ASGBD

Master 1 IL
Groupe 2
2018

Rapport de TP N°3 ASGBD : DICTIONNAIRE DE DONNÉES ORACLE

BOUDOUR Mehdi / 201500008386/ TP: ORACLE (3) DICTIONNAIRE DE DONNÉES ORACLE

[ARCHITECTURE DES S.G.B.D. RELATIONNELS]

E-mail : geronimotoutcourt@gmail.com

1) Connecter en tant que « System ». Lister le catalogue « DICT ».

```
--1)
connect SYSTEM/pwd;

SELECT * FROM DICT;
```

```
GV$_LOCK
Synonym for GV_$_LOCK


TABLE_NAME
------------------------------
COMMENTS
--------------------------------------------------------------------------------
IND
Synonym for USER_INDEXES

ALL_JOBS
Synonym for USER_JOBS


2551 rows selected.

SQL>
```

Il contient combien d'instances ? **NOMBRE D'INSTANCES = 2551 rows selected.**

Donner sa structure ?

```
DESC DICT;
```

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

REMARQUE :
Structure(DICT) = **DICT** (**TABLE_NAME** VARCHAR2(30) , **COMMECNTS** VARCHAR2(4000)).

2) Donner le rôle et la structure des tables (ou vues) suivantes : ALL_TAB_COLUMNS, USER_USERS, ALL_CONSTRAINTS et USER_TAB_PRIVS.

```
--2)
--ALL_TAB_COLUMNS
SELECT COMMENTS AS ROLE_DE_ALL_TAB_COLUMNS
FROM   DICT
WHERE TABLE_NAME = 'ALL_TAB_COLUMNS';
```

```
SQL> --2)
SQL>
SQL> --ALL_TAB_COLUMNS
SQL> SELECT COMMENTS AS ROLE_DE_ALL_TAB_COLUMNS
  2  FROM   DICT
  3  WHERE TABLE_NAME = 'ALL_TAB_COLUMNS';

ROLE_DE_ALL_TAB_COLUMNS
--------------------------------------------------------------------------------
Columns of user's tables, views and clusters
```

REMARQUE :
- **ALL_TAB_COLUMNS** : Contient les informations des colonnes des tables et vues et clusters accessible par le USER (que l'on peut voir).

```sql
--USER_USERS
SELECT COMMENTS AS ROLE_DE_USER_USERS
FROM   DICT
WHERE TABLE_NAME = 'USER_USERS';
```

```
SQL> --USER_USERS
SQL> SELECT COMMENTS AS ROLE_DE_USER_USERS
  2  FROM   DICT
  3  WHERE TABLE_NAME = 'USER_USERS';

ROLE_DE_USER_USERS
--------------------------------------------------------------------------------
Information about the current user
```

REMARQUE :
- **USER_USERS** : Contient des informations sur l'utilisateur courant.

```sql
--ALL_CONSTRAINTS
SELECT COMMENTS AS ROLE_DE_ALL_CONSTRAINTS
FROM   DICT
WHERE TABLE_NAME = 'ALL_CONSTRAINTS';
```

```
SQL> --ALL_CONSTRAINTS
SQL> SELECT COMMENTS AS ROLE_DE_ALL_CONSTRAINTS
  2  FROM   DICT
  3  WHERE TABLE_NAME = 'ALL_CONSTRAINTS';

ROLE_DE_ALL_CONSTRAINTS
--------------------------------------------------------------------------------
Constraint definitions on accessible tables
```

REMARQUE :
- **ALL_CONSTRAINTS** : Contient les définitions des contraintes sur les table accessibles par le USER (que l'on peut voir) .

```sql
--USER_TAB_PRIVS
SELECT COMMENTS AS ROLE_DE_USER_TAB_PRIVS
FROM   DICT
WHERE TABLE_NAME = 'USER_TAB_PRIVS';
```

```
SQL> --USER_TAB_PRIVS
SQL> SELECT COMMENTS AS ROLE_DE_USER_TAB_PRIVS
  2  FROM   DICT
  3  WHERE TABLE_NAME = 'USER_TAB_PRIVS';

ROLE_DE_USER_TAB_PRIVS
--------------------------------------------------------------------------------
Grants on objects for which the user is the owner, grantor or grantee
```

REMARQUE :
- **USER_TAB_PRIVS** : Contient les informations des privilèges sur les objets dont l'utilisateur est propriétaire, bénéficiaire ou cédant du privilège.

3) Trouver le nom d'utilisateur avec lequel vous êtes connecté ?

REMARQUE :
Réponse : le nom de l'utilisateur est une information qui concerne le USER courant (connecté) donc elle devrait se trouver dans **"USER_USERS"**d'après (Q2).**Alors :** on fait un **DESC USER_USERS** pour trouver la colonne correspondante puis un **SELECT** sur cette colonne.

```
--3)
DESC USER_USERS;
```

```
SQL> --3)
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)
```

REMARQUE :
La colonne qui nous intéresse est donc : **USERNAME**.

```
SELECT USERNAME FROM USER_USERS;
```

```
SQL> SELECT USERNAME FROM USER_USERS;

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

REMARQUE :
Le nom du USER avec lequel je suis connecté = **SYSTEM**.

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

```
--4)
--Comparer la structure
-------------------------------------------------------------
--STRUCTURE DE ALL_TAB_COLUMNS
-------------------------------------------------------------
DESC ALL_TAB_COLUMNS;
```

```
SQL> --4)
SQL> --Comparer la structure
SQL>
SQL> ------------------------------------------------------------
SQL> --STRUCTURE DE ALL_TAB_COLUMNS
SQL> ------------------------------------------------------------
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)
```

```
------------------------------------------------------------
--STRUCTURE DE USER_TAB_COLUMNS
------------------------------------------------------------
DESC USER_TAB_COLUMNS;
```

```
SQL> ------------------------------------------------------------
SQL> --STRUCTURE DE USER_TAB_COLUMNS
SQL> ------------------------------------------------------------
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)
```

REMARQUE :
ALL_TAB_COLUMNS possède exactement les mêmes colonnes que **USER_TAB_COLUMNS** avec en plus la colonne **OWNER** .
Structure(**ALL_TAB_COLUMNS**) = Structure(**USER_TAB_COLUMNS**) + [**OWNER VARCHAR2(30) NOT NULL**].

```
--Comparer le contenu
-------------------------------------------------------
--CONTENU DE USER_TAB_COLUMNS
-------------------------------------------------------
SELECT TABLE_NAME ,COLUMN_NAME FROM USER_TAB_COLUMNS;
```

```
SQL> --Comparer le contenu
SQL>
SQL> -------------------------------------------------------
SQL> --CONTENU DE USER_TAB_COLUMNS
SQL> -------------------------------------------------------
SQL> SELECT TABLE_NAME ,COLUMN_NAME FROM USER_TAB_COLUMNS;

TABLE_NAME                     COLUMN_NAME
------------------------------ ------------------------------
AQ$_QUEUES                     SUBSCRIBERS
AQ$_INTERNET_AGENTS            AGENT_NAME
AQ$_INTERNET_AGENTS            PROTOCOL
AQ$_INTERNET_AGENTS            SPARE1
AQ$_INTERNET_AGENT_PRIVS       AGENT_NAME
AQ$_INTERNET_AGENT_PRIVS       DB_USERNAME
AQ$_QUEUES                     OID
AQ$_QUEUES                     EVENTID
AQ$_QUEUES                     NAME
AQ$_QUEUES                     TABLE_OBJNO
AQ$_QUEUES                     USAGE

TABLE_NAME                     COLUMN_NAME
------------------------------ ------------------------------
AQ$_QUEUES                     ENABLE_FLAG
AQ$_QUEUES                     MAX_RETRIES
AQ$_QUEUES                     RETRY_DELAY
AQ$_QUEUES                     PROPERTIES
AQ$_QUEUES                     RET_TIME
AQ$_QUEUES                     QUEUE_COMMENT
AQ$_QUEUES                     MEMORY_THRESHOLD

TABLE_NAME                     COLUMN_NAME
------------------------------ ------------------------------
AQ$_DEF$_AQCALL_F              EXPIRATION
MVIEW_RECOMMENDATIONS          MVIEW_NAME
AQ$_DEF$_AQCALL_F              ENQ_TIME
AQ$DEF$_AQCALL                 ORIGINAL_QUEUE_OWNER
AQ$DEF$_AQCALL                 MSG_PRIORITY

1699 rows selected.
```

```
-------------------------------------------------------
--CONTENU DE ALL_TAB_COLUMNS
-------------------------------------------------------
SELECT OWNER,TABLE_NAME ,COLUMN_NAME FROM ALL_TAB_COLUMNS;
```

```
OWNER                          TABLE_NAME
------------------------------ ------------------------------
COLUMN_NAME
------------------------------
APEX_040000                    APEX_APPLICATION_PAGE_PROC
COMPONENT_SIGNATURE

SYS                            USER_FILE_GROUP_FILES
FILE_DIRECTORY

SYS                            EXU9GSAS
VALUE


OWNER                          TABLE_NAME
------------------------------ ------------------------------
COLUMN_NAME
------------------------------
SYS                            GV_$SYS_OPTIMIZER_ENV
INST_ID


73615 rows selected.
```

REMARQUE :

USER_TAB_COLUMNS: contient les infos de toute les colonnes des tables du **USER** courant (soit **SYSTEM**).

ALL_TAB_COLUMNS: contient les infos de toute les colonnes des tables accessibles par le USER courant avec en plus précision du propriétaire de la table via le champ **OWNER.**

REMARQUE :

Du coup Nombre(Lignes de **ALL_TAB_COLUMNS**) > Nombre(Lignes de **USER_TAB_COLUMNS**)
Exécutons un COUNT(*) afin de mieux cerner la différence entre les 2 Tables :

```
SELECT COUNT(*) AS COLONNES_DU_USER_COURANT FROM USER_TAB_COLUMNS;

SELECT OWNER , COUNT(*) AS NB_COLUMNS FROM ALL_TAB_COLUMNS GROUP BY OWNER ORDER BY NB_COLUMNS;
```

```
SQL> SELECT COUNT(*) AS COLONNES_DU_USER_COURANT FROM USER_TAB_COLUMNS;

COLONNES_DU_USER_COURANT
------------------------
                    1699

SQL>
SQL> SELECT OWNER , COUNT(*) AS NB_COLUMNS FROM ALL_TAB_COLUMNS GROUP BY OWNER ORDER BY NB_COLUMNS;

OWNER                          NB_COLUMNS
------------------------------ ----------
USER3                                   2
CASANOVA                               10
USER1                                  13
ALILAPOINTE                            14
ETUDIANT                               14
DONQUICHOTTE                           15
APPQOSSYS                              16
FLOWS_FILES                            22
ETUDIANT2                              25
DBHOPITAL                              31
FACSYS                                 36

OWNER                          NB_COLUMNS
------------------------------ ----------
OUTLN                                  43
HR                                     51
MEHDI                                  66
XDB                                   200
DBSNMP                                211
CTXSYS                                649
MDSYS                                 882
SYSTEM                               1699
APEX_040000                         13783
SYS                                 55833

21 rows selected.
```

REMARQUE :

Ainsi : Contenu(**USER_TAB_COLUMNS**) est inclu dans le Contenu(**ALL_TAB_COLUMNS**) et pour preuve le requête suivante est censé retourner **zéro ligne**.

```
(SELECT TABLE_NAME ,COLUMN_NAME FROM USER_TAB_COLUMNS)
MINUS
(SELECT TABLE_NAME ,COLUMN_NAME FROM ALL_TAB_COLUMNS
WHERE OWNER = (SELECT USERNAME FROM USER_USERS));
```

```
SQL> (SELECT TABLE_NAME ,COLUMN_NAME FROM USER_TAB_COLUMNS)
  2  MINUS
  3  (SELECT TABLE_NAME ,COLUMN_NAME FROM ALL_TAB_COLUMNS
  4  WHERE OWNER = (SELECT USERNAME FROM USER_USERS));

no rows selected
```

Resultat = **no rows selected.**

5) Vérifiez que les tables du TP1 ont été réellement créées **?**

```
--5)
SELECT TABLE_NAME
FROM ALL_TABLES
WHERE OWNER = 'DBHOPITAL';
```

```
SQL> --5)
SQL> SELECT TABLE_NAME
  2  FROM ALL_TABLES
  3  WHERE OWNER = 'DBHOPITAL';

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

8 rows selected.
```

REMARQUE :

les tables du TP1 ont été réellement créées.

Donner toutes les informations sur ces tables ?

```
SELECT *
FROM ALL_TABLES
WHERE OWNER = 'DBHOPITAL';
```

```
OWNER                          TABLE_NAME
------------------------------ ------------------------------
TABLESPACE_NAME                CLUSTER_NAME
------------------------------ ------------------------------
IOT_NAME                       STATUS     PCT_FREE    PCT_USED  INI_TRANS
------------------------------ -------- ---------- ---------- ----------
 MAX_TRANS INITIAL_EXTENT NEXT_EXTENT MIN_EXTENTS MAX_EXTENTS PCT_INCREASE
---------- -------------- ----------- ----------- ----------- ------------
 FREELISTS FREELIST_GROUPS LOG B    NUM_ROWS     BLOCKS EMPTY_BLOCKS  AVG_SPACE
---------- --------------- --- - ---------- ---------- ------------ ----------
 CHAIN_CNT AVG_ROW_LEN AVG_SPACE_FREELIST_BLOCKS NUM_FREELIST_BLOCKS
---------- ----------- ------------------------- -------------------
DEGREE
----------------------------------------
INSTANCES                              CACHE                TABLE_LO
-------------------------------------- -------------------- --------
SAMPLE_SIZE LAST_ANA PAR IOT_TYPE     T S NES BUFFER_ FLASH_C CELL_FL ROW_MOVE
----------- -------- --- ------------ - - --- ------- ------- ------- --------
GLO USE DURATION        SKIP_COR MON CLUSTER_OWNER                   DEPENDEN
--- --- --------------- -------- --- ------------------------------ --------
COMPRESS COMPRESS_FOR DRO REA SEG RESULT_
-------- ------------ --- --- --- -------


8 rows selected.
```

6) Lister les tables de l'utilisateur « system » et celles de l'utilisateur DBA_HOPITAL (l'utilisateur de TP1).

```
--6)
SELECT OWNER,TABLE_NAME
FROM  ALL_TABLES
WHERE OWNER = 'SYSTEM'
OR    OWNER = 'DBHOPITAL'
ORDER BY OWNER;
```

```
SQL> --6)
SQL> SELECT OWNER,TABLE_NAME
  2  FROM  ALL_TABLES
  3  WHERE OWNER = 'SYSTEM'
  4  OR    OWNER = 'DBHOPITAL'
  5  ORDER BY OWNER;

OWNER                          TABLE_NAME
------------------------------ ------------------------------
DBHOPITAL                      INFIRMIER
DBHOPITAL                      CHAMBRE
DBHOPITAL                      EMPLOYE
DBHOPITAL                      HOSPITALISATION
DBHOPITAL                      MEDECIN
DBHOPITAL                      SOIGNE
DBHOPITAL                      SERVICE
DBHOPITAL                      PATIENT
SYSTEM                         LOGMNR_GLOBAL$
SYSTEM                         LOGMNR_RESTART_CKPT_TXINFO$
SYSTEM                         LOGMNR_SESSION_ACTIONS$
```

```
OWNER                          TABLE_NAME
------------------------------ ------------------------------
SYSTEM                         LOGSTDBY$APPLY_PROGRESS

166 rows selected.
```

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

```
--7)
connect DBHOPITAL/pwd;

SELECT *
FROM USER_TAB_COLUMNS
WHERE TABLE_NAME = 'PATIENT';
```

```
SQL> --7)
SQL> connect DBHOPITAL/pwd;
Connected.
SQL>
SQL> SELECT *
  2  FROM USER_TAB_COLUMNS
  3  WHERE TABLE_NAME = 'PATIENT';

TABLE_NAME                     COLUMN_NAME
------------------------------ ------------------------------
DATA_TYPE
--------------------------------------------------------------------------------
DAT
---
DATA_TYPE_OWNER
--------------------------------------------------------------------------------
DATA_LENGTH DATA_PRECISION DATA_SCALE N  COLUMN_ID DEFAULT_LENGTH
----------- -------------- ---------- - ---------- --------------
DATA_DEFAULT
--------------------------------------------------------------------------------
NUM_DISTINCT LOW_VALUE
------------ ----------------------------------------------------------------
HIGH_VALUE                                                          DENSITY
------------------------------------------------------------------- ----------
 NUM_NULLS NUM_BUCKETS LAST_ANA SAMPLE_SIZE
---------- ----------- -------- -----------
CHARACTER_SET_NAME                          CHAR_COL_DECL_LENGTH GLO USE
------------------------------------------- -------------------- --- ---
AVG_COL_LEN CHAR_LENGTH C V80 DAT HISTOGRAM
----------- ----------- - --- --- ---------------
PATIENT                        NUM_PATIENT
```

```
TABLE_NAME                     COLUMN_NAME
------------------------------ ------------------------------
DATA_TYPE
--------------------------------------------------------------------------------
DAT
---
DATA_TYPE_OWNER
--------------------------------------------------------------------------------
DATA_LENGTH DATA_PRECISION DATA_SCALE N  COLUMN_ID DEFAULT_LENGTH
----------- -------------- ---------- - ---------- --------------
DATA_DEFAULT
--------------------------------------------------------------------------------
NUM_DISTINCT LOW_VALUE
------------ ----------------------------------------------------------------
HIGH_VALUE                                                          DENSITY
------------------------------------------------------------------- ----------
 NUM_NULLS NUM_BUCKETS LAST_ANA SAMPLE_SIZE
---------- ----------- -------- -----------
CHARACTER_SET_NAME                          CHAR_COL_DECL_LENGTH GLO USE
------------------------------------------- -------------------- --- ---
AVG_COL_LEN CHAR_LENGTH C V80 DAT HISTOGRAM
----------- ----------- - --- --- ---------------


6 rows selected.
```

```
SELECT *
FROM USER_TAB_COLUMNS
WHERE TABLE_NAME = 'HOSPITALISATION';
```

```
SQL> SELECT *
  2  FROM USER_TAB_COLUMNS
  3  WHERE TABLE_NAME = 'HOSPITALISATION';

TABLE_NAME                     COLUMN_NAME
------------------------------ ------------------------------
DATA_TYPE
--------------------------------------------------------------------------------
DAT
---
DATA_TYPE_OWNER
--------------------------------------------------------------------------------
DATA_LENGTH DATA_PRECISION DATA_SCALE N  COLUMN_ID DEFAULT_LENGTH
----------- -------------- ---------- - ---------- --------------
DATA_DEFAULT
--------------------------------------------------------------------------------
NUM_DISTINCT LOW_VALUE
------------ ------------------------------------------------------------------
HIGH_VALUE                                                          DENSITY
------------------------------------------------------------------- ----------
 NUM_NULLS NUM_BUCKETS LAST_ANA SAMPLE_SIZE
---------- ----------- -------- -----------
CHARACTER_SET_NAME                        CHAR_COL_DECL_LENGTH GLO USE
----------------------------------------- -------------------- --- ---
AVG_COL_LEN CHAR_LENGTH C V80 DAT HISTOGRAM
----------- ----------- - --- --- ---------------
HOSPITALISATION                NUM_PATIENT
```

8) Comment peut-on vérifie qu'il y a une référence de clé étrangère entre les tables PATIENT et HOSPITALISATION ?

REMARQUE :

On execute un **SELECT** sur la table **ALL_CONSTRAINTS** de manière à afficher les contraintes de clé étrangère (cad **CONSTRAINT_TYPE = 'R'**) appartenant à **HOSPITALISATION** ou **PATIENT** qui référence une clé primaire ou un attribut **UNIQUE** de **PATIENT** ou **HOSPITALISATION** (respectivement) où **R_CONSTRAINT_NAME** appartient à l'autre table.

```
--8)
connect SYSTEM/pwd;

SELECT  CONSTRAINT_NAME,CONSTRAINT_TYPE ,TABLE_NAME,R_OWNER,R_CONSTRAINT_NAME
FROM ALL_CONSTRAINTS
WHERE OWNER = 'DBHOPITAL'
AND   TABLE_NAME IN ('PATIENT' , 'HOSPITALISATION')
AND   CONSTRAINT_TYPE = 'R'
AND R_CONSTRAINT_NAME IN
(SELECT CONSTRAINT_NAME
FROM ALL_CONSTRAINTS
WHERE OWNER = 'DBHOPITAL'
AND   CONSTRAINT_TYPE IN ('P' , 'U')
AND   TABLE_NAME IN ('PATIENT' , 'HOSPITALISATION'));
```

```
SQL> --8)
SQL> connect SYSTEM/pwd;
Connected.
SQL> SELECT  CONSTRAINT_NAME,CONSTRAINT_TYPE ,TABLE_NAME,R_OWNER,R_CONSTRAINT_NAME
  2  FROM ALL_CONSTRAINTS
  3  WHERE OWNER = 'DBHOPITAL'
  4  AND   TABLE_NAME IN ('PATIENT' , 'HOSPITALISATION')
  5  AND   CONSTRAINT_TYPE = 'R'
  6  AND R_CONSTRAINT_NAME IN
  7  (SELECT CONSTRAINT_NAME
  8  FROM ALL_CONSTRAINTS
  9  WHERE OWNER = 'DBHOPITAL'
 10  AND   CONSTRAINT_TYPE IN ('P' , 'U')
 11  AND   TABLE_NAME IN ('PATIENT' , 'HOSPITALISATION'));

CONSTRAINT_NAME                C TABLE_NAME
------------------------------ - ------------------------------
R_OWNER
--------------------------------------------------------------------------------
R_CONSTRAINT_NAME
------------------------------
FK_NUM_PATIENT_HOSPITALISATION R HOSPITALISATION
DBHOPITAL
PK_NUM_PATIENT
```

REMARQUE :

Il y a donc une contrainte de clé étrangère **FK_NUM_PATIENT_HOSPITALISATION** de la table **HOSPITALISATION** référençant la table PATIENT.

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

```
--9)
connect DBHOPITAL/pwd

SELECT * FROM USER_CONSTRAINTS;
```

```
SQL> --9)
SQL> connect DBHOPITAL/pwd
Connected.
SQL>
SQL> SELECT * FROM USER_CONSTRAINTS;

OWNER
------------------------------------------------------------------------------
CONSTRAINT_NAME                C TABLE_NAME
------------------------------ - ------------------------------
SEARCH_CONDITION
------------------------------------------------------------------------------
R_OWNER
------------------------------------------------------------------------------
R_CONSTRAINT_NAME              DELETE_RU STATUS   DEFERRABLE     DEFERRED
------------------------------ --------- -------- -------------- ---------
VALIDATED     GENERATED      BAD RELY LAST_CHA INDEX_OWNER
------------- -------------- --- ---- -------- ------------------------------
INDEX_NAME                     INVALID VIEW_RELATED
------------------------------ ------- --------------
DBHOPITAL
```

…… etc.

```
25 rows selected.
```

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

REMARQUE :

Pour ce qui est de cette question je présente 2 solutions :

Solution I : mais cette solution ne donne pas les contraintes vu que les donnée présente dans ALL_CNSTRAINTS et ALL_CONS_COLUMNS ne sufusent pas pour reprendre la définition de toutes le CI.

Extraire de **USER_TAB_COLUMNS** les **Noms**, **Types**, **Longueurs** des colonnes de la tables **HOSPITALISATION** et si elles peuvent être NULL ou non avec un **SELECT**.

```
--10)
/*Attributs*/
SELECT COLUMN_NAME, DATA_TYPE, DATA_LENGTH, NULLABLE ,COLUMN_ID
FROM USER_TAB_COLUMNS
WHERE TABLE_NAME = 'HOSPITALISATION';
```

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

COLUMN_NAME
------------------------------
DATA_TYPE
--------------------------------------------------------------------------------
DATA_LENGTH N  COLUMN_ID
----------- - ----------
NUM_PATIENT
NUMBER
         22 N          1

CODE_SERVICE
VARCHAR2
         20 Y          2

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

NUM_CHAMBRE
NUMBER
         22 Y          3

LIT
NUMBER

COLUMN_NAME
------------------------------
DATA_TYPE
--------------------------------------------------------------------------------
DATA_LENGTH N  COLUMN_ID
----------- - ----------
         22 Y          4
```

Solution II : Après une recherche sur le package **DBMS_METADATA.**

Utiliser la fonction **GET_DDL** du package **DBMS_METADATA** qui donne la commande DDL de création de la table ce qui constitue l'ensemble des informations, contraintes comprises, nécessaire à la création de la table **HOSPITALISATION**.

```
SET SERVEROUTPUT ON;
BEGIN
   DBMS_OUTPUT.PUT_LINE(DBMS_METADATA.GET_DDL( OBJECT_TYPE => 'TABLE', NAME =>
'HOSPITALISATION'));
END;
```

```
SQL> SET SERVEROUTPUT ON;
SQL> BEGIN
  2     DBMS_OUTPUT.PUT_LINE(DBMS_METADATA.GET_DDL( OBJECT_TYPE => 'TABLE'
  3                                               , NAME => 'HOSPITALISATION')
  4                          );
  5  END;
  6  /

  CREATE TABLE "DBHOPITAL"."HOSPITALISATION"
   (    "NUM_PATIENT" NUMBER(*,0),

        "CODE_SERVICE" VARCHAR2(20),
        "NUM_CHAMBRE" NUMBER(*,0),
        "LIT" NUMBER(*,0),

        "DATE_HOST" DATE,
         CONSTRAINT "PK_NUM_PATIENT_HOSPITALISATION" PRIMARY KEY
("NUM_PATIENT")
  USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT FLASH_CACHE
DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "HOPITAL_TBS"  ENABLE,

CONSTRAINT "FK_NUM_PATIENT_HOSPITALISATION" FOREIGN KEY ("NUM_PATIENT")

REFERENCES "DBHOPITAL"."PATIENT" ("NUM_PATIENT") ENABLE,
         CONSTRAINT
"FK_CODE_SERVICE_NUM_CHAMBRE" FOREIGN KEY ("CODE_SERVICE", "NUM_CHAMBRE")

REFERENCES "DBHOPITAL"."CHAMBRE" ("CODE_SERVICE", "NUM_CHAMBRE") ENABLE
   )
SEGMENT CREATION IMMEDIATE
  PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
NOCOMPRESS LOGGING
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS
2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT
FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "HOPITAL_TBS"

PL/SQL procedure successfully completed.
```

REMARQUE :
La commande de création contient effectivement les informations nécessaires à la création.

```
CREATE TABLE "DBHOPITAL"."HOSPITALISATION"
("NUM_PATIENT" NUMBER(*,0),
 "CODE_SERVICE" VARCHAR2(20),
 "NUM_CHAMBRE" NUMBER(*,0),
 "LIT" NUMBER(*,0),
 "DATE_HOST" DATE,
 CONSTRAINT "PK_NUM_PATIENT_HOSPITALISATION" PRIMARY KEY("NUM_PATIENT")
  -- USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255
--STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
--PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT FLASH_CACHE
--DEFAULT CELL_FLASH_CACHE DEFAULT) TABLESPACE "HOPITAL_TBS"  ENABLE,
CONSTRAINT "FK_NUM_PATIENT_HOSPITALISATION" FOREIGN KEY ("NUM_PATIENT")
REFERENCES "DBHOPITAL"."PATIENT" ("NUM_PATIENT") ENABLE,
CONSTRAINT "FK_CODE_SERVICE_NUM_CHAMBRE" FOREIGN KEY ("CODE_SERVICE", "NUM_CHAMBRE")
REFERENCES "DBHOPITAL"."CHAMBRE" ("CODE_SERVICE", "NUM_CHAMBRE") ENABLE )
/*SEGMENT CREATION IMMEDIATE
  PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
NOCOMPRESS LOGGING
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS
2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT
FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "HOPITAL_TBS"*/.
```

11) Trouver tous les privilèges accordés à AdminHopital.

```
--11)
connect ADMINHOPITAL/pwd;
--Executer un DESC afin de retrouver les colonne nécessaire à la requête
DESC USER_TAB_PRIVS;
```

```
SQL> --11)
SQL> connect ADMINHOPITAL/pwd;
Connected.
SQL>
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)
```

REMARQUE :
L'attribut **GRANTEE** représente le bénéficiaire du privilège, donc chercher les privilèges dont **'AdminHopital'** est bénéficiaire revient à faire un **SELECT** sur **USER_TAB_PRIVS** avec pour filtre **GRANTEE = 'ADMINHOPITAL'**.

```
SELECT *
FROM USER_TAB_PRIVS
WHERE  GRANTEE = 'ADMINHOPITAL';
```

```
SQL> SELECT *
  2  FROM USER_TAB_PRIVS
  3  WHERE  GRANTEE = 'ADMINHOPITAL';

GRANTEE                        OWNER
------------------------------ ------------------------------
TABLE_NAME                     GRANTOR
------------------------------ ------------------------------
PRIVILEGE                                GRA HIE
---------------------------------------- --- ---
ADMINHOPITAL                   DBHOPITAL
EMPLOYE                        DBHOPITAL
INDEX                                    NO  NO

ADMINHOPITAL                   DBHOPITAL
EMPLOYE                        DBHOPITAL
UPDATE                                   NO  NO

GRANTEE                        OWNER
------------------------------ ------------------------------
TABLE_NAME                     GRANTOR
------------------------------ ------------------------------
PRIVILEGE                                GRA HIE
---------------------------------------- --- ---
```

REMARQUE :
ADMINHOPITALE a un privilège **UPDATE** et **INDEX** sur la table **EMPLOYE** de **DBHOPITAL**.

12) Trouver les rôles donnés à l'utilisateur AdminHopital.

```
--12)
connect ADMINHOPITAL/pwd;

SELECT USERNAME , GRANTED_ROLE
FROM USER_ROLE_PRIVS;
```

```
SQL> --12)
SQL> connect ADMINHOPITAL/pwd;
Connected.
SQL>
SQL> SELECT USERNAME , GRANTED_ROLE
  2  FROM USER_ROLE_PRIVS;

USERNAME                       GRANTED_ROLE
------------------------------ ------------------------------
ADMINHOPITAL                   GESTIONNAIREPATIENT
```

REMARQUE :
ADMINHOPITALE a le rôle **GESTIONNAIREPATIENT**.

13) Trouver tous les objets appartenant à AdminHopital.

```
--13)
SELECT * FROM USER_OBJECTS ;
```

```
SQL> --13)
SQL> SELECT * FROM USER_OBJECTS ;

no rows selected
```

REMARQUE :
ADMINHOPITALE n'a créait aucun objet donc la requête ne retourne pas de ligne.

14) L'administrateur cherche le propriétaire de la table HOSPITALISATION, comment il pourra le trouver ?

```
--14)
connect SYSTEM/pwd;

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

```
SQL> --14)
SQL> connect SYSTEM/pwd;
Connected.
SQL>
SQL> SELECT OWNER FROM ALL_TABLES WHERE TABLE_NAME = 'HOSPITALISATION';

OWNER
------------------------------
DBHOPITAL
```

REMARQUE :
Propriétaire(**HOSPITALISATION**) = **DBHOPITAL**.

15) Donner la taille en Ko de la table HOSPITALISATION.

```
--15)
connect DBHOPITAL/pwd

SELECT BYTES/1000 TAILLE_DE_HOSPITALISATION
FROM USER_EXTENTS
WHERE SEGMENT_NAME = 'HOSPITALISATION';
```

```
SQL> --15)
SQL> connect DBHOPITAL/pwd
Connected.
SQL>
SQL> SELECT BYTES/1000 TAILLE_DE_HOSPITALISATION
  2  FROM USER_EXTENTS
  3  WHERE SEGMENT_NAME = 'HOSPITALISATION';

TAILLE_DE_HOSPITALISATION
-------------------------
                   65,536
```

REMARQUE :
Taille(**HOSPITALISATION**) = **65,536 Ko**.

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

REMARQUE :
Dans cette question j'affiche le contenu des catalogues avant chaque commande **DDL** du TP1 puis la commande en elle-même et enfin leurs contenu après l'exécution de la commande.

/*Etat des catalogues avant les Commande DDL*/

```
--16)
connect DBHOPITAL/pwd

--Catalogue USER_TABLES,USER_TAB_COLUMNS,USER_CONSTRAINTS AVANT LES COMMANDES
DDL*/
SELECT TABLE_NAME
FROM   USER_TABLES ;

SELECT COLUMN_NAME , TABLE_NAME
FROM USER_TAB_COLUMNS;

SELECT CONSTRAINT_NAME , CONSTRAINT_TYPE
FROM USER_CONSTRAINTS;
```

```
SQL> --16)
SQL> connect DBHOPITAL/pwd
Connected.
SQL> --Catalogue USER_TABLES ,USER_TAB_COLUMNS ,USER_CONSTRAINTS AVANT LES COMMANDES DDL*/
SQL> SELECT TABLE_NAME
  2  FROM   USER_TABLES ;

no rows selected

SQL>
SQL> SELECT COLUMN_NAME , TABLE_NAME
  2  FROM USER_TAB_COLUMNS;

no rows selected

SQL>
SQL> SELECT CONSTRAINT_NAME , CONSTRAINT_TYPE
  2  FROM USER_CONSTRAINTS;

no rows selected
```

/*Commande DDL 1*/

```
--Commande DDL 1
CREATE TABLE EMPLOYE (
      NUM_EMP           INTEGER,
      NOM_EMP           VARCHAR2(50),
      PRENOM_EMP  VARCHAR2(50),
      ADRESSE_EMP       VARCHAR2(120),
      TEL_EMP           VARCHAR2(20),
      CONSTRAINT PK_NUM_EMP PRIMARY KEY (NUM_EMP)
);
```

```
SQL> --Commande DDL 1
SQL> CREATE TABLE EMPLOYE (
  2  NUM_EMP INTEGER,
  3  NOM_EMP VARCHAR2(50),
  4  PRENOM_EMP VARCHAR2(50),
  5  ADRESSE_EMP VARCHAR2(120),
  6  TEL_EMP VARCHAR2(20),
  7  CONSTRAINT PK_NUM_EMP PRIMARY KEY (NUM_EMP)
  8  );

Table created.
```

/*Etat des catalogues après la Commande DDL1*/

```
--Catalogue USER_TABLES ,USER_TAB_COLUMNS ,USER_CONSTRAINTS APRES COMMANDE
DDL1*/
SELECT TABLE_NAME
FROM   USER_TABLES ;

SELECT COLUMN_NAME , TABLE_NAME
FROM USER_TAB_COLUMNS;

SELECT CONSTRAINT_NAME , CONSTRAINT_TYPE
FROM USER_CONSTRAINTS;
```

```
SQL> --Catalogue USER_TABLES ,USER_TAB_COLUMNS ,USER_CONSTRAINTS APRES COMMANDE DDL1*/
SQL> SELECT TABLE_NAME
  2  FROM   USER_TABLES ;

TABLE_NAME
------------------------------
EMPLOYE

SQL>
SQL> SELECT COLUMN_NAME , TABLE_NAME
  2  FROM USER_TAB_COLUMNS;

COLUMN_NAME                    TABLE_NAME
------------------------------ ------------------------------
NUM_EMP                        EMPLOYE
NOM_EMP                        EMPLOYE
PRENOM_EMP                     EMPLOYE
ADRESSE_EMP                    EMPLOYE
TEL_EMP                        EMPLOYE

SQL>
SQL> SELECT CONSTRAINT_NAME , CONSTRAINT_TYPE
  2  FROM USER_CONSTRAINTS;

CONSTRAINT_NAME                C
------------------------------ -
PK_NUM_EMP                     P
```

/* la Commande DDL 2*/

```
--Commande DDL 2
CREATE TABLE PATIENT (
      NUM_PATIENT       INTEGER,
      NOM_PATIENT            VARCHAR2(50),
      PRENOM_PATIENT    VARCHAR2(50),
      ADRESSE_PATIENT   VARCHAR2(120),
      TEL_PATIENT            VARCHAR2(20),
      MUTUELLE          VARCHAR2(10),
      CONSTRAINT PK_NUM_PATIENT PRIMARY KEY (NUM_PATIENT)
);
```

```
SQL> --Commande DDL 2
SQL> CREATE TABLE PATIENT (
  2  NUM_PATIENT INTEGER,
  3  NOM_PATIENT VARCHAR2(50),
  4  PRENOM_PATIENT VARCHAR2(50),
  5  ADRESSE_PATIENT VARCHAR2(120),
  6  TEL_PATIENT VARCHAR2(20),
  7  MUTUELLE  VARCHAR2(10),
  8  CONSTRAINT PK_NUM_PATIENT PRIMARY KEY (NUM_PATIENT)
  9  );

Table created.
```

/*Etat des catalogues après la Commande DDL2*/

```
SQL> --Catalogue USER_TABLES ,USER_TAB_COLUMNS ,USER_CONSTRAINTS APRES COMMANDE DDL 2*/
SQL> SELECT TABLE_NAME
  2  FROM   USER_TABLES ;

TABLE_NAME
------------------------------
EMPLOYE
PATIENT

SQL>
SQL> SELECT COLUMN_NAME , TABLE_NAME
  2  FROM USER_TAB_COLUMNS;

COLUMN_NAME                    TABLE_NAME
------------------------------ ------------------------------
NUM_EMP                        EMPLOYE
NOM_EMP                        EMPLOYE
PRENOM_EMP                     EMPLOYE
ADRESSE_EMP                    EMPLOYE
TEL_EMP                        EMPLOYE
NUM_PATIENT                    PATIENT
NOM_PATIENT                    PATIENT
PRENOM_PATIENT                 PATIENT
ADRESSE_PATIENT                PATIENT
TEL_PATIENT                    PATIENT
MUTUELLE                       PATIENT

11 rows selected.

SQL>
SQL> SELECT CONSTRAINT_NAME , CONSTRAINT_TYPE
  2  FROM USER_CONSTRAINTS;

CONSTRAINT_NAME                C
------------------------------ -
PK_NUM_EMP                     P
PK_NUM_PATIENT                 P
```

/* la Commande DDL3*/

```
--Commande DDL 3
CREATE TABLE MEDECIN (
      NUM_MED           INTEGER,
      SPECIALITE        VARCHAR2(30),
      CONSTRAINT PK_NUM_MED PRIMARY KEY (NUM_MED),
      CONSTRAINT FK_NUM_MED FOREIGN KEY (NUM_MED)
      REFERENCES EMPLOYE (NUM_EMP),
      CONSTRAINT CHK_SPECIALITE CHECK (SPECIALITE IN (
      'Anesthesiste','Cardiologue','Generaliste',
      'Orthopediste','Radiologue','Traumatologue','Pneumologue'))
);
```

```
SQL> --Commande DDL 3
SQL> CREATE TABLE MEDECIN (
  2  NUM_MED INTEGER,
  3  SPECIALITE  VARCHAR2(30),
  4  CONSTRAINT PK_NUM_MED PRIMARY KEY (NUM_MED),
  5  CONSTRAINT FK_NUM_MED FOREIGN KEY (NUM_MED)
  6  REFERENCES EMPLOYE (NUM_EMP),
  7  CONSTRAINT CHK_SPECIALITE CHECK (SPECIALITE IN (
  8  'Anesthesiste','Cardiologue','Generaliste',
  9  'Orthopediste','Radiologue','Traumatologue','Pneumologue'))
 10  );

Table created.
```

/*Etat des catalogues après la Commande DDL3*/

```
SQL> --Catalogue USER_TABLES ,USER_TAB_COLUMNS ,USER_CONSTRAINTS APRES COMMANDE DDL3*/
SQL> SELECT TABLE_NAME
  2  FROM   USER_TABLES ;

TABLE_NAME
------------------------------
EMPLOYE
PATIENT
MEDECIN

SQL>
SQL> SELECT COLUMN_NAME , TABLE_NAME
  2  FROM USER_TAB_COLUMNS;

COLUMN_NAME                    TABLE_NAME
------------------------------ ------------------------------
NUM_EMP                        EMPLOYE
NOM_EMP                        EMPLOYE
PRENOM_EMP                     EMPLOYE
ADRESSE_EMP                    EMPLOYE
TEL_EMP                        EMPLOYE
NUM_MED                        MEDECIN
SPECIALITE                     MEDECIN
NUM_PATIENT                    PATIENT
NOM_PATIENT                    PATIENT
PRENOM_PATIENT                 PATIENT
ADRESSE_PATIENT                PATIENT

COLUMN_NAME                    TABLE_NAME
------------------------------ ------------------------------
TEL_PATIENT                    PATIENT
MUTUELLE                       PATIENT

13 rows selected.

SQL>
SQL> SELECT CONSTRAINT_NAME , CONSTRAINT_TYPE
  2  FROM USER_CONSTRAINTS;

CONSTRAINT_NAME                C
------------------------------ -
CHK_SPECIALITE                 C
FK_NUM_MED                     R
PK_NUM_EMP                     P
PK_NUM_PATIENT                 P

PK_NUM_MED                     P
```

/*Commande DDL4*/

```
--Commande DDL 4
CREATE TABLE SOIGNE (
      NUM_PATIENT             INTEGER,
      NUM_MED                 INTEGER,
```

```
      CONSTRAINT PK_NUM_PATIENT_MED PRIMARY KEY (NUM_PATIENT,NUM_MED),
      CONSTRAINT FK_NUM_PATIENT FOREIGN KEY (NUM_PATIENT)
      REFERENCES PATIENT (NUM_PATIENT),
      CONSTRAINT FK_NUM_MED_SOIGNE FOREIGN KEY (NUM_MED)
      REFERENCES MEDECIN (NUM_MED)
);
```

```
SQL> --Commande DDL 4
SQL> CREATE TABLE SOIGNE (
  2  NUM_PATIENT INTEGER,
  3  NUM_MED INTEGER,
  4  CONSTRAINT PK_NUM_PATIENT_MED PRIMARY KEY (NUM_PATIENT,NUM_MED),
  5  CONSTRAINT FK_NUM_PATIENT FOREIGN KEY (NUM_PATIENT)
  6  REFERENCES PATIENT (NUM_PATIENT),
  7  CONSTRAINT FK_NUM_MED_SOIGNE FOREIGN KEY (NUM_MED)
  8  REFERENCES MEDECIN (NUM_MED)
  9  );

Table created.
```

/*Etat des catalogues après la Commande DDL4*/

```
SQL> --Catalogue USER_TABLES ,USER_TAB_COLUMNS ,USER_CONSTRAINTS APRES COMMANDE DDL 4*/
SQL> SELECT TABLE_NAME
  2  FROM   USER_TABLES ;

TABLE_NAME
------------------------------
EMPLOYE
PATIENT
MEDECIN
SOIGNE

SQL>
SQL> SELECT COLUMN_NAME , TABLE_NAME
  2  FROM USER_TAB_COLUMNS;

COLUMN_NAME                    TABLE_NAME
------------------------------ ------------------------------
NUM_EMP                        EMPLOYE
NOM_EMP                        EMPLOYE
PRENOM_EMP                     EMPLOYE
ADRESSE_EMP                    EMPLOYE
TEL_EMP                        EMPLOYE
NUM_MED                        MEDECIN
SPECIALITE                     MEDECIN
NUM_PATIENT                    PATIENT
NOM_PATIENT                    PATIENT
PRENOM_PATIENT                 PATIENT
ADRESSE_PATIENT                PATIENT

COLUMN_NAME                    TABLE_NAME
------------------------------ ------------------------------
TEL_PATIENT                    PATIENT
MUTUELLE                       PATIENT
NUM_PATIENT                    SOIGNE
NUM_MED                        SOIGNE

15 rows selected.

SQL> SELECT CONSTRAINT_NAME , CONSTRAINT_TYPE
  2  FROM USER_CONSTRAINTS;

CONSTRAINT_NAME                C
------------------------------ -
CHK_SPECIALITE                 C
FK_NUM_MED_SOIGNE              R
FK_NUM_PATIENT                 R
FK_NUM_MED                     R
PK_NUM_EMP                     P
PK_NUM_PATIENT                 P
PK_NUM_MED                     P
PK_NUM_PATIENT_MED             P

8 rows selected.
```

/*Commande DDL5*/

```
--Commande DDL 5
CREATE TABLE SERVICE (
      CODE_SERVICE      VARCHAR2(20),
      NOM_SERVICE       VARCHAR2(50),
      BATIMENT          VARCHAR2(10),
      DIRECTEUR         INTEGER,
      CONSTRAINT PK_CODE_SERVICE PRIMARY KEY (CODE_SERVICE),
      CONSTRAINT FK_DIRECTEUR FOREIGN KEY (DIRECTEUR)
      REFERENCES MEDECIN (NUM_MED),
      CONSTRAINT UNQ_NOM_SERVICE UNIQUE (NOM_SERVICE)
```

```
);
```

```
SQL> --Commande DDL 5
SQL> CREATE TABLE SERVICE (
  2  CODE_SERVICE VARCHAR2(20),
  3  NOM_SERVICE  VARCHAR2(50),
  4  BATIMENT VARCHAR2(10),
  5  DIRECTEUR  INTEGER,
  6  CONSTRAINT PK_CODE_SERVICE PRIMARY KEY (CODE_SERVICE),
  7  CONSTRAINT FK_DIRECTEUR FOREIGN KEY (DIRECTEUR)
  8  REFERENCES MEDECIN (NUM_MED),
  9  CONSTRAINT UNQ_NOM_SERVICE UNIQUE (NOM_SERVICE)
 10  );

Table created.
```

/*Etat des catalogues après la Commande DDL5*/

```
SQL> --Catalogue USER_TABLES ,USER_TAB_COLUMNS ,USER_CONSTRAINTS APRES COMMANDE DDL*/
SQL> SELECT TABLE_NAME
  2  FROM   USER_TABLES ;

TABLE_NAME
------------------------------
EMPLOYE
PATIENT
MEDECIN
SOIGNE
SERVICE

SQL>
SQL> SELECT COLUMN_NAME , TABLE_NAME
  2  FROM USER_TAB_COLUMNS;

COLUMN_NAME                    TABLE_NAME
------------------------------ ------------------------------
NUM_EMP                        EMPLOYE
NOM_EMP                        EMPLOYE
PRENOM_EMP                     EMPLOYE
ADRESSE_EMP                    EMPLOYE
TEL_EMP                        EMPLOYE
NUM_MED                        MEDECIN
SPECIALITE                     MEDECIN
NUM_PATIENT                    PATIENT
NOM_PATIENT                    PATIENT
PRENOM_PATIENT                 PATIENT
ADRESSE_PATIENT                PATIENT

COLUMN_NAME                    TABLE_NAME
------------------------------ ------------------------------
TEL_PATIENT                    PATIENT
MUTUELLE                       PATIENT
CODE_SERVICE                   SERVICE
NOM_SERVICE                    SERVICE
BATIMENT                       SERVICE
DIRECTEUR                      SERVICE
NUM_PATIENT                    SOIGNE
NUM_MED                        SOIGNE

19 rows selected.

SQL> SELECT CONSTRAINT_NAME , CONSTRAINT_TYPE
  2  FROM USER_CONSTRAINTS;

CONSTRAINT_NAME                C
------------------------------ -
CHK_SPECIALITE                 C
FK_NUM_MED_SOIGNE              R
FK_DIRECTEUR                   R
FK_NUM_PATIENT                 R
FK_NUM_MED                     R
PK_NUM_EMP                     P
PK_NUM_PATIENT                 P
PK_NUM_MED                     P
PK_NUM_PATIENT_MED             P
PK_CODE_SERVICE                P
UNQ_NOM_SERVICE                U

11 rows selected.
```

/*Commande DDL6*/

```
--Commande DDL 6
CREATE TABLE INFIRMIER (
      NUM_INF                 INTEGER,
      CODE_SERVICE      VARCHAR2(20),
      ROTATION          VARCHAR2(15),
      SALAIRE           FLOAT,
      CONSTRAINT PK_NUM_INF PRIMARY KEY (NUM_INF),
      CONSTRAINT FK_NUM_INF FOREIGN KEY (NUM_INF)
      REFERENCES EMPLOYE (NUM_EMP),
      CONSTRAINT FK_CODE_SERVICE FOREIGN KEY (CODE_SERVICE)
      REFERENCES SERVICE (CODE_SERVICE),
```

```
        CONSTRAINT CHK_ROTATION CHECK (ROTATION IN ('JOUR','NUIT'))
);
```

```
SQL> --Commande DDL 6
SQL> CREATE TABLE INFIRMIER (
  2  NUM_INF INTEGER,
  3  CODE_SERVICE  VARCHAR2(20),
  4  ROTATION VARCHAR2(15),
  5  SALAIRE  FLOAT,
  6  CONSTRAINT PK_NUM_INF PRIMARY KEY (NUM_INF),
  7  CONSTRAINT FK_NUM_INF FOREIGN KEY (NUM_INF)
  8  REFERENCES EMPLOYE (NUM_EMP),
  9  CONSTRAINT FK_CODE_SERVICE FOREIGN KEY (CODE_SERVICE)
 10  REFERENCES SERVICE (CODE_SERVICE),
 11  CONSTRAINT CHK_ROTATION CHECK (ROTATION IN ('JOUR','NUIT'))
 12  );

Table created.
```

/*Etat des catalogues après la Commande DDL6*/

```
SQL> --Catalogue USER_TABLES ,USER_TAB_COLUMNS ,USER_CONSTRAINTS APRES COMMANDE DDL 6*/
SQL> SELECT TABLE_NAME
  2  FROM   USER_TABLES ;

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

6 rows selected.

SQL> SELECT COLUMN_NAME , TABLE_NAME
  2  FROM USER_TAB_COLUMNS;

COLUMN_NAME                    TABLE_NAME
------------------------------ ------------------------------
NUM_EMP                        EMPLOYE
NOM_EMP                        EMPLOYE
PRENOM_EMP                     EMPLOYE
ADRESSE_EMP                    EMPLOYE
TEL_EMP                        EMPLOYE
NUM_INF                        INFIRMIER
CODE_SERVICE                   INFIRMIER
ROTATION                       INFIRMIER
SALAIRE                        INFIRMIER
NUM_MED                        MEDECIN
SPECIALITE                     MEDECIN

COLUMN_NAME                    TABLE_NAME
------------------------------ ------------------------------
NUM_PATIENT                    PATIENT
NOM_PATIENT                    PATIENT
PRENOM_PATIENT                 PATIENT
ADRESSE_PATIENT                PATIENT
TEL_PATIENT                    PATIENT
MUTUELLE                       PATIENT
CODE_SERVICE                   SERVICE
NOM_SERVICE                    SERVICE
BATIMENT                       SERVICE
DIRECTEUR                      SERVICE
NUM_PATIENT                    SOIGNE

COLUMN_NAME                    TABLE_NAME
------------------------------ ------------------------------
NUM_MED                        SOIGNE

23 rows selected.

SQL> SELECT CONSTRAINT_NAME , CONSTRAINT_TYPE
  2  FROM USER_CONSTRAINTS;

CONSTRAINT_NAME                C
------------------------------ -
CHK_SPECIALITE                 C
CHK_ROTATION                   C
FK_CODE_SERVICE                R
FK_NUM_MED_SOIGNE              R
FK_DIRECTEUR                   R
FK_NUM_PATIENT                 R
FK_NUM_MED                     R
FK_NUM_INF                     R
PK_NUM_EMP                     P
PK_NUM_PATIENT                 P
PK_NUM_MED                     P

CONSTRAINT_NAME                C
------------------------------ -
PK_NUM_PATIENT_MED             P
PK_CODE_SERVICE                P
UNQ_NOM_SERVICE                U
PK_NUM_INF                     P

15 rows selected.
```

/*Commande DDL7*/

```
--Commande DDL 7
CREATE TABLE CHAMBRE (
      CODE_SERVICE      VARCHAR2(20),
      NUM_CHAMBRE       INTEGER,
      SURVEILLANT             INTEGER,
      NB_LITS                 INTEGER,
      CONSTRAINT PK_CODE_SERVIC_NUM_CHAMBRE PRIMARY KEY
(CODE_SERVICE,NUM_CHAMBRE),
      CONSTRAINT FK_CODE_SERVICE_CHAMBRE FOREIGN KEY (CODE_SERVICE)
      REFERENCES SERVICE (CODE_SERVICE),
      CONSTRAINT FK_SURVEILLANT FOREIGN KEY (SURVEILLANT)
      REFERENCES INFIRMIER (NUM_INF)
);
```

```
SQL> --Commande DDL 7
SQL> CREATE TABLE CHAMBRE (
  2  CODE_SERVICE  VARCHAR2(20),
  3  NUM_CHAMBRE  INTEGER,
  4  SURVEILLANT INTEGER,
  5  NB_LITS INTEGER,
  6  CONSTRAINT PK_CODE_SERVIC_NUM_CHAMBRE PRIMARY KEY (CODE_SERVICE,NUM_CHAMBRE),
  7  CONSTRAINT FK_CODE_SERVICE_CHAMBRE FOREIGN KEY (CODE_SERVICE)
  8  REFERENCES SERVICE (CODE_SERVICE),
  9  CONSTRAINT FK_SURVEILLANT FOREIGN KEY (SURVEILLANT)
 10  REFERENCES INFIRMIER (NUM_INF)
 11  );

Table created.
```

/*Etat des catalogues après la Commande DDL7*/

```
SQL> SELECT TABLE_NAME
  2  FROM   USER_TABLES ;

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

7 rows selected.
```

```
SQL> SELECT COLUMN_NAME , TABLE_NAME
  2  FROM USER_TAB_COLUMNS;

COLUMN_NAME                    TABLE_NAME
------------------------------ ------------------------------
CODE_SERVICE                   CHAMBRE
NUM_CHAMBRE                    CHAMBRE
SURVEILLANT                    CHAMBRE
NB_LITS                        CHAMBRE
NUM_EMP                        EMPLOYE
NOM_EMP                        EMPLOYE
PRENOM_EMP                     EMPLOYE
ADRESSE_EMP                    EMPLOYE
TEL_EMP                        EMPLOYE
NUM_INF                        INFIRMIER
CODE_SERVICE                   INFIRMIER

COLUMN_NAME                    TABLE_NAME
------------------------------ ------------------------------
ROTATION                       INFIRMIER
SALAIRE                        INFIRMIER
NUM_MED                        MEDECIN
SPECIALITE                     MEDECIN
NUM_PATIENT                    PATIENT
NOM_PATIENT                    PATIENT
PRENOM_PATIENT                 PATIENT
ADRESSE_PATIENT                PATIENT
TEL_PATIENT                    PATIENT
MUTUELLE                       PATIENT
CODE_SERVICE                   SERVICE

COLUMN_NAME                    TABLE_NAME
------------------------------ ------------------------------
NOM_SERVICE                    SERVICE
BATIMENT                       SERVICE
DIRECTEUR                      SERVICE
NUM_PATIENT                    SOIGNE
NUM_MED                        SOIGNE

27 rows selected.

SQL> SELECT CONSTRAINT_NAME , CONSTRAINT_TYPE
  2  FROM USER_CONSTRAINTS;

CONSTRAINT_NAME                C
------------------------------ -
CHK_SPECIALITE                 C
CHK_ROTATION                   C
FK_SURVEILLANT                 R
FK_CODE_SERVICE                R
FK_CODE_SERVICE_CHAMBRE        R
FK_NUM_MED_SOIGNE              R
FK_DIRECTEUR                   R
FK_NUM_PATIENT                 R
FK_NUM_MED                     R
FK_NUM_INF                     R
PK_NUM_EMP                     P

CONSTRAINT_NAME                C
------------------------------ -
PK_NUM_PATIENT                 P
PK_NUM_MED                     P
PK_NUM_PATIENT_MED             P
PK_CODE_SERVICE                P
UNQ_NOM_SERVICE                U
PK_NUM_INF                     P
PK_CODE_SERVIC_NUM_CHAMBRE     P

18 rows selected.
```

/*Etat des catalogues après la Commande DDL8*/

```
--Commande DDL 8
CREATE TABLE HOSPITALISATION (
      NUM_PATIENT       INTEGER,
      CODE_SERVICE      VARCHAR2(20),
      NUM_CHAMBRE       INTEGER,
      LIT                     INTEGER,
      CONSTRAINT PK_NUM_PATIENT_HOSPITALISATION PRIMARY KEY (NUM_PATIENT),
      CONSTRAINT FK_NUM_PATIENT_HOSPITALISATION FOREIGN KEY (NUM_PATIENT)
      REFERENCES PATIENT (NUM_PATIENT),
      CONSTRAINT FK_CODE_SERVICE_NUM_CHAMBRE FOREIGN KEY
(CODE_SERVICE,NUM_CHAMBRE)
      REFERENCES CHAMBRE (CODE_SERVICE,NUM_CHAMBRE)
);
```

```
SQL> CREATE TABLE HOSPITALISATION (
  2  NUM_PATIENT  INTEGER,
  3  CODE_SERVICE  VARCHAR2(20),
  4  NUM_CHAMBRE  INTEGER,
  5  LIT INTEGER,
  6  CONSTRAINT PK_NUM_PATIENT_HOSPITALISATION PRIMARY KEY (NUM_PATIENT),
  7  CONSTRAINT FK_NUM_PATIENT_HOSPITALISATION FOREIGN KEY (NUM_PATIENT)
  8  REFERENCES PATIENT (NUM_PATIENT),
  9  CONSTRAINT FK_CODE_SERVICE_NUM_CHAMBRE FOREIGN KEY (CODE_SERVICE,NUM_CHAMBRE)
 10  REFERENCES CHAMBRE (CODE_SERVICE,NUM_CHAMBRE)
 11  );

Table created.
```

/*Etat des catalogues après la Commande DDL8*/

```
SQL> SELECT TABLE_NAME
  2  FROM   USER_TABLES ;

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

8 rows selected.

SQL> SELECT COLUMN_NAME , TABLE_NAME
  2  FROM USER_TAB_COLUMNS;

COLUMN_NAME                    TABLE_NAME
------------------------------ ------------------------------
CODE_SERVICE                   CHAMBRE
NUM_CHAMBRE                    CHAMBRE
SURVEILLANT                    CHAMBRE
NB_LITS                        CHAMBRE
NUM_EMP                        EMPLOYE
NOM_EMP                        EMPLOYE
PRENOM_EMP                     EMPLOYE
ADRESSE_EMP                    EMPLOYE
TEL_EMP                        EMPLOYE
NUM_PATIENT                    HOSPITALISATION
CODE_SERVICE                   HOSPITALISATION

COLUMN_NAME                    TABLE_NAME
------------------------------ ------------------------------
NUM_CHAMBRE                    HOSPITALISATION
LIT                            HOSPITALISATION
NUM_INF                        INFIRMIER
CODE_SERVICE                   INFIRMIER
ROTATION                       INFIRMIER
SALAIRE                        INFIRMIER
NUM_MED                        MEDECIN
SPECIALITE                     MEDECIN
NUM_PATIENT                    PATIENT
NOM_PATIENT                    PATIENT
PRENOM_PATIENT                 PATIENT

COLUMN_NAME                    TABLE_NAME
------------------------------ ------------------------------
ADRESSE_PATIENT                PATIENT
TEL_PATIENT                    PATIENT
MUTUELLE                       PATIENT
CODE_SERVICE                   SERVICE
NOM_SERVICE                    SERVICE
BATIMENT                       SERVICE
DIRECTEUR                      SERVICE
NUM_PATIENT                    SOIGNE
NUM_MED                        SOIGNE

31 rows selected.

SQL> SELECT CONSTRAINT_NAME , CONSTRAINT_TYPE
  2  FROM USER_CONSTRAINTS;

CONSTRAINT_NAME                C
------------------------------ -
CHK_SPECIALITE                 C
CHK_ROTATION                   C
FK_CODE_SERVICE_NUM_CHAMBRE    R
FK_SURVEILLANT                 R
FK_CODE_SERVICE                R
FK_CODE_SERVICE_CHAMBRE        R
FK_NUM_MED_SOIGNE              R
FK_DIRECTEUR                   R
FK_NUM_PATIENT                 R
FK_NUM_PATIENT_HOSPITALISATION R
FK_NUM_MED                     R

CONSTRAINT_NAME                C
------------------------------ -
FK_NUM_INF                     R
PK_NUM_EMP                     P
PK_NUM_PATIENT                 P
PK_NUM_MED                     P
PK_NUM_PATIENT_MED             P
PK_CODE_SERVICE                P
UNQ_NOM_SERVICE                U
PK_NUM_INF                     P
PK_CODE_SERVIC_NUM_CHAMBRE     P
PK_NUM_PATIENT_HOSPITALISATION P

21 rows selected.
```

/*Etat des catalogues avant la Commande DDL9*/

```
--------------------------------------------------------------------------------
--Catalogue USER_TAB_COLUMNS  AVANT COMMANDE DDL 9*/

SELECT COLUMN_NAME , TABLE_NAME
FROM USER_TAB_COLUMNS
WHERE  TABLE_NAME = 'HOSPITALISATION';
```

```
SQL>
SQL> SELECT COLUMN_NAME , TABLE_NAME
  2  FROM USER_TAB_COLUMNS
  3  WHERE  TABLE_NAME = 'HOSPITALISATION';

COLUMN_NAME                    TABLE_NAME
------------------------------ ------------------------------
NUM_PATIENT                    HOSPITALISATION
CODE_SERVICE                   HOSPITALISATION
NUM_CHAMBRE                    HOSPITALISATION
LIT                            HOSPITALISATION
```

/*Commande DDL9*/

```
--Commande DDL 9
ALTER TABLE HOSPITALISATION
ADD DATE_HOST DATE ;
```

```
SQL> ALTER TABLE HOSPITALISATION
  2  ADD DATE_HOST DATE ;

Table altered.
```

/*Etat des catalogues après la Commande DDL9*/

```
--Catalogue USER_TAB_COLUMNS  APRES COMMANDE DDL 9*/
SELECT COLUMN_NAME , TABLE_NAME
FROM USER_TAB_COLUMNS
WHERE  TABLE_NAME = 'HOSPITALISATION';
```

```
SQL> SELECT COLUMN_NAME , TABLE_NAME
  2  FROM USER_TAB_COLUMNS
  3  WHERE  TABLE_NAME = 'HOSPITALISATION';

COLUMN_NAME                    TABLE_NAME
------------------------------ ------------------------------
NUM_PATIENT                    HOSPITALISATION
CODE_SERVICE                   HOSPITALISATION
NUM_CHAMBRE                    HOSPITALISATION
LIT                            HOSPITALISATION
DATE_HOST                      HOSPITALISATION
```

/*Etat des catalogues avant la Commande DDL10*/

```
--Catalogue USER_TABLES ,USER_TAB_COLUMNS ,USER_CONSTRAINTS AVANT LA COMMANDE
DDL 10*/

SELECT CONSTRAINT_NAME , CONSTRAINT_TYPE
FROM USER_CONSTRAINTS
WHERE TABLE_NAME = 'PATIENT';
```

```
SQL>
SQL> SELECT CONSTRAINT_NAME , CONSTRAINT_TYPE
  2  FROM USER_CONSTRAINTS
  3  WHERE TABLE_NAME = 'PATIENT';

CONSTRAINT_NAME                C
------------------------------ -
PK_NUM_PATIENT                 P
```

/*Commande DDL10*/

```
--Commande DDL 10
ALTER TABLE PATIENT
ADD CONSTRAINT NT_NLL_MUTUELLE CHECK (MUTUELLE IS NOT NULL);
```

```
SQL> ALTER TABLE PATIENT
  2  ADD CONSTRAINT NT_NLL_MUTUELLE CHECK (MUTUELLE IS NOT NULL);

Table altered.
```

/*Etat des catalogues après la Commande DDL10*/

```
--Catalogue USER_CONSTRAINTS APRES COMMANDE DDL 10
SELECT CONSTRAINT_NAME , CONSTRAINT_TYPE
FROM USER_CONSTRAINTS
WHERE TABLE_NAME = 'PATIENT';
```

```
SQL> SELECT CONSTRAINT_NAME , CONSTRAINT_TYPE
  2  FROM USER_CONSTRAINTS
  3  WHERE TABLE_NAME = 'PATIENT';

CONSTRAINT_NAME                C
------------------------------ -
PK_NUM_PATIENT                 P
NT_NLL_MUTUELLE                C
```

/*Etat des catalogues avant la Commande DDL11*/

```
--------------------------------------------------------------------------------
--Catalogue USER_TAB_COLUMNS  AVANT LA COMMANDE DDL 11*/
SELECT COLUMN_NAME ,DATA_TYPE,DATA_LENGTH, TABLE_NAME
FROM USER_TAB_COLUMNS
WHERE TABLE_NAME = 'PATIENT'
AND COLUMN_NAME = 'PRENOM_PATIENT';
```

```
SQL> SELECT COLUMN_NAME ,DATA_TYPE,DATA_LENGTH, TABLE_NAME
  2  FROM USER_TAB_COLUMNS
  3  WHERE TABLE_NAME = 'PATIENT'
  4  AND COLUMN_NAME = 'PRENOM_PATIENT';

COLUMN_NAME
------------------------------
DATA_TYPE
--------------------------------------------------------------------------------
DATA_LENGTH TABLE_NAME
----------- ------------------------------
PRENOM_PATIENT
VARCHAR2
         50 PATIENT
```

/*Commande DDL11*/

```
--Commande DDL 11
ALTER TABLE PATIENT
MODIFY PRENOM_PATIENT VARCHAR2(100);
```

```
SQL> ALTER TABLE PATIENT
  2  MODIFY PRENOM_PATIENT VARCHAR2(100);

Table altered.
```

/*Etat des catalogues après la Commande DDL11*/

```
--Catalogue USER_TAB_COLUMNS  APRES LA COMMANDE DDL 11*/
SELECT COLUMN_NAME ,DATA_TYPE,DATA_LENGTH, TABLE_NAME
FROM USER_TAB_COLUMNS
WHERE TABLE_NAME = 'PATIENT'
AND COLUMN_NAME = 'PRENOM_PATIENT';
```

```
SQL> SELECT COLUMN_NAME ,DATA_TYPE,DATA_LENGTH, TABLE_NAME
  2  FROM USER_TAB_COLUMNS
  3  WHERE TABLE_NAME = 'PATIENT'
  4  AND COLUMN_NAME = 'PRENOM_PATIENT';

COLUMN_NAME
------------------------------
DATA_TYPE
--------------------------------------------------------------------------------
DATA_LENGTH TABLE_NAME
----------- ------------------------------
PRENOM_PATIENT
VARCHAR2
        100 PATIENT
```

/*Etat des catalogues avant la Commande DDL12*/

```
--------------------------------------------------------------------------------
--Catalogue USER_TAB_COLUMNS AVANT LA COMMANDE DDL 12*/

SELECT COLUMN_NAME , TABLE_NAME
FROM USER_TAB_COLUMNS
WHERE TABLE_NAME = 'EMPLOYE';
```

```
SQL>
SQL> SELECT COLUMN_NAME , TABLE_NAME
  2  FROM USER_TAB_COLUMNS
  3  WHERE TABLE_NAME = 'EMPLOYE';

COLUMN_NAME                    TABLE_NAME
------------------------------ ------------------------------
NUM_EMP                        EMPLOYE
NOM_EMP                        EMPLOYE
PRENOM_EMP                     EMPLOYE
ADRESSE_EMP                    EMPLOYE
TEL_EMP                        EMPLOYE
```

/* Commande DDL12*/

```
--Commande DDL 12
ALTER TABLE EMPLOYE
DROP COLUMN TEL_EMP;
```

```
SQL> ALTER TABLE EMPLOYE
  2  DROP COLUMN TEL_EMP;

Table altered.
```

/*Etat des catalogues après la Commande DDL12*/

```
--Catalogue USER_TAB_COLUMNS  APRES COMMANDE DDL 12*/
SELECT COLUMN_NAME , TABLE_NAME
FROM USER_TAB_COLUMNS
WHERE TABLE_NAME = 'EMPLOYE';
```

```
SQL> SELECT COLUMN_NAME , TABLE_NAME
  2  FROM USER_TAB_COLUMNS
  3  WHERE TABLE_NAME = 'EMPLOYE';

COLUMN_NAME                    TABLE_NAME
------------------------------ ------------------------------
NUM_EMP                        EMPLOYE
NOM_EMP                        EMPLOYE
PRENOM_EMP                     EMPLOYE
ADRESSE_EMP                    EMPLOYE
```

/*Etat des catalogues avant la Commande DDL13*/

```
--------------------------------------------------------------------------------
----------
--Catalogue USER_TAB_COLUMNS  AVANT LA COMMANDE DDL 13*/
SELECT COLUMN_NAME , TABLE_NAME
FROM USER_TAB_COLUMNS
WHERE TABLE_NAME = 'PATIENT';
```

```
SQL> SELECT COLUMN_NAME , TABLE_NAME
  2  FROM USER_TAB_COLUMNS
  3  WHERE TABLE_NAME = 'PATIENT';

COLUMN_NAME                    TABLE_NAME
------------------------------ ------------------------------
NUM_PATIENT                    PATIENT
NOM_PATIENT                    PATIENT
PRENOM_PATIENT                 PATIENT
ADRESSE_PATIENT                PATIENT
TEL_PATIENT                    PATIENT
MUTUELLE                       PATIENT

6 rows selected.
```

/* Commande DDL13*/

```
--Commande DDL 13
ALTER TABLE PATIENT
RENAME COLUMN ADRESSE_PATIENT
TO adr_pat;
```

```
SQL> ALTER TABLE PATIENT
  2  RENAME COLUMN ADRESSE_PATIENT
  3  TO adr_pat;

Table altered.
```

/*Etat des catalogues après la Commande DDL13*/

```
--Catalogue USER_TAB_COLUMNS  APRES COMMANDE DDL 13*/
SELECT COLUMN_NAME , TABLE_NAME
FROM USER_TAB_COLUMNS
WHERE TABLE_NAME = 'PATIENT';
```

```
SQL> SELECT COLUMN_NAME , TABLE_NAME
  2  FROM USER_TAB_COLUMNS
  3  WHERE TABLE_NAME = 'PATIENT';

COLUMN_NAME                    TABLE_NAME
------------------------------ ------------------------------
NUM_PATIENT                    PATIENT
NOM_PATIENT                    PATIENT
PRENOM_PATIENT                 PATIENT
ADR_PAT                        PATIENT
TEL_PATIENT                    PATIENT
MUTUELLE                       PATIENT

6 rows selected.
```

/*Etat des catalogues avant la Commande DDL14*/

```
--------------------------------------------------------------------------------
 --Catalogue USER_CONSTRAINTS AVANT LA COMMANDE DDL 14*/
SELECT CONSTRAINT_NAME , CONSTRAINT_TYPE
FROM USER_CONSTRAINTS
WHERE TABLE_NAME = 'INFIRMIER';
```

```
SQL> SELECT CONSTRAINT_NAME , CONSTRAINT_TYPE
  2  FROM USER_CONSTRAINTS
  3  WHERE TABLE_NAME = 'INFIRMIER';

CONSTRAINT_NAME                C
------------------------------ -
CHK_ROTATION                   C
PK_NUM_INF                     P
FK_NUM_INF                     R
FK_CODE_SERVICE                R
```

/*Commande DDL14*/

```
--Commande DDL 13
ALTER TABLE INFIRMIER
ADD CONSTRAINT CHK_SALAIRE CHECK (SALAIRE BETWEEN 10000 AND 30000);
```

```
SQL> ALTER TABLE INFIRMIER
  2  ADD CONSTRAINT CHK_SALAIRE CHECK (SALAIRE BETWEEN 10000 AND 30000);

Table altered.
```

/*Etat des catalogues après la Commande DDL14*/

```
--Catalogue USER_CONSTRAINTS APRES COMMANDE DDL 14*/
SELECT CONSTRAINT_NAME , CONSTRAINT_TYPE
FROM USER_CONSTRAINTS
WHERE TABLE_NAME = 'INFIRMIER';
```

```
SQL> SELECT CONSTRAINT_NAME , CONSTRAINT_TYPE
  2  FROM USER_CONSTRAINTS
  3  WHERE TABLE_NAME = 'INFIRMIER';

CONSTRAINT_NAME                C
------------------------------ -
CHK_ROTATION                   C
PK_NUM_INF                     P
FK_NUM_INF                     R
FK_CODE_SERVICE                R
CHK_SALAIRE                    C
```

/*Etat des catalogues avant la Commande DDL15*/

```
--------------------------------------------------------------------------------
 --Catalogue USER_CONSTRAINTS AVANT LA COMMANDE DDL 15*/
SELECT CONSTRAINT_NAME , CONSTRAINT_TYPE
FROM USER_CONSTRAINTS
WHERE TABLE_NAME = 'MEDECIN';
```

```
SQL> SELECT CONSTRAINT_NAME , CONSTRAINT_TYPE
  2  FROM USER_CONSTRAINTS
  3  WHERE TABLE_NAME = 'MEDECIN';

CONSTRAINT_NAME                C
------------------------------ -
CHK_SPECIALITE                 C
PK_NUM_MED                     P
FK_NUM_MED                     R
```

/*Commande DDL15*/

```
--Commande DDL 14
ALTER TABLE MEDECIN
ADD CONSTRAINT NT_NLL_SPECIALITE CHECK (SPECIALITE IS NOT NULL);
```

```
SQL> ALTER TABLE MEDECIN
  2  ADD CONSTRAINT NT_NLL_SPECIALITE CHECK (SPECIALITE IS NOT NULL);

Table altered.
```

/*Etat des catalogues après la Commande DDL15*/

```
--Catalogue USER_CONSTRAINTS APRES COMMANDE DDL 15*/
SELECT CONSTRAINT_NAME , CONSTRAINT_TYPE
FROM USER_CONSTRAINTS
WHERE TABLE_NAME = 'MEDECIN';
```

```
SQL> SELECT CONSTRAINT_NAME , CONSTRAINT_TYPE
  2  FROM USER_CONSTRAINTS
  3  WHERE TABLE_NAME = 'MEDECIN';

CONSTRAINT_NAME               C
----------------------------- -
CHK_SPECIALITE                C
PK_NUM_MED                    P
FK_NUM_MED                    R
NT_NLL_SPECIALITE             C
```

/*Etat des catalogues avant la Commande DDL16*/

```
--------------------------------------------------------------------------------
 --Catalogue USER_CONSTRAINTS AVANT LA COMMANDE DDL 16*/
SELECT CONSTRAINT_NAME , CONSTRAINT_TYPE , STATUS
FROM USER_CONSTRAINTS
WHERE TABLE_NAME = 'INFIRMIER'
AND CONSTRAINT_NAME = 'CHK_SALAIRE';
```

```
SQL> SELECT CONSTRAINT_NAME , CONSTRAINT_TYPE , STATUS
  2  FROM USER_CONSTRAINTS
  3  WHERE TABLE_NAME = 'INFIRMIER'
  4  AND CONSTRAINT_NAME = 'CHK_SALAIRE';

CONSTRAINT_NAME               C STATUS
----------------------------- - --------
CHK_SALAIRE                   C ENABLED
```

/*Commande DDL16*/

```
--Commande DDL 16
ALTER TABLE INFIRMIER
DISABLE CONSTRAINT CHK_SALAIRE;
```

```
SQL> ALTER TABLE INFIRMIER
  2  DISABLE CONSTRAINT CHK_SALAIRE;

Table altered.
```

/*Etat des catalogues après la Commande DDL16*/

```
--Catalogue USER_CONSTRAINTS APRES COMMANDE DDL 16*/
SELECT CONSTRAINT_NAME , CONSTRAINT_TYPE , STATUS
FROM USER_CONSTRAINTS
WHERE TABLE_NAME = 'INFIRMIER'
AND CONSTRAINT_NAME = 'CHK_SALAIRE';
```

```
SQL> SELECT CONSTRAINT_NAME , CONSTRAINT_TYPE , STATUS
  2  FROM USER_CONSTRAINTS
  3  WHERE TABLE_NAME = 'INFIRMIER'
  4  AND CONSTRAINT_NAME = 'CHK_SALAIRE';

CONSTRAINT_NAME               C STATUS
----------------------------- - --------
CHK_SALAIRE                   C DISABLED
```

/*Etat des catalogues avant la Commande DDL17*/

```
--Commande DDL 17
ALTER TABLE INFIRMIER
ENABLE CONSTRAINT CHK_SALAIRE;
```

```
SQL> ALTER TABLE INFIRMIER
  2  ENABLE CONSTRAINT CHK_SALAIRE;

Table altered.
```

```
--Catalogue USER_CONSTRAINTS APRES COMMANDE DDL 17*/
SELECT CONSTRAINT_NAME , CONSTRAINT_TYPE , STATUS
FROM USER_CONSTRAINTS
WHERE TABLE_NAME = 'INFIRMIER'
AND CONSTRAINT_NAME = 'CHK_SALAIRE';
```

```
SQL> SELECT CONSTRAINT_NAME , CONSTRAINT_TYPE , STATUS
  2  FROM USER_CONSTRAINTS
  3  WHERE TABLE_NAME = 'INFIRMIER'
  4  AND CONSTRAINT_NAME = 'CHK_SALAIRE';

CONSTRAINT_NAME                C S STATUS
------------------------------ - --------
CHK_SALAIRE                    C ENABLED
```