Rapport TP 2 ASGBD

Etudiante : Ferchichi Manel

Section : SII A

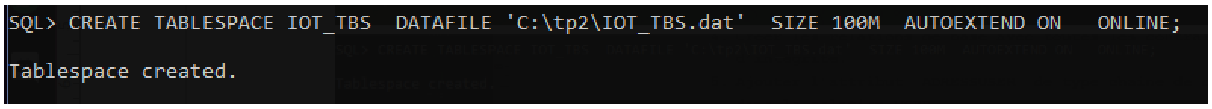
Groupe TP : 1

Partie I : Création des Tablespaces et des utilisateurs

1. **Créer deux Tablespaces   IOT_TBS et   IOT_TempTBS**

Requête :

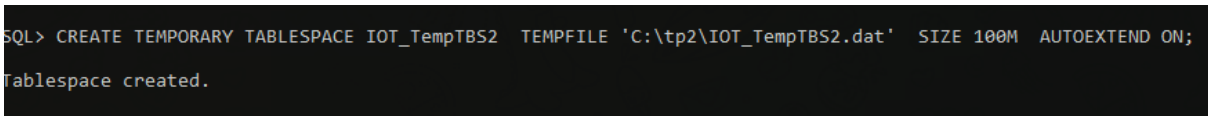
CREATE TABLESPACE **IOT_TBS** DATAFILE 'C:\tp2\**IOT_TBS**.dat' SIZE 100M AUTOEXTEND ON   ONLINE;



```
SQL> CREATE TABLESPACE IOT_TBS  DATAFILE 'C:\tp2\IOT_TBS.dat'  SIZE 100M  AUTOEXTEND ON   ONLINE;

Tablespace created.
```

Requête :

CREATE TEMPORARY TABLESPACE **IOT_TempTBS2** TEMPFILE 'C:\tp2\**IOT_TempTBS2**.dat' SIZE 100M AUTOEXTEND ON;

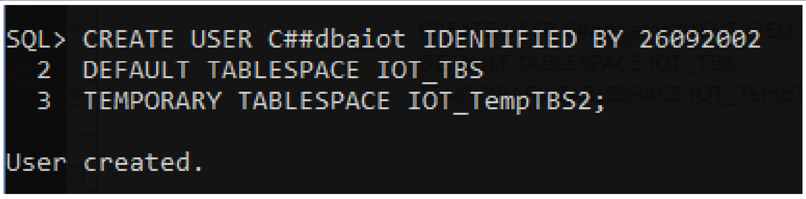


```
SQL> CREATE TEMPORARY TABLESPACE IOT_TempTBS2  TEMPFILE 'C:\tp2\IOT_TempTBS2.dat'  SIZE 100M  AUTOEXTEND ON;

Tablespace created.
```

2. **Créer un utilisateur dbaiot en lui attribuant les deux tablespaces créés précédemment**

Requête :

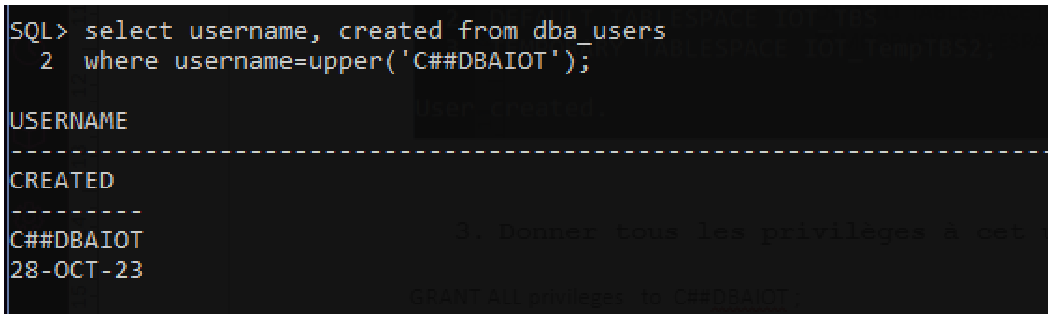
```
CREATE USER dbaiot IDENTIFIED BY 26092002
DEFAULT TABLESPACE IOT_TBS
TEMPORARY TABLESPACE IOT_TempTBS2;
```



```
SQL> CREATE USER C##dbaiot IDENTIFIED BY 26092002
  2  DEFAULT TABLESPACE IOT_TBS
  3  TEMPORARY TABLESPACE IOT_TempTBS2;

User created.
```

Requête :

```
SELECT USERNAME, CREATED FROM DBA_USERS WHERE USERNAME=UPPER('C##DBAIOT');
```

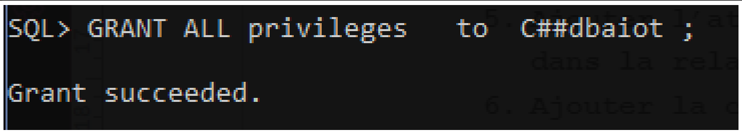


```
SQL> select username, created from dba_users
  2  where username=upper('C##DBAIOT');

USERNAME
--------------------------------------------------------------------------------
CREATED
---------
C##DBAIOT
28-OCT-23
```

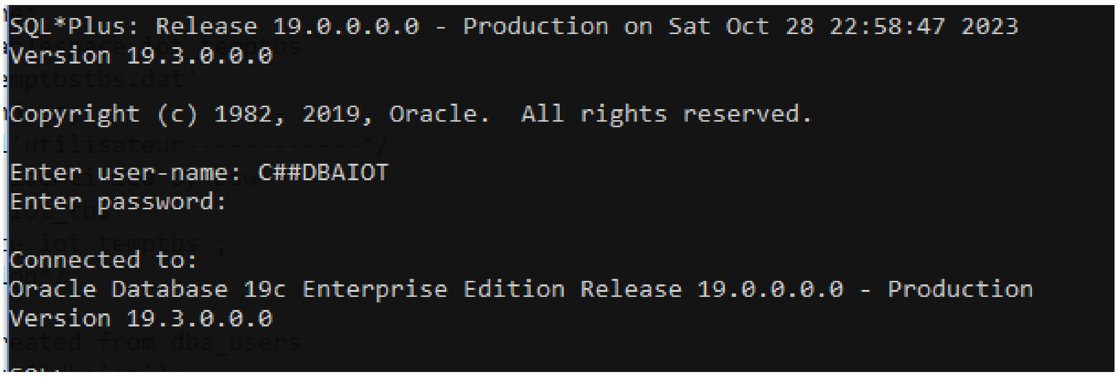
3. **Donner tous les privilèges à cet utilisateur.**

Requête :

```
GRANT ALL privileges   to  C##DBAIOT ;
```



```
SQL> GRANT ALL privileges   to  C##dbaiot ;

Grant succeeded.
```



```
SQL*Plus: Release 19.0.0.0.0 - Production on Sat Oct 28 22:58:47 2023
Version 19.3.0.0.0

Copyright (c) 1982, 2019, Oracle.  All rights reserved.

Enter user-name: C##DBAIOT
Enter password:

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.3.0.0.0
```

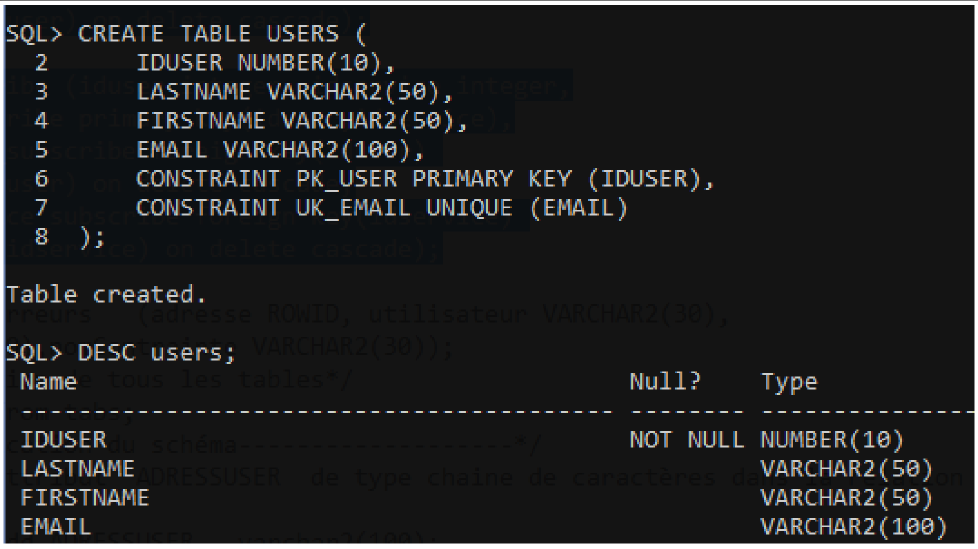
Partie II : Langage de définition de données

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

- **USERS table**

Requête :

```
CREATE TABLE USERS ( IDUSER NUMBER(10),LASTNAME VARCHAR2(50),
    FIRSTNAME VARCHAR2(50),EMAIL VARCHAR2(100),
    CONSTRAINT PK_USER PRIMARY KEY (IDUSER),
    CONSTRAINT UK_EMAIL UNIQUE (EMAIL)
);
```

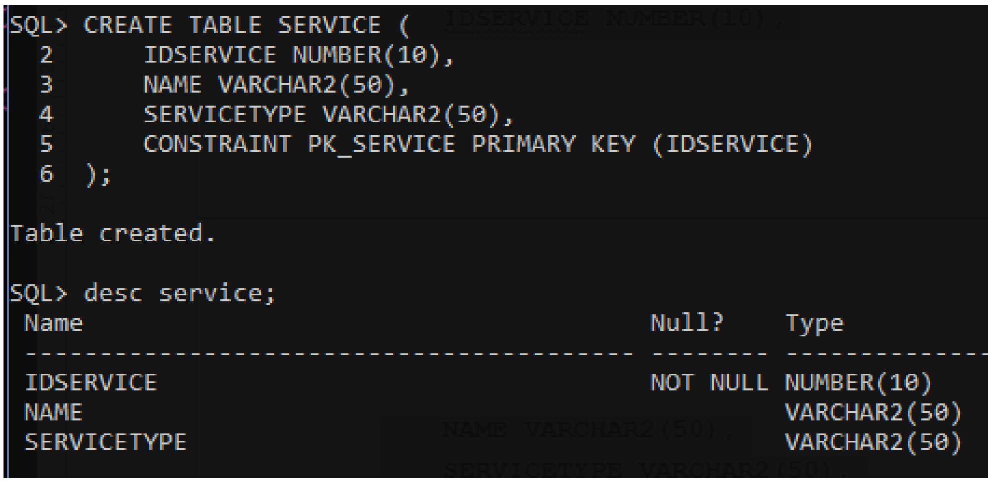


```
SQL> CREATE TABLE USERS (
         IDUSER NUMBER(10),
         LASTNAME VARCHAR2(50),
         FIRSTNAME VARCHAR2(50),
         EMAIL VARCHAR2(100),
         CONSTRAINT PK_USER PRIMARY KEY (IDUSER),
         CONSTRAINT UK_EMAIL UNIQUE (EMAIL)
     );

Table created.

SQL> DESC users;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 IDUSER                                    NOT NULL NUMBER(10)
 LASTNAME                                           VARCHAR2(50)
 FIRSTNAME                                          VARCHAR2(50)
 EMAIL                                              VARCHAR2(100)
```

- **SERVICE table**

Requête :

```
CREATE TABLE SERVICE (
    IDSERVICE NUMBER(10),
    NAME VARCHAR2(50),
    SERVICETYPE VARCHAR2(50),
    CONSTRAINT PK_SERVICE PRIMARY KEY (IDSERVICE)
);
```

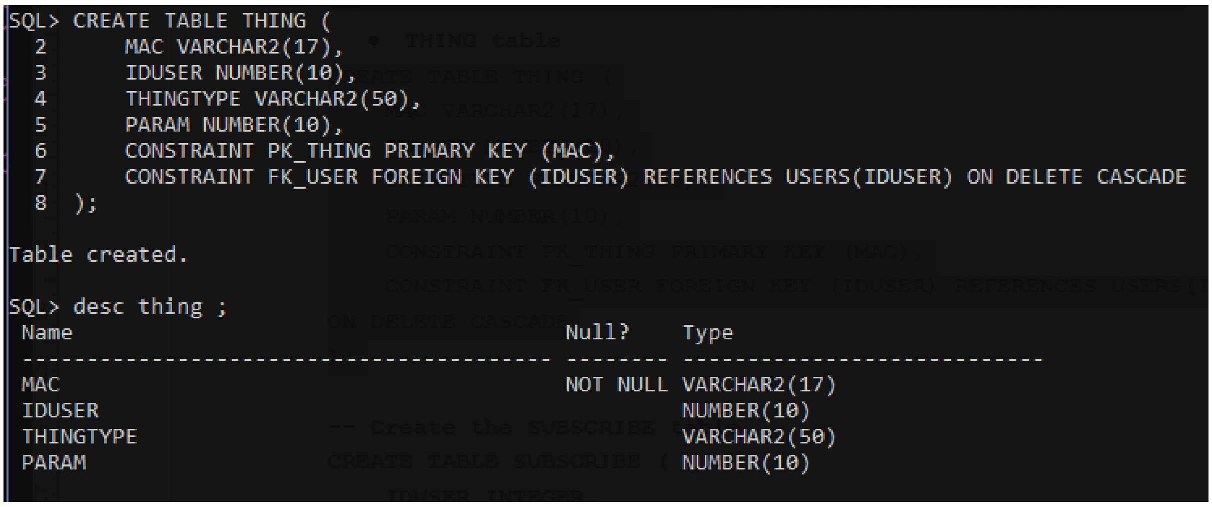


```
SQL> CREATE TABLE SERVICE (
         IDSERVICE NUMBER(10),
         NAME VARCHAR2(50),
         SERVICETYPE VARCHAR2(50),
         CONSTRAINT PK_SERVICE PRIMARY KEY (IDSERVICE)
     );

Table created.

SQL> desc service;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 IDSERVICE                                 NOT NULL NUMBER(10)
 NAME                                               VARCHAR2(50)
 SERVICETYPE                                        VARCHAR2(50)
```

- **THING table**

Requête :

```
CREATE TABLE THING (
    MAC VARCHAR2(17),
    IDUSER NUMBER(10),
    THINGTYPE VARCHAR2(50),
    PARAM NUMBER(10),
    CONSTRAINT PK_THING PRIMARY KEY (MAC),
    CONSTRAINT FK_USER FOREIGN KEY (IDUSER) REFERENCES USERS(IDUSER) ON
DELETE CASCADE
);
```

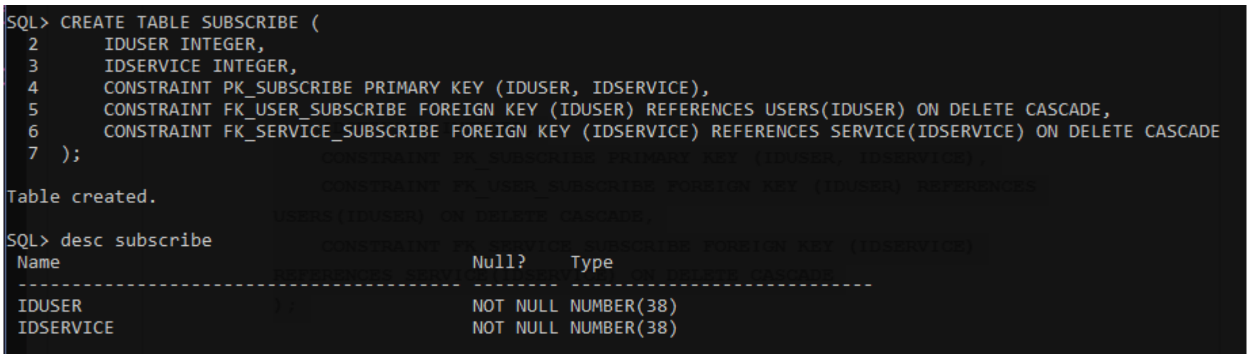


```
SQL> CREATE TABLE THING (
  2      MAC VARCHAR2(17),
  3      IDUSER NUMBER(10),
  4      THINGTYPE VARCHAR2(50),
  5      PARAM NUMBER(10),
  6      CONSTRAINT PK_THING PRIMARY KEY (MAC),
  7      CONSTRAINT FK_USER FOREIGN KEY (IDUSER) REFERENCES USERS(IDUSER) ON DELETE CASCADE
  8  );

Table created.

SQL> desc thing ;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 MAC                                       NOT NULL VARCHAR2(17)
 IDUSER                                             NUMBER(10)
 THINGTYPE                                          VARCHAR2(50)
 PARAM                                              NUMBER(10)
```

- **SUBSCRIBE table**

Requête :

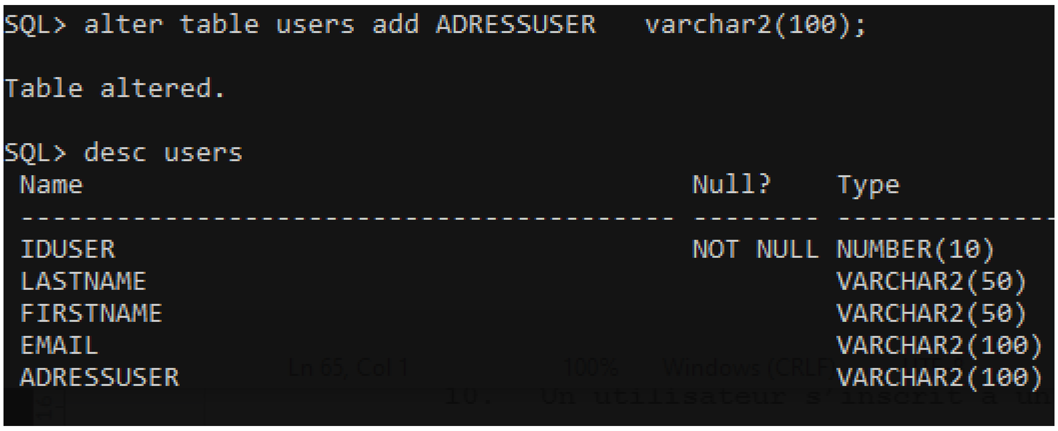
```
CREATE TABLE SUBSCRIBE (
    IDUSER INTEGER,
    IDSERVICE INTEGER,
    CONSTRAINT PK_SUBSCRIBE PRIMARY KEY (IDUSER, IDSERVICE),
    CONSTRAINT FK_USER_SUBSCRIBE FOREIGN KEY (IDUSER) REFERENCES
USERS(IDUSER) ON DELETE CASCADE,
    CONSTRAINT FK_SERVICE_SUBSCRIBE FOREIGN KEY (IDSERVICE) REFERENCES
SERVICE(IDSERVICE) ON DELETE CASCADE
);
```



```
SQL> CREATE TABLE SUBSCRIBE (
  2      IDUSER INTEGER,
  3      IDSERVICE INTEGER,
  4      CONSTRAINT PK_SUBSCRIBE PRIMARY KEY (IDUSER, IDSERVICE),
  5      CONSTRAINT FK_USER_SUBSCRIBE FOREIGN KEY (IDUSER) REFERENCES USERS(IDUSER) ON DELETE CASCADE,
  6      CONSTRAINT FK_SERVICE_SUBSCRIBE FOREIGN KEY (IDSERVICE) REFERENCES SERVICE(IDSERVICE) ON DELETE CASCADE
  7  );

Table created.

SQL> desc subscribe
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 IDUSER                                    NOT NULL NUMBER(38)
 IDSERVICE                                 NOT NULL NUMBER(38)
```

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

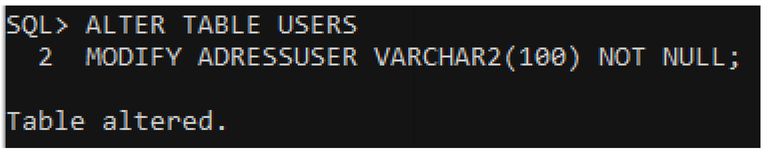
Requête :
ALTER TABLE USERS ADD ADRESSUSER   VARCHAR2(100);



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

Table altered.

SQL> desc users
 Name                                      Null?    Type
 ----------------------------------------- -------- -------------
 IDUSER                                    NOT NULL NUMBER(10)
 LASTNAME                                           VARCHAR2(50)
 FIRSTNAME                                          VARCHAR2(50)
 EMAIL                                              VARCHAR2(100)
 ADRESSUSER                                         VARCHAR2(100)
```

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

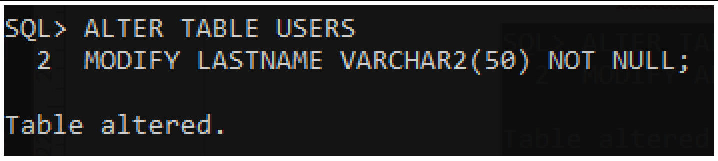
Requête :
ALTER TABLE USERS MODIFY ADRESSUSER VARCHAR2(100) NOT NULL;



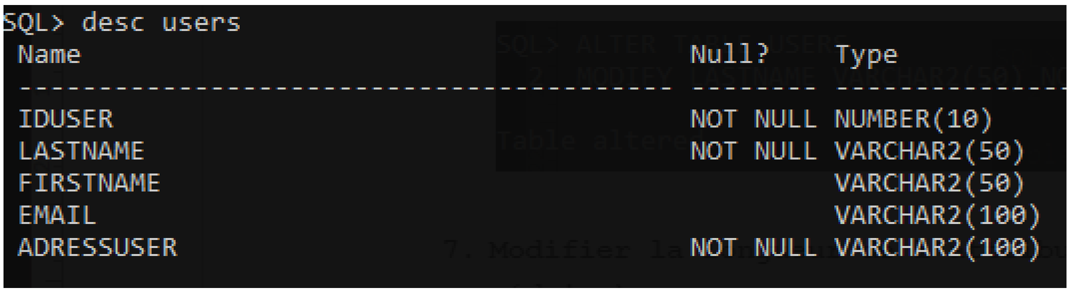
```
SQL> ALTER TABLE USERS
  2  MODIFY ADRESSUSER VARCHAR2(100) NOT NULL;

Table altered.
```

Requête :

```
ALTER TABLE USERS MODIFY LASTNAME VARCHAR2(50) NOT NULL;
```



```
SQL> ALTER TABLE USERS
  2  MODIFY LASTNAME VARCHAR2(50) NOT NULL;

Table altered.
```



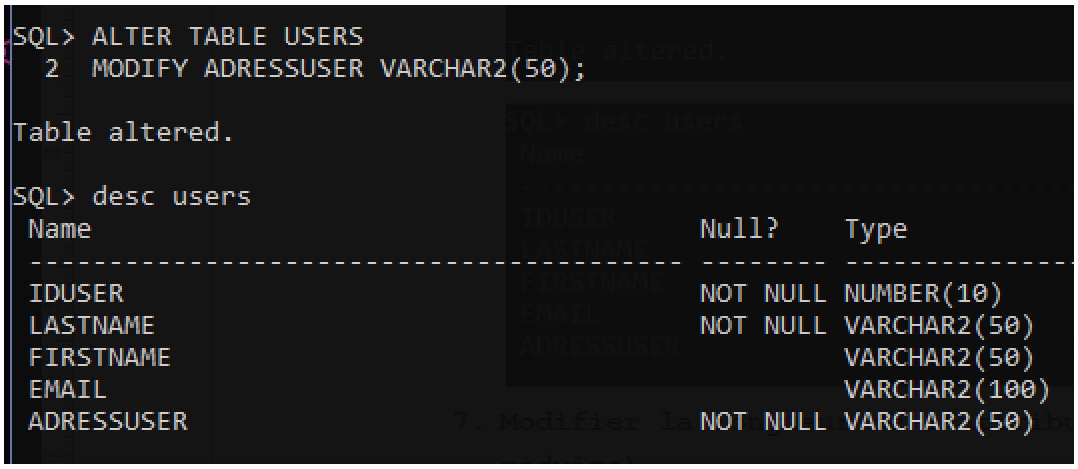
```
SQL> desc users
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 IDUSER                                    NOT NULL NUMBER(10)
 LASTNAME                                  NOT NULL VARCHAR2(50)
 FIRSTNAME                                          VARCHAR2(50)
 EMAIL                                              VARCHAR2(100)
 ADRESSUSER                                NOT NULL VARCHAR2(100)
```

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

Requête :

```
ALTER TABLE USERS MODIFY LASTNAME VARCHAR2(50) NOT NULL;
```

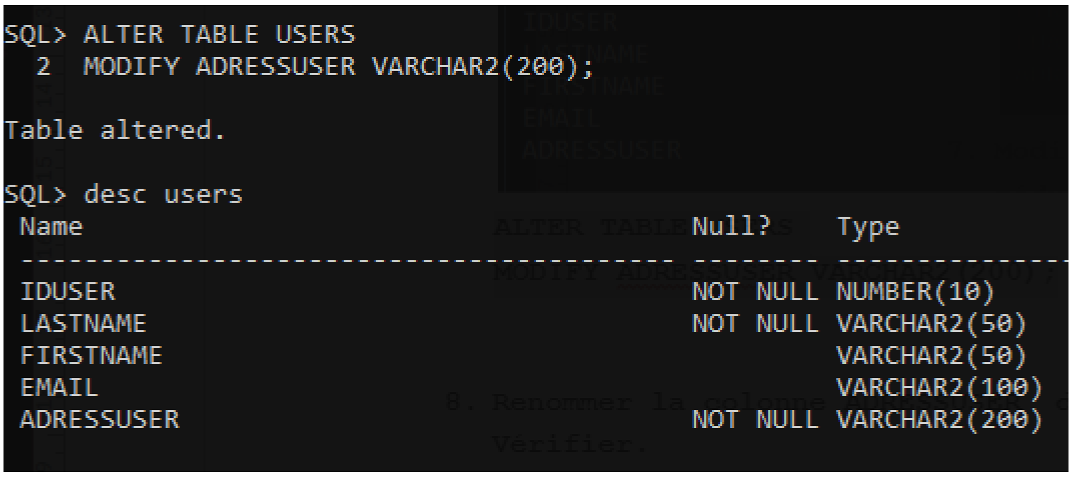
ALTER TABLE USERS
MODIFY ADRESSUSER VARCHAR2(50);



```
SQL> ALTER TABLE USERS
  2  MODIFY ADRESSUSER VARCHAR2(50);

Table altered.

SQL> desc users
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 IDUSER                                    NOT NULL NUMBER(10)
 LASTNAME                                  NOT NULL VARCHAR2(50)
 FIRSTNAME                                          VARCHAR2(50)
 EMAIL                                              VARCHAR2(100)
 ADRESSUSER                                NOT NULL VARCHAR2(50)
```

Requête :

```
ALTER TABLE USERS MODIFY ADRESSUSER VARCHAR2(200);
```

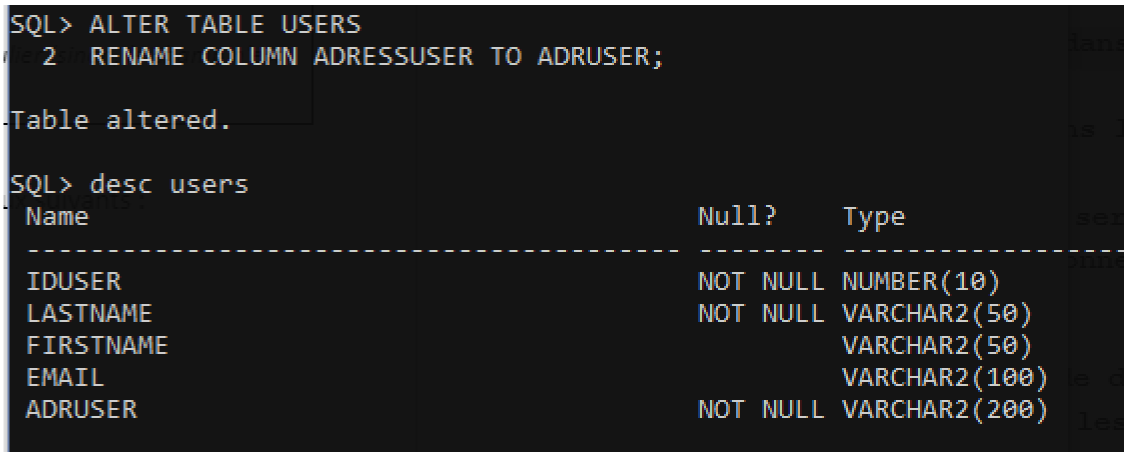


```
SQL> ALTER TABLE USERS
  2  MODIFY ADRESSUSER VARCHAR2(200);

Table altered.

SQL> desc users
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------
 IDUSER                                    NOT NULL NUMBER(10)
 LASTNAME                                  NOT NULL VARCHAR2(50)
 FIRSTNAME                                          VARCHAR2(50)
 EMAIL                                              VARCHAR2(100)
 ADRESSUSER                                NOT NULL VARCHAR2(200)
```

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

Requête :

```
ALTER TABLE USERS MODIFY RENAME COLUMN ADRESSUSER TO ADRUSER;
```

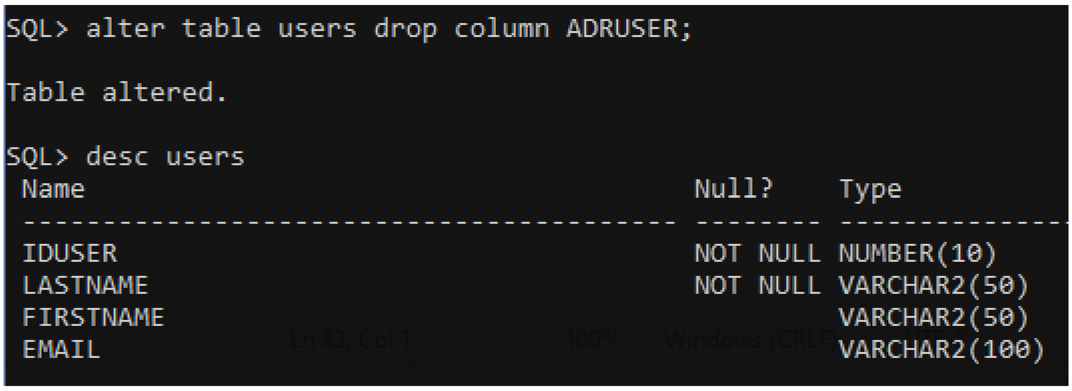


```
SQL> ALTER TABLE USERS
  2  RENAME COLUMN ADRESSUSER TO ADRUSER;

Table altered.

SQL> desc users
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------
 IDUSER                                    NOT NULL NUMBER(10)
 LASTNAME                                  NOT NULL VARCHAR2(50)
 FIRSTNAME                                          VARCHAR2(50)
 EMAIL                                              VARCHAR2(100)
 ADRUSER                                   NOT NULL VARCHAR2(200)
```

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

Requête :

```
ALTER TABLE USERS DROP COLUMN ADRUSER;
```

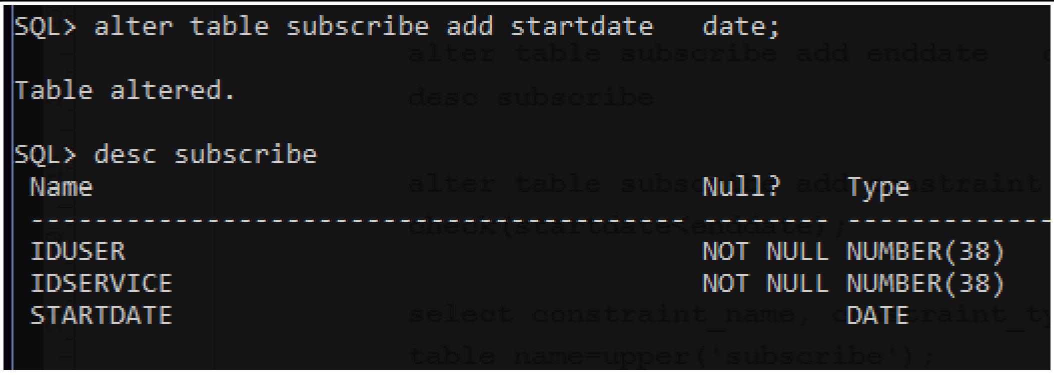


```
SQL> alter table users drop column ADRUSER;

Table altered.

SQL> desc users
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------
 IDUSER                                    NOT NULL NUMBER(10)
 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.**

Requête :

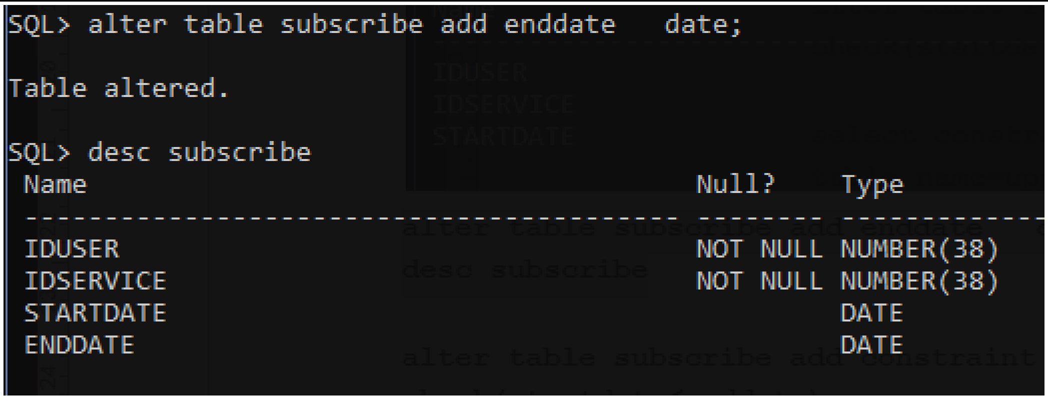
```
ALTER TABLE SUBSCRIBE ADD STARTDATE   DATE;
DESC SUBSCRIBE;
```



```
SQL> alter table subscribe add startdate   date;

Table altered.

SQL> desc subscribe
 Name                                      Null?    Type
 ----------------------------------------- -------- ------------
 IDUSER                                    NOT NULL NUMBER(38)
 IDSERVICE                                 NOT NULL NUMBER(38)
 STARTDATE                                          DATE
```

Requête :

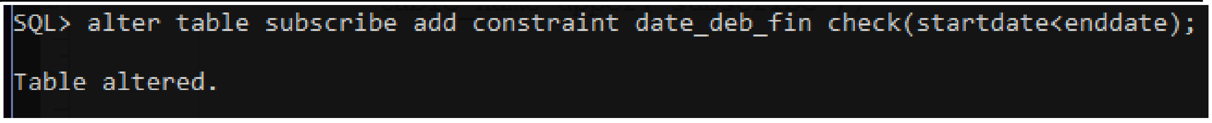
```
ALTER TABLE SUBSCRIBE ADD ENDDATE   DATE;
DESC SUBSCRIBE
```



```
SQL> alter table subscribe add enddate   date;

Table altered.

SQL> desc subscribe
 Name                                      Null?    Type
 ----------------------------------------- -------- ------------
 IDUSER                                    NOT NULL NUMBER(38)
 IDSERVICE                                 NOT NULL NUMBER(38)
 STARTDATE                                          DATE
 ENDDATE                                            DATE
```

Requête :

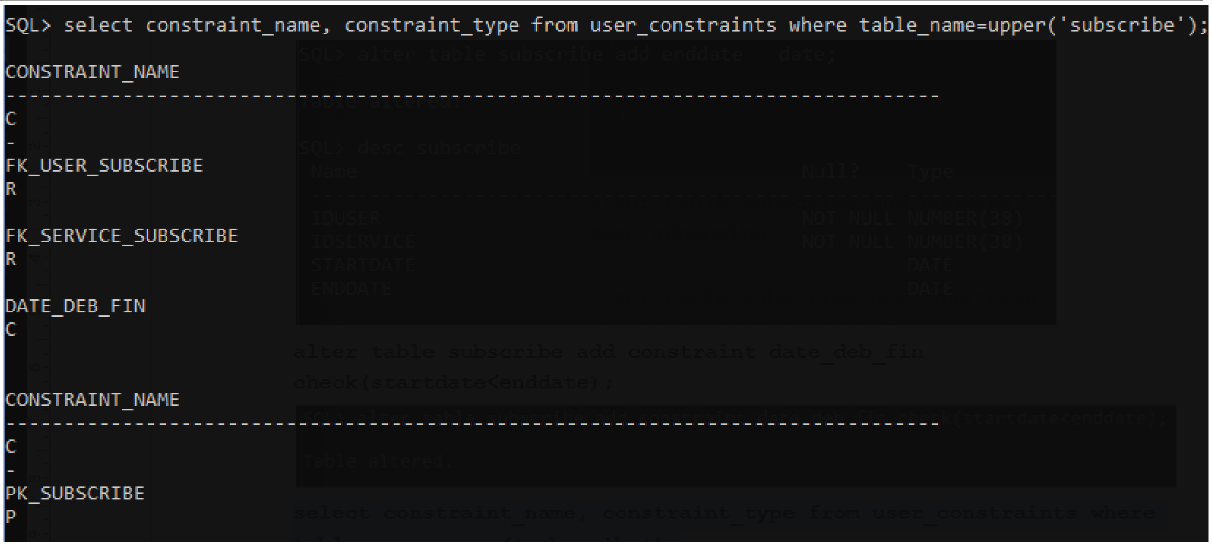
```
ALTER TABLE SUBSCRIBE ADD CONSTRAINT DATE_DEB_FIN CHECK(STARTDATE<ENDDATE);
```



```
SQL> alter table subscribe add constraint date_deb_fin check(startdate<enddate);

Table altered.
```

Requête :

```
SELECT CONSTRAINT_NAME, CONSTRAINT_TYPE FROM USER_CONSTRAINTS WHERE
TABLE_NAME=UPPER('SUBSCRIBE');
```



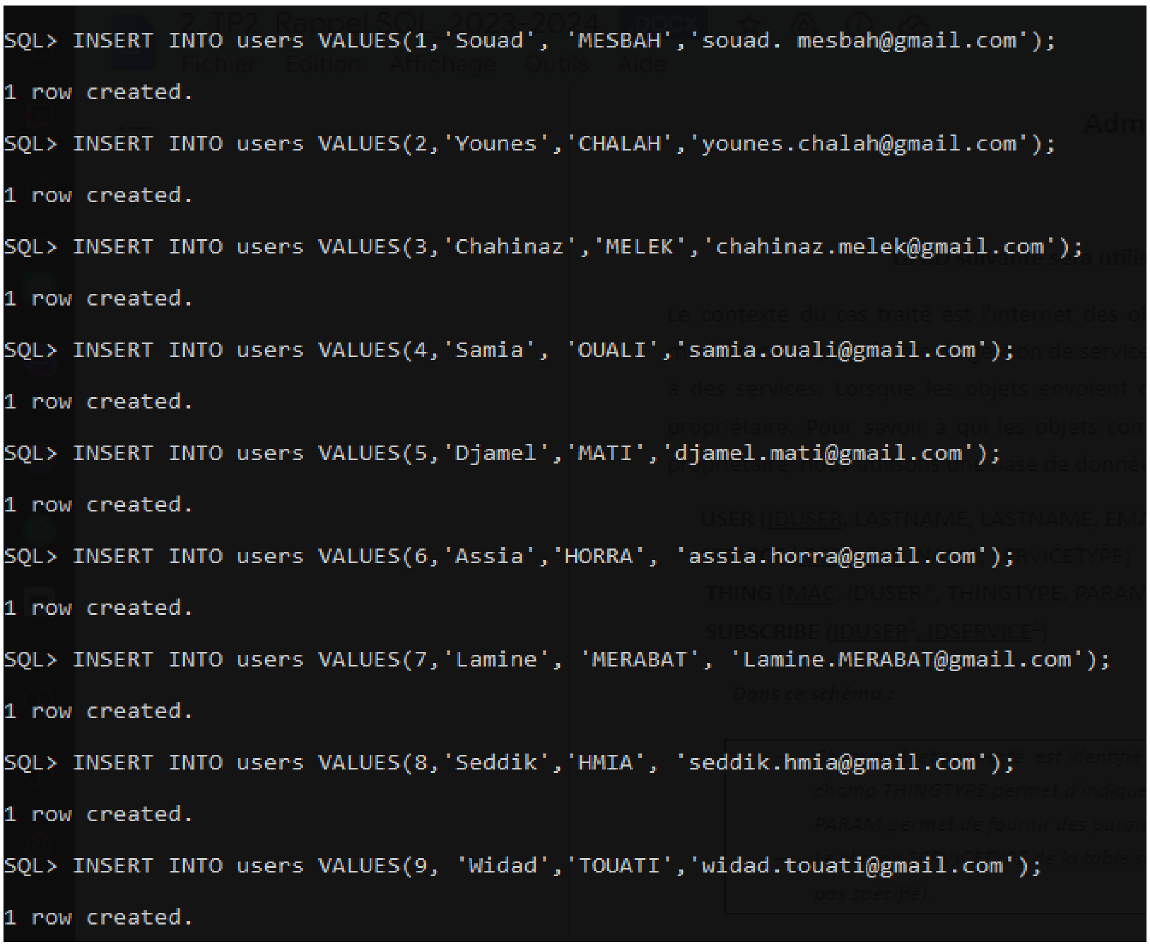
```
SQL> select constraint_name, constraint_type from user_constraints where table_name=upper('subscribe');

CONSTRAINT_NAME
--------------------------------------------------------------------------------
C
-
FK_USER_SUBSCRIBE
R

FK_SERVICE_SUBSCRIBE
R

DATE_DEB_FIN
C


CONSTRAINT_NAME
--------------------------------------------------------------------------------
C
-
PK_SUBSCRIBE
P
```

Partie III : Langage de manipulation de données

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

Requête :

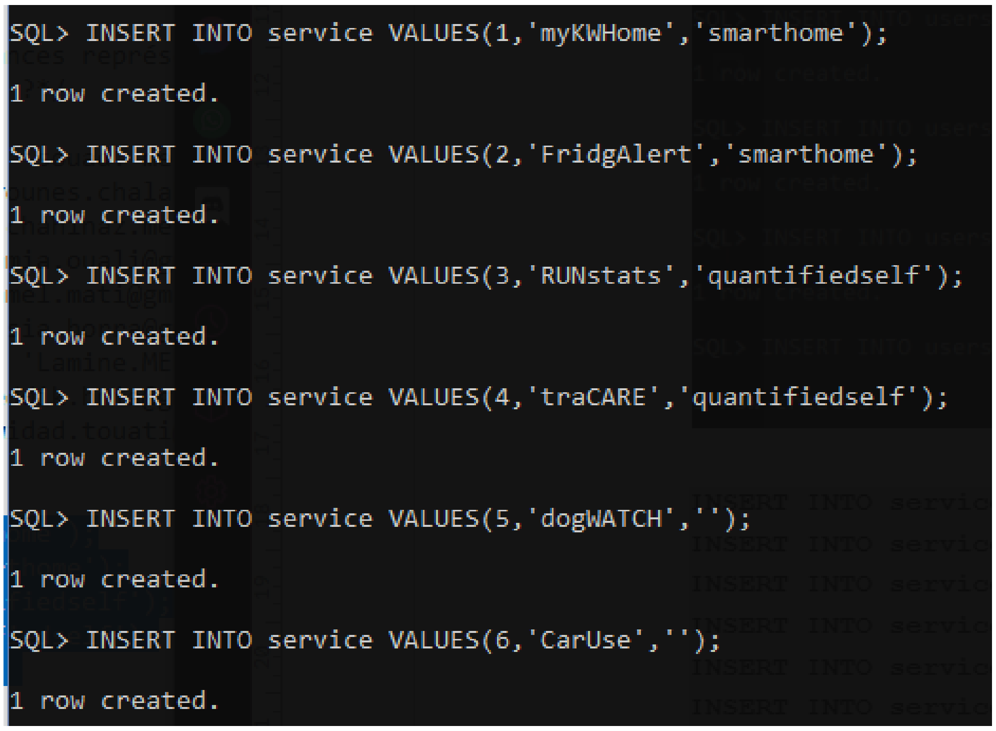
```
INSERT INTO users VALUES(1,'Souad', 'MESBAH',    'souad.
mesbah@gmail.com');
INSERT INTO users VALUES(2,'Younes','CHALAH','younes.chalah@gmail.com');
INSERT INTO users VALUES(3,'Chahinaz','MELEK','chahinaz.melek@gmail.com');
INSERT INTO users VALUES(4,'Samia', 'OUALI','samia.ouali@gmail.com');
INSERT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



```
SQL> INSERT INTO users VALUES(1,'Souad', 'MESBAH','souad. mesbah@gmail.com');

1 row created.

SQL> INSERT INTO users VALUES(2,'Younes','CHALAH','younes.chalah@gmail.com');

1 row created.

SQL> INSERT INTO users VALUES(3,'Chahinaz','MELEK','chahinaz.melek@gmail.com');

1 row created.

SQL> INSERT INTO users VALUES(4,'Samia', 'OUALI','samia.ouali@gmail.com');

1 row created.

SQL> INSERT INTO users VALUES(5,'Djamel','MATI','djamel.mati@gmail.com');

1 row created.

SQL> INSERT INTO users VALUES(6,'Assia','HORRA', 'assia.horra@gmail.com');

1 row created.

SQL> INSERT INTO users VALUES(7,'Lamine', 'MERABAT', 'Lamine.MERABAT@gmail.com');

1 row created.

SQL> INSERT INTO users VALUES(8,'Seddik','HMIA', 'seddik.hmia@gmail.com');

1 row created.

SQL> INSERT INTO users VALUES(9, 'Widad','TOUATI','widad.touati@gmail.com');

1 row created.
```

Requête :

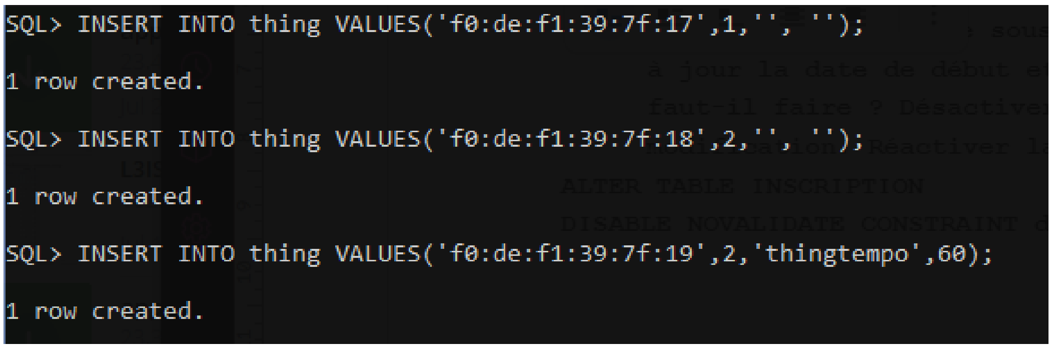
```
INSERT INTO service VALUES(1,'myKWHome','smarthome');
INSERT INTO service VALUES(2,'FridgAlert','smarthome');
INSERT INTO service VALUES(3,'RUNstats','quantifiedself');
INSERT INTO service VALUES(4,'traCARE','quantifiedself');
INSERT INTO service VALUES(5,'dogWATCH','');
INSERT INTO service VALUES(6,'CarUse','');
```



```
SQL> INSERT INTO service VALUES(1,'myKWHome','smarthome');

1 row created.

SQL> INSERT INTO service VALUES(2,'FridgAlert','smarthome');

1 row created.

SQL> INSERT INTO service VALUES(3,'RUNstats','quantifiedself');

1 row created.

SQL> INSERT INTO service VALUES(4,'traCARE','quantifiedself');

1 row created.

SQL> INSERT INTO service VALUES(5,'dogWATCH','');

1 row created.

SQL> INSERT INTO service VALUES(6,'CarUse','');

1 row created.
```

Requête :

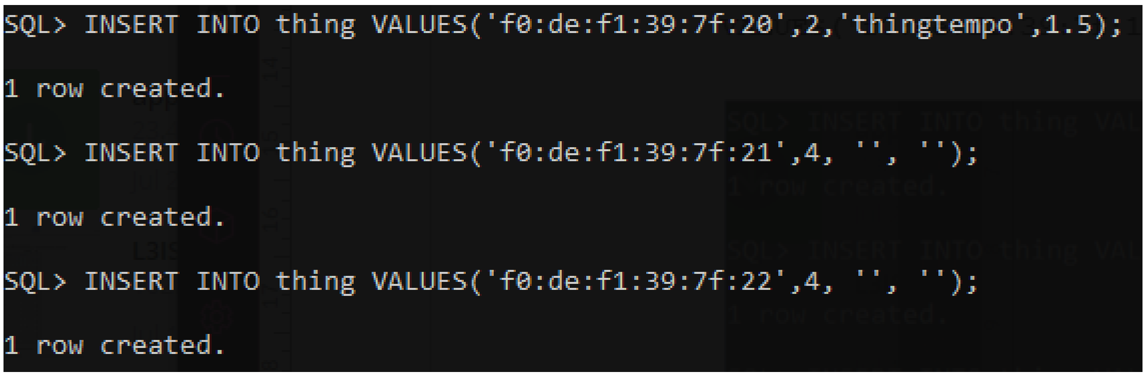
```
INSERT INTO thing VALUES('f0:de:f1:39:7f:17',1,'', '');
INSERT INTO thing VALUES('f0:de:f1:39:7f:18',2,'', '');
INSERT INTO thing VALUES('f0:de:f1:39:7f:19',2,'thingtempo',60);
```



```
SQL> INSERT INTO thing VALUES('f0:de:f1:39:7f:17',1,'', '');

1 row created.

SQL> INSERT INTO thing VALUES('f0:de:f1:39:7f:18',2,'', '');

1 row created.

SQL> INSERT INTO thing VALUES('f0:de:f1:39:7f:19',2,'thingtempo',60);

1 row created.
```

Requête :

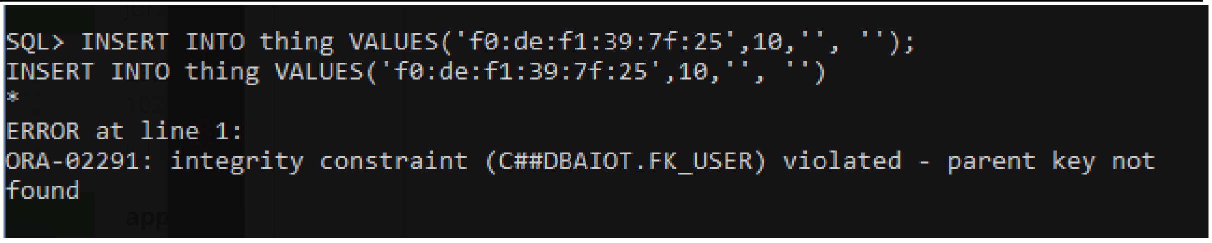
```
INSERT INTO thing VALUES('f0:de:f1:39:7f:20',2,'thingtempo',1.5);
INSERT INTO thing VALUES('f0:de:f1:39:7f:21',4, '', '');
INSERT INTO thing VALUES('f0:de:f1:39:7f:22',4, '', '');
```



```
SQL> INSERT INTO thing VALUES('f0:de:f1:39:7f:20',2,'thingtempo',1.5);

1 row created.

SQL> INSERT INTO thing VALUES('f0:de:f1:39:7f:21',4, '', '');

1 row created.

SQL> INSERT INTO thing VALUES('f0:de:f1:39:7f:22',4, '', '');

1 row created.
```

Requête :

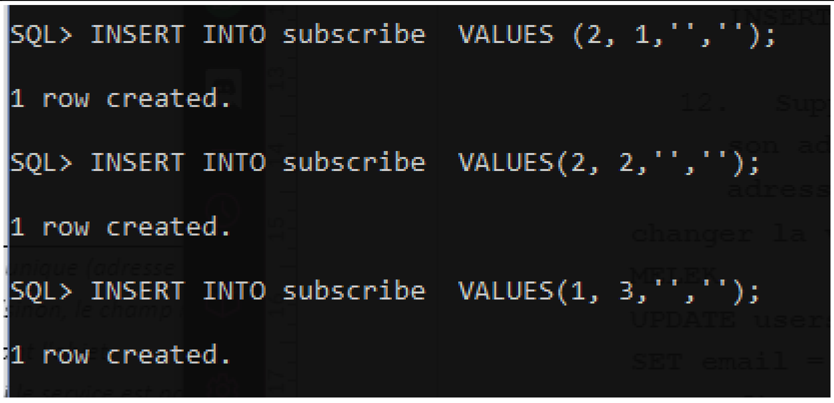
```
INSERT INTO thing VALUES('f0:de:f1:39:7f:25',10,'', '');
```



```
SQL> INSERT INTO thing VALUES('f0:de:f1:39:7f:25',10,'', '');
INSERT INTO thing VALUES('f0:de:f1:39:7f:25',10,'', '')
*
ERROR at line 1:
ORA-02291: integrity constraint (C##DBAIOT.FK_USER) violated - parent key not
found
```

Requête :

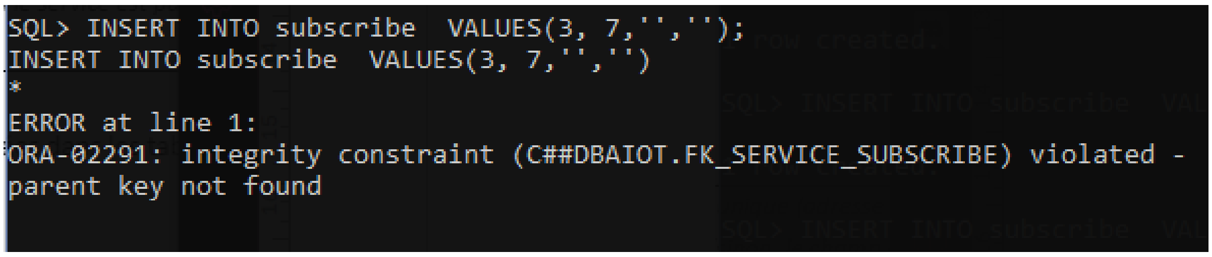
```
INSERT INTO subscribe  VALUES (2, 1,'','');
INSERT INTO subscribe  VALUES(2, 2,'','');
INSERT INTO subscribe  VALUES(1, 3,'','');
```



```
SQL> INSERT INTO subscribe  VALUES (2, 1,'','');

1 row created.

SQL> INSERT INTO subscribe  VALUES(2, 2,'','');

1 row created.

SQL> INSERT INTO subscribe  VALUES(1, 3,'','');

1 row created.
```

Requête :

```
INSERT INTO subscribe  VALUES(3, 7,'','');
```



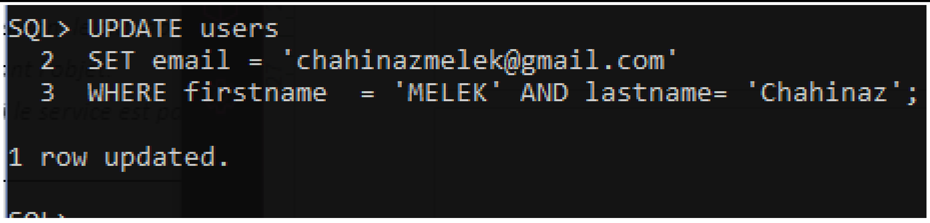
```
SQL> INSERT INTO subscribe  VALUES(3, 7,'','');
INSERT INTO subscribe  VALUES(3, 7,'','')
*
ERROR at line 1:
ORA-02291: integrity constraint (C##DBAIOT.FK_SERVICE_SUBSCRIBE) violated -
parent key not found
```

12. **Supposons que l'utilisateur Chahinaz MELEK a perdu l'accès à son adresse mail et elle veut le remplacer par la nouvelle adresse chahinazmelek@gmail.com. Que faut-il faire ?**

changer la valeur de email dans la table ou l'utilisateur Chahinaz MELEK

Requête :

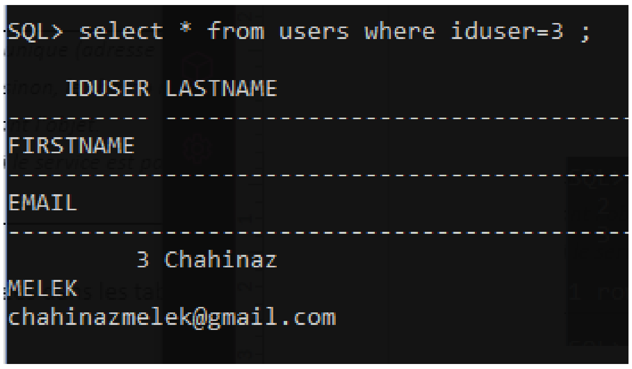
```
UPDATE users
SET email = 'chahinazmelek@gmail.com'
WHERE firstname  = 'MELEK' AND lastname= 'Chahinaz';
```



```
SQL> UPDATE users
  2  SET email = 'chahinazmelek@gmail.com'
  3  WHERE firstname  = 'MELEK' AND lastname= 'Chahinaz';

1 row updated.
```

Requête :

```
SELECT * FROM USERS WHERE IDUSER=3 ;
```

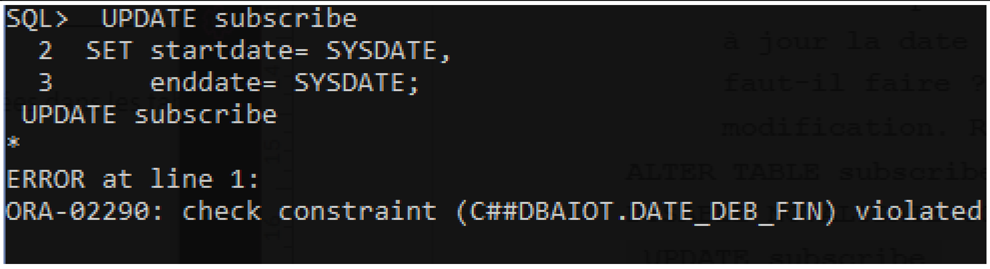


```
SQL> select * from users where iduser=3 ;

    IDUSER LASTNAME
---------- --------------------------------
FIRSTNAME
----------------------------------------
EMAIL
----------------------------------------
         3 Chahinaz
MELEK
chahinazmelek@gmail.com
```

13. **Pour la période de souscription à un service, on veut mettre à jour la date de début et fin par la date de système. Que faut-il faire ? Désactiver la contrainte pour autoriser la modification. Réactiver la contrainte.**

Requête :

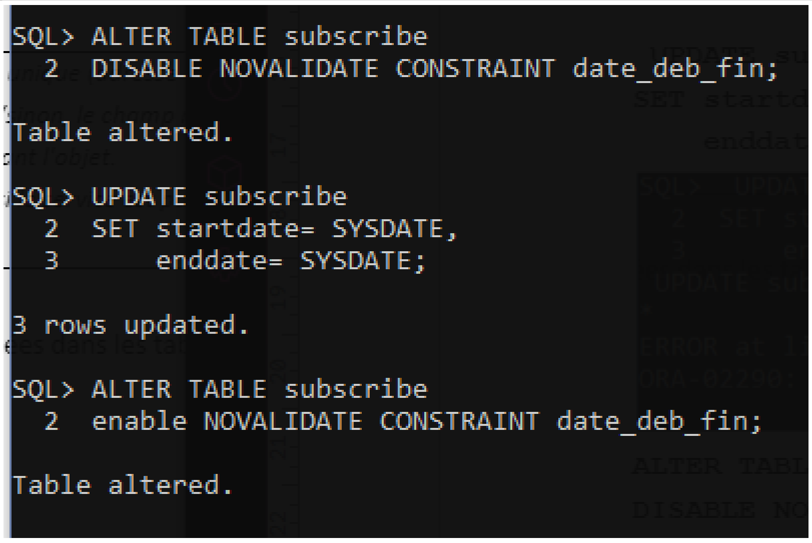
```
UPDATE subscribe SET startdate= SYSDATE, enddate= SYSDATE;
```



```
SQL>  UPDATE subscribe
  2  SET startdate= SYSDATE,
  3      enddate= SYSDATE;
 UPDATE subscribe
*
ERROR at line 1:
ORA-02290: check constraint (C##DBAIOT.DATE_DEB_FIN) violated
```

Requête :

```
ALTER TABLE subscribe
DISABLE NOVALIDATE CONSTRAINT date_deb_fin;

ALTER TABLE subscribe
enable NOVALIDATE CONSTRAINT date_deb_fin;
```



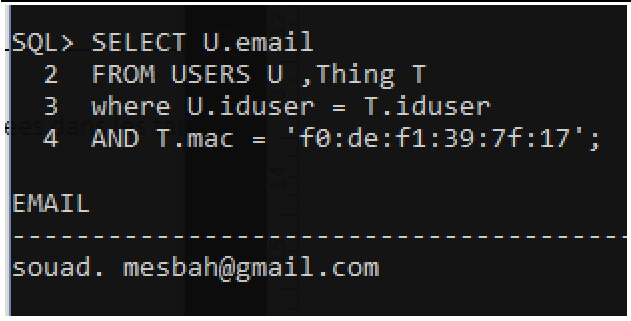
```
SQL> ALTER TABLE subscribe
  2  DISABLE NOVALIDATE CONSTRAINT date_deb_fin;

Table altered.

SQL> UPDATE subscribe
  2  SET startdate= SYSDATE,
  3      enddate= SYSDATE;

3 rows updated.

SQL> ALTER TABLE subscribe
  2  enable NOVALIDATE CONSTRAINT date_deb_fin;

Table altered.
```

Partie IV : Langage d'interrogation de données

14. Quel est l'adresse email de l'utilisateur qui possède l'objet d'adresse MAC f0:de:f1:39:7f:17 ?

Requête :

```
SELECT U.email
FROM USERS U ,Thing T
where U.iduser = T.iduser
AND T.mac = 'f0:de:f1:39:7f:17';
```

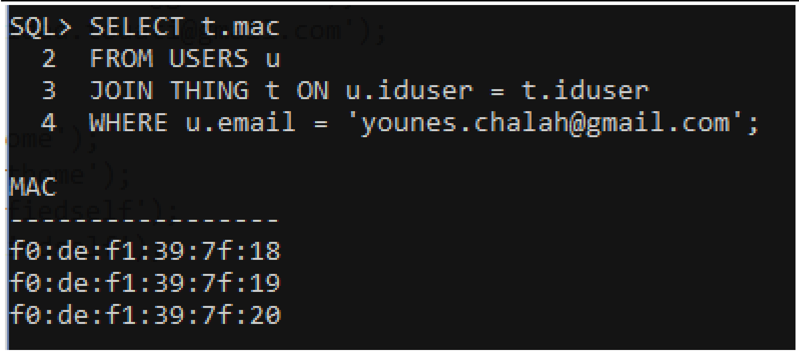


```
SQL> SELECT U.email
  2  FROM USERS U ,Thing T
  3  where U.iduser = T.iduser
  4  AND T.mac = 'f0:de:f1:39:7f:17';

EMAIL
----------------------------------------
souad. mesbah@gmail.com
```

15. Quels sont les adresses MAC des objets appartenant à l'utilisateur dont l'adresse email est younes.chalah@gmail.com?

Requête :

```
SELECT t.mac
FROM USERS u
JOIN THING t ON u.iduser = t.iduser
WHERE u.email = 'younes.chalah@gmail.com';
```

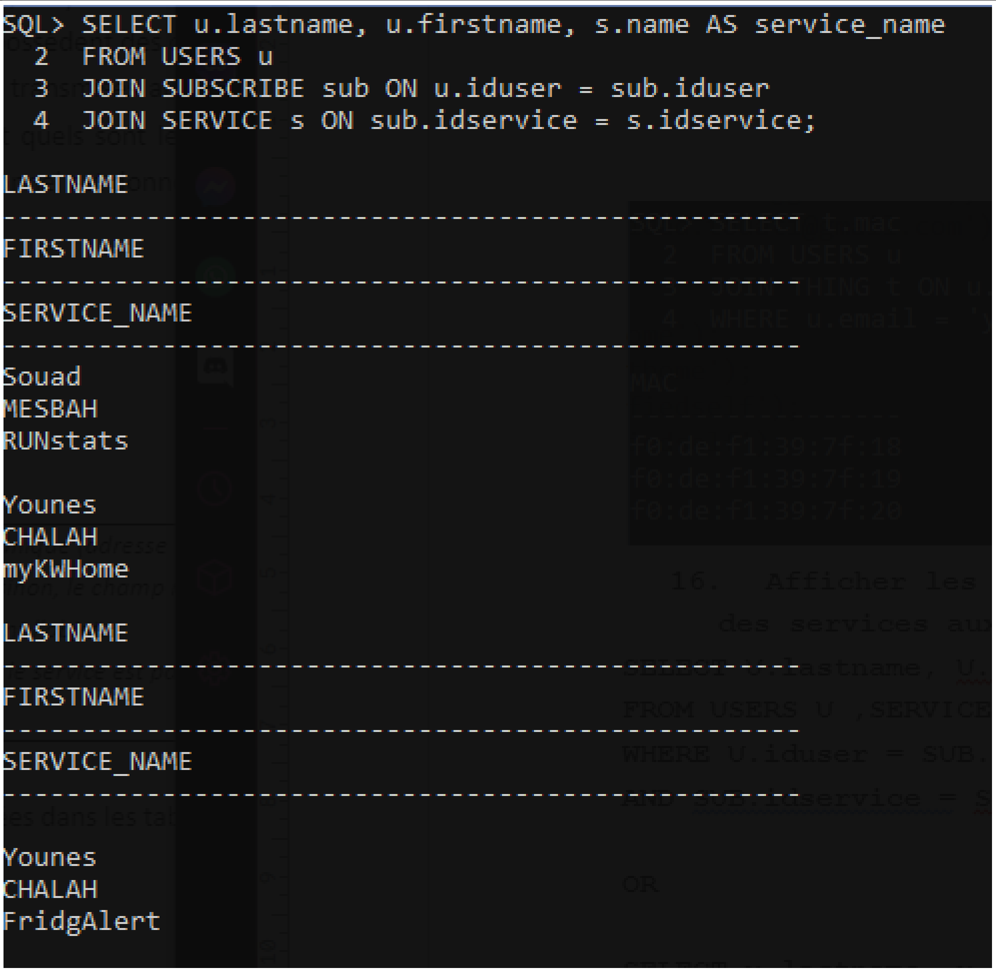


```
SQL> SELECT t.mac
  2  FROM USERS u
  3  JOIN THING t ON u.iduser = t.iduser
  4  WHERE u.email = 'younes.chalah@gmail.com';

MAC
-----------------
f0:de:f1:39:7f:18
f0:de:f1:39:7f:19
f0:de:f1:39:7f:20
```

16. Afficher les noms et prénoms des utilisateurs avec les noms des services auxquels ils sont abonnés.

Requête :

```
SELECT U.lastname, U.firstname, S.name AS service_name
FROM USERS U ,SERVICE S,SUBSCRIBE SUB
WHERE U.iduser = SUB.iduser
AND SUB.idservice = S.idservice;

OR

SELECT u.lastname, u.firstname, s.name AS service_name
FROM USERS u
JOIN SUBSCRIBE sub ON u.iduser = sub.iduser
JOIN SERVICE s ON sub.idservice = s.idservice;
```

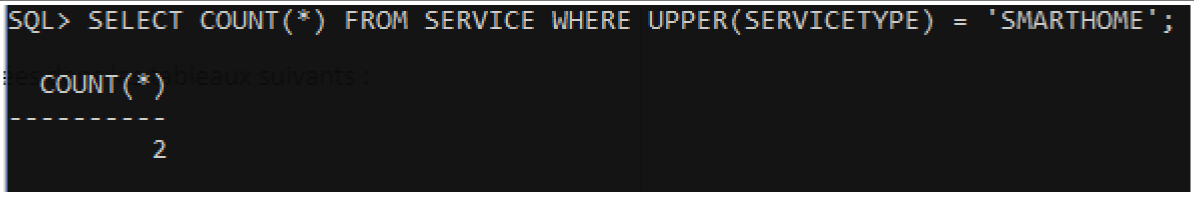


```
SQL> SELECT u.lastname, u.firstname, s.name AS service_name
  2  FROM USERS u
  3  JOIN SUBSCRIBE sub ON u.iduser = sub.iduser
  4  JOIN SERVICE s ON sub.idservice = s.idservice;

LASTNAME
--------------------------------------------------
FIRSTNAME
--------------------------------------------------
SERVICE_NAME
--------------------------------------------------
Souad
MESBAH
RUNstats

Younes
CHALAH
myKWHome

LASTNAME
--------------------------------------------------
FIRSTNAME
--------------------------------------------------
SERVICE_NAME
--------------------------------------------------
Younes
CHALAH
FridgAlert
```

17. **Combien de services sont de type smarthome ?**

Requête :

```
SELECT COUNT(*) FROM SERVICE WHERE UPPER(SERVICETYPE) = 'SMARTHOME';
```

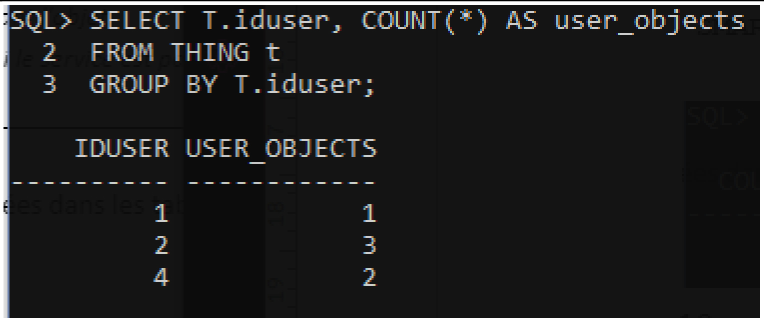


```
SQL> SELECT COUNT(*) FROM SERVICE WHERE UPPER(SERVICETYPE) = 'SMARTHOME';

  COUNT(*)
----------
         2
```

18. **Afficher les id des propriétaires d'objets avec le nombre d'objets qu'ils possèdent.**

Requête :

```
SELECT T.iduser, COUNT(*) AS user_objects
FROM THING t
GROUP BY T.iduser;
```

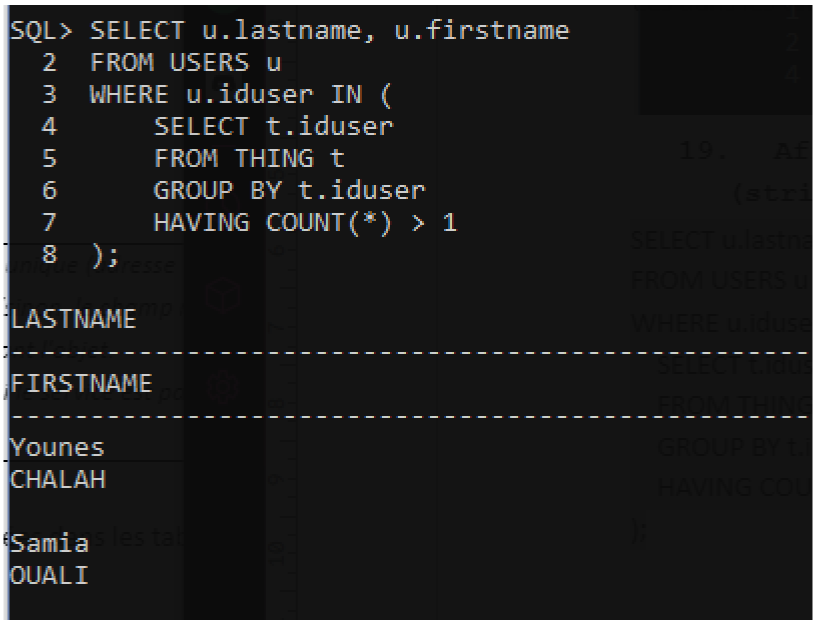


```
SQL> SELECT T.iduser, COUNT(*) AS user_objects
  2  FROM THING t
  3  GROUP BY T.iduser;

    IDUSER USER_OBJECTS
---------- ------------
         1            1
         2            3
         4            2
```

19. **Afficher les noms et prénoms des propriétaires de (strictement) plus de 1 objet.**

Requête :

```
SELECT u.lastname, u.firstname
FROM USERS u
WHERE u.iduser IN (
    SELECT t.iduser
    FROM THING t
    GROUP BY t.iduser
    HAVING COUNT(*) > 1
);
```



```
SQL> SELECT u.lastname, u.firstname
  2  FROM USERS u
  3  WHERE u.iduser IN (
  4      SELECT t.iduser
  5      FROM THING t
  6      GROUP BY t.iduser
  7      HAVING COUNT(*) > 1
  8  );

LASTNAME
--------------------------------------------------
FIRSTNAME
--------------------------------------------------
Younes
CHALAH

Samia
OUALI
```