

FIAP GRADUAÇÃO

Tecnologia em Análise e Desenvolvimento de Sistemas

Application Development For Databases

Prof. Alan Reis

Interagindo com o Oracle Database Server: Instruções SQL em Programas PL/SQL

Objetivos

Ao concluir esta lição, você será capaz de:

- Determinar quais instruções SQL podem ser incluídas diretamente em um bloco executável PL/SQL
- Manipular dados com instruções DML em blocos PL/SQL
- Usar instruções de controle de transação em blocos PL/SQL
- Usar a cláusula `INTO` para armazenar os valores retornados por uma instrução SQL
- Fazer a distinção entre cursores explícitos e implícitos
- Usar atributos de cursores SQL

Agenda

- Recuperando dados com PL/SQL
- Manipulando dados com PL/SQL
- Apresentando cursores SQL

Instruções SQL em Blocos PL/SQL

- Recupere uma linha do banco de dados usando o comando `SELECT`.
- Faça alterações em linhas do banco de dados usando comandos DML.
- Controle uma transação com o comando `COMMIT`, `ROLLBACK` ou `SAVEPOINT`.

Instruções **SELECT** em Blocos PL/SQL

Recupere dados do banco de dados com uma instrução `SELECT`.

Sintaxe:

```
SELECT  select_list
INTO    {variable_name[, variable_name]...
        | record_name}
FROM    table
[WHERE  condition];
```

Instruções SELECT em Blocos PL/SQL

- A cláusula INTO é obrigatória.
- As consultas devem retornar apenas uma linha.

```
DECLARE
  v_fname VARCHAR2(25);
BEGIN
  SELECT first_name INTO v_fname
  FROM employees WHERE employee_id=200;
  DBMS_OUTPUT.PUT_LINE(' First Name is : '||v_fname);
END;
/
```

```
anonymous block completed
First Name is : Jennifer
```


Recuperando Dados em Blocos PL/SQL: Exemplo

Recupere `hire_date` e `salary` para o funcionário especificado.

```
DECLARE
  v_emp_hiredate    employees.hire_date%TYPE;
  v_emp_salary      employees.salary%TYPE;
BEGIN
  SELECT    hire_date, salary
  INTO      v_emp_hiredate, v_emp_salary
  FROM employees
  WHERE employee_id = 100;
  DBMS_OUTPUT.PUT_LINE ('Hire date is :'|| v_emp_hiredate);
  DBMS_OUTPUT.PUT_LINE ('Salary is :'|| v_emp_salary);
END;
/
```

```
anonymous block completed
Hire date is : 17-JUN-87
Salary  is : 24000
```

Recuperando Dados em Blocos PL/SQL FIAP

Retorne a soma dos salários de todos os funcionários do departamento especificado.

Exemplo:

```
DECLARE
    v_sum_sal    NUMBER(10,2);
    v_deptno     NUMBER NOT NULL := 60;
BEGIN
    SELECT SUM(salary) -- group function
    INTO v_sum_sal FROM employees
    WHERE department_id = v_deptno;
    DBMS_OUTPUT.PUT_LINE ('The sum of salary is ' || v_sum_sal);
END;
```

```
anonymous block completed
The sum of salary is 28800
```

Ambiguidades de Nomeação

```
DECLARE
  hire_date      employees.hire_date%TYPE;
  sysdate        hire_date%TYPE;
  employee_id    employees.employee_id%TYPE := 176;
BEGIN
  SELECT         hire_date, sysdate
  INTO           hire_date, sysdate
  FROM           employees
  WHERE          employee_id = employee_id;
END;
/
```

```
Error report:
ORA-01422: exact fetch returns more than requested number of rows
ORA-06512: at line 6
01422. 00000 - "exact fetch returns more than requested number of rows"
*Cause:      The number specified in exact fetch is less than the rows returned.
*Action:     Rewrite the query or change number of rows requested
```

Convenções de Nomeação

- Use uma convenção de nomeação para evitar ambiguidade na cláusula `WHERE`.
- Evite usar nomes de colunas do banco de dados como identificadores.
- Podem ocorrer erros de sintaxe porque o código PL/SQL verifica primeiro se há uma coluna na tabela do banco de dados.
- Os nomes de variáveis locais e parâmetros formais têm precedência sobre os nomes de *tabelas* do banco de dados.
- Os nomes de *colunas* de tabelas do banco de dados têm precedência sobre os nomes de variáveis locais.

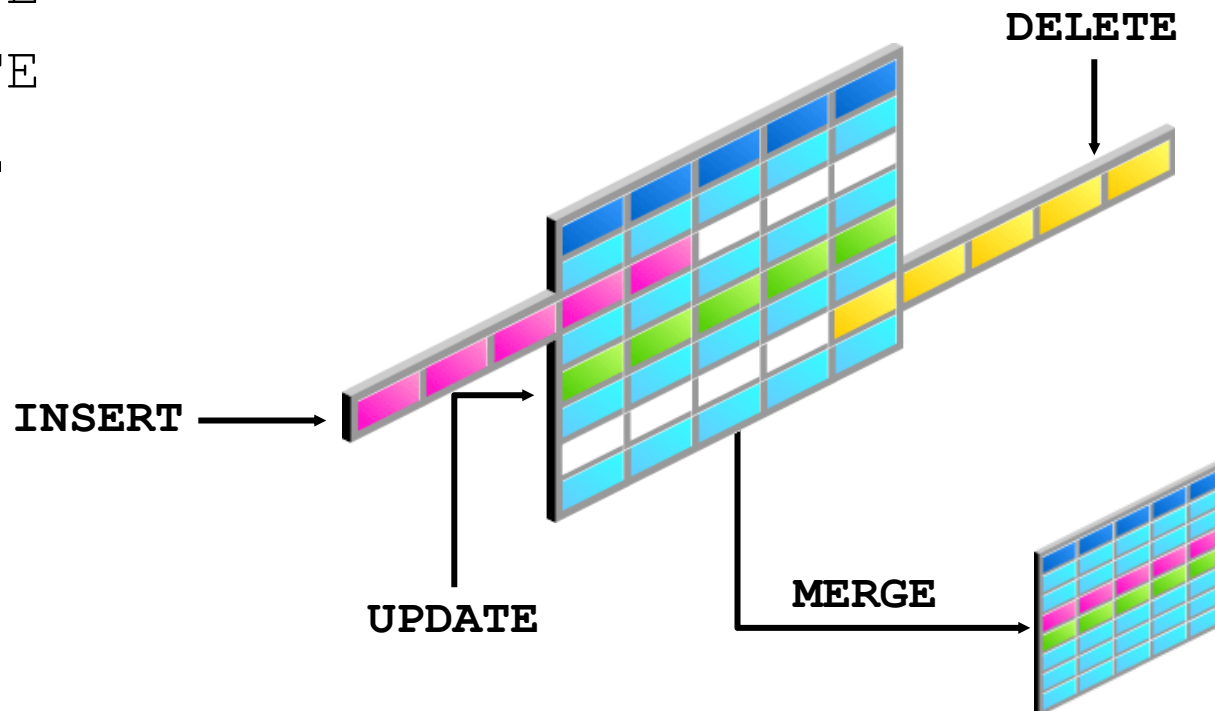
Agenda

- Recuperando dados com PL/SQL
- Manipulando dados com PL/SQL
- Apresentando cursores SQL

Usando o Código PL/SQL para Manipular Dados

Faça alterações em tabelas do banco de dados usando comandos DML:

- INSERT
- UPDATE
- DELETE
- MERGE



Inserindo Dados: Exemplo

Adicione informações sobre um novo funcionário à tabela EMPLOYEES.

```
BEGIN
  INSERT INTO employees
    (employee_id, first_name, last_name, email,
     hire_date, job_id, salary)
    VALUES (employees_seq.NEXTVAL, 'Ruth', 'Cores',
            'RCORES', CURRENT_DATE, 'AD_ASST', 4000);
END;
/
```

Atualizando Dados: Exemplo

Aumente o salário de todos os funcionários que trabalham no estoque.

```
DECLARE
    sal_increase    employees.salary%TYPE := 800;
BEGIN
    UPDATE          employees
    SET              salary = salary + sal_increase
    WHERE            job_id = 'ST_CLERK';
END;
/
```

```
anonymous block completed
FIRST_NAME      SALARY
-----
Julia           4000
Irene           3500
James           3200
Steven          3000
```

...

```
Curtis          3900
Randall         3400
Peter           3300
```

```
20 rows selected
```


Deletando Dados: Exemplo

Delete linhas que pertencem ao departamento 10 da tabela employees.

```
DECLARE
    deptno    employees.department_id%TYPE := 10;
BEGIN
    DELETE FROM employees
    WHERE    department_id = deptno;
END;
/
```

Intercalando Linhas

Insira ou atualize as linhas na tabela `COPY_EMP3` para que corresponda à tabela `employees`.

```
BEGIN
MERGE INTO copy_emp c
  USING employees e
  ON (e.employee_id = c.empno)
  WHEN MATCHED THEN
    UPDATE SET
      c.first_name = e.first_name,
      c.last_name = e.last_name,
      c.email = e.email,
      . . .
  WHEN NOT MATCHED THEN
    INSERT VALUES(e.employee_id, e.first_name, e.last_name,
      . . . ,e.department_id);
END;
/
```

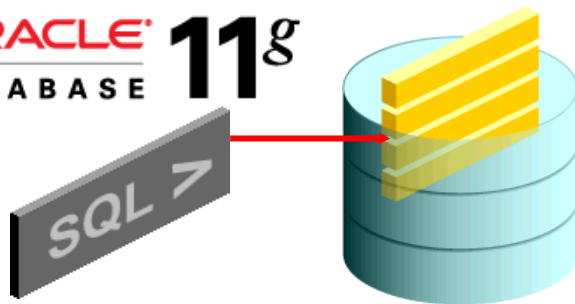
Agenda

- Recuperando dados com PL/SQL
- Manipulando dados com PL/SQL
- Apresentando cursores SQL

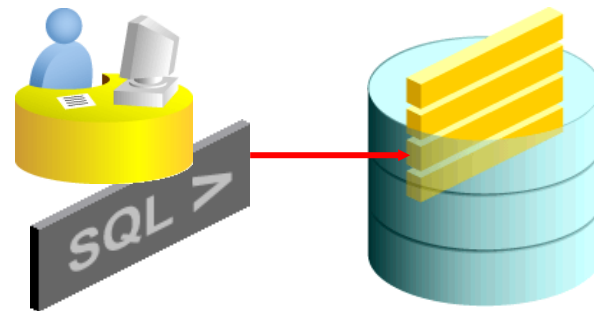
Cursor SQL

- Um cursor é um ponteiro para a área de memória particular alocada pelo Oracle Server. Ele é usado para tratar o conjunto de resultados de uma instrução `SELECT`.
- Há dois tipos de cursores: implícitos e explícitos.
 - **Implícitos:** Criados e gerenciados internamente pelo Oracle Server para processar instruções SQL
 - **Explícitos:** Declarados explicitamente pelo programador

ORACLE[®] 11^g
DATABASE



Cursor implícito



Cursor explícito

Atributos de Cursores SQL para Cursores Implícitos

Usando os atributos de cursores SQL, você pode testar o resultado das instruções SQL.

SQL%FOUND	Atributo booleano que será avaliado como <code>TRUE</code> se a instrução SQL mais recente afetar pelo menos uma linha
SQL%NOTFOUND	Atributo booleano que será avaliado como <code>TRUE</code> se a instrução SQL mais recente não afetar nenhuma linha
SQL%ROWCOUNT	Um valor inteiro que representa o número de linhas afetadas pela instrução SQL mais recente

Atributos de Cursores SQL para Cursores Implícitos

Delete linhas que possuem o ID de funcionário especificado da tabela `employees`. Imprima o número de linhas deletadas.

Exemplo:

```
DECLARE
  v_rows_deleted VARCHAR2(30)
  v_empno employees.employee_id%TYPE := 176;
BEGIN
  DELETE FROM employees
  WHERE employee_id= v_empno;
  v_rows_deleted := (SQL%ROWCOUNT ||
                    ' row deleted. ');
  DBMS_OUTPUT.PUT_LINE (v_rows_deleted);

END;
```

Questionário

Quando você usa a instrução `SELECT` em blocos PL/SQL, a cláusula `INTO` é necessária e as consultas podem retornar uma ou mais linhas.

- a. Verdadeiro
- b. Falso

Sumário

Nesta lição, você aprendeu a:

- Incorporar instruções DML, instruções de controle de transações e instruções DDL aos blocos PL/SQL
- Usar a cláusula `INTO`, obrigatória para todas as instruções `SELECT`, em blocos PL/SQL
- Fazer a distinção entre cursores explícitos e implícitos
- Usar atributos de cursores SQL para determinar o resultado de instruções SQL

Exercício 4: Visão Geral

Este exercício aborda os seguintes tópicos:

- Selecionando dados de uma tabela
- Inserindo linhas em uma tabela
- Atualizando linhas de uma tabela
- Deletando um registro de uma tabela