

# ☐ / ☐ GRADUAÇÃO



# Tecnologia em Análise e Desenvolvimento de Sistemas

Prof<sup>o</sup> Ms. Alexandre Barcelos profalexandre.barcelos@fiap.com.br

2019



# Database Application Development

Prof⁰ Ms. Alexandre Barcelos profalexandre.barcelos@fiap.com.br

2019



# **Objetivos**



Depois de completar esta lição, você poderá fazer o seguinte:

- Criar registros PL/SQL definidos pelo usuário
- Criar um registro com o atributo %ROWTYPE
- Criar uma tabela PL/SQL

# **Tipos de Dados Compostos**



- Tipos:
  - PL/SQL RECORDS
  - PL/SQL TABLES

1-6

#### **RECORDS e TABLES**

Como variáveis escalares, as variáveis compostas também possuem um tipo de dados. Os tipos de dados compostos (também chamados *conjuntos*) são RECORD, TABLE, Nested TABLE, e VARRAY. Utilize o tipo de dados RECORD para manipular os dados relacionados, porém diferentes, como uma unidade lógica. Utilize o tipo de dados TABLE para fazer referência e manipular conjuntos de dados como um objeto inteiro. Os tipos de dados Nested TABLE e VARRAY não são abordados neste curso.

Um registro é um grupo de itens de dados relacionados armazenados em campos, cada um com seu próprio nome e tipo de dados. Uma tabela contém uma coluna e uma chave primária para fornecer a você acesso a linhas semelhante a array. Uma vez definidos, as tabelas e registros podem ser reutilizados.

# Registros PL/SQL



- Devem conter um ou mais componentes de qualquer tipo de dados escalar, RECORD ou PL/SQL TABLE, chamados campos
- São semelhantes em estrutura a registros em um 3GL
- Não são iguais a linhas em uma tabela de banco de dados
- Tratam um conjunto de campos como uma unidade lógica
- São convenientes para extrair uma linha de dados de uma tabela para processamento

1-7

Um registro é um grupo de itens de dados relacionados armazenados em campos, cada um com seu próprio nome e tipo de dados. Por exemplo, suponha que você tenha tipos diferentes de dados sobre um funcionário, como nome, salário, data de admissão etc. Esses dados são diferentes em tipo porém estão relacionados logicamente. Um registro que contém campos, como o nome, o salário e a data de admissão de um funcionário permite manipular os dados como uma unidade lógica. Quando você declara um tipo de registro para esses campos, eles podem ser manipulados como uma unidade.

- Cada registro definido pode ter tantos campos quantos forem necessários.
- Os registros podem receber atribuição de valores iniciais e podem ser definidos como NOT NULL.
- Os campos sem valores iniciais são inicializados para NULL.
- A palavra-chave DEFAULT também pode ser usada ao definir campos.
- Você pode definir tipos RECORD e declarar registros definidos pelo usuário na parte declarativa de qualquer bloco, subprograma ou pacote.

<ul> <li>Pode-se declarar e fazer referência a registros aninhados. Um registro pod ser o componente de outro registro.</li> </ul>	le

# Criando um Registro PL/SQL



#### Sintaxe:

```
TYPE type_name IS RECORD (field_declaration]...);
```

```
2 identifier type_name;
```

#### field declaration:

1-8

### Definindo e Declarando um Registro PL/SQL

Para criar um registro, defina um tipo RECORD e, em seguida, declare registros desse tipo.

Na sintaxe:

*type\_name* é o nome do tipo RECORD. (Esse identificador é usado para declarar registros.)

field\_name é o nome de um campo dentro do registro

field\_type é o tipo de dados do campo. (Ele representa qualquer tipo de dados PL/SQL exceto REF CURSOR. Você pode usar os atributos %TYPE e ROWTYPE.)

expr é o field type ou um valor inicial

A restrição NOT NULL impede a atribuição de nulos a esses campos. Certifique-se de inicializar campos NOT NULL.

# Criando um Registro PL/SQL



Declarar variáveis para armazenar o nome, a tarefa e o salário de um novo funcionário.

### Exemplo

```
TYPE emp_record_type IS RECORD

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

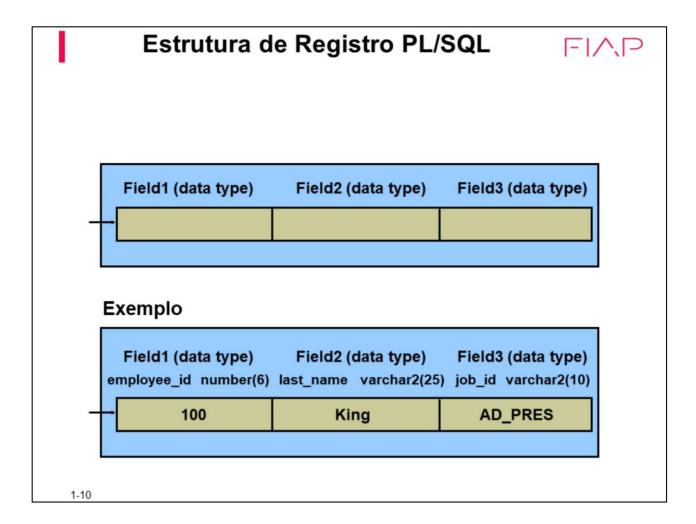
1-9

### Criando um Registro PL/SQL

Declarações de campo são semelhantes a declarações de variável. Cada campo tem um nome exclusivo e um tipo de dados específico. Não existem tipos de dados predefinidos para registros PL/SQL, como existem para variáveis escalares. Assim, você deve primeiro criar o tipo de dados e, em seguida, declarar um identificador usando esse tipo de dados.

O exemplo a seguir mostra que você pode usar o atributo %TYPE para especificar um tipo de dados de campo:

**Observação:** Você pode adicionar a restrição NOT NULL a qualquer declaração de campo para impedir a atribuição de nulos a esse campo. Lembre-se, os campos declarados como NOT NULL devem ser inicializados.



### Fazendo Referência e Inicializando Registros

Os campos em um registro são acessados por nome. Para fazer referência ou inicializar um campo individual, use notação em pontos e a seguinte sintaxe:

Por exemplo, você faz referência ao campo no registro emp\_record .

É possível, em seguida, atribuir um valor ao campo de registro.

Em um bloco ou subprograma, os registros definidos pelo usuário são concretizados quando você inclui o bloco ou subprograma e deixam de existir quando você sai do bloco ou subprograma.

### **Atribuindo Valores aos Registros**

Você pode atribuir uma lista de valores comuns a um registro usando a instrução SELECT ou FETCH. Certifique-se de que os nomes de colunas aparecem na mesma ordem dos campos no registro. Você também pode atribuir um registro a outro se eles tiverem o mesmo tipo de dados.

Um registro definido pelo usuário e um registro %ROWTYPE nunca têm o mesmo tipo de dados.



- Declarar uma variável segundo um conjunto de colunas em uma view ou tabela de banco de dados.
- Prefixar %ROWTYPE com a tabela de banco de dados.
- Campos no registro obtêm seus nomes e Sintaxe:

#### DECLARE

identifier reference%ROWTYPE;

1-11

### **Declarando Registros com o Atributo %ROWTYPE**

Para declarar um registro com base em um conjunto de colunas em uma view ou tabela de banco de dados, use o atributo %ROWTYPE. Os campos no registro obtêm seus nomes e tipos de dados a partir das colunas da tabela ou view. O registro também pode armazenar uma linha inteira de dados extraída

de um cursor ou variável de cursor.

No exemplo a seguir, um registro é declarado usando %ROWTYPE como um especificador de tipo de dados.

O registro, *emp\_record*, terá uma estrutura consistindo nos campos a seguir, cada um deles representando uma coluna na tabela EMP.

Observação: Isso não é código, mas simplesmente a estrutura da variável composta.

#### O Atributo %ROWTYPE

• Declarar uma variável segundo um conjunto de colunas em uma view ou tabela de banco de dados.

- Prefixar %ROWTYPE com a tabela de banco de dados.
- Campos no registro obtêm seus nomes e tipos de dados a partir das colunas da tabela ou view.

# Vantagens de Usar %ROWTYPE



- O número e tipos de dados das colunas de banco de dados subjacentes podem não ser conhecidas.
- O número e tipos de dados da coluna de banco de dados subjacente pode alterar no tempo de execução.
- O atributo é útil ao recuperar uma linha com a instrução SELECT \*.

1-12

### Declarando Registros com o Atributo %ROWTYPE (continuação)

#### **Sintaxe**

**onde:** *identificador* é o nome escolhido para o registro como um todo *referência* é o nome da tabela, view, cursor ou variável de cursor em que o

registro deve ser baseado. (Você deve certificar-se de que essa referência é válida ao declarar o registro, isto é, a tabela ou view precisa existir.)

Pode-se, em seguida, atribuir um valor ao campo de registro.

### Vantagens de Usar %ROWTYPE

- O número e tipos de dados das colunas de banco de dados subjacentes podem nãoser conhecidas.
- O número e tipos de dados da coluna de banco de dados subjacente pode alterar no tempo de execução.
- O atributo é útil ao recuperar uma linha com a instrução SELECT.



## **Exemplos**

Declarar uma variável para armazenar a mesma informação sobre um departamento que está armazenada na tabela DEPT.

dept\_record dept%ROWTYPE;

Declarar uma variável para armazenar a mesma informação sobre um funcionário que está armazenada na tabela EMP.

emp record emp%ROWTYPE;

1-13

### **Exemplos**

A primeira declaração no slide cria um registro com os mesmos nomes de campo e tipos de dados de campo da linha na tabela DEPT. Os campos são DEPTNO, DNAME e LOCATION.

A segunda declaração cria um registro com os mesmos nomes de campo e tipos de dados de campo de uma linha na tabela EMP. Os campos são EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, COMM e DEPTNO.



## **Exemplos**

Declarar uma variável para armazenar a mesma informação sobre um departamento que está armazenada na tabela DEPT.

dept\_record dept%ROWTYPE;

Declarar uma variável para armazenar a mesma informação sobre um funcionário que está armazenada na tabela EMP.

emp record emp%ROWTYPE;

1-14

No exemplo a seguir, um funcionário está se aposentando. As informações sobre esse funcionário são adicionadas a uma tabela que contém informações sobre funcionários aposentados. O usuário fornece o número do funcionário.

#### **DECLARE**

emp\_rec emp%ROWTYPE;

**BEGIN** 

SELECT \* INTO emp\_rec

FROM emp

WHERE empno = &employee\_number;

INSERT INTO retired\_emps(empno, ename, job, mgr, hiredate,

leavedate, sal, comm, deptno)

VALUES (emp\_rec.empno, emp\_rec.ename, emp\_rec.job, emp\_rec.mgr, emp\_rec.hiredate, SYSDATE, emp\_rec.sal, emp\_rec.comm,

emp\_rec.deptno); COMMIT;

**END** 



```
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.job_id,emp_rec.manager_id, emp_rec.hire_date, SYSDATE, emp_rec.salary, emp_rec.commission_pct, emp_rec.department_id);

END;

/
```

1-15

# Inserindo um registro com %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;
```

# Atualizando um linha em uma Tabela 🕞 🦯 🏳 utilizando um Registro



```
SET SERVEROUTPUT ON
SET VERIFY OFF
DEFINE employee number = 124
DECLARE
   emp rec retired emps%ROWTYPE;
 SELECT * INTO emp rec FROM retired emps;
emp rec.leavedate:=SYSDATE;
UPDATE retired emps SET ROW = emp rec WHERE
 empno=&employee number;
END;
SELECT * FROM retired emps;
```

1-17

#### **Exercícios**

# Escreva um bloco PL/SQL para imprimir informações sobre um determinado país.. Declare um registro PL/SQL baseado na estrutura da tabela de countries.

Use o comando DEFINE para definir uma variável c **countryid**. Atribua CA a countryid. Passe o valor para o bloco PL/SQL através de uma variável de substituição.

Na seção declare, use o atributo%ROWTYPE e declare a variável country\_record de tipo countries.

Na seção executável, obter todas as informações da tabela **countries** usando countryid. Mostra as informações selecionadas sobre o país.