User-Defined Data types

- Jayendra Khatod

Objectives

- **Create user-defined PL/SQL records**
- **Create a record with the %ROWTYPE attribute**
- **Practical exercises to work with Record Types.**

PL/SQL Records

- Must contain one or more components of any scalar or RECORD type called fields
- Are similar in structure in C
- Are not the same as rows in a database table
- Treat a collection of fields as a logical unit
- Are convenient for fetching a row of data from a table for processing

Creating a PL/SQL Record

- **Syntax**

```
TYPE type_name IS RECORD
     (field_declaration[,field_declaration]…);
identifiertype_name;
```

- **Where *field_declaration* is**

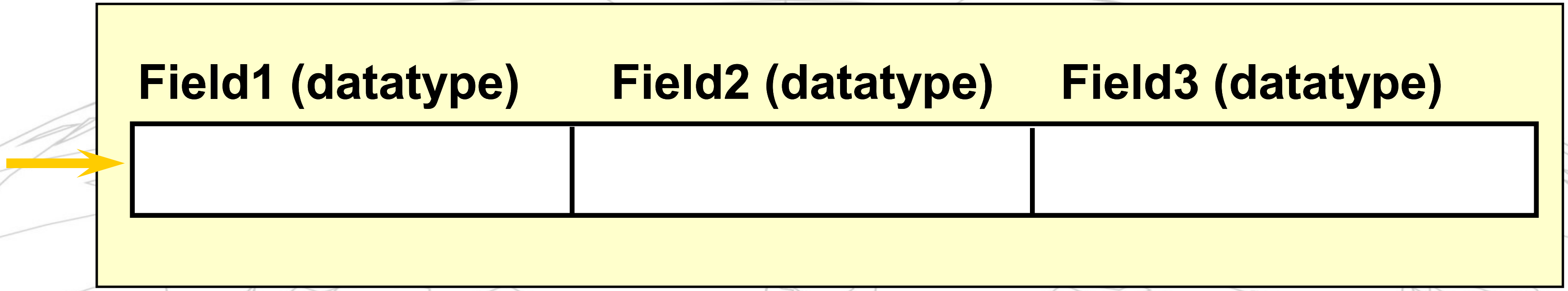
```
field_name {field_type | variable%TYPE
           | table.column%TYPE |
table%ROWTYPE}
           [[NOT NULL] {:= | DEFAULT} expr]
```

Creating a PL/SQL Record

- **Declare variables to store the name, job, and salary of a new employee.**
- **Example**

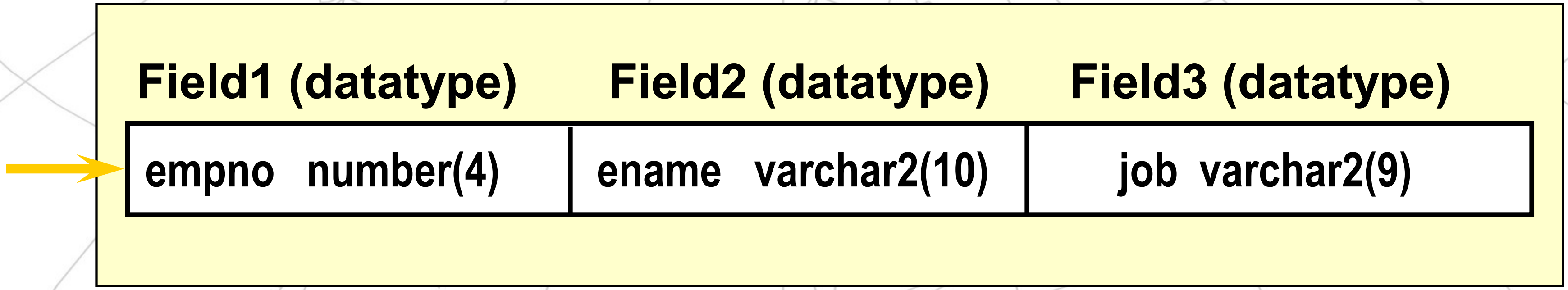
```
...
  TYPE emp_record_type IS RECORD
     (ename   VARCHAR2(10),
      job        VARCHAR2(9),
      sal        NUMBER(7,2));
  emp_record   emp_record_type;
...
```

PL/SQL Record Structure



Field1 (datatype)	Field2 (datatype)	Field3 (datatype)

Example



Field1 (datatype)	Field2 (datatype)	Field3 (datatype)
empno number(4)	ename varchar2(10)	job varchar2(9)

The %ROWTYPE Attribute

- **Declare a variable according to a collection of columns in a database table or view.**
- **Prefix %ROWTYPE with the database table.**
- **Fields in the record take their names and datatypes from the columns of the table or view.**

Declaring Records with the %ROWTYPE Attribute

- **Syntax**

```
DECLARE
   identifier   reference%ROWTYPE;
```

- ***identifier*** **is the name chosen for the record as a whole**
- ***reference*** **is the name of the table, view, cursor on which therecord is to be based**

Advantages of Using %ROWTYPE

- The number and datatypes of the underlying database columns may not be known.
- The number and datatypes of the underlying database column may change at runtime.
- The attribute is useful when retrieving a row with the SELECT statement.

The %ROWTYPE Attribute

- **Examples**
- **Declare a variable to store the same information about a department as it is stored in the DEPT table.**

```
dept_record    dept%ROWTYPE;
```

- **Declare a variable to store the same information about an employee as it is stored in the EMP table.**

```
emp_record     emp%ROWTYPE;
```

Example

```
DECLARE
        Emp_rec employee%ROWTYPE;
BEGIN
    Select  * into emp_rec from employees
    Where employee_id = 101;

    Insert into retired_emp (emp_id, salary,
    ……)
    Values  (emp_rec.employee_id,
    emp_rec.salary,…..);
    Commit;
END;
```

Summary

- Define and reference PL/SQL variables of composite data types:
  - PL/SQL records
- Define a PL/SQL record by using the %ROWTYPE attribute.