

Работа с составными типами данных



Рассматриваемые вопросы

- Создание пользовательских записей PL/SQL
- Создание записи с атрибутом %ROWTYPE
- Создание INDEX BY таблицы
- Создание INDEX BY таблицы с записями
- Различия между записями, таблицами и таблицами записей

Составные типы данных

- В отличие от скалярных типов могут хранить множество значений
- Два вида составных типов данных:
 - Записи PL/SQL
 - Коллекции PL/SQL
 - INDEX BY таблица или матрица ассоциативных элементов
 - Вложенная таблица
 - VARRAY

Составные типы данных

- Используйте записи PL/SQL, если необходимо в данный момент времени хранить только один экземпляр данных различного типа.
- Используйте коллекции PL/SQL, если необходимо хранить несколько значений одного и того же типа данных.

Записи PL/SQL

- Должны содержать один или более компонентов, называемых полями, скалярного типа, типа `RECORD` или `INDEX BY` таблица.
- По структуре сходны с записями в языках третьего поколения (3GL), включая `C` и `C++` .
- Определяются пользователем и могут отображать подмножество полей строки таблицы.
- Совокупность полей рассматривается как логическая единица.
- Удобны для выборки строки данных из таблицы при обработке.

Создание записи PL/SQL

Синтаксис:

1

```
TYPE имя_типа IS RECORD  
    (объявление_поля[, объявление_поля]...);
```

2

```
идентификатор    имя_типа;
```

объявление_поля:

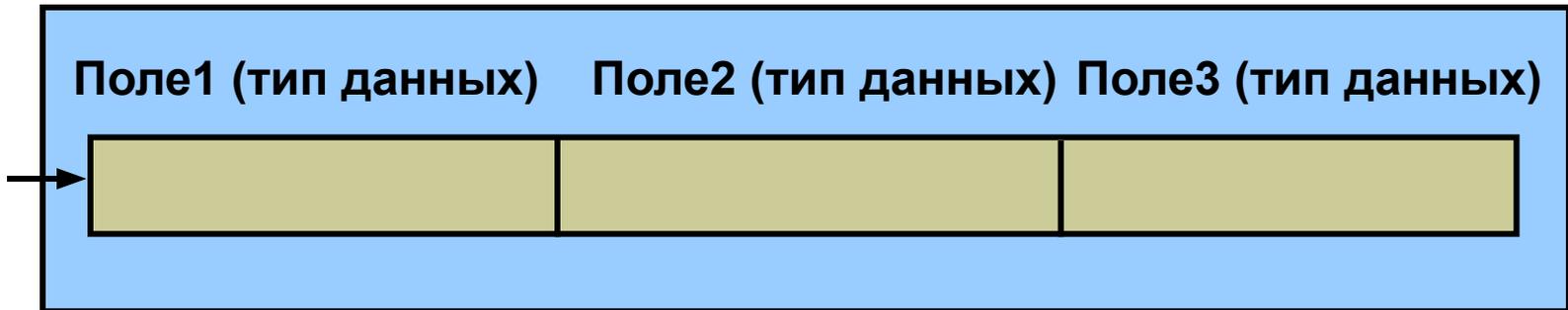
```
имя_поля {тип_поля | переменная%TYPE  
          | таблица.столбец%TYPE | таблица%ROWTYPE}  
[[NOT NULL] {:= | DEFAULT} выражение]
```

Создание записи PL/SQL

Пример объявления переменной для хранения имени, должности и оклада:

```
...  
  TYPE emp_record_type IS RECORD  
    (last_name  VARCHAR2(25),  
     job_id     VARCHAR2(10),  
     salary     NUMBER(8,2));  
  emp_record   emp_record_type;  
...
```

Структура записи PL/SQL



Пример:



Атрибут %ROWTYPE

- Используется для объявления переменной на основе совокупности столбцов в таблице или представлении базы данных.
- Перед %ROWTYPE указывается имя таблицы или представления.
- Поля записи наследуют имена и типы данных от столбцов таблицы или представления.

Синтаксис:

```
DECLARE  
    идентификатор    ссылка%ROWTYPE ;
```

Преимущества использования атрибута `%ROWTYPE`

- Количество базовых столбцов и типы данных в них могут быть неизвестны.
- Количество и типы данных базовых столбцов могут меняться во время выполнения.
- Полезен при выборке строки с помощью команды `SELECT *`

Атрибут %ROWTYPE

```
...  
DEFINE employee_number = 124  
DECLARE  
    emp_rec    employees%ROWTYPE;  
BEGIN  
    SELECT * INTO emp_rec FROM employees  
    WHERE employee_id = &employee_number;  
    INSERT INTO retired_emps(empno, ename, job, mgr,  
    hiredate, leavedate, sal, comm, deptno)  
    VALUES (emp_rec.employee_id, emp_rec.last_name,  
    emp_rec.job_id,emp_rec.manager_id,  
    emp_rec.hire_date, SYSDATE, emp_rec.salary,  
    emp_rec.commission_pct, emp_rec.department_id);  
END;  
/
```

Вставка строки с атрибутом %ROWTYPE

```
...  
DEFINE employee_number = 124  
DECLARE  
    emp_rec    retired_emps%ROWTYPE;  
BEGIN  
    SELECT employee_id, last_name, job_id, manager_id,  
           hire_date, hire_date, salary, commission_pct,  
           department_id INTO emp_rec FROM employees  
    WHERE  employee_id = &employee_number;  
    INSERT INTO retired_emps VALUES emp_rec;  
END;  
/  
SELECT * FROM retired_emps;
```

Обновление строки таблицы с помощью записи

```
SET SERVEROUTPUT ON
SET VERIFY OFF
DEFINE employee_number = 124
DECLARE
    emp_rec retired_emps%ROWTYPE;
BEGIN
    SELECT * INTO emp_rec FROM retired_emps
        WHERE empno=&employee_number;
    emp_rec.leavedate:=SYSDATE;
    UPDATE retired_emps SET ROW = emp_rec
        WHERE empno=&employee_number;
END;
/
SELECT * FROM retired_emps;
```

INDEX BY таблицы или матрицы ассоциативных элементов

- Структуры PL/SQL, содержащие два столбца:
 - Главный ключ целого или строкового типа
 - Столбец скалярного типа или типа `record`
- Без ограничений на размер; однако максимально возможный размер зависит от типа данных столбца с главным ключом.

Создание INDEX BY таблицы

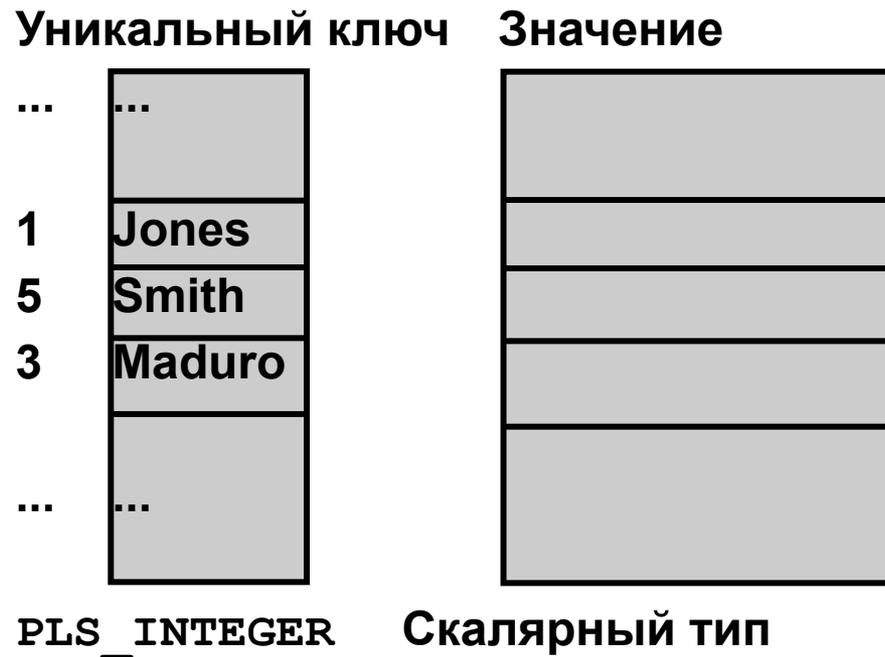
Синтаксис:

```
TYPE имя_типа IS TABLE OF
  {тип_столбца | переменная%TYPE
  | таблица.столбец%TYPE} [NOT NULL]
  | таблица.%ROWTYPE
  [INDEX BY PLS_INTEGER | BINARY_INTEGER
  | VARCHAR2 (<размер>)] ;
идентификатор  имя_типа ;
```

Объявление INDEX BY таблицы для хранения фамилий сотрудников.

```
...
TYPE ename_table_type IS TABLE OF
  employees.last_name%TYPE
  INDEX BY PLS_INTEGER;
...
ename_table ename_table_type;
```

Структура INDEX ВУ таблицы



Пример создание INDEX BY таблицы

```
DECLARE
  TYPE ename_table_type IS TABLE OF
    employees.last_name%TYPE
    INDEX BY PLS_INTEGER;
  TYPE hiredate_table_type IS TABLE OF DATE
    INDEX BY PLS_INTEGER;
  ename_table          ename_table_type;
  hiredate_table       hiredate_table_type;
BEGIN
  ename_table(1)       := 'CAMERON';
  hiredate_table(8)    := SYSDATE + 7;
  IF ename_table.EXISTS(1) THEN
    INSERT INTO ...
    ...
END;
/
```

Использование методов INDEX BY таблиц

Для упрощения использования INDEX BY таблиц в PL/SQL имеются следующие методы:

- EXISTS
- COUNT
- FIRST **and** LAST
- PRIOR
- NEXT
- DELETE

INDEX BY таблица записей

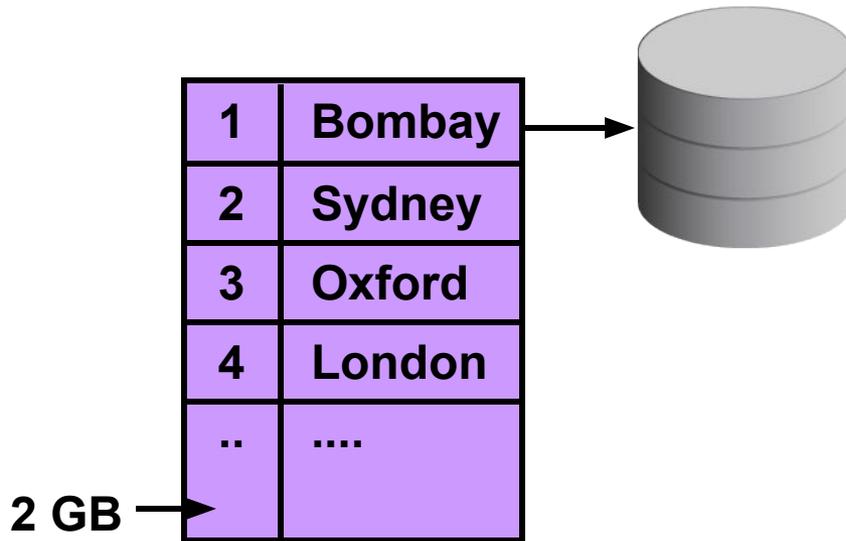
Пример определения переменной с типом INDEX BY таблицы для хранения целиком всей строки таблицы БД.

```
DECLARE
  TYPE dept_table_type IS TABLE OF
    departments%ROWTYPE
      INDEX BY PLS_INTEGER;
  dept_table dept_table_type;
  -- Каждый элемент dept_table - запись
```

Пример INDEX BY таблицы записей

```
SET SERVEROUTPUT ON
DECLARE
  TYPE emp_table_type IS TABLE OF
    employees%ROWTYPE INDEX BY PLS_INTEGER;
  my_emp_table  emp_table_type;
  max_count     NUMBER(3) := 104;
BEGIN
  FOR i IN 100..max_count
  LOOP
    SELECT * INTO my_emp_table(i) FROM employees
    WHERE employee_id = i;
  END LOOP;
  FOR i IN my_emp_table.FIRST..my_emp_table.LAST
  LOOP
    DBMS_OUTPUT.PUT_LINE(my_emp_table(i).last_name);
  END LOOP;
END;
/
```

Вложенные таблицы (Nested Tables)



```
Syntax:  
TYPE type_name IS TABLE OF  
{column_type | variable%TYPE  
| table.column%TYPE} [NOT NULL]  
| table.%ROWTYPE
```

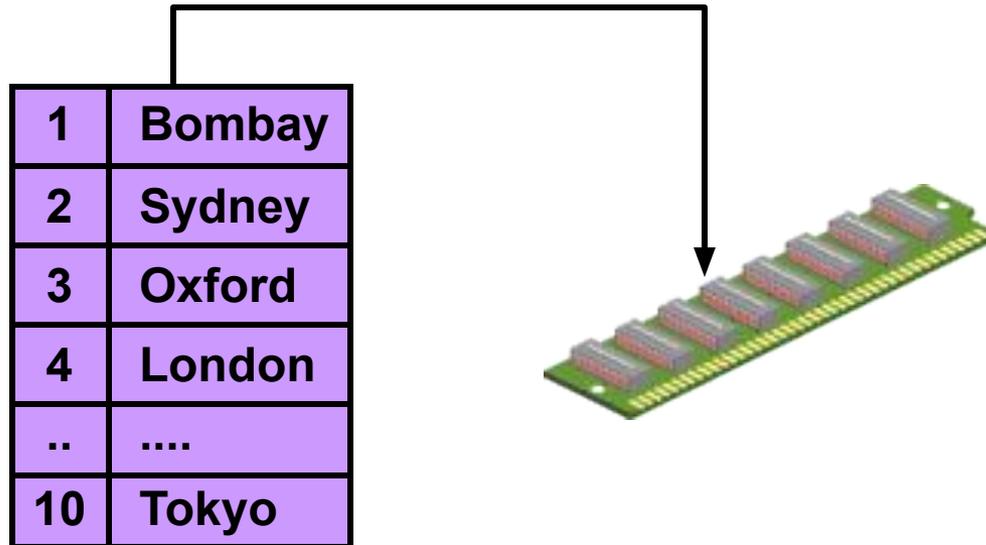
```
Example:  
TYPE location_type IS TABLE OF locations.city%TYPE;  
offices location_type;
```

Nested Tables: пример

```
DECLARE
  TYPE location_type IS TABLE OF locations.city%TYPE;
  offices location_type;
  table_count NUMBER;
BEGIN
  offices := location_type('Bombay', 'Tokyo','Singapore',
    'Oxford');
  offices.extend;
  offices(5) := 'London';
  FOR i in 1.. offices.count() LOOP
    DBMS_OUTPUT.PUT_LINE(offices(i));
  END LOOP;
END;
/
```

VARRAY

имеет ограниченный размер



Example:

```
TYPE location_type IS VARRAY(3) OF locations.city%TYPE;  
offices location_type;
```

Итоги

- PL/SQL позволяет объявлять и использовать переменные следующих составных типов данных:
 - Записи PL/SQL
 - INDEX BY таблицы
 - INDEX BY таблицы записей
- Объявление записи PL/SQL с помощью атрибута %ROWTYPE

Обзор практического занятия 6

- Объявление INDEX BY таблиц
- Обработка данных с помощью INDEX BY таблиц
- Объявление записи PL/SQL
- Обработка данных с помощью записей PL/SQL