18/12/2024

Compte rendu TP oracle

BD architecture & administration

NOM : Nait-Cherif

PRENOM : Sabrinel

MATRICULE : 222231346615

Niveau : 3 éme année ING Software

Travail demandé par: M. Ferrahi

Table des matières

Partie I : Création des Tablespaces et des utilisateurs

1) Créer deux Tablespaces IOT_TBS et IOT_TempTBS

```
--IOT_TBS --
create  tablespace IOT_TBS
datafile 'C:\IOT_TBS.dat'
size 100M
Autoextend on
online;
--IOT_TempTBS--
create  temporary  tablespace IOT_TempTBS
tempfile  'C:\IOT_TempTBS.dat'
size 100M
Autoextend on ;
```

```
SQL> create  tablespace IOT_TBS
  2  datafile 'C:\IOT_TBS.dat'
  3  size 100M
  4  Autoextend on
  5  online;

Tablespace created.
```

```
SQL> --IOT_TempTBS--
SQL> create  temporary  tablespace IOT_TempTBS
  2  tempfile  'C:\IOT_TempTBS.dat'
  3  size 100M
  4  Autoextend on ;

Tablespace created.
```

2) Créer un utilisateur DBAIOT en lui attribuant les deux tablespaces créés précédemment

```
create user DBAIOT  identified by sabrinel
default tablespace IOT_TBS
temporary tablespace  IOT_TempTBS;
```

```
SQL> create user DBAIOT  identified by sabrinel
  2  default tablespace IOT_TBS
  3  temporary tablespace  IOT_TempTBS;

User created.
```

3) Donner tous les privilèges à cet utilisateur

```
grant all privileges to DBAIOT;
```

```
SQL> grant all privileges to DBAIOT;

Grant succeeded.
```

Partie 2 : Langage de définition de données

4) Créer les relations de base avec toutes les contraintes d'intégrité.

USER (**IDUSER**, LASTNAME, LASTNAME, EMAIL)

SERVICE (**IDSERVICE**, NAME, SERVICETYPE)

THING (**MAC**, IDUSER*, THINGTYPE, PARAM)

SUBSCRIBE (**IDUSER*, IDSERVICE***)

```
-- création de la table user
create table utilisateur (-- ici, j'ai changé le nom de la table de "user" en "utilisateur" pour éviter un conflit avec le mot réservé "USER" dans Oracle. (voir figure 1)qa
    iduser number primary key,
    lastname varchar2(50) ,
    firstname varchar2(50),
    email varchar2(50)
);
-- création de la table service
create table service (
    idservice number primary key,
    name varchar2(50),
    servicetype varchar2(50)
);
-- création de la table thing
create table thing (
    mac char(17) primary key,
    iduser number ,
    thingtype varchar2(50),
    param number ,
    constraint fk_iduser foreign key (iduser) references utilisateur(iduser) on delete cascade
);
-- création de la table subscribe
create table subscribe (
    iduser number ,
    idservice number ,
    constraint pk_cleprimaire primary key (iduser, idservice),
    constraint fk_iduser2 foreign key (iduser) references utilisateur(iduser) on delete cascade,
    constraint fk_idservice foreign key (idservice) references service(idservice) on delete cascade
);
```

```
create table user (
             *
ERROR at line 1:
ORA-00903: invalid table name
```

Figure 1

```
SQL> -- création de la table user
SQL> create table utilisateur (-- ici, j'ai changé
avec le mot réservé "USER" dans Oracle.
  2      iduser number primary key,
  3      lastname varchar2(50) ,
  4      firstname varchar2(50),
  5      email varchar2(100)
  6  );

Table created.

SQL>
SQL> -- création de la table service
SQL> create table service (
  2      idservice number primary key,
  3      name varchar2(50),
  4      servicetype varchar2(50)
  5  );

Table created.
```

```
SQL> -- création de la table thing
SQL> create table thing (
  2      mac char(17) primary key,
  3      iduser number ,
  4      thingtype varchar2(50),
  5      param varchar2(50),
  6      constraint fk_iduser foreign key (iduser) references utilisateur(iduser) on delete cascade
  7  );

Table created.

SQL>
SQL> -- création de la table subscribe
SQL> create table subscribe (
  2      iduser number ,
  3      idservice number ,
  4      constraint pk_cleprimaire primary key (iduser, idservice),
  5      constraint fk_iduser2 foreign key (iduser) references utilisateur(iduser) on delete cascade,
  6      constraint fk_idservice foreign key (idservice) references service(idservice) on delete cascade
  7  );

Table created.
```

5) Ajouter l'attribut ADRESSUSER de type chaine de caractères dans la relation USER.

```
alter table utilisateur
add adressuser varchar2(50);
```

```
SQL> alter table utilisateur
  2  add adressuser varchar2(50);

Table altered.
```

6) Ajouter la contrainte not null pour les attributs ADRESSUSER et LASTNAME de la relation USER.

```
-- Ajouter la contrainte NOT NULL pour l'attribut ADRESSUSER
alter table utilisateur
modify adressuser varchar2(50) not null;
-- Ajouter la contrainte NOT NULL pour l'attribut LASTNAME
alter table utilisateur
modify lastname varchar2(50) not null;
```

```
SQL> -- Ajouter la contrainte NOT NULL pour l'attribut ADRESSUSER
SQL> alter table utilisateur
  2  modify adressuser varchar2(50)not null;

Table altered.
SQL> -- Ajouter la contrainte NOT NULL pour l'attribut LASTNAME
SQL> alter table utilisateur
  2  modify lastname varchar2(50) not null;

Table altered.
```

7) Modifier la longueur de l'attribut ADRESSUSER (agrandir, réduire).

```
-- agrandir
alter table utilisateur
modify adressuser varchar2(150);
--réduire
alter table utilisateur
modify adressuser varchar2(45);
```

```
SQL> -- agrandir
SQL> alter table utilisateur
  2  modify adressuser varchar2(150);

Table altered.

SQL> --réduire
SQL> alter table utilisateur
  2  modify adressuser varchar2(45);

Table altered.
```

8) Renommer la colonne ADRESSUSER dans la table USER par ADRUSER. Vérifier.

```
alter table utilisateur
rename column adressuser to adruser;
--verification
describe utilisateur;
```

```
SQL> alter table utilisateur
  2  rename column adressuser to adruser;

Table altered.

SQL> --verification
SQL> describe utilisateur;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------
 IDUSER                                    NOT NULL NUMBER
 LASTNAME                                  NOT NULL VARCHAR2(50)
 FIRSTNAME                                          VARCHAR2(50)
 EMAIL                                              VARCHAR2(100)
 ADRUSER                                   NOT NULL VARCHAR2(45)
```

9) Supprimer la colonne ADRUSER dans la table USER. Vérifier la suppression.

```
alter table utilisateur
drop column adruser;
--verification
describe utilisateur;
```

```
SQL> alter table utilisateur
  2  drop column adruser;

Table altered.

SQL> --verification
SQL> describe utilisateur;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------
 IDUSER                                    NOT NULL NUMBER
 LASTNAME                                  NOT NULL VARCHAR2(50)
 FIRSTNAME                                          VARCHAR2(50)
 EMAIL                                              VARCHAR2(100)
```

10) Un utilisateur s'inscrit à un service pour une période délimitée par un début et fin. Donner les instructions SQL pour répondre à ce besoin.

```
alter table subscribe
add ( debut_date date  default sysdate, --la date actuelle par defaut dans le cas ou  les champs ne sont pas renseignés
      fin_date date default (sysdate + 365) --un an après la date actuelle
);
```

```
SQL> alter table subscribe
  2  add (
  3      debut_date date  default sysdate,
  4      fin_date date default (sysdate + 365)
  5  );

Table altered.
```

Partie 3 : Langage de manipulation de données

11) Remplir toutes les tables par les instances représentées ci-dessus. Quels sont les problèmes rencontrés ?

```
-- Insertion des données dans la table utilisateur

insert into utilisateur (iduser, lastname, firstname, email) values (1, 'Souad', 'MESBAH', 'souad.mesbah@gmail.com');
```

```
insert into utilisateur (iduser, lastname, firstname, email) values (2,  'Younes', 'CHALAH','younes.chalah@gmail.com');

insert into utilisateur (iduser, lastname, firstname, email) values (3, 'Chahinaz', 'MELEK','chahinaz.melek@gmail.com');

insert into utilisateur (iduser, lastname, firstname, email) values (4, 'Samia', 'OUALI', 'samia.ouali@gmail.com');

insert into utilisateur (iduser, lastname, firstname, email) values (5, 'Djamel', 'MATI','djamel.mati@gmail.com');

insert into utilisateur (iduser, lastname, firstname, email) values (6, 'Assia', 'HORRA','assia.horra@gmail.com');

insert into utilisateur (iduser, lastname, firstname, email) values (7, 'Lamine', 'MERABAT','Lamine.MERABAT@gmail.com');

insert into utilisateur (iduser, lastname, firstname, email) values (8, 'Seddik', 'HMIA', 'seddik.hmia@gmail.com');

insert into utilisateur (iduser, lastname, firstname, email) values (9, 'Widad', 'TOUATI', 'widad.touati@gmail.com');

-- Insertion des données dans la table service

insert into service (idservice, name, servicetype) values (1, 'myKWHome', 'smarthome');

insert into service (idservice, name, servicetype) values (2, 'FridgAlert', 'smarthome');

insert into service (idservice, name, servicetype) values (3, 'RUNstats', 'quantifiedself');

insert into service (idservice, name, servicetype) values (4, 'traCARE', 'quantifiedself');

insert into service (idservice, name) values (5, 'dogWATCH');

insert into service (idservice, name) values (6, 'CarUse');

-- Insertion des données dans la table thing

insert into thing (mac, iduser, thingtype, param) values ('f0:de:f1:39:7f:17', 1,NULL, NULL);
```

```
insert into thing (mac, iduser, thingtype, param) values ('f0:de:f1:39:7f:18', 2, NULL, NULL);
insert into thing (mac, iduser, thingtype, param) values ('f0:de:f1:39:7f:19', 2, 'thingtempo', 60);
insert into thing (mac, iduser, thingtype, param) values ('f0:de:f1:39:7f:25', 10, NULL, NULL);
insert into thing (mac, iduser, thingtype, param) values ('f0:de:f1:39:7f:20', 2, 'thingtempo', 1.5);
insert into thing (mac, iduser, thingtype, param) values ('f0:de:f1:39:7f:21', 4, NULL,NULL);
insert into thing (mac, iduser, thingtype, param) values ('f0:de:f1:39:7f:22', 4, NULL, NULL);
-- Insertion des données dans la table subscribe
insert into subscribe (iduser, idservice) values (2, 1);
insert into subscribe (iduser, idservice) values (2, 2);
insert into subscribe (iduser, idservice) values (1, 3);
insert into subscribe (iduser, idservice) values (3, 7);
```

IDUSER	LASTNAME	FIRSTNAME	EMAIL
1	Souad	MESBAH	souad.mesbah@gmail.com
2	Younes	CHALAH	younes.chalah@gmail.com
3	Chahinaz	MELEK	chahinaz.melek@gmail.com
4	Samia	OUALI	samia.ouali@gmail.com
5	Djamel	MATI	djamel.mati@gmail.com
6	Assia	HORRA	assia.horra@gmail.com
7	Lamine	MERABAT	Lamine.MERABAT@gmail.com
8	Seddik	HMIA	seddik.hmia@gmail.com
9	Widad	TOUATI	widad.touati@gmail.com

9 rows selected.

IDSERVICE	NAME	SERVICETYPE
1	myKWHome	smarthome
2	FridgAlert	smarthome
3	RUNstats	quantifiedself
4	traCARE	quantifiedself
5	dogWATCH	
6	CarUse	

6 rows selected.

MAC	IDUSER	THINGTYPE	PARAM
f0:de:f1:39:7f:17	1		
f0:de:f1:39:7f:18	2		
f0:de:f1:39:7f:19	2	thingtempo	60
f0:de:f1:39:7f:20	2	thingtempo	1.5
f0:de:f1:39:7f:21	4		
f0:de:f1:39:7f:22	4		

6 rows selected.

```
SQL> select * from subscribe;

    IDUSER  IDSERVICE DEBUT_DAT FIN_DATE
---------- ---------- --------- ---------
         2          1 13-DEC-24 13-DEC-25
         2          2 13-DEC-24 13-DEC-25
         1          3 13-DEC-24 13-DEC-25
```

- **les problèmes rencontrés sont:**
  - Pour l'insertion de la ligne :

```
insert into thing (mac, iduser, thingtype, param) values ('f0:de:f1:39:7f:25', 10, NULL, NULL);
```

Elle entraînera une erreur à l'insertion car iduser = 10 ne correspond à aucun utilisateur existant dans la table utilisateur.
La contrainte de clé étrangère (**fk_iduser**) empêche l'insertion d'une ligne dans la table **thing** si **iduser** ne correspond pas à un utilisateur valide d+ans la table utilisateur.

  - Pour l'insertion de la ligne :

```
insert into subscribe (iduser, idservice) values (3, 7);
```

Elle entraînera une erreur à l'insertion car idservice = 7 ne correspond à aucun service existant dans la table service.
La contrainte de clé étrangère (**fk_idservice**) empêche l'insertion d'une ligne dans la table **subscribe** si **idservice** ne correspond pas à un service valide dans la table service.

Partie 4 : Gestion des utilisateurs

1) Connectez-vous avec l'utilisateur DBAIOT et créez un autre utilisateur : Admin en lui donnant les mêmes tablespace que DBAIOT.

```
connect  DBAIOT/sabrinel
create user admin identified by adm
default tablespace IOT_TBS
temporary tablespace IOT_TempTBS;
```

```
SQL> connect  DBAIOT/sabrinel
Connected.
```

```
SQL> create user admin identified by adm
  2  default tablespace IOT_TBS
  3  temporary tablespace IOT_TempTBS;

User created.
```

2) Connectez-vous à l'aide cet utilisateur. Que remarquez-vous ?

```
SQL> connect admin/adm
ERROR:
ORA-01045: user ADMIN lacks CREATE SESSION privilege; logon denied
```

Remarque : Il est impossible de se connecter avec l'utilisateur **admin** car il ne dispose pas des privilèges nécessaires pour ouvrir une session.

3) Donnez le droit de création d'une session pour cet utilisateur (Create Session) et reconnectez-vous.

```
grant create session to admin;

connect admin/adm
```

```
SQL> connect  DBAIOT/sabrinel
Connected.
SQL> grant create session to admin;

Grant succeeded.

SQL> connect admin/adm
Connected.
```

4) Donnez les privilèges suivants à Admin: créer des tables, des utilisateurs. Vérifiez.

```
connect DBAIOT/sabrinel
grant create table, create user to admin;
```

```
SQL> connect DBAIOT/sabrinel
Connected.
SQL> grant create table, create user to admin;

Grant succeeded.
```

Pour la verification:

```
select * from user_sys_privs;
```

```
SQL> select * from USER_SYS_PRIVS;

USERNAME                       PRIVILEGE                                ADM
------------------------------ ---------------------------------------- ---
ADMIN                          CREATE TABLE                             NO
ADMIN                          CREATE SESSION                           NO
ADMIN                          CREATE USER                              NO
```

5) Exécutez la requête Q1 suivante : Select * from DBAIOT.Utilisateur ; Que remarquez-vous ?

Remarque : L'utilisateur **admin** ne dispose pas des privilèges nécessaires pour accéder (lecture) à la table **DBAIOT.Utilisateur** .

```
SQL> connect admin/adm
Connected.
SQL> Select * from DBAIOT.utilisateur ;
Select * from DBAIOT.utilisateur
                     *
ERROR at line 1:
ORA-00942: table or view does not exist
```

6) Donnez le droit de lecture à cet utilisateur pour la table Utilisateur. Exécutez la requête Q1 maintenant.

```
connect  DBAIOT/sabrinel
grant select on utilisateur to admin;
Select * from DBAIOT.Utilisateur ;
```

```
SQL> connect  DBAIOT/sabrinel
Connected.
SQL> grant select on utilisateur to admin;

Grant succeeded.
```

```
SQL>  Select * from DBAIOT.Utilisateur ;

    IDUSER LASTNAME                                           FIRSTNAME                                           EMAIL
---------- -------------------------------------------------- --------------------------------------------------
         1 Souad                                              MESBAH                                              souad.mesbah@gmail.com
         2 Younes                                             CHALAH                                              younes.chalah@gmail.com
         3 Chahinaz                                           MELEK                                               chahinaz.melek@gmail.com
         4 Samia                                              OUALI                                               samia.ouali@gmail.com
         5 Djamel                                             MATI                                                djamel.mati@gmail.com
         6 Assia                                              HORRA                                               assia.horra@gmail.com
         7 Lamine                                             MERABAT                                             Lamine.MERABAT@gmail.com
         8 Seddik                                             HMIA                                                seddik.hmia@gmail.com
         9 Widad                                              TOUATI                                              widad.touati@gmail.com

9 rows selected.
```

7) On veut créer une vue USER_THING qui sauvegarde pour chaque utilisateur ses objets connectés. Que faut-il faire ? Que remarquez-vous ?

Remarque : L'utilisateur **admin** ne pourra pas créer la vue **USER_THING**, car il ne dispose pas des privilèges nécessaires pour créer des vues.

```
create view user_thing as
select u.iduser, u.firstname, u.lastname, t.thing_name, t.thing_type
from utilisateur u
join things t on u.iduser = t.user_id;
```

```
SQL> create view user_thing as
  2  select u.iduser, u.firstname, u.lastname, t.thing_name, t.thing_type
  3  from utilisateur u
  4  join things t on u.iduser = t.user_id;
join things t on u.iduser = t.user_id
     *
ERROR at line 4:
ORA-00942: table or view does not exist
```

8) Donnez le droit de création de vue à cet utilisateur, le droit de lecture sur la table THING et réessayez de refaire la création de la vue.

```
connect DBAIOT/sabrinel
grant create view to admin;
grant select on DBAIOT.utilisateur to admin;
grant select on DBAIOT.thing to admin;
connect admin/adm
```

```
create view user_thing as
select u.iduser, u.lastname, t.mac, t.thingtype
from DBAIOT.utilisateur u
join DBAIOT.thing t on u.iduser = t.iduser;
```

```
SQL> connect DBAIOT/sabrinel
Connected.
SQL>
SQL> grant create view to admin;

Grant succeeded.

SQL> grant select on thing to admin;

Grant succeeded.
```

```
SQL> connect admin/adm
Connected.
SQL> create view user_thing as
  2  select u.iduser, u.lastname, t.mac, t.thingtype
  3  from DBAIOT.utilisateur u
  4  join DBAIOT.thing t on u.iduser = t.iduser;

View created.
```

9) Créez un index NAMESERVICE_IX sur l'attribut NAME de la table SERVICE. Que remarquez-vous ?

```
create index nameservice_ix on service(name);
```

Remarque : l'utilisateur **admin** ne pourra pas créer l'index **NAMESERVICE_IX**, car il ne dispose pas des privilèges nécessaires pour créer un index.

```
SQL> create index nameservice_ix on service(name);
create index nameservice_ix on service(name)
                               *
ERROR at line 1:
ORA-00942: table or view does not exist
```

10) Donnez le droit de création d'index à Admin pour la table SERIVCE, ensuite réessayez de créer l'index. Que se passe-til ?

```
connect DBAIOT/sabrinel;
grant create  any index to admin;
grant select on DBAIOT.service to admin;
connect admin/adm;
create index nameservice_ix on DBAIOT.service(name);
```

```
SQL> grant create  any index to admin;

Grant succeeded.

SQL> grant select on DBAIOT.service to admin;

Grant succeeded.

SQL> connect admin/adm;
Connected.
SQL> create index nameservice_ix on DBAIOT.service(name);
create index nameservice_ix on DBAIOT.service(name)
                                       *
ERROR at line 1:
ORA-01950: no privileges on tablespace 'IOT_TBS'
```

Cette erreur:

Indique que l'utilisateur **admin** ne dispose pas des privilèges nécessaires pour créer un index dans le **tablespace IOT_TBS.**

```
connect dbaiot/sabrinel
grant unlimited tablespace to admin;
Alter user admin QUOTA UNLIMITED on IOT_TBS
connect admin/adm;
create index nameservice_ix on DBAIOT.service(name);
```

Remarque :

Alter user admin QUOTA UNLIMITED on IOT_TBS ne compromet pas la sécurité des donnée**s** (l'accès non autorisé aux données),mais elle peut entraîner des problèmes de gestion des ressources, de performance, et d'utilisation excessive de l'espace donc pouvez accorder un quota limité à l'utilisateur.

```
SQL> connect dbaiot/sabrinel
Connected.
SQL> ALTER USER admin QUOTA UNLIMITED ON IOT_TBS;

User altered.

SQL> connect admin/adm
Connected.
SQL> create index nameservice_ix on DBAIOT.service(name);

Index created.
```

11) Enlevez les privilèges précédemment accordés.

```
connect DBAIOT/sabrinel;
-- privilege de creation d'index
revoke create any index from admin;
-- le privilege SELECT sur la table SERVICE
revoke select on DBAIOT.service from admin;
-- le privilege SELECT sur la table UTILISATEUR
revoke select on DBAIOT.utilisateur from admin;
-- le privilege SELECT sur la table thing
grant select on DBAIOT.thing to admin;
--le privilege de creation de vue
revoke create view from admin;
-- le privilege de creation de session
revoke create session from admin;
-- les privileges de creation de table et d'utilisateur
revoke create table, create user from admin;
 -- les privileges table space
revoke unlimited tablespace from admin;
```

```
SQL> connect DBAIOT/sabrinel;
Connected.
SQL> revoke create any index from admin;

Revoke succeeded.

SQL> revoke select on DBAIOT.service from admin;

Revoke succeeded.

SQL> revoke select on DBAIOT.utilisateur from admin;

Revoke succeeded.

SQL> revoke create view from admin;

Revoke succeeded.

SQL> revoke create session from admin;

Revoke succeeded.

SQL> revoke create table, create user from admin;

Revoke succeeded.
```

```
SQL> revoke unlimited tablespace from admin;

Revoke succeeded.
```

12) Vérifiez que les privilèges ont bien été supprimés.

```
connect DBAIOT/sabrinel;
grant create session to admin;
connect admin/adm
select * from user_sys_privs;
revoke create session from admin;
```

```
SQL> connect admin/adm
Connected.
SQL> select * from user_sys_privs;

USERNAME                       PRIVILEGE                                ADM
------------------------------ ---------------------------------------- ---
ADMIN                          CREATE SESSION                           NO
```

13) Créez un profil « IOT_Profil » qui est caractérisé par : (3 sessions simultanés autorisées, Un appel système ne peut pas consommer plus de 35 secondes de CPU, Chaque session ne peut excéder 90 minutes, Un appel système ne peut lire plus de 1200 blocs de donnes en mémoire et sur le disque, Chaque session ne peut allouer plus de 25 ko de mémoire en SGA, Pour chaque session, 30 minutes d'inactivité maximum sont autorisés, 5 tentatives de connexion avant blocage du compte, Le mot de passe est valable pendant 50 jours et il faudra attendre 40 jours avant qu'il puisse être utilisé à nouveau, 1 seul jour d'interdiction d'accès après que les 5 tentatives de connexion ont été atteintes, La période de grâce qui prolonge l'utilisation du mot de passe avant son changement est de 5 jours).

```
Create profile IOT_Profil limit
-- 3 sessions simultanées autorisées
   SESSIONS_PER_USER          3
-- Un appel système ne peut pas consommer plus de 35 secondes de CPU
   CPU_PER_SESSION            35
-- Chaque session ne peut excéder 90 minutes (60*90=5400 secondes)
   CONNECT_TIME               5400
-- Un appel système ne peut lire plus de 1200 blocs de données en mémoire et sur le disque
   LOGICAL_READS_PER_SESSION  1200
-- Chaque session ne peut allouer plus de 25 Ko de mémoire en SGA
PRIVATE_SGA                   25600
-- 30 minutes d'inactivité maximum autorisées par session
   IDLE_TIME                  30
-- 5 tentatives de connexion avant blocage du compte
 FAILED_LOGIN_ATTEMPTS        5
-- Le mot de passe est valable pendant 50 jours
   PASSWORD_LIFE_TIME         50
-- Il faut attendre 40 jours avant qu'un mot de passe puisse être réutilisé
PASSWORD_REUSE_TIME           40
-- 1 jour d'interdiction d'accès après 5 tentatives échouées de connexion
```

```
   PASSWORD_LOCK_TIME        1
-- La période de grâce pour prolonger l'utilisation du mot de passe avant son changement est de 5 jours
   PASSWORD_GRACE_TIME       5;
```

```
SQL> CREATE PROFILE IOT_Profil LIMIT
  2       SESSIONS_PER_USER           3
  3       CPU_PER_SESSION             35
  4       CONNECT_TIME                5400
  5       LOGICAL_READS_PER_SESSION   1200
  6       PRIVATE_SGA                 25600
  7       IDLE_TIME                   30
  8       FAILED_LOGIN_ATTEMPTS       5
  9       PASSWORD_LIFE_TIME          50
 10       PASSWORD_REUSE_TIME         40
 11       PASSWORD_LOCK_TIME          1
 12       PASSWORD_GRACE_TIME         5;

Profile created.
```

14) Affectez ce profil à l'utilisateur Admin.

```
alter user admin profile iot_profil;
```

```
SQL> alter user admin profile iot_profil;

User altered.
```

15) Créez le rôle : « SUBSCRIBE_MANAGER » qui peut voir les tables USERS, SERVICE et peut modifier les lignes de la table SUBSCRIBE.

```
create role subscribe_manager;

grant select on dbaiot.utilisateur to subscribe_manager;
grant select on dbaiot.service to subscribe_manager;
grant update on dbaiot.subscribe to subscribe_manager;
```

```
SQL> create role subscribe_manager;

Role created.

SQL> grant select on dbaiot.utilisateur to subscribe_manager;

Grant succeeded.

SQL> grant select on dbaiot.service to subscribe_manager;

Grant succeeded.

SQL> grant update on dbaiot.subscribe to subscribe_manager;

Grant succeeded.
```

16) Assignez ce rôle à Admin. Vérifier que les autorisations assignées au rôle SUBSCRIBE_MANAGER , ont été bien transférées sur l'utilisateur à Admin.

```
grant subscribe_manager to admin;
connect admin/adm
select * from user_tab_privs where grantee = 'ADMIN';
select * from user_sys_privs where grantee = 'ADMIN';
select * from user_role_privs where grantee = 'ADMIN';
```

```
SQL> select * from user_sys_privs;

USERNAME                       PRIVILEGE                                ADM
------------------------------ ---------------------------------------- ---
ADMIN                          CREATE ANY INDEX                         NO
ADMIN                          CREATE SESSION                           NO
```

```
SQL> select * from user_role_privs;

USERNAME                       GRANTED_ROLE                   ADM DEF OS_
------------------------------ ------------------------------ --- --- ---
ADMIN                          SUBSCRIBE_MANAGER              NO  YES NO
```

Partie 5 : Dictionnaire de données

1) Connecter en tant que « System ». Lister le catalogue « DICT ». Il contient combien d'instances ? Donner sa structure? (Describe DICT; select * from dict;)

```
connect system/1234
select * from dict;
Le dictionnaire contient

select count(*) from dict;
describe dict;
```

Le catalogue DICT est un ensemble de vues et de tables qui contiennent des informations système concernant la base de données Oracle.

La structure du catalogue DICT peut être représentée comme suit :

Nom de la table	**Commentaires**
TABLE_NAME	Contient des métadonnées sur tous les objets définis dans la base de données, tels que les tables, les vues et les colonnes.
COMMENTS	Les tables et vues dans le catalogue DICT décrivent des objets internes de la base de données, comme les définitions de tables, les contraintes, les index, les colonnes et les types de données.

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

ALL_JOBS
Synonym for USER_JOBS


2551 rows selected.
```

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

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

SQL> DESCRIBE DICT;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 TABLE_NAME                                         VARCHAR2(30)
 COMMENTS                                           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

Remarque : dans cette question j'ai utilisé le site oracle comme référence

- ALL_TAB_COLUMNS

Rôle : La vue ALL_TAB_COLUMNS fournit des informations sur les colonnes des tables, vues et clusters accessibles à l'utilisateur actuel. Elle permet d'obtenir des détails sur la structure des objets de la base de données auxquels l'utilisateur a accès.

```
describe all_tab_columns;

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
```

Structure :

Colonne	**Description**
OWNER	Nom de l'utilisateur propriétaire de la table, vue ou cluster.
TABLE_NAME	Nom de la table, vue ou cluster.
COLUMN_NAME	Nom de la colonne.
DATA_TYPE	Type de données de la colonne (par exemple, VARCHAR2, NUMBER).
DATA_TYPE_MOD	Modificateur du type de données de la colonne.
DATA_TYPE_OWNER	Propriétaire du type de données de la colonne.
DATA_LENGTH	Longueur de la colonne en octets.
DATA_PRECISION	Précision décimale pour le type de données NUMBER; précision binaire pour FLOAT; NULL pour les autres types.
DATA_SCALE	Nombre de chiffres à droite de la virgule décimale pour le type NUMBER.

Colonne	Description
NULLABLE	Indique si la colonne peut accepter des valeurs NULL (Y pour oui, N pour non).
COLUMN_ID	Identifiant de la colonne dans la table.
DEFAULT_LENGTH	Longueur de la valeur par défaut de la colonne.
DATA_DEFAULT	Valeur par défaut de la colonne, si définie.
NUM_DISTINCT	Nombre de valeurs distinctes dans la colonne.
NUM_NULLS	Nombre de valeurs NULL dans la colonne.
NUM_BUCKETS	Nombre de compartiments dans l'histogramme pour la colonne.
LAST_ANALYZED	Date de la dernière analyse de la colonne.
SAMPLE_SIZE	Taille de l'échantillon utilisée lors de l'analyse de la colonne.
CHARACTER_SET_NAME	Nom du jeu de caractères utilisé pour les colonnes de type chaîne.
CHAR_COL_DECL_LENGTH	Longueur déclarée de la colonne de type caractère.

- **USER_USERS**

Rôle : La vue USER_USERS fournit des informations sur l'utilisateur actuel connecté à la base de données. Elle permet d'obtenir des détails sur l'utilisateur, tels que son nom, son statut et d'autres attributs associés.

```
describe user_users;

select comments from dict where table_name = ' USER_USERS ';
```

```
SQL> describe 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)
```

```
SQL> SELECT comments FROM dict WHERE table_name = 'USER_USERS';

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

Structure :

Colonne	Description
USERNAME	Nom de l'utilisateur.
USER_ID	Identifiant de l'utilisateur dans la base de données.
CREATED	Date de création de l'utilisateur.
ACCOUNT_STATUS	Statut du compte de l'utilisateur (par exemple, OPEN, LOCKED).
EXPIRY_DATE	Date d'expiration du compte, si définie.
PROFILE	Nom du profil utilisé par l'utilisateur.
DEFAULT_TABLESPACE	Tablespace par défaut de l'utilisateur.

- **ALL_CONSTRAINTS**

Rôle : La vue ALL_CONSTRAINTS fournit des informations sur les contraintes définies sur les tables accessibles à l'utilisateur actuel. Elle permet d'obtenir des détails sur les contraintes telles que les clés primaires, les clés étrangères, les contraintes uniques, etc.

```
describe all_constraints;
```

```
select comments from dict where table_name = ' ALL_CONSTRAINT';
```

```
SQL> describe 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)
```

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

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

Structure :

Colonne	**Description**
OWNER	Nom de l'utilisateur propriétaire de la contrainte.
CONSTRAINT_NAME	Nom de la contrainte.
CONSTRAINT_TYPE	Type de contrainte (par exemple, P pour primaire, R pour clé étrangère, U pour unique).
TABLE_NAME	Nom de la table sur laquelle la contrainte est définie.
SEARCH_CONDITION	Condition associée à la contrainte, si applicable.
STATUS	Statut de la contrainte (par exemple, ENABLED ou DISABLED).

Colonne	Description
IS_DEFERRABLE	Indique si la contrainte peut être différée.
DELETE_RULE	Règle de suppression de la contrainte (par exemple, CASCADE, SET NULL).

- **USER_TAB_PRIVS**

Rôle : La vue USER_TAB_PRIVS fournit des informations sur les privilèges accordés sur les objets de la base de données (tables, vues, etc.) de l'utilisateur actuel. Elle permet de voir quels privilèges (tels que SELECT, INSERT, UPDATE, DELETE) l'utilisateur a sur des objets spécifiques.

```
describe user_tab_privs;

select comments from dict where table_name = 'user_tab_privs';
```

```
SQL> describe 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)
```

```
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
```

Structure :

Colonne	Description
GRANTEE	Nom de l'utilisateur ou rôle qui possède les privilèges.
TABLE_NAME	Nom de la table ou de la vue sur laquelle les privilèges sont accordés.
GRANTOR	Nom de l'utilisateur qui a accordé le privilège.

Colonne	Description
PRIVILEGE	Type de privilège accordé (par exemple, SELECT, INSERT, UPDATE, DELETE).
GRANTABLE	Indique si le privilège est transférable à d'autres utilisateurs (YES ou NO).
HIERARCHY	Indique si le privilège est accordé à l'utilisateur au niveau de la hiérarchie d'un objet.

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 ?

```
describe all_tab_columns;
describe user_tab_columns;
```

Comparaison :

ALL_TAB_COLUMNS pour obtenir des informations sur les colonnes de toutes les tables et vues auxquelles vous avez accès, quel que soit le schéma.

USER_TAB_COLUMNS pour consulter les colonnes des tables et vues que vous possédez.

5) Vérifiez que les tables de la partie 1 ont été réellement créées (afficher la liste des tables de l'utilisateur connecté) ? Donner toutes les informations sur ces tables ?

```
select table_name from user_tables;
--table utilisateur
select table_name, column_name, data_type, data_length, nullable
from all_tab_columns
where owner = 'DBAIOT'
 and  table_name = 'UTILISATEUR ';
```

```
--table service
select table_name, column_name, data_type, data_length, nullable
from all_tab_columns
where owner = 'DBAIOT'
 and  table_name = 'SERVICE';
--table thing
select table_name, column_name, data_type, data_length, nullable
from all_tab_columns
where owner = 'DBAIOT'
 and  table_name = 'THING';
--table suscribe
select table_name, column_name, data_type, data_length, nullable
from all_tab_columns
where owner = 'DBAIOT'
 and  table_name = 'SUBSCRIBE ';
```

```
TABLE_NAME                     COLUMN_NAME                    DATA_TYPE
                                                            DATA_LENGTH N
------------------------------ ------------------------------ ----------
---------------------------------------------- ----------- -
UTILISATEUR                    IDUSER                         NUMBER
                                                                  22 N
UTILISATEUR                    LASTNAME                       VARCHAR2
                                                                  50 N
UTILISATEUR                    FIRSTNAME                      VARCHAR2
                                                                  50 Y
UTILISATEUR                    EMAIL                          VARCHAR2
                                                                  50 Y
```

```
TABLE_NAME                     COLUMN_NAME                    DATA_TYPE
                                                            DATA_LENGTH N
------------------------------ ------------------------------ ----------
-------------------------------------------- ----------- -
SERVICE                        IDSERVICE                      NUMBER
                                                                  22 N
SERVICE                        NAME                           VARCHAR2
                                                                  50 Y
SERVICE                        SERVICETYPE                    VARCHAR2
                                                                  50 Y
```

```
TABLE_NAME                     COLUMN_NAME                    DATA_TYPE
                                                            DATA_LENGTH N
------------------------------ ------------------------------ ----------
-------------------------------------------- ----------- -
THING                          MAC                            CHAR
                                                                  17 N
THING                          IDUSER                         NUMBER
                                                                  22 Y
THING                          THINGTYPE                      VARCHAR2
                                                                  50 Y
THING                          PARAM                          NUMBER
                                                                  22 Y
```

```
TABLE_NAME                     COLUMN_NAME                    DATA_TYPE
                                                            DATA_LENGTH N
------------------------------ ------------------------------ ----------
-------------------------------------------- ----------- -
SUBSCRIBE                      IDUSER                         NUMBER
                                                                  22 N
SUBSCRIBE                      IDSERVICE                      NUMBER
                                                                  22 N
SUBSCRIBE                      DEBUT_DATE                     DATE
                                                                   7 Y
SUBSCRIBE                      FIN_DATE                       DATE
                                                                   7 Y
```

6) Lister les tables de l'utilisateur « system » et celles de l'utilisateur DBAIOT (l'utilisateur de la partie 1).

```
select table_name from user_tables;

select  table_name from all_tables

where owner = 'DBAIOT';
```

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

TABLE_NAME
------------------------------
UTILISATEUR
SERVICE
THING
SUBSCRIBE
```

```
SQL> select table_name from all_

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
MVIEW$_ADV_CLIQUE

TABLE_NAME
------------------------------
MVIEW$_ADV_ELIGIBLE
MVIEW$_ADV_OUTPUT
MVIEW$_ADV_EXCEPTIONS
MVIEW$_ADV_PARAMETERS
MVIEW$_ADV_INFO
MVIEW$_ADV_JOURNAL
MVIEW$_ADV_PLAN
AQ$_QUEUE_TABLES
AQ$_QUEUES
AQ$_SCHEDULES

TABLE_NAME
------------------------------
LOGMNR_TAB$
LOGMNR_COL$
LOGMNR_ATTRCOL$
LOGMNR_TS$
LOGMNR_IND$
LOGMNR_USER$
LOGMNR_TABPART$
LOGMNR_TABSUBPART$
LOGMNR_TABCOMPART$
LOGMNR_TYPE$
LOGMNR_COLTYPE$

TABLE_NAME
------------------------------
LOGMNR_ATTRIBUTE$
LOGMNR_LOB$
LOGMNR_CDEF$
LOGMNR_CCOL$
LOGMNR_ICOL$
LOGMNR_LOBFRAG$
LOGMNR_INDPART$
LOGMNR_INDSUBPART$
LOGMNR_INDCOMPART$
LOGMNR_LOGMNR_BUILDLOG
LOGMNR_NTAB$

TABLE_NAME
------------------------------
LOGMNR_OPQTYPE$
LOGMNR_SUBCOLTYPE$
LOGMNR_KOPM$
LOGMNR_PROPS$
LOGMNR_ENC$
LOGMNR_REFCON$
LOGMNR_PARTOBJ$
LOGMNRP_CTAS_PART_MAP
LOGSTDBY$APPLY_PROGRESS

163 rows selected.
```

7) Donner la description des attributs des tables THING et SUBSCRIBE (Exploiter la table USER_TAB_COLUMNS).

```
select table_name, column_name, data_type, data_length, nullable
from all_tab_columns
where owner = 'DBAIOT' and table_name in ('THING', 'SUBSCRIBE')
order by table_name, column_id;
```

```
TABLE_NAME                     COLUMN_NAME                    DATA_TYPE
                                                            DATA_LENGTH N
------------------------------ ------------------------------ ----------
---------------------------------------------- ----------- -
SUBSCRIBE                      IDUSER                         NUMBER
                                                                  22 N
SUBSCRIBE                      IDSERVICE                      NUMBER
                                                                  22 N
SUBSCRIBE                      DEBUT_DATE                     DATE
                                                                   7 Y
SUBSCRIBE                      FIN_DATE                       DATE
                                                                   7 Y
THING                          MAC                            CHAR
                                                                  17 N
THING                          IDUSER                         NUMBER
                                                                  22 Y
THING                          THINGTYPE                      VARCHAR2
                                                                  50 Y
THING                          PARAM                          NUMBER
                                                                  22 Y

8 rows selected.
```

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

```
select constraint_name, constraint_type from user_constraints where table_name in ('thing','service');
```

```
SQL> select constraint_name, constraint_type from user_constraints where table_name in ('THING','SERVICE');

CONSTRAINT_NAME                C
------------------------------ -
SYS_C007355                    P
SYS_C007356                    P
FK_IDUSER                      R
```

9) Donner toutes les contraintes créées lors de la partie et les informations qui les caractérisent (Exploitez la table USER_CONSTRAINTS);

```
select constraint_name,constraint_type,table_name,status,deferrable,deferred,validated
from  all_constraints
where owner = 'DBAIOT';
```

```
SQL> select constraint_name,constraint_type,table_name,status,deferrable,deferred,validated
  2  from  all_constraints
  3  where  owner = 'DBAIOT';

CONSTRAINT_NAME                C TABLE_NAME                     STATUS   DEFERRABLE     DEFERRED  VALIDATED
------------------------------ - ------------------------------ -------- -------------- --------- -------------
SYS_C007362                    C UTILISATEUR                    ENABLED  NOT DEFERRABLE IMMEDIATE VALIDATED
FK_IDUSER                      R THING                          ENABLED  NOT DEFERRABLE IMMEDIATE VALIDATED
FK_IDUSER2                     R SUBSCRIBE                      ENABLED  NOT DEFERRABLE IMMEDIATE VALIDATED
FK_IDSERVICE                   R SUBSCRIBE                      ENABLED  NOT DEFERRABLE IMMEDIATE VALIDATED
SYS_C007354                    P UTILISATEUR                    ENABLED  NOT DEFERRABLE IMMEDIATE VALIDATED
SYS_C007355                    P SERVICE                        ENABLED  NOT DEFERRABLE IMMEDIATE VALIDATED
SYS_C007356                    P THING                          ENABLED  NOT DEFERRABLE IMMEDIATE VALIDATED
PK_CLEPRIMAIRE                 P SUBSCRIBE                      ENABLED  NOT DEFERRABLE IMMEDIATE VALIDATED

8 rows selected.
```

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

```
--attribut+type
 Select column_name,  data_type,  data_length,  nullable From user_tab_columns
Where  owner = 'DBAIOT' and table_name = 'SUBSCRIBE';
```

```
 --contraint
select  constraint_name,  constraint_type, table_name, status, deferrable, deferred, validated
from user_constraints
where owner = 'DBAIOT' and table_name = 'SUBSCRIBE';
 --cle primaire
select column_name from all_cons_columns
where constraint_name = ( select  constraint_name  from all_constraints
  where owner = 'DBAIOT' and table_name = 'SUBSCRIBE' constraint_type = 'P');
```

```
COLUMN_NAME                    DATA_TYPE
                             DATA_LENGTH N
------------------------------ ----------------------
---------------- ----------- -
IDUSER                         NUMBER
                                    22 N
IDSERVICE                      NUMBER
                                    22 N
DEBUT_DATE                     DATE
                                     7 Y
FIN_DATE                       DATE
                                     7 Y
```

```
CONSTRAINT_NAME              C TABLE_NAME                   STATUS   DEFERRABLE     DEFERRED  VALIDATED
---------------------------- - ------------------------------ -------- -------------- --------- -------------
PK_CLEPRIMAIRE               P SUBSCRIBE                    ENABLED  NOT DEFERRABLE IMMEDIATE VALIDATED
FK_IDUSER2                   R SUBSCRIBE                    ENABLED  NOT DEFERRABLE IMMEDIATE VALIDATED
FK_IDSERVICE                 R SUBSCRIBE                    ENABLED  NOT DEFERRABLE IMMEDIATE VALIDATED
```

```
COLUMN_NAME
------------------------------------
------------------------------------
IDUSER
IDSERVICE
```

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

```
grant create session to admin;
grant create table to admin;
grant select on dbaiot.utilisateur to admin;
connect admin/adm
select * from user_tab_privs;
connect system/1234
select  * from dba_sys_privs where grantee = 'ADMIN';
```

```
SQL> connect admin/adm
Connected.
SQL> SELECT * FROM user_tab_privs
  2  ;

GRANTEE                        OWNER                          TABLE_NAME                     GRANTOR                        PRIVILEGE                                GRA
HIE
------------------------------ ------------------------------ ------------------------------ ------------------------------ ---------------------------------------- ---
---
ADMIN                          DBAIOT                         UTILISATEUR                    DBAIOT                         SELECT                                   NO
NO
```

```
SQL> SELECT * FROM dba_sys_privs WHERE grantee = 'ADMIN';

GRANTEE                        PRIVILEGE                                ADM
------------------------------ ---------------------------------------- ---
ADMIN                          CREATE TABLE                             NO
ADMIN                          CREATE SESSION                           NO
```

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

```
select *  from  dba_role_privs where  grantee = 'ADMIN';
```

```
SQL> connect system/1234
Connected.
SQL> select *  from  dba_role_privs where  grantee = 'ADMIN';

GRANTEE                        GRANTED_ROLE                   ADM DEF
------------------------------ ------------------------------ --- ---
ADMIN                          SUBSCRIBE_MANAGER              NO  YES
```

13) Trouver tous les objets appartenant à Admin.

```
select object_name, object_type from all_objects
where owner = 'ADMIN';
```

```
OBJECT_NAME                    OBJECT_TYPE
------------------------------ -------------------
USER_THING                     VIEW
NAMESERVICE_IX                 INDEX
```

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

```
connect admin/adm
select owner, table_name from all_tables
where table_name = 'SUBSCRIBE';
```

```
SQL> connect admin/adm
Connected.
SQL> select owner, table_name from all_tables
  2  where table_name = 'SUBSCRIBE';

OWNER                          TABLE_NAME
------------------------------ ----------------------------
DBAIOT                         SUBSCRIBE
```

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

```
connect system/1234
select segment_name,   segment_type,   bytes / 1024 as size_kb
from dba_segments
where segment_name = 'SUBSCRIBE' and owner = 'DBAIOT';
```

```
SQL> connect system/1234
Connected.
SQL> select segment_name,   segment_type,   bytes / 1024 as size_kb
  2  from dba_segments
  3  where segment_name = 'SUBSCRIBE' and owner = 'DBAIOT';

SEGMENT_NAME                                                                     SEGMENT_TYPE          SIZE_KB
-------------------------------------------------------------------------------- ------------------ ----------
SUBSCRIBE                                                                        TABLE                      64
```

16) Vérifier l'effet produit par chacune des commandes de définition de données de la partie 1 sur le dictionnaire :

Créez un nouvel utilisateur comme dans la partie 1, donner lui tous les privilèges ensuite connectez-vous avec cet utilisateur que vous venez de créer

```
connect system/1234;
create user newUser identified by saber;
grant connect , resource,dba to newUser;

connect newUser/saber
select * from all_users;
select * from user_sys_privs;
select * from user_tab_privs;
select * from user_role_privs;
select * from user_objects;
```

```
SQL> connect system/1234;
Connected.
SQL> create user newUser identified by saber;

User created.

SQL> grant connect , resource,dba to newUser;

Grant succeeded.
```

```
SQL> select * from all_users;

USERNAME                          USER_ID CREATED
------------------------------ ---------- ---------
XS$NULL                        2147483638 29-MAY-14
NEWUSER                                58 18-DEC-24
DBACOMPTOIRS                           52 05-NOV-24
MEE                                    51 21-OCT-24
APEX_040000                            47 29-MAY-14
APEX_PUBLIC_USER                       45 29-MAY-14
FLOWS_FILES                            44 29-MAY-14
HR                                     43 29-MAY-14
MDSYS                                  42 29-MAY-14
ANONYMOUS                              35 29-MAY-14
XDB                                    34 29-MAY-14

USERNAME                          USER_ID CREATED
------------------------------ ---------- ---------
CTXSYS                                 32 29-MAY-14
APPQOSSYS                              30 29-MAY-14
DBSNMP                                 29 29-MAY-14
ORACLE_OCM                             21 29-MAY-14
DIP                                    14 29-MAY-14
OUTLN                                   9 29-MAY-14
SYSTEM                                  5 29-MAY-14
SYS                                     0 29-MAY-14
ADMIN                                  56 17-DEC-24
DBAIOT                                 53 13-DEC-24
```

```
SQL> select * from user_sys_privs;

USERNAME                       PRIVILEGE                                ADM
------------------------------ ---------------------------------------- ---
NEWUSER                        UNLIMITED TABLESPACE                     NO
```

```
SQL> select * from user_sys_privs;

USERNAME                       PRIVILEGE                                ADM
------------------------------ ---------------------------------------- ---
NEWUSER                        UNLIMITED TABLESPACE                     NO

SQL> select * from user_tab_privs;

no rows selected

SQL> select * from user_role_privs;

USERNAME                       GRANTED_ROLE                   ADM DEF OS_
------------------------------ ------------------------------ --- --- ---
NEWUSER                        CONNECT                        NO  YES NO
NEWUSER                        DBA                            NO  YES NO
NEWUSER                        RESOURCE                       NO  YES NO

SQL> select * from user_objects;

no rows selected
```

Reference

- **Support de cours**
- **Site oracle** :https://www.oracle.com/fr/