

## TP1

-- 1. Créer le type objet T\_ADR défini par :

```
-----  
create or replace type t_adr as object(  
    rue          varchar2(20),  
    mais         number(2),  
    ville        varchar2(20),  
    pays         varchar2(20));
```

--test du schéma du type desc t\_adr;

--- 2. Créer la table objet des adresses TAB\_ADRS à partir du type T\_ADR

```
-----  
create table tab_adrs of t_adr(  
    constraint pk_tab_adr primary key (rue, mais, ville),  
    constraint c_ville_nn CHECK (ville IS not null),  
    constraint c_pays_nn CHECK (pays IS not null));
```

--test du schéma desc tab\_adrs;

--- 3. Insérer des données dans la table TAB\_ADRS

```
insert into tab_adrs values ('rue1', 1, 'ville1', 'maroc');
```

--validation commit;

--- 4. Afficher le contenu de la table SELECT \* FROM TAB\_ADRS;

--- 5. Afficher les références des objets de la table tab\_adrs SELECT REF(D) FROM TAB\_ADRS D;

--- 6. Afficher les différents champs de la table TAB\_ADRS avec leurs références

SELECT REF(D), D.\* FROM TAB\_ADRS D; ou bien SELECT REF(D), VALUE(D) FROM TAB\_ADRS D;

--- 7. Afficher les adresses d'une ville avec leurs références --SELECTIONNER LES ADRESSES DE LA MEME VILLE VILLE2 AVEC LEURS REFERENCES

--SELECT REF(D), D.\* FROM TAB\_ADRS D WHERE D.VILLE='ville2';-- ou bien SELECT REF(D), VALUE(D) FROM TAB\_ADRS D WHERE D.VILLE='ville2';

--- 8. Créer le type personne T\_PERSONNE définir par :

```
-----  
CREATE OR REPLACE TYPE T_PERSONNE AS OBJECT(  
    CIN          VARCHAR2(20),  
    NOM          VARCHAR2(20),  
    PRENOM      VARCHAR2(20),  
    AGE         NUMBER(2),  
    ADR_PRIVÉ   T_ADR,  
    ADR_PUB     REF T_ADR);
```

--- 9. Créer la table objet TAB\_PERS à partir du type T\_PERSONNE

```

CREATE TABLE TAB_PERS OF T_PERSONNE(
    CONSTRAINT PK_TAB_PERS PRIMARY KEY (CIN),
    CONSTRAINT C_NOM_NN CHECK (NOM IS not null),
    CONSTRAINT C_AGE_NN CHECK (AGE IS NOT NULL),
    CONSTRAINT C_ADR_PRIV_NN CHECK (ADR_PRIVÉ IS NOT NULL),
    CONSTRAINT C_REF_ADR_PUB SCOPE IS TAB_ADRS);

-----

--10. INSERTION DANS LA TABLE TAB_PERS
-- MEME ADRESSE PUBLIQUES
INSERT INTO TAB_PERS VALUES ('k12434', 'nom1', 'prenom1', 34, T_ADR('r1', 1, 'v1', 'maroc'),
    (SELECT REF(D) FROM TAB_ADRS D WHERE D.RUE='rue1' AND D.MAIS=1 AND D.VILLE='ville1'));
-- ADRESSE PUBLIQUES DIFFERENTE
INSERT INTO TAB_PERS VALUES ('X12434', 'nom3', 'prenom3', 24, T_ADR('r3', 3, 'v3', 'maroc'),
    (SELECT REF(D) FROM TAB_ADRS D WHERE D.RUE='rue1' AND D.MAIS=2 AND D.VILLE='ville2'));
-- LES PERSONNES DONT L ADRESSE PUBLIQUE EST NULL
INSERT INTO TAB_PERS VALUES ('AQQ12434', 'nom11', 'prenom11', 63, T_ADR('rue11', 11, 'ville11', 'maroc'),
    (SELECT REF(D) FROM TAB_ADRS D WHERE D.RUE='rue11' AND D.MAIS=11 AND D.VILLE='ville11'));
-- PERSONNES POUR SUPPRIMER SON ADRESSE REFERENCE
INSERT INTO TAB_PERS VALUES ('QQQQQ', 'nom55', 'prenom55', 43, T_ADR('rue55', 55, 'ville55', 'maroc'),
    (SELECT REF(D) FROM TAB_ADRS D WHERE D.RUE='rueQQ' AND D.MAIS=55 AND D.VILLE='ville55'));
INSERT INTO TAB_PERS VALUES ('XXXXX', 'nom88', 'prenom88', 88, T_ADR('rue88', 88, 'ville88', 'maroc'),
    NULL);

-----

-- 11.    Afficher les adresses privées de toutes les personnes SELECT ADR_PRIVÉ FROM TAB_PERS;

-----

-- 12.    Afficher la rue et le numéro de maison des adresses privées de toutes les personnes
--APPLATISSEMENT DES RUE ET MAISON-- REMARQUE: SANS ALIAS CA NE MARCHE PAS
SELECT D.ADR_PRIVÉ.RUE, D.ADR_PRIVÉ.MAIS FROM TAB_PERS D;

-----

-- 13.    Afficher les adresses publiques de toutes les personnes-- directement les références stockées dans la colonne
ADR_PUB SELECT ADR_PUB FROM TAB_PERS;
-- si on veut les adresses référencées et non les références--LES ADRESSES PUBLIQUES DES PERSONNES
SELECT Deref(ADR_PUB) FROM TAB_PERS;

-----

-- 14.    Afficher la rue et le numéro de maison des adresses publiques de toutes les personnes
SELECT D.ADR_PUB.RUE, D.ADR_PUB.MAIS FROM TAB_PERS D;
-- on peut le faire également en passant par la table des adresses
select d.rue, d.mais from tab_adrs d where REF(d) IN (select ADR_PUB from tab_pers);
-- ESSAI AVEC Deref ET PROJECTION-- REMARQUE: AVEC Deref CA NE MARCHE PAS
SELECT Deref(ADR_PRIVÉ) AS AP, AP.RUE, AP.MAIS FROM TAB_PERS;

-----

-- 15.    Afficher les personnes qui habitent la même rue de la même ville (adresses privées)
SELECT DISTINCT(A.CIN), A.NOM, A.PRENOM, A.ADR_PRIVÉ.RUE, B.ADR_PRIVÉ.RUE, A.ADR_PRIVÉ.VILLE, B.ADR_PRIVÉ.VILLE
FROM TAB_PERS A, TAB_PERS B WHERE A.ADR_PRIVÉ.RUE=B.ADR_PRIVÉ.RUE AND A.ADR_PRIVÉ.VILLE=B.ADR_PRIVÉ.VILLE
AND A.CIN<>B.CIN;

```

-----  
-- 16. Afficher les personnes qui ont la même adresse privée que la personne de cin t12434

```
SELECT DISTINCT(B.CIN), B.NOM, B.PRENOM --,A.ADR_PRIVÉ.RUE,  
B.ADR_PRIVÉ.RUE,A.ADR_PRIVÉ.VILLE,B.ADR_PRIVÉ.VILLE FROM TAB_PERS A, TAB_PERS B  
WHERE A.CIN='T12434' AND A.ADR_PRIVÉ.RUE=B.ADR_PRIVÉ.RUE AND A.ADR_PRIVÉ.VILLE=B.ADR_PRIVÉ.VILLE AND  
A.CIN<>B.CIN;
```

-----  
-- 17. Afficher les personnes qui ont la même adresse publique

```
SELECT DISTINCT(CIN), NOM, PRENOM FROM TAB_PERS WHERE ADR_PUB =(SELECT ADR_PUB FROM TAB_PERS WHERE  
CIN='T12434');
```

-- MEME CHOISE AVEC in

```
SELECT DISTINCT(CIN), NOM, PRENOM FROM TAB_PERS WHERE ADR_PUB IN (SELECT ADR_PUB FROM TAB_PERS WHERE  
CIN='T12434');
```

-- autre façon: jointure

```
select distinct(CIN), nom, prenom from tab_pers p1, tab_pers p2 where p1.cin='T12434' AND p1.ADR_PUB=p2.ADR_PUB  
AND p1.CIN<>p2.CIN;
```

-- LES PERSONNES QUI ONT LA MEME ADRESSE PUBLIQUE

```
SELECT DISTINCT(A.CIN), A.NOM, A.PRENOM FROM TAB_PERS A, TAB_PERS B WHERE A.ADR_PUB = B.ADR_PUB AND  
A.CIN<>B.CIN;
```

-----  
-- 18. Afficher les personnes qui ont la même adresse publique et privée

```
SELECT DISTINCT(A.CIN), A.NOM, A.PRENOM FROM TAB_PERS A, TAB_PERS B  
WHERE A.ADR_PUB.RUE = B.ADR_PRIVÉ.RUE AND A.ADR_PUB.VILLE = B.ADR_PRIVÉ.VILLE  
AND A.ADR_PUB.PAYS = B.ADR_PRIVÉ.PAYS AND A.ADR_PUB.MAIS = B.ADR_PRIVÉ.MAIS AND A.CIN=B.CIN;
```

-----  
---19. Afficher les personnes dont l'adresse publique est null

```
SELECT DISTINCT(A.CIN), A.NOM, A.PRENOM FROM TAB_PERS A WHERE A.ADR_PUB IS NULL;
```

-----  
-- 20. Dans la table TAB\_ADRS, supprimer quelques adresses connaissant les personnes

-- qui les utilisent

-- TEST DE SUPPRESSION D UNE ADRESSE REFERENCEE

-- TEST DES PERSONNES CONCERNEES

-----  
SELECT CIN, NOM, PRENOM, ADR\_PUB FROM TAB\_PERS WHERE CIN='QQQQQ' OR CIN='WWWWW';

-- SUPPRESSION DES ADRESSES PUBLIQUES

```
DELETE TAB_ADRS D WHERE REF(D) IN (SELECT ADR_PUB FROM TAB_PERS WHERE CIN='QQQQQ' OR CIN='WWWWW');
```

-- TEST DES PERSONNES DONT L ADRESSE PUBLIQUES EST NULL

```
SELECT DISTINCT(A.CIN), A.NOM, A.PRENOM FROM TAB_PERS A WHERE A.ADR_PUB IS NULL;
```

-- 21. Afficher les personnes dont l'adresse publique est dégligné

-- (l'objet référencé n'existe plus dans la table TAB\_ADRS)

-----  
SELECT CIN, NOM, PRENOM FROM TAB\_PERS WHERE ADR\_PUB NOT IN (SELECT REF(D) FROM TAB\_ADRS D) AND ADR\_PUB  
IS NOT NULL;

-- SI ON ELEVE NULL-- CETTE REQUETE NE PREND QUE LES REF NOT NULL

```
SELECT CIN, NOM, PRENOM FROM TAB_PERS WHERE ADR_PUB NOT IN (SELECT REF(D) FROM TAB_ADRS D);
```

-- CETTE REQUETE DONNE D'AUTRES NOMS--> LE NULL N EST PAS CONSIDRE COMME FAISANT PARTIE DES REF DE  
TAB\_ADRS

```

SELECT CIN, NOM, PRENOM FROM TAB_PERS WHERE ADR_PUB IS NULL;
-- METTRE A JOUR LES REF DEGLINGUEES A NULL
UPDATE TAB_PERS SET ADR_PUB=NULL WHERE ADR_PUB NOT IN (SELECT REF(D) FROM TAB_ADRS D);

```

## TP2

--1.1

```

create or replace type t_adr as object(
    rue          varchar2(20),
    mais         number(2),
    ville        varchar2(20),
    pays         varchar2(20),
    constructor function t_adr(r IN varchar2, m IN Number, v IN varchar2) RETURN SELF AS
RESULT,
    constructor function t_adr(r IN varchar2, m IN Number) RETURN SELF AS RESULT,
    --- LES METHODES ORDER DOIVENT RENVOYER UN INTEGER ET NON UN BOOLEAN
    -- LA FONCTION ORDER RETOURNE -1 POUR <; 0 POUR = ET 1 POUR >
    ORDER MEMBER FUNCTION COMPARE_ADR(ADR IN T_ADR) RETURN INTEGER,
    member procedure modifier_rue (r IN varchar2),
    static procedure ajouter_adr(r IN varchar2, m IN number, v IN varchar2, p IN varchar2),
    static procedure ajouter_adr(A IN T_ADR),
    static procedure supprimer_adr(r IN varchar2, m IN number),
    static procedure supprimer_avec_maj(r IN varchar2, m IN number));

```

--1.2 TP précédent

--2.1 T\_PERS

```

CREATE OR REPLACE TYPE T_PERS AS OBJECT(
    CIN          VARCHAR2(20),
    NOM          VARCHAR2(20),
    PRENOM       VARCHAR2(20),
    AGE          NUMBER(2),
    ADR_PRIV     T_ADR,
    ADR_PUB      REF T_ADR,
    constructor function T_PERS(c IN varchar2, n IN varchar2, p IN varchar2, a IN NUMBER) RETURN SELF
AS RESULT,
    constructor function T_PERS(c IN varchar2, n IN varchar2, p IN varchar2) RETURN SELF AS
RESULT,
    member function AGE_PERS RETURN NUMBER,
    member procedure modifier_ADR_PRIV_PERS (a IN T_ADR),
    member procedure modifier_ADR_PUB_PERS (R IN REF T_ADR),
    member FUNCTION MEME_ADR_PRIVEE_PERS(ADR_PRIV IN T_ADR) RETURN BOOLEAN,
    static procedure ajouter_PERS(c IN varchar2, n IN varchar2, p IN varchar2, a IN number),
    static procedure ajouter_PERS(P IN T_PERS),
    static procedure supprimer_PERS(P IN T_PERS));

```

--2.2 TP précédent

--3.1 les méthodes T\_ADR

```

create or replace type body t_adr as
    constructor function t_adr(r IN varchar2, m IN Number, v IN varchar2) RETURN SELF AS
RESULT IS
    BEGIN
        SELF.rue:=r;SELF.mais:=m;SELF.ville:=v;SELF.PAYS:='maroc';
        RETURN;
    END;

```

```

END t_adr;
constructor function t_adr(r IN varchar2, m IN Number) RETURN SELF AS RESULT IS
    BEGIN
        SELF.rue:=r;SELF.mais:=m;SELF.ville:='rabat';SELF.PAYS:='maroc';
        RETURN;
END t_adr;
ORDER MEMBER FUNCTION COMPARE_ADR(ADR IN T_ADR) RETURN INTEGER IS
    BEGIN
        IF      SELF.RUE=ADR.RUE AND SELF.MAIS=ADR.MAIS AND SELF.VILLE=ADR.VILLE
            AND SELF.PAYS=ADR.PAYS
        THEN RETURN 0;
        ELSE RETURN -1;
        END IF;
    END COMPARE_ADR;
member procedure modif_rue (r IN varchar2) IS
    BEGIN
        UPDATE TAB_ADRS SET rue=r WHERE rue=SELF.rue;
    END modif_rue;
static procedure ajouter_adr(r IN varchar2, m IN number, v IN varchar2, p IN varchar2) IS
    BEGIN
        insert into tab_adrs values (r, m, v, p);
    END ajouter_adr;
static procedure ajouter_adr(A IN T_ADR) IS
    BEGIN
        --insert into tab_adrs values (A.rue, A.mais, A.ville, A.pays);
        insert into tab_adrs values (A);commit;
    END ajouter_adr;
static procedure supprimer_adr(r IN varchar2, m IN number) IS
    REF_ADR          REF T_ADR; -- LA REF DE L'OBJET A SUPPRIMER
    BEGIN
        DELETE TAB_ADRS WHERE RUE=r AND MAIS=m;COMMIT;
    END supprimer_adr;
static procedure supprimer_avec_maj(r IN varchar2, m IN number) IS
    REF_ADR          REF T_ADR;
    BEGIN
        -- AU LIEU D'UTILISER UN TRIGGER, JE FAIS LE TRAITEMENT DIRECTEMENT DANS CETTE FONCTION
        SELECT REF(A) INTO REF_ADR FROM TAB_ADRS A WHERE A.RUE=R AND A.MAIS=M;
        UPDATE TAB_PERS SET ADR_PUB=NULL WHERE ADR_PUB=REF_ADR;
        DELETE TAB_ADRS WHERE RUE=r AND MAIS=m;COMMIT;
    END supprimer_avec_maj;
END;
--3.2.3 Test des méthodes
-- TEST DES CONSTRUCTEUR
-- ON PEUT INVOQUER UN CONSTRUCTEUR DIRECTEMENT DANS INSERT INTO COMME:
-- IL FAUT JUSTE QUE LE CONSTRUTEUR CONSTRUIT LES DIFFERENTS CHAMPS DE L'OBJET
INSERT INTO TAB_ADRS VALUES ( T_ADR('XXX',11) );
--test de ajouter_adr
t_adr.ajouter_adr('marue', 11, 'maville', 'mon pays');commit;
--test d'insertion select * from tab_adrs;
--test de supprimer
t_adr.supprimer_adr('marue', 11);commit;
--tester de la suppression d'un objet select * from tab_adrs;

```

```

--test des constructeur
    ADR    T_ADR
    ADR:=NEW T_ADR('sarue', 11);t_adr.ajouter_adr(ADR);commit;
--TEST AVEC UNIQUEMENT DES VALEURS DE L OBJET ADR
DECLARE
    ADR    T_ADR;
BEGIN
    ADR:=NEW T_ADR('sarue', 22);t_adr.ajouter_adr(ADR.RUE, ADR.MAIS, ADR.VILLE,
ADR.PAYS);commit;
END;        SELECT * FROM TAB_ADRS;
--TEST D INSERTION AVEC UN OBJET COMME ARGUMENT
DECLARE
    ADR    T_ADR;
BEGIN
    ADR:=NEW T_ADR('RIEN', 99);t_adr.ajouter_adr(ADR);commit;END;
-- la description des méthodes
COL TYPE_NAME FORMAT A15
COL METHOD_NAME FORMAT A20
COL METHOD_TYPE FORMAT A15
SELECT TYPE_NAME, METHOD_NAME, METHOD_TYPE, PARAMETERS, RESULTS
    FROM USER_TYPE_METHODS WHERE TYPE_NAME = 'T_ADR';
COL PARAM_NAME FORMAT A15
COL METHOD_NAME FORMAT A15
COL PARAM_TYPE_NAME FORMAT A20
SELECT METHOD_NAME, METHOD_NO, PARAM_NAME, PARAM_TYPE_NAME
    FROM USER_METHOD_PARAMS WHERE TYPE_NAME = 'T_ADR';
SELECT METHOD_NAME, METHOD_NO,RESULT_TYPE_NAME
    FROM USER_METHOD_RESULTS WHERE TYPE_NAME = 'T_ADR';
--TEST DE REF select ref(d) from tab_adrs d;
--test de value select value(d) from tab_adrs d;
--test de value et le lien de l'objet avec l'objet dans la table
DECLARE
    ADR            T_ADR;
    PAYSTEST       varchar2(20);
BEGIN
    SELECT VALUE(D) INTO ADR FROM TAB_ADRS D WHERE D.RUE='rue1' and D.MAIS=1;
    DBMS_OUTPUT.PUT_LINE('objet recupere : '||ADR.RUE ||' '||ADR.MAIS||' '||ADR.VILLE||'
'||ADR.PAYS);
    ADR.PAYS:='TOTO';
    DBMS_OUTPUT.PUT_LINE('objet modif : '||ADR.RUE ||' '||ADR.MAIS||' '||ADR.VILLE||'
'||ADR.PAYS);
    SELECT VALUE(D) INTO ADR FROM TAB_ADRS D WHERE D.RUE='rue1' and D.MAIS=1;
    DBMS_OUTPUT.PUT_LINE('objet table : '||ADR.RUE ||' '||ADR.MAIS||' '||ADR.VILLE||'
'||ADR.PAYS);
END;

```