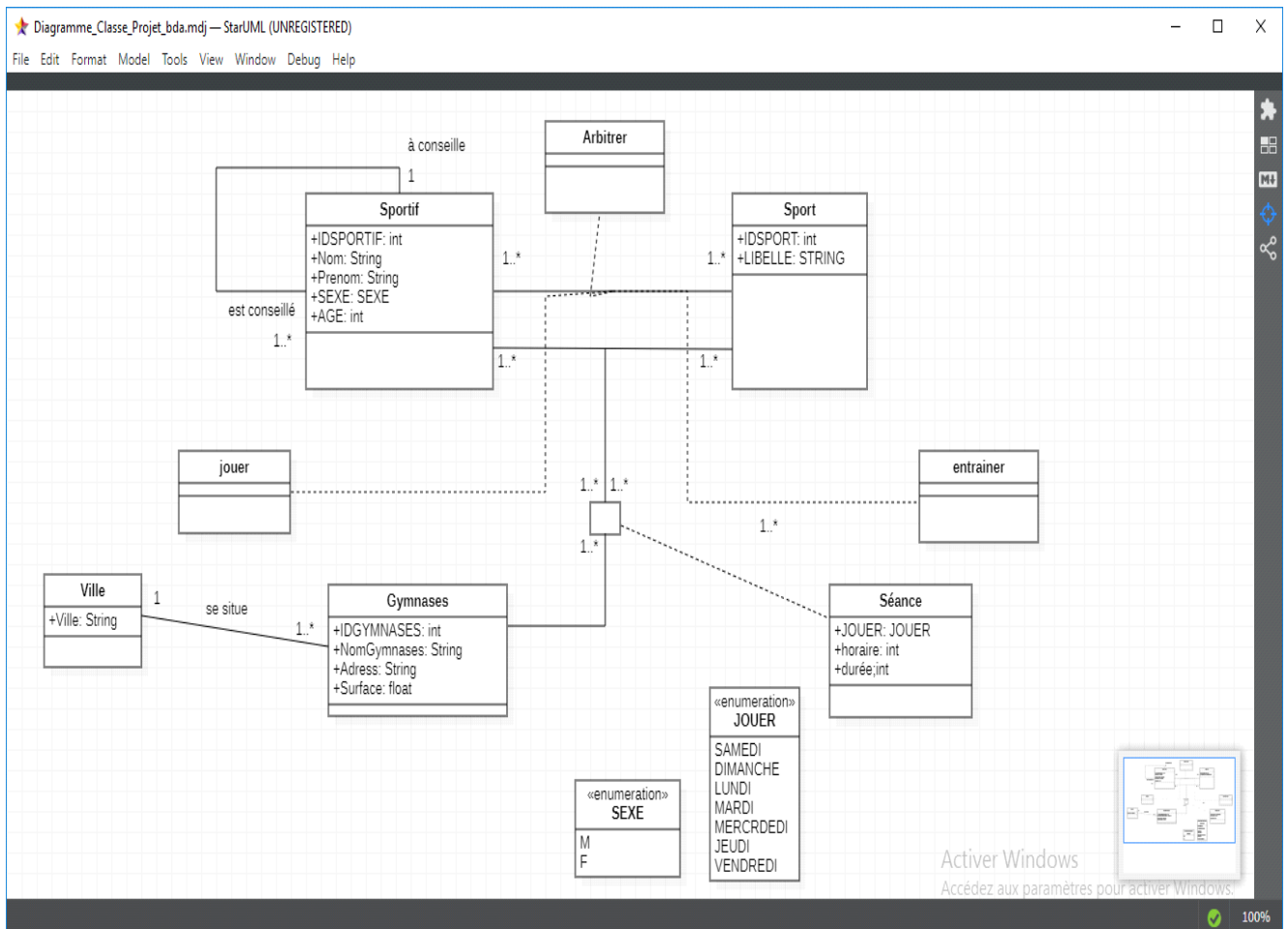


## Partie I : Modélisation orientée objet

### 1-1-Transformation de schéma relationnel vers le digramme de classe :



### 1-2-Transformation de digramme de classe :



leur ville de localisation.

\*)La table ARBITRER indique quels sportifs arbitrent quels sports.

\*)La table ENTRAÎNER indique quels sportifs entraînent quels sports.

\*)La table JOUER indique quels sportifs jouent quels sports.

\*)La table SEANCES indique les séances d'entraînement pour un sport donné dans un gymnase donné à un moment donné.

----> Les associations sont :

\*)Un sportif peut conseiller un autre sportif (relation "Conseiller").

\*)Un gymnase est situé dans une ville (relation "Ville").

\*)Un sport peut être pratiqué par plusieurs sportifs et un sportif peut pratiquer plusieurs sports (relation "Jouer").

\*)Un sport peut être arbitré par plusieurs sportifs et un sportif peut arbitrer plusieurs sports (relation "Arbitrer").

\*)Un sport peut être entraîné par plusieurs sportifs et un sportif peut entraîner plusieurs sports (relation "Entraîner").

\*)ainsi que la relation entre Séance et Sport , Séance et Gymnase , Séance et Sportif .

## **Partie II : Création des TableSpaces et utilisateur**

### **2-1- Création des deux TableSpaces SQL3 TBS et SQL3 TempTBS:**

```
Run SQL Command Line

Copyright (c) 1982, 2014, Oracle. All rights reserved.

SQL> connect / as sysdba;
Connected.
SQL> create tablespace SQL33_TBS datafile 'C:\tbs\SQL33_TBS.dat' size 100M autoextend on online;

Tablespace created.
```

```
Sélection Run SQL Command Line

SQL> create temporary tablespace SQL33_TempTBS tempfile 'C:\tbs\SQL33_TempTBS.dat' size 100M autoextend on;

Tablespace created.

SQL>
```

### Description :

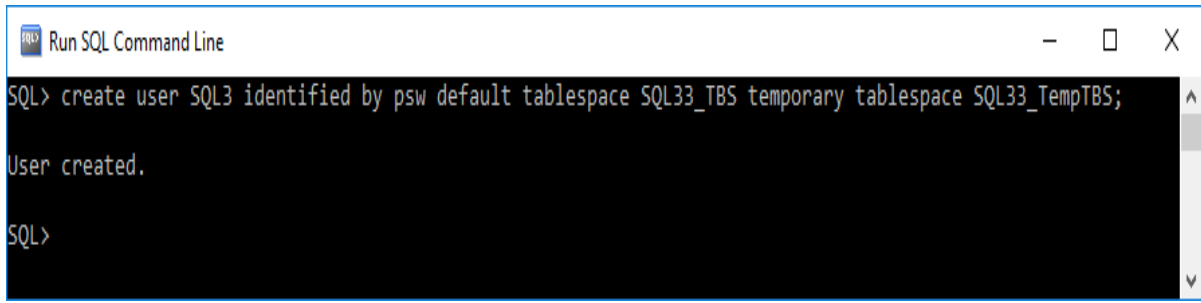
**CREATE TABLESPACE xxx\_tbs** Spécifie le nom du tablespace de la base.

**DATAFILE 'C:\tbs\_xxx.dat' SIZE 100M** Spécifie le nom complet du fichier système ainsi que sa taille en méga octets.

**AUTOEXTEND ON** La taille est augmentée automatiquement en cas de saturation

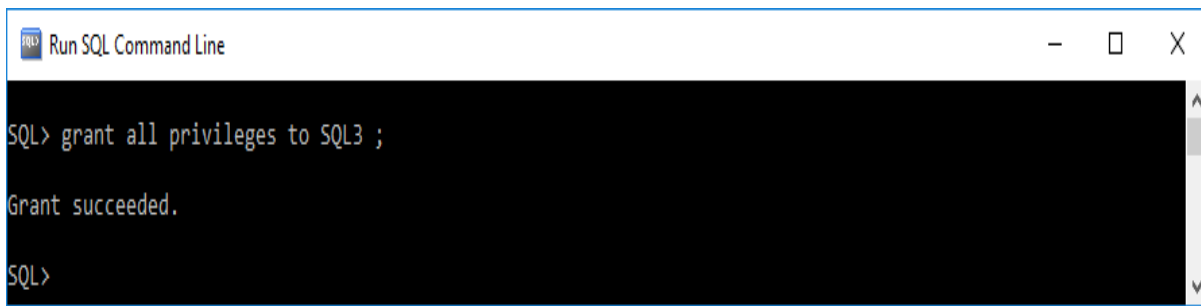
**ONLINE** Disponible immédiatement suite à sa création

### 2-2- Création de l'utilisateur SQL3 en lui attribuant les deux tablespaces créés précédemment :

A screenshot of a 'Run SQL Command Line' window. The window has a title bar with a blue icon and the text 'Run SQL Command Line'. The main area is a black terminal with white text. The text shows a SQL command to create a user, followed by a confirmation message and a new prompt.

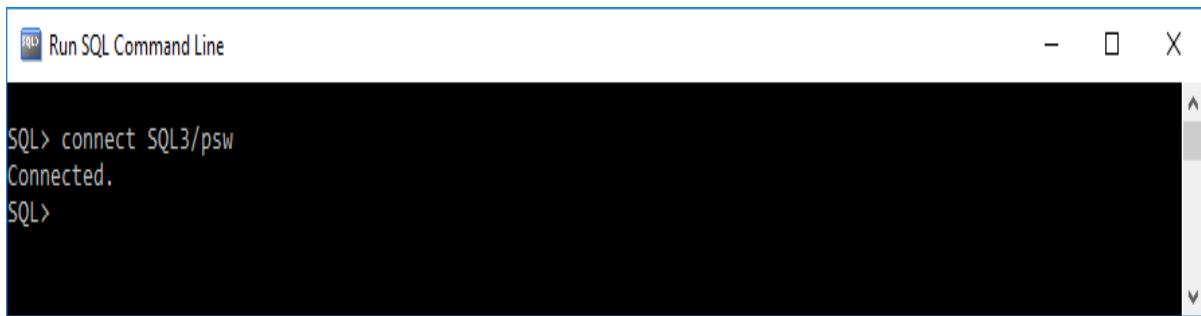
```
SQL> create user SQL3 identified by psw default tablespace SQL33_TBS temporary tablespace SQL33_TempTBS;  
User created.  
SQL>
```

### 2-3- Donner tous les privilèges à cet utilisateur:

A screenshot of a 'Run SQL Command Line' window. The window has a title bar with a blue icon and the text 'Run SQL Command Line'. The main area is a black terminal with white text. The text shows a SQL command to grant all privileges to a user, followed by a confirmation message and a new prompt.

```
SQL> grant all privileges to SQL3 ;  
Grant succeeded.  
SQL>
```

### 2-4- connecter au système avec cet utilisateur:

A screenshot of a 'Run SQL Command Line' window. The window has a title bar with a blue icon and the text 'Run SQL Command Line'. The main area is a black terminal with white text. The text shows a SQL command to connect to the system as a specific user, followed by a confirmation message and a new prompt.

```
SQL> connect SQL3/psw  
Connected.  
SQL>
```

## **Partie III : Langage de définition de données:**

### 3-1-définition des types incomplets nécessaires:

```
Run SQL Command Line
SQL> create type T_Sportif;
2 /
Type created.
SQL> create type T_Sport;
2 /
Type created.
SQL> create type T_Arbitrer;
2 /
Type created.
SQL> create type T_Jouer;
2 /
Type created.
SQL> create type T_Entraîner;
2 /
Type created.
SQL> create type T_ville;
2 /
Type created.
SQL> create type T_Gymnass;
2 /
Type created.
SQL> create type T_Seance;
2 /
Type created.
```

### explication :

tous les types qu'on a créés sont de type incomplet car tous les classes ont au moins une association entre eux

### 3-1-1- définition des types définis par utilisateur:

-----> table sportif :

**sportif\_arbitrer** --> collection de références des objets de type T\_Arbitrer

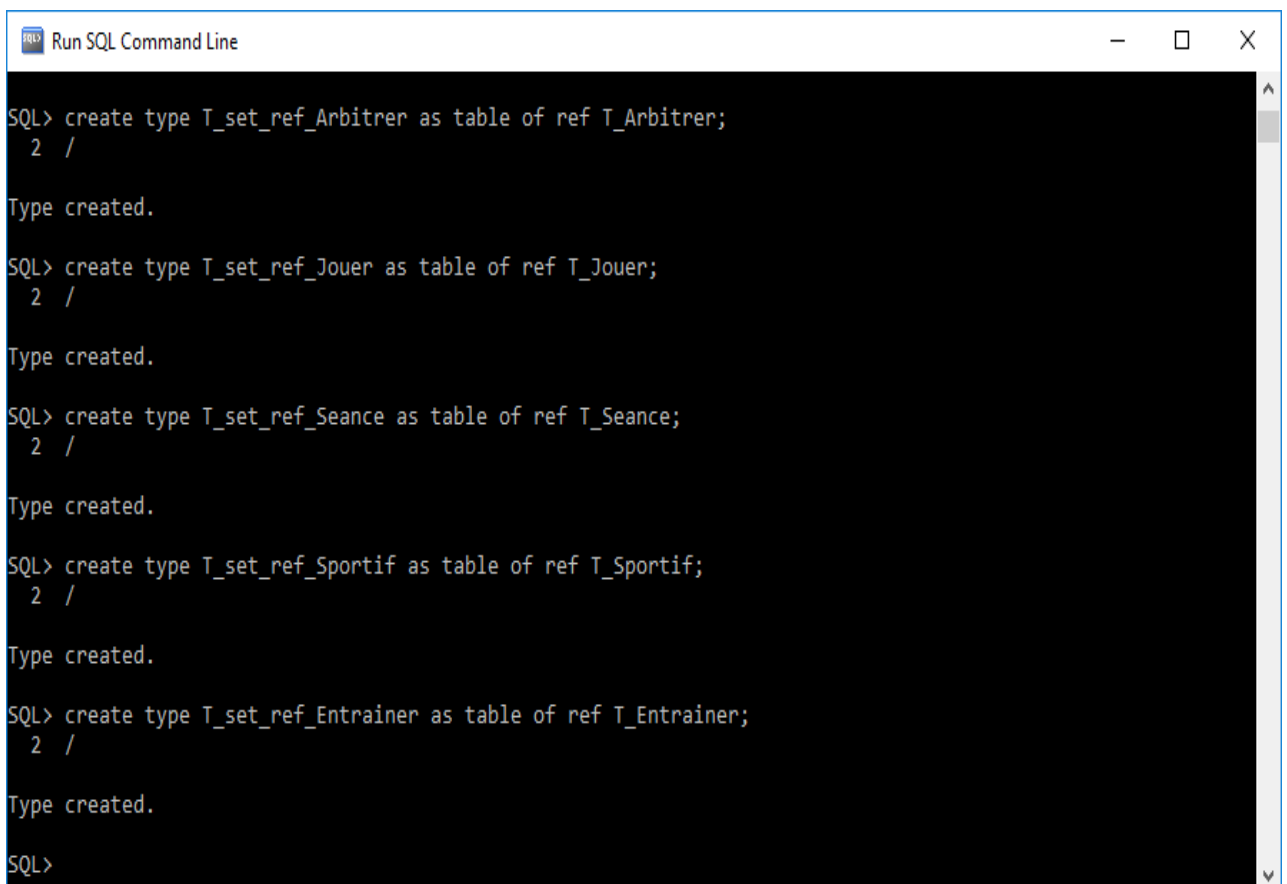
**sportif\_jouer** --> collection de références des objets de type T\_Jouer

**est\_conseillé** --> collection de références des objets de type  
T\_Sportif

**sportif\_seance** --> collection de références des objets de type  
T\_Seance

**sportif\_Entraîner** --> collection de références des objets de type  
T\_Entraîner

**sportif\_Sport** --> collection de références des objets de type  
T\_Sport

A screenshot of a 'Run SQL Command Line' window with a black background and white text. The window title bar shows a small icon and the text 'Run SQL Command Line'. The command prompt shows five SQL commands to create table types, each followed by a confirmation message. The commands are: 1. 'create type T\_set\_ref\_Arbitrer as table of ref T\_Arbitrer;' followed by '2 /' and 'Type created.'; 2. 'create type T\_set\_ref\_Jouer as table of ref T\_Jouer;' followed by '2 /' and 'Type created.'; 3. 'create type T\_set\_ref\_Seance as table of ref T\_Seance;' followed by '2 /' and 'Type created.'; 4. 'create type T\_set\_ref\_Sportif as table of ref T\_Sportif;' followed by '2 /' and 'Type created.'; 5. 'create type T\_set\_ref\_Entraîner as table of ref T\_Entraîner;' followed by '2 /' and 'Type created.'. The prompt 'SQL>' is visible at the end of the last command and at the bottom of the window.

```
SQL> create type T_set_ref_Arbitrer as table of ref T_Arbitrer;
2 /
Type created.

SQL> create type T_set_ref_Jouer as table of ref T_Jouer;
2 /
Type created.

SQL> create type T_set_ref_Seance as table of ref T_Seance;
2 /
Type created.

SQL> create type T_set_ref_Sportif as table of ref T_Sportif;
2 /
Type created.

SQL> create type T_set_ref_Entraîner as table of ref T_Entraîner;
2 /
Type created.

SQL>
```

```
Run SQL Command Line
SQL> create type T_set_ref_sport as table of ref T_Sport;
2 /
```

```
Run SQL Command Line
SQL> create type T_set_ref_Gymnass as table of ref T_Gymnass;
2 /
Type created.
```

-----> Mise à jour de type incomplet T\_Sportif :

**sportif\_arbitrer** --> attribut référence de type T\_Arbitrer

**sportif\_jouer** --> collection de références des objets de type T\_Jouer

**sportif\_seance** --> collection de références des objets de type T\_Seance

**sportif\_Entrainer** --> collection de références des objets de type T\_Entrainer

**est\_Conseille** --> collection de références des objets de type T\_Sportif.



```
Run SQL Command Line

SQL> create or replace type T_Sportif as object (
  2   IDSPORTIF number,
  3   Nom varchar2(50),
  4   prenom varchar2(50),
  5   sexe varchar2(50),
  6   Age number,
  7   sportif_arbitrer T_set_ref_Arbitrer,
  8   sportif_jouer T_set_ref_Jouer,
  9   est_conseile T_set_ref_Sportif,
 10   sportif_seance T_set_ref_Seance,
 11   sportif Entraîner T_set_ref Entraîner
 12 );
 13 /

Type created.

SQL>
```

**nb :** j'ai oublié d'ajouter la collection **Sportif\_sport** qui est une collection de référence de type T\_sport

```
Run SQL Command Line

SQL> alter type T_Sportif add attribut sportif_Sport T_set_ref_Sport CASCADE;
```

```
Run SQL Command Line

Name                                Null?    Type
-----
IDSPORTIF                          NOT NULL NUMBER
NOM                                VARCHAR2(50)
PRENOM                             VARCHAR2(50)
SEXE                               VARCHAR2(50)
AGE                                NUMBER
SPORTIF_ARBITRER                   T_SET_REF_ARBITRER
SPORTIF_JOUER                      T_SET_REF_JOUER
EST_CONSEILE                       T_SET_REF_SPORTIF
SPORTIF_SEANCE                     T_SET_REF_SEANCE
SPORTIF_ENTRAÎNER                  T_SET_REF_ENTRAÎNER
SPORTIF_SPORT                      T_SET_REF_SPORT

SQL>
```

-----> table sport :

**sport\_arbitrer** --> attribut référence de type T\_Arbitrer

**sport\_jouer** --> collection de références des objets de type T\_Jouer

**sport\_seance** --> collection de références des objets de type T\_Seance

**sport\_Entraîner** --> collection de références des objets de type T\_Entraîner

**sport\_Gymnass** --> collection de références des objets de type T\_Gymnass.

**sport\_sportif** --> collection de références des objets de type T\_Sportif.

**nb :** tous ces types de collection on a défini déjà au niveau de la table Sportif donc on les utilisent directement

-----> Mise à jour de type incomplet T\_Sport :

```
Run SQL Command Line

SQL> create or replace type T_Sport as object (
  2   IDSPORT number,
  3   Libelle varchar2(50),
  4   sport_arbitrer ref T_Arbitrer,
  5   sport_jouer    T_set_ref_Jouer,
  6   sport_seance   T_set_ref_Seance,
  7   sport_Entraîner T_set_ref_Entraîner
  8 );
  9 /

Type created.

SQL>
```

**nb** : j'ai oublié la collection **sport\_sportif** et la collection **sport\_gymnass**

```
Run SQL Command Line

SQL> alter type T_Sport add attribute sport_Sportif T_set_ref_sportif CASCADE;

Type altered.

SQL>
```

```
Run SQL Command Line

SQL> alter type T_sport add attribute sport_gymnass T_set_ref_gymnass cascade;
```

| Name            | Null?    | Type                |
|-----------------|----------|---------------------|
| IDSPO           | NOT NULL | NUMBER              |
| LIBELLE         |          | VARCHAR2(50)        |
| SPORT_ARBITRER  |          | REF OF T_ARBITRER   |
| SPORT_JOUER     |          | T_SET_REF_JOUER     |
| SPORT_SEANCE    |          | T_SET_REF_SEANCE    |
| SPORT_ENTRAINER |          | T_SET_REF_ENTRAINER |
| SPORT_GYMNASS   |          | T_SET_REF_GYMNASS   |
| SPORT_SPORTIF   |          | T_SET_REF_SPORTIF   |

SQL>

-----> table Arbitrer :

**arbitrer\_sportif** --> collection de références des objets de type T\_Sportif

**arbitrer\_sport** --> collection de références des objets de type T\_Sport

**nb:** le type collection de T\_Sportif on a défini déjà au niveau de la table Sportif donc on va l'utiliser directement. par contre le type collection de T\_Sport on va le définir

```
SQL> create type T_set_ref_Sport as table of ref T_Sport;
2 /

Type created.

SQL>
```

-----> Mise à jour de type incomplet T\_Arbitrer:

```
Run SQL Command Line

SQL> create or replace type T_Arbitrer as object(
  2   arbitrer_sportif      T_set_ref_Sportif,
  3   arbitrer_sport       T_set_ref_Sport
  4 );
  5 /

Type created.

SQL>
```

```
Run SQL Command Line

SQL> desc T_Arbitrer;
Name                                Null?    Type
-----
ARBITRER_SPORTIF                    T_SET_REF_SPORTIF
ARBITRER_SPORT                      T_SET_REF_SPORT

SQL>
```

**nb:** il faut qu'on déconnecte pour qu'on puissent afficher la description de type T\_Arbitrer

-----> table Jouer :

**jouer\_sportif** --> collection de références des objets de type T\_Sportif

**jouer\_sport** --> collection de références des objets de type T\_Sport

**nb:** le type collection de T\_Sportif on a défini déjà au niveau de la table Sportif donc on va l'utiliser directement. et la même chose pour le type collection T\_Sport on a défini au niveau de la table Arbitrer.

-----> Mise à jour de type icomplet T\_Jouer :

```
SQL> create or replace type T_Jouer as object(
  2   joueur_sportif      T_set_ref_Sportif,
  3   joueur_sport        T_set_ref_Sport
  4 );
  5 /

Type created.
```

```
SQL> desc T_Jouer;
Name                                     Null?    Type
-----
JOUER_SPORTIF                           T_SET_REF_SPORTIF
JOUER_SPORT                              T_SET_REF_SPORT
SQL>
```

-----> table Entraîner :

**entraîner\_sportif** --> référence des objets de type T\_Sportif

**entraîner\_sport** --> collection de références des objets de type T\_Sport

**nb:** le type collection de T\_Sport on a défini déjà au niveau de la table Arbitrer donc on va l'utiliser directement

-----> Mise à jour de type icomplet T\_Entraîner :

```
Run SQL Command Line

SQL> create or replace type T_Entrainer as object(
  2   entrainer_sportif      T_set_ref_Sportif,
  3   entrainer_sport       T_set_ref_Sport
  4 );
  5 /

Type created.
```

```
Run SQL Command Line

SQL> alter type T_Entrainer add attribute entrainer_sportif ref T_Sportif cascade;

Type altered.
```

```
Run SQL Command Line

SQL> desc T_Entrainer;
Name                                Null?    Type
-----
ENTRAINER_SPORT                    T_SET_REF_SPORT
ENTRAINER_SPORTIF                  REF OF T_SPORTIF

SQL>
```

-----> table Ville :

ville\_gymnass --> collection de références des objets de type T\_Gymnass.

-----> Mise à jour de type icomplet T\_ville :

```
Run SQL Command Line

SQL> create or replace type T_ville as object(
2   ville          varchar2(50),
3   ville_gymnass   T_set_ref_Gymnass
4 );
5 /

Type created.
```

```
Run SQL Command Line

SQL> desc T_ville;
Name                                Null?     Type
-----
VILLE                             V
VILLE_GYMNASS                     T_SET_REF_GYMNASS

SQL>
```

-----> table Gymnass :

**gymnass\_ville** --> attribut référence de type T\_ville

**gymnass\_seance** --> collection de références des objets de type T\_Seance

**nb:** le type collection de T\_Seance on a défini déjà au niveau de la table Sportif donc on va l'utiliser directement.

-----> Mise à jour de type icomplet T\_Gymnass :



```
Run SQL Command Line
SQL> desc T_Gymnass;
Name                               Null?   Type
-----
IDGYMNAISE                         NUMBER
NOMGYMNAISE                        VARCHAR2(50)
ADRESSE                           VARCHAR2(50)
SURFACE                           FLOAT(20)
GYMNASS_VILLE                      REF OF T_VILLE
GYMNASS_SEANCE                     T_SET_REF_SEANCE
SQL>
```

-----> table Seance :

Seance\_Sportif --> attribut référence de type T\_Sportif

Seance\_gymnass --> attribut référence de type  
T\_Gymnass

Seance\_Sport --> attribut référence de type T\_Sport

-----> Mise à jour de type incomplet T\_Seance :

```
Run SQL Command Line
SQL> create or replace type T_Seance as object(
2   Jour          varchar2(50),
3   Horaire       FLOAT(20),
4   Duree         number,
5   Seance_Sportif ref T_Sportif,
6   Seance_gymnass ref T_Gymnass,
7   Seance_Sport  ref T_Sport
8 );
9 /
Type created.
```

```
Run SQL Command Line

SQL> desc T_Seance;
Name                                Null?    Type
-----
JOUR                                VARCHA2(50)
HORAIRE                             FLOAT(20)
DUREE                               NUMBER
SEANCE_SPORTIF                      REF OF T_SPORTIF
SEANCE_GYMNASS                      REF OF T_GYMNASS
SEANCE_SPORT                        REF OF T_SPORT

SQL>
```

### 3-2-définition des méthodes aux types:

-----> **3-2-1: la méthode nbr sport entraînés:**

**nb:** cet méthode on va la mettre au type T\_Sportif

```
Run SQL Command Line

SQL> alter type T_Sportif add member function nbr_sports_entraines return number cascade;

Type altered.
```

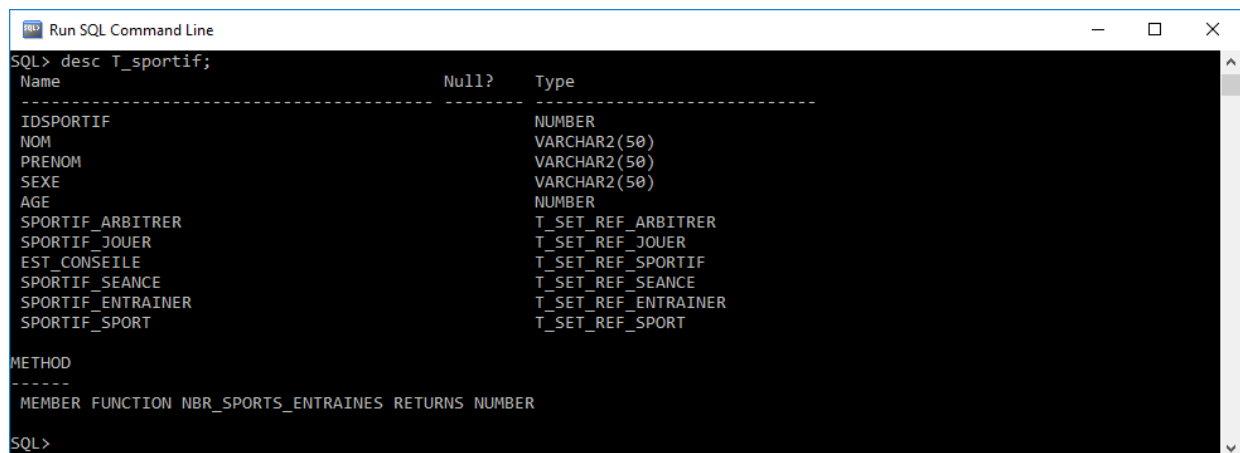
```
Run SQL Command Line

SQL> create or replace type body T_Sportif as
2   member function nbr_sports_entraines return number is -- Ajout du "is" ici
3   nbr_sports_entr number;
4   begin
5   select count(deref(value(T1))) into nbr_sports_entr
6   from sportif s, table(s.sportif_Sport) T1
7   where s.IDSPORTIF = self.IDSPORTIF;
8   return nbr_sports_entr;
9   end nbr_sports_entraines;
10 end;
11 /

Type body created.

SQL>
```

**nb:** il faut qu'on déconnecte pour qu'on puissent afficher la méthode



```
Run SQL Command Line
SQL> desc T_sportif;
Name                                Null?    Type
-----
IDSPORTIF                           NUMBER
NOM                                  VARCHAR2(50)
PRENOM                               VARCHAR2(50)
SEXE                                 VARCHAR2(50)
AGE                                  NUMBER
SPORTIF_ARBITRER                     T_SET_REF_ARBITRER
SPORTIF_JOUER                         T_SET_REF_JOUER
EST_CONSEIL                           T_SET_REF_SPORTIF
SPORTIF_SEANCE                       T_SET_REF_SEANCE
SPORTIF_ENTRAINER                     T_SET_REF_ENTRAINER
SPORTIF_SPORT                        T_SET_REF_SPORT

METHOD
-----
MEMBER FUNCTION NBR_SPORTS_ENTRAINES RETURNS NUMBER

SQL>
```

### explication :

Le code que nous avons fourni ajoute une nouvelle méthode "nbr\_sports\_entraines" au type T\_Sportif en utilisant l'instruction "alter type" et définit le corps de cette méthode en utilisant l'instruction "create or replace type body".

Cette requête crée ou remplace le corps (type body) d'un type utilisateur (user-defined type) nommé "T\_Sportif". Le type T\_Sportif a une fonction membre (member function) nommée "nbr\_sports\_entraines" qui renvoie le nombre de sports entraînés par le sportif correspondant à l'instance actuelle du type.

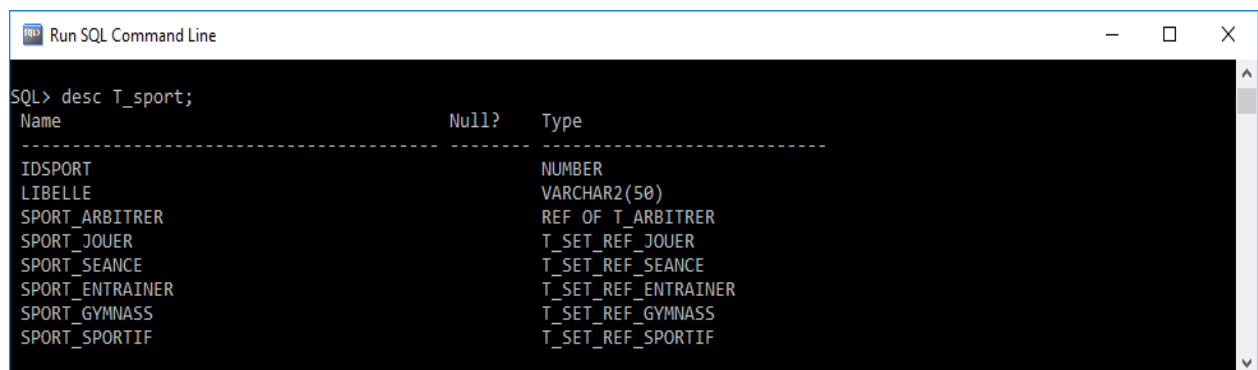
La fonction membre utilise une requête SQL pour compter le nombre de sports entraînés par le sportif. La requête sélectionne tous les éléments de la table "sportif\_Sport" du sportif actuel,

puis compte le nombre de résultats de cette sélection. Le résultat est stocké dans la variable "nbr\_sports\_entr" et renvoyé par la fonction.

-----> 3-2-2: la méthode nbr\_gymnass\_sport:

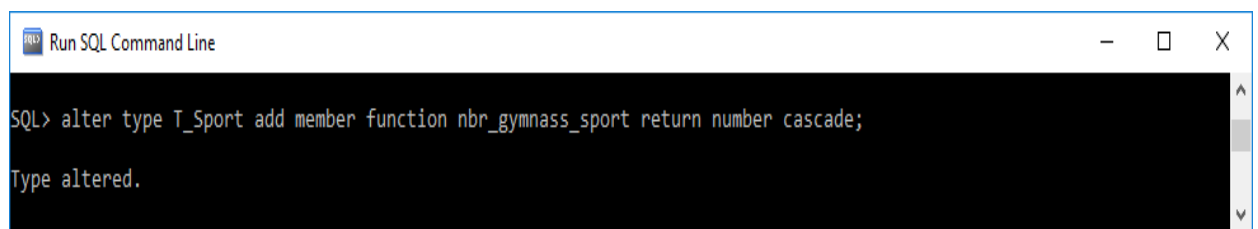
d'abord on va ajouter l'attribut **sport\_gymnass** qui est une collection de références de type T\_Gymnass.

**nb:** cet méthode on va la mettre au type T\_Sport



```
SQL> desc T_sport;
```

| Name            | Null? | Type                |
|-----------------|-------|---------------------|
| IDSPO           |       | NUMBER              |
| LIBELLE         |       | VARCHAR2(50)        |
| SPORT_ARBITRER  |       | REF OF T_ARBITRER   |
| SPORT_JOUER     |       | T_SET_REF_JOUER     |
| SPORT_SEANCE    |       | T_SET_REF_SEANCE    |
| SPORT_ENTRAINER |       | T_SET_REF_ENTRAINER |
| SPORT_GYMNASS   |       | T_SET_REF_GYMNASS   |
| SPORT_SPORTIF   |       | T_SET_REF_SPORTIF   |



```
SQL> alter type T_Sport add member function nbr_gymnass_sport return number cascade;
```

Type altered.

```
Run SQL Command Line
SQL> create or replace type body T_Sport as
2   member function nbr_gymnass_sport return number is -- Ajout du "is" ici
3   nbr_gym_sport number;
4   begin
5   select count(deref(value(t2))) into nbr_gym_sport
6   from sport s, table(s.sport_gymnass) t2
7   where s.IDSPORT = self.IDSPORT;
8   return nbr_gym_sport;
9   end nbr_gymnass_sport;
10 end;
11 /
Type body created.
SQL>
```

**nb:** il faut qu'on déconnecte pour qu'on puissent afficher la méthode.

```
Run SQL Command Line
SQL> desc T_sport;
Name                                Null?    Type
-----
IDSPORT                             NUMBER
LIBELLE                             VARCHAR2(50)
SPORT_ARBITRER                       REF OF T_ARBITRER
SPORT_JOUER                          T_SET_REF_JOUER
SPORT_SEANCE                         T_SET_REF_SEANCE
SPORT_ENTRAINER                     T_SET_REF_ENTRAINER
SPORT_GYMNASS                       T_SET_REF_GYMNASS
SPORT_SPORTIF                       T_SET_REF_SPORTIF

METHOD
-----
MEMBER FUNCTION NBR_GYMNASS_SPORT RETURNS NUMBER
SQL>
```

### explication :

Le code que nous avons fourni ajoute une nouvelle méthode "nbr\_gymnass\_sport" au type T\_Sport en utilisant l'instruction "alter type" et définit le corps de cette méthode en utilisant l'instruction "create or replace type body".

Cette requête crée ou remplace le corps (type body) d'un type

utilisateur (user-defined type) nommé "T\_Sport". Le type T\_Sport a une fonction membre (member function) nommée "nbr\_gymnass\_sport" qui renvoie le nombre de gymnases dans lesquels le sport correspondant à l'instance actuelle du type est pratiqué.

La fonction membre utilise une requête SQL pour compter le nombre de gymnases dans lesquels le sport est pratiqué. La requête sélectionne tous les éléments de la table "sport\_gymnass" pour le sport actuel, puis compte le nombre de résultats de cette sélection. Le résultat est stocké dans la variable "nbr\_gym\_sport" et renvoyé par la fonction.

-----> 3-2-3: la méthode calcul\_sup\_moy\_gymnass:

**nb:** cet méthode on va la mettre au type T\_Ville

```
Run SQL Command Line
SQL> alter type T_ville add member function calcul_sup_moy_gymnass return float cascade;
Type altered.
```

```
Run SQL Command Line
SQL> create or replace type body T_ville as
2  member function calcul_sup_moy_gymnass return float is -- Ajout du "is" ici
3      sup_moy_gym float;
4  begin
5      select AVG(deref(value(t1)).surface) into sup_moy_gym
6      from ville v, table(v.ville_gymnass) t1
7      where v.ville = self.ville;
8      return sup_moy_gym;
9  end calcul_sup_moy_gymnass;
10 end;
11 /
Type body created.
```

**nb:** il faut qu'on déconnecte pour qu'on puissent afficher la méthode.

```
Run SQL Command Line
SQL> desc T_ville;
Name                                Null?    Type
-----
VILLE                              VARCHA2(50)
VILLE_GYMNASS                      T_SET_REF_GYMNASS

METHOD
-----
MEMBER FUNCTION CALCUL_SUP_MOY_GYMNASS RETURNS NUMBER

SQL>
```

### explication :

Le code que nous avons fourni ajoute une nouvelle méthode "calcul\_sup\_moy\_gymnass" au type T\_ville en utilisant l'instruction "alter type" et définit le corps de cette méthode en utilisant l'instruction "create or replace type body".

Cette requête crée ou remplace le corps de la méthode membre calcul\_sup\_moy\_gymnass de l'objet de type T\_ville. Cette méthode calcule la moyenne des surfaces des gymnases dans une

ville donnée en utilisant la clause AVG dans une requête SQL et retourne le résultat en tant que float. L'ajout du is est nécessaire pour indiquer que la définition de la fonction commence à cet endroit. Le / final est utilisé pour exécuter la commande dans SQL\*Plus ou SQL Developer.

### 3-3- Création des tables:

-----> table sportif :

```
Run SQL Command Line
SQL> create table Sportif of T_Sportif(primary key(IDSPORTIF),foreign key(Conseille) references Sportif)
2 nested table sportif_arbitrer store as table_sportif_arbitrer,
3 nested table sportif_jouer store as table_sportif_jouer,
4 nested table est_conseille store as table_est_conseille,
5 nested table sportif_seance store as table_sportif_seance,
6 nested table sportif_Entraîner store as table_sportif_Entraîner;

Table created.
```

```
Run SQL Command Line
SQL> desc Sportif;
Name                                Null?    Type
-----
IDSPORTIF                          NOT NULL NUMBER
NOM                                VARCHAR2(50)
PRENOM                             VARCHAR2(50)
SEXE                               VARCHAR2(50)
AGE                                NUMBER
SPORTIF_ARBITRER                   T_SET_REF_ARBITRER
SPORTIF_JOUER                      T_SET_REF_JOUER
EST_CONSEILLE                      T_SET_REF_SPORTIF
SPORTIF_SEANCE                    T_SET_REF_SEANCE
SPORTIF_ENTRAÎNER                 T_SET_REF_ENTRAÎNER
SPORTIF_SPORT                     T_SET_REF_SPORT
CONSEILLE                         REF OF T_SPORTIF
```

### explication:

on va créer une table "Sportif" qui utilise le type utilisateur "T\_Sportif" défini précédemment.



La clause "of T\_Sportif" indique que la table "Sportif" est basée sur le type utilisateur "T\_Sportif".

La colonne "IDSPORTIF" est déclarée comme la clé primaire de la table avec la clause "primary key(IDSPORTIF)".

La colonne "Conseille" est déclarée comme une clé étrangère qui fait référence à la table "Sportif" elle-même, en utilisant la syntaxe "foreign key(Conseille) references Sportif". Cela permet d'établir une relation de conseil entre deux sportifs.

En outre, cinq colonnes de type collection sont définies pour stocker des informations supplémentaires sur les sportifs. Ces collections sont "sportif\_arbitrer", "sportif\_jouer", "est\_conseile", "sportif\_seance", et "sportif\_Entrainer".

Les collections sont stockées en tant que tables imbriquées dans la table "Sportif". Chaque collection est associée à une table particulière qui est le store (stockage physique), et qui est créée en utilisant la syntaxe "nested table nom\_de\_collection store as nom\_de\_table".

-----> table Arbitrer :

```
Run SQL Command Line

SQL> create table Arbitrer of T_Arbitrer
  2  nested table arbitrer_sportif store as table_arbitrer_sportif,
  3  nested table arbitrer_sport store as table_arbitrer_sport;

Table created.

SQL>
```

```
Run SQL Command Line

SQL> desc Arbitrer;
Name                                Null?    Type
-----
ARBITRER_SPORTIF                    T_SET_REF_SPORTIF
ARBITRER_SPORT                      T_SET_REF_SPORT

SQL>
```

### explication:

on va créer une table "Arbitrer" qui utilise le type utilisateur "T\_Arbitrer" défini précédemment.

La clause "of T\_Arbitrer" indique que la table "Arbitrer" est basée sur le type utilisateur "T\_Arbitrer".

En outre, deux colonnes de type collection sont définies pour stocker des informations supplémentaires sur les Arbitres. Ces collections sont "arbitrer\_sportif", "arbitrer\_sport".

Les collections sont stockées en tant que tables imbriquées dans

la table "Arbitrer". Chaque collection est associée à une table particulière qui est le store (stockage physique), et qui est créée en utilisant la syntaxe "nested table nom\_de\_collection store as nom\_de\_table".

-----> table Sport :

```
Run SQL Command Line

SQL> create table Sport of T_Sport(primary key(IDSPORT),foreign key(sport_arbitrer) references Arbitrer)
  2 nested table sport_jouer store as table_sport_jouer,
  3 nested table sport_seance store as table_sport_seance,
  4 nested table sport Entraîner store as table_sport Entraîner,
  5 nested table SPORT_GYMNASS store as table_sport_gymnass;

Table created.

SQL>
```

```
Run SQL Command Line

SQL> desc Sport;
Name                               Null?   Type
-----
IDSPORT                            NOT NULL NUMBER
LIBELLE                            VARCHAR2(50)
SPORT_ARBITRER                     REF OF T_ARBITRER
SPORT_JOUER                        T_SET_REF_JOUER
SPORT_SEANCE                       T_SET_REF_SEANCE
SPORT_ENTRAÎNER                   T_SET_REF_ENTRAÎNER
SPORT_GYMNASS                      T_SET_REF_GYMNASS
SPORT_SPORTIF                     T_SET_REF_SPORTIF

SQL>
```

### explication:

on va créer une table "Sport" qui utilise le type utilisateur "T\_Sport" défini précédemment.

La clause "of T\_Sport" indique que la table "Sport" est basée sur le type utilisateur "T\_Sport".

La colonne "IDSPORT" est déclarée comme la clé primaire de la table avec la clause "primary key(IDSPORT)".

La colonne "sport\_arbitrer" est déclarée comme une clé étrangère qui fait référence à la table "Arbitrer", en utilisant la syntaxe "foreign key(sport\_arbitrer) references Arbitrer". Cela permet d'établir une relation entre la table Arbitrer et Sport.

En outre, quatre colonnes de type collection sont définies pour stocker des informations supplémentaires sur les sports. Ces collections sont "sport\_jouer", "sport\_seance", "sport\_entrainer", et "sport\_gymnass".

Les collections sont stockées en tant que tables imbriquées dans la table "Sport". Chaque collection est associée à une table particulière qui est le store (stockage physique), et qui est créée en utilisant la syntaxe "nested table nom\_de\_collection store as nom\_de\_table".

-----> table Jouer :

```
Run SQL Command Line
SQL>
SQL> create table Jouer of T_Jouer
  2 nested table jouer_sport store as table_jouer_sport,
  3 nested table jouer_sportif store as table_jouer_sportif;

Table created.
```

```
Run SQL Command Line
SQL> desc Jouer;
Name                                Null?    Type
-----
JOUER_SPORTIF                       T_SET_REF_SPORTIF
JOUER_SPORT                         T_SET_REF_SPORT
SQL>
```

### explication:

on va créé une table "Jouer" qui utilise le type utilisateur "T\_Jouer" défini précédemment.

La clause "of T\_Jouer" indique que la table "Jouer" est basée sur le type utilisateur "T\_Jouer".

En outre, deux colonnes de type collection sont définies pour stocker des informations supplémentaires sur les Joueurs. Ces collections sont "jouer\_sport", "jouer\_sportif".

Les collections sont stockées en tant que tables imbriquées dans

la table "Jouer". Chaque collection est associée à une table particulière qui est le store (stockage physique), et qui est créée en utilisant la syntaxe "nested table nom\_de\_collection store as nom\_de\_table".

-----> table Entrainer :

```
Run SQL Command Line
SQL> create table Entrainer of T_Entrainer(foreign key(entrainer_sportif) references Sportif)
  2  nested table entrainer_sport store as table_entrainer_sport;

Table created.

SQL>
```

```
Run SQL Command Line
SQL> desc entrainer;
Name                                Null?    Type
-----
ENTRAINER_SPORT                     T_SET_REF_SPORT
ENTRAINER_SPORTIF                   REF OF T_SPORTIF

SQL>
```

### explication:

on va créer une table "Entrainer" qui utilise le type utilisateur "T\_Entrainer" défini précédemment.

La clause "of T\_Entrainer" indique que la table "Entrainer" est

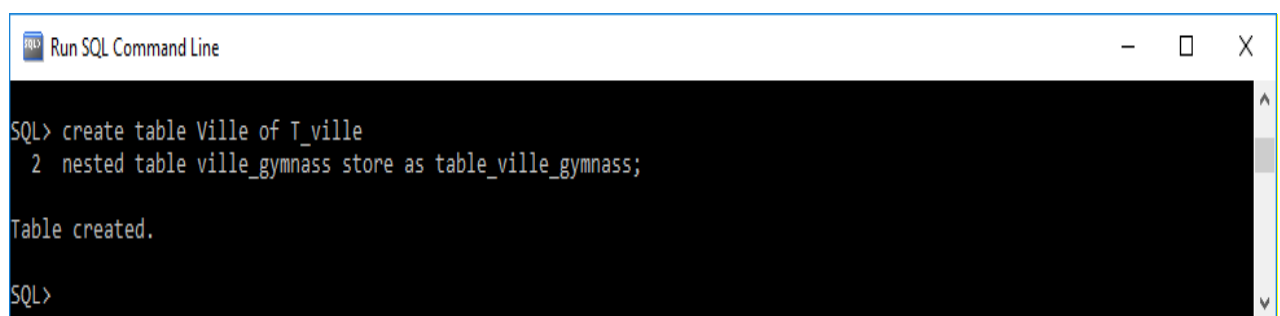
basée sur le type utilisateur "T\_Entrainer".

La colonne "Entrainer\_sportif" est déclarée comme une clé étrangère qui fait référence à la table "Sportif", en utilisant la syntaxe "foreign key(Entrainer\_sportif) references Sportif". Cela permet d'établir une relation entre la table Entrainer et Sportif.

En outre, une seule colonne de type collection est définie pour stocker des informations supplémentaires sur les Entraîneurs. Cette collection est "Entrainer\_sport".

La collection est stockée en tant que table imbriquée dans la table "Entrainer". La collection est associée à une table particulière qui est le store (stockage physique), et qui est créée en utilisant la syntaxe "nested table nom\_de\_collection store as nom\_de\_table".

-----> table ville :

A screenshot of a SQL Command Line window titled "Run SQL Command Line". The window has a black background with white text. The SQL command entered is "create table Ville of T\_ville" followed by an indented line "2 nested table ville\_gymnass store as table\_ville\_gymnass;". The response from the database is "Table created." followed by a new prompt "SQL>".

```
SQL> create table Ville of T_ville
      2 nested table ville_gymnass store as table_ville_gymnass;

Table created.

SQL>
```

```
Run SQL Command Line
SQL> desc Ville;
Name                               Null?    Type
-----
VILLE                             VARCHA2(50)
VILLE_GYMNASS                     T_SET_REF_GYMNASS
SQL>
```

### explication:

on va créer une table "Ville" qui utilise le type utilisateur "T\_Ville" défini précédemment.

La clause "of T\_Ville" indique que la table "Ville" est basée sur le type utilisateur "T\_Ville".

En outre, une seule colonne de type collection est définie pour stocker des informations supplémentaires sur les Entraîneurs. Cette collection est "Ville\_gymnass".

La collection est stockée en tant que table imbriquée dans la table "Ville". La collection est associée à une table particulière qui est le store (stockage physique), et qui est créée en utilisant la syntaxe "nested table nom\_de\_collection store as nom\_de\_table".

-----> table Gymnass :



**nb**: on va ajouter l'attribut **gymnass\_sport** qui est une collection de référence de type Sport car j'ai fait une mise à jour au niveau de diagramme de classe lors de la création de la méthode **nbr\_gym\_sport**.

```
Run SQL Command Line

SQL> alter type T_Gymnass add attribute gymnass_sport T_set_ref_Sport cascade;

Type altered.
```

```
Run SQL Command Line

SQL> desc T_Gymnass;
Name                                Null?    Type
-----
IDGYMNASE                           NUMBER
NOMGYMNASE                          VARCHAR2(50)
ADRESSE                             VARCHAR2(50)
SURFACE                             FLOAT(20)
GYMNASS_VILLE                       REF OF T_VILLE
GYMNASS_SEANCE                      T_SET_REF_SEANCE
GYMNASS_SPORT                       T_SET_REF_SPORT

SQL>
```

----- et maintenant on va créer la table -----

```
Run SQL Command Line

SQL> create table Gymnass of T_Gymnass(primary key(IDGYMNASE),foreign key(gymnass_ville) references Ville)
2 nested table gymnass_seance store as table_gymnass_seance,
3 nested table gymnass_sport store as table_gymnass_sport;

Table created.

SQL>
```

```
Run SQL Command Line

SQL> desc Gymnass;

Name                               Null?   Type
-----
IDGYMNASE                          NOT NULL NUMBER
NOMGYMNASE                         VARCHAR2(50)
ADRESSE                           VARCHAR2(50)
SURFACE                           FLOAT(20)
GYMNASS_VILLE                     REF OF T_VILLE
GYMNASS_SEANCE                    T_SET_REF_SEANCE
GYMNASS_SPORT                     T_SET_REF_SPORT

SQL>
```

### explication:

on va créer une table "Gymnass" qui utilise le type utilisateur "T\_Gymnass" défini précédemment.

La clause "of T\_Gymnass" indique que la table "Gymnass" est basée sur le type utilisateur "T\_Gymnass".

La colonne "IDGYMNASE" est déclarée comme la clé primaire de la table avec la clause "primary key(IDGYMNASE)".

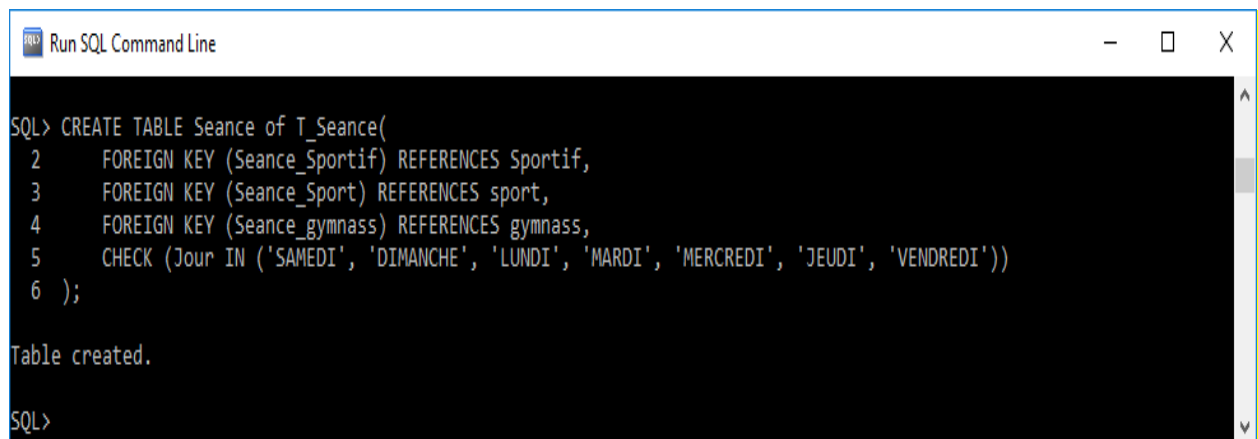
La colonne "gymnass\_ville" est déclarée comme une clé étrangère qui fait référence à la table "Ville", en utilisant la syntaxe "foreign

key(gymnass\_ville) references Ville". Cela permet d'établir une relation entre la table Gymnass et Ville.

En outre, deux colonnes de type collections sont définies pour stocker des informations supplémentaires sur les Salles. Ces collections sont "gymnass\_seance" et "gymnass\_seance".

Les collections est stockées en tant que tables imbriquées dans la table "Gymnass". Les collection sont associées à des tables particulières qui est le store (stockage physique), et qui est créée en utilisant la syntaxe "nested table nom\_de\_collection store as nom\_de\_table".

-----> table Seance :



```
Run SQL Command Line

SQL> CREATE TABLE Seance of T_Seance(
2     FOREIGN KEY (Seance_Sportif) REFERENCES Sportif,
3     FOREIGN KEY (Seance_Sport) REFERENCES sport,
4     FOREIGN KEY (Seance_gymnass) REFERENCES gymnass,
5     CHECK (Jour IN ('SAMEDI', 'DIMANCHE', 'LUNDI', 'MARDI', 'MERCREDI', 'JEUDI', 'VENDREDI'))
6 );

Table created.

SQL>
```

```
Run SQL Command Line
SQL> desc Seance;
Name                               Null?    Type
-----
JOUR                               VARCHA2(50)
HORAIRE                            FLOAT(20)
DUREE                              NUMBER
SEANCE_SPORTIF                     REF OF T_SPORTIF
SEANCE_GYMNASS                     REF OF T_GYMNASS
SEANCE_SPORT                       REF OF T_SPORT
SQL>
```

### explication:

on va créer une table "Seance" qui utilise le type utilisateur "T\_Seance" défini précédemment.

La clause "of T\_Seance" indique que la table "Seance" est basée sur le type utilisateur "T\_Seance".

La colonne "seance\_sport" est déclarée comme une clé étrangère qui fait référence à la table "sport", en utilisant la syntaxe "foreign key(seance\_sport) references sport". Cela permet d'établir une relation entre la table seance et Sport.

La colonne "seance\_sportif" est déclarée comme une clé étrangère qui fait référence à la table "Sportif", en utilisant la syntaxe "foreign key(seance\_sportif) references Sportif". Cela permet d'établir une relation entre la table seance et Sportif

La colonne "seance\_gymnass" est déclarée comme une clé étrangère qui fait référence à la table "gymnass", en utilisant la syntaxe "foreign key(gymnass) references gymnass". Cela permet

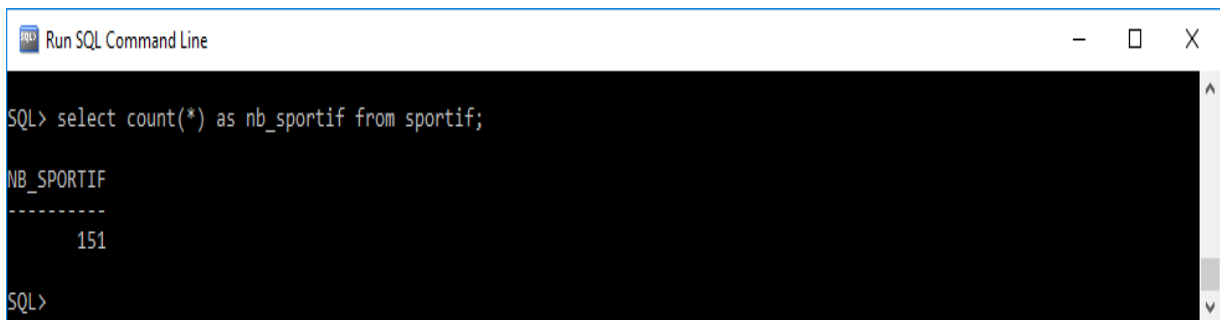
d'établir une relation entre la table seance et gymnass

on a ajouter une contrainte de type check qui verifier que l'attribut jour est dans les jours de la semaine.

## **Partie VI : Langage de manipulation de données**

### 1-1- les insertions:

-----> table sportif :



```
Run SQL Command Line
SQL> select count(*) as nb_sportif from sportif;

NB_SPORTIF
-----
          151

SQL>
```

**nb:** le code sql des insertions il est au niveau de fichier :

----- CodeSql3\_projet\_bda -----

### explication:

----> on va prendre un exemple:

```
Run SQL Command Line

SQL> INSERT INTO Sportif VALUES(
2     154, 'BOUTAHAR', 'Abderahim', 'M', 30,
3     T_set_ref_Arbitrer(),
4     T_set_ref_Jouer(),
5     T_set_ref_Sportif(
6         (select ref(s) from Sportif s where s.IDSPORTIF = 2),
7         (select ref(s) from Sportif s where s.IDSPORTIF = 3),
8         (select ref(s) from Sportif s where s.IDSPORTIF = 4)
9     ),
10    T_set_ref_Seance(),
11    T_set_ref_Entraîner(),
12    T_set_ref_Sport(),
13    NULL
14 );

1 row created.

SQL>
```

Cette requête SQL insère une nouvelle ligne dans la table "Sportif" avec les valeurs suivantes:

IDSPORTIF = 154

NOM = 'BOUTAHAR

PRENOM = 'Abderahim'

SEXE = 'M'

AGE = 30

Les valeurs pour les colonnes suivantes sont des appels de fonction:

T\_set\_ref\_Arbitrer() : renvoie les références à une table liant des arbitres à des matchs

T\_set\_ref\_Jouer() : renvoie les référence à une table liant des sportifs à des matchs

T\_set\_ref\_Sportif() : renvoie les référence à une table liant des sportifs à d'autres sportifs (en l'occurrence, les IDSPORTIF 2, 3 et 4)

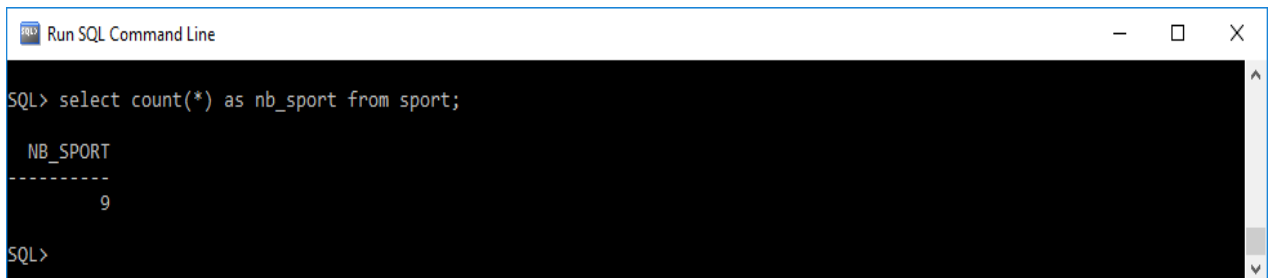
T\_set\_ref\_Seance() : renvoie les référence à une table liant des séances d'entraînement à des sportifs

T\_set\_ref\_Entraîner() : renvoie les référence à une table liant des entraîneurs à des sportifs

T\_set\_ref\_Sport() : renvoie les référence à une table liant des Sport à des sportifs

Enfin, la dernière colonne contient une sous-requête qui renvoie une référence au sportif avec l'IDSPORTIF NULL.

-----> table sport:



```
Run SQL Command Line
SQL> select count(*) as nb_sport from sport;

  NB_SPORT
-----
         9

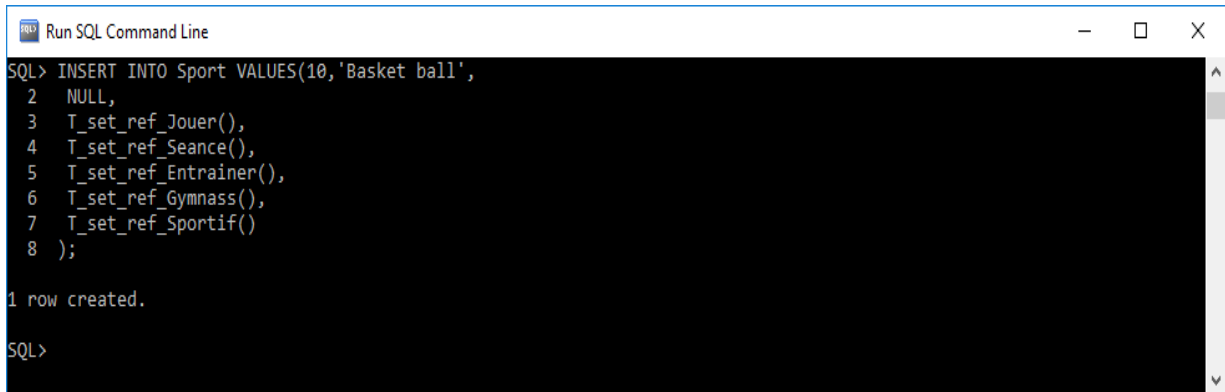
SQL>
```

**nb:** le code sql des insertions il est au niveau de fichier :

----- CodeSql3\_projet\_bda -----

explication:

----> on va prendre un exemple:

A screenshot of a 'Run SQL Command Line' window. The window has a title bar with a small icon and standard window controls (minimize, maximize, close). The main area is a black terminal with white text. The text shows an SQL INSERT statement for a table named 'Sport'. The statement is: SQL> INSERT INTO Sport VALUES(10, 'Basket ball', 2 NULL, 3 T\_set\_ref\_Jouer(), 4 T\_set\_ref\_Seance(), 5 T\_set\_ref\_Entrainer(), 6 T\_set\_ref\_Gymnass(), 7 T\_set\_ref\_Sportif() 8 );. Below the statement, it says '1 row created.' and then 'SQL>' on a new line. The window has a vertical scrollbar on the right side.

```
SQL> INSERT INTO Sport VALUES(10, 'Basket ball',
2 NULL,
3 T_set_ref_Jouer(),
4 T_set_ref_Seance(),
5 T_set_ref_Entrainer(),
6 T_set_ref_Gymnass(),
7 T_set_ref_Sportif()
8 );

1 row created.

SQL>
```

Cette requête SQL insère une nouvelle ligne dans la table "Sport" avec les valeurs suivantes :

L'identifiant de sport est 10.

Le nom de sport est 'Basket ball'.

Le champ pour la référence à la ligne de la table Arbitrer est NULL (vide).

Le champ pour la référence à la ligne de la table "Jouer" est défini en appelant le constructeur "T\_set\_ref\_Jouer()" qui renvoie la référence à une ligne dans la table Jouer.

Le champ pour la référence à la ligne de la table "Seance" est défini en appelant le constructeur "T\_set\_ref\_Seance()" qui renvoie la référence à une ligne dans la table Seance.

Le champ pour la référence à la ligne de la table "Entrainer" est défini en appelant le constructeur "T\_set\_ref\_Entrainer()" qui renvoie la référence à une ligne dans la table Entrainer.

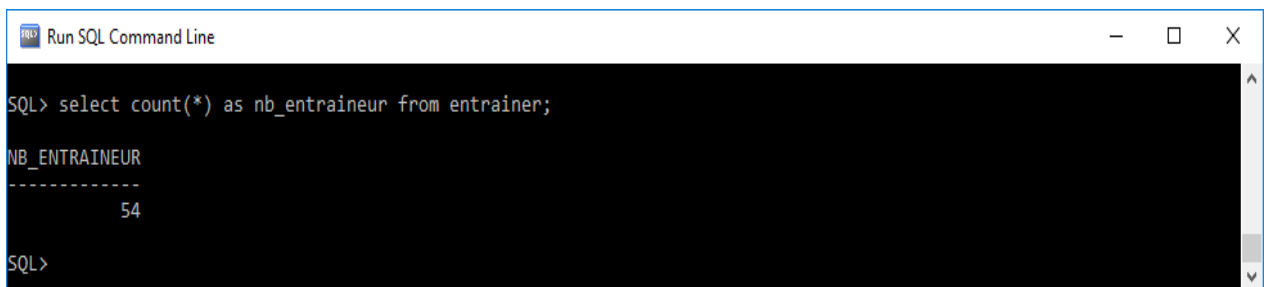
Le champ pour la référence à la ligne de la table "Gymnass" est



défini en appelant le constructeur "T\_set\_ref\_Gymnass()" qui renvoie la référence à une ligne dans la table Gymnass.

Le champ pour la référence à la ligne de la table "Sportif" est défini en appelant le constructeur "T\_set\_ref\_Sportif()" qui renvoie la référence à une ligne dans la table Sportif.

-----> table Entraîner:



```
Run SQL Command Line
SQL> select count(*) as nb_entraineur from entraineur;

NB_ENTRAINEUR
-----
              54

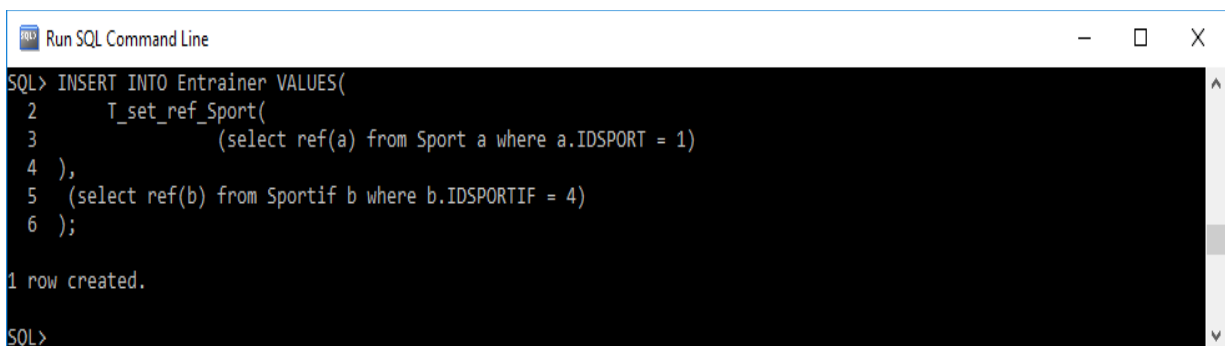
SQL>
```

**nb:** le code sql des insertions il est au niveau de fichier :

----- CodeSql3\_projet\_bda -----

explication:

----> on va prendre un exemple:



```
Run SQL Command Line
SQL> INSERT INTO Entraîner VALUES(
2     T_set_ref_Sport(
3         (select ref(a) from Sport a where a.IDSPORT = 1)
4     ),
5     (select ref(b) from Sportif b where b.IDSPORTIF = 4)
6 );

1 row created.

SQL>
```

Cette requête SQL insère une nouvelle ligne dans la table "Entrainer" avec les valeurs suivantes :

Le champ pour la référence à la ligne de la table "Sport" est défini en utilisant une sous-requête qui sélectionne la référence à la ligne de la table "Sport" qui a un IDSPORT égal à 1. Cela signifie que cette entrée d'"Entrainer" est associée à la ligne de la table "Sport" qui a un IDSPORT égal à 1.

Le champ pour la référence à la ligne de la table "Sportif" est défini en appelant le constructeur "T\_set\_ref\_Sportif()" qui prend la référence à la ligne de la table "Sportif" obtenue à partir de la sous-requête précédente en tant que paramètre. Cela signifie que cette entrée d'"Entrainer" est associée à une ligne de la table "Sportif" qui entraîne le sport associé à la ligne de la table "Sport" qui a un IDSPORIF égal à 4.

-----> table Jouer:

```
Run SQL Command Line

SQL> select count(*) as nb_joueur from jouer;

NB_JOUEUR
-----
      205

SQL>
```

**nb:** le code sql des insertions il est au niveau de fichier :

----- CodeSql3\_projet\_bda -----

explication:

----> on va prendre un exemple:

```
Run SQL Command Line

SQL> INSERT INTO Jouer VALUES(
2      T_set_ref_Sportif(
3          (select ref(a) from Sportif a where a.IDSPORTIF = 27)
4      ),
5      T_set_ref_Sport(
6          (select ref(b) from Sport b where b.IDSPORT = 2),
7      (select ref(b) from Sport b where b.IDSPORT = 3),
8      (select ref(b) from Sport b where b.IDSPORT = 6),
9      (select ref(b) from Sport b where b.IDSPORT = 7)
10     )
11 );

1 row created.

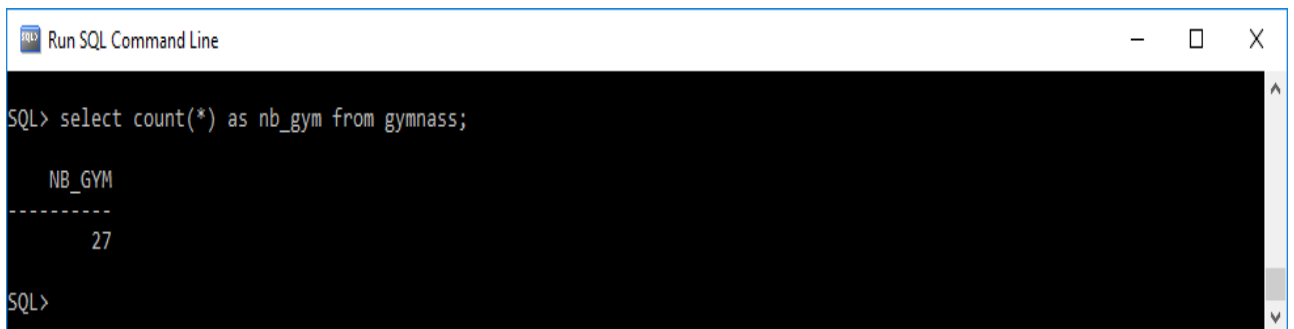
SQL>
```

Cette requête SQL insère une nouvelle ligne dans la table "Jouer" en utilisant les valeurs fournies. La table "Jouer" doit avoir deux colonnes de type REF, chacune faisant référence à des types d'objets différents : "Sportif" et "Sport".

La première valeur à insérer dans la table "Jouer" est une référence à un objet "Sportif". Cette référence est obtenue en sélectionnant l'objet "Sportif" qui a un "IDSPORTIF" de 27, et en utilisant le constructeur T\_set\_ref\_Sportif.

La deuxième valeur à insérer est une référence à un ou plusieurs objets "Sport". Cette référence est obtenue en sélectionnant les objets "Sport" qui ont les "IDSPORT" 2, 3, 6 et 7, et en utilisant le constructeur T\_set\_ref\_Sport pour créer une référence à chacun de ces objets. Les quatre références sont ensuite combinées en un seul objet de type tableau, qui est ensuite inséré dans la colonne de référence à "Sport" de la table "Jouer".

-----> table Gymnases:



```
Run SQL Command Line
SQL> select count(*) as nb_gym from gymnass;

  NB_GYM
-----
      27

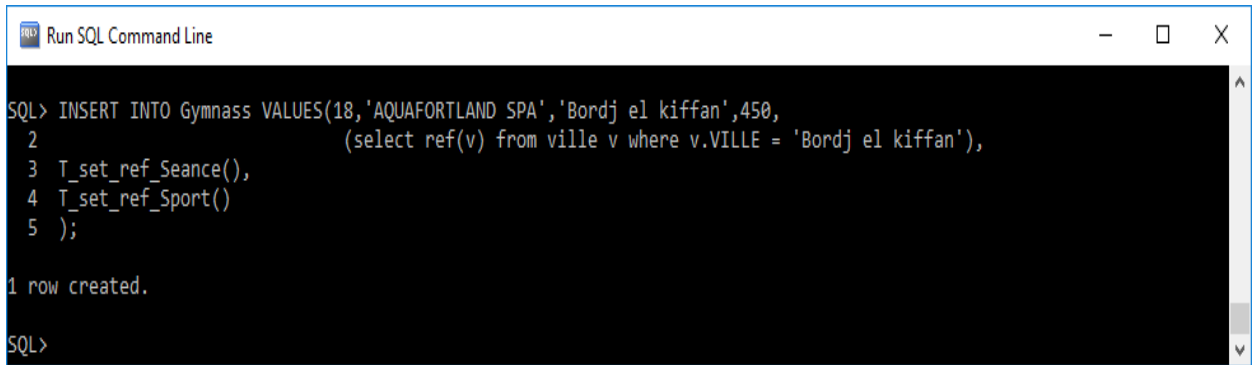
SQL>
```

**nb:** le code sql des insertions il est au niveau de fichier :

----- CodeSql3\_projet\_bda -----

explication:

----> on va prendre un exemple:

A screenshot of a 'Run SQL Command Line' window. The window has a title bar with a minus, maximize, and close button. The background is black with white text. The SQL command is: 

```
SQL> INSERT INTO Gymnass VALUES(18, 'AQUAFORTLAND SPA', 'Bordj el kiffan', 450,  
2 (select ref(v) from ville v where v.VILLE = 'Bordj el kiffan'),  
3 T_set_ref_Seance(),  
4 T_set_ref_Sport()  
5 );
```

 The result of the command is: 

```
1 row created.
```

 The prompt 'SQL>' is visible at the bottom.

Cette requête SQL insère une nouvelle ligne dans la table "Gymnass" avec les valeurs suivantes pour les colonnes :

"id\_gymnase" : 18

"nom\_gymnase" : 'AQUAFORTLAND SPA'

"adresse\_gymnase" : 'Bordj el kiffan'

"surface" : 450

Les valeurs pour les deux dernières colonnes sont obtenues en appelant des fonctions de référence dans la requête :

La colonne "ref\_ville" utilise la sous-requête (select ref(v) from ville v where v.VILLE = 'Bordj el kiffan') pour rechercher la ville de Bordj el kiffan dans la table "ville" et retourner sa référence.

La colonne "ref\_seance" utilise le constructeur T\_set\_ref\_Seance() pour générer une nouvelle référence de séance.

La colonne "ref\_sport" utilise le constructeur T\_set\_ref\_Sport() pour générer une nouvelle référence de sport.

En résumé, cette requête insère une nouvelle entrée dans la table

"Gymnass" pour représenter un gymnase nommé "AQUAFORTLAND SPA" situé à "Bordj el kiffan" avec une surface de 450 . Les références de la ville, de la séance et du sport associés à ce gymnase sont également stockées dans la table.

-----> table Ville:

```
Run SQL Command Line

SQL> select count(*) as nb_ville from ville;

  NB_VILLE
-----
        17

SQL>
```

**nb:** le code sql des insertions il est au niveau de fichier :

----- CodeSql3\_projet\_bda -----

explication:

----> on va prendre un exemple:

```
Run SQL Command Line

SQL> INSERT INTO Ville VALUES('El Mouradia',
2      T_set_ref_Gymnass(
3      (select ref(g) from gymnass g where g.IDGYMNASE= 6),
4      (select ref(g) from gymnass g where g.IDGYMNASE= 11)
5      );
6

1 row created.

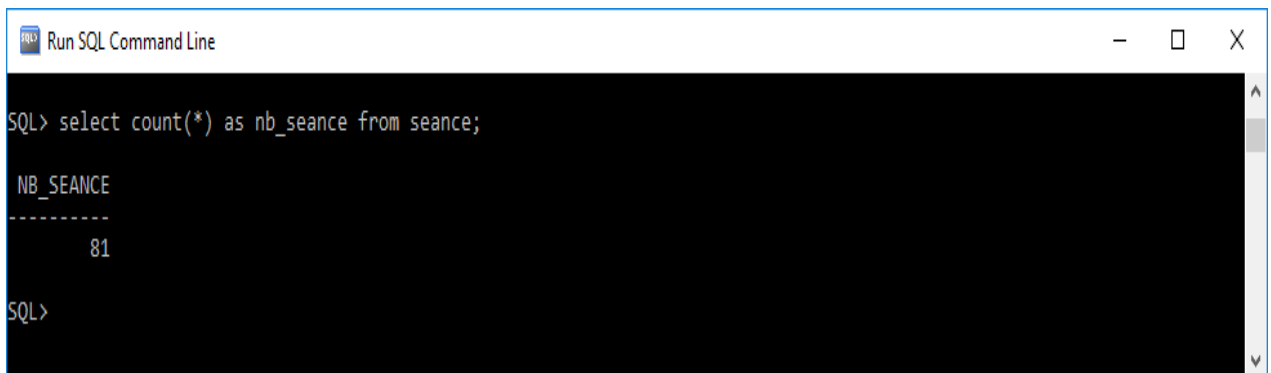
SQL>
```

Cette requête SQL insère une nouvelle ligne dans la table "Ville" avec une valeur pour la colonne "nom" de "El Mouradia" et une

valeur pour la colonne "gymnases" qui est un type de données référence à un ensemble de gymnases. L'ensemble de gymnases spécifié contient les références aux gymnases ayant les ID 6 et 11 dans la table "gymnass".

La syntaxe utilisée pour spécifier l'ensemble de gymnases est spécifique à Oracle et utilise le constructeur T\_set\_ref\_Gymnass pour créer une collection de références à des objets de type "gymnass".

-----> table Seance:



```
Run SQL Command Line
SQL> select count(*) as nb_seance from seance;

NB_SEANCE
-----
          81

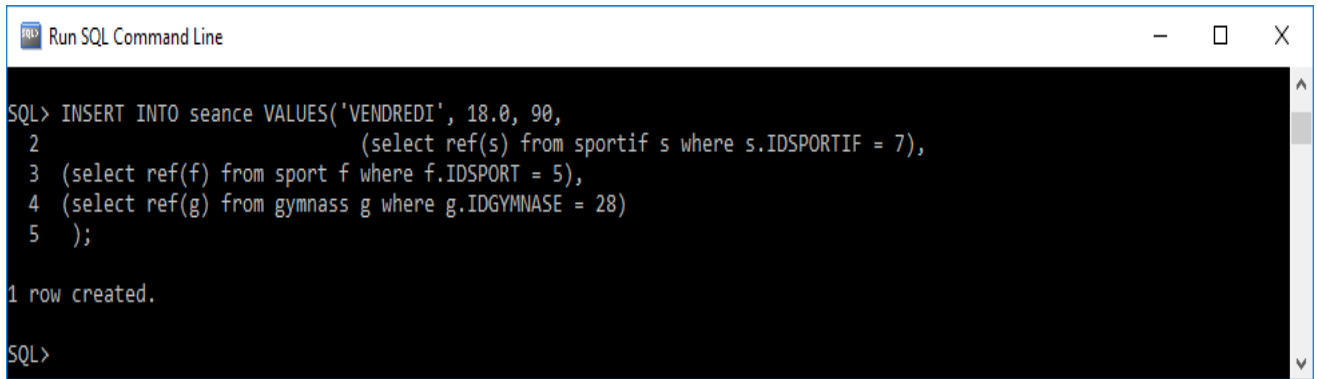
SQL>
```

**nb:** le code sql des insertions il est au niveau de fichier :

----- CodeSql3\_projet\_bda -----

explication:

----> on va prendre un exemple:



```
Run SQL Command Line
SQL> INSERT INTO seance VALUES('VENDREDI', 18.0, 90,
2      (select ref(s) from sportif s where s.IDSPORTIF = 7),
3      (select ref(f) from sport f where f.IDSPORT = 5),
4      (select ref(g) from gymnass g where g.IDGYMNASE = 28)
5  );

1 row created.

SQL>
```

Cette requête insère une nouvelle séance dans une table appelée "seance". Les valeurs insérées pour cette séance sont les suivantes :

Jour : "VENDREDI"

Heure de début : 18.0

Durée : 90 minutes

Sportif : il s'agit d'une référence vers le sportif dont l'IDSPORTIF est égal à 7 dans la table "sportif"

Sport : il s'agit d'une référence vers le sport dont l'IDSPORT est égal à 5 dans la table "sport"

Gymnase : il s'agit d'une référence vers le gymnase dont l'IDGYMNASE est égal à 28 dans la table "gymnass"

cette requête utilise des sous-requêtes pour récupérer les références vers les enregistrements correspondants dans les tables "sportif", "sport" et "gymnass".

## **Partie V : Langage d'interrogation de données**



// requete 1 : Quels sont les sportifs (identifiant, nom et prénom) qui ont un âge entre 20 et 30 ans ?

-----> la requete:

```
Run SQL Command Line
SQL> select s.IDSPORTIF, s.nom, s.prenom
2  from sportif s
3  where s.age <= 30 and s.age >= 20
4  order by s.age;
```

-----> le résultat:

```
Run SQL Command Line
SQL> select s.IDSPORTIF, s.nom, s.prenom,s.age
2  from sportif s
3  where s.age <= 30 and s.age >= 20
4  order by s.age;
```

| IDSPORTIF | NOM      | PRENOM    | AGE |
|-----------|----------|-----------|-----|
| 5         | AAKOUB   | Linda     | 22  |
| 13        | ABBACI   | Mohamed   | 22  |
| 17        | ACHAIBOU | Rachid    | 22  |
| 18        | HOSNI    | Leila     | 22  |
| 21        | BABACI   | Mourad    | 22  |
| 29        | ROUSSELI | Lamice    | 22  |
| 31        | SETIHA   | Moustapha | 22  |
| 41        | DELHOUME | Elina     | 22  |

Activer Windows  
Accédez aux paramètres pour activer Windows.

|               |               |     |
|---------------|---------------|-----|
| Yasser        | 50 KALI       | 22  |
| IDSPORTIF NOM |               |     |
| PRENOM        |               | AGE |
| Rami          | 52 DANDOUR    | 22  |
| Houcine       | 53 DEMMERA    | 22  |
| Bouzid        | 72 MARTALI    | 22  |
| IDSPORTIF NOM |               |     |
| PRENOM        |               | AGE |
| Asma          | 114 MICHALIKH | 22  |
| Nadia         | 9 ABDELMOUMEN | 23  |
| Abdelhamid    | 10 ABAD       | 23  |
| IDSPORTIF NOM |               |     |
| PRENOM        |               | AGE |
| Nassima       | 24 AMARA      | 23  |
| Lyes          | 25 AROUEL     | 23  |
|               | 26 BAALI      |     |

Activer Windows  
Accédez aux paramètres pour activer Windows.

|               |              |     |
|---------------|--------------|-----|
| Leila         | 26 BAALI     | 23  |
| IDSPORTIF NOM |              |     |
| PRENOM        |              | AGE |
| Hatem         | 27 BADI      | 23  |
| Daouad        | 32 COTERI    | 23  |
| Sami          | 33 RAMELI    | 23  |
| IDSPORTIF NOM |              |     |
| PRENOM        |              | AGE |
| Majid         | 40 FATAHI    | 23  |
| Nadir         | 42 BEHADI    | 23  |
| Dalia         | 43 MATI      | 23  |
| IDSPORTIF NOM |              |     |
| PRENOM        |              | AGE |
| Zohra         | 48 HEDDI     | 23  |
| Mohammed      | 54 ELKABBADJ | 23  |

Activer Windows  
Accédez aux paramètres pour activer Windows.

|               |                |     |
|---------------|----------------|-----|
| Omer          | 55 FEROLI      | 23  |
| IDSPORTIF NOM |                |     |
| -----         |                |     |
| PRENOM        |                | AGE |
| -----         |                |     |
| Adam          | 58 GUITENI     | 23  |
| Samia         | 59 KACI        | 23  |
| Amine         | 11 ABAYAHIA    | 24  |
| IDSPORTIF NOM |                |     |
| -----         |                |     |
| PRENOM        |                | AGE |
| -----         |                |     |
| Riad          | 12 ABBACI      | 24  |
| Lamia         | 14 ABDELOUAHAB | 24  |
| Lamine        | 16 BENOUDAHA   | 24  |
| IDSPORTIF NOM |                |     |
| -----         |                |     |
| PRENOM        |                | AGE |
| -----         |                |     |
| Youssef       | 23 BEHADI      | 24  |
| Nidal         | 30 CHIKHI      | 24  |
|               | 34 LEHIRACHE   |     |

Activer Windows  
Accédez aux paramètres pour activer Windows.

|               |                |     |
|---------------|----------------|-----|
| Oussama       | 34 LEHIRACHE   | 24  |
| IDSPORTIF NOM |                |     |
| -----         |                |     |
| PRENOM        |                | AGE |
| -----         |                |     |
| Yacine        | 35 TERIKI      | 24  |
| Malika        | 38 LAZARI      | 24  |
| Otheman       | 39 MORELI      | 24  |
| IDSPORTIF NOM |                |     |
| -----         |                |     |
| PRENOM        |                | AGE |
| -----         |                |     |
| Islam         | 46 DOUDOU      | 24  |
| Sandra        | 49 JADI        | 24  |
| Fouad         | 51 LAJEL       | 24  |
| IDSPORTIF NOM |                |     |
| -----         |                |     |
| PRENOM        |                | AGE |
| -----         |                |     |
| Amel          | 3 BOUZIDI      | 25  |
| Zouhir        | 7 HADJ         | 25  |
|               | 15 ABDEMEZIANE |     |

Activer Windows  
Accédez aux paramètres pour activer Windows.

|               |     |             |     |
|---------------|-----|-------------|-----|
| Majid         | 15  | ABDEMEZIANE | 25  |
| IDSPORTIF NOM |     |             |     |
| -----         |     |             |     |
| PRENOM        |     |             | AGE |
| -----         |     |             |     |
| Adel          | 19  | ABERKANE    | 25  |
| Racim         | 20  | AZOUG       | 25  |
| Ayoub         | 22  | BAKIR       | 25  |
| IDSPORTIF NOM |     |             |     |
| -----         |     |             |     |
| PRENOM        |     |             | AGE |
| -----         |     |             |     |
| Malika        | 37  | LAZARI      | 25  |
| Céline        | 47  | Grine       | 25  |
| Mouad         | 151 | HENDI       | 25  |
| IDSPORTIF NOM |     |             |     |
| -----         |     |             |     |
| PRENOM        |     |             | AGE |
| -----         |     |             |     |
| Zohra         | 56  | GUERRAOUI   | 25  |
| Aziz          | 57  | BOUACHA     | 25  |

Activer Windows  
Accédez aux paramètres pour activer Windows.

|               |     |           |     |
|---------------|-----|-----------|-----|
| Hadjer        | 69  | MARADI    | 25  |
| IDSPORTIF NOM |     |           |     |
| -----         |     |           |     |
| PRENOM        |     |           | AGE |
| -----         |     |           |     |
| Jamel         | 92  | HADJI     | 25  |
| Maria         | 145 | LAAMARA   | 25  |
| Ramezi        | 146 | DABOUB    | 25  |
| IDSPORTIF NOM |     |           |     |
| -----         |     |           |     |
| PRENOM        |     |           | AGE |
| -----         |     |           |     |
| Nadia         | 147 | HASSINI   | 25  |
| Maria         | 148 | KALOUNE   | 25  |
| Besma         | 149 | BELHAOUA  | 25  |
| IDSPORTIF NOM |     |           |     |
| -----         |     |           |     |
| PRENOM        |     |           | AGE |
| -----         |     |           |     |
| Fouad         | 150 | BELAID    | 25  |
| Dalila        | 67  | FATTIMI   | 26  |
|               | 136 | LABOULAIS |     |

Activer Windows  
Accédez aux paramètres pour activer Windows.

|               |           |     |
|---------------|-----------|-----|
| 136           | LABOULAIS |     |
| Fadia         |           | 26  |
| IDSPORTIF NOM |           |     |
| -----         |           |     |
| PRENOM        |           | AGE |
| -----         |           |     |
| 137           | DOUDOU    |     |
| Faiza         |           | 26  |
| 138           | MAALEM    |     |
| Lamia         |           | 26  |
| 139           | BESNARD   |     |
| Salma         |           | 26  |
| IDSPORTIF NOM |           |     |
| -----         |           |     |
| PRENOM        |           | AGE |
| -----         |           |     |
| 140           | BELHAMID  |     |
| Hadjer        |           | 26  |
| 141           | BOUAAZA   |     |
| Asma          |           | 26  |
| 142           | CORCHI    |     |
| Melissa       |           | 26  |
| IDSPORTIF NOM |           |     |
| -----         |           |     |
| PRENOM        |           | AGE |
| -----         |           |     |
| 143           | BELAID    |     |
| Jaouida       |           | 26  |
| 144           | GASMI     |     |
| Souad         |           | 26  |

Activer Windows  
Accédez aux paramètres pour activer Windows.

|               |            |     |
|---------------|------------|-----|
| 61            | LAZARRI    |     |
| Jamel         |            | 27  |
| IDSPORTIF NOM |            |     |
| -----         |            |     |
| PRENOM        |            | AGE |
| -----         |            |     |
| 2             | BOUROUBI   |     |
| Anis          |            | 28  |
| 36            | DJELOUDANE |     |
| Zinedine      |            | 28  |
| 44            | ADIBOU     |     |
| Ibrahim       |            | 28  |
| IDSPORTIF NOM |            |     |
| -----         |            |     |
| PRENOM        |            | AGE |
| -----         |            |     |
| 45            | CHALI      |     |
| Karim         |            | 28  |
| 85            | SALLADj    |     |
| Miloud        |            | 28  |
| 1             | BOUTAHAR   |     |
| Abderahim     |            | 30  |
| IDSPORTIF NOM |            |     |
| -----         |            |     |
| PRENOM        |            | AGE |
| -----         |            |     |
| 6             | ABBAS      |     |
| ABBAS         |            | 30  |
| 8             | HAMADI     |     |
| Hani          |            | 30  |
| 63            | AMANI      |     |

Activer Windows  
Accédez aux paramètres pour activer Windows.

|               |    |           |    |
|---------------|----|-----------|----|
| Fadi          | 63 | AMANI     | 30 |
| IDSPORTIF NOM |    |           |    |
| PRENOM        |    | AGE       |    |
| Faiza         | 64 | LANORI    | 30 |
| Mourad        | 65 | CHAADI    | 30 |
| Mohamed       | 66 | DANDANE   | 30 |
| IDSPORTIF NOM |    |           |    |
| PRENOM        |    | AGE       |    |
| Jazia         | 68 | REGHI     | 30 |
| Nadji         | 70 | BELMADI   | 30 |
| Racim         | 71 | DELAROCHI | 30 |
| IDSPORTIF NOM |    |           |    |
| PRENOM        |    | AGE       |    |
| Douad         | 73 | DALLIMI   | 30 |
| Adel          | 74 | OUBACHA   | 30 |
|               | 76 | HALGATTI  |    |

|               |    |           |    |
|---------------|----|-----------|----|
| Camelia       | 76 | HALGATTI  | 30 |
| IDSPORTIF NOM |    |           |    |
| PRENOM        |    | AGE       |    |
| Farid         | 77 | HALGATTI  | 30 |
| Jamel         | 78 | CHAOUAH   | 30 |
| Jamel         | 79 | CHAOUAH   | 30 |
| IDSPORTIF NOM |    |           |    |
| PRENOM        |    | AGE       |    |
| Ramezi        | 80 | HOCHET    | 30 |
| Jaouida       | 81 | DROULLONI | 30 |
| Ahmed         | 83 | LOUATI    | 30 |
| IDSPORTIF NOM |    |           |    |
| PRENOM        |    | AGE       |    |
| Anes          | 84 | HAMARI    | 30 |
| Boualem       | 86 | GALLOTI   | 30 |
|               | 87 | KASBADJI  |    |

|               |               |     |
|---------------|---------------|-----|
| Fateh         | 87 KASBADJI   | 30  |
| IDSPORTIF NOM |               |     |
| -----         |               |     |
| PRENOM        |               | AGE |
| -----         |               |     |
| Rachid        | 88 JENOURI    | 30  |
| Jamel         | 89 RIHABI     | 30  |
| Nadir         | 90 DERARNI    | 30  |
| IDSPORTIF NOM |               |     |
| -----         |               |     |
| PRENOM        |               | AGE |
| -----         |               |     |
| Zinedine      | 91 BATERAOUI  | 30  |
| Nabil         | 93 CAUCHARDI  | 30  |
| Mazine        | 95 ESTANBOULI | 30  |
| IDSPORTIF NOM |               |     |
| -----         |               |     |
| PRENOM        |               | AGE |
| -----         |               |     |
| Lamine        | 96 JANID      | 30  |
| Bassim        | 97 BONHOMMANE | 30  |
|               | 98 RIADI      |     |

|               |              |     |
|---------------|--------------|-----|
| Walid         | 98 RIADI     | 30  |
| IDSPORTIF NOM |              |     |
| -----         |              |     |
| PRENOM        |              | AGE |
| -----         |              |     |
| Djamil        | 100 LESOIFI  | 30  |
| Esslam        | 101 SWAMI    | 30  |
| Adel          | 102 DAOUDI   | 30  |
| IDSPORTIF NOM |              |     |
| -----         |              |     |
| PRENOM        |              | AGE |
| -----         |              |     |
| Nassim        | 103 LAAMOURI | 30  |
| Dihia         | 104 SEHIER   | 30  |
| Fouad         | 105 STITOUAH | 30  |
| IDSPORTIF NOM |              |     |
| -----         |              |     |
| PRENOM        |              | AGE |
| -----         |              |     |
| Hani          | 106 BAADI    | 30  |
| Nazim         | 107 BOURAS   | 30  |

```
Run SQL Command Line
118 IGHOLI
Lyes
30
IDSPORTIF NOM
-----
PRENOM AGE
-----
119 GUEMEZ
Jaouad
30
120 LECOM
Aissa
30
121 HOUAT
Aziz
30
IDSPORTIF NOM
-----
PRENOM AGE
-----
122 BEQUETA
Aicha
30
123 RATENI
Walid
30
124 TOUAT
Yasmine
30
IDSPORTIF NOM
-----
PRENOM AGE
-----
125 JALONI
Aimad
30
126 DEBOUBA
yasser
30
127 GASTAB
```

Activer Windows  
Accédez aux paramètres pour activer Windows.

```
Run SQL Command Line
127 GASTAB
Chouaib
30
IDSPORTIF NOM
-----
PRENOM AGE
-----
128 GASTAB
Chouaib
30
129 DABONI
Rachid
30
130 LACHOUBI
Kamel
30
IDSPORTIF NOM
-----
PRENOM AGE
-----
131 GALLOI
Nadira
30
132 DORONI
Yanis
30
133 LENOUCI
Youssef
30
IDSPORTIF NOM
-----
PRENOM AGE
-----
134 LERICHE
Hadi
30
135 MANSOUR
Lamine
30
```

Activer Windows  
Accédez aux paramètres pour activer Windows.



```
Run SQL Command Line
135 MANSOUR
Lamine 30

143 rows selected.

SQL>
```

// requete 2: Afficher la superficie moyenne des gymnases, pour chaque ville.

```
Run SQL Command Line

SQL> SELECT distinct v.ville, v.CALCUL_SUP_MOY_GYMNASS() FROM ville v;

VILLE                                V.CALCUL_SUP_MOY_GYMNASS()
-----
Belouizdad                            400
El Biar                               620
Bordj el Kiffan                        450
El Mouradia                           350
Chéraga                               510
Bordj El Bahri                        450
Les sources                           450
Sidi Mhamed                           500
Hydra                                  310
Hussein Dey                           500
Kouba                                  400

VILLE                                V.CALCUL_SUP_MOY_GYMNASS()
-----
El Achour                             600
Béni Messous                          520
Alger centre                          280
Dely Brahim                           620
Bir Mourad Raïs                       480
Birkhadem                             450
Baba hassen                           450
Alger                                  400

19 rows selected.

SQL>
```

// requete 3: Quels sont les sportifs qui sont des conseillers ?

-----> la requete:

```
Run SQL Command Line

SQL> select s.nom,s.prenom
2  from sportif s
3  where s.conseille is not null
4  order by s.IDSPORTIF;
```

-----> le résultat:

```
Run SQL Command Line

SQL> select s.nom,s.prenom
2  from sportif s
3  where s.conseille is not null
4  order by s.IDSPORTIF;

NOM
-----
PRENOM
-----
BOUROUBI
Anis

BOUZIDI
Amel

LACHEMI
Bouzid

NOM
-----
PRENOM
-----
AAKOUB
Linda

ABBAS
ABBAS

HADJ
Zouhir

NOM
-----
PRENOM
-----
HAMADI
Hani

ABDELMOUNEN
Nadia

Activer Windows
Accédez aux paramètres pour activer Windows.
```

```
Run SQL Command Line
ABAD
Abdelhamid

NOM
-----
PRENOM
-----
ABAYAHIA
Amine

ABBACI
Riad

ABBACI
Mohamed

NOM
-----
PRENOM
-----
ABDELOUAHAB
Lamia

ABDEMEZIANE
Majid

BENOUADAH
Lamine

NOM
-----
PRENOM
-----
ACHAIBOU
Rachid

HOSNI
Leila

ABERKANE
```

Activer Windows  
Accédez aux paramètres pour activer Windows.

Run SQL Command Line

ABERKANE  
Adel

NOM  
-----  
PRENOM  
-----  
AZOUG  
Racim

BABACI  
Mourad

BAKIR  
Ayoub

NOM  
-----  
PRENOM  
-----  
BEHADI  
Youssef

AMARA  
Nassima

AROUEL  
Lyes

NOM  
-----  
PRENOM  
-----  
BAALI  
Leila

BADI  
Hatem

Activer Windows  
Accédez aux paramètres pour activer Windows.

```
Run SQL Command Line
BADI
Hatem

RABAHI
Rabah

NOM
-----
PRENOM
-----
ROUSSELI
Lamice

CHIKHI
Nidal

SETIHA
Moustapha

NOM
-----
PRENOM
-----
COTERI
Daoud

RAMELI
Sami

LEHIRACHE
Oussama

NOM
-----
PRENOM
-----
TERIKI
Yacine

DJELOUDANE
```

Activer Windows  
Accédez aux paramètres pour activer Windows.

```
Run SQL Command Line
DJELOUDANE
Zinedine

LAZARI
Malika

NOM
-----
PRENOM
-----
MORELI
Otheman

FATAHI
Majid

DELHOUME
Elina

NOM
-----
PRENOM
-----
BEHADI
Nadir

MATI
Dalia

ADIBOU
Ibrahim

NOM
-----
PRENOM
-----
DOUDOU
Islam
```

Activer Windows  
Accédez aux paramètres pour activer Windows.

```
Run SQL Command Line

Grine
Céline

HEDDI
Zohra

NOM
-----
PRENOM
-----
JADI
Sandra

KALI
Yasser

LAJEL
Fouad

NOM
-----
PRENOM
-----
DANDOUR
Rami

DEMMEHA
Houcine

ELKABBADJ
Mohammed

NOM
-----
PRENOM
-----
FEROLI
Omer
```

Activer Windows  
Accédez aux paramètres pour activer Windows.

Run SQL Command Line

GUERRAOUI

Zohra

BOUACHA

Aziz

NOM

-----

PRENOM

-----

GUITENI

Adam

TIZEGHAT

Badis

LAZARRI

Jamel

NOM

-----

PRENOM

-----

BAZOUZI

Daoud

AMANI

Fadi

LANORI

Faiza

NOM

-----

PRENOM

-----

DANDANE

Mohamed

Activer Windows

Accédez aux paramètres pour activer Windows.



Run SQL Command Line

```
FATTIMI
Dalila

REGHI
Dazia

NOM
-----
PRENOM
-----
MARADI
Hadjer

BELMADI
Nadji

DELAROCHI
Racim

NOM
-----
PRENOM
-----
MARTALI
Bouzid

DALLIMI
Douad

OUBACHA
Adel

NOM
-----
PRENOM
-----
SAADI
Nihal
```

Activer Windows  
Accédez aux paramètres pour activer Windows.

```
Run SQL Command Line
HALGATTI
Camelia

HALGATTI
Farid

NOM
-----
PRENOM
-----
CHAOUAH
Jamel

HOCHET
Ramezi

DROULLONI
Jaouida

NOM
-----
PRENOM
-----
HOULEMI
Lyes

LOUATI
Ahmed

HAMARI
Anes

NOM
-----
PRENOM
-----
SALLADj
Miloud

GALLOTI
```

Activer Windows  
Accédez aux paramètres pour activer Windows.

```
Run SQL Command Line
GALLOTI
Boualem

KASBADJI
Fateh

NOM
-----
PRENOM
-----
JENOURI
Rachid

DERARWI
Nadir

HADJI
Jamel

NOM
-----
PRENOM
-----
CAUCHARDI
Nabil

LEROUDI
Moussa

ESTANBOULI
Mazine

NOM
-----
PRENOM
-----
JANID
Lamine

RIADI
```

Activer Windows  
Accédez aux paramètres pour activer Windows.

```
Run SQL Command Line
RIADI
Walid

LESOIFI
Djamil

NOM
-----
PRENOM
-----
SWAMI
Esslam

DAOUDI
Adel

LAAMOURI
Nassim

NOM
-----
PRENOM
-----
SEHIER
Dihia

STITOUAH
Fouad

BAADI
Hani

NOM
-----
PRENOM
-----
BOURAS
Nazim
```

Activer Windows  
Accédez aux paramètres pour activer Windows.

Run SQL Command Line

AIT AMARA  
Salim  
  
AIT AMARA  
Salim  
  
NOM  
-----  
PRENOM  
-----  
ROULLADI  
Aïssa  
  
ROUTINE  
Mohamed  
  
LOUATI  
Islam  
  
NOM  
-----  
PRENOM  
-----  
AID  
Naim  
  
MICHALIKH  
Asma  
  
LEMOUSSI  
Amine  
  
NOM  
-----  
PRENOM  
-----  
BELIFA  
Samia

Activer Windows  
Accédez aux paramètres pour activer Windows.

```
Run SQL Command Line

FERRIRA
Manel

IGHOLI
Lyes

NOM
-----
PRENOM
-----
GUEMEZ
Jaouad

LECOM
Aïssa

HOUAT
Aziz

NOM
-----
PRENOM
-----
BEQUETA
Aïcha

RATENI
Walid

TOUAT
Yasmine

NOM
-----
PRENOM
-----
JALONI
Aïmad
```

Activer Windows  
Accédez aux paramètres pour activer Windows.

```
Run SQL Command Line
DEBOUBA
yasser

GASTAB
Chouaib

NOM
-----
PRENOM
-----
GASTAB
Chouaib

DABONI
Rachid

LACHOUBI
Kamel

NOM
-----
PRENOM
-----
GALLOI
Nadira

DORONI
Yanis

LENOUCHI
Youcef

NOM
-----
PRENOM
-----
LERICHE
Hadi

Activer Windows
Accédez aux paramètres pour activer Windows.
```

```
Run SQL Command Line
MANSOUR
Lamine

LABOULAIS
Fadia

NOM
-----
PRENOM
-----
DOUDOU
Faiza

MAALEM
Lamia

BESNARD
Salma

NOM
-----
PRENOM
-----
BELHAMID
Hadjer

BOUAAZA
Asma

CORCHI
Melissa

NOM
-----
PRENOM
-----
BELAID
Jaouida

GASMI
```

Activer Windows  
Accédez aux paramètres pour activer Windows.

```
Run SQL Command Line
NOM
-----
PRENOM
-----
BELAID
Jaouida

GASMI
Souad

LAAMARA
Maria

NOM
-----
PRENOM
-----
DABOUB
Ramezi

HASSINI
Nadia

KALOUNE
Maria

NOM
-----
PRENOM
-----
BELHAOUA
Besma

BELAID
Fouad

HENDI
Mouad

141 rows selected.
```

// requete 4 Quels entraîneurs n'entraînent que du hand ball ou du basket ball ?

-----> la requete

```
Run SQL Command Line
SQL> select distinct deref(e.entraineur_sportif).nom , deref(e.entraineur_sportif).prenom
2  from entraineur e , table(e.entraineur_sport) t1
3  where deref(t1.column_value).LIBELLE = 'Hand ball' or
4         deref(t1.column_value).LIBELLE = 'Basket ball';
```



-----> le résultat

```
Run SQL Command Line
SQL> select distinct deref(e.entraineur_sportif).nom , deref(e.entraineur_sportif).prenom
2  from entraineur e , table(e.entraineur_sport) t1
3  where deref(t1.column_value).LIBELLE = 'Hand ball' or
4         deref(t1.column_value).LIBELLE = 'Basket ball';

DEREF(E.ENTRAINEUR_SPORTIF).NOM
-----
DEREF(E.ENTRAINEUR_SPORTIF).PRENOM
-----
BOUTAHAR
Abderahim

LACHEMI
Bouzid

HADJ
Zouhir

DEREF(E.ENTRAINEUR_SPORTIF).NOM
-----
DEREF(E.ENTRAINEUR_SPORTIF).PRENOM
-----
BOUROUBI
Anis

BELHAOUA
Besma

BOUZIDI
Amel

HENDI
Mouad

7 rows selected.
```

// requete 5: Quels sont les sportifs les plus jeunes?

----->on va déviser la requete en 4 sous requetes :

sous-requete1:

```
Run SQL Command Line
SQL> SELECT age, COUNT(age) AS age_count
2 FROM sportif
3 GROUP BY age;

  AGE  AGE_COUNT
-----
    30         66
    25         19
    22         13
    28          5
    26         10
    32          4
    24         13
    23         16
    40          2
    27          1
    36          1

  AGE  AGE_COUNT
-----
    39          1

12 rows selected.

SQL>
```

cet requete permet d'extraire pour chaque age ses nombres d'occurence.

### sous-requete2:

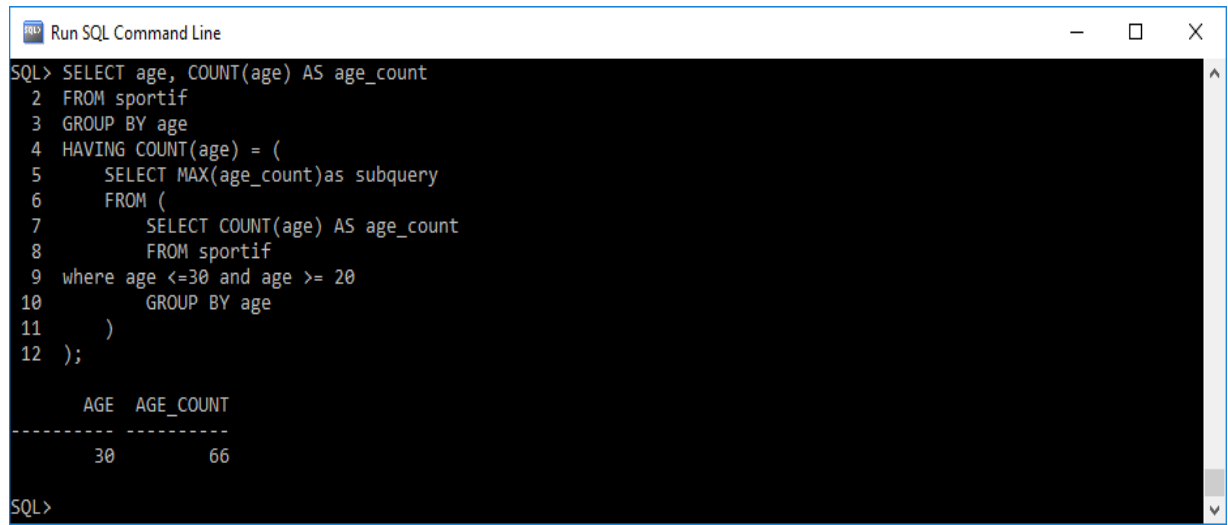
```
Run SQL Command Line
SQL> SELECT age, COUNT(age) AS age_count
2 FROM sportif
3 where age <=30 and age >= 20
4 GROUP BY age;

  AGE  AGE_COUNT
-----
    30         66
    25         19
    22         13
    28          5
    26         10
    24         13
    23         16
    27          1

8 rows selected.
```

----> cet requete permet d'extraire pour chaque age entre 20 et 30 ses nombres d'occurence.

### sous-requete3:



```
SQL> SELECT age, COUNT(age) AS age_count
2 FROM sportif
3 GROUP BY age
4 HAVING COUNT(age) = (
5     SELECT MAX(age_count) as subquery
6     FROM (
7         SELECT COUNT(age) AS age_count
8         FROM sportif
9     where age <=30 and age >= 20
10         GROUP BY age
11     )
12 );
```

| AGE | AGE_COUNT |
|-----|-----------|
| 30  | 66        |

SQL>

---> cet requete permet d'extraire à partir de l'ensemble précédent l'age le plus fréquent

### sous-requete4:

```
Run SQL Command Line
SQL> SELECT nom, prenom
  2 FROM sportif
  3 WHERE age IN (
  4     SELECT age
  5     FROM sportif
  6     GROUP BY age
  7     HAVING COUNT(*) = (
  8         SELECT MAX(age_count) as subquery
  9         FROM (
 10             SELECT COUNT(*) AS age_count
 11             FROM sportif
 12             GROUP BY age
 13         )
 14     )
 15 );
```

-----> le résultat

```
Run SQL Command Line
NOM
-----
PRENOM
-----
MANSOUR
Lamine
LERICHE
Hadi
LENOUCHI
Youcef
NOM
-----
PRENOM
-----
DORONI
Yanis
GALLOI
Nadira
LACHOUBI
Kamel
NOM
-----
PRENOM
-----
DABONI
Rachid
GASTAB
Chouaib
GASTAB
Chouaib
NOM
```

Activer Windows  
Accédez aux paramètres pour activer Windows.

Run SQL Command Line

NOM

-----

PRENOM

-----

DEBOUBA

yasser

JALONI

Aimad

TOUAT

Yasmine

NOM

-----

PRENOM

-----

RATENI

Walid

BEQUETA

Aïcha

HOUAT

Aziz

NOM

-----

PRENOM

-----

LECOM

Aïssa

GUEMEZ

Jaouad

IGHOLI

Lyes

Activer Windows

Accédez aux paramètres pour activer Windows.

```
Run SQL Command Line

NOM
-----
PRENOM
-----
FERRIRA
Manel

BELIFA
Samia

LEMOUSSI
Amine

NOM
-----
PRENOM
-----
AID
Naim

LOUATI
Islam

BOUTINE
Mohamed

NOM
-----
PRENOM
-----
ROULLADI
Aïssa

AIT AMARA
Salim

AIT AMARA
Salim
```

Activer Windows  
Accédez aux paramètres pour activer Windows.

```
Run SQL Command Line

NOM
-----
PRENOM
-----
BOURAS
Nazim

BAADI
Hani

STITOUAH
Fouad

NOM
-----
PRENOM
-----
SEHIER
Dihia

LAAMOURI
Nassim

DAOUDI
Adel

NOM
-----
PRENOM
-----
SWAMI
Esslam

LESOIFI
Djamil

RIADI
Walid
```

Activer Windows  
Accédez aux paramètres pour activer Windows.

```
Run SQL Command Line

NOM
-----
PRENOM
-----
GALLOTI
Boualem

HAMARI
Anes

LOUATI
Ahmed

NOM
-----
PRENOM
-----
DROULLONI
Jaouida

HOCHET
Ramezi

CHAQUAH
Jamel

NOM
-----
PRENOM
-----
CHAQUAH
Jamel

HALGATTI
Farid

HALGATTI
Camelia
```

Activer Windows  
Accédez aux paramètres pour activer Windows.

```
Run SQL Command Line

NOM
-----
PRENOM
-----
OUBACHA
Adel

DALLIMI
Douad

DELAROCHI
Racim

NOM
-----
PRENOM
-----
BELMADI
Nadji

REGHI
Jazia

DANDANE
Mohamed

NOM
-----
PRENOM
-----
CHAADI
Mourad

LANORI
Faiza

AMANI
Fadi

Activer Windows
Accédez aux paramètres pour activer Windows.
```

```
Sélection Run SQL Command Line

NOM
-----
PRENOM
-----
HAMADI
Hani

ABBAS
ABBAS

BOUTAHAR
Abderahim

66 rows selected.
```





