les contraintes doivent être nommées de la manière suivante :

InitialeTypeContrainte\_\_nomTable[\_nomTableReference].

[Q9] En utilisant la vue user\_objects, vérifier la présence des objets créés. Utiliser :  
select object\_name, object\_type from ...

[Q10] Créer un index sur une clé primaire et un autre sur une des clés étrangères de la table prêt. Que peut-on déduire ?

[Q11] Augmenter la taille de la colonne nom de la table membre.

[Q12] Ajouter le champ email à la table membre.

[Q13] Ajouter la colonne adresse à la table membre.

[Q14] Supprimer les contraintes de la table prêt.

[Q15] Modifier la table prêt en y ajoutant les contraintes suivantes :

- a. codeMembre est une clé étrangère de la table membre.
- b. codeLivre est une clé étrangère de la table livre.

[Q16] Ajouter la contrainte de clé primaire à prêt.

[Q17] Modifier la table livre en y ajoutant la colonne codeAuteur et la contrainte indiquant que cette colonne est une clé étrangère.

[Q18] Supprimer la table possède.

- [Q19] Ajouter une contrainte à la table livre qui assure que titre de livre est une valeur bien déterminée.
- [Q20] Ajoute les champs langue et NbreExemplaires à la table livre.
- [Q21] Ajouter la contrainte qui assure l'unicité du nom et prénom du membre.
- [Q22] Créer les index sur les champs suivants :
- a. nom et prénom de l'auteur,
  - b. nom et prénom du membre,
  - c. titre du livre.
- [Q23] Supprimer l'index déjà créé sur la table auteur.
- [Q24] Ajouter la contrainte qui spécifie que dateRetour est  $\geq$  datePret.
- [Q25] Afficher les contraintes en utilisant la vue userConstraints.

## TP 2 : Langage de Manipulation des Données

Nous considérons ci-dessous les tables d'une base de données supposée déjà construite:

**Etudiant** (codeEtudiant, nom, prénom, dateNaissance, ville)

codeEtudiant : entier,  
nom : caractères variable de longueur 20,  
prénom : caractères variable de longueur 20,  
dateNaissance : date,  
ville : caractère variable de longueur 10.

**Cours** (codeCours, nom, nbHeures)

codeCours : entier,  
nom : caractère variable de longueur 20,  
nbHeures : entier sur 2 chiffres.

**Enseignant** (codeEnseignant, nom, prénom, spécialité)

codeEnseignant : entier,  
nom : caractère variable de longueur 20,  
prénom : caractère variable de longueur 20,  
spécialité : caractère variable de longueur 20.

**Résultat** (codeEtudiant, codeCours, note)

note : nombre de longueur 5 avec 2 chiffres après la virgule.

**Charge** (codeCours, codeEnseignant)

Les commandes correctement exécutées doivent être enregistrées dans un fichier à rendre.

### LDD

- [Q1] Construire un utilisateur nommé Etud avec quota dans l'espace de table users.
- [Q2] Accorder à cet utilisateur les privilèges 'all privileges'.
- [Q3] Indiquer les clés primaires et les clés étrangères de chaque table.
- [Q4] Construire les tables dans le schéma de Etud.

### LMD

- [Q5] Insérer des données dans les tables en utilisant la commande Insert.
- [Q6] Faire les mises à jour suivantes (n'oubliez pas de valider les mises à jour par la commande commit) :
  - a. Augmenter de 5% toutes les notes des étudiant.
  - b. Changer le nom du cours 2 par 'SGBD'.
  - c. Supprimer le cours n° 5.

### LID

- [Q7] Afficher séparément les tables déjà créées.
- [Q8] Afficher le nom et la ville des étudiants.
- [Q9] Afficher le nom et la ville des étudiants ordonnés par nom.
- [Q10] Afficher le nom des étudiants des villes agadir et tiznit.

- [Q11] Afficher le nom et la ville des étudiants des villes agadir, tiznit et taroudant ordonnés par ville et par nom.
- [Q12] Afficher le nom et l'âge des étudiants ordonnés par ville et par âge descendant.
- [Q13] Afficher le nom et l'âge des étudiants dont l'âge est compris entre 17 et 20, et ordonnés par âge descendant.
- [Q14] Afficher le nom et la ville des étudiants venants de Casa, Taroudant ou Safi.
- [Q15] Lister les étudiants dont le nom contient 'Ben' ou 'oui'.

FSA/Département Informatique/SM15-LP21

### TP 3 : Langage de Manipulation des Données (2)

Nous reprenons les tables de la série 2 :

- Etudiant (codeEtudiant, nom, prénom, dateNaissance, ville)
- Cours (codeCours, nom, nbHeures)
- Enseignant (codeEnseignant, nom, prénom, spécialité)
- Résultat (codeEtudiant, codeCours, note)
- Charge (codeCours, codeEnseignant)

Répondre en SQL aux requêtes suivantes :

NB. Les requêtes justes doivent être sauvegardées dans un fichier à rendre.

- [Q1] Afficher le nom, le prénom et la date de naissance (le nom du jour, celui du mois et l'année) des étudiants.
- [Q2] Calculer de deux façons différentes l'âge moyen des étudiants.
- [Q3] Afficher le nombre d'étudiants de l'établissement et le moyen de leurs âges.
- [Q4] Afficher le nombre d'étudiants de l'établissement, le minimum, le maximum et la moyenne de leurs âges.
- [Q5] Lister par ville le nombre d'étudiants.
- [Q6] Donner pour chaque étudiant la moyenne, le maximum et le minimum de notes.
- [Q7] Donner pour chaque étudiant des villes Agadir et Tiznit la moyenne, le maximum et le minimum de notes.
- [Q8] Lister le nom des étudiants ayant la moyenne comprise entre 12 et 16.
- [Q9] Lister, ordonnés par moyenne de notes, le nom des étudiants ayant la moyenne comprise entre 12 et 16.
- [Q10] Donner la liste de noms des enseignants dont le nom contient 'i' à la fin. On préférera que la première lettre du nom soit en majuscule et les autres lettres soient en minuscules.
- [Q11] Donner le nom des enseignants qui assure plus de deux cours.
- [Q12] Afficher le nom et la spécialité des enseignants dont la spécialité est connue.
- [Q13] Lister les noms des enseignants ayant la même spécialité.
- [Q14] Lister les noms des enseignants qui font le même cours.
- [Q15] Créer une autre table ETUDIANTS semblable à Etudiant et la charger par des tuples dont les codes sont distincts de ceux de ETUDIANT. Faire l'union de ETUDIANTS et ETUDIANT.
- [Q16] Faire l'union de ETUDIANT et de ETUDIANTS pour la ville d'Agadir.
- [Q17] Faire l'union de : les étudiants de la ville de casa de la table ETUDIANT et les étudiants de la ville d'agadir de la table ETUDIANTS).
- [Q18] Donner le nom la moyenne le minimum et le maximum de notes de chaque étudiant.
- [Q19] Donner le nom la moyenne le minimum et le maximum de notes de chaque étudiant de la ville Agadir.

- [Q20] Donner le nom la moyenne le minimum et le maximum de notes de chaque étudiant dont la moyenne est supérieure à 11.
- [Q21] Donner le nom la moyenne le minimum et le maximum de notes de chaque étudiant de la ville Agadir et dont la moyenne est supérieure 12.
- [Q22] Afficher les enseignants qui assurent tous les cours sauf le cours de réseaux.
- [Q23] Vérifier que les enseignants ayant une charge, figurent bien dans la table Enseignant.
- [Q24] Construire une vue qui va contenir les codes, les noms et les moyennes des étudiants de la ville Agadir.
- [Q25] Afficher le contenu de la vue créée à la question précédente.



### TP 4 : Langage de Manipulation des Données (3)

Soit les tables suivantes :

Usine (NU, NomU, Ville)

Produit (NP, NomP, Couleur, Poids)

Fournisseur (NF, NomF, Statut, Ville) et

Livraison (NP, NU, NF, quantité)

#### LDD

[Q1] Construire un utilisateur nommé tp4 et lui donner des droits pour créer des tables.

[Q2] Après avoir indiqué les clés primaires et étrangères, construire les tables ci-dessus dans le schéma de tp4.

#### LMD

[Q3] Remplir les tables par des données vraisemblables.

#### LID

[Q4] Donner en capitale le nom et la ville de toutes les usines.

[Q5] Donner le numéro, le nom et la ville de toutes les usines dont le nom contient plast.

[Q6] Donner les numéros des fournisseurs qui approvisionnent l'usine n°1.

[Q7] Donner le nom des fournisseurs qui approvisionnent l'usine n°1 en produit n°1.

[Q8] Donner les numéros des fournisseurs qui approvisionnent les usines n°1 ou n°2.

[Q9] Donner les numéros des usines qui ne reçoivent aucun produit du fournisseur n°1.

[Q10] Donner les numéros des usines qui s'approvisionnent chez le fournisseur n°3.

[Q11] Donner les numéros des fournisseurs qui fournissent au moins un produit fourni par le fournisseur n°1.

[Q12] Donner les numéros des usines qui s'approvisionnent uniquement chez le fournisseur n°3.

[Q13] Chercher pour chaque fournisseur le nom et le nombre d'usines qu'il approvisionne.  
L'affichage sera ordonné par nombre d'usines.

[Q14] Chercher le nom des fournisseurs qui approvisionne plus de 4 usines.

[Q15] Trouver le nom des produits fournis à plus de 4 usines d'Agadir.

[Q16] Afficher pour chaque produit le nom (seule l'initiale en majuscule) et la quantité totale fournie aux usines.

[Q17] Chercher, ordonnée par nom, les fournisseurs qui approvisionnent toutes les usines.

**TP 5 : Intégrité et sécurité**

- [Q1] Construire un utilisateur nommé TP4 et lui accorder uniquement les droits de se connecter et créer des tables dans son schéma.
- [Q2] Construire les relations suivantes dans le schéma de tp4, en respectant les contraintes données ci-après (toutes les contraintes doivent être nommées):  
Emp(nemp, nom, prénom, sexe, salaire, primes, taxes, nserv) et  
Service (nserv, libellés, locaux, nchefs)
- i. nemp est clé primaire ;
  - ii. nserv est clé primaire dans service et est étrangère dans emp (indiquer aussi l'action référentielle qui assure la suppression en cascade) ;
  - iii. nchefs est clé étrangère dans la relation service ;
  - iv. Les valeurs possibles de sexe sont : null, f, et m ;
  - v. Salaire est  $\geq 2000$  ;
  - vi. Les primes sont inférieures à 20% de salaire ;
  - vii. Les taxes sont comprises entre 20% et 40 % de la somme de salaire et de prime ;
- [Q3] Afficher les contraintes (nom, type et texte) des tables emp et service.
- [Q4] Insérer un service et des employés avec des salaires  $<$  et  $>$  10000
- [Q5] Construire un autre utilisateur TP, en utilisant le compte system, et lui accorder uniquement le droit de se connecter à la base.
- [Q6] Accorder, à partir du compte de TP4, à l'utilisateur TP, le droit de lire le nom et le prénom des employés ayant un salaire  $> 10000$ .
- [Q7] Vérifier si l'utilisateur TP peut exercer les droits accordés à la question précédente.
- [Q8] Retirer de TP les droits accordés à la question 7.
- [Q9] Construire une vue qui permet à TP de lire le nom, le prénom et le salaire des employés ayant un salaire  $< 10000$  mais uniquement le nom et le prénom des employés ayant un salaire  $\geq 10000$  (remplacer les salaires de ces derniers par une chaîne 'XXXX').
- [Q10] Vérifier si TP peut exercer ce droit.

## TP 6 : Les triggers

Cette série est basée sur la table emp de la série précédente.

- [Q1] Construire la table services définie par le schéma suivant :  
Services (nserv, libellés).
- [Q2] Ajouter à emp le champ nserv sans indiquer de contrainte référentielle.
- [Q3] Construire un trigger SupprCasc qui réalise la suppression en cascade entre services et emp.
- [Q4] Vérifier si le trigger SupprCasc est fonctionnel ou non.
- [Q5] Construire un trigger MajCasc qui réalise la mise à jour en cascade entre services et emp.
- [Q6] Vérifier si le trigger MajCasc est fonctionnel ou non.
- [Q7] Ajouter à la table services le champ nbre\_empls qui stocke le nombre d'employés de chaque service.
- [Q8] En utilisant une seule instruction update, mettre à jour le champ nbre\_empls.
- [Q9] Ecrire un trigger qui à la suite d'un ajout d'un employé à un service, incrémente le nbre\_empls de ce service. Vérifier la fonctionnalité de votre trigger.
- [Q10] IDEM que précédente, mais pour une opération de suppression.
- [Q11] IDEM que précédente, mais pour une modification de nserv dans emp.
- [Q12] Construire une relation audit\_emp contenant le champ de nemp et les champs :
  - opération: nature de l'opération effectuée sur emp (insertion, suppression et modification),
  - utilisateur: le nom de l'utilisateur qui a effectué l'opération,
  - date\_opération: date de l'opération.
- [Q13] Construire les triggers nécessaires qui alimentent la table audit\_emp et vérifier s'ils sont fonctionnels.