

# ŚCIGAWEKAZKA BAZOWADWA 3

Adam Sawicki  
2006-11-12

## Typ

```
create or replace type Typ1 as object (  
    Pole1 TypPolal,  
    ...  
);  
  
drop type Typ1;  
  
desc Typ1;  
  
create table Tabelal of Typ1;  
  
create table Tabelal of Typ1 (  
    [Ograniczenie1], ...  
);  
  
select * from Tabelal;  
  
select Pole1, ... from Tabelal;  
  
select value(A) from Tabelal A;
```

## Metoda

```
create or replace type Typ1 as object (  
    ...  
    member procedure Procedural,  
    member function Funkcj1  
        return TypFunkcji  
);  
  
create or replace type body Typ1 as  
member procedure Procedural is  
    ...  
begin  
    ...  
end [Procedural];  
end;  
  
create or replace type body Typ1 as  
member function Funkcj1  
return TypFunkcji is  
    ...  
begin  
    ...  
end [Funkcj1];  
end;
```

## Typ zbiorowy

```
create type Typy1 as varray(5) of Typ1;  
  
create table Tabelal (  
    Pole1 Typy1  
);  
  
insert into Tabelal values (  
    Typ1(Wartosc11, Wartosc12),  
    Typ1(Wartosc21, Wartosc22)  
);  
  
select T.Pole1 from Tabelal T;  
  
select * from the (  
    select T.Pole1 from Tabelal where ...  
);
```

## Tabele zagnieżdżone

```
create type Projekt as object (  
    Nazwa varchar2(32)  
);  
  
create type Projekty as table of Projekt;  
  
create table Ludzie (  
    Imie varchar2(32),
```

```
    CoRobi Projekty  
)  
nested table CoRobi store as Ludzie_CoRobi;  
  
insert into Ludzie values (  
    'Adam',  
    Projekty (  
        Projekt('The Final Quest'),  
        Projekt('Return to Black Forest')  
    )  
);  
  
select * from the (  
    select CoRobi from Ludzie where Imie='Adam'  
);  
  
insert into the (  
    select CoRobi from Ludzie where Imie='Adam'  
) values (Projekt('Zaliczenie'));  
  
update the (...  
  
delete from the (...
```

## Perspektywa obiektowa

```
create table TblLudzie (  
    CzlowiekId number primary key,  
    Imie varchar2(32)  
);  
  
create table TblProjekty  
(  
    ProjektId number primary key,  
    CzlowiekId number,  
    Nazwa varchar2(32),  
    foreign key (CzlowiekId)  
        references TblLudzie(CzlowiekId)  
);  
  
insert into TblLudzie values (1, 'Adam');  
insert into TblLudzie values (2, 'Lamer');  
  
insert into TblProjekty values  
    (1, 1, 'The Final Quest');  
insert into TblProjekty values  
    (2, 1, 'Return to Black Forest');  
insert into TblProjekty values  
    (3, 2, 'Hello World');  
  
select * from TblLudzie natural join TblProjekty;  
  
create or replace type TypProjekt as object (  
    ProjektId number,  
    Nazwa varchar2(32)  
);  
  
create or replace type TypProjekty as  
table of TypProjekt;  
  
create or replace type TypCzlowiek as object (  
    CzlowiekId number,  
    Imie varchar2(32),  
    Projekty TypProjekty  
);  
  
create or replace view ViewLudzie of TypCzlowiek  
with object identifier (CzlowiekId) as  
select L.CzlowiekId, L.Imie, cast(multiset(  
    select ProjektId, Nazwa  
    from TblProjekty  
    where CzlowiekId = L.CzlowiekId  
) as TypProjekty)  
from TblLudzie L;  
  
select * from ViewLudzie;
```

## Dziedziczenie

```
create or replace type Osoba as object (  
    Imie varchar2(32)  
) not final;  
  
create or replace type Student under Osoba (  
    Kierunek varchar2(32)  
);  
  
create table Osoby of Osoba;  
  
insert into Osoby  
values (Osoba('Lamer'));  
insert into Osoby  
values (Student('Adam', 'Informatyka'));  
  
select * from Osoby;  
  
select treat(value(T) as Student)  
from Osoby T  
where value(T) is of (Student);  
  
select Imie, treat(value(T) as Student).Kierunek  
from Osoby T  
where value(T) is of (Student);
```

## Metoda sortująca

```
create or replace type Projekt as object (  
    Numer number,  
    Nazwa varchar2(32),  
    order member function Sortuj (p Projekt)  
        return integer  
);  
  
create or replace type body Projekt as  
order member function Sortuj (p Projekt)  
return integer  
is  
begin  
    return p.Numer - Numer;  
end;  
end;  
  
create or replace type Projekt2 as object (  
    Numer number,  
    Nazwa varchar2(32),  
    map member function Sortuj  
        return integer  
);  
  
create or replace type body Projekt2 as  
map member function Sortuj  
return integer is  
begin  
    return Numer;  
end;  
end;  
  
create table Osoby (  
    ImieOsoby varchar2(32),  
    ProjektOsoby Projekt  
);  
  
insert into Osoby values  
('Adam', Projekt(20, 'The Final Quest'));
```

```
insert into Osoby values  
('Adam', Projekt(30, 'Return to Black Forest'));  
insert into Osoby values  
('Lamer', Projekt(1, 'Hello World'));
```

```
select * from Osoby order by ProjektOsoby;
```

## Metoda zliczająca

```
create or replace type Projekt as object (  
    Nazwa varchar2(32)  
);  
  
create or replace type Projekty as table of Projekt;  
  
create or replace type Osoba as object (  
    Imie varchar2(32),  
    CoRobi Projekty,  
    member function Liczba  
        return integer  
);  
  
create or replace type body Osoba as  
member function Liczba  
return integer  
is  
begin  
    return CoRobi.count();  
end;  
end;  
  
create table Osoby of Osoba  
nested table CoRobi store as Osoby_Projekty;  
  
insert into Osoby values (  
    'Adam', Projekty(  
        Projekt('The Final Quest'),  
        Projekt('Return to Black Forest')  
    )  
);  
insert into Osoby values (  
    'Lamer', Projekty(  
        Projekt('Hello World')  
    )  
);  
  
select T.Imie, T.Liczba() from Osoby T;  
  
create or replace type Osoba as object (  
    Imie varchar2(32),  
    CoRobi Projekty,  
    member function LiczbaZnakow  
        return integer  
);  
  
create or replace type body Osoba as  
member function LiczbaZnakow  
return integer  
is  
    R integer := 0;  
    I integer;  
begin  
    for I in 1..self.CoRobi.count loop  
        R := R + length(self.CoRobi(I).Nazwa);  
    end loop;  
    return R;  
end;  
end;
```