

TP n° 1 de Bases de Données / ORACLE

Création et restructuration d'une base

1) Travail préliminaire

Définir tous les chemins d'accès à ORACLE sur chaque répertoire de travail en ajoutant dans le .profile :

```
PATH=/usr/bin:/etc:/usr/sbin:/usr/ucb:$HOME/bin:/usr/bin/X11:/sbin/./usr/local/bin
export PATH
```

```
ORACLE_SID='bddo'
export ORACLE_SID
```

```
ORACLE_BASE=/usr/home/dba/oracle
export ORACLE_BASE
```

```
ORACLE_HOME=$ORACLE_BASE/product
export ORACLE_HOME
```

```
ORACLE_TERM=3151
export ORACLE_TERM
```

```
PATH=$PATH:$ORACLE_HOME/bin
export PATH
```

Accès à SQLPLUS depuis AIX : SQLPLUS ↵, puis donner votre *login* et votre *password*.

Sortie par EXIT ou QUIT.

Définir un éditeur dans SQLPLUS : `define_editor = /usr/local/bin/emacs .`

Utilisation : `edit <nomfichier>`.

Lancer une commande SQL depuis un fichier : `start cmd.sql ;`

Editeur incorporé à SQLPLUS :

Commande	Abréviation	Effet
APPEND chaîne	A chaîne	Ajoute la chaîne en fin de ligne courante
CHANGE /ch1 /ch2	C /ch1 /ch2	Change ch1 en ch2 dans la ligne courante
DEL	DEL	Supprime la ligne courante
INPUT	I	Ajoute des lignes après la ligne courante
LIST	L	Liste toutes les lignes du buffer
LIST n	L n	Liste la ligne n
LIST *	L *	Liste la ligne courante
LIST LAST	L LAST	Liste la dernière ligne
LIST m n	L m n	Liste les lignes m à n
CLEAR BUFFER	CL BUFF	Supprime toutes les lignes
RUN	RUN	Exécute la commande stockée dans le buffer
SAVE nomfichier	SAVE nomfichier	Sauvegarde le buffer dans un fichier
GET nomfichier	GET nomfichier	Charge le buffer avec le contenu d'un fichier

2) Création d'une base

Créer la base de données « FABRICATION » étudiée en cours. Peupler chaque table avec quelques tuples. **Ne pas introduire de contrainte d'intégrité dans la définition des tables.**

Base FABRICATION :

PIECE (NOP, DESIGNATION, COULEUR, POIDS) ;
SERVICE (NOS, INTITULE, LOCALISATION) ;
ORDRE (NOP, NOS, QUANTITE) ;
NOMENCLATURE (NOPA, NOPC, QUANTITE) ;

3) Contraintes d'intégrité

Tester l'insertion de tuples *incohérents* vis-à-vis des contraintes d'intégrité :

- insérer des pièces de même numéro dans la table PIECE ;
 - insérer des services de même numéro dans la table SERVICE ;
 - insérer des ordres dans la table ORDRE pour un service qui n'existe pas dans la table SERVICE ;
 - insérer dans la table NOMENCLATURE des tuples faisant référence à des pièces inexistantes dans la table PIECE.
-
- Ajouter à toutes les tables les contraintes d'intégrité concernant les clés primaires et étrangères.
 - Corriger ou supprimer les tuples incorrects.
 - Ajouter une contrainte de domaine sur l'attribut COULEUR, qui ne doit prendre que les valeurs ROUGE, VERTE, BLEUE et JAUNE.

4) Modification de la structure de la base

- Augmenter le type des attributs DESIGNATION et INTITULE de 10 caractères.
- Diminuer le type de l'attribut LOCALISATION de 10 caractères. *Prévoir le cas des troncatures.*
- Ajouter l'attribut DIRECTEUR à la table SERVICE.
- Supprimer l'attribut LOCALISATION de la table SERVICE.
- Supprimer la clé primaire de la table PIECE.
- Supprimer la contrainte de domaine sur l'attribut COULEUR.
- Supprimer la table NOMENCLATURE.
- Supprimer la table ORDRE en utilisant l'option CASCADE CONSTRAINTS et vérifier l'effet sur les tables PIECE et SERVICE.

2)

```
CREATE TABLE PIECE ( NOP NUMBER(3),
                     DESIGNATION CHAR(20),
                     COULEUR CHAR(15),
                     POIDS NUMBER(5,2) );
```

```
CREATE TABLE SERVICE ( NOS NUMBER(3),
                       INTITULE CHAR(20),
                       LOCALISATION CHAR(30) );
```

```
CREATE TABLE ORDRE ( NOP NUMBER(3),
                     NOS NUMBER(3),
                     QUANTITE NUMBER(3) );
```

```
CREATE TABLE NOMENCLATURE ( NOPA NUMBER(3),
                              NOPC NUMBER(3),
                              QUANTITE NUMBER(3) );
```

```
INSERT INTO PIECE VALUES (1, 'P1', 'ROUGE', 10.5);
```

...

3)

```
ALTER TABLE PIECE
ADD CONSTRAINT PIECE_CLP PRIMARY KEY (NOP);
```

```
ALTER TABLE SERVICE
ADD CONSTRAINT SERVICE_CLP PRIMARY KEY (NOS);
```

```
ALTER TABLE ORDRE
ADD CONSTRAINT ORDRE_CLP PRIMARY KEY (NOP, NOS);
```

```
ALTER TABLE ORDRE
ADD CONSTRAINT ORDRE_CLE1 FOREIGN KEY (NOP) REFERENCES PIECE (NOP);
```

```
ALTER TABLE ORDRE
ADD CONSTRAINT ORDRE_CLE2 FOREIGN KEY (NOS) REFERENCES SERVICE (NOS);
```

```
ALTER TABLE NOMENCLATURE
ADD CONSTRAINT NOM_CLP PRIMARY KEY (NOPA, NOPC);
```

```
ALTER TABLE NOMENCLATURE
ADD CONSTRAINT NOM_CLE1 FOREIGN KEY (NOPA) REFERENCES PIECE (NOP);
```

```
ALTER TABLE NOMENCLATURE
ADD CONSTRAINT NOM_CLE2 FOREIGN KEY (NOPC) REFERENCES PIECE (NOP);
```

```
ALTER TABLE PIECE
ADD CONSTRAINT PIECE_COUL
CHECK (COULEUR IN ('ROUGE', 'VERTE', 'BLEUE', 'JAUNE')) ;
```

4)

```
ALTER TABLE PIECE MODIFY (DESIGNATION CHAR(30)) ;
ALTER TABLE SERVICE MODIFY (INTITULE CHAR(30)) ;
```

```
CREATE TABLE TMP AS      Sauvegarde de la table SERVICE dans la table TMP
SELECT * FROM SERVICE;
```

```
UPDATE SERVICE          Suppression de la colonne LOCALISATION
SET LOCALISATION = NULL;
```

```
ALTER TABLE SERVICE    Restructuration de la table
MODIFY (LOCALISATION CHAR(20));
```

```
UPDATE SERVICE          Recopie de TMP vers SERVICE avec troncature
SET LOCALISATION = ( SELECT SUBSTR(LOCALISATION,1,20)
                     FROM TMP
                     WHERE TMP.NOS = SERVICE.NOS);
```

```
DROP TABLE TMP;        Suppression de la table temporaire TMP
```

```
ALTER TABLE SERVICE ADD (DIRECTEUR CHAR(20)) ;
```

```
ALTER TABLE SERVICE DROP UNIQUE (LOCALISATION) ;
```

```
ALTER TABLE PIECE DROP PRIMARY KEY ;
```

```
ALTER TABLE PIECE DROP CONSTRAINT PIECE_COUL ;
```

```
DROP TABLE NOMENCLATURE ;
```

```
DROP TABLE ORDRE CASCADE CONSTRAINTS ;
SELECT * FROM PIECE ;
SELECT * FROM SERVICE ;
```