USTHB
Université des Sciences et de la Technologie
Houari Boumediene

SQL

Rapport du TP n°3 Dictionnaire

Rougab Imene
M1 SII Groupe 01
Année scolaire : 2016/2017

Note : *A cause de quelques problèmes d'affichage sur sqlplus certains résultats ne sont pas bien affichés.*

1. On se connecte en tant que « system » :

a. Listons le catalogue « DICT »

Requête :

```
SELECT * FROM DICT ;
```

Résultat :

Cette requête donne un résultat de plus de 1000 lignes, voici donc une petite partie du résultat.

```
TABLE_NAME
------------------------------
COMMENTS
--------------------------------------------------------------------------------
USER_TABLESPACES
Description of accessible tablespaces

USER_UPDATABLE_COLUMNS
Description of updatable columns

USER_XML_TABLES
Description of the user's own XMLType tables


TABLE_NAME
------------------------------
COMMENTS
--------------------------------------------------------------------------------
V$DB_TRANSPORTABLE_PLATFORM
Synonym for V_$DB_TRANSPORTABLE_PLATFORM

V$FIXED_TABLE
Synonym for V_$FIXED_TABLE

V$TABLESPACE
Synonym for V_$TABLESPACE


TABLE_NAME
------------------------------
COMMENTS
--------------------------------------------------------------------------------
V$TRANSPORTABLE_PLATFORM
Synonym for V_$TRANSPORTABLE_PLATFORM
```

b. Nombre d'instances de « DICT » :

Requête :

```
SELECT COUNT(*) FROM DICT ;
```

Résultat :

Le dictionnaire contient 1821 lignes.

```
SQL> SELECT COUNT(*) FROM DICT;

  COUNT(*)
----------
      1821
```

c. STRUCTURE DE « DICT » :

Requête :

```
DESC DICT ;
```

Résultat :

Le dictionnaire est représenté sous la forme d'un tableau contenant les noms de toutes les structures de la base de données, ainsi que des commentaires sur ces dernières. On le voit bien dans sur le résultat obtenu, le tableau contient « TABLE_NAME » et « COMMENTS ».

```
SQL> DESC DICT;
 Nom                                       NULL ?   Type
 ----------------------------------------- -------- ----------------
 TABLE_NAME                                         VARCHAR2(30)
 COMMENTS                                           VARCHAR2(4000)
```

NOTE : On peut limiter l'affichage en listant par exemples des structures avec des noms précis, on va afficher par exemple les données du dictionnaire comportant le mot « TABLE » dans leurs noms :

Requête:

```
SELECT TABLE_NAME, COMMENTS
FROM DICT
WHERE TABLE_NAME LIKE '%TABLE%'
ORDER BY TABLE_NAME;
```

Résultat : Il existe 75 structures comportant le mot "TABLE" dans le nom.

(On affiche une partie)

```
TABLE_NAME                     COMMENTS
------------------------------ --------------------------------------------------
USER_TABLESPACES               Description of accessible tablespaces
USER_UPDATABLE_COLUMNS         Description of updatable columns
USER_XML_TABLES                Description of the user's own XMLType tables
V$DB_TRANSPORTABLE_PLATFORM    Synonym for V_$DB_TRANSPORTABLE_PLATFORM
V$FIXED_TABLE                  Synonym for V_$FIXED_TABLE
V$TABLESPACE                   Synonym for V_$TABLESPACE
V$TRANSPORTABLE_PLATFORM       Synonym for V_$TRANSPORTABLE_PLATFORM

75 ligne(s) sÚlectionnÚe(s).
```

2. Rôles et structures des tables et vues :

a. ALL_TAB_COLUMNS : Cette vue rassemble les colonnes des tables de la base de données.

```
SQL> DESC ALL_TAB_COLUMNS;
 Nom                                       NULL ?   Type
 ----------------------------------------- -------- ------------
 OWNER                                     NOT NULL VARCHAR2(30)
 TABLE_NAME                                NOT NULL VARCHAR2(30)
 COLUMN_NAME                               NOT NULL VARCHAR2(30)
 DATA_TYPE                                          VARCHAR2(106)
 DATA_TYPE_MOD                                      VARCHAR2(3)
 DATA_TYPE_OWNER                                    VARCHAR2(30)
 DATA_LENGTH                               NOT NULL NUMBER
 DATA_PRECISION                                     NUMBER
 DATA_SCALE                                         NUMBER
 NULLABLE                                           VARCHAR2(1)
 COLUMN_ID                                          NUMBER
 DEFAULT_LENGTH                                     NUMBER
 DATA_DEFAULT                                       LONG
 NUM_DISTINCT                                       NUMBER
 LOW_VALUE                                          RAW(32)
 HIGH_VALUE                                         RAW(32)
 DENSITY                                            NUMBER
 NUM_NULLS                                          NUMBER
 NUM_BUCKETS                                        NUMBER
 LAST_ANALYZED                                      DATE
 SAMPLE_SIZE                                        NUMBER
 CHARACTER_SET_NAME                                 VARCHAR2(44)
 CHAR_COL_DECL_LENGTH                               NUMBER
 GLOBAL_STATS                                       VARCHAR2(3)
 USER_STATS                                         VARCHAR2(3)
 AVG_COL_LEN                                        NUMBER
 CHAR_LENGTH                                        NUMBER
 CHAR_USED                                          VARCHAR2(1)
 V80_FMT_IMAGE                                      VARCHAR2(3)
 DATA_UPGRADED                                      VARCHAR2(3)
 HISTOGRAM                                          VARCHAR2(15)
```

b. USER_USERS : Contient les informations sur tous les utilisateurs.

```
SQL> DESC USER_USERS;
 Nom                                       NULL ?   Type
 ----------------------------------------- -------- ------------
 USERNAME                                  NOT NULL VARCHAR2(30)
 USER_ID                                   NOT NULL NUMBER
 ACCOUNT_STATUS                            NOT NULL VARCHAR2(32)
 LOCK_DATE                                          DATE
 EXPIRY_DATE                                        DATE
 DEFAULT_TABLESPACE                        NOT NULL VARCHAR2(30)
 TEMPORARY_TABLESPACE                      NOT NULL VARCHAR2(30)
 CREATED                                   NOT NULL DATE
 INITIAL_RSRC_CONSUMER_GROUP                        VARCHAR2(30)
 EXTERNAL_NAME                                      VARCHAR2(4000)
```

c. **ALL_CONSTRAINTS :** Contient les informations sur toutes les contraintes définies dans la base de données.

```
SQL> DESC ALL_CONSTRAINTS;
 Nom                                       NULL ?   Type
 ----------------------------------------- -------- ------------
 OWNER                                     NOT NULL VARCHAR2(30)
 CONSTRAINT_NAME                           NOT NULL VARCHAR2(30)
 CONSTRAINT_TYPE                                    VARCHAR2(1)
 TABLE_NAME                                NOT NULL VARCHAR2(30)
 SEARCH_CONDITION                                   LONG
 R_OWNER                                            VARCHAR2(30)
 R_CONSTRAINT_NAME                                  VARCHAR2(30)
 DELETE_RULE                                        VARCHAR2(9)
 STATUS                                             VARCHAR2(8)
 DEFERRABLE                                         VARCHAR2(14)
 DEFERRED                                           VARCHAR2(9)
 VALIDATED                                          VARCHAR2(13)
 GENERATED                                          VARCHAR2(14)
 BAD                                                VARCHAR2(3)
 RELY                                               VARCHAR2(4)
 LAST_CHANGE                                        DATE
 INDEX_OWNER                                        VARCHAR2(30)
 INDEX_NAME                                         VARCHAR2(30)
 INVALID                                            VARCHAR2(7)
 VIEW_RELATED                                       VARCHAR2(14)
```

d. **USER_TAB_PRIVS :** rassemble les informations concernant les privilèges des utilisateurs sur les tables de la base de données.

```
SQL> DESC USER_TAB_PRIVS;
 Nom                                       NULL ?   Type
 ----------------------------------------- -------- ------------
 GRANTEE                                   NOT NULL VARCHAR2(30)
 OWNER                                     NOT NULL VARCHAR2(30)
 TABLE_NAME                                NOT NULL VARCHAR2(30)
 GRANTOR                                   NOT NULL VARCHAR2(30)
 PRIVILEGE                                 NOT NULL VARCHAR2(40)
 GRANTABLE                                          VARCHAR2(3)
 HIERARCHY                                          VARCHAR2(3)
```

3. Nom d'utilisateur avec lequel on est connecté :

Requête :

```
SELECT USER FROM USER_USERS;
```

Résultat :

```
SQL> SELECT USERNAME FROM USER_USERS;

USERNAME
------------------------------
SYSTEM
```

4. Comparaison de « ALL_TAB_COLUMNS » ET « USER_TAB_COLUMNS » :

Comme on le voit dans les résultats, la structure des deux vues est la même. La différence réside dans le contenu. « ALL_TAB_COLUMNS » contient des informations sur toutes les tables de la base de données, tandis que « USER_TAB_COLUMNS » contient des informations sur les colonnes des tables d'un utilisateur précis.

```
SQL> DESC USER_TAB_COLUMNS;
 Nom                                       NULL ?   Type
 ----------------------------------------- -------- ----------------
 TABLE_NAME                                NOT NULL VARCHAR2(30)
 COLUMN_NAME                               NOT NULL VARCHAR2(30)
 DATA_TYPE                                          VARCHAR2(106)
 DATA_TYPE_MOD                                      VARCHAR2(3)
 DATA_TYPE_OWNER                                    VARCHAR2(30)
 DATA_LENGTH                               NOT NULL NUMBER
 DATA_PRECISION                                     NUMBER
 DATA_SCALE                                         NUMBER
 NULLABLE                                           VARCHAR2(1)
 COLUMN_ID                                          NUMBER
 DEFAULT_LENGTH                                     NUMBER
 DATA_DEFAULT                                       LONG
 NUM_DISTINCT                                       NUMBER
 LOW_VALUE                                          RAW(32)
 HIGH_VALUE                                         RAW(32)
 DENSITY                                            NUMBER
 NUM_NULLS                                          NUMBER
 NUM_BUCKETS                                        NUMBER
 LAST_ANALYZED                                      DATE
 SAMPLE_SIZE                                        NUMBER
 CHARACTER_SET_NAME                                 VARCHAR2(44)
 CHAR_COL_DECL_LENGTH                               NUMBER
 GLOBAL_STATS                                       VARCHAR2(3)
 USER_STATS                                         VARCHAR2(3)
 AVG_COL_LEN                                        NUMBER
 CHAR_LENGTH                                        NUMBER
 CHAR_USED                                          VARCHAR2(1)
 V80_FMT_IMAGE                                      VARCHAR2(3)
 DATA_UPGRADED                                      VARCHAR2(3)
 HISTOGRAM                                          VARCHAR2(15)
```

```
SQL> DESC ALL_TAB_COLUMNS;
 Nom                                       NULL ?   Type
 ----------------------------------------- -------- ------------
 OWNER                                     NOT NULL VARCHAR2(30)
 TABLE_NAME                                NOT NULL VARCHAR2(30)
 COLUMN_NAME                               NOT NULL VARCHAR2(30)
 DATA_TYPE                                          VARCHAR2(106)
 DATA_TYPE_MOD                                      VARCHAR2(3)
 DATA_TYPE_OWNER                                    VARCHAR2(30)
 DATA_LENGTH                               NOT NULL NUMBER
 DATA_PRECISION                                     NUMBER
 DATA_SCALE                                         NUMBER
 NULLABLE                                           VARCHAR2(1)
 COLUMN_ID                                          NUMBER
 DEFAULT_LENGTH                                     NUMBER
 DATA_DEFAULT                                       LONG
 NUM_DISTINCT                                       NUMBER
 LOW_VALUE                                          RAW(32)
 HIGH_VALUE                                         RAW(32)
 DENSITY                                            NUMBER
 NUM_NULLS                                          NUMBER
 NUM_BUCKETS                                        NUMBER
 LAST_ANALYZED                                      DATE
 SAMPLE_SIZE                                        NUMBER
 CHARACTER_SET_NAME                                 VARCHAR2(44)
 CHAR_COL_DECL_LENGTH                               NUMBER
 GLOBAL_STATS                                       VARCHAR2(3)
 USER_STATS                                         VARCHAR2(3)
 AVG_COL_LEN                                        NUMBER
 CHAR_LENGTH                                        NUMBER
 CHAR_USED                                          VARCHAR2(1)
 V80_FMT_IMAGE                                      VARCHAR2(3)
 DATA_UPGRADED                                      VARCHAR2(3)
 HISTOGRAM                                          VARCHAR2(15)
```

5. Vérifions que les tables du TP1 ont bien été créées :

Requête :

```
SELECT DISTINCT TABLE_NAME FROM ALL_TABLES;
```

Résultat :

```
SQL> SELECT DISTINCT TABLE_NAME FROM ALL_TABles WHERE OWNER='ROUGAB';

TABLE_NAME
------------------------------
HOTEL
CHAMBRE
RESERVATION
CLIENT
```

Affichons maintenant les informations sur ces tables :

Requête : SELECT * FROM USER_TABLES ;

On ne va pas afficher le résultat de cette requête, vu la taille de ce dernier.

On va préciser des informations à afficher, par exemple les noms des tables et le nom de la TablesSpaces à laquelle elles appartiennent.

SELECT TABLE_NAME, TABLESPACE_NAME CLUSTER_NAME FROM USER_TABLES ;

Résultat :

```
SQL> SELECT TABLE_NAME, TABLESPACE_NAME FROM USER_TABLES ;

TABLE_NAME                     TABLESPACE_NAME
------------------------------ ------------------------------
HOTEL                          TPS
CHAMBRE                        TPS
CLIENT                         TPS
RESERVATION                    TPS
```

6. Listons les tables des utilisateurs « SYSTEM » et « ADMINHOTEL »

Requêtes:

SELECT DISTINCT TABLE_NAME FROM ALL_TABLES WHERE OWNER='SYSTEM';

SELECT DISTINCT TABLE_NAME FROM ALL_TABLES WHERE OWNER='ADMINHOTEL';

Résultats: (en ce qui concerne les tables de « system » on a pris juste une partie du résultat)

```
SQL> SELECT DISTINCT TABLE_NAME FROM ALL_TABLES WHERE OWNER='ADMINHOTEL';

TABLE_NAME
------------------------------
HOTEL
ETUDIANT
```

```
SQL> SELECT DISTINCT TABLE_NAME FROM ALL_TABLES WHERE OWNER='SYSTEM';

TABLE_NAME
------------------------------
DEF$_LOB
LOGMNRC_DBNAME_UID_MAP
LOGMNR_AGE_SPILL$
LOGSTDBY$PARAMETERS
LOGSTDBY$APPLY_MILESTONE
LOGSTDBY$SCN
REPCAT$_FLAVORS
REPCAT$_SNAPGROUP
REPCAT$_CONFLICT
REPCAT$_OBJECT_TYPES
REPCAT$_USER_PARM_VALUES

TABLE_NAME
```

7. Description des attributs des tables « CLIENT » et « RESERVATION »

On va afficher les attributs de chaque table, leurs types ainsi que leurs tailles

Requêtes :

SELECT COLUMN_NAME, DATA_TYPE, DATA_LENGTH
FROM USER_TAB_COLUMNS
WHERE TABLE_NAME='RESERVATION';

SELECT COLUMN_NAME, DATA_TYPE, DATA_LENGTH
FROM USER_TAB_COLUMNS
WHERE TABLE_NAME='CLIENT';

Résultats:

```
SQL> SELECT COLUMN_NAME, DATA_TYPE, DATA_LENGTH FROM USER_TAB_COLUMNS WHERE TABLE_NAME='RESERVATION';

COLUMN_NAME                    DATA_TYPE                                                    DATA_LENGTH
------------------------------ ------------------------------------------------------------ -----------
DATEARRIVEE                    DATE                                                                   7
DATEDEPART                     DATE                                                                   7
NUMCHAMBRE                     NUMBER                                                                22
NUMCLIENT                      NUMBER                                                                22
NUMHOTEL                       NUMBER                                                                22
```

```
SQL> SELECT COLUMN_NAME, DATA_TYPE, DATA_LENGTH FROM USER_TAB_COLUMNS WHERE TABLE_NAME='CLIENT';

COLUMN_NAME                    DATA_TYPE                                                    DATA_LENGTH
------------------------------ ------------------------------------------------------------ -----------
ADRESSECLIENT                  VARCHAR2                                                              50
NUMCLIENT                      NUMBER                                                                22
NOM                            VARCHAR2                                                              20
PRENOM                         VARCHAR2                                                              20
```

8. Vérifions qu'il y a une référence de clé étrangère entre la table « CLIENT » et « RESERVATION »

Il suffit d'afficher les contraintes de la table « RESERVATION »

Requête :

SELECT CONSTRAINT_NAME, CONSTRAINT_TYPE FROM USER_CONSTRAINTS WHERE TABLE_NAME='RESERVATION';

Résultat:

```
SQL> SELECT CONSTRAINT_NAME, constraint_type FROM USER_CONSTRAINTS WHERE TABLE_NAME='RESERVATION';

CONSTRAINT_NAME                C
------------------------------ -
CHECK_DATE                     C
PK_RESERVATION                 P
FK_CHAMBRE                     R
FK_CLIENT                      R
```

9. Affichages des contraintes du TP1 et de leurs caractéristiques :

On ne va afficher que le nom de la contrainte, le nom de la table dans laquelle elle est définit et son type (pour des raisons d'affichage).

Requête :

```
//Pour afficher toutes les caractéristiques
SELECT * FROM USER_CONSTRAINTS ;
//on ne va afficher que table_name, contraint_name et constraint_type
SELECT TABLE_NAME, CONSTRAINT_NAME, CONSTRAINT_TYPE FROM
USER_CONSTRAINTS;
```

Résultat:

```
SQL> SELECT TABLE_NAME, CONSTRAINT_NAME, CONSTRAINT_TYPE FROM USER_CONSTRAINTS;

TABLE_NAME                     CONSTRAINT_NAME                C
------------------------------ ------------------------------ -
RESERVATION                    FK_CLIENT                      R
RESERVATION                    FK_CHAMBRE                     R
RESERVATION                    CHECK_DATE                     C
CHAMBRE                        FK_HOTEL                       R
CHAMBRE                        CHECK_TYPECHAMBRE              C
BIN$7b/WaOfLSeSiinSFuZWO4g==$0 BIN$i+CuJR0xQs2ict568Wsl3Q==$0 C
BIN$7b/WaOfLSeSiinSFuZWO4g==$0 BIN$elScYC3TS5+p/oF6MWfWKg==$0 C

TABLE_NAME                     CONSTRAINT_NAME                C
------------------------------ ------------------------------ -
BIN$SyiWAXRgTNC1BM2yehf0wQ==$0 BIN$+wrYxcJPQ92KTBXvLughjg==$0 C
HOTEL                          CHECK_ETOILES                  C
BIN$0KNsJe70QQCgUD8mjM77AQ==$0 BIN$OkDnnXq/RDStCKVgdmmEhg==$0 C
BIN$qLqQ/mAER62MwLsft1pLsQ==$0 BIN$Ty4Sb0L7QXOB30eT89/R+A==$0 C
BIN$7b/WaOfLSeSiinSFuZWO4g==$0 BIN$GDIKdEF4S6+r3uw81M/P5Q==$0 C
BIN$SyiWAXRgTNC1BM2yehf0wQ==$0 BIN$g2oBiQNQTguceuk2p5GnVQ==$0 C
BIN$7WKu3SjMQIeaWqiHRpdg1g==$0 BIN$pkvAama6S/migYeeY/162g==$0 C

TABLE_NAME                     CONSTRAINT_NAME                C
------------------------------ ------------------------------ -
BIN$XH8XxZiHSRyiuR5e7CUNwg==$0 BIN$YEgitPiRRO6e6w0UoLRriA==$0 C
BIN$OAcEi1x5QfqfwnNipq/EuA==$0 BIN$nIwb2OOJSSCGM391dlHLNQ==$0 C
BIN$Q3ckJH97S6m7r9tMq0jJZA==$0 BIN$dOJPHpWjR6KEPsVWSeZeMg==$0 C
BIN$OrTRJnrpRnmPkLAcWQOB9Q==$0 BIN$0NF+RvWOTAevVs0eASYQIg==$0 P
BIN$7b/WaOfLSeSiinSFuZWO4g==$0 BIN$T18QQ7AGS3KsGMkZHijFBA==$0 P
BIN$SyiWAXRgTNC1BM2yehf0wQ==$0 BIN$ZcZstE0nQrmHn45SaObWgQ==$0 P
HOTEL                          PK_HOTEL                       P

TABLE_NAME                     CONSTRAINT_NAME                C
------------------------------ ------------------------------ -
CHAMBRE                        PK_CHAMBRE                     P
CLIENT                         PRK_CLIENT                     P
RESERVATION                    PK_RESERVATION                 P
BIN$Q3ckJH97S6m7r9tMq0jJZA==$0 BIN$h+Gtjv54TcOsWR1vXCqeWw==$0 P
BIN$OAcEi1x5QfqfwnNipq/EuA==$0 BIN$xb3k2601REm/yAJSfWVChQ==$0 P
BIN$XH8XxZiHSRyiuR5e7CUNwg==$0 BIN$Jr583FuXT9+O81CKXaLmDg==$0 P
BIN$7WKu3SjMQIeaWqiHRpdg1g==$0 BIN$WWVGAWWfRxCW46DhvPIpOQ==$0 P

TABLE_NAME                     CONSTRAINT_NAME                C
------------------------------ ------------------------------ -
BIN$Dn5pkewiRPCMp9DL47yQxA==$0 BIN$qXk1gOD1QEWrT+zcIVo28A==$0 P
BIN$qLqQ/mAER62MwLsft1pLsQ==$0 BIN$tAPDw2S2Tp+L0+6aK8DDrA==$0 P
BIN$0KNsJe70QQCgUD8mjM77AQ==$0 BIN$yt61V3T1S0qL6kaM3a6oEA==$0 P
```

10. Les informations permettant de créer la table « RESERVATION »

On a besoin de connaitre les attributs (Noms, types, tailles) et els contraintes définies sur la table (Noms, types, et les colonnes sur lesquelles elles sont définies)

Requête :

```
// Les attributs
SELECT COLUMN_NAME, DATA_TYPE, DATA_LENGTH
FROM USER_TAB_COLUMNS
WHERE TABLE_NAME='RESERVATION';
// Les contraintes
SELECT U.CONSTRAINT_NAME, U.CONSTRAINT_TYPE, A.COLUMN_NAME
FROM USER_CONSTRAINTS U, ALL_CONS_COLUMNS A
WHERE A.TABLE_NAME='RESERVATION'
AND U.CONSTRAINT_NAME=A.CONSTRAINT_NAME;
```

Résultats:

```
SQL> SELECT COLUMN_NAME, DATA_TYPE, DATA_LENGTH
  2  FROM USER_TAB_COLUMNS
  3  WHERE TABLE_NAME='RESERVATION';

COLUMN_NAME                    DATA_TYPE                                    DATA_LENGTH
------------------------------ -------------------------------------------- -----------
DATEARRIVEE                    DATE                                                   7
DATEDEPART                     DATE                                                   7
NUMCHAMBRE                     NUMBER                                                22
NUMCLIENT                      NUMBER                                                22
NUMHOTEL                       NUMBER                                                22
```

```
SQL> SELECT U.CONSTRAINT_NAME, U.CONSTRAINT_TYPE, A.COLUMN_NAME
  2  FROM USER_CONSTRAINTS U, ALL_CONS_COLUMNS A
  3  WHERE A.TABLE_NAME='RESERVATION'
  4  AND U.CONSTRAINT_NAME=A.CONSTRAINT_NAME;

CONSTRAINT_NAME                C COLUMN_NAME
------------------------------ - -----------------------------------
FK_CLIENT                      R NUMCLIENT
FK_CLIENT                      R NUMCLIENT
FK_CHAMBRE                     R NUMHOTEL
FK_CHAMBRE                     R NUMCHAMBRE
FK_CHAMBRE                     R NUMHOTEL
FK_CHAMBRE                     R NUMCHAMBRE
PK_RESERVATION                 P NUMHOTEL

CONSTRAINT_NAME                C COLUMN_NAME
------------------------------ - -----------------------------------
PK_RESERVATION                 P NUMCLIENT
PK_RESERVATION                 P DATEARRIVEE
PK_RESERVATION                 P NUMHOTEL
PK_RESERVATION                 P NUMCLIENT
PK_RESERVATION                 P DATEARRIVEE
CHECK_DATE                     C DATEARRIVEE
CHECK_DATE                     C DATEDEPART

CONSTRAINT_NAME                C COLUMN_NAME
------------------------------ - -----------------------------------
CHECK_DATE                     C DATEARRIVEE
CHECK_DATE                     C DATEDEPART

16 ligne(s) sÚlectionnÚe(s).
```

11. Les privilèges accordés à l'utilisateur « ADMINHOTEL »

Requête :

```
//si on n'est pas connecté avec « ADMINHOTEL »
SELECT PRIVILEGE, TABLE_NAME FROM USER_TAB_PRIVS WHERE
GRANTEE='ADMINHOTEL';
// si on est connecté en tant que « ADMINHOTEL »
SELECT PRIVILEGE, TABLE_NAME FROM USER_TAB_PRIVS;
```

Résultat :

```
SQL> select PRIVILEGE,TABLE_NAME from USER_TAB_PRIVS where GRANTEE='ADMINHOTEL';

PRIVILEGE                                TABLE_NAME
---------------------------------------- ------------------------------
SELECT                                   HOTEL
UPDATE                                   CHAMBRE
SELECT                                   CHAMBRE
```

12. Les rôles donnés à « ADMINHOTEL »

Requête :

```
//On est connecté en tant que « ADMINHOTEL »
SELECT GRANTED_ROLE FROM USER_ROLE_PRIVS;//Afficher uniquement le rôle
SELECT * FROM USER_ROLE_PRIVS; //Afficher tout le détails concernant
le rôle
```

Résultat:

```
SQL> SELECT GRANTED_ROLE FROM USER_ROLE_PRIVS;

GRANTED_ROLE
------------------------------
GESTIONNAIRE_DE_RESERVATIONS
```

```
SQL> select * from USER_ROLE_PRIVS;

USERNAME                       GRANTED_ROLE                   ADM DEF OS_
------------------------------ ------------------------------ --- --- ---
ADMINHOTEL                     GESTIONNAIRE_DE_RESERVATIONS   NO  YES NO
```

13. Les objets de « ADMINHOTEL »

Requête :

```
SELECT * FROM USER_OBJECTS; //Afficher toutes les informations sur les objets
SELECT OBJECT_NAME FROM USER_OBJECTS; //Afficher les noms des objets
SELECT OBJECT_NAME FROM ALL_OBJECTS WHERE OWNER='ADMINHOTEL'; //Si on n'est pas connecté avec "AdminHotel"
```

Résultat:

```
SQL> SELECT OBJECT_NAME FROM ALL_OBJECTS WHERE OWNER='ADMINHOTEL';

OBJECT_NAME
------------------------------
HOTEL
PK_HOTEL
CHAMBRETYPE_IX
ETUDIANT
```

```
SQL> select * from USER_OBJECTS;

OBJECT_NAME
--------------------------------------------------------------------------------
SUBOBJECT_NAME                 OBJECT_ID DATA_OBJECT_ID OBJECT_TYPE
------------------------------ ---------- -------------- -------------------
CREATED  LAST_DDL TIMESTAMP           STATUS  T G S
-------- -------- ------------------- ------- - - -
HOTEL
                                   14157          14157 TABLE
16/10/16 16/10/16 2016-10-16:10:52:12 VALID   N N N

PK_HOTEL
                                   14158          14158 INDEX
16/10/16 16/10/16 2016-10-16:10:52:13 VALID   N N N

OBJECT_NAME
--------------------------------------------------------------------------------
SUBOBJECT_NAME                 OBJECT_ID DATA_OBJECT_ID OBJECT_TYPE
------------------------------ ---------- -------------- -------------------
CREATED  LAST_DDL TIMESTAMP           STATUS  T G S
-------- -------- ------------------- ------- - - -

CHAMBRETYPE_IX
                                   14166          14166 INDEX
16/10/16 16/10/16 2016-10-16:19:06:31 VALID   N N N

ETUDIANT
                                   14167          14167 TABLE
```

14. Pour que l'administrateur puisse retrouver le propriétaire de la table « RESERVATION » :

Il doit executer la requête suivante :

Requête :

```
SELECT OWNER FROM ALL_TABLES WHERE TABLE_NAME='RESERVATION';
```

Résultat:

```
SQL> SELECT OWNER FROM ALL_TABLES WHERE TABLE_NAME='RESERVATION';

OWNER
----------------------------
ROUGAB
```

15. Taille en KO de la table « RESERVATION »

Requête :

```
SELECT BYTES/1024 AS SIZE_TABLE_KO
FROM USER_SEGMENTS
WHERE SEGMENT_NAME='RESERVATION';
```

Résultat:

```
SQL> SELECT BYTES/1024 AS SIZE_TABLE_KO FROM USER_SEGMENTS WHERE SEGMENT_NAME='RESERVATION';

SIZE_TABLE_KO
-------------
           64
```

16. Vérifions l'effet produit par les commandes du TP1 sur le dictionnaire

1- Créons un nouvel utilisateur : on crée des TablesSpaces puis un utilisateur et on lui attribue tous les privilèges

```
SQL> create tablespace tp3_tbs datafile 'C:\Users\im\Desktop\M1 SII\ASGBD\TP2\tp3_tbs.dat' size 100M autoextend on online;

Tablespace crÚÚ.

SQL> create temporary tablespace tp3_temp tempfile 'C:\Users\im\Desktop\M1 SII\ASGBD\TP2\tp3_temp.dat' size 100M autoextend on;

Tablespace crÚÚ.

SQL> create user user_tp3 identified by test default tablespace tp3_tbs temporary tablespace tp3_temp;

Utilisateur crÚÚ.

SQL> GRANT ALL PRIVILEGES TO user_tp3;

Autorisation de privilÞges (GRANT) acceptÚe.

SQL> disconnect
DÚconnectÚ de Oracle Database 10g Express Edition Release 10.2.0.1.0 - Production
SQL> connect
Entrez le nom utilisateur : user_tp3
Entrez le mot de passe :
ConnectÚ.
```

2- Vérifions ce qu'il y a dans le dictionnaire avant de créer les tables :

Requête :

```
SELECT TABLE_NAME  FROM USER_TABLES;

SELECT COLUMN_NAME FROM USER_TAB_COLUMNS;

SELECT CONSTRAINT_NAME  FROM USER_CONSTRAINTS;
```

Résultat :

```
SQL> select TABLE_NAME  from USER_TABLES;

aucune ligne sÚlectionnÚe

SQL> select COLUMN_NAME from USER_TAB_COLUMNS;

aucune ligne sÚlectionnÚe

SQL>
SQL> select CONSTRAINT_NAME  from USER_CONSTRAINTS;

aucune ligne sÚlectionnÚe
```

3- Créons désormais les tables du TP1 et vérifions encore le contenu du dictionnaire en utilisant les requêtes précédentes :

Création des tables :

```
SQL> create table HOTEL (
  2  NumHotel Number(3),
  3  Nom varchar(50),
  4  Ville varchar(20),
  5  Etoiles number(1),
  6  constraint pk_hotel primary key (NumHotel),
  7  constraint check_etoiles check (Etoiles>0 AND Etoiles<6)
  8  );

Table crÚÚe.

SQL>
SQL> create table CHAMBRE (
  2  NumChambre Number(3),
  3  NumHotel number(3),
  4  Etage number(3),
  5  TypeChambre varchar(10),
  6  PrixNuit number(5),
  7  constraint pk_chambre primary key (NumHotel,NumChambre),
  8  constraint fk_hotel foreign key (NumHotel) references HOTEL(NumHotel) on delete cascade,
  9  constraint check_TypeChambre check (TypeChambre IN ('simple','double','triple','suite','autre'))
 10  );

Table crÚÚe.

SQL>
SQL> create table CLIENT(
  2  NumClient number(5),
  3  Nom varchar(20),
  4  Prenom varchar(20),
  5  constraint prk_client primary key (NumClient)
  6  );

Table crÚÚe.

SQL>
SQL> create table RESERVATION (
  2  NumClient number(5),
  3  NumHotel number(3),
  4  DateArrivee date,
  5  DateDepart date,
  6  NumChambre number(3),
  7  constraint pk_reservation primary key(NumHotel, NumClient, DateArrivee),
  8  constraint fk_chambre foreign key (NumHotel,NumChambre) references CHAMBRE(NumHotel,NumChambre) on delete cascade,
  9  constraint fk_client foreign key (NumClient) references CLIENT(NumClient) on delete cascade,
 10  constraint check_date check (DateArrivee<DateDepart)
 11  );

Table crÚÚe.
```

Vérification du contenu du dictionnaire après création des tables :

Le dictionnaire contient désormais les informations sur les tables créées, les colonnes, les contraintes…

```
SQL> select TABLE_NAME  from USER_TABLES;

TABLE_NAME
------------------------------
HOTEL
CHAMBRE
CLIENT
RESERVATION

SQL>
SQL> select COLUMN_NAME from USER_TAB_COLUMNS;

COLUMN_NAME
------------------------------
PRENOM
NUMCHAMBRE
ETOILES
ETAGE
DATEDEPART
NUMHOTEL
NUMHOTEL
NUMCHAMBRE
NOM
TYPECHAMBRE
VILLE

COLUMN_NAME
------------------------------
NOM
NUMCLIENT
NUMHOTEL
NUMCLIENT
PRIXNUIT
DATEARRIVEE

17 ligne(s) sÚlectionnÚe(s).
```

4- Vérifions l'effet de la requête « Alter » sur le dictionnaire :

Exécutons ces 2 requêtes :

```
Alter table CLIENT ADD AdresseClient varchar(50);
Alter table CHAMBRE RENAME COLUMN ETAGE TO ETAGECHAMBRE;
```

```
SQL> Alter table CLIENT ADD AdresseClient varchar(50);

Table modifiÚe.

SQL> Alter table CHAMBRE rename column ETAGE to ETAGECHAMBRE;

Table modifiÚe.
```

Vérifions maintenant le contenu du dictionnaire : on remarque dans le résultat que la colonne « AdresseClient » a bien été ajoutée, et la colonne « Etage » a été renommée en « EtageChambre »

```
SQL> SELECT COLUMN_NAME FROM USER_TAB_COLUMNS;

COLUMN_NAME
------------------------------
PRENOM
NUMCHAMBRE
ETOILES
ETAGECHAMBRE
ADRESSECLIENT
DATEDEPART
NUMHOTEL
NUMHOTEL
NUMCHAMBRE
NOM
TYPECHAMBRE

COLUMN_NAME
------------------------------
VILLE
NOM
NUMCLIENT
NUMHOTEL
NUMCLIENT
PRIXNUIT
DATEARRIVEE

18 ligne(s) sÚlectionnÚe(s).
```

4- Enfin, vérifions l'effet de l'ajout d'une contrainte sur le dictionnaire :

On va prendre comme exemple la contrainte « Check» sur l'attribut « PrixNuit »

Avant l'ajout de la contrainte :

```
SQL> select CONSTRAINT_NAME  from USER_CONSTRAINTS;

CONSTRAINT_NAME
------------------------------
FK_CLIENT
FK_CHAMBRE
CHECK_DATE
FK_HOTEL
CHECK_TYPECHAMBRE
CHECK_ETOILES
PK_HOTEL
PK_CHAMBRE
PRK_CLIENT
PK_RESERVATION

10 ligne(s) sÚlectionnÚe(s).
```

Ajout de la contrainte et vérifications du contenu du dictionnaire :

On remarque après exécution que la contrainte « Check_Prix » a été ajouté à la liste des contraintes contenues dans le dictionnaire de données.

```
SQL> Alter table CHAMBRE add constraint check_prix check( PrixNuit>2000 and PrixNuit<20000);

Table modifiÚe.

SQL>
SQL> select CONSTRAINT_NAME  from USER_CONSTRAINTS;

CONSTRAINT_NAME
------------------------------
CHECK_PRIX
FK_CLIENT
FK_CHAMBRE
CHECK_DATE
FK_HOTEL
CHECK_TYPECHAMBRE
CHECK_ETOILES
PK_HOTEL
PK_CHAMBRE
PRK_CLIENT
PK_RESERVATION

11 ligne(s) sÚlectionnÚe(s).
```