

EXERCICES : Pratiquons la manipulation

- Pour ces exercices, vous devez faire une copie des tables EMP et SERV.
- 121 : Augmenter de 10% ceux qui ont un salaire inférieur au salaire moyen. Valider.
- 122 : Insérez vous comme nouvel employé embauché aujourd'hui au salaire que vous désirez. Valider.
- 123 : Effacer les employés ayant le métier de SECRETAIRE. Valider.
- 124 : Insérer le salarié dont le nom est MOYEN, prénom Toto, no 1010, embauché le 12/12/99, supérieur 1000, pas de comm, service no 1, salaire vaut le salaire moyen des employés. Valider.
- 125 : Supprimer tous les employés ayant un A dans leur nom. Faire un SELECT sur votre table pour vérifier cela. Annuler les modifications et vérifier que cette action s'est bien déroulée.
- /*126 : Les verrous. Supprimer l'employé créé à l'exercice 122 de votre voisin. Ne pas valider. Vérifiez tous les deux le contenu de la table. Demander à votre voisin d'augmenter son propre salaire de 1000 €. Valider la suppression. Chacun vérifie l'action. Refaire l'exercice en échangeant les rôles.*/*

Le Langage de Manipulation de Données

Créer une table

- Deux façons de créer une table
- Pour créer une table à partir de rien
- **CREATE TABLE** nom table (col1 type1 [NOT NULL], col2 type2 [NOT NULL],)
[TABLE SPACE nomtbs]
- Pour créer une table à partir d'une autre table
- **CREATE TABLE** nom table [AS select];

Le Langage de Manipulation de Données

Ajouter une colonne.

- **ALTER TABLE nom_table ADD (col1 type1, col2 type2, ..);**
- La nouvelle colonne est placée à droite des anciennes.
- Tous les champs de la nouvelle colonne sont à NULL.

Modifier une colonne.

- **ALTER TABLE nom_table MODIFY (nom_col nouv_type);**
- le nouveau type doit être compatible avec l'ancien, on peut augmenter la taille maximale d'une colonne VARCHAR2(n) par exemple.
- on peut déclarer NOT NULL une colonne qui ne contient pas de valeur NULL.

Le Langage de Manipulation de Données

Renomme et supprimer une table

la commande **DROP TABLE**

- Permet de supprimer la table.
- Aucune récupération possible, la place est toute de suite réutilisée

la commande **RENAME**

- Permet de renommer une table
- EX: **RENAME TABLE serv to services;**
- Permet de renommer une colonne
- EX : **ALTER TABLE EMP
CHANGE EMPLOI EMPLOI1 int(5)**

(change ancien_nom nouveau_nom type_données)

EXERCICES : Créons !

- **127 : Créer les tables EMP1 et SERV1 comme copie des tables EMP et SERV.**
- **128 : Vérifier que la table PROJ n'existe pas.**
- **129 :**
 - Créer une table PROJET avec les colonnes suivantes:
numéro de projet (noproj), type numérique 3 chiffres, doit contenir une valeur. nom de projet (nomproj), type caractère, longueur = 10
budget du projet (budget), type numérique, 6 chiffres significatifs et 2 décimales.
 - Vérifier l'existence de la table PROJET.
 - Faire afficher la description de la table PROJET.
 - C'était une erreur!!! Renommez la table PROJET en PROJ
- **130 :**
 - Insérer trois lignes de données dans la table PROJ: numéros des projets = 101, 102, 103
noms des projets = alpha, beta, gamma budgets = 250000, 175000, 950000
 - Afficher le contenu de la table PROJ.
 - Valider les insertions faites dans la table PROJ.

EXERCICES : Créons !

- **131 :**
 - Modifier la table PROJ en donnant un budget de 1.500.000 Euros au projet 103.
 - Modifier la colonne budget afin d'accepter des projets jusque 10.000.000.000 d'Euros
 - Retenter la modification demandée 2 lignes au dessus.
- **132 :**
 - Ajouter une colonne NOPROJ (type numérique) à la table EMP.
 - Afficher le contenu de la table EMP.
- **133 : Affecter les employés du service 2 et les directeurs au projet 101.**
- **134 : Affecter les employés dont le numéro est supérieur à 1350 au projet 102, sauf ceux qui sont déjà affectés à un projet.**
- **135 : Affecter les employés n'ayant pas de projet au projet 103.**
- **136 : Sélectionner les noms d'employés avec le nom de leur projet et le nom de leur service.**
- **137 : Ecrire une requête qui affiche les noms de toutes les tables de la BDD 'emploi' « Use & Show »**

Le langage de définition de données

Les vues : la création

- Le résultat d'une commande select est une table virtuelle, cette table créée dynamiquement peut être vue comme une table réelle par les utilisateurs par les biais d'une vue.
- **CREATE VIEW nom_vue [(col1, col2,..... AS SELECT;**
- **Il n'est pas obligatoire de préciser les noms de colonnes, par défaut les colonnes de la vue ont les mêmes noms que les colonnes du select qui a défini cette vue, les colonnes résultat d'une expression doivent être renommées dans le select.**
- **EX: créer la vue SUBA de la table employée limitée au non directeurs.**
 - **create view SUBA**
as select * from emp
where emploi <> 'DIRECTEUR';
- **Généralement une vue est créée pour empêcher l'accès à certaines colonnes.**

Le langage de définition de données Les vues : l'utilisation

- Elle s'interroge comme une table
- Pour détruire une vue
 - **DROP VIEW** nom_vue;
- Pour renommer une vue
 - **RENAME** ancien_nom TO nouveau_nom;

EXERCICES : Ajustez votre vue

- **138** : Créer la vue EMP1 de la table EMP contenant les colonnes: numéro d'employé, nom et emploi et limitée aux employés du service numéro 1,
- **139** : Sélectionner cette vue EMP1 entièrement,
- **140** : Sélectionner les vendeurs de la vue emp1.
- **141** : Modifier la table EMP en transformant l'emploi de NYS en comptable.
- **142** : Sélectionner la vue EMP1.

Le Langage de Manipulation de Données

Les contraintes

- Il y a trois types de contraintes :
 - Déclaration de clef primaire
 - ⊙ **constraint pk_serv primary key (noserv)**
 - Contrainte d'appartenance à un domaine
 - ⊙ **constraint ck_sal check (SAL between 1 and 9999999)**
 - ⊙ **constraint ck_comm check (COMM >= 0)**
 - Contrainte référentielle (Clef étrangère)
 - ⊙ **constraint fk1_emp foreign key (SUP) references EMP(NOEMP)**
 - ⊙ **constraint fk2_emp foreign key (NOSERV) references SERV(NOSERV)**
- La clef peut être multiple, les champs participant à la clef sont alors séparés par des virgules.

Le Langage de Manipulation de Données

Les contraintes

- **La déclaration des contraintes se placent :**
 - Les contraintes de domaine : après la déclaration du champ
 - ⊙ **create table serv**
(noserv number(4) not null,
constraint ck_noser check(NOSERV between 1 and 9999,.....);
 - Les autres contraintes :
 - ⊙ Soit à la fin du create table :
 - ⊙ **Create table emp** (.....,
constraint pk_serv primary key (noserv),
constraint fk1_emp foreign key (SUP) references EMP(NOEMP), constraint fk2_emp
foreign key (NOSERV) references SERV(NOSERV));
 - Soit dans un alter table :
 - ⊙ **Alter table emp**
add constraint fk1_emp foreign key (SUP) references EMP(NOEMP);
- **Généralement la contrainte de clef primaire se trouve dans le create table, celle de clef étrangère dans un alter table, les "create table" sont regroupés en début de script.**

•Exercices de contraintes :

- Contrainte d'unicité de la table proj (NOPROJ)
- Contrainte clé étrangère table EMP (numéro de service)
- Contrainte de domaine table PROJ (budget > 20 000)
- Contrainte clé étrangère table EMP (numéro de supérieur)
- Contrainte de domaine table EMP (numéro de supérieur entre 0 et 9999)
- Contrainte de domaine table SERV (numéro de service < 10)