

TP3

DDL – Vues – contraintes

Quelques Commandes utiles

⇒⇒ A la découverte du dictionnaire d'Oracle

```
show user ;
select * from all_users;
desc user_tables ;
select table_name from user_tables;
column table_name heading Nom_de_l_utilisateur
select table_name from user_tables;
column object_name format a30
set pages 30
set lines 150
select object_name, object_type from user_objects order by object_type;
desc user_constraints;
column constraint_name format a30
select constraint_name from user_constraints where table_name='&matable';
```

⇒⇒ A la redécouverte de quelques fonctions SQL/Oracle

```
SELECT RPAD('Soleil',17,'bla') "RPAD exemple" FROM DUAL;
SELECT LPAD('DESS EID',15,'*.*') "LPAD exemple" FROM DUAL;
SELECT SUBSTR('DESS EID',6,3) "SUBSTR exemple" FROM DUAL;
SELECT SUBSTR('ABCDEFGH IJ',-5,4) "SUBSTR exemple" FROM DUAL;
SELECT TO_CHAR (SYSDATE, 'MM-DD-YYYY HH24:MI:SS') "Now" FROM DUAL;
SELECT LENGTH('WEB WAREHOUSE') "Longueur en caractères" FROM DUAL;
SELECT DECODE(ANNEE,1,'Première',2,'Seconde','Valeur différente de 1 et de 2 !!') AS
ANETUDE FROM ELEVES;
SELECT NVL(SPECIALITE, 'Valeur NON renseignée') FROM PROFESSEURS;
SELECT NVL(SPECIALITE, 'Valeur NON renseignée') AS SPEC_PROF FROM PROFESSEURS;
SELECT ROUND(17.0958,1) "ROUND exemple" FROM DUAL;
SELECT ROUND(17.58,2) "ROUND exemple" FROM DUAL;
SELECT TRUNC(1958.0917,1) "TRUNC exemple" FROM DUAL;
SELECT TRUNC(1958.0917,2) "TRUNC exemple" FROM DUAL;
SELECT ROUND(TO_DATE('17-SEP-2002'), 'YEAR') "New Year" FROM DUAL;
SELECT EXTRACT(YEAR FROM SYSDATE) FROM DUAL;
SELECT ADD_MONTHS(SYSDATE,7) FROM DUAL;
SELECT TRUNC(MONTHS_BETWEEN(SYSDATE, TO_DATE('19-JUN-2001'))) AS AGEBB FROM DUAL;
SELECT TO_NUMBER(TO_CHAR(SYSDATE, 'YYYY'))FROM DUAL;
```

Exercice 1 : DDL, valeurs nulles et valeurs par défaut

Soit la table Emp .

```
emp (  
  nom varchar2(10),  
  num number(4) not null primary key,  
  fonction varchar2(12),  
  n_sup number(5),  
  embauche date,  
  salaire number(5),  
  n_dept number(5)) ;
```

1) Un peu de nettoyage :

- Supprimer la table Emp si elle existe

2) Créer la table Emp tel que.

- Le nom de l'employé et son numéro n'acceptent pas la valeur nulle.
- la fonction peut prendre des valeurs nulles.
- La date d'embauche prend la date du système (SYSDATE) par défaut et elle est unique.
- Si le salaire n'est pas renseigné on met 9000 par défaut.
- Si le département n'est pas renseigné on met la valeur 0 par défaut

3) Insertion de données

- Insérer MARTIN 16712 (tableau ci-dessous) dans la table emp.
- Insérer DUPONT 17574 en essayant d'attribuer à la colonne nom la valeur nulle.
- Réinsérer DUPONT 17574 mais cette fois-ci sans renseigner la fonction.
- Insérer DUPOND 26691 sans renseigner le salaire (salaire prendra la valeur par défaut)
- Insérer l'employé numéro 17574 (on ne connaît pas son salaire)
- Vérifier si les insertions étaient correctement effectuées

NOM	NUM	FONCTION	N_SUP	EMBAUCHE	SALAIRE	N_DEPT
MARTIN	16712	administratif	25717	23-MAY-90	4000	30
DUPONT	17574		16712	03-MAY-95	1900	30
DUPOND	26691	commercial	27047	04-APR-88	900	20
LAMBERT	25012	administratif	27047	14-APR-91	1200	20
JOUBERT	25717	president		10-OCT-82	5000	30
LEBRETON	16034	commercial	27047	01-JUN-91	900	20
PAQUEL	27546	commercial	27047	03-SEP-83	2200	20
MARTIN	16712	directeur	25717	23-MAY-90	4000	30
DUPONT	17574		16712			
LEFEBVRE	25935	commercial	27047		2350	
GARDARIN	15155	ingenieur	24533	22-MAR-03		20
SIMON	26834		24533			20

3).

- ✓ Supprimer les employés dont la fonction est '**administratif**'
- ✓ Afficher le contenu de la table emp ;
- ✓ Faire un ROLLBACK ;
- ✓ Réafficher le contenu de la table emp.
- ✓ Supprimer l'employé ayant le plus grand salaire.
- ✓ Supprimer l'employé n'ayant pas un supérieur
- ✓ Supprimer la table emp.

Exercice 2 : Les vues

Soit le schéma relationnel de la base de données Ecole vue dans les TP précédents :

ELEVES(NUM_ELEVE, NOM, PRENOM, DATE_NAISSANCE, POIDS, ANNEE)
COURS(NUM_COURS, NOM, NBHEURES, ANNEE)
PROFESSEURS(NUM_PROF, NOM, SPECIALITE, DATE_ENTREE, DER_PROM,
SALAIRE_BASE, SALAIRE_ACTUEL)
ACTIVITES (NIVEAU,NOM , EQUIPE)

RESULTATS(NUM_ELEVE,NUM_COURS, POINTS)
CHARGE(NUM_PROF, NUM_COURS)
ACTIVITES_PRATIQUEES(NUM_ELEVE, NIVEAU, NOM)

Schéma relationnel de la BD Ecole

Remarque :

Créez la « BD » ECOLE ; Lancez le script SQL ecole.sql

Les vues :

Une vue est souvent appelée une "table virtuelle". Elle peut être interrogée exactement comme une table, par contre il existe des restrictions sur les mises à jour.

1. Création de vue à partir d'une table :

- a. Créer une vue Vue_El1 qui permet de donner les numéros, noms, prénoms et poids des élèves pesant moins de 50 kg.
- b. Afficher le contenu de la vue.

2. Impact d'une modification d'une relation sur une vue et vice versa :

- a. Mettre-à-jour la vue Vue_El1 en mettant le poids de 'Benoit' à 60 kg et insérer la ligne suivante 11, 'Berrabah', 'Djamel', '15-02-1973', 85, 2) ;
- b. Vérifier le résultat de ces deux opérations dans Vue_El1 et la table Eleves.
- c. Mettre-à-jour la vue Vue_El1 en remettant le poids de 'Benoit' à 35 kg et insérer le nouvel élève : (11, 'Achvar', 'Didier', 45) ;
- d. Vérifier dans Vue_El1 et la table Eleves.

- e. proposez un moyen pour que les mises-à-jours dans la vue Vue_El1 vérifient la condition de la requête de sa création.
- f. Faites des tests pour vérifier l'efficacité de ce moyen.

3. Création de vues Multi-tables :

- (a) Créer une vue Elev_Result contenant les numéros et les noms des élèves, les numéros et noms des cours qu'ils suivent et leurs résultats (points) dans ces derniers. Afficher son contenu.
- (b) insérer dans cette vue le n-uplet suivant : (11, 'Mark',6,'BDD',14);
- (c) qu'est ce que vous remarquez ?

4. Création de vue à partir d'une autre vue :

- (a) En utilisant la vue Vue_El1, créer une autre vue Vue_El1_admis donnant les noms des élèves admis, leurs cours et leurs résultats. Lister ensuite le contenu de cette vue.

5. Création de vue avec utilisation de champs calculés :

- (a) Créer une vue Vue_El1_Infos contenant les noms des élèves et leurs âges.
- (b) Lister ensuite ceux ayant plus de 35 ans.
- (c) Peut-on modifier cette vue ?

6. Création de vue en utilisant des agrégats :

- (a) Créer une vue Vue_El1_moyenne contenant pour chaque élève, son nom et sa moyenne générale.

7. Interrogation par jointure d'une table et une vue

- (a) Donner les noms des élèves admis et les noms de leurs activités.

Exercice 3 : Les Contraintes d'intégrité :

Soit le schéma relationnel suivant :

EMP(ENO, NOM, PROF, DATEEMB, SAL, COMM, DNO)

DEPT(DNO, DNOM, DIR, VILLE)

Remarques :

DNO dans EMP est le numéro du département de l'employé.

DIR dans DEPT est le numéro de l'employé directeur de ce département

Le script SQL de création de la base de données emploi est dans le fichier emp-dept.sql.

Ouvrir ce fichier et vérifier que les contraintes de clés primaires et de clés étrangères n'ont pas été intégrées et seront traitées dans les questions ci-dessous.

Pour valider les travaux en cours : **COMMIT ;**

1. Nettoyer la base de données en exécutant le fichier clean_DB.sql
2. Créer la base de données en exécutant le fichier emp_dept.sql
3. Vérifier la création et le remplissage des tables EMP et DEPT.

4. Visualiser la liste des contraintes en affichant le contenu de USER_CONSTRAINTS du dictionnaire de données. Quelles contraintes peut-on remarquer ?
5. Ajouter les clés primaires et les clés étrangères.
6. Tester l'insertion d'un employé dont on ne connaît que le nom (DUPONT), le salaire (600 €) et le département (30).
7. Modifier le schéma de EMP de manière à ce que l'attribut NUMEMP comporte 5 chiffres au maximum. Peut-on remettre NUMEMP à 4 chiffres ?
8. Modifier la table EMP de manière à ne pas permettre des salaires inférieurs à 800€. Tester la contrainte.
9. Modifier la table EMP de manière à ce que la date d'entrée soit par défaut la date courante. Tester la contrainte.
10. Vérifier, en affichant le contenu de DEPT, si NUMDEPT satisfait la contrainte de clé primaire dans la table DEPT.
11. Insérer un département ayant la même valeur de NUMDEPT qu'un département existant.
12. Tenter d'ajouter une contrainte sur la table DEPT définissant NUMDEPT comme clé primaire.
13. Supprimez le tuple que vous venez d'insérer. Retentez la modification définissant NUMDEPT comme clé primaire de DEPT. Testez cette contrainte en réinsérant à nouveau le tuple.
14. Modifier la table EMP de manière à ce que le cumul du salaire et de la commission soit obligatoirement compris entre 800 et 8000 €. Tester la contrainte.
15. Modifier la table EMP de manière à ce que le numéro de département NUMDEPT définisse une clé étrangère correspondant à NUMDEPT de DEPT. Tester la contrainte.
16. Modifier la table EMP de manière à ce que le numéro d'employé soit clé primaire. Tester la contrainte.
17. Tenter de définir la contrainte que COMM ne puisse pas avoir de valeur nulle.
18. Tenter de supprimer la contrainte de clé primaire sur DEPT (Utiliser la clause DROP? Avec et sans la clause CASCADE).
19. Y'a-t-il un moyen pour exprimer la contrainte que les départements ont tous des employés ?
20. Pour récapituler, visualisez la liste des contraintes en affichant la vue du dictionnaire de données TABLE_CONSTRAINTS.

Exercice 4 (Facultatif)

Parmi les rôles d'un administrateur de bases de données est la création des utilisateurs, le contrôle des droits attribués à ces derniers et la sécurité des données. Connectez vous avec deux comptes différents sur la même machine (ou travaillez par groupes sur deux machines différentes). Dans la suite de ce TP, les deux comptes seront appelés user1 et user2.

Chargez les fichiers dept.sql et emp.sql à partir de l'espace Campus.

User1 crée la table dept sur le compte de User2. Ce dernier crée la table emp sur le compte de user1.

Que se passe-t-il ?

Proposez une solution en essayant de donner le droit de création de table à User1 et User2. Est-il toujours possible de créer les tables ?

Chaque utilisateur crée les deux tables (dept et emp) sur son propre compte.

Chaque utilisateur (par exemple user1) crée la vue : nom des employés, lieu de travail à partir des table emp et dept de l'autre utilisateur (par exemple user2). Attention, pour accéder à une table de l'autre utilisateur, il faut la préfixer par son nom.

Vous avez remarqué qu'il y a un problème de droits. Donnez les droits nécessaires pour résoudre le problème.

Transmettez les droits que vous avez obtenus à un autre utilisateur.

S'il y a toujours un problème de droits, résolvez-le.

Chaque utilisateur donne la possibilité d'insérer, supprimer et modifier les enregistrements de sa table emp à l'autre utilisateur.

Donnez les requêtes qui permettent de vérifier les différentes possibilités de droits d'accès à cette table.

Chaque utilisateur donne la possibilité de mettre à jour uniquement la colonne LOC de sa table dept à l'autre utilisateur.

Donnez les requêtes qui permettent de vérifier que seule la colonne LOC est accessible.

Vérifier dans le dictionnaire de données (dba_tab_privs) la liste des privilèges de user1 et user2.

user1 remplace la valeur du nom de l'employé (de sa table emp) SMITH par la valeur KIMO.

User2 remet valeur du nom de l'employé (de la table emp de user1) à SMITH

Que se passe-t-il ?

user1 commit.

Que se passe-t-il ?

Créez une situation d'inter-blocage.