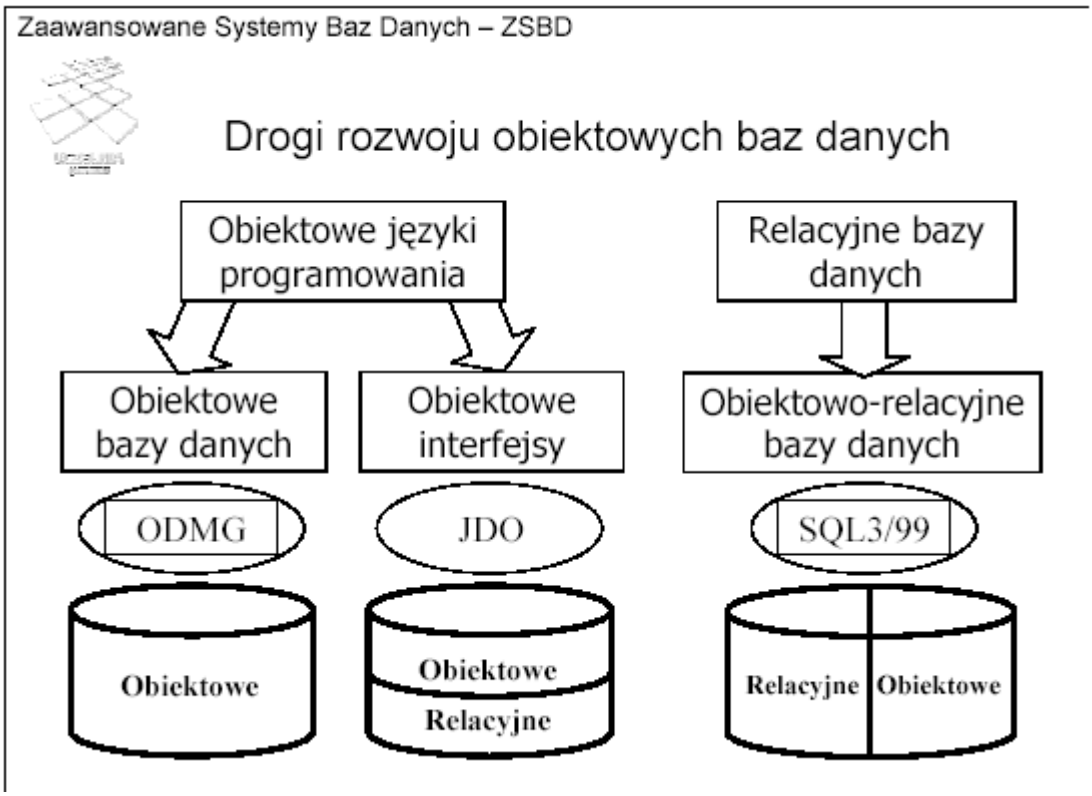
Wykład VIII



KOLEKCJE - to typy masowe,zawierające pewną liczbę jednorodnych elementów

SQL3 wprowadza następujące kolekcje:

- zbiory (**SETS)** - zestaw elementów bez powtórzeń, kolejność nieistotna
- listy (**LISTS)** - zestaw elementów ewentualnie z powtórzeniami, kolejność istotna
- wielozbiory (**MULTISETS**) – możliwe powtórzenia, kolejność nieistotna

W Oracle zaimplementowano dwa rodzaje kolekcji

- *tabele zagnieżdżone* - wartości typu **TABLE,** brak maksymalnego rozmiaru, kolejność elementów nieistotna

CREATE [OR REPLACE] TYPE *Nazwa_kolekcji* **AS TABLE OF** *Typ_elementu* **[NOT NULL]**

- *tablice zmiennej długości* - wartości typu **VARRAY**, określony maksymalny rozmiar, kolejność elementów istotna

CREATE [OR REPLACE] TYPE *Nazwa_kolekcji*
AS VARRAY (*max_rozmiar*) **OF** *Typ_elementu* **[NOT NULL]**

Wykład VIII

przykłady:

1)

/* *utworzenie kolekcji Telefony z elementów typu tekst 15 znakowy* */

CREATE OR REPLACE TYPE Telefony **AS VARRAY(3) OF VARCHAR2(15)**
/

2)

/* *utworzenie typu elementu kolekcji* */

CREATE OR REPLACE TYPE Filia **AS OBJECT**
(Oddzial **NUMBER**(1) **,**
Miasto **VARCHAR2**(25) ,
Telefon **VARCHAR2**(10) **)**
/

/* *utworzenie kolekcji* Filie *o pięciu elementach* */
CREATE OR REPLACE TYPE Filie **AS VARRAY(5) OF** Filia
/

/* *utworzenie tabeli z kolumną typu kolekcja* **/**
CREATE TABLE Banki
(Nazwa **VARCHAR2**(20) ,
Centrala **VARCHAR2**(15) ,
Oddzialy **Filie**) ;

DESC Banki

Nazwa	Wartość NULL?	Typ
NAZWA		VARCHAR2(20)
CENTRALA		VARCHAR2(15)
ODDZIALY		FILIE

INSERT INTO Banki **VALUES** ('PEKAO SA', 'Warszawa',

Filie (**Filia** (2, 'Częstochowa','0-34256-31'), **Filia**(3,'Kraków','0-18134-76'))) ;

Wykład VIII

/* *blok anonimowy; uzupełnie kolekcji - dodanie nowego elementu* */

```
DECLARE
Fil Filie:=Filie();
I INTEGER;
IL INTEGER;
BEGIN
 SELECT Oddzialy  INTO Fil  FROM  Banki WHERE Nazwa='PEKAO SA';
 Fil.EXTEND(1);
IL:= Fil.LAST();
 Fil(IL):=filia(5,'Kielce','04212531');
 UPDATE  Banki  SET Oddzialy=Fil WHERE Nazwa='PEKAO SA';
 FOR  I  IN 1..Fil.LAST() LOOP
  dbms_output.put_line(Fil(I).Oddzial||' '||Fil(i).Miasto);
    END  LOOP;
END;
/
```

/* *prezentacja elementów kolekcji* */

```
SELECT * FROM TABLE
        (SELECT Oddzialy  FROM  Banki  WHERE  Nazwa='PEKAO SA');
SELECT  *  FROM  Banki;

SELECT  b.Nazwa , e.*  FROM  Banki b , TABLE (b.Oddzialy) e;
```

3) /* *przykład kolekcji typu zagnieżdżona tabela* */

```
CREATE OR  REPLACE  TYPE Lokal  AS OBJECT
( Nazwa VARCHAR2( 20 ),
Powierzchnia NUMBER( 6,1 ) )
/

CREATE OR  REPLACE TYPE  Lokale  AS  TABLE  OF  Lokal
/

CREATE OR REPLACE TYPE Budynek  AS OBJECT
( Adres VARCHAR2( 30 ),
 Zasob Lokale ,
 MEMBER  FUNCTION  Liczba_lokali RETURN NUMBER ,
 MEMBER FUNCTION Cal_powierzchnia RETURN NUMBER )
/
```

Wykład VIII

```
CREATE OR REPLACE TYPE BODY Budynek AS
MEMBER FUNCTION Liczba_lokali RETURN NUMBER
IS
BEGIN
  RETURN Zasob.COUNT( ) ;
END Liczba_lokali ;
MEMBER  FUNCTION Cal_powierzchnia RETURN  NUMBER
IS
I NUMBER;
Sumuj NUMBER :=0;
BEGIN
  FOR I IN 1..Zasob.COUNT LOOP
   Sumuj:=Sumuj + Zasob(I).Powierzchnia;
 END LOOP ;
 RETURN Sumuj ;
END Cal_powierzchnia ;
END ;
/
```

```
CREATE  TABLE  Budynki OF  Budynek NESTED TABLE Zasob STORE  AS zas_lokale ;
```

```
INSERT  INTO  Budynki VALUES ( Budynek ('Częstochowa Al. NMP 23', Lokale ( Lokal
('Mieszkanie',63.4), Lokal ('Biuro',78) ) ) );
```

```
SELECT Adres , b.Liczba_lokali( ) , b.Cal_powierzchnia( ) FROM  Budynki b;
```

```
SELECT b.Nazwa, b.Powierzchnia  FROM  TABLE
                ( SELECT Zasob FROM Budynki WHERE Adres LIKE '%K%' ) b;
```

```
INSERT  INTO  TABLE ( SELECT Zasob FROM  Budynki WHERE  Adres LIKE '%K%')
                        VALUES ( Lokal ( 'Biuro PZU' , 125 ) ) ;
```

Kolekcje można tworzyć:

- trwale w bazie danych
- nietrwale – w blokach PL/SQL

Elementami kolekcji mogą być instancje typów obiektowych i na odwrót..

Instancje kolekcji mogą być przekazywane poprzez parametry procedur i funkcji.

Wykład VIII

Operacje na kolekcjach

- możliwe jest przypisanie: kolekcja1 := kolekcja2 o ile obie instancje kolekcji są dokładnie tego samego typu
- kolekcje nie mogą być używane w porównaniach oraz klauzulach DISTINCT, GROUP BY, ORDER BY
- niezainicjowana kolekcja jest równa NULL – nie ma żadnych elementów
- mogą być testowane względem NULL przy użyciu operatorów IS NULL oraz IS NOT NULL
- w zapytaniach można do kolekcji odwoływać się jak do tabel, stosując operator TABLE (THE)
- można budować perspektywy relacyjne udostępniające dane z kolekcji tak jak dane w pełni relacyjne

Metody predefiniowane operujące na kolekcjach

- EXISTS(n) - zwraca prawdę jeżeli n-ty element kolekcji istnieje
- COUNT – zwraca liczbę elementów kolekcji
- LIMIT – zwraca maksymalny rozmiar VARRAY lub NULL dla tabel zagnieżdżonych
- FIRST – zwraca indeks pierwszego elementu kolekcji
- LAST – zwraca indeks ostatniego elementu kolekcji
- PRIOR (n) - zwraca indeks poprzednika n-tego elementu
- NEXT (n) - zwraca indeks następnego po n-tym elemencie
- EXTEND – dodaje pusty element do kolekcji
- EXTEND (n) - dodaje n pustych elementów
- EXTEND (n , e) - dodaje n kopii elementu e
- TRIM – przycina kolekcję o element końcowy
- TRIM (n) - przycina kolekcję o n-elementów końcowych
- DELETE – usuwa wszystkie elementy kolekcji
- DELETE (n) - usuwa n-ty element kolekcji
- DELETE (n1 , n2) - usuwa elementy od n1 do n2

Wykład VIII

przykłady:

/ utworzenie kolekcji Nazwiska, gdzie elementem jest napis na 30 znakach */*

CREATE OR REPLACE TYPE Nazwiska **AS TABLE OF VARCHAR2(30)**
/

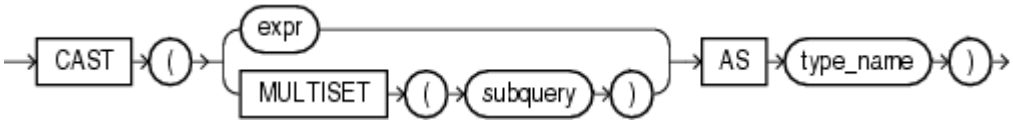
/ prezentacja nazw etatów wraz z kolekcją Nazwisk pracowników zatrudnionych na danym etacie */*

SELECT Nazwa **, CAST (MULTISET**
(SELECT Nazwisko **FROM** Pracownicy **WHERE** Etat = Nazwa **) AS** Nazwiska **)**
Kolekcja
FROM Etaty **;**

/ prezentacja nazwisk szefów wraz z kolekcją nazwisk podwładnych */*

SELECT Nazwisko **, CAST (MULTISET**
(SELECT Nazwisko **FROM** Pracownicy p **WHERE** p.Szef = s.Numer **) AS** Nazwiska **)**
Podwładni
FROM Pracownicy s **WHERE EXISTS**
(SELECT Nazwisko **FROM** Pracownicy p **WHERE** p.szef = s.Numer **) ;**

/* *działanie operatorów* CAST, MULTISET
*konwersja danych z wyrzżenia lub podzapytania do typu kolekcja */*



Wykorzystano
Wykłady dr inż. Olga Siedlecka-Lamch – Systemy baz danych z roku 2012