# ASGBD

## Rapport du TP N°3: « Dictionnaire Oracle »

**CHIKH Khadidja**

Master 1 SII
Groupe 4

19/10/2018

**1) La structure du catalogue DICT:**

```
SQL> desc dict
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 TABLE_NAME                                         VARCHAR2(30)
 COMMENTS                                           VARCHAR2(4000)
```

**Le nombre de ses instances:**

```
SQL> select count(*) from dict;

  COUNT(*)
----------
      2551
```

**Exemple d'instance:**

```
SQL> select * from dict;

TABLE_NAME
------------------------------
COMMENTS
--------------------------------------------------------------------------------
USER_CONS_COLUMNS
Information about accessible columns in constraint definitions

ALL_CONS_COLUMNS
Information about accessible columns in constraint definitions

DBA_CONS_COLUMNS
Information about accessible columns in constraint definitions
```

**2) Les rôles et les structures des tables ALL_TAB_COLUMNS, USER_USERS, ALL_CONSTRAINTS et USER_TAB_PRIVS:**

ALL_TAB_COLUMNS:

```
SQL> select * from dict where table_name='ALL_TAB_COLUMNS';

TABLE_NAME
------------------------------
COMMENTS
--------------------------------------------------------------------------------
ALL_TAB_COLUMNS
Columns of user's tables, views and clusters
```

Elle contient les informations concernant toutes les colonnes de toutes les tables,vues ou clusters auxquels l'utilisateur peut acceder.

Sa structure :

```
SQL> desc all_tab_columns;
 Name                                      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(120)
 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)
```

USER_USERS :

```
SQL> select * from dict where table_name='USER_USERS';

TABLE_NAME
------------------------------
COMMENTS
--------------------------------------------------------------------------------
USER_USERS
Information about the current user
```

Liste les informations concernant l'utilisateur connecté.

Sa structure :

```
SQL> desc user_users;
 Name                                      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)
```

ALL_CONSTRAINTS :

```
SQL> select * from dict where table_name='ALL_CONSTRAINTS';

TABLE_NAME
------------------------------
COMMENTS
--------------------------------------------------------------------------------
ALL_CONSTRAINTS
Constraint definitions on accessible tables
```

Elle liste les informations concernant les contraintes definies sur les tables accessibles par l'utilisateur.

Sa structure :

```
SQL> desc all_constraints
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 OWNER                                              VARCHAR2(120)
 CONSTRAINT_NAME                           NOT NULL VARCHAR2(30)
 CONSTRAINT_TYPE                                    VARCHAR2(1)
 TABLE_NAME                                NOT NULL VARCHAR2(30)
 SEARCH_CONDITION                                   LONG
 R_OWNER                                            VARCHAR2(120)
 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)
```

USER_TAB_PRIVS :

```
SQL> select * from dict where table_name='USER_TAB_PRIVS';

TABLE_NAME
------------------------------
COMMENTS
--------------------------------------------------------------------------------
USER_TAB_PRIVS
Grants on objects for which the user is the owner, grantor or grantee
```

Liste les informations concernant les objets dont l'utilisateur est propriétaire,beneficière ou createur.

Sa structure :

```
SQL> desc user_tab_privs
 Name                                      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) Trouver le nom d'utilisateur avec lequel nous sommes connectés :**
Nous utilisons la table USER_USERS

```
SQL> select username from user_users;

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

**4) Comparer la structure et le contenu des tables ALL_TAB_COLUMNS et USER_TAB_COLUMNS:**

USER_TAB_COLUMNS :
Elle liste les informations concernant toutes les colonnes de toutes les tables que POSSEDES l'utilisateur, ce qui fait que ALL_TAB_COLUMNS possède une colonne en plus 'OWNER' qui spécifie à qui appartient la table.

```
SQL> desc user_tab_columns
 Name                                      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(120)
 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) Vérifier que les tables du TP1 ont été réellement créées et donner toutes les informations sur ces tables :**

Nous utilisons la table ALL_TABLES

```
SQL> desc all_tables
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 OWNER                                     NOT NULL VARCHAR2(30)
 TABLE_NAME                                NOT NULL VARCHAR2(30)
 TABLESPACE_NAME                                    VARCHAR2(30)
 CLUSTER_NAME                                       VARCHAR2(30)
 IOT_NAME                                           VARCHAR2(30)
 STATUS                                             VARCHAR2(8)
 PCT_FREE                                           NUMBER
 PCT_USED                                           NUMBER
 INI_TRANS                                          NUMBER
 MAX_TRANS                                          NUMBER
 INITIAL_EXTENT                                     NUMBER
 NEXT_EXTENT                                        NUMBER
 MIN_EXTENTS                                        NUMBER
 MAX_EXTENTS                                        NUMBER
 PCT_INCREASE                                       NUMBER
 FREELISTS                                          NUMBER
 FREELIST_GROUPS                                    NUMBER
 LOGGING                                            VARCHAR2(3)
 BACKED_UP                                          VARCHAR2(1)
 NUM_ROWS                                           NUMBER
 BLOCKS                                             NUMBER
 EMPTY_BLOCKS                                       NUMBER
 AVG_SPACE                                          NUMBER
 CHAIN_CNT                                          NUMBER
 AVG_ROW_LEN                                        NUMBER
 AVG_SPACE_FREELIST_BLOCKS                          NUMBER
 NUM_FREELIST_BLOCKS                                NUMBER
 DEGREE                                             VARCHAR2(40)
 INSTANCES                                          VARCHAR2(40)
 CACHE                                              VARCHAR2(20)
 TABLE_LOCK                                         VARCHAR2(8)
 SAMPLE_SIZE                                        NUMBER
 LAST_ANALYZED                                      DATE
 PARTITIONED                                        VARCHAR2(3)
 IOT_TYPE                                           VARCHAR2(12)
```

Nous recherchons alors les tables appartenant à DBAHOPITAL :

```
SQL> select table_name from all_tables where owner='DBAHOPITAL';

TABLE_NAME
------------------------------
HOSPITALISATION
MEDECIN
EMPLOYE
SERVICE
INFIRMIER
CHAMBRE
PATIENT
SOIGNE

8 rows selected.
```

Puisque le nombre d'informations est important, nous nous limitons à afficher uniquement les noms des tables et de leurs tablespaces (Nous appliquons de même pour les questions qui suivent) :

```
SQL> select table_name,tablespace_name from all_tables where owner='DBAHOPITAL';

TABLE_NAME                     TABLESPACE_NAME
------------------------------ ------------------------------
HOSPITALISATION                HOPITAL_TBS
MEDECIN                        HOPITAL_TBS
EMPLOYE                        HOPITAL_TBS
SERVICE                        HOPITAL_TBS
INFIRMIER                      HOPITAL_TBS
CHAMBRE                        HOPITAL_TBS
PATIENT                        HOPITAL_TBS
SOIGNE                         HOPITAL_TBS

8 rows selected.
```

**6) Lister les tables de l'utilisateur « SYSTEM » et celles de l'utilisateur DBAHOPITAL (l'utilisateur de TP1) :**

SYSTEM :

```
SQL> select table_name from all_tables where owner='SYSTEM';

TABLE_NAME
------------------------------
LOGMNR_GLOBAL$
LOGMNR_RESTART_CKPT_TXINFO$
LOGMNR_SESSION_ACTIONS$
LOGMNR_SESSION_EVOLVE$
LOGSTDBY$FLASHBACK_SCN
LOGMNR_PARAMETER$
LOGMNR_SESSION$
LOGMNR_FILTER$
MVIEW$_ADV_WORKLOAD
MVIEW$_ADV_BASETABLE
MVIEW$_ADV_SQLDEPEND

TABLE_NAME
------------------------------
MVIEW$_ADV_PRETTY
MVIEW$_ADV_TEMP
MVIEW$_ADV_FILTER
MVIEW$_ADV_LOG
MVIEW$_ADV_FILTERINSTANCE
MVIEW$_ADV_LEVEL
MVIEW$_ADV_ROLLUP
MVIEW$_ADV_AJG
MVIEW$_ADV_FJG
MVIEW$_ADV_GC
```

DBAHOPITAL :

```
SQL> select table_name from all_tables where owner='DBAHOPITAL';

TABLE_NAME
------------------------------
HOSPITALISATION
MEDECIN
EMPLOYE
SERVICE
INFIRMIER
CHAMBRE
PATIENT
SOIGNE

8 rows selected.
```

**7) Donner la description des attributs des tables PATIENT et HOSPITALISATION (Exploiter la table USER_TAB_COLUMNS) :**

Pour lister les descriptions des attributs des tables "Hospitalisation" et "Patient" en exploitant USER_TAB_COLUMNS, nous devons d'abord nous connecter à l'utilisateur DBAHOPITAL, ou bien en utilisant la table ALL_TAB_COLUMNS étant connectés depuis l'utilisateur SYSTEM.

Avec USER_TAB_COLUMNS :

```
SQL> select column_name,data_type,data_length,nullable from user_tab_columns where ta
ble_name='HOSPITALISATION';

COLUMN_NAME
------------------------------
DATA_TYPE
--------------------------------------------------------------------------------
DATA_LENGTH N
----------- -
NUM_PATIENT
NUMBER
         22 N

CODE_SERVICE
VARCHAR2
          3 Y

COLUMN_NAME
------------------------------
DATA_TYPE
--------------------------------------------------------------------------------
DATA_LENGTH N
----------- -

NUM_CHAMBRE
NUMBER
         22 Y

LIT
NUMBER

COLUMN_NAME
------------------------------
DATA_TYPE
--------------------------------------------------------------------------------
DATA_LENGTH N
----------- -
         22 Y
```

Avec ALL_TAB_COLUMNS :

```
SQL> select column_name,data_type,data_length,nullable from ALL_tab_columns where tab
le_name='PATIENT';

COLUMN_NAME
------------------------------
DATA_TYPE
--------------------------------------------------------------------------------
DATA_LENGTH N
----------- -
NUM_PATIENT
NUMBER
         22 N

NOM_PATIENT
VARCHAR2
         20 Y

COLUMN_NAME
------------------------------
DATA_TYPE
--------------------------------------------------------------------------------
DATA_LENGTH N
----------- -

PRENOM_PATIENT
VARCHAR2
         20 Y

ADR_PAT
VARCHAR2

COLUMN_NAME
------------------------------
DATA_TYPE
--------------------------------------------------------------------------------
DATA_LENGTH N
----------- -
         60 Y

TEL_PATIENT
NUMBER
         22 Y
```

**8) Verifier s'il y a une référence de clé étrangère entre les tables PATIENT et HOSPITALISATION :**

Pour cela nous utilisons la table ALL_CONSTRAINTS. Nous remarquons qu'elle porte information, dans chaque tuple, sur le nom de la contrainte référencée dans l'attribut «R_CONSTRAINT_NAME», nous allons donc rechercher les contraintes definies sur la table HOSPITALISATION qui sont dejà des contraintes de la table PATIENT:

```
SQL> select constraint_name,constraint_type,r_constraint_name from all_constraints
where table_name='HOSPITALISATION' and r_constraint_name in (select constraint_name
 from all_constraints where table_name='PATIENT');

CONSTRAINT_NAME               C R_CONSTRAINT_NAME
------------------------------ - ------------------------------
CE1HOSPITALISATION            R CPPATIENT
```

**9) Donner toutes les contraintes créées lors du TP1 et les informations qui les caractérisent (Exploitez la table USER_CONSTRAINTS) :**

Sa structure:

```
SQL> desc user_constraints
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 OWNER                                              VARCHAR2(120)
 CONSTRAINT_NAME                           NOT NULL VARCHAR2(30)
 CONSTRAINT_TYPE                                    VARCHAR2(1)
 TABLE_NAME                                NOT NULL VARCHAR2(30)
 SEARCH_CONDITION                                   LONG
 R_OWNER                                            VARCHAR2(120)
 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)
```

Les informations des contraintes du TP1 :

```
SQL> select constraint_name,constraint_type,table_name from user_constraints where
table_name in('HOSPITALISATION','SOIGNE','PATIENT','MEDECIN','SERVICE','EMPLOYE','C
HAMBRE','INFIRMIER');

CONSTRAINT_NAME               C TABLE_NAME
------------------------------ - ------------------------------
SYS_C007035                   C PATIENT
SYS_C007056                   C MEDECIN
CHMEDECIN                     C MEDECIN
SYS_C007034                   C INFIRMIER
CHINFIRMIER                   C INFIRMIER
CE2SOIGNE                     R SOIGNE
CE1SOIGNE                     R SOIGNE
CESERVICE                     R SERVICE
CEMEDECIN                     R MEDECIN
CE2INFIRMIER                  R INFIRMIER
CEINFIRMIER                   R INFIRMIER

CONSTRAINT_NAME               C TABLE_NAME
------------------------------ - ------------------------------
CE2HOSPITALISATION            R HOSPITALISATION
CE1HOSPITALISATION            R HOSPITALISATION
CE2CHAMBRE                    R CHAMBRE
CE1CHAMBRE                    R CHAMBRE
CPSOIGNE                      P SOIGNE
USERVICE                      U SERVICE
CPSERVICE                     P SERVICE
CPPATIENT                     P PATIENT
CPMEDECIN                     P MEDECIN
CPINFIRMIER                   P INFIRMIER
UHOSPITALISATION              U HOSPITALISATION

CONSTRAINT_NAME               C TABLE_NAME
------------------------------ - ------------------------------
CPHOSPITALISATION             P HOSPITALISATION
CPEMPLOYE                     P EMPLOYE
CPCHAMBRE                     P CHAMBRE
CHSALAIREINFIRMIER            C INFIRMIER

26 rows selected.
```

**10) Retrouver toutes les informations permettant de recréer la table HOSPITALISATION :**

Pour cela nous devons retrouver la structure, les contraintes, les indexes et les privilèges qui sont definis sur cette table.

Sa structure :

```
SQL> desc dbahopital.hospitalisation
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 NUM_PATIENT                               NOT NULL NUMBER(3)
 CODE_SERVICE                                       VARCHAR2(3)
 NUM_CHAMBRE                                        NUMBER(3)
 LIT                                                NUMBER(1)
```

Ses contraintes :

```
SQL> select constraint_name,constraint_type,r_constraint_name,status,r_owner from all
_constraints where table_name='HOSPITALISATION';

CONSTRAINT_NAME               C R_CONSTRAINT_NAME             STATUS
----------------------------- - ------------------------------ --------
R_OWNER
--------------------------------------------------------------------------------
CE1HOSPITALISATION            R CPPATIENT                      ENABLED
DBAHOPITAL

CE2HOSPITALISATION            R CPCHAMBRE                      ENABLED
DBAHOPITAL

CPHOSPITALISATION             P                                ENABLED



CONSTRAINT_NAME               C R_CONSTRAINT_NAME             STATUS
----------------------------- - ------------------------------ --------
R_OWNER
--------------------------------------------------------------------------------
UHOSPITALISATION              U                                ENABLED
```

Ses indexes :

```
SQL> select index_owner, index_name,table_owner,column_name from all_ind_columns where
 table_name='HOSPITALISATION';

INDEX_OWNER                    INDEX_NAME
------------------------------ ------------------------------
TABLE_OWNER
------------------------------
COLUMN_NAME
--------------------------------------------------------------------------------
DBAHOPITAL                     CPHOSPITALISATION
DBAHOPITAL
NUM_PATIENT

DBAHOPITAL                     UHOSPITALISATION
DBAHOPITAL
CODE_SERVICE

INDEX_OWNER                    INDEX_NAME
------------------------------ ------------------------------
TABLE_OWNER
------------------------------
COLUMN_NAME
--------------------------------------------------------------------------------

DBAHOPITAL                     UHOSPITALISATION
DBAHOPITAL
NUM_CHAMBRE

DBAHOPITAL                     UHOSPITALISATION
DBAHOPITAL

INDEX_OWNER                    INDEX_NAME
------------------------------ ------------------------------
TABLE_OWNER
------------------------------
COLUMN_NAME
--------------------------------------------------------------------------------
LIT
```

Les privilèges :

```
SQL> desc all_tab_privs
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 GRANTOR                                   NOT NULL VARCHAR2(30)
 GRANTEE                                   NOT NULL VARCHAR2(30)
 TABLE_SCHEMA                              NOT NULL VARCHAR2(30)
 TABLE_NAME                                NOT NULL VARCHAR2(30)
 PRIVILEGE                                 NOT NULL VARCHAR2(40)
 GRANTABLE                                          VARCHAR2(3)
 HIERARCHY                                          VARCHAR2(3)

SQL> select grantor,grantee,table_schema,privilege,grantable,hierarchy from all_tab_pr
ivs where table_name='HOSPITALISATION';

GRANTOR                        GRANTEE
------------------------------ ------------------------------
TABLE_SCHEMA                   PRIVILEGE                                GRA HIE
------------------------------ ---------------------------------------- --- ---
DBAHOPITAL                     GESTIONNAIREPATIENT
DBAHOPITAL                     SELECT                                   NO  NO

DBAHOPITAL                     GESTIONNAIREPATIENT
DBAHOPITAL                     UPDATE                                   NO  NO
```

**11) Trouver tous les privilèges accordés à ADMINHOPITAL :**

Nous exploitons alors les tables DBA_SYS_PRIVS,DBA_TAB_PRIVS,DBA_COL_PRIVS dont les rôles sont trouvés depuis le catalogue DICT

```
SQL> select * from dict where table_name='DBA_SYS_PRIVS';

TABLE_NAME
------------------------------
COMMENTS
--------------------------------------------------------------------------------
DBA_SYS_PRIVS
System privileges granted to users and roles


SQL> select * from dict where table_name='DBA_TAB_PRIVS';

TABLE_NAME
------------------------------
COMMENTS
--------------------------------------------------------------------------------
DBA_TAB_PRIVS
All grants on objects in the database


SQL> select * from dict where table_name='DBA_COL_PRIVS';

TABLE_NAME
------------------------------
COMMENTS
--------------------------------------------------------------------------------
DBA_COL_PRIVS
All grants on columns in the database
```

Les privilèges accordés à ADMINHOPITAL :

```
SQL> select * from dba_sys_privs where grantee='ADMINHOPITAL';

no rows selected

SQL> select * from dba_tab_privs where grantee='ADMINHOPITAL';

no rows selected

SQL> select * from dba_col_privs where grantee='ADMINHOPITAL';

no rows selected
```

**12) Trouver les rôles donnés à l'utilisateur ADMINHOPITAL :**

Nous executons la commande :'select * from dict where comments like '%role%';' ,parmi les résultats nous trouvons la table "DBA_ROLE_PRIVS", qui contient les rôles donnés aux utilisateurs et aux rôles.

```
SQL> select * from dict where table_name='DBA_ROLE_PRIVS';

TABLE_NAME
------------------------------
COMMENTS
--------------------------------------------------------------------------------
DBA_ROLE_PRIVS
Roles granted to users and roles
```

Sa structure et les rôles donnés à ADMINHOPITAL :

```
SQL> desc dba_role_privs
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 GRANTEE                                            VARCHAR2(30)
 GRANTED_ROLE                              NOT NULL VARCHAR2(30)
 ADMIN_OPTION                                       VARCHAR2(3)
 DEFAULT_ROLE                                       VARCHAR2(3)

SQL> select * from dba_role_privs where grantee='ADMINHOPITAL';

GRANTEE                        GRANTED_ROLE                   ADM DEF
------------------------------ ------------------------------ --- ---
ADMINHOPITAL                   GESTIONNAIREPATIENT             NO  YES
```

**13) Trouver tous les objets appartenant à ADMINHOPITAL :**

Pour cela nous utilisons la table ALL_OBJECTS

```
SQL> desc all_objects
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 OWNER                                     NOT NULL VARCHAR2(30)
 OBJECT_NAME                               NOT NULL VARCHAR2(30)
 SUBOBJECT_NAME                                     VARCHAR2(30)
 OBJECT_ID                                 NOT NULL NUMBER
 DATA_OBJECT_ID                                     NUMBER
 OBJECT_TYPE                                        VARCHAR2(19)
 CREATED                                   NOT NULL DATE
 LAST_DDL_TIME                             NOT NULL DATE
 TIMESTAMP                                          VARCHAR2(19)
 STATUS                                             VARCHAR2(7)
 TEMPORARY                                          VARCHAR2(1)
 GENERATED                                          VARCHAR2(1)
 SECONDARY                                          VARCHAR2(1)
 NAMESPACE                                 NOT NULL NUMBER
 EDITION_NAME                                       VARCHAR2(30)

SQL> select * from all_objects where owner='ADMINHOPITAL';

OWNER                          OBJECT_NAME
------------------------------ ------------------------------
SUBOBJECT_NAME                  OBJECT_ID DATA_OBJECT_ID OBJECT_TYPE
------------------------------ ---------- -------------- -------------------
CREATED  LAST_DDL TIMESTAMP           STATUS  T G S  NAMESPACE
-------- -------- ------------------- ------- - - - ----------
EDITION_NAME
------------------------------
ADMINHOPITAL                   EMPLOYENOM_IX
                                    20412          20412 INDEX
12/10/18 12/10/18 2018-10-12:12:21:46 VALID   N N N          4
```

**14) L'administrateur cherche le propriétaire de la table HOSPITALISATION :**

Nous utilisons la table alors ALL  TABLES

```
SQL> select owner from all_tables where table_name='HOSPITALISATION';

OWNER
------------------------------
DBAHOPITAL
```

**15) Donner la taille en Ko de la table HOSPITALISATION :**

Nous avons la table DBA_EXTENTS qui contient les informations sur les caractèristiques du stockage pour tous les segments dans la base de données ; elle comporte l'attribut «Bytes» qui représente la taille du segment en octet.
Nous utilisons donc cet attribut pour avoir la taille de la table HOSPITALISATION, et en le divisant par 1024 pour obtenir cette taille en KO.

```
SQL> select * from dict where table_name='DBA_EXTENTS';

TABLE_NAME
------------------------------
COMMENTS
--------------------------------------------------------------------------------
DBA_EXTENTS
Extents comprising all segments in the database


SQL> DESC dba_extents
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 OWNER                                              VARCHAR2(30)
 SEGMENT_NAME                                       VARCHAR2(81)
 PARTITION_NAME                                     VARCHAR2(30)
 SEGMENT_TYPE                                       VARCHAR2(18)
 TABLESPACE_NAME                                    VARCHAR2(30)
 EXTENT_ID                                          NUMBER
 FILE_ID                                            NUMBER
 BLOCK_ID                                           NUMBER
 BYTES                                              NUMBER
 BLOCKS                                             NUMBER
 RELATIVE_FNO                                       NUMBER

SQL> select bytes/1024 as Table_size_KO from dba_extents where segment_name='HOSPITALI
SATION' and owner='DBAHOPITAL';

TABLE_SIZE_KO
-------------
           64
```

**16) Vérifier l'effet produit par chacune des commandes de définition de données du TP1 sur le dictionnaire :**

Les opérations effectuées dans le TP1 ont eu comme effet des repercussions sur le dictionnaire. Comme nous n'avons pas enregistré des verifications de l'état de ce dernier avant ces opérations, nous allors donc créer des objets pour tester quelques opérations du TP1 et verifier leurs repercussions.

La création d'un utilisateur:

Avant :

```
SQL> select username from all_users;

USERNAME
------------------------------
XS$NULL
APEX_040000
APEX_PUBLIC_USER
FLOWS_FILES
HR
MDSYS
ANONYMOUS
XDB
CTXSYS
APPQOSSYS
DBSNMP

USERNAME
------------------------------
ORACLE_OCM
DIP
OUTLN
SYSTEM
SYS
ADMINHOPITAL
DBAHOPITAL

18 rows selected.
```

Après :

```
SQL> create user dbahopitaltest identified by chikh3;

User created.

SQL> select username from all_users
  2  ;

USERNAME
------------------------------
XS$NULL
DBAHOPITALTEST
APEX_040000
APEX_PUBLIC_USER
FLOWS_FILES
HR
MDSYS
ANONYMOUS
XDB
CTXSYS
APPQOSSYS

USERNAME
------------------------------
DBSNMP
ORACLE_OCM
DIP
OUTLN
SYSTEM
SYS
ADMINHOPITAL
DBAHOPITAL

19 rows selected.
```

La creation d'une table :

```
SQL> select * from all_tables where table_name='EMPTEST';

no rows selected

SQL> select count(*) from all_tables;

  COUNT(*)
----------
      1697

SQL> create table emptest (num number(10) primary key,nom varchar2(20),prenom varch
ar2(20));

Table created.

SQL> select owner,tablespace_name,status from all_tables where table_name='EMPTEST'
;

OWNER                          TABLESPACE_NAME                STATUS
------------------------------ ------------------------------ --------
SYSTEM                         SYSTEM                         VALID

SQL> select count(*) from all_tables;

  COUNT(*)
----------
      1698
```

L'ajout d'une colonne :

```
SQL> select table_name,column_name from all_tab_columns where table_name='EMPTEST'
;

TABLE_NAME                     COLUMN_NAME
------------------------------ ------------------------------
EMPTEST                        NUM
EMPTEST                        NOM
EMPTEST                        PRENOM

SQL> select count(*) from all_tab_columns;

  COUNT(*)
----------
     73412

SQL> alter table emptest add salaire number(6);

Table altered.

SQL> select table_name,column_name from all_tab_columns where table_name='EMPTEST'
;

TABLE_NAME                     COLUMN_NAME
------------------------------ ------------------------------
EMPTEST                        NUM
EMPTEST                        NOM
EMPTEST                        PRENOM
EMPTEST                        SALAIRE

SQL> select count(*) from all_tab_columns;

  COUNT(*)
----------
     73413
```

L'ajout d'une contrainte :

```
SQL> select constraint_name,constraint_type from all_constraints where table_name=
'EMPTEST';

CONSTRAINT_NAME                C
------------------------------ -
SYS_C007135                    P

SQL> select count(*) from all_constraints;

  COUNT(*)
----------
      6996

SQL> alter table emptest add constraint chemptest check (salaire>50000);

Table altered.

SQL> select constraint_name,constraint_type from all_constraints where table_name=
'EMPTEST';

CONSTRAINT_NAME                C
------------------------------ -
SYS_C007135                    P
CHEMPTEST                      C

SQL> select count(*) from all_constraints;

  COUNT(*)
----------
      6997
```

L'accord d'un privilège système:

```
SQL> select * from dba_sys_privs where grantee='DBAHOPITALTEST';

no rows selected

SQL> select count(*) from dba_sys_privs;

  COUNT(*)
----------
       843

SQL> grant create session to dbahopitaltest;

Grant succeeded.

SQL> select * from dba_sys_privs where grantee='DBAHOPITALTEST';

GRANTEE                        PRIVILEGE                                ADM
------------------------------ ---------------------------------------- ---
DBAHOPITALTEST                 CREATE SESSION                           NO

SQL> select count(*) from dba_sys_privs;

  COUNT(*)
----------
       844
```

L'accord d'un rôle :

```
SQL> select * from dba_role_privs where grantee='DBAHOPITALTEST';

no rows selected

SQL> create role dbaroletest;

Role created.

SQL> grant select on system.emptest to dbaroletest;

Grant succeeded.

SQL> grant dbaroletest to dbahopitaltest;

Grant succeeded.

SQL> conn dbahopitaltest/chikh3
Connected.
SQL> select * from user_role_privs;

USERNAME                       GRANTED_ROLE                   ADM DEF OS_
------------------------------ ------------------------------ --- --- ---
DBAHOPITALTEST                 DBAROLETEST                    NO  YES NO

SQL> conn system
Enter password:
Connected.
SQL> select * from dba_role_privs where grantee='DBAHOPITALTEST';

GRANTEE                        GRANTED_ROLE                   ADM DEF
------------------------------ ------------------------------ --- ---
DBAHOPITALTEST                 DBAROLETEST                    NO  YES

SQL> revoke dbaroletest from dbahopitaltest;

Revoke succeeded.

SQL> conn dbahopitaltest/chikh3
Connected.
SQL> select * from user_role_privs;

no rows selected
```