



جامعة هواري بومدين
Université Houari Boumedienne

Rapport BDD TP3

NOM : YENNEK

Prénom : Kamel

Matricule : 191931066409

Groupe : 02

1).Connecter en tant que « System ». Lister le catalogue « DICT ». Il contient combien d’instances ? Donner sa structure ?

- Select * from dict ;

```
SQL> connect
Enter user-name: SYSTEM
Enter password:
Connected.
SQL> select * from dict;
```

```
GV$XSTREAM_OUTBOUND_SERVER
Synonym for GV_$XSTREAM_OUTBOUND_SERVER

GV$XSTREAM_TRANSACTION
Synonym for GV_$XSTREAM_TRANSACTION

TABLE_NAME
-----
COMMENTS
-----
GV$_LOCK
Synonym for GV_$LOCK

IND
Synonym for USER_INDEXES

ALL_JOBS
Synonym for USER_JOBS

2551 rows selected.
```

-Nombre d’instances:

Select count(*) from dict ;

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

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

-Sa structure :

Desc dict ;

```
SQL> Desc dict ;

Name                                         Null?    Type
-----
TABLE_NAME                                  VARCHA2(30)
COMMENTS                                    VARCHA2(4000)
```

2) Donner le rôle et la structure des tables (ou vues) suivantes ALL TAB COLUMNS, USER USERS, ALL CONSTRAINTS et USER TAB PRIVS. (exemple : describe USER USERS; select COMMENTS from DICT where TABLE NAME='USER USERS';)

select COMMENTS from DICT where TABLE_NAME='ALL_TAB_COLUMNS';

```
SQL> select COMMENTS from DICT where TABLE_NAME='ALL_TAB_COLUMNS';
```

```
COMMENTS
```

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

Desc ALL_TAB_COLUMNS;

```
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)

select COMMENTS from DICT where TABLE_NAME='USER_USERS';

```
SQL> select COMMENTS from DICT where TABLE_NAME='USER_USERS';
```

```
COMMENTS
```

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

desc USER_USERS;

```
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)

select COMMENTS from DICT where TABLE_NAME='ALL_CONSTRAINTS';

```
SQL> select COMMENTS from DICT where TABLE_NAME='ALL_CONSTRAINTS';
```

COMMENTS

Constraint definitions on accessible tables

desc ALL_CONSTRAINTS;

```
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)

select COMMENTS from DICT where TABLE_NAME='USER_TAB_PRIVS';

```
SQL> select COMMENTS from DICT where TABLE_NAME='USER_TAB_PRIVS';
```

```
COMMENTS
```

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

```
desc USER_TAB_PRIVS;
```

```
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 vous êtes connecté (sans utiliser show user, en utilisant le dictionnaire)?

```
select username from 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 ?

```
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)

```
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). Vérifiez que les tables du TP1 ont été réellement créées (afficher la liste des tables de l'utilisateur connecté) ? Donner toutes les informations sur ces tables ?

Pour consulter les tables de l'utilisateur connecté veut dire les tables système, on exécute la requête suivante :

- Liste des tables de l'utilisateur connecté qui est Système

Select table_name from user_tables;

toutes les informations sur ces tables

Select * from user_tables;

```
SQL> Select table_name from user_tables;
```

```
TABLE_NAME
```

```
-----  
LOGMNR_PARAMETER$  
LOGMNR_SESSION$  
MVIEW$_ADV_WORKLOAD  
MVIEW$_ADV_Basetable  
MVIEW$_ADV_SQLDEPEND  
MVIEW$_ADV_PRETTY  
MVIEW$_ADV_TEMP  
MVIEW$_ADV_FILTER  
MVIEW$_ADV_LOG  
MVIEW$_ADV_FILTERINSTANCE  
MVIEW$_ADV_LEVEL
```

```
TABLE_NAME
```

```
-----  
MVIEW$_ADV_ROLLUP  
MVIEW$_ADV_AJG  
MVIEW$_ADV_FJG  
MVIEW$_ADV_GC  
MVIEW$_ADV_CLIQUE  
MVIEW$_ADV_ELIGIBLE  
MVIEW$_ADV_OUTPUT  
MVIEW$_ADV_EXCEPTIONS  
MVIEW$_ADV_PARAMETERS  
MVIEW$_ADV_INFO  
MVIEW$_ADV_JOURNAL
```

```
TABLE_NAME
```

```
-----  
MVIEW$_ADV_PLAN  
AQ$_QUEUE_TABLES  
AQ$_QUEUES  
AQ$_SCHEDULES  
AQ$_INTERNET_AGENTS  
AQ$_INTERNET_AGENT_PRIVS  
DEF$_ERROR  
DEF$_DESTINATION  
DEF$_CALLDEST  
DEF$_DEFAULTDEST  
DEF$_LOB
```

```
TABLE_NAME
```

```
-----  
DEF$_PROPAGATOR  
DEF$_ORIGIN
```

```
LOGMNR_FILTER$  
LOGMNR_AGE_SPILL$  
LOGMNR_SPILL$  
LOGMNR_RESTART_CKPT_TXINFO$  
LOGMNR_SESSION_ACTIONS$
```

```
154 rows selected.
```


select count(*) from user_tables;

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

COUNT(*)
-----
        154
```

-Vérification que les tables de user TP1 on été bien crée :

connect dbaintervention/psw ;

select table_name from user_tables;

```
SQL> connect dbaintervention/psw
Connected.
SQL> select table_name from user_tables;

TABLE_NAME
-----
CLIENT
EMPLOYE
MARQUE
MODELE
VEHICULE
INTERVENTIONS
INTERVENANTS
TABLEERREURS

8 rows selected.
```

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

-A partir de SYSTEM :

Select table_name from user_tables;

select table_name from all_tables where owner='DBAINTERVENTION';

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

TABLE_NAME
-----
CLIENT
EMPLOYE
MARQUE
MODELE
VEHICULE
INTERVENTIONS
INTERVENANTS
TABLEERREURS

8 rows selected.
```

7) Donner la description des attributs des tables VEHICULE et INTERVENTIONS (Exploiter la table USER_TAB_COLUMNS).

SELECT * from USER_TAB_COLUMNS where TABLE_NAME='VEHICULE';

**SELECT table_name,column_name,data_type,data_length from
USER_TAB_COLUMNS where TABLE_NAME='VEHICULE';**

```
SQL> connect dbaintervention/psw
Connected.
SQL> SELECT * from USER_TAB_COLUMNS where TABLE_NAME='VEHICULE'
2 ;
```

```
SQL> SELECT table_name,column_name,data_type,data_length from USER_TAB_COLUMNS where TABLE_NAME='VEHICULE';
```

TABLE_NAME	COLUMN_NAME	DATA_TYPE	DATA_LENGTH
VEHICULE	NUMVEHICULE	NUMBER	22
VEHICULE	NUMCLIENT	NUMBER	22
VEHICULE	NUMMODELE	NUMBER	22
VEHICULE	NUMIMMAT	VARCHAR2	20
VEHICULE	ANNEE	VARCHAR2	4

**SELECT table_name, column_name, data_type, data_length from
USER_TAB_COLUMNS where TABLE_NAME='INTERVENTIONS';**

```
SQL> SELECT table_name, column_name, data_type, data_length from USER_TAB_COLUMNS where TABLE_NAME='INTERVENTIONS';
```

TABLE_NAME	COLUMN_NAME	DATA_TYPE	DATA_LENGTH
INTERVENTIONS	NUMINTERVENTION	NUMBER	22
INTERVENTIONS	NUMVEHICULE	NUMBER	22
INTERVENTIONS	TYPEINTERVENTION	VARCHAR2	50
INTERVENTIONS	DATEDEBINTERV	DATE	7
INTERVENTIONS	DATEFININTERV	DATE	7
INTERVENTIONS	COUTINTERV	NUMBER	22

6 rows selected.

8) Comment peut-on vérifier qu'il y a une référence de clé étrangère entre les tables VEHICULE et INTERVENTIONS?

R → referential integrity ou Foreign key

```
SQL> SELECT constraint_name, constraint_type FROM USER_CONSTRAINTS WHERE table_name = 'VEHICULE';
```

CONSTRAINT_NAME	C
-----	-
PK_VEHICULE	P
FK_CLIENT	R
FK_MODELE	R

```
SQL> SELECT constraint_name, constraint_type FROM USER_CONSTRAINTS WHERE table_name = 'INTERVENTIONS';
```

CONSTRAINT_NAME	C
-----	-
PK_INTERVENTION	P
FK_VEHICULE	R
CK_DATEINTER	C

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

connect dbaintervention/psw

SELECT constraint_name, table_name , constraint_type FROM USER_CONSTRAINTS;

```
SQL> connect dbintervention/psw
Connected.
SQL>
SQL> SELECT constraint_name, table_name , constraint_type FROM USER_CONSTRAINTS;
```

CONSTRAINT_NAME	TABLE_NAME	C
CK_CIV	CLIENT	C
PK_CLIENT	CLIENT	P
CK_CATEGORIE	EMPLOYE	C
CK_EMPLOYE	EMPLOYE	P
PK_MARQUE	MARQUE	P
PK_MODELE	MODELE	P
FK_MARQUE	MODELE	R
PK_VEHICULE	VEHICULE	P
FK_CLIENT	VEHICULE	R
FK_MODELE	VEHICULE	R
PK_INTERVENTION	INTERVENTIONS	P

CONSTRAINT_NAME	TABLE_NAME	C
FK_VEHICULE	INTERVENTIONS	R
PK_INTERVENANTS	INTERVENANTS	P
FK_INTERVENTIONS	INTERVENANTS	R
FK_EMPLOYE	INTERVENANTS	R
SYS_C009180	EMPLOYE	C
CK_SALAIRE_NULL	EMPLOYE	C
CK_DATEINTER	INTERVENTIONS	C

18 rows selected.

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

select column_name,data_type,data_length,nullable from all_tab_columns where table_name='INTERVENTIONS' and owner='DBAINTERVENTION';

```
SQL> select column_name,data_type,data_length,nullable from all_tab_columns where table_name='INTERVENTIONS' and owner='DBAINTERVENTION';
```

COLUMN_NAME	DATA_TYPE	DATA_LENGTH	N
NUMINTERVENTION	NUMBER	22	N
NUMVEHICULE	NUMBER	22	Y
TYPEINTERVENTION	VARCHAR2	50	Y
DATEDEBINTERV	DATE	7	Y
DATEFININTERV	DATE	7	Y
COUTINTERV	NUMBER	22	Y

6 rows selected.

11). Trouver tous les privilèges accordés à Admin (comme on les a supprimé dans le TP2, recréez 2 privilèges système et un privilège objet pour admin et les afficher en tant que admin et en tant que system).

```
SQL> connect
Enter user-name: SYSTEM
Enter password:
Connected.
```

Select privilege, admin_option from dba_sys_privs where grantee=upper('admin');

```
SQL> select privilege , admin_option from dba_sys_privs where grantee=upper('admin');

no rows selected
```

-Ajout de 2 privilèges système:

Grant create session, create table to admin;

```
SQL> grant create session, create table to admin;

Grant succeeded.
```

-Affichage en tant que System :

select privilege from dba_sys_privs where grantee='ADMIN';

```
SQL> select privilege from dba_sys_privs where grantee='ADMIN';

PRIVILEGE
-----
CREATE TABLE
CREATE SESSION
```

-Ajout d'un privilège objet :

connect admin/psw

select privilege from user_tab_privs;

```
SQL> connect admin/psw
Connected.
SQL> select privilege from user_tab_privs;

no rows selected
```

connect dbaintervention/psw

grant select on INTERVENANTS to admin;

```
SQL> connect dbaintervention/psw
Connected.
SQL> grant select on INTERVENANTS to admin;

Grant succeeded.
```

-Affichage en tant que Admin :

connect admin/psw

select privilege from user_sys_privs;

```
SQL> connect admin/psw
Connected.
SQL> select privilege from user_sys_privs;

PRIVILEGE
-----
CREATE TABLE
CREATE SESSION
```

select privilege from user_tab_privs;

```
SQL> select privilege from user_tab_privs;

PRIVILEGE
-----
SELECT
```

-Affichage en tant que System :

select privilege from dba_tab_privs where grantee='ADMIN';

```
SQL> connect
Enter user-name: SYSTEM
Enter password:
Connected.
SQL> select privilege from dba_tab_privs where grantee='ADMIN';

PRIVILEGE
-----
SELECT
```

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

-A partir du System

select granted_role from dba_role_privs where grantee='ADMIN';

```
SQL> select granted_role from dba_role_privs where grantee='ADMIN';

GRANTED_ROLE
-----
GESTIONNAIRE_DES_INTERVENTIONS
```

-A partir du Admin

select granted_role from user_role_privs;

```
SQL> connect admin/psw
Connected.
SQL> select granted_role from user_role_privs;

GRANTED_ROLE
-----
GESTIONNAIRE_DES_INTERVENTIONS
```

13). Trouver tous les objets appartenant à Admin.

select object_name, object_type from user_objects;

```
SQL> select object_name, object_type from user_objects;

OBJECT_NAME                                OBJECT_TYPE
-----
NOMEMP_IX                                INDEX
VIEW1                                    VIEW
TEST                                    TABLE
```

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

select owner from dba_tables where table_name='INTERVENTIONS';

```
SQL> select owner from dba_tables where table_name='INTERVENTIONS';

OWNER
-----
DBAINTERVENTION
```

15). Donner la taille en Ko de la table INTERVENTIONS (utiliser desc user_segments;).

connect dbaintervention/psw

select Bytes/1024 from user_segments where

```
SQL> connect dbaintervention/psw
Connected.
SQL> select Bytes/1024 from user_segments where segment_name='INTERVENTIONS';

BYTES/1024
-----
64
```

segment_name='INTERVENTIONS';

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

Créez un nouvel utilisateur comme dans le tp1, donner lui tous les privilèges ensuite connectez-vous avec cet utilisateur que vous venez de créer

Create user tp1 identified by psw;

User created.

Grant all privileges to tp1;

Grant succeeded.

select TABLE_NAME from USER_TABLES;

select COLUMN_NAME from USER_TAB_COLUMNS;

select CONSTRAINT_NAME from USER_CONSTRAINTS;

```
SQL> select TABLE_NAME from USER_TABLES;

no rows selected

SQL>
SQL> select COLUMN_NAME from USER_TAB_COLUMNS;

no rows selected

SQL>
SQL> select CONSTRAINT_NAME from USER_CONSTRAINTS;

no rows selected
```

Créez les tables du tp1, toujours en étant connecté avec ce nouvel utilisateur

(Les requêtes de création des tables dans le Tp1).

vérifiez ce qu'il y a dans le dictionnaire en exécutant les requêtes précédentes:

select TABLE_NAME from USER_TABLES;

```
Connected.
SQL> select TABLE_NAME from USER_TABLES;

TABLE_NAME
-----
CLIENT
EMPLOYE
MARQUE
MODELE
VEHICULE
INTERVENTIONS
INTERVENANTS
TABLEERREURS

8 rows selected.
```

select COLUMN_NAME from USER_TAB_COLUMNS;

```
SQL> select COLUMN_NAME from USER_TAB_COLUMNS;
```

```
COLUMN_NAME
```

```
-----
```

```
FAX  
TELPRIV  
TELPROF  
ADRESSE  
DATENAISSANCE  
PRENOM  
NOM  
CIV  
NUMCLIENT  
SALAIRE  
CATEGORIE
```

```
COLUMN_NAME
```

```
-----
```

```
PRENOMEMP  
NOMEMP  
NUMEMPLOYE  
DATEFIN  
DATEDEBUT  
NUMEMPLOYE  
NUMINTERVENTION  
COUTINTERV  
DATEFININTERV  
DATEDEBINTERV  
TYPEINTERVENTION
```

```
COLUMN_NAME
```

```
-----
```

```
NUMVEHICULE  
NUMINTERVENTION  
PAYS  
MARQUE  
NUMMARQUE  
MODELE  
NUMMARQUE  
NUMMODELE  
NOMCONTRAINT  
NOMTABLE  
UTILISATEUR
```

```
COLUMN_NAME
```

```
-----
```

```
ADRESSE  
ANNEE  
NUMIMMAT
```



```
COLUMN_NAME
-----
ADRESSE
ANNEE
NUMIMMAT
NUMMODELE
NUMCLIENT
NUMVEHICULE

39 rows selected.
```

select CONSTRAINT_NAME from USER_CONSTRAINTS;

```
SQL> select CONSTRAINT_NAME  from USER_CONSTRAINTS;

CONSTRAINT_NAME
-----
CK_CATEGORIE
CK_CIV
FK_INTERVENTIONS
FK_VEHICULE
FK_MODELE
FK_MARQUE
FK_EMPLOYE
FK_CLIENT
PK_CLIENT
CK_EMPLOYE
PK_MARQUE

CONSTRAINT_NAME
-----
PK_MODELE
PK_VEHICULE
PK_INTERVENTION
PK_INTERVENANTS

15 rows selected.
```

Modifier le nom d'un attribut :

Alter table client rename column TELPROF to TELEPHO;

Ajouter un attribut comme dans le tp1 :

alter table employe add DATEINSTALLATION date;

Ensuite vérifiez si le changement s'est produit dans le dictionnaire :

```
SQL> select column_name from user_tab_columns;
```

```
COLUMN_NAME
```

```
-----
```

```
FAX  
TELPRIV  
TELEPHO  
ADRESSE  
DATENAISSANCE  
PRENOM  
NOM  
CTV  
NUMCLIENT  
DATEINSTALLATION  
SALAIRE
```

```
COLUMN_NAME
```

```
-----
```

```
CATEGORIE  
PRENOMEMP  
NOMEMP  
NUMEMPLOYE  
DATEFIN  
DATEDEBUT  
NUMEMPLOYE  
NUMINTERVENTION  
COUTINTERV  
DATEFININTERV  
DATEDEBINTERV
```

```
COLUMN_NAME
```

```
-----
```

```
TYPEINTERVENTION  
NUMVEHICULE  
NUMINTERVENTION  
PAYS  
MARQUE  
NUMMARQUE  
MODELE  
NUMMARQUE  
NUMMODELE  
NOMCONTRAINTTE
```

```

NOMCONTRAINTE
NOMTABLE

COLUMN_NAME
-----
UTILISATEUR
ADRESSE
ANNEE
NUMIMMAT
NUMMODELE
NUMCLIENT
NUMVEHICULE

40 rows selected.

```

-Le nombres de ligne de dictionnaire des colonnes est changé de 39 à 40 car on ajouté un attribut qui est DATEINSTALLATION.

Et pour finir ajoutez une contrainte check comme dans le tp1 et vérifier encore une fois le dictionnaire

select CONSTRAINT_NAME from USER_CONSTRAINTS; (afficher les contraintes de la table concernée)

```

SQL> alter table employe add constraint ck_salaire_null check(SALAIRE is not null);

Table altered.

SQL> select CONSTRAINT_NAME from USER_CONSTRAINTS;

CONSTRAINT_NAME
-----
CK_CATEGORIE
CK_CIV
CK_SALAIRE_NULL
FK_INTERVENTIONS
FK_VEHICULE
FK_MODELE
FK_MARQUE
FK_EMPLOYE
FK_CLIENT
PK_CLIENT
CK_EMPLOYE

CONSTRAINT_NAME
-----
PK_MARQUE
PK_MODELE
PK_VEHICULE
PK_INTERVENTION
PK_INTERVENANTS

16 rows selected.

```

```

SQL> select constraint_name, constraint_type from user_constraints where table_name=upper('employe');

CONSTRAINT_NAME          C
-----
CK_CATEGORIE              C
CK_EMPLOYE                P
CK_SALAIRE_NULL           C

```

Connectez-vous en tant que system et exécutez :

Desc ALL_TAB_COLUMNS

select OWNER, TABLE_NAME, COLUMN_NAME, DATA_TYPE from
ALL_TAB_COLUMNS where table_name='EMPLOYE';

```
SQL> connect
Enter user-name: SYSTEM
Enter password:
Connected.
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)