

Bases de Données Avancées (Suite du cours : l'héritage).

Considérons à nouveau la syntaxe de création d'un type.

```
CREATE [OR REPLACE] TYPE schema.nomType [AS OBJECT |  
UNDER schema.nomSurType ]  
(colonne1 type1, ...  
methode1 (paramètres), ...)  
[[NOT] INSTANTIABLE]  
[[NOT] FINAL]  
/
```

Figure 01

Employe_type

codemp	etat_civil_t		adresse_t			paye
	nom	datenais	nrue	rue	ville	

hérite de

Pilote_type

codemp	etat_civil_t		dresse_t			paye	nbHvol	compagnie
	nom	datenais	nrue	rue	ville			

hérite de

Pil_instructeur_type

codemp	etat_civil_t		adresse_t			paye	nbHvol	compagnie	nbHvolIns	expireQualif
	nom	datenais	nrue	rue	ville					

Code 01

```
CREATE OR REPLACE TYPE etat_civil_type AS OBJECT  
(nom VARCHAR(30), datenais DATE)  
/  
CREATE OR REPLACE TYPE adresse_type AS OBJECT  
(nrue NUMBER(3), rue VARCHAR(40), ville VARCHAR(30))  
/  
CREATE OR REPLACE TYPE Employe_type AS OBJECT  
(codemp CHAR(6), etat_civil_t etat_civil_type, adresse_t adresse_type,  
paye NUMBER(6,2))  
NOT FINAL  
/  
CREATE TYPE Pilote_type UNDER Employe_type  
(nbHvol NUMBER, compagnie VARCHAR(6))  
NOT FINAL  
/  
CREATE TYPE Pil_instructeur_type UNDER Pilote_type  
(nbHvolIns NUMBER, expireQualif DATE)  
FINAL/  
/
```

Code 02

```
CREATE OR REPLACE TYPE etat_civil_type AS OBJECT  
(nom VARCHAR(30), datenais DATE)  
/  
CREATE OR REPLACE TYPE adresse_type AS OBJECT  
(nrue NUMBER(3), rue VARCHAR(40), ville VARCHAR(30))  
/  
CREATE OR REPLACE TYPE Employe_type AS OBJECT  
(codemp CHAR(6), etat_civil_t etat_civil_type, adresse_t adresse_type,  
paye NUMBER(6,2))  
INSTANTIABLE NOT FINAL  
/  
/
```

```

CREATE TYPE Pilote_type UNDER Employe_type
(nbHvol NUMBER, compagnie VARCHAR(6))
NOT INSTANTIABLE NOT FINAL
/
CREATE TYPE Pil_instructeur_type UNDER Pilote_type
(nbHvolIns NUMBER, expireQualif DATE)
INSTANTIABLE FINAL
/

```

Code 03

```

DECLARE
    p2 Pilote_type;
BEGIN
    p2 := NEW Pilote_type('E1',etat_civil_type('Diffis','05-02-1970'),adresse_type(1,'G. Brassens','Blagnac'),3500, 1000, 'ALIB');
END;

```

Erreur a la ligne : tentative d'instanciation d'un type NOT INSTANTIABLE

Code 04

```

CREATE TABLE Employe OF Employe_type
(CONSTRAINT pk_Employe PRIMARY KEY(codemp));

CREATE TABLE Pilote OF Pilote_type
(CONSTRAINT pk_Pilote PRIMARY KEY(codemp));

CREATE TABLE Pil_instructeur OF Pil_instructeur_type
(CONSTRAINT pk_Pil_instructeur PRIMARY KEY(codemp));

```

Code 05

```

INSERT INTO Pilote
VALUES ('E1',
        etat_civil_type('Diffis','05-02-1970'),
        adresse_type(1,'G. Brassens','Blagnac'),3500, 1000, 'ALIB');

INSERT INTO Pil_instructeur
VALUES ('E2',
        etat_civil_type('Ente','11-09-1968'),
        adresse_type(10,'Perantony','B/leze'),5500,
        1500, 'ALIB', 500, '28-02-2003');

INSERT INTO Employe
VALUES ('E3',
        etat_civil_type('lacombe','05-02-1960'),
        adresse_type(1,'Di Roma','Aucamville'),8500);

```

erreur a la ligne 1 :.....

Code 06

```

CREATE TYPE Pilote_type AS OBJECT
(brevet VARCHAR(6), nom VARCHAR(20))
/
CREATE TYPE PiloteChasse_type UNDER Pilote_type
(baseAerienne VARCHAR(20), surnom VARCHAR(10))
/
CREATE TYPE Avion_type AS OBJECT
(immat VARCHAR(6), pilotePilote_type)
/
CREATE TABLE Avion OF Avion_type
(CONSTRAINT pk_Avion PRIMARY KEY(immat));

```

Insérons trios objets:

Code 07

```
INSERT INTO Avion VALUES
    (Avion_type('F-WTSS', Pilote_type('PL-1','Miranda')));
INSERT INTO Avion VALUES
    (Avion_type('F-ENTE', Pilote_type('PL-2','Ente')));
INSERT INTO Avion VALUES
    (Avion_type('F-PAUL', PiloteChasse_type('PL-3','Soutou','Mt de
Marsan','Faucon')));
```

Code 08

SELECT * FROM Avion;

IMMAT, PILOTE (BREVET, NOM)

WTSS	PILOTE_TYPE('PL-1','Miranda)
F-ENTE	PILOTE_TYPE ('PL-2','Ente)
F-PAUL	PILOTECHASSE_TYPE('PL-3','Soutou','Mt de Marsan','Faucon')

Code 09

SELECT **TREAT(pilote AS PiloteChasse_type) FROM Avion;**

TREAT(PILOTEASPILOTECHASSE_TYPE)(BREVET,
NOM ,BASEAERIENNE, SURNOM)

PILOTECHASSE_TYPE('PL-3', 'soutou', 'Mt de Marsan', 'Faucon')

Code 10

```
DECLARE
    var_pil_chasse PiloteChasse_type;
BEGIN
    SELECT TREAT(pilote AS PiloteChasse_type)
    INTO var_pil_chasse
        FROM Avion WHERE immat ='F-PAUL';
--
    DBMS_OUTPUT.PUT_LINE('Base aérienne : ' ||
var_pil_chasse.baseAerienne);
    EXCEPTION
        WHEN NO_DATA_FOUND THEN
            DBMS_OUTPUT.PUT_LINE('pas de pilote de chasse
pour l'avion F-PAUL.');
```

END;

/

Base aérienne : Mt de Marsan

Procédure PL/SQL terminée avec succès.

Code 11

```
CREATE TYPE Pilote_type AS OBJECT
    (brevet VARCHAR(6), nom VARCHAR(20), nbHvol
NUMBER(5))
    NOT FINAL
/

CREATE TYPE PiloteChasse_type UNDER Pilote_type
    (baseAerienne VARCHAR(20), surnom VARCHAR(10))/

CREATE TYPE PiloteLigne_type UNDER Pilote_type
    (compagnie VARCHAR(15))
/

CREATE TYPE Avion_type AS OBJECT
    (immat VARCHAR(6), pilote Pilote_type /* substitutable */)
/

CREATE TABLE Avion OF Avion_type
    (CONSTRAINT pk_Avion PRIMARY KEY(immat));
```

Code 12

```
INSERT INTO Avion VALUES
    (Avion_type('F-WTSS', Pilote_type('PL-1','Miranda',3500)));
INSERT INTO Avion VALUES
    (Avion_type('F-ENTE', Pilote_type('PL-2','Ente',1000)));
INSERT INTO Avion VALUES
    (Avion_type('F-PAUL', PiloteChasse_type('PL-
3','Soutou',2400,'Mt de Marsan','Faucon')));
```

```
INSERT INTO Avion VALUES
    (Avion_type('F-GLFS', PiloteLigne_type('PL-
4','SanFilippo',7800,'Air-Littoral')));
```

Code 13

```
SELECT VALUE(a) FROM Avion a;

VALUE(A) (IMMAT, PILOTE(BREVET, NOM, NBHVOL)
-----
AVION_TYPE('F-WTSS', PILOTE_TYPE('PL-1','Miranda',3500))
AVION_TYPE ('F-ENTE', PILOTE_TYPE ('PL-2','Ente',1000))
AVION_TYPE ('F-PAUL', PILOTECHASSE_TYPE('PL-
3','Soutou',2400,'Mt de Marsan','Faucon'))
AVION_TYPE ("F-GLFS", PILOTELIGNE_TYPE('PL-
4','SanFilippo',7800,'Air-Littoral'))
```

Code 14

```
SELECT VALUE(a) FROM Avion a WHERE a.pilote IS OF
    (PiloteLigne_type, PiloteChasse_type);

VALUE(A) (IMMAT, PILOTE(BREVET, NOM, NBHVOL)
-----
AVION_TYPE ('F-PAUL', PILOTECHASSE_TYPE('PL-
3','Soutou',2400,'Mt de Marsan','Faucon'))
AVION_TYPE ("F-GLFS", PILOTELIGNE_TYPE('PL-
4','SanFilippo',7800,'Air-Littoral'))
```

Code 15

```
DECLARE
    Var_avi avion_type;

BEGIN
    SELECT VALUE(a) INTO Var_avi FROM Avion a
    WHERE pilote IS OF (PiloteLigne_type) and immat = 'F-GLFS';
    DBMS_OUTPUT.PUT_LINE('brevet' :'| | Var_avi.pilote. brevet
    | |"Avion:"| | Var_avi.immat) ;

END;
Brevet : PL-4 avion : F-GLFS
Procédure PL/SQL terminer avec succès
```

Code 16

```
SELECT a.immat,a.pilote.nom FROM Avion a WHERE a.pilote
IS OF (ONLY Pilote_type);
IMMAT      PILOTE.NOM
-----
F-WTSS     Miranda
F-ENTE     Entre
```