
ORACLE
Academy

Database Programming with PL/SQL

1-1

Introduction to PL/SQL

ORACLE Academy

Objectives

- This lesson covers the following objectives:
  - Describe PL/SQL
  - Differentiate between SQL and PL/SQL
  - Explain the need for PL/SQL

Purpose

- PL/SQL is Oracle Corporation's procedural programming language for relational databases
- To describe PL/SQL, you learn its characteristics and identify the differences between PL/SQL and SQL
- Identifying limitations of SQL and how PL/SQL addresses those limitations will help you to understand why PL/SQL is needed

PL/SQL Description

- Procedural Language extension to SQL
- A third-generation programming language (3GL)
- An Oracle proprietary programming language
- Combines program logic and control flow with SQL

Evolution/Generations of Programming Languages:

- 1GL: First-generation programming languages; these are machine level languages specific to a particular CPU
- 2GL: Second-generation programming languages; assembly languages specific to a particular CPU; converted by an assembler into a machine language; commonly used for performance-oriented and processing-intensive applications such as firmware interfaces and hardware drivers
- 3GL: Third-generation programming languages; converted into machine language by a compiler; less cryptic and thus more programmer-friendly than 2GLs (ex., Visual Basic, C, C++, COBOL, FORTRAN, Java, Pascal, PL/SQL)

Evolution/Generations of Programming Languages:

- 4GL: Fourth-generation programming languages; less cryptic and thus more programmer-friendly than 2GLs; unlike the broad applicability of 3GLs, most 4GLs are used with databases, for queries, report generation, data manipulation, etc. (ex., SQL, MySQL)
- 5GL: Fifth-generation programming languages; generally involve a visual or graphical development environment that exports source language to a 3GL or 4GL compiler; 5GL may also refer to languages that define a problem and a set of constraints, then let the computer find a solution; (ex., AI systems such as PROLOG used with IBM's Watson)
- As an Oracle proprietary programming language, PL/SQL is only used with an Oracle database

SQL Description

- A nonprocedural language
- Also known as a "declarative language," allows programmer to focus on input and output rather than the program steps
- A fourth-generation programming language (4GL)
- Primary language used to access and modify data in relational databases
- Standardized by the American National Standards Institute (ANSI)
- Vendors such as Oracle typically include some proprietary SQL features in their database environments

Although PL/SQL was developed after SQL, SQL is a 4GL while PL/SQL is a 3GL.

SQL and PL/SQL are both needed. They are not alternatives to each other. Only SQL can be used to access database tables and only PL/SQL can be used to write the procedural logic.

SQL Statement

- The SQL statement shown is simple and straightforward
- However, if you need to modify a data item in a conditional manner, you come across a limitation of SQL

```
SELECT employee_id, job_id, hire_date
     FROM employees;
```

- For example, how would you write an SQL statement to update the job_id data with a new value determined by the current job_id and the hire_date?

Limitations of SQL

- Assume the company decides to promote all sales representatives, marketing representatives, and stock clerks employed for at least ten years to senior representatives and clerks
- If the current date is 05-Feb-2015, sales reps 174, 176, and 178 qualify for the promotion

EMPLOYEE_ID	JOB_ID	HIRE_DATE	"NEW" JOB_ID
174	SA_REP	11-May-1996	SR_SA_REP
176	SA_REP	24-Mar-1998	SR_SA_REP
178	SA_REP	24-May-1999	SR_SA_REP
240	SA_REP	02-Oct-2005	
242	SA_REP	09-Dec-2007	

Limitations of SQL

- If the current date is 05-FEB-2015, stock clerks 141, 142, 143, and 144 also qualify for the promotion

EMPLOYEE_ID	JOB_ID	HIRE_DATE	"NEW" JOB_ID
141	ST_CLERK	17-Oct-1995	SR_ST_CLERK
142	ST_CLERK	29-Jan-1997	SR_ST_CLERK
143	ST_CLERK	15-Mar-1998	SR_ST_CLERK
144	ST_CLERK	09-Jul-1998	SR_ST_CLERK
244	ST_CLERK	07-Sep-2009	

Limitations of SQL

- One solution to updating the job_id data is shown
- How many SQL statements do you need to write for sales representatives, marketing representatives, and stock clerks?
- What if there are other job_ids to update?

```
UPDATE employees
     SET job_id = 'SR_SA_REP'
     WHERE job_id = 'SA_REP' AND
          hire_date <= '05-Feb-2005'
```

```
UPDATE employees
     SET job_id = 'SR_ST_CLERK'
     WHERE job_id = 'ST_CLERK' AND
          hire_date <= '05-Feb-2005'
```

Limitations of SQL

- You would need to write a separate SQL statement for each job_id that needs to be updated
- Depending on the number of job_ids, this could be a tedious task
- It would be easier to write a single statement to accomplish this task
- The statement would require logic, otherwise known as procedural logic
- PL/SQL extends SQL with procedural logic and makes it possible to write one statement to accomplish this task

PL/SQL Extends SQL with Procedural Logic

- Using PL/SQL, you can write one statement to promote the sales representatives, marketing representatives, and stock clerks.

```
DECLARE
   CURSOR c_employees IS SELECT * FROM employees;
BEGIN
   FOR c_emp in c_employees
   LOOP
     IF c_emp.job_id = 'SA_REP' AND c_emp.hire_date <= '05-Feb-2005' THEN
       UPDATE employees
       SET job_id = 'SR_SA_REP' WHERE employee_id = c_emp.employee_id;
       ELSIF c_emp.job_id = 'MK_REP' AND c_emp.hire_date <= '05-Feb-2005' THEN
         UPDATE employees
         SET job_id = 'SR_MK_REP' WHERE employee_id = c_emp.employee_id;
       ELSIF c_emp.job_id = 'ST_CLERK' AND c_emp.hire_date <= '05-Feb-2005' THEN
         UPDATE employees
         SET job_id = 'SR_ST_CLRK' WHERE employee_id = c_emp.employee_id;
     END IF;
   END LOOP;
END;
```

Procedural Constructs

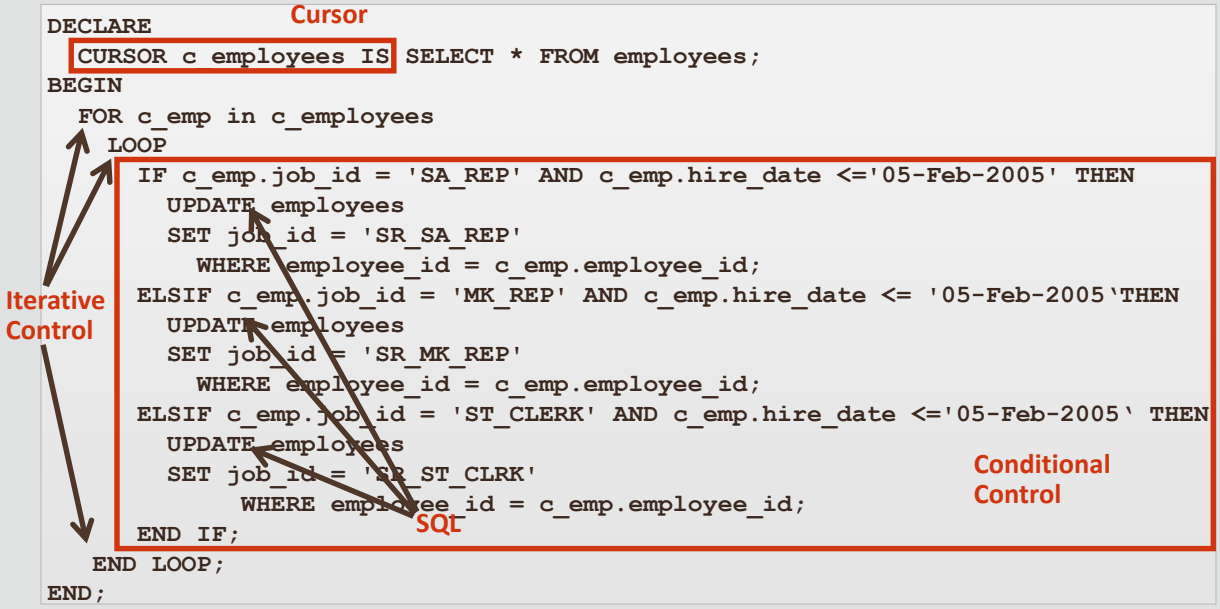
- You use PL/SQL to write the procedural code and embed SQL statements within the PL/SQL code
- The procedural code includes variables, constants, cursors, conditional logic, and iteration
- PL/SQL code blocks can be saved and named, then executed whenever needed

ORACLE Academy

As powerful as SQL is, it simply does not offer the flexibility and power developers need to create full-blown applications. Oracle's PL/SQL language ensures that we can stay entirely within the operating system-independent Oracle environment and still write highly efficient applications that meet our users' requirements.

Procedural Constructs Highlighted

- Several PL/SQL constructs are highlighted below




```
DECLARE
  CURSOR c_employees IS SELECT * FROM employees;
BEGIN
  FOR c_emp in c_employees
   LOOP
    IF c_emp.job_id = 'SA_REP' AND c_emp.hire_date <='05-Feb-2005' THEN
      UPDATE employees
      SET job_id = 'SR_SA_REP'
        WHERE employee_id = c_emp.employee_id;
    ELSIF c_emp.job_id = 'MK_REP' AND c_emp.hire_date <= '05-Feb-2005'THEN
      UPDATE employees
      SET job_id = 'SR_MK_REP'
        WHERE employee_id = c_emp.employee_id;
    ELSIF c_emp.job_id = 'ST_CLERK' AND c_emp.hire_date <='05-Feb-2005' THEN
      UPDATE employees
      SET job_id = 'SR_ST_CLRK'
          WHERE employee_id = c_emp.employee_id;
    END IF;
   END LOOP;
END;
```

Characteristics of PL/SQL

- PL/SQL:
  - Is a highly structured, readable, and accessible language
  - Is a standard and portable language for Oracle development
  - Is an embedded language and it works with SQL
  - Is a high-performance, highly integrated database language
  - Is based on the Ada Programming Language and has many similarities in syntax

Terminology

- Key terms used in this lesson included:
  - PL/SQL
  - Procedural Constructs

- PL/SQL is Oracle Corporation's procedural language for relational databases which allows program logic and control flow to be combined with SQL statements.
- The procedural constructs include variables, constants, cursors, conditional logic, and iteration.

Summary

- In this lesson, you should have learned how to:
  - Describe PL/SQL
  - Differentiate between SQL and PL/SQL
  - Explain the need for PL/SQL


ORACLE
Academy