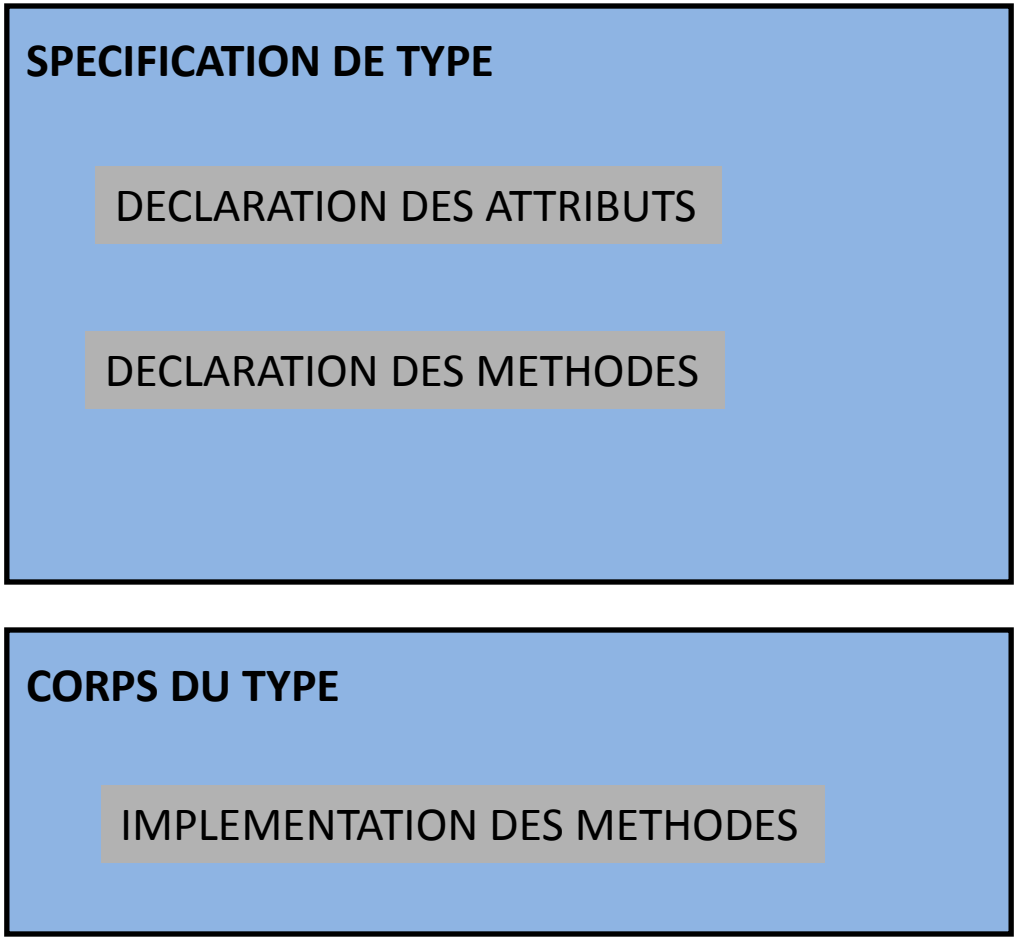
LE MODELE RELATIONNEL ETENDU
TYPE ABSTRAIT
TP1

Les types abstraits (ADT)



Les types abstraits: exemple

```
SQL>
SQL> create or replace type tadr as object(
  2  numero          number(3),
  3  rue             varchar2(20),
  4  code_postal     number(5),
  5  ville           varchar2(20)
  6  );
  7  /

Type créé.

SQL>
```

Les types abstraits: exemple

```
CREATE TYPE tadr AS OBJECT(
        Numero          NUMBER(4),
        Rue             VARCHAR2(20),
        Code_postal     NUMBER(5),
        Ville           VARCHAR2(20)
);
```

Numero	Rue	Code_Postal	Ville

```
CREATE TABLE adresses  OF tadr;
```

```
CREATE TABLE ADRS OF TADR (
 CONSTRAINT PK primary key Numero,
 CONSTRAINT CNN CHECK (rue IS NOT NULL)
);
```

Les types abstraits: exemple

```
CREATE TYPE tadr AS OBJECT(
        Numero          NUMBER(4),
        Rue             VARCHAR2(20),
        Code_postal     NUMBER(5),
        Ville           VARCHAR2(20));
```

REF	Numero	Rue	Code_Postal
XXXXX			
YYYYYY			
ZZZZZZ			

```
CREATE TABLE adresses OF tadr;
```

Les types abstraits: exemple

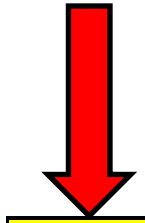
```
Insert into adresses values (tadr (15,'rue',1234,'ville'));
```

```
Insert into adresses values ( 15,'rue',1234,'ville');
```

```
Select * from adresses;
```

```
Select REF(d)
from adresses d;
```

```
Select REF(d), d.*
from adresses d;
```



REF	Numero	Rue	Code_Postal
XXXXX			
YYYYYY			
ZZZZZZ			

Les types abstraits: exemple

```
Insert into adresses values (tadr (15,'rue',1234,'ville'));
```

```
Insert into adresses values ( 15,'rue',1234,'ville');
```

```
Select VALUE(d) from adresses d;
```

REF	Numero	Rue	Code_Postal
XXXXX			
YYYYYY			
ZZZZZZ			

```
Select VALUE(d).rue from adresses d;
```

Les types abstraits: exemple

```
DECLARE
        ADR             TADR;
        REFADR          REF  TADR;
BEGIN
        SELECT REF(D), VALUE(D) INTO ADR, REFADR
        FROM ADRESSES D
        WHERE D.RUE='sarue';
END;
/
```

Les types abstraits: exemple

```
CREATE OR REPLACE TYPE t_personne AS OBEJCT(
        PL              NUMBER(4),
        Nom             VARCHAR2(20),
        Adresse         tadr,
        ……..
        Salaire    NUMBER(10,2)
);
```

PL	NOM	Adresse				….	Salaire
		Numero	Rue	Code_Postal	Ville		

```
CREATE TABLE Employes OF t_personne;
```

Les types abstraits: exemple

```
Insert into Employes values (t_personne(11,'Zaatar',
        tadr (15,'rue',1234,'ville'), …,1000000)

Select * from Employes;

Select Adresse from Employes;

Select d.adresse.rue, d.adresse.ville
From Employes d;
```

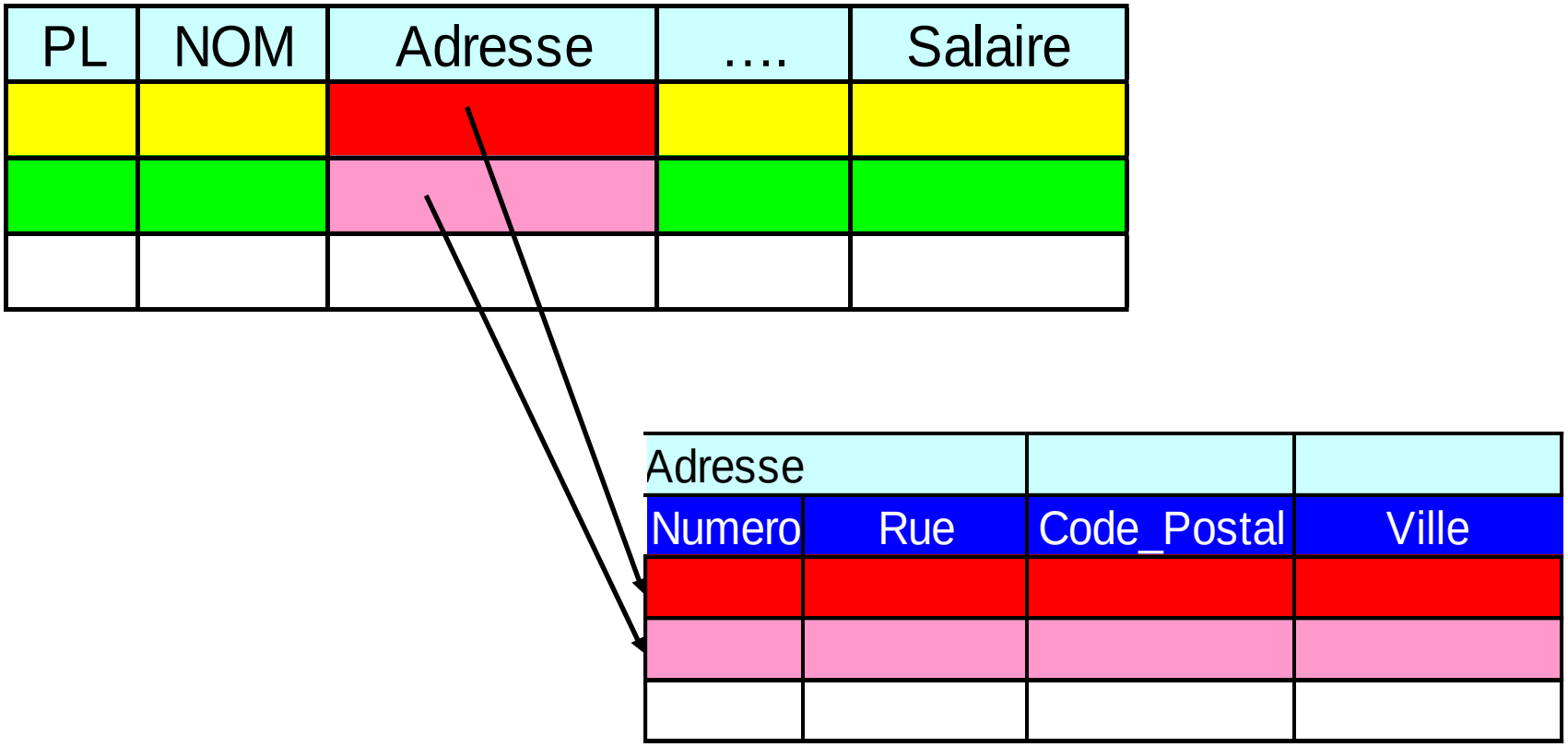
Les types abstraits: exemple

```
CREATE OR REPLACE TYPE t_personne AS OBEJCT(
        PL              NUMBER(4),
        Nom             VARCHAR2(20),
        Adresse         REF tadr,
        ……..
        Salaire    NUMBER(10,2));
```

```
CREATE TABLE PERSONNES OF t_personne(
CONSTRAINT IREF ADRESSE SCOPE IS ADRESSES
);
```

Les types abstraits: exemple

CREATE TABLE PERSONNES OF t_personne;



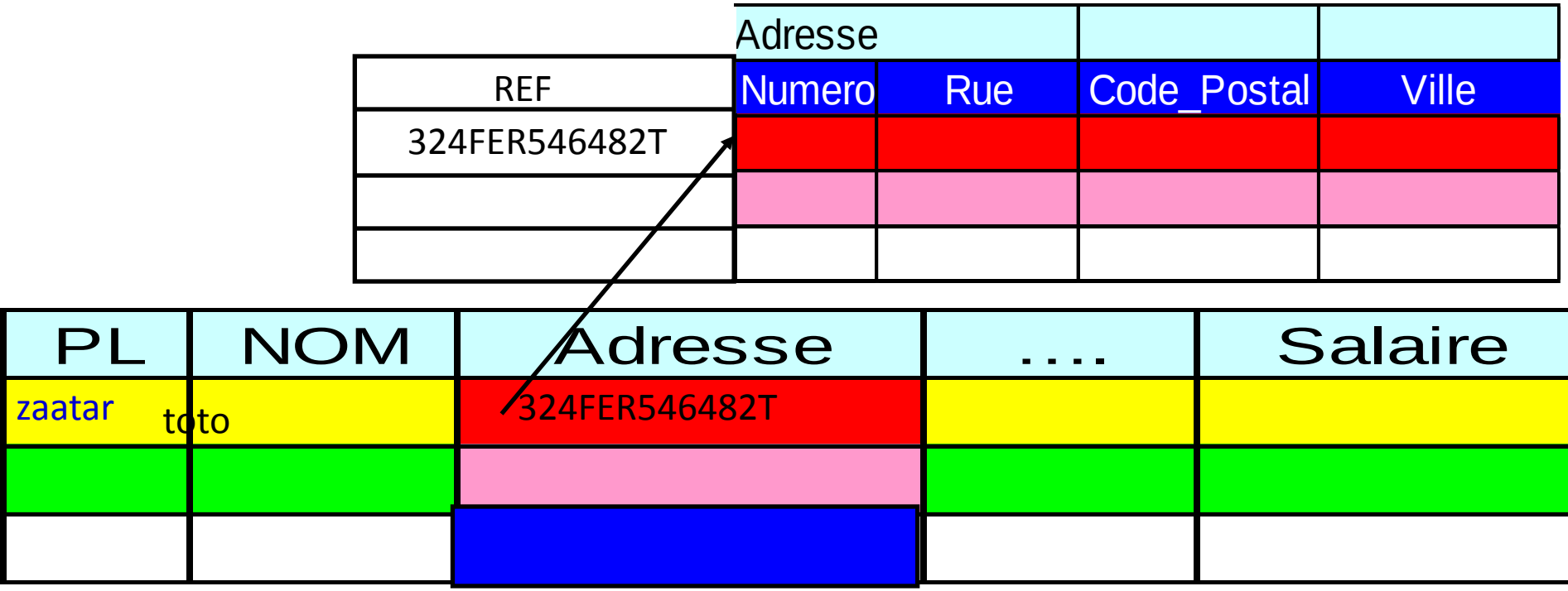
CREATE TABLE OF tadr;

Type abstrait: constructeur

```
CREATE TABLE Employes OF t_personne;


INSERT INTO Employes VALUES (
t_personne(13,'toto',
(select REF(d) from adresses d where d.ville='saville'),
…
100000)
);
```

Les types abstraits: exemple



INSERT INTO Employes VALUES (t_personne(13,'toto',
(select REF(d) from adresses d where d.ville='saville'),
…
1000000);

SELECT **DEREF(ADRESSE)** FROM EMPLOYES;

SELECT **D.ADRESSE.RUE** FROM EMPLOYES **D**;

TYPE ABSTRAIT et PL/SQL

```
SQL>
SQL> declare
  2    VADR  Tadr;
  3  begin
  4    VADR:=NEW T_ADR('r',11,'v','p');
  5    insert into adresses values(VADR);
  6    commit;
  7  end;
  8  /

Procédure PL/SQL terminée avec succès.
```

Les types abstraits (ADT)/ Conclusion

- Permet à l'utilisateur de définir lui même ses propres types
- Pas de vraie identité d'objet mais des pointeurs invariables
- Le développeur gère lui-même les références des objets
- Navigation à travers les références au lieu de la jointure