

Compte-rendu du TP n°6

I) Les références

1) Créez un type « Intervenant_type » puis un type « mission_type ».

SQL]

```
CREATE OR REPLACE TYPE Intervenant_type AS OBJECT (  
    mat integer,  
    nom varchar(64),  
    prenom varchar(64),  
    qualification varchar(64),  
    tarifApplique number  
);  
/
```

```
CREATE OR REPLACE TYPE mission_type AS OBJECT (  
    code varchar(64),  
    intitule varchar(64),  
    nbJours integer,  
    intervenant ref intervenant_type  
);  
/
```

2) Créez une table des intervenants et une table des missions en mode ligne. Vous devez avoir une référence sur les intervenants.

SQL]

1) Création de la table des intervenants

```
CREATE TABLE intervenant of intervenant_type;
```

2) Création de la table des missions

```
CREATE TABLE mission of mission_type;
```

3) Renseignez la table des intervenants avec les données du TP 5.

SQL]

```
INSERT INTO intervenant VALUES (2516, 'Dupont', 'Pierre', 'Développeur', 550);  
INSERT INTO intervenant VALUES (7655, 'Henri', 'Jacques', 'Consultant', 990.90);  
INSERT INTO intervenant VALUES (7687, 'Triolet', 'Elsa', 'Consultant', 1029.00);
```

4) Renseignez la table des missions en créant une référence sur les intervenants concernés.

SQL]

```
INSERT INTO mission VALUES (mission_type('Varalpain033', 'Etude technique du
passage de PEL en CEL', 54,
(SELECT REF(i) from intervenant i where i.mat = 7655)));
INSERT INTO mission VALUES (mission_type('Armoni002', 'Prise de contact avec le
Directeur', 2,
(SELECT REF(i) from intervenant i where i.mat = 1000)));
```

R]

CODE	INTITULE	NBJOURS	INTERVENANT
Varalpain033	Etude technique du passage de PEL en CEL	54	0000220208421E29BDFC82EEDFE05 0820A0EA35BF0421E29BDFC7FEED FE050820A0EA35BF0
Armoni002	Prise de contact avec le Directeur	2	

5) Ajouter l'intervenant 1000 qui se nomme Philippe Michelin, qui est Directeur avec un tarif associé de 3000 euros la journée.

SQL]

```
INSERT INTO intervenant VALUES (1000, 'Michelin', 'Philippe', 'Directeur',
3000);
```

R]

MAT	NOM	PRENOM	QUALIFICATION	TARIFAPPLIQUE
2516	Dupont	Pierre	Développeur	550
7655	Henri	Jacques	Consultant	990,9
7687	Triolet	Elsa	Consultant	1029
1000	Michelin	Philippe	Directeur	3000

6) Affichez la référence de l'objet « Intervenant 1000 ». Expliquez le message obtenu.

SQL]

```
SELECT REF(i) from intervenant i where i.mat = 1000;
```

R] 0000280209421E29BDFC86EEDFE050820A0EA35BF0421E29BDFC7FEEDFE050820A0EA35BF0010049
4F0003

Dû à la création de la table intervenant, aucun message d'erreur n'apparaît, et la référence correcte est renvoyée.

10) Mettez à jour la table missions afin d'insérer la mission en ayant une référence sur l'intervenant 1000.

SQL]

```
UPDATE mission SET intervenant = (SELECT REF(i) from intervenant i where i.mat = 1000)
WHERE code = 'Armoni002';
```

11) Affichez la description puis le contenu de la table « mission ».

SQL] DESC mission;

R]

Name	Null?	Type
CODE		VARCHAR2(64)
INTITULE		VARCHAR2(64)
NBJOURS		NUMBER(38)
INTERVENANT		REF OF INTERVENANT_TYPE

SQL] SELECT * FROM mission;

R]

CODE	INTITULE	NBJOURS	INTERVENANT
Varalpain033	Etude technique du passage de PEL en CEL	54	0000220208421E29BDFC82EEDFE05 0820A0EA35BF0421E29BDFC7FEED FE050820A0EA35BF0
Armoni002	Prise de contact avec le Directeur	2	0000220208421E29BDFC86EEDFE05 0820A0EA35BF0421E29BDFC7FEED FE050820A0EA35BF0

12) Affichez le code et le coût de la mission (coût = nb de jours * tarif).

SQL] SELECT code, (nbjours * deref(intervenant).tarifApplique) cout FROM mission;

R]

CODE	COUT
Varalpain033	53508,6
Armoni002	6000

13) Augmentez le nombre de jours de la mission.

SQL]

1) Procédure de prolongement d'une mission

```
CREATE OR REPLACE PROCEDURE addDays(id mission.code%TYPE, n mission.nbjours
%TYPE) IS
```

```
  BEGIN
```

```
    UPDATE mission
```

```
      SET nbjours = nbjours + n
```

```
      WHERE code = id;
```

```
  END;
```

```
 /
```

2) Essai

```
execute addDays('Armoni002', 3);
```

R]

CODE	INTITULE	NBJOURS	INTERVENANT
Varalpain033	Etude technique du passage de PEL en CEL	54	0000220208421E29BDFC82EEDFE05 0820A0EA35BF0421E29BDFC7FEED FE050820A0EA35BF0
Armoni002	Prise de contact avec le Directeur	5	0000220208421E29BDFC86EEDFE05 0820A0EA35BF0421E29BDFC7FEED FE050820A0EA35BF0

14) Réduisez le tarif de Michelin à 2800 euros et affichez de nouveau le code et le coût de la mission.

SQL]

1) Modification du tarif de Philippe Michelin

```
UPDATE intervenant set tarifApplique = 2800 where nom = 'Michelin' and prenom =  
'Philippe';
```

2) Affichage du code et du coût de la mission

```
SELECT code, (nbjours * deref(intervenant).tarifApplique) cout FROM mission;
```

R]

CODE	COUT
Varalpain033	53508,6
Armoni002	14000

15) Supprimez l'intervenant Michelin puis visualisez la mission et son coût

SQL]

1) Suppression de l'intervenant

```
DELETE FROM intervenant where nom = 'Michelin' and prenom = 'Philippe';
```

2) Affichage du code et du coût de la mission

```
SELECT code, (nbjours * deref(intervenant).tarifApplique) cout FROM mission;
```

R]

CODE	COUT
Varalpain033	53508,6
Armoni002	

II) Tables imbriquées

1) Créez un type « missions type » qui sera une table de missions (type « mission type »).

SQL]

```
CREATE OR REPLACE TYPE missions_type AS TABLE OF mission_type;  
/
```

R] Type created.

2) Créez la table “projet” qui aura pour clé primaire le nom du projet et un attribut “lesMissions” qui sera du type “missions type”. « lesMissions » sera le nom du champ dans la table « projet » ; « missionsTable » sera le nom de la table imbriquée.

SQL]

```
CREATE TABLE projet (  
    nom varchar(64),  
    dateDebut date,  
    dateFin date,  
    lesMissions missions_type  
) NESTED TABLE lesMissions STORE AS missionsTable;
```

R] Table created.

3) En une seule requête, insérer le projet avec ses 2 missions.

SQL]

```
INSERT INTO projet VALUES ('CA1', '12/12/2009', '13/07/2011',
    missions_type(
        mission_type('SFG', 'Spécifications générales', 30, (select ref(i) from
intervenant i where mat=7655)),
        mission_type('SFD', 'Spécifications détaillées', 60, (select ref(i) from
intervenant i where mat=2516))
    )
);
```

R] 1 row created

4) Affichez une description de la table “projet” puis les caractéristiques du projet.

SQL]

```
desc projet;
```

R]

Name	Null?	Type
NOM		VARCHAR2(64)
DATEDEBUT		DATE
DATEFIN		DATE
LESMISSIONS		MISSIONS_TYPE

SQL]

```
select nom, dateDebut, dateFin from projet where nom='CA1';
```

R]

NOM	DATEDEBUT	DATEFIN
CA1	12/12/09	13/07/11

5) Affichez les différentes missions du projet. Il faut pour cela d'abord accéder à la table imbriquée.

SQL]

```
select m.* from projet p, table(p.lesMissions) m where p.nom = 'CA1';
```

R]

CODE	INTITULE	NBJOURS	INTERVENANT
SFG	Spécifications générales	30	0000220208421E29BDFC82EEDFE05 0820A0EA35BF0421E29BDFC7FEED FE050820A0EA35BF0
SFD	Spécifications détaillées	60	0000220208421E29BDFC81EEDFE05 0820A0EA35BF0421E29BDFC7FEED FE050820A0EA35BF0

6) Affichez les différentes missions du projet avec les noms d'intervenant et leur qualification.

SQL]

```
select m.code, m.intitule, m.nbjours, deref(m.intervenant).nom nom,
deref(m.intervenant).qualification qualification
from projet p, table(p.lesMissions) m where p.nom = 'CA1';
```

R]

CODE	INTITULE	NBJOURS	NOM	QUALIFICATION
SFG	Spécifications générales	30	Henri	Consultant
SFD	Spécifications détaillées	60	Dupont	Développeur

7) Affichez les différentes missions du projet avec les noms d'intervenant et leur qualification.

SQL]

```
select p.nom, p.dateFin, m.code, m.nbjours, deref(m.intervenant).nom nom_int,
deref(m.intervenant).qualification qualif_int
from projet p, table(p.lesMissions) m where p.nom = 'CA1';
```

R]

NOM	DATEFIN	CODE	NBJOURS	NOM_INT	QUALIF_INT
CA1	13/07/11	SFG	30	Henri	Consultant
CA1	13/07/11	SFD	60	Dupont	Développeur