

1. Exercice : Tablespaces

Préambule du corrigé : dans ce qui suit, on utilise le compte bdda pour créer les différents objets de la base de données.

1.1 Questions

En récupérant la définition de quelques tables de la documentation technique de la base de données d'AREL, on vous demande :

1. de créer un autre tablespace que Users nommé Users2. Il faut avoir le privilège système CREATE TABLESPACE.

La commande ci-dessous permet de créer le tablespace users2 associé au fichier 'C:\oracle\app\oracle\oradata\XE\users2.1.dbf' de taille 100K

```
CREATE TABLESPACE users2 DATAFILE 'c:\oracle\app\oracle\oradata\xe\users2.1.dbf' SIZE 100K;
```

2. de créer les différentes tables associées à ce tablespace.

On crée la table types_groupes dans le tablespace users2

```
CREATE TABLE TYPES_GROUPS (P_ID NUMBER NOT NULL, LIBELLE VARCHAR2(50 BYTE) NOT NULL, CONSTRAINT PK_TYPES_GROUPS PRIMARY KEY (P_ID)) TABLESPACE USERS2;
```

On crée la table groupes dans le tablespace users2

```
CREATE TABLE GROUPES (P_ID NUMBER NOT NULL, F_TYPE NUMBER NOT NULL, LIBELLE VARCHAR2(255 BYTE), CONSTRAINT PK_GROUPS PRIMARY KEY (P_ID), CONSTRAINT FK_GROUPS_1 FOREIGN KEY (F_TYPE) REFERENCES TYPES_GROUPS (P_ID) ) TABLESPACE USERS2 ;
```

3. d'utiliser les métavues USER_TABLESPACES et USER_TABLES pour constater les résultats de vos actions. Il est conseillé d'appliquer auparavant la commande desc pour extraire les colonnes intéressantes pour la question.

```
select tablespace_name from user_tablespaces
```

```
TABLESPACE_NAME
```

```
-----
```

```
SYSTEM
```

```
SYSAUX
```

```
UNDOTBS1
```

```
TEMP
```

```
USERS
```

```
USERS2
```

SIE : CORRIGE DU TD BASES DE DONNEES AVANCEES : TABLESPACES - CLUSTERS - INDEXS

```
select table_name,tablespace_name from user_tables;
```

TABLE_NAME	TABLESPACE_NAME
TYPES_GROUPES	USERS2
GROUPES	USERS2

4. De changer l'état du tablespace (OFFLINE et ONLINE) et constater le résultat en essayant des requêtes adéquates. Il faut avoir le privilège système ALTER TABLESPACE.

On se place dans le compte system, on donne des droits au compte bdda
grant alter tablespace to bdda;

On se remet dans le compte bdda

```
alter tablespace users2 offline;
```

Le select génère une erreur

```
select * from groupes;
```

Erreur SQL : ORA-01539: tablespace 'USERS2' is not online

```
alter tablespace users2 online;
```

Le select peut s'effectuer et ne génère plus d'erreur

```
select * from groupes;
```

aucune ligne sélectionnée

1.2 Documentation technique sur les tablespaces

- CREATE TABLESPACE nomTablespace DATAFILE 'patheFile.dbf' SIZE ???M EXTENT MANAGEMENT LOCAL AUTOALLOCATE;
- ALTER TABLESPACE nomTablespace OFFLINE ;
- ALTER TABLESPACE nomTablespace ONLINE ;
- ALTER TABLESPACE nomTablespace READ ONLY ;
- ALTER TABLESPACE nomTablespace READ WRITE;
- CREATE TABLE (???) TABLESPACE nomTablespace;

2. Exercice : Indexs

2.1 Questions

Dans la base de données AREL, on étudie les tables Rels_Relations et Acteurs

2.1.1 Indexs sur Acteurs

La table Acteurs a un champs f_statut qui est une clé étrangère vers la table Statuts. Il y à l'heure actuel, 2650 lignes dans la table Acteurs. Le contenu de la table Statut vaut :

P_ID	LIBELLE
3	administratif
4	etudiant
5	parent
6	professeur
7	scolarite
21	invite

- Pourquoi, si on crée un index sur f_statut, il faut créer un index bitmap.
Le nombre de valeurs différentes du champs f_statut est fixé et petit. Le taux de sélectivité de ce champ sera donc faible. Il faut donc privilégier un index bitmap.
- Créer cet index.
Rappel : La création d'un index de ce type n'est pas possible avec ORACLE XE (version bridée)
Create bitmap index indx_statut on Acteurs(f_statut);
- Créer un index adéquat sur (nom,prenom) en justifiant votre choix.
En revanche, le taux de sélectivité du couple (nom, prenom) dans la table Acteurs est très proche de 1. Il faut donc créer un index b-tree.
Create index indx_nom_prenom on Acteurs(nom,prenom);

2.1.2 Autres indexs

- Trouver d'autres indexs à créer en étudiant les tables qui gravitent autour de la table RELS.
- Créer les indexs adéquats.

Les bonnes questions à se poser pour créer les indexs ont été énoncées dans la question précédente.

2.2 Documentation technique sur les indexs

- CREATE INDEX nomSchema.nomIndex ON nomSchema.nomTable (liste des colonnes)
TABLESPACE nomTablespace ; // b-tree
- CREATE BITMAP INDEX nomSchema.nomIndex ON nomSchema.nomTable (liste des colonnes)
TABLESPACE nomTablespace ; //bitmap

3. Exercice : Cluster

3.1 Questions

3.1.1 RELS et les tables associées

La table Rels est reliée en (0,1) (1,1) avec chacune des tables suivantes :

- Documents
- Matieres
- Travaux
- Programmes
- Créer un cluster adéquat et écrire les requêtes LDD afin que le contenu de ces 5 tables soit stocké automatiquement par Oracle dans ce cluster.
- Constater le résultat de votre travail avec la métavue User_Tables.
- Constater qu'on ne peut rien faire sur les tables tant que le cluster n'a pas d'index sur ces colonnes.
- Créer l'index adéquat.

La question suivante est assez similaire donc vous pouvez vous référer à son corrigé.

3.1.2 Les groupes

Les tables Groupes et Groupes_parentes sont liées par une relation Maitre-Détails.

- Créer les tables Groupes et Groupes_parentes dans le tablespace Users2 sans les clustériser.

// On crée la table types_groupes

```
CREATE TABLE TYPES_GROUPES
(
  P_ID NUMBER NOT NULL,
  LIBELLE VARCHAR2(50 BYTE) NOT NULL,
  CONSTRAINT PK_TYPES_GROUPES PRIMARY KEY (P_ID)
) TABLESPACE users2;
```

// On crée la table groupes

```
CREATE TABLE GROUPES
(
  P_ID NUMBER NOT NULL,
  F_TYPE NUMBER NOT NULL,
  LIBELLE VARCHAR2(255 BYTE),
  CONSTRAINT PK_GROUPES PRIMARY KEY (P_ID),
  CONSTRAINT FK_GROUPES_1 FOREIGN KEY (F_TYPE)
  REFERENCES TYPES_GROUPES (P_ID)
) TABLESPACE users2;
```

// On crée la table groupes_parentes

```
CREATE TABLE GROUPES_PARENTES
(
  PF_PERE NUMBER NOT NULL,
```

SIE : CORRIGE DU TD BASES DE DONNEES AVANCEES : TABLESPACES - CLUSTERS - INDEXS

```
PF_FILS NUMBER NOT NULL,  
CONSTRAINT PK_GROUPES_PARENTES PRIMARY KEY (PF_PERE,  
PF_FILS),  
CONSTRAINT FK_GROUPES_PARENTES_2 FOREIGN KEY (PF_FILS)  
REFERENCES GROUPES (P_ID) ON DELETE CASCADE,  
CONSTRAINT FK_GROUPES_PARENTES_1 FOREIGN KEY (PF_PERE)  
REFERENCES GROUPES (P_ID) ON DELETE CASCADE  
) TABLESPACE users2;
```

- Remplir ces tables avec un jeu d'essai
- Ecrire les requêtes LDD pour créer un cluster adéquat pour clusteriser ces deux tables.

// On crée un cluster avec une colonne de jointure

```
create cluster clu_groupes(p_id_formelle number) tablespace users2;
```

// On crée la table groupes2 dans le cluster

```
CREATE TABLE GROUPES2  
( P_ID NUMBER NOT NULL,  
F_TYPE NUMBER NOT NULL,  
LIBELLE VARCHAR2(255 BYTE),  
CONSTRAINT PK_GROUPES2 PRIMARY KEY (P_ID),  
CONSTRAINT FK_GROUPES2_1 FOREIGN KEY (F_TYPE)  
REFERENCES TYPES_GROUPES (P_ID)  
) CLUSTER clu_groupes(p_id);
```

// On crée la table groupes_parentes2 dans le cluster

```
CREATE TABLE GROUPES_PARENTES2  
( PF_PERE NUMBER NOT NULL,  
PF_FILS NUMBER NOT NULL,  
CONSTRAINT PK_GROUPES_PARENTES2 PRIMARY KEY (PF_PERE,  
PF_FILS),  
CONSTRAINT FK_GROUPES_PARENTES2_2 FOREIGN KEY (PF_FILS)  
REFERENCES GROUPES (P_ID) ON DELETE CASCADE,  
CONSTRAINT FK_GROUPES_PARENTES2_1 FOREIGN KEY (PF_PERE)  
REFERENCES GROUPES2 (P_ID) ON DELETE CASCADE  
) CLUSTER clu_groupes(pf_pere);
```

// On regarde une partie du résultat de ces différentes créations de clusters et de tables avec la métavue USER_TABLES.

```
SELECT TABLE_NAME, TABLESPACE_NAME, CLUSTER_NAME FROM  
USER_TABLES;
```

TABLE_NAME	TABLESPACE_NAME	CLUSTER_NAME
PLAN_TABLE	USERS	
GROUPES2	USERS2	CLU_GROUPES
GROUPES_PARENTES2	USERS2	CLU_GROUPES

SIE : CORRIGE DU TD BASES DE DONNEES AVANCEES : TABLESPACES - CLUSTERS - INDEXS

```
TYPES_GROUPES      USERS2
GROUPES            USERS2
GROUPES_PARENTES  USERS2
```

- Ecrire les requêtes pour transférer les deux tables dans le cluster.

// Il faut d'abord créer l'index sur le cluster sinon aucune ligne ne peut être insérée dans les tables clusterisées

```
create index indx_clu_groupes on cluster clu_groupes;
```

// On recopie la table groupes dans la table groupes2

```
insert into groupes2 select * from groupes;
```

// On recopie la table groupes_parentes dans la table groupes_parentes2

```
insert into groupes_parentes2 select * from groupes_parentes;
```

// On détruit les tables groupes et groupes_parentes

```
DROP TABLE groupes;
```

```
DROP TABLE groupes_parentes;
```

// On renomme les tables groupes2 et groupes_parentes2

```
ALTER TABLE Groupes2 RENAME TO Groupes;
```

```
ALTER TABLE Groupes_parentes2 RENAME TO Groupes_parentes;
```

// On regarde une partie du résultat de ces différentes créations de clusters et de tables avec la métavue USER_TABLES.

```
SELECT TABLE_NAME, TABLESPACE_NAME, CLUSTER_NAME FROM
USER_TABLES;
```

TABLE_NAME	TABLESPACE_NAME	CLUSTER_NAME
PLAN_TABLE	USERS	
GROUPES	USERS2	CLU_GROUPES
GROUPES_PARENTES2	USERS2	CLU_GROUPES
TYPES_GROUPES	USERS2	

3.2 Documentation technique sur les clusters

- CREATE CLUSTER nomCluster(liste des colonnes formelles) TABLESPACE nomTablespace;
- CREATE TABLE nomTable(???) CLUSTER nomCluster(liste des colonnes effectives);
- CREATE INDEX nomIndex ON CLUSTER nomCluster;