Partie 1

On doit d'abord se connecter en tant que system.

```
sqlplus system/a2b48a3bfb254f04@//localhost:1521/XE
```

```
bash-4.2# sqlplus system/a2b48a3bfb254f04@//localhost:1521/XE

SQL*Plus: Release 18.0.0.0.0 - Production on Fri Dec 13 16:50:43 2024
Version 18.4.0.0.0

Copyright (c) 1982, 2018, Oracle.  All rights reserved.

Last Successful login time: Fri Dec 13 2024 16:43:04 +00:00

Connected to:
Oracle Database 18c Express Edition Release 18.0.0.0.0 - Production
Version 18.4.0.0.0

SQL>
```

1.a

Creation de la tablespace IOT_TBS

```
create tablespace IOT_TBS DATAFILE '\tablespace_Data.dat' size 100M autoextend on online;
```

```
SQL> create tablespace IOT_TBS DATAFILE '\tablespace_Data.dat' size 100M autoextend on
 online;

Tablespace created.

SQL>
```

1.b

Creation de la tablespace temporaire IOT_TempTBS

```
create temporary tablespace IOT_TempTBS TEMPFILE '\tablespace_Temp.dat' size 100M autoextend on;
```

```
SQL> create temporary tablespace IOT_TempTBS TEMPFILE '\tablespace_Temp.dat' size 100M
 autoextend on;

Tablespace created.

SQL>
```

2.

Création de l'utilisateur DBAIOT, en lui attribuant les tablespaces que nous avons créés et en définissant '1234' comme mot de passe.

```
alter session set "_oracle_script" = true;
create user DBAIOT identified by "1234" Default tablespace IOT_TBS temporary tablespace IOT_TempTBS;
```

```
SQL> alter session set "_oracle_script" = true;

Session altered.

SQL> create user DBAIOT identified by "1234" Default tablespace IOT_TBS temporary tabl
espace IOT_TempTBS;

User created.

SQL>
```

3.

On donne tous les droits et privilèges à l'utilisateur DBAIOT.

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

```
SQL> grant all privileges to DBAIOT;

Grant succeeded.

SQL>
```

Partie 2

On doit d'abord se connecter en tant que DBAIOT.

```
sqlplus dbaiot/1234@//localhost:1521/XE
```

```
bash-4.2# sqlplus dbaiot/1234@//localhost:1521/XE

SQL*Plus: Release 18.0.0.0.0 - Production on Fri Dec 13 17:30:12 2024
Version 18.4.0.0.0

Copyright (c) 1982, 2018, Oracle.  All rights reserved.


Connected to:
Oracle Database 18c Express Edition Release 18.0.0.0.0 - Production
Version 18.4.0.0.0

SQL>
```

1.a

Creation de la table users

```
create table users (
iduser integer,
lastname varchar2(50),
firstname varchar2(50),
email varchar2(50),
constraint pk_users primary key (iduser)
);
```

```
SQL> create table users (
iduser integer,
lastname varchar2(50),
firstname varchar2(50),
email varchar2(50),
constraint pk_users primary key (iduser)
);   2    3    4    5    6    7

Table created.

SQL>
```

1.b

Creation de la table service

```
create table service (
idservice integer,
name varchar2(50),
servicetype varchar2(50),
constraint pk_service primary key (idservice),
constraint chk_service_type check (servicetype in ('quantifiedself','smarthome'))
);
```

```
SQL> create table service (
idservice integer,
name varchar2(50),
servicetype varchar2(50),
constraint pk_service primary key (idservice),
constraint chk_service_type check (servicetype in ('quantifiedself','smarthome'))
);   2    3    4    5    6    7

Table created.

SQL>
```

1.c

Creation de la table thing

```
create table thing (
mac varchar2(20),
iduser integer,
thingtype varchar2(50),
param number,
constraint pk_thing primary key (mac),
constraint fk_thing_users foreign key (iduser) references users(iduser)
);
```

```
SQL> create table thing (
mac varchar2(20),
iduser integer,
thingtype varchar2(50),
param number,
constraint pk_thing primary key (mac),
constraint fk_thing_users foreign key (iduser) references users(iduser)
);   2    3    4    5    6    7    8

Table created.

SQL>
```

1.d

Creation de la table subscribe

```
create table subscribe (
iduser integer,
idservice integer,
constraint pk_subscribe primary key (iduser,idservice),
constraint fk_subscribe_users foreign key (iduser) references users(iduser),
constraint fk_subscribe_service foreign key (idservice) references service(idservice)
);
```

```
SQL> create table subscribe (
iduser integer,
idservice integer,
constraint pk_subscribe primary key (iduser,idservice),
constraint fk_subscribe_users foreign key (iduser) references users(iduser),
constraint fk_subscribe_service foreign key (idservice) references service(idservice)
);   2    3    4    5    6    7

Table created.

SQL>
```

2.

Ajoutter l'attribut adressuser de la table users

```
1 alter table users add (adressuser varchar2(100));
```

```
SQL> alter table users add (adressuser varchar2(100));

Table altered.

SQL>
```

3.a

Ajouter la contrainte not null pour l'attribut lastname de la table users

```
1 alter table users add constraint nn_users_lastname check (lastname is NOT null);
```

```
SQL> alter table users add constraint nn_users_lastname check (lastname is not null);

Table altered.

SQL>
```

3.b

Ajouter la contrainte not null pour l'attribut adressuser de la table users

```
1 alter table users add constraint nn_users_adressuser check (adressuser is NOT null);
```

```
SQL> alter table users add constraint nn_users_adressuser check (adressuser is NOT nul
l);

Table altered.

SQL>
```

4.a

Agrandir la taille de l'attribut adressuser de la table users

```
1 alter table users modify (adressuser varchar2(150));
```

```
SQL> alter table users modify (adressuser varchar2(150));

Table altered.

SQL>
```

4.b

Reduire la taille de l'attribut adressuser de la table users

```
alter table users modify (adressuser varchar2(90));
```

```
SQL> alter table users modify (adressuser varchar2(90));

Table altered.

SQL>
```

5.a

Renommer l'attribut adressuser de la table users a adruser

```
alter table users rename column adressuser to adruser;
```

```
SQL> alter table users rename column adressuser to adruser;

Table altered.
```

5.b

On verifie le changement du nom en utilisant desc pour afficher la structure de la table

```
desc users;
```

```
SQL> desc users;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 IDUSER                                    NOT NULL NUMBER(38)
 LASTNAME                                           VARCHAR2(50)
 FIRSTNAME                                          VARCHAR2(50)
 EMAIL                                              VARCHAR2(50)
 ADRUSER                                            VARCHAR2(90)

SQL>
```

6.a

Supprimer l'attribut adruser de la table users

```
alter table users drop column adruser;
```

```
SQL> alter table users drop column adruser;

Table altered.

SQL>
```

6.b

On verifie la suppression de l'attribut adruser utilisant desc pour afficher la structure de la table

```
desc users;
```

```
SQL> desc users;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 IDUSER                                    NOT NULL NUMBER(38)
 LASTNAME                                           VARCHAR2(50)
 FIRSTNAME                                          VARCHAR2(50)
 EMAIL                                              VARCHAR2(50)

SQL>
```

7.

Pour reprondre a ce besoin on doit ajouter les deux attributs date_fin et date_deb a la table subscribe

7.a

Ajouter l'attribut date_deb, supprimer l'ancienne contrainte pk_subscribe, puis la recréer en incluant date_deb dans la clé primaire de la table subscribe.

```
alter table subscribe add (date_deb date);
alter table subscribe drop constraint pk_subscribe;
alter table subscribe add constraint pk_subscribe primary key (iduser,idservice,date_deb);
```

```
SQL> alter table subscribe add (date_deb date);

Table altered.

SQL> alter table subscribe drop constraint pk_subscribe;

Table altered.

SQL> alter table subscribe add constraint pk_subscribe primary key (iduser,idservice,d
ate_deb);

Table altered.

SQL>
```

7.b

Ajouter l'attribut date_fin et ajouter une contrainte pour verifie que date_deb<date_fin

```
alter table subscribe add (date_fin date);
alter table subscribe add constraint chk_subscribe_date check(date_deb<date_fin);
```

```
SQL> alter table subscribe add (date_fin date);

Table altered.

SQL> alter table subscribe add constraint chk_subscribe_date check(date_deb<date_fin);

Table altered.

SQL>
```

Partie 3

1.a

Insertion dans la table users

```
insert all
    into users values(1,'Souad','MESBAH','souad.mesbah@gmail.com')
    into users values(2,'Younes','CHALAH','younes.chalah@gmail.com')
    into users values(3,'Chahinaz','MELEK','chahinaz.melek@gmail.com')
    into users values(4,'Samia','OUALI','samia.ouali@gmail.com')
    into users values(5,'Djamel','MATI','djamel.mati@gmail.com')
    into users values(6,'Assia','HORRA','assia.horra@gmail.com')
    into users values(7,'Lamine','MERABAT','Lamine.MERABAT@gmail.com')
    into users values(8,'Seddik','HMIA','seddik.hmia@gmail.com')
    into users values(9,'Widad','TOUATI','widad.touati@gmail.com')
select * from dual;
```

```
SQL> insert all
    into users values(1,'Souad','MESBAH','souad.mesbah@gmail.com')
    into users values(2,'Younes','CHALAH','younes.chalah@gmail.com')
    into users values(3,'Chahinaz','MELEK','chahinaz.melek@gmail.com')
    into users values(4,'Samia','OUALI','samia.ouali@gmail.com')
    into users values(5,'Djamel','MATI','djamel.mati@gmail.com')
    into users values(6,'Assia','HORRA','assia.horra@gmail.com')
    into users values(7,'Lamine','MERABAT','Lamine.MERABAT@gmail.com')
    into users values(8,'Seddik','HMIA','seddik.hmia@gmail.com')
    into users values(9,'Widad','TOUATI','widad.touati@gmail.com')
select * from dual;  2    3    4    5    6    7    8    9   10   11

9 rows created.

SQL>
```

1.b

Insertion dans la table service

```
insert all
    into service values(1,'myKWHome','smarthome')
    into service values(2,'FridgAlert','smarthome')
    into service values(3,'RUNstats','quantifiedself')
    into service values(4,'traCARE','quantifiedself')
    into service values(5,'dogWATCH',NULL)
    into service values(6,'CarUse',NULL)
select * from dual;
```

```
SQL> insert all
    into service values(1,'myKWHome','smarthome')
    into service values(2,'FridgAlert','smarthome')
    into service values(3,'RUNstats','quantifiedself')
    into service values(4,'traCARE','quantifiedself')
    into service value  2    3    4    5    6  s(5,'dogWATCH',NULL)
    into service values(6,'CarUse',NULL)
select * from dual;  7    8

6 rows created.

SQL>
```

1.c

Insertion dans la table thing

```
insert all
    into thing values('f0:de:f1:39:7f:17',1,NULL,NULL)
    into thing values('f0:de:f1:39:7f:18',2,NULL,NULL)
    into thing values('f0:de:f1:39:7f:19',2,'thingtempo',60)
    into thing values('f0:de:f1:39:7f:25',10,NULL,NULL)
    into thing values('f0:de:f1:39:7f:20',2,'thingtempo',1.5)
    into thing values('f0:de:f1:39:7f:21',4,NULL,NULL)
    into thing values('f0:de:f1:39:7f:22',4,NULL,NULL)
select * from dual;
```

```
SQL> insert all
    into thing values('f0:de:f1:39:7f:17',1,NULL,NULL)
    into thing values('f0:de:f1:39:7f:18',2,NULL,NULL)
    into thing values('f0:de:f1:39:7f:19',2,'thingtempo',60)
    into thing values('f0:de:f1:39:7f:25',10,NULL,NULL)
    into thing values('f0:de:f1:39:7f:20',2,'thingtempo',1.5)
    into thing values('f0:de:f1:39:7f:21',4,NULL,NULL)
    into thing values('f0:de:f1:39:7f:22',4,NULL,NULL)
select * from dual;  2    3    4    5    6    7    8    9
insert all
*
ERROR at line 1:
ORA-02291: integrity constraint (DBAIOT.FK_THING_USERS) violated - parent key
not found


SQL>
```

Le problème rencontré est qu'on a enfreint la contrainte FK_THING_USERS, car iduser = 10 n'existe pas dans la table users.

1.d

Insertion dans la table subscribe

```
insert all
    into subscribe values(2,1,to_date('2020/01/12','YYYY/MM/DD'),to_date('2020/02/12','YYYY/MM/DD'))
    into subscribe values(2,2,to_date('2020/01/19','YYYY/MM/DD'),to_date('2020/02/07','YYYY/MM/DD'))
    into subscribe values(1,3,to_date('2020/02/05','YYYY/MM/DD'),to_date('2020/04/05','YYYY/MM/DD'))
    into subscribe values(3,7,to_date('2020/02/01','YYYY/MM/DD'),to_date('2020/02/15','YYYY/MM/DD'))
select * from dual;
```

```
SQL> insert all
    into subscribe values(2,1,to_date('2020/01/12','YYYY/MM/DD'),to_date('2020/02/12','YYYY/MM/DD'))
    into subscribe values(2,2,to_date('2020/01/19','YYYY/MM/DD'),to_date('2020/02/07','YYYY/MM/DD'))
    into subscribe values(1,3,to_date('2020/02/05','YYYY/MM/DD'),to_date('2020/04/05','YYYY/MM/DD'))
    into subscribe values(3,7,to_date('2020/02/01','YYYY/MM/DD'),to_date('2020/02/15','YYYY/MM/DD'))
select * from dual;  2    3    4    5    6
insert all
*
ERROR at line 1:
ORA-02291: integrity constraint (DBAIOT.FK_SUBSCRIBE_SERVICE) violated - parent
key not found


SQL>
```

Le problème rencontré est qu'on a enfreint la contrainte FK_SUBSCRIBE_SERVICE, car idservice = 7 n'existe pas dans la table service.

Partie 4

1.

Création de l'utilisateur Admin, en lui attribuant les tablespaces que nous avons créés et en définissant '1234' comme mot de passe.

```
alter session set "_oracle_script" = true;
create user Admin identified by "1234" Default tablespace IOT_TBS temporary tablespace IOT_TempTBS;
```

```
SQL> alter session set "_oracle_script" = true;

Session altered.

SQL> create user Admin identified by "1234" Default tablespace IOT_TBS temporary table
space IOT_TempTBS;

User created.

SQL>
```

2.

On se connecte en tant que admin.

```
sqlplus admin/1234@//localhost:1521/XE
```

```
bash-4.2# sqlplus admin/1234@//localhost:1521/XE

SQL*Plus: Release 18.0.0.0.0 - Production on Fri Dec 13 23:53:03 2024
Version 18.4.0.0.0

Copyright (c) 1982, 2018, Oracle.  All rights reserved.

ERROR:
ORA-01045: user ADMIN lacks CREATE SESSION privilege; logon denied


Enter user-name:
```

Il est impossible de se connecter en tant qu'admin car cet utilisateur ne dispose pas des privilèges nécessaires pour créer une session.

3.a

On donne le droit de créer une session a admin depuis DBAIOT

```
grant create session to admin;
```

```
SQL> grant create session to admin;

Grant succeeded.

SQL>
```

3.b

On se connecte en tant que admin.

```
sqlplus admin/1234@//localhost:1521/XE
```

```
bash-4.2# sqlplus admin/1234@//localhost:1521/XE

SQL*Plus: Release 18.0.0.0.0 - Production on Sat Dec 14 00:05:22 2024
Version 18.4.0.0.0

Copyright (c) 1982, 2018, Oracle.  All rights reserved.


Connected to:
Oracle Database 18c Express Edition Release 18.0.0.0.0 - Production
Version 18.4.0.0.0

SQL>
```

4.a

On donne le droit de créer des tables et utilisateurs a admin depuis DBAIOT

```
grant create table , create user to admin;
```

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

Grant succeeded.

SQL>
```

4.b

Pour verifie les droit de admin on va afficher la table USER_SYS_PRIVS

```
SELECT * FROM USER_SYS_PRIVS;
```

```
SQL> SELECT * FROM USER_SYS_PRIVS;

USERNAME
--------------------------------------------------------------------------------
PRIVILEGE                                ADM COM INH
---------------------------------------- --- --- ---
ADMIN
CREATE TABLE                             NO  YES NO

ADMIN
CREATE SESSION                           NO  YES NO

ADMIN
CREATE USER                              NO  YES NO


SQL>
```

Remarque

La table USER_SYS_PRIVS affiche les privilèges du user avec lequel on est connecté. La colonne ADM (admin) indique que l'utilisateur peut accorder ou révoquer ce privilège, COM (command) signifie que l'utilisateur a droit à ce privilège, et INH (inherit) indique que l'utilisateur a obtenu ce privilège indirectement via un role.

5.

On va executer **Q1** depuis admin

```
Select * from DBAIOT.USERS;
```

```
SQL> Select * from DBAIOT.USERS;
Select * from DBAIOT.USERS
                     *
ERROR at line 1:
ORA-00942: table or view does not exist


SQL>
```

Le problem est que l'utilisateur admin n'a pas le privilege de selectionner la table DBAIOT.USERS

6.a

On donne le privelege de selectionner la table DBAIOT.USERS a admin depuis DBAIOT On va executer **Q1** depuis admin

```
grant select on DBAIOT.USERS to admin;
```

```
SQL> grant select on DBAIOT.USERS to admin;

Grant succeeded.

SQL>
```

6.b

On va executer **Q1** depuis admin

```
Select * from DBAIOT.USERS;
```

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

    IDUSER LASTNAME
---------- --------------------------------------------------
FIRSTNAME
--------------------------------------------------
EMAIL
--------------------------------------------------
         1 Souad
MESBAH
souad.mesbah@gmail.com

         2 Younes
CHALAH
younes.chalah@gmail.com

    IDUSER LASTNAME
---------- --------------------------------------------------
FIRSTNAME
--------------------------------------------------
EMAIL
--------------------------------------------------

         3 Chahinaz
MELEK
chahinaz.melek@gmail.com

         4 Samia
OUALI

    IDUSER LASTNAME
---------- --------------------------------------------------
FIRSTNAME
--------------------------------------------------
EMAIL
```

7.

On va créer la vue USER_THING avec une jointure entre les tables DBAIOT.USERS et DBAIOT.THING depuis l'utilisateur admin

```
create view USER_THING(iduser,objet) as select us.iduser,th.mac from
DBAIOT.USERS us , DBAIOT.THING th where th.iduser=us.iduser
group by us.iduser,th.mac;
```

```
SQL> create view USER_THING(iduser,objet) as select us.iduser,th.mac from
DBAIOT.USERS us , DBAIOT.THING th where th.iduser=us.iduser
group by us.iduser,th.mac;  2    3
DBAIOT.USERS us , DBAIOT.THING th where th.iduser=us.iduser
                  *
ERROR at line 2:
ORA-00942: table or view does not exist


SQL>
```

Le problem est que l'admin n'a pas le privelege de select la table DBAIOT.THING ni le privelge de creer des views

8.a

On donne le droit de créer des view et selectionner la table DBAIOT.THING a admin depuis DBAIOT

```
grant create view to admin;
grant select on DBAIOT.THING to admin;
```

```
SQL> grant create view to admin;

Grant succeeded.

SQL> grant select on DBAIOT.THING to admin;

Grant succeeded.

SQL>
```

8.b

Creation de la vue USER_THING depuis admin

```
1 create view USER_THING(iduser,objet) as select us.iduser,th.mac from
2 DBAIOT.USERS us , DBAIOT.THING th where th.iduser=us.iduser
3 group by us.iduser,th.mac;
```

```
SQL> create view USER_THING(iduser,objet) as select us.iduser,th.mac from

DBAIOT.USERS us , DBAIOT.THING th where th.iduser=us.iduser
group by us.iduser,th.mac;  2    3

View created.

SQL>
```

9.

On va créer l'index NAMESERVICE_IX sur l'attribut NAME de la table DBAIOT.SERVICE depuis l'utilisateur admin

```
1 create index NAMESERVICE_IX on DBAIOT.SERVICE(name);
```

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


SQL>
```

Le problem est que l'admin n'a pas le privelege de select la table DBAIOT.SERVICE ni le privelge de creer des indexes

10.a

On donne le droit de créer des indexes et selectionner la table DBAIOT.SERVICE a admin depuis DBAIOT

```
grant create any index to admin;
grant select on DBAIOT.SERVICE to admin;
```

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

Grant succeeded.

SQL> grant select on DBAIOT.SERVICE to admin;

Grant succeeded.

SQL>
```

10.b

Creation de l'index NAMESERVICE_IX depuis admin

```
create index NAMESERVICE_IX on DBAIOT.SERVICE(name);
```

```
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'


SQL>
```

Remarque

- Admin ne peut pas creer l'index NAMESERVICE_IX puisqu'il n'a pas les privileges pour ecrire sur la tablespace IOT_TBS
- Admin a pu creer la vue externe USER_THING car les views externes sont stockes dans la tablespace system

11.

revoker tous les droits qu'on a donner a admin depuis DBAIOT

```
revoke create view from admin;
revoke create any index from admin;
revoke create user from admin;
revoke create table from admin;
revoke select on DBAIOT.USERS from admin;
revoke select on DBAIOT.SERVICE from admin;
revoke select on DBAIOT.THING from admin;
revoke create session from admin;
```

```
SQL> revoke create view from admin;
Revoke succeeded.
SQL> revoke create any index from admin;
Revoke succeeded.
SQL> revoke create user from admin;
Revoke succeeded.
SQL> revoke create table from admin;
Revoke succeeded.
SQL> revoke select on DBAIOT.USERS from admin;
Revoke succeeded.
SQL> revoke select on DBAIOT.SERVICE from admin;
Revoke succeeded.
SQL> revoke select on DBAIOT.THING from admin;
Revoke succeeded.
SQL> revoke create session from admin;
Revoke succeeded.
SQL>
```

12.

On verifie que tous les droits d'admin on ete revoke en selectionnant la table ALL_TAB_PRIVS avec garentee='ADMIN' depuis DBAIOT puisque admin n'a plus le droit de creer une session

```
SELECT * FROM ALL_TAB_PRIVS where grantee='ADMIN';
```

```
SQL> SELECT * FROM ALL_TAB_PRIVS where grantee='ADMIN';
no rows selected
SQL>
```

13.

Création du profil IOT_Profil, en lui attribuant les limits donner.

```
create profile C##IOT_Profil limit
  sessions_per_user 3               -- 3 sessions simultanees autorisees par utilisateur
  cpu_per_call 35                   -- Un appel systeme ne peut pas consommer plus de 35 secondes de CPU
  connect_time 5400                 -- La session ne peut exceder 90 minutes (5400 secondes)
  logical_reads_per_call 1200       -- Un appel systeme ne peut lire plus de 1200 blocs de donnees
  private_sga 25k                   -- Chaque session ne peut allouer plus de 25 Ko en memoire SGA
  idle_time 30                      -- I'inactivite maximale de 30 minutes avant deconnexion de la session
  failed_login_attempts 5           -- 5 tentatives de connexion echouees avant verrouillage du compte
  password_life_time 50             -- Le mot de passe est valable pendant 50 jours
  password_reuse_time 40            -- Il faut attendre 40 jours avant qu'un mot de passe puisse etre reutilise
  password_grace_time 5             -- 5 jours de periode de grace avant que le mot de passe doive etre change
  password_lock_time 1/24;          -- 1 jour d'interdiction d'acces apres avoir atteint 5 tentatives de connexion
      echouees
```

```
SQL> create profile C##IOT_Profil limit
  sessions_per_user 3
  cpu_per_call 35
  connect_time 540
  logical_reads_per_call 1200
  private_sga 25k
   2    3    4    5    6    7   idle_time 30
  failed_login_attempts 5
  password_life_time 50
  password_reuse_time 40
  password_grace_time 5
  password_lock_time 1/24;   8    9   10   11   12

Profile created.
```

> **Remarque**
>
> J'ai ajouté le préfixe **C##** parce que je suis dans le conteneur racine **CDB$ROOT** de la base de données XE.

14.

On Affecte le profile IOT_PROFIL a l'utilisateur admin

```
1 alter user admin profile C##IOT_Profil;
```

```
SQL> alter user admin profile C##IOT_Profil;

User altered.
```

15.

Creation du role SUBSCRIBE_MANAGER depuis dbaiot en lui donnant les droits demandes

```
1 alter session set "_oracle_script" = true;
2 create role C##SUBSCRIBE_MANAGER;
3 grant select on DBAIOT.SERVICE to C##SUBSCRIBE_MANAGER;
4 grant select on DBAIOT.USERS to C##SUBSCRIBE_MANAGER;
5 grant update on DBAIOT.SUBSCRIBE to C##SUBSCRIBE_MANAGER;
```

```
SQL> alter session set "_oracle_script" = true;
Session altered.
SQL> create role C##SUBSCRIBE_MANAGER;
Role created.
SQL> grant select on DBAIOT.SERVICE to C##SUBSCRIBE_MANAGER;
Grant succeeded.
SQL> grant select on DBAIOT.USERS to C##SUBSCRIBE_MANAGER;
Grant succeeded.
SQL> grant update on DBAIOT.SUBSCRIBE to C##SUBSCRIBE_MANAGER;
Grant succeeded.
SQL>
```

16.a

On affecte le role SUBSCRIBE_MANAGER a l'utilisateur admin

```
grant C##SUBSCRIBE_MANAGER to admin;
```

```
SQL> grant C##SUBSCRIBE_MANAGER to admin;
Grant succeeded.
SQL>
```

16.b

les role des utilisteur sont stockes dans les table USER_ROLE_PRIVS et DBA_ROLE_PRIVS , probleme et que admin ne peut pas creer de session donc on ne peut pas se connecter en tant que admin et afficher USER_ROLE_PRIVS , et dbaiot n'est pas DBA pour acceder a DBA_ROLE_PRIVS , donc on se connect en tant que sysdba et afficher DBA_ROLE_PRIVS avec WHERE GRANTEE = 'ADMIN' pour afficher just celle d'admin

```
sqlplus sys/a2b48a3bfb254f04@//localhost:1521/XE as sysdba
SELECT * FROM DBA_ROLE_PRIVS WHERE GRANTEE = 'ADMIN';
```

```
bash-4.2# sqlplus sys/a2b48a3bfb254f04@//localhost:1521/XE as sysdba

SQL*Plus: Release 18.0.0.0.0 - Production on Sun Dec 15 02:16:09 2024
Version 18.4.0.0.0

Copyright (c) 1982, 2018, Oracle.  All rights reserved.


Connected to:
Oracle Database 18c Express Edition Release 18.0.0.0.0 - Production
Version 18.4.0.0.0

SQL> SELECT * FROM DBA_ROLE_PRIVS WHERE GRANTEE = 'ADMIN';

GRANTEE
--------------------------------------------------------------------------------
GRANTED_ROLE
--------------------------------------------------------------------------------
ADM DEL DEF COM INH
--- --- --- --- ---
ADMIN
C##SUBSCRIBE_MANAGER
NO  NO  YES YES NO
```

Depuis l'output on remarque que admin a le role C##SUBSCRIBE_MANAGER

Partie 5

1.a

On se connect en tant que system

```
sqlplus system/a2b48a3bfb254f04@//localhost:1521/XE
```

```
bash-4.2# sqlplus system/a2b48a3bfb254f04@//localhost:1521/XE

SQL*Plus: Release 18.0.0.0.0 - Production on Sun Dec 15 02:47:28 2024
Version 18.4.0.0.0

Copyright (c) 1982, 2018, Oracle.  All rights reserved.

Last Successful login time: Sun Dec 15 2024 02:44:53 +00:00

Connected to:
Oracle Database 18c Express Edition Release 18.0.0.0.0 - Production
Version 18.4.0.0.0

SQL>
```

1.b

On affiche la structure de la table dict en utilisant desc

```
desc dict;
```

```
SQL> desc dict;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 TABLE_NAME                                         VARCHAR2(128)
 COMMENTS                                           VARCHAR2(4000)

SQL>
```

1.c

On affiche la table dict , elle a 4623 instances

```
select * from dict;
```

```
TABLE_NAME
--------------------------------------------------------------------------------
COMMENTS
--------------------------------------------------------------------------------
CAT
Synonym for USER_CATALOG

CLU
Synonym for USER_CLUSTERS

DICT
Synonym for DICTIONARY


TABLE_NAME
--------------------------------------------------------------------------------
COMMENTS
--------------------------------------------------------------------------------
CDB_XS_ENB_AUDIT_POLICIES
Synonym for CDB_XS_ENABLED_AUDIT_POLICIES

EXT_TO_OBJ
Synonym for EXT_TO_OBJ_VIEW


4623 rows selected.

SQL>
```

2.a

On affiche la structure des tables ALL_TAB_COLUMNS, USER_USERS, ALL_CONSTRAINTS et USER_TAB_PRIVS avec desc

```
desc ALL_TAB_COLUMNS;
desc USER_USERS;
desc ALL_CONSTRAINTS;
desc USER_TAB_PRIVS;
```

```
SQL> desc ALL_TAB_COLUMNS;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 OWNER                                     NOT NULL VARCHAR2(128)
 TABLE_NAME                                NOT NULL VARCHAR2(128)
 COLUMN_NAME                               NOT NULL VARCHAR2(128)
 DATA_TYPE                                          VARCHAR2(128)
 DATA_TYPE_MOD                                      VARCHAR2(3)
 DATA_TYPE_OWNER                                    VARCHAR2(128)
 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(2000)
 HIGH_VALUE                                         RAW(2000)
 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)
 DEFAULT_ON_NULL                                    VARCHAR2(3)
 IDENTITY_COLUMN                                    VARCHAR2(3)
 EVALUATION_EDITION                                 VARCHAR2(128)
 UNUSABLE_BEFORE                                    VARCHAR2(128)
 UNUSABLE_BEGINNING                                 VARCHAR2(128)
 COLLATION                                          VARCHAR2(100)
```

```
SQL> desc USER_USERS;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 USERNAME                                  NOT NULL VARCHAR2(128)
 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)
 LOCAL_TEMP_TABLESPACE                              VARCHAR2(30)
 CREATED                                   NOT NULL DATE
 INITIAL_RSRC_CONSUMER_GROUP                        VARCHAR2(128)
 EXTERNAL_NAME                                      VARCHAR2(4000)
 PROXY_ONLY_CONNECT                                 VARCHAR2(1)
 COMMON                                             VARCHAR2(3)
 ORACLE_MAINTAINED                                  VARCHAR2(1)
 INHERITED                                          VARCHAR2(3)
 DEFAULT_COLLATION                                  VARCHAR2(100)
 IMPLICIT                                           VARCHAR2(3)
 ALL_SHARD                                          VARCHAR2(3)

SQL>
```

```
SQL> desc ALL_CONSTRAINTS;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 OWNER                                              VARCHAR2(128)
 CONSTRAINT_NAME                           NOT NULL VARCHAR2(128)
 CONSTRAINT_TYPE                                    VARCHAR2(1)
 TABLE_NAME                                NOT NULL VARCHAR2(128)
 SEARCH_CONDITION                                   LONG
 SEARCH_CONDITION_VC                                VARCHAR2(4000)
 R_OWNER                                            VARCHAR2(128)
 R_CONSTRAINT_NAME                                  VARCHAR2(128)
 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(128)
 INDEX_NAME                                         VARCHAR2(128)
 INVALID                                            VARCHAR2(7)
 VIEW_RELATED                                       VARCHAR2(14)
 ORIGIN_CON_ID                                      NUMBER

SQL>
```

```
SQL> desc USER_TAB_PRIVS;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 GRANTEE                                            VARCHAR2(128)
 OWNER                                              VARCHAR2(128)
 TABLE_NAME                                         VARCHAR2(128)
 GRANTOR                                            VARCHAR2(128)
 PRIVILEGE                                          VARCHAR2(40)
 GRANTABLE                                          VARCHAR2(3)
 HIERARCHY                                          VARCHAR2(3)
 COMMON                                             VARCHAR2(3)
 TYPE                                               VARCHAR2(24)
 INHERITED                                          VARCHAR2(3)

SQL>
```

2.b

Role

- **ALL_TAB_COLUMNS** : stocke les colonnes de toutes les vues et tables accessibles à chaque utilisateur.
- **USER_USERS** : contient une description de l'utilisateur actuellement connecté.
- **ALL_CONSTRAINTS** : contient toutes les contraintes accessibles à chaque utilisateur.
- **USER_TAB_PRIVS** : contient les privilèges objets des tables de l'utilisateur actuellement connecté.

3.

On va utiliser la table USER_USERS pour afficher le nom d'utilisatuer avec lequelle on s'est connecter

```
1 select username from user_users;
```

```
SQL> select username from user_users;

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

4.

On affiche la structure des tables ALL_TAB_COLUMNS,USER_TAB_COLUMNS avec desc

```
1 desc all_tab_columns;
2 desc user_tab_columns;
```

```
SQL> desc ALL_TAB_COLUMNS;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 OWNER                                     NOT NULL VARCHAR2(128)
 TABLE_NAME                                NOT NULL VARCHAR2(128)
 COLUMN_NAME                               NOT NULL VARCHAR2(128)
 DATA_TYPE                                          VARCHAR2(128)
 DATA_TYPE_MOD                                      VARCHAR2(3)
 DATA_TYPE_OWNER                                    VARCHAR2(128)
 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(2000)
 HIGH_VALUE                                         RAW(2000)
 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)
 DEFAULT_ON_NULL                                    VARCHAR2(3)
 IDENTITY_COLUMN                                    VARCHAR2(3)
 EVALUATION_EDITION                                 VARCHAR2(128)
 UNUSABLE_BEFORE                                    VARCHAR2(128)
 UNUSABLE_BEGINNING                                 VARCHAR2(128)
 COLLATION                                          VARCHAR2(100)
```

```
SQL> desc user_tab_columns;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 TABLE_NAME                                NOT NULL VARCHAR2(128)
 COLUMN_NAME                               NOT NULL VARCHAR2(128)
 DATA_TYPE                                          VARCHAR2(128)
 DATA_TYPE_MOD                                      VARCHAR2(3)
 DATA_TYPE_OWNER                                    VARCHAR2(128)
 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(2000)
 HIGH_VALUE                                         RAW(2000)
 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)
 DEFAULT_ON_NULL                                    VARCHAR2(3)
 IDENTITY_COLUMN                                    VARCHAR2(3)
 EVALUATION_EDITION                                 VARCHAR2(128)
 UNUSABLE_BEFORE                                    VARCHAR2(128)
 UNUSABLE_BEGINNING                                 VARCHAR2(128)
 COLLATION                                          VARCHAR2(100)
```

Difference

les deux tables ont les memes attributes , la seul difference est que la table ALL_TAB_COLUMNS a un attribut owner en plus compare a USER_TAB_COLUMNS

5.

On doit d'abord se connecter en tant que DBAIOT, puis afficher l'attribut `table_name` de la table `USER_TABLES`, qui contient les tables appartenant à l'utilisateur actuellement connecté.

```
1 sqlplus dbaiot/1234@//localhost:1521/XE
2 select table_name name from user_tables;
```

```
bash-4.2# sqlplus dbaiot/1234@//localhost:1521/XE

SQL*Plus: Release 18.0.0.0.0 - Production on Sun Dec 15 06:29:48 2024
Version 18.4.0.0.0

Copyright (c) 1982, 2018, Oracle.  All rights reserved.

Last Successful login time: Sun Dec 15 2024 02:26:05 +00:00

Connected to:
Oracle Database 18c Express Edition Release 18.0.0.0.0 - Production
Version 18.4.0.0.0

SQL> select table_name name from user_tables;

NAME
--------------------------------------------------------------------------------
SERVICE
USERS
THING
SUBSCRIBE

SQL>
```

6.

On va afficher les tables de system et de DBAIOT depuis user system en utilisant la table ALL_TABLES qui contient les tables de chaque user et en ajoute un where clause avec owner = 'SYSTEM' pour afficher les tables du user system, puis where owner = 'DBAIOT' pour les tables DBAIOT

6.a

```
1 select table_name name from all_tables where owner = 'SYSTEM';
```

```
NAME
--------------------------------------------------------------------------------
LOGMNR_TABPART$
LOGMNR_TABSUBPART$
LOGMNR_TABCOMPART$
LOGMNR_TYPE$
LOGMNR_COLTYPE$
LOGMNR_ATTRIBUTE$
LOGMNR_LOB$
LOGMNR_CON$
LOGMNR_CONTAINER$
LOGMNR_CDEF$
LOGMNR_CCOL$

NAME
--------------------------------------------------------------------------------
LOGMNR_ICOL$
LOGMNR_LOBFRAG$
LOGMNR_INDPART$
LOGMNR_INDSUBPART$
LOGMNR_INDCOMPART$
LOGMNR_LOGMNR_BUILDLOG
LOGMNR_NTAB$
LOGMNR_OPQTYPE$
LOGMNR_SUBCOLTYPE$
LOGMNR_KOPM$
LOGMNR_PROPS$

NAME
--------------------------------------------------------------------------------
LOGMNR_ENC$
LOGMNR_REFCON$
LOGMNR_IDNSEQ$
LOGMNR_PARTOBJ$
LOGMNRP_CTAS_PART_MAP
LOGMNR_SHARD_TS
LOGSTDBY$APPLY_PROGRESS

128 rows selected.
```

6.b

```
1 select table_name name from all_tables where owner = 'DBAIOT';
```

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

NAME
--------------------------------------------------------------------------------
SERVICE
USERS
THING
SUBSCRIBE

SQL>
```

7.

On va afficher les attributs des tables thing et subscribe en utilisant la table USER_TAB_COLUMNS depuis DBAIOT , en va afficher le nom , id , type , longeur de chaque attributs et on fait un WHERE table_name='THING' pour afficher les attributs de la table THING puis WHERE table_name='SUBSCRIBE' pour les attributs de la table SUBSCRIBE

7.a

```
select column_name col ,
data_type  type,
data_length len,
nullable n ,
column_id id
from user_tab_columns where table_name='THING';
```

```
SQL> select column_name col ,
data_type  type,
data_length len,
nullable n ,
column_id id
from user_tab_columns where table_name='THING';
  2    3    4    5    6
COL
--------------------------------------------------------------------------------
TYPE
--------------------------------------------------------------------------------
       LEN N         ID
---------- - ----------
MAC
VARCHAR2
        20 N          1

IDUSER
NUMBER
        22 Y          2

COL
--------------------------------------------------------------------------------
TYPE
--------------------------------------------------------------------------------
       LEN N         ID
---------- - ----------

THINGTYPE
VARCHAR2
        50 Y          3

PARAM
NUMBER

COL
--------------------------------------------------------------------------------
TYPE
--------------------------------------------------------------------------------
       LEN N         ID
---------- - ----------
        22 Y          4


SQL>
```

7.b

```
select column_name col ,
data_type  type,
data_length len,
nullable n ,
column_id id
from user_tab_columns where table_name='SUBSCRIBE';
```

```
SQL> select column_name col ,
data_type  type,
data_length len,
nullable n ,
column_id id
from user_tab_columns where table_name='SUBSCRIBE';
  2    3    4    5    6
COL
--------------------------------------------------------------------------------
TYPE
--------------------------------------------------------------------------------
       LEN N         ID
---------- - ----------
IDUSER
NUMBER
        22 N          1

IDSERVICE
NUMBER
        22 N          2

COL
--------------------------------------------------------------------------------
TYPE
--------------------------------------------------------------------------------
       LEN N         ID
---------- - ----------

DATE_DEB
DATE
         7 N          3

DATE_FIN
DATE

COL
--------------------------------------------------------------------------------
TYPE
--------------------------------------------------------------------------------
       LEN N         ID
---------- - ----------
         7 Y          4
```

8.

Pour vérifier s'il y a une référence de la table `SUBSCRIBE` dans la table `THING`, nous allons effectuer une jointure de la table `USER_CONSTRAINTS` avec elle-même. Nous utilisons un *select imbriqué* ou la *sous-requête* récupère les noms des contraintes référencées de la table `THING`, c'est-à-dire les contraintes de clé primaire des table references par `THING`. La requête principale compte ensuite le nombre de contraintes de clé primaire dans la table `SUBSCRIBE` qui figurent parmi les contraintes obtenues dans la sous-requête. si count = 0 , pas de references sinon si 1 ya une reference

```
1 select count(constraint_name) from user_constraints where
2 table_name = 'SUBSCRIBE' and constraint_type='P'
3 and constraint_name in (select r_constraint_name from user_constraints
4 where table_name='THING' and constraint_type = 'R');
```

```
SQL> select count(constraint_name) from user_constraints where
table_name = 'SUBSCRIBE' and constraint_type='P'
and constraint_name in (select r_constraint_name from user_constraints
where table_name='THING' and constraint_type = 'R');  2    3    4

COUNT(CONSTRAINT_NAME)
----------------------
                     0

SQL>
```

Remarque

- Constraint_Type = 'R' : contrainte cle etrangere
- Constraint_Type = 'P' : contrainte cle primaire
- R_Constraint_Name : nom des contrainte cle primaire reference par la table

9.

Pour afficher les contraintes creer par DBAIOT on va utiliser la table USER_CONSTRAINTS , et afficher les noms des contraintes(constraint_name) , type (constraint_name) , les tables associe (table_name)

```
select constraint_name name,
constraint_type type,
table_name tab
from user_constraints;
```

```
NAME
--------------------------------------------------------------------------------
T
-
TAB
--------------------------------------------------------------------------------
PK_THING
P
THING

NN_USERS_LASTNAME
C
USERS

NAME
--------------------------------------------------------------------------------
T
-
TAB
--------------------------------------------------------------------------------

PK_SUBSCRIBE
P
SUBSCRIBE

CHK_SUBSCRIBE_DATE
C

NAME
--------------------------------------------------------------------------------
T
-
TAB
--------------------------------------------------------------------------------
SUBSCRIBE


11 rows selected.
```

10.a

On va afficher les id(column_id) , nom(column_name) , type de donnes(data_type) , nullabilite(nullable) , valeur par defaut(data_default) des attributs de la table SUBSCRIBE en utilisant la table USER_TAB_COLUMNS et bien sure le where pour just selectionner la table SUBSCRIBE where table_name='SUBSCRIBE'

```
select column_id,
column_name col,
data_type type,
nullable ,
data_default def
from user_tab_columns where table_name='SUBSCRIBE';
```

10.b

Pour obtenir le nom de la contrainte cle primaire et les nom de ces attributs on va faire une jointure entre USER_CONSTRIANTS et USER_CONS_COLUMNS , la premiere table va nous aider a filterer les contraintes pour just avoir des contraintes de type cle primaire t1.constraint_type = 'P' de la table SUBSCRIBE t1.table_name='SUBSCRIBE' puis on utilise la deuxieme table pour afficher les noms des attributs de la cle primaire t2.column_name avec la jointure t1.constraint_name=t2.constraint_name

```
select t1.constraint_name const , t2.column_name from
user_constraints t1,user_cons_columns t2 where
t1.table_name='SUBSCRIBE' and t1.constraint_type='P' and
t2.constraint_name = t1.constraint_name;
```

```
user_constraints t1,user_cons_columns t2 where
t1.table_name='SUBSCRIBE' and t1.constraint_type='P' and
t2.constraint_name = t1.constraint_name;  2    3    4

CONST
------------------------------------------------------------------------
-------
COLUMN_NAME
------------------------------------------------------------------------
-------
PK_SUBSCRIBE
IDUSER

PK_SUBSCRIBE
IDSERVICE

PK_SUBSCRIBE
DATE_DEB
```

10.c

Pour afficher les nom des contraintes cle etrangers et le nom de l'attribut , table reference , colonne reference on doit utiliser des imbriquation de select toujours avec les deux table USER_CONSTRIANTS et USER_CONS_COLUMNS , le outer select fait une jointure entre les tables et affiche le nom de la contrainte et son attribut en filtron le type pour foreign key t1.constraint_type = 'R' de la table SUBSCRIBE t1.table_name='SUBSCRIBE' , pour afficher la table reference on va selectioner le nom de la table de la contrainte cle primaire reference par la table SUBSCRIBE t1.r_constraint_name=t11.constraint_name depuis la table USER_CONSTRAINTS, pour affiche les colonnes reference en selectionne les nom des colonnes de la contrainte cle primaire reference par la table SUBSCRIBE de puis la table USER_CONS_COLUMNS t1.r_constraint_name=t22.constraint_name

```sql
select t1.constraint_name const , t2.column_name ,
(select t22.column_name from
user_cons_columns t22 where
t1.r_constraint_name=t22.constraint_name) ref_col,
(select table_name from user_constraints t11
where t1.r_constraint_name=t11.constraint_name) ref_table
from
user_constraints t1,user_cons_columns t2 where
t1.table_name='SUBSCRIBE' and t1.constraint_type='R' and
t2.constraint_name = t1.constraint_name;
```

```
SQL> select t1.constraint_name const , t2.column_name ,
(select t22.column_name from
user_cons_columns t22 where
t1.r_constraint_name=t22.constraint_name) ref_col,
(select table_name from user_constraints t11
where t1.r_constraint_name=t11.constraint_name) ref_table
from
user_constraints t1,user_cons_columns t2 where
t1.table_name='SUBSCRIBE' and t1.constraint_type='R' and
t2.constraint_name = t1.constraint_name;  2    3    4    5    6    7    8    9   10

CONST
------------------------------------------------------------------------------
COLUMN_NAME
------------------------------------------------------------------------------
REF_COL
------------------------------------------------------------------------------
REF_TABLE
------------------------------------------------------------------------------
FK_SUBSCRIBE_USERS
IDUSER
IDUSER
USERS


CONST
------------------------------------------------------------------------------
COLUMN_NAME
------------------------------------------------------------------------------
REF_COL
------------------------------------------------------------------------------
REF_TABLE
------------------------------------------------------------------------------
FK_SUBSCRIBE_SERVICE
IDSERVICE
IDSERVICE
SERVICE
```

10.d

Pour obtenir les nom des contraintes uniques et les nom des attributs associes on va faire une jointure entre USER_CONSTRIANTS et USER_CONS_COLUMNS , la premiere table va nous aider a filterer les contraintes pour just avoir des contraintes de type unique t1.constraint_type = 'U' de la table SUBSCRIBE t1.table_name='SUBSCRIBE' puis on utilise la deuxieme table pour afficher le nom de l'attribut unique t2.column_name avec la jointure t1.constraint_name=t2.constraint_name

```sql
select t1.constraint_name const , t2.column_name from
user_constraints t1,user_cons_columns t2 where
t1.table_name='SUBSCRIBE' and t1.constraint_type='U' and
t2.constraint_name = t1.constraint_name;
```

```
SQL> select t1.constraint_name const , t2.column_name from
user_constraints t1,user_cons_columns t2 where
t1.table_name='SUBSCRIBE' and t1.constraint_type='U' and
t2.constraint_name = t1.constraint_name;  2    3    4

no rows selected

SQL>
```

10.e

On va obtenir les nom des contraints check + leur condition avec la table USER_CONSTRAINTS , on selectionnant l'attribut constraint_name(nom de la contraint) , search_condition(condition du check) et pour avoir que les contraintes de la table subscribe de type check on utilise le where clause table_name='SUBSCRIBE' and constraint_type='C'(type check)

```
select t1.constraint_name const , t1.SEARCH_CONDITION from
user_constraints t1
where t1.table_name='SUBSCRIBE' and t1.constraint_type='C';
```

```
SQL> select t1.constraint_name const , t1.SEARCH_CONDITION from
user_constraints t1
where t1.table_name='SUBSCRIBE' and t1.constraint_type='C';  2    3

CONST
------------------------------------------------------------------------------
SEARCH_CONDITION
------------------------------------------------------------------------------
CHK_SUBSCRIBE_DATE
date_deb<date_fin


SQL>
```

11.a

On donne 2 privilegs system a admin(creation session,user) et 1 privilege objet (select on DBAIOT.USERS) depuis DBAIOT

```
--2 privileges system
grant create session to admin;
grant create user to admin;

--1 privilges objet
grant select on DBAIOT.USERS to admin;
```

```
SQL> grant create session to admin;

Grant succeeded.

SQL> grant create user to admin;

Grant succeeded.

SQL> grant select on DBAIOT.USERS to admin;

Grant succeeded.

SQL>
```

11.b

On se connect d'abord en tant qu'admin

```
sqlplus admin/1234@//localhost:1521/XE
```

Pour afficher les privileges on va utilise la table USER_TAB_PRIV pour les privileges objets et la table USER_SYS_PRIV pour les privileges system

```sql
--privileges objets
SELECT privilege priv,table_name tab FROM USER_TAB_PRIVS;

--privileges system
SELECT privilege priv FROM USER_SYS_PRIVS;
```

```
SQL> SELECT privilege priv,table_name tab FROM USER_TAB_PRIVS;

PRIV
----------------------------------------
TAB
--------------------------------------------------------------------------------
SELECT
USERS

INHERIT PRIVILEGES
ADMIN
```

```
SQL> SELECT privilege priv FROM USER_SYS_PRIVS;

PRIV
----------------------------------------
CREATE SESSION
CREATE USER

SQL>
```

Remarque

On remarque que admin a comme droits system :

- creation session
- creation user

Et comme droits object :

- select on DBAIOT.USERS

12.

Pour afficher les roles d'admin on va utiliser la table USER_ROLE_PRIVS et just afficher l'attribut granted_name (nom du role) depuis admin

```sql
select granted_role from user_role_privs;
```

```
SQL> select granted_role from user_role_privs;

GRANTED_ROLE
--------------------------------------------------------------------------------
C##SUBSCRIBE_MANAGER

SQL>
```

Remarque

Le role C##SUBSCRIBE_MANAGER a ete creer dans la partie4

13.

Pour afficher les objets dont ADMIN est propriétaire on va utiliser la table ALL_OBJECTS , selectionner l'attribut OBJECT_NAME et filtrer avec where owner='ADMIN' pour seulement avoir les objets d'ADMIN

```
1 select object_name name from all_objects
2 where owner='ADMIN';
```

```
SQL> select object_name name from all_objects
where owner='ADMIN';  2

NAME
--------------------------------------------------------------------------------
--------
USER_THING

SQL>
```

Remarque

L'objet USER_THING est une vue qu'on a creer dans la partie4

14.

Pour trouver le propriétaire de la table SUBSCRIBE, on va afficher l'attribut OWNER de la table ALL_OBJECTS et filtre avec where object_name='SUBSCRIBE' pour seulement avoir le propietaire de la table SUBSCRIBE

```
1 select owner from all_objects
2 where object_name='SUBSCRIBE';
```

```
SQL> select owner from all_objects
where object_name='SUBSCRIBE';  2

OWNER
------------------------------------------------------------------------------
-------
DBAIOT

SQL>
```

15.

Pour afficher la taille de la table SUBSCRIBE en KO , on va selectionner la table USER_SEGMENTS depuis DBAIOT , affichant l'attribut BYTES qui donne la taille du segment en byte(octet) en va le diviser sur 2^{10}(1024) pour converitre au KO et en filtre avec where segment_name='SUBSCRIBE' pour just avoir la taille de la table SUBSCRIBE

```
select bytes/1024 size_KO from user_segments where
segment_name='SUBSCRIBE';
```

```
SQL> select bytes/1024 size_KO from user_segments where
segment_name='SUBSCRIBE';  2

   SIZE_KO
----------
        64

SQL>
```

16.a

Creer utilisateur Test et lui donner tous les droits

```
alter session set "_oracle_script" = true;
create user TEST identified by "1234" Default tablespace IOT_TBS temporary tablespace IOT_TempTBS;
grant all privileges to test;
```

```
SQL> alter session set "_oracle_script" = true;

Session altered.

SQL> create user TEST identified by "1234" Default tablespace IOT_TBS tem
porary tablespace IOT_TempTBS;

User created.

SQL> grant all privileges to test;

Grant succeeded.

SQL>
```

16.b

Connecter en tant que Test

```
1 sqlplus test/1234@//localhost:1521/XE
```

```
bash-4.2# sqlplus test/1234@//localhost:1521/XE

SQL*Plus: Release 18.0.0.0.0 - Production on Mon Dec 16 10:43:11 2024
Version 18.4.0.0.0

Copyright (c) 1982, 2018, Oracle.  All rights reserved.


Connected to:
Oracle Database 18c Express Edition Release 18.0.0.0.0 - Production
Version 18.4.0.0.0

SQL>
```

16.c

On va afficher les tables USER_OBJECTS , USER_TAB_COLUMNS , USER_CONSTRAINTS avant et apres des query DDL

```
1 select object_name name from user_objects;
```

```
SQL> select object_name name from user_objects;

no rows selected

SQL>
```

```
select column_id,
column_name col,
data_type type,
nullable ,
data_default def
from user_tab_columns;
```

```
SQL> select column_id,
column_name col,
data_type type,
nullable ,
data_default def
from user_tab_columns;  2    3    4    5    6

no rows selected
```

```
select constraint_name name,
constraint_type type,
table_name tab
from user_constraints;
```

```
SQL> select constraint_name name,
constraint_type type,
table_name tab
from user_constraints;  2    3    4

no rows selected
```

Remarque

Puisque test n'a jamais creer de contraints , attribut, objets toute les tables renvoit 0 lignes

On va creer deux Table t1 et t2

```
create table t1 (id1 int);
create table t2 (id2 int);
```

```
SQL> create table t1 (id1 int);

Table created.

SQL> create table t2 (id2 int);

Table created.
```

On va afficher les object et attribut de Test

```
select object_name name from user_objects;
```

```
SQL> select object_name name from user_objects;

NAME
--------------------------------------------------------------------------------
--------

T2
T1
```

```
select column_id,
column_name col,
data_type type,
nullable ,
data_default def
from user_tab_columns;
```

```
SQL> select column_id,
column_name col,
data_type type,
nullable ,
data_default def
from user_tab_columns; 2   3   4   5   6

 COLUMN_ID
----------
COL
--------------------------------------------------------------------------------
TYPE
--------------------------------------------------------------------------------
N
-
DEF
--------------------------------------------------------------------------------
         1
ID2
NUMBER
 COLUMN_ID
----------
COL
--------------------------------------------------------------------------------
TYPE
--------------------------------------------------------------------------------
N
-
DEF
--------------------------------------------------------------------------------
Y


 COLUMN_ID
----------
COL
--------------------------------------------------------------------------------
TYPE
--------------------------------------------------------------------------------
N
-
DEF
--------------------------------------------------------------------------------
         1
ID1
NUMBER
 COLUMN_ID
----------
COL
--------------------------------------------------------------------------------
TYPE
--------------------------------------------------------------------------------
N
-
DEF
--------------------------------------------------------------------------------
Y
```

Remarque

On remarque apres la creation des deux tables , les tables USER_OBJECTS , USER_TAB_COLUMNS on ete mise a jour

On va ajouter une colonne a t1 puis afficher les attributes de Test

```
alter table t1 add (name varchar2(20) default 'admin');
```

```
SQL> alter table t1 add (name varchar2(20) default 'admin');

Table altered.
```

```
select column_id,
column_name col,
data_type type,
nullable ,
data_default def
from user_tab_columns;
```

```
 COLUMN_ID
----------
COL
--------------------------------------------------------------------------------
TYPE
--------------------------------------------------------------------------------
N
-
DEF
--------------------------------------------------------------------------------
         2
NAME
VARCHAR2
 COLUMN_ID
----------
COL
--------------------------------------------------------------------------------
TYPE
--------------------------------------------------------------------------------
N
-
DEF
--------------------------------------------------------------------------------
Y
'admin'
```

Remarque

On appercoie que l'attribut name a ete ajoute a la table USER_TAB_COLUMNS et avec sa valeur par defaut 'admin'

On va ajouter une contraint de cle primaire pour t1 et une unique pour t2

```
alter table t1 add constraint pk_t1 primary key (id1);
alter table t2 add constraint uq_t2 unique(id2);
```

```
SQL> alter table t1 add constraint pk_t1 primary key (id1);

Table altered.

SQL> alter table t2 add constraint uq_t2 unique(id2);

Table altered.
```

```
select column_id,
column_name col,
data_type type,
nullable ,
data_default def
from user_tab_columns;
```

```
SQL> select constraint_name name,
constraint_type type,
table_name tab
from user_constraints;  2    3    4

NAME
--------------------------------------------------------------------------------
T
-
TAB
--------------------------------------------------------------------------------
PK_T1
P
T1

UQ_T2
U
T2

NAME
--------------------------------------------------------------------------------
T
-
TAB
--------------------------------------------------------------------------------
```

Remarque

On remarque qu'apres avoir ajouter les contraintes la table USER_CONSTRAINTS a ete misajour