

2. Tworzenie obiektów

Czas trwania: 60 minut

Przystąpimy teraz do przeglądu mechanizmów obiektowych oferowanych przez RDBMS Oracle

1. Utwórz typ obiektowy PRACOWNIK.

```
CREATE TYPE Pracownik AS OBJECT (  
    nazwisko VARCHAR2(20),  
    pensja    NUMBER(6,2),  
    etat      VARCHAR2(15),  
    data_ur   DATE  
);  
/
```

2. Utwórz tabelę obiektową i wstaw do niej obiekt reprezentujący pracownika Kowalskiego

```
CREATE TABLE PracownicyObjTab OF PRACOWNIK;  
  
INSERT INTO PracownicyObjTab VALUES (  
    NEW Pracownik('Kowalski',2500,'ASYSTENT',DATE '1965-07-01'));
```

3. Wyświetl zawartość tabeli obiektowej w trybie dostępu relacyjnego i obiektowego.

```
SELECT * FROM PracownicyObjTab;  
  
SELECT VALUE(p) FROM PracownicyObjTab p;
```

4. Utwórz tabelę, która będzie zawierała obiektowy atrybut. Wstaw do tabeli krotkę zawierającą obiekt.

```
CREATE TABLE ProjektyTab (  
    symbol CHAR(6),  
    nazwa VARCHAR(100),  
    budzet NUMBER,  
    kierownik Pracownik);  
  
INSERT INTO ProjektyTab VALUES  
('AB 001','Projekt X',20000,  
    NEW Pracownik('Nowak',3200,'ADIUNKT',null));
```

5. Wyświetl zawartość nowoutworzonej tabeli. Sprawdź, jak funkcjonuje dostęp do składowych obiektów za pomocą notacji kropkowej.

```
SELECT nazwa, kierownik FROM ProjektyTab;  
  
SELECT p.kierownik.nazwisko FROM ProjektyTab p;
```

6. Dodaj do definicji typu Pracownik funkcję wyznaczającą wiek pracownika oraz metodę służącą do przyznania pracownikowi podwyżki

```
ALTER TYPE Pracownik REPLACE AS OBJECT (  
  nazwisko VARCHAR2(20),  
  pensja    NUMBER(6,2),  
  etat      VARCHAR2(15),  
  data_ur   DATE,  
  MEMBER FUNCTION wiek RETURN NUMBER,  
  MEMBER PROCEDURE podwyzka(p_kwota NUMBER) );  
/
```

```
CREATE OR REPLACE TYPE BODY Pracownik AS  
  MEMBER FUNCTION wiek RETURN NUMBER IS  
  BEGIN  
    RETURN EXTRACT (YEAR FROM CURRENT_DATE) - EXTRACT (YEAR FROM  
data_ur);  
  END wiek;  
  
  MEMBER PROCEDURE podwyzka(p_kwota NUMBER) IS  
  BEGIN  
    pensja := pensja + p_kwota;  
  END podwyzka;  
END;  
/
```

7. Wyświetl wiek pracowników umieszczonych w tabeli PracownicyObjTab. Następnie, przyznaj 200 zł podwyżki kierownikowi projektu 'AB 001' (tabela ProjektyTab)

```
SELECT p.nazwisko, p.data_ur, p.wiek() FROM PracownicyObjTab p;  
  
DECLARE  
  l_kierownik Pracownik;  
BEGIN  
  SELECT kierownik INTO l_kierownik  
  FROM ProjektyTab WHERE symbol = 'AB 001';  
  
  l_kierownik.podwyzka(200);  
  
  UPDATE ProjektyTab  
  SET kierownik = l_kierownik  
  WHERE symbol = 'AB 001';  
END;
```

8. Wyświetl unikalne identyfikatory (OIDs) obiektów przechowywanych w tabeli PracownicyObjTab.

```
SELECT VALUE(p), REF(p) FROM PracownicyObjTab p;
```

9. Dodaj do typu obiektowego Pracownik metodę, za pomocą której będzie można porównywać ze sobą pracowników korzystając z ich pensji i wieku.

```
ALTER TYPE Pracownik ADD MAP MEMBER FUNCTION odwzoruj  
RETURN NUMBER CASCADE INCLUDING TABLE DATA;
```

```
CREATE OR REPLACE TYPE BODY Pracownik AS  
MEMBER FUNCTION wiek RETURN NUMBER IS  
BEGIN  
    RETURN EXTRACT (YEAR FROM CURRENT_DATE) -  
        EXTRACT (YEAR FROM data_ur);  
END wiek;  
MEMBER PROCEDURE podwyzka(p_kwota NUMBER) IS  
BEGIN  
    pensja := pensja + p_kwota;  
END podwyzka;  
MAP MEMBER FUNCTION odwzoruj RETURN NUMBER IS  
BEGIN  
    RETURN ROUND(pensja,-3) + wiek();  
END odwzoruj;  
END;  
/
```

10. Dodaj do tabeli PracownicyObjTab dwóch nowych pracowników i sprawdź działanie funkcji odwzorowującej.

```
INSERT INTO PracownicyObjTab VALUES  
(NEW Pracownik('Nowak',2000,'ADIUNKT',DATE '1961-02-15'));  
INSERT INTO PracownicyObjTab VALUES  
(NEW Pracownik('Janiak',1800,'ASYSTENT',DATE '1973-12-02'));
```

```
SELECT p.nazwisko, p.pensja, p.wiek()  
FROM PracownicyObjTab p  
ORDER BY VALUE(p);
```

```
SELECT * FROM PracownicyObjTab p  
WHERE VALUE(p) > ( SELECT VALUE(r) FROM PracownicyObjTab r  
                    WHERE r.nazwisko = 'Janiak' );
```

11. Do definicji typu obiektowego Pracownik dodaj konstruktor. W ciele typu obiektowego umieść następnie implementację konstruktora.

```
ALTER TYPE Pracownik REPLACE AS OBJECT (  
  nazwisko VARCHAR2(20),  
  pensja    NUMBER(6,2),  
  etat     VARCHAR2(15),  
  data_ur  DATE,  
  MEMBER FUNCTION wiek RETURN NUMBER,  
  MEMBER PROCEDURE podwyzka(p_kwota NUMBER),  
  MAP MEMBER FUNCTION odwzoruj RETURN NUMBER,  
  CONSTRUCTOR FUNCTION Pracownik(p_nazwisko VARCHAR2)  
    RETURN SELF AS RESULT );  
  
CREATE OR REPLACE TYPE BODY Pracownik AS  
  MEMBER FUNCTION wiek RETURN NUMBER IS  
  BEGIN  
    RETURN EXTRACT (YEAR FROM CURRENT_DATE) -  
          EXTRACT (YEAR FROM data_ur);  
  END wiek;  
  MEMBER PROCEDURE podwyzka(p_kwota NUMBER) IS  
  BEGIN  
    pensja := pensja + p_kwota;  
  END podwyzka;  
  MAP MEMBER FUNCTION odwzoruj RETURN NUMBER IS  
  BEGIN  
    RETURN ROUND(pensja,-3) + wiek();  
  END odwzoruj;  
  
  CONSTRUCTOR FUNCTION Pracownik(p_nazwisko VARCHAR2)  
    RETURN SELF AS RESULT IS  
  BEGIN  
    SELF.nazwisko := p_nazwisko; SELF.pensja := 1000;  
    SELF.etat := null; SELF.data_ur := null;  
    RETURN;  
  END;  
END;  
/
```

12. Przetestuj działanie konstruktora tworząc nowy obiekt i wstawiając go do tabeli PracownicyObjTab.

```
INSERT INTO PracownicyObjTab VALUES (  
  NEW Pracownik('Dziamdziak') );
```

13. Utwórz typy obiektowe Osoba i Adres. Wykorzystaj referencje do powiązania osób z adresami zamieszkania. Następnie, utwórz tabele obiektowe i wypełnij je przykładowymi danymi.

```
CREATE TYPE Adres AS OBJECT (  
    ulica      VARCHAR2(15),  
    dom        NUMBER(4),  
    mieszkanie NUMBER(3));  
/  
  
CREATE TYPE Osoba AS OBJECT (  
    Nazwisko  VARCHAR2(20),  
    Imie      VARCHAR2(15),  
    GdzieMieszka REF Adres);  
/  
  
CREATE TABLE AdresyObjTab OF Adres;  
CREATE TABLE OsobyObjTab OF Osoba;  
  
ALTER TABLE OsobyObjTab ADD  
SCOPE FOR(GdzieMieszka) IS AdresyObjTab;  
  
INSERT INTO AdresyObjTab VALUES  
    (NEW Adres('Kolejowa',2,18));  
INSERT INTO OsobyObjTab VALUES  
    (NEW Osoba('Kowalska', 'Anna', null));  
INSERT INTO OsobyObjTab VALUES  
    (NEW Osoba('Kowalski', 'Jan', null));  
  
UPDATE OsobyObjTab o  
SET o.gdziemieszka = (  
    SELECT REF(a) FROM AdresyObjTab a  
    WHERE a.ulica = 'Kolejowa' );
```

14. Sprawdź różne sposoby wykorzystania nawigacji po referencjach.

```
SELECT o.imie, o.nazwisko, Deref(o.gdziemieszka)  
FROM OsobyObjTab o;  
  
SELECT o.imie, o.nazwisko, o.gdziemieszka.ulica,  
    o.gdziemieszka.dom  
FROM OsobyObjTab o;  
  
SELECT o.imie, o.nazwisko, a.ulica, a.dom, a.mieszkanie  
FROM OsobyObjTab o JOIN AdresyObjTab a  
    ON o.gdziemieszka = REF(a);
```

15. Zaobserwuj zjawisko wiszących referencji.

```
DELETE FROM AdresyObjTab a WHERE a.ulica = 'Kolejowa';  
  
SELECT * FROM OsobyObjTab o WHERE o.gdziemieszka IS NULL;  
  
SELECT * FROM OsobyObjTab o WHERE o.gdziemieszka IS Dangling;
```

3. Opcja obiektowa: ćwiczenia samodzielne

Tworzenie typów obiektowych

1. Zdefiniuj typ obiektowy reprezentujący SAMOCHODY. Każdy samochód powinien mieć markę, model, liczbę kilometrów oraz datę produkcji i cenę. Stwórz tablicę obiektową i wprowadź kilka przykładowych obiektów, obejrzyj zawartość tablicy

```
SQL> desc samochod
```

Nazwa	Wartość NULL?	Typ
MARKA		VARCHAR2 (20)
MODEL		VARCHAR2 (20)
KILOMETRY		NUMBER
DATA_PRODUKCJI		DATE
CENA		NUMBER (10, 2)

```
SQL> select * from samochody;
```

MARKA	MODEL	KILOMETRY	DATA_PRODU	CENA
FIAT	BRAVA	60000	30-11-1999	25000
FORD	MONDEO	80000	10-05-1997	45000
MAZDA	323	12000	22-09-2000	52000

2. Stwórz tablicę WLASCIKIELE zawierającą imiona i nazwiska właścicieli oraz atrybut obiektowy SAMOCHOD. Wprowadź do tabeli przykładowe dane i wyświetl jej zawartość.

```
SQL> desc wlasciciele
```

Nazwa	Wartość NULL?	Typ
IMIE		VARCHAR2 (100)
NAZWISKO		VARCHAR2 (100)
AUTO		SAMOCHOD

```
SQL> select * from wlasciciele;
```

IMIE	NAZWISKO	AUTO(MARKA, MODEL, KILOMETRY, DATA_PRODUKCJI, CENA)
JAN	KOWALSKI	SAMOCHOD('FIAT', 'SEICENTO', 30000, '02-12-0010', 19500)
ADAM	NOWAK	SAMOCHOD('OPEL', 'ASTRA', 34000, '01-06-0009', 33700)

3. Wartość samochodu maleje o 10% z każdym rokiem. Dodaj do typu obiektowego SAMOCHOD metodę wyliczającą na podstawie wieku i przebiegu aktualną wartość samochodu.

```
SQL> SELECT s.marka, s.cena, s.wartosc() FROM SAMOCHODY s;
```

MARKA	CENA	S.WARTOSC()
FIAT	25000	12500
FORD	45000	9000
MAZDA	52000	26000