Oracle Academy
Introduction to Database Programming with PL/SQL
Instructor Resource Guide

INSTRUCTOR NOTES FOR SLIDES

SECTION 1 LESSON 1 – Introduction to PL/SQL

Slide 1: Introduction to PL/SQL
No instructor notes for this slide

Slide 2: What Will I Learn?
No instructor notes for this slide

Slide 3: Why Learn It?
No instructor notes for this slide

Slide 4: Tell Me / Show Me – What is PL/SQL?
- PL/SQL is a proprietary language, that is, it is owned by Oracle and can be used only with an Oracle database or tool.
- PL/SQL is part of a class of programming languages called procedural languages. Other procedural languages include C, Perl, Java and Visual Basic.
- PL/SQL is a row-based language; it works on data one row at a time.

Evolution of Programming Languages
- **1GL:** Low level or machine language that a computer processor can interpret (for example, machine languages)
- **2GL:** Assembler or assembly language that is converted by an assembler into a machine language (for example, assembly languages)
- **3GL:** A "high-level" programming language that is converted into machine language by a compiler. 3GLs require extensive knowledge of programming language syntax (for example, Basic, C, Fortran, Pascal, Java, PL/SQL)
- **4GL:** A language that is closer to natural language than a programming language; most 4GLs are used to access databases (for example, SQL)
- **5GL:** Two possible definitions:
  - A language that uses a visual or graphical development interface to create source language that is usually compiled with a 3GL or 4GL language compiler (for example, Microsoft, Borland, IBM have visual languages for developing applications in Java)
  - A language used for artificial intelligence or neural networks

Oracle's PL/SQL language has several defining characteristics:
- It is a highly structured, readable, and accessible language.
- It is a standard and portable language for Oracle development.
- It is an embedded language, works with SQL.
- It is a high-performance, highly integrated database language.

Copyright © 2009, Oracle. All rights reserved.

- It is based on the ADA language and has many similarities in syntax

Slide 5: Tell Me / Show Me – What is SQL? Structured Query Language (SQL)

- Oracle generally pronounces SQL as “sequel”. Other vendors may pronounce SQL as separate letters, “S”, “Q”, “L”.
- For a long time, SQL was enough to satisfy developers. But as time passed, and databases matured, PL/SQL’s procedural code was needed. PL/SQL was introduced to overcome some limitations in SQL and to provide a more complete programming solution for those who sought to build procedural applications to run against the Oracle database.
- PL/SQL is a SET based language – works on sets of data
- Stress that we need both SQL and PL/SQL. They are not alternatives to each other. Only SQL can be used to access the table data, and only PL/SQL is used to write the procedural logic.

Slide 6: Tell Me / Show Me – SQL Statement

No instructor notes for this slide

Slide 7: Tell Me / Show Me – Limitations of SQL

For every student, check the class_id and the final_numeric_grade. Depending on the performance of all the students in the course, you may want to update the letter_grade column by assigning varying final letter grades to the students. For example:

- For class 1, assign a letter grade of
  - A to those students who receive a number grade of 66-75
  - B to those students who receive a number grade of 56-65
  - C to those students who receive a number grade of 46-55
  - D to those students who receive a number grade of 36-45
  - F to those students who receive a number grade less than or equal to 35
- For class 2, assign a letter grade of
  - A to those students who receive a number grade of of 91-100
  - B to those students who receive a number grade of 81-90
  - C to those students who receive a number grade of 71-80
  - D to those students who receive a number grade of 61-70
  - F to those students who receive a number grade less than or equal to 60

Using SQL, how would you do this?

The following SELECT statement is used to display the rows of data from the class_enrollments table. SQL statements are written within PL/SQL statements.

```
SELECT class_id, student_id, dance_number, dance_routine
FROM class_enrollments;
```

The SQL statement shown above is simple and straightforward. However, if you want to alter any of the retrieved data in a conditional manner, you come across the limitations of SQL.

Copyright © 2009, Oracle. All rights reserved.

How would you write a SQL statement to update the dance_routine with varying dance routines for students in different classes?

CLASS_ID	**STUDENT_ID**	**DANCE_GRADE**	**DANCE_ROUTINE**
1	101	75	
1	107	71	
1	131	65	
2	155	91	
2	114	93	

Depending on the performance of all the students in the course, you may want to update the dance_routine column by assigning varying final dance routines to the students. For example:
For class 1, assign a dance routine of
A to those students who receive a dance grade of 66 - 75
B to those students who receive a dance grade of 56 - 65
C to those students who receive a dance grade of 46 - 55
D to those students who receive a dance grade of 36 - 45
For class 2, assign a dance routine of
A to those students who receive a dance grade of 36 - 45
B to those students who receive a dance grade of 46 - 55
C to those students who receive a dance grade of 56 - 65
D to those students who receive a dance grade of 66 – 75
Using SQL, how would you accomplish this?

You would need to write one SQL statement for each class_id plus dance_grade combination. This would result in four SQL statements for class_id=1:

- UPDATE class_enrollments SET dance_routine=‘A’
  WHERE class_id=1 AND dance_grade BETWEEN 66 and 75;
- UPDATE class_enrollments SET dance_routine =‘B’
  WHERE class_id=1 AND dance_grade BETWEEN 56 and 65;
- UPDATE class_enrollments SET dance_routine =‘C’
  WHERE class_id=1 AND dance_grade BETWEEN 46 and 55;
- UPDATE class_enrollments SET dance_routine =‘D’
  WHERE class_id=1 AND dance_grade BETWEEN 36 and 45

There are a lot of statements to code for this one small example. There has to be a better way to complete this task. Just think how long it would take the system to run through all this code for all the data in a table.

Copyright © 2009, Oracle. All rights reserved.

This is a lot of statements and it doesn't even include the statements for the other classes! Similarly, there would be four statements for class_id=2.

Wouldn't it be easier to write a single statement to accomplish this task? The statement would require logic, otherwise known as conditional or procedural logic. PL/SQL allows you to incorporate procedural logic with SQL. PL/SQL extends SQL with procedural logic.

These statements can and will be condensed using one PL/SQL conditional statement. Just think how long it would take the system NOW to run through all this code for all the data in a table….a lot less time would be needed.

Slide 8: Tell Me / Show Me – Limitations of SQL
No instructor notes for this slide

Slide 9: Tell Me / Show Me – Limitations of SQL
No instructor notes for this slide

Slide 10: Tell Me / Show Me – Limitations of SQL
No instructor notes for this slide

Slide 11: Tell Me / Show Me – PL/SQL extends SQL with Procedural Logic
No instructor notes for this slide

Slide 12: Tell Me / Show Me – Procedural Constructs
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.

Slide 13: Tell Me / Show Me – Procedural Constructs
In fact everything in PL/SQL is a procedural construct. The slide highlights only some of them.

Slide 14: Tell Me / Show Me – Terminology
PL/SQL – Oracle Corporations standard procedural language for relational databases which allows basic program logic and control flow to be combined with SQL statements

Slide 15: Tell Me / Show Me – Summary
No instructor notes for this slide

Slide 16: Try It / Solve It
No instructor notes for this slide

Copyright © 2009, Oracle. All rights reserved.

SECTION 1 LESSON 2 – Benefits of PL/SQL

Slide 1: Benefits of PL/SQL
No instructor notes for this slide

Slide 2: What Will I Learn?
No instructor notes for this slide

Slide 3: Why Learn It?
No instructor notes for this slide

Slide 4: Tell Me / Show Me – Benefits of PL/SQL
All of these benefits have a major impact on why PL/SQL was developed and how it is used.

Slide 5: Tell Me / Show Me – Benefit 1: Integration of procedural constructs with SQL
Integration is a fantastic process. You can get your data to accomplish all types of tasks that are needed for businesses.

Slide 6: Tell Me / Show Me – Benefit 2: Modularized program development
Modular programs are considered to be old news. Good programming practice uses modular programs to break program control into usable and understand sections.

Slide 7: Tell Me / Show Me – Benefit 2: Modularized program development (continued)
No instructor notes for this slide

Slide 8: Tell Me / Show Me – Benefit 2: Modularized program development (continued)
No instructor notes for this slide

Slide 9: Tell Me / Show Me – Benefit 3: Improved performance
When sending programs through the system, one block (section) of code, instead of all separate lines of code, makes the program process faster.

The following features also result in improved performance:
1. PL/SQL **variables** store data in the same internal binary format as the database does, so no data conversion is needed.
2. PL/SQL is executed in the same memory space as the Oracle server and therefore there is no communications overhead between the two programs.
3. PL/SQL **functions** can be called directly from SQL.
4. A special kind of PL/SQL procedure, called a **trigger**, can execute automatically whenever something important happens in the database.

Slide 10: Tell Me / Show Me – Benefit 4: Integration with Oracle tools
Oracle has a set of tools that are very usable. PL/SQL is used by all these tools.

Copyright © 2009, Oracle. All rights reserved.

Slide 11: Tell Me / Show Me – Benefit 5: Portability
Portability speaks for itself. PL/SQL can be used anywhere Oracle runs.

Slide 12: Tell Me / Show Me – Benefit 6: Exception handling
One good programming practice is to write code so that an abrupt end of program does not happen. This is very hard to maintain when there are many users. Error handling is used to keep the abrupt end of program from happening.

Slide 13: Tell Me / Show Me – PL/SQL Compared to Other Languages
Requires Oracle database or tool: With PL/SQL you cannot create a PL/SQL program that runs all by itself; PL/SQL programs must be run from the Oracle server or an Oracle tool. C and Java programs do not require an Oracle database to run or compile. You can develop standalone programs using Java and C.
Object-oriented: Java is an object-oriented language and C is not. PL/SQL has included some object-oriented features such as abstract data types, multi-level collections, encapsulation, function overloading, and inheritance.
Performance against an Oracle database: PL/SQL is tightly integrated with the database and as such can result in highly efficient programs. Java and C are less efficient because they are not as integrated with the database.
Portable to different operating systems: PL/SQL is portable to Oracle databases on different operating systems (assuming version compatibility). Java is also highly portable. Different C compilers and libraries are not 100% compatible on different operating systems.
Ease of learning: PL/SQL is fairly easy to learn, whereas Java requires knowledge of object-oriented programming. C also is a more complex programming language than PL/SQL.

Slide 14: Tell Me / Show Me – PL/SQL in Oracle Products
No instructor notes for this slide

Slide 15: Tell Me / Show Me – Terminology
Blocks – The basic unit of PL/SQL programs- also known as modules.
Portability – The ability for PL/SQL programs to run anywhere an Oracle server runs.
Exceptions – An error that occurs in the database or in a user's program during runtime.

Slide 16: Summary
No instructor notes for this slide

Slide 17: Try It / Solve It
No instructor notes for this slide

Copyright © 2009, Oracle. All rights reserved.

SECTION 1 LESSON 3 – Creating PL/SQL Blocks

Slide 1: Creating PL/SQL Blocks
No instructor notes for this slide

Slide 2: What Will I Learn?
No instructor notes for this slide

Slide 3: Why Learn It?
No instructor notes for this slide

Slide 4: Tell Me / Show Me – PL/SQL Block Structure
Students will learn about nested blocks later in the course.

Slide 5: Tell Me / Show Me – PL/SQL Block Structure (continued)
No instructor notes for this slide

Slide 6: Tell Me / Show Me – PL/SQL Block Structure (continued)
DECLARE is not needed if no variables, constants, cursors or user-defined exceptions are required. But nearly all real-life blocks will need variables and/or cursors, therefore nearly all real-life blocks will need a DECLARE section.

Slide 7: Tell Me / Show Me – The PL/SQL Compiler
Students will learn later that we can recompile stored subprograms (procedures, functions and packages) on demand by (for example): ALTER PROCEDURE procedure_name COMPILE;

Students will learn more about the needed privileges in Section 8.

Slide 8: Tell Me / Show Me – Block Types
No instructor notes for this slide

Slide 9: Tell Me / Show Me – Anonymous Blocks
No instructor notes for this slide

Slide 10: Tell Me / Show Me – Examples of Anonymous Blocks
DBMS_OUTPUT.PUT_LINE is explained at the end of this lesson.

Slide 11: Tell Me / Show Me – Examples of Anonymous Blocks (continued)
If students ask “what is TOO_MANY_ROWS ?”, tell them it means that a SELECT statement fetched more than one row from the database; in this example, it would mean that the COUNTRIES table contains at least two rows whose country_id = 'CA'.

Copyright © 2009, Oracle. All rights reserved.

Slide 12: Tell Me / Show Me – Subprograms

Every language has a syntax, a vocabulary, and a character set. You have to learn the rules that govern its usage. PL/SQL is made up of many BLOCKS of code. A PL/SQL block has up to four sections, only one is mandatory.

Header - optional
Declaration section – DECLARE identifiers - optional
Executable section – BEGIN…END; - mandatory
Exception - Error Handling - optional

Named Program:

```
PROCEDURE name (Header, Optional)
IS
DECLARE (Optional)
        Variables, cursors, user-     defined exceptions
BEGIN ( Executable, start, Mandatory)
        - SQL statements
        - PL/SQL statements
EXCEPTION (Optional)
        Actions to perform when     errors occur
END; (Executable, end, Mandatory)
```

Anonymous Block with a declaration section:

```
/*Declaration section program*/
DECLARE
        words varchar2(15) := ‘Hello World’;

BEGIN
        DBMS_OUTPUT.PUT_LINE(words);
END;
```

Anonymous Block without a declaration section:

```
/*No Declaration section program*/
BEGIN
        DBMS_OUTPUT.PUT_LINE(‘Hello’ || ‘ World’);
END;
```

Slide 13: Tell Me / Show Me – Examples of Subprograms

No instructor notes for this slide. Note that this slide is providing an example of a procedure and a function merely for you to compare and contrast, not necessarily for you or the students to execute.

Slide 14: Tell Me / Show Me – Program Constructs

No instructor notes for this slide

Copyright © 2009, Oracle. All rights reserved.

Slide 15: Tell Me / Show Me – PL/SQL Programming Environments
No instructor notes for this slide

*Slide 16: Tell Me / Show Me – PL/SQL Programming Environments: iSQL*Plus*
No instructor notes for this slide

Slide 17: Tell Me / Show Me – PL/SQL Programming Environments: Oracle JDeveloper
JDeveloper is a Windows- or UNIX-based application. It is not a browser-based application.
JDeveloper is intended mainly for programmers writing Java applications, but can be used to write and debug PL/SQL also.

Slide 18: Tell Me / Show Me – PL/SQL Programming Environments: Oracle Application Express
No instructor notes for this slide

Slide 19: Tell Me / Show Me – Developing with SQL Workshop
No instructor notes for this slide

Slide 20: Tell Me / Show Me – SQL Commands
The slide shows an example of a PL/SQL anonymous block in SQL Commands. The following is an example of a SQL Script, containing two SQL statements and one anonymous PL/SQL block. Note that in a script, anonymous PL/SQL blocks must be followed by a forward slash (/). This is not needed in SQL Commands (although it can be used).

```
SELECT count(*) FROM employees;              -- SQL statement

DECLARE                                  -- PL/SQL block
  v_count     NUMBER(6,0);
BEGIN
  SELECT count(*) INTO v_count FROM departments;
  DBMS_OUTPUT.PUT_LINE(v_count);
END;
/
SELECT sysdate FROM dual;
```

Slide 21: Tell Me / Show Me – Using DBMS_OUTPUT.PUT_LINE
No instructor notes for this slide

Slide 22: Tell Me / Show Me – Using DBMS_OUTPUT.PUT_LINE (continued)
No instructor notes for this slide

Slide 23: Tell Me / Show Me – Using DBMS_OUTPUT.PUT_LINE (continued)
Students will learn more about DBMS_OUTPUT in Section 9.

Copyright © 2009, Oracle. All rights reserved.

The second call in the slide shows that number values (v_emp_count) can be displayed by DBMS_OUTPUT. In this example, the Oracle server has performed an implicit datatype conversion (TO_CHAR(v_emp_count)) to convert the number to a character string for concatenation.

Slide 24: Tell Me / Show Me – Terminology
Anonymous PL/SQL block – unnamed blocks of code not stored in the database and do not exist after they are executed
Compiler – software that checks and translates programs written in high-level programming languages into binary code to execute
Subprograms – named PL/SQL blocks that are stored in the database and can be declared as procedures or functions
Procedures – programs that perform an action and may return values
Functions – programs that compute and return a single value

Slide 25: Summary
No instructor notes for this slide

Slide 26: Try It / Solve It
No instructor notes for this slide

Copyright © 2009, Oracle. All rights reserved.

SECTION 1 LESSON 4 – Review of SQL Select Statements

Slide 1: Review of SQL Select Statements
No instructor notes for this slide

Slide 2: What Will I Learn?
This lesson is a quick review of basic SQL statements. Database Design and Database Programming with SQL are prerequisites to this course, therefore it is assumed that students will be familiar with most of the content.

Slide 3: Why Learn It?
No instructor notes for this slide

Slide 4: Tell Me / Show Me – Selecting Data
No instructor notes for this slide

Slide 5: Tell Me / Show Me – Case Study
Please refer to Section 0 for a description of the World Facts case study.

Slide 6: Tell Me / Show Me – Select Data
Ask the class the following questions about the World Facts dataset:

What SQL command will return all the data in the WF_COUNTRIES table?
Answer: SELECT * FROM wf_countries;

What SQL command will retrieve the name and area of the country with ID 245?
Answer: SELECT area, country_name FROM wf_countries WHERE country_id = 245;

What SQL command will retrieve the ID, name, and area of all the countries?
Answer: SELECT country_id, country_name, area FROM wf_countries;

Slide 7: Tell Me / Show Me – Sorting
No instructor notes for this slide

Slide 8: Tell Me / Show Me – Calculations
No instructor notes for this slide

Slide 9: Tell Me / Show Me – Column aliases
No instructor notes for this slide

Slide 10: Tell Me / Show Me – Concatenation
Remind students that the concatenation operator does not automatically insert spaces before and after a literal.

Copyright © 2009, Oracle. All rights reserved.

Slide 11: Tell Me / Show Me – DISTINCT
Clarify that the results shown in the example are a partial list from the actual data. It is probably best to demonstrate this by executing the statements in Application Express so that students can see all the data.

Slide 12: Tell Me / Show Me – DISTINCT (continued)
Clarify that the results shown in the example are a partial list from the actual data. It is probably best to demonstrate this by executing the statements in Application Express so that students can see all the data.

Slide 13: Tell Me / Show Me – BETWEEN...AND
Ask the class to point out the difference between the example and the following statement:

```
SELECT country_name, coastline FROM wf_countries
  WHERE coastline >= 500 AND coastline <= 550;
```

Answer: No difference. The results are the same and there is no difference in performance.

Slide 14: Tell Me / Show Me – IN
Ask the class to point out the difference between the example and the following statement:

```
SELECT region_id, country_name FROM wf_countries
  WHERE region_id = 5 OR region_id = 9;
```

Answer: No difference. The results are the same and there is no difference in performance.

Slide 15: Tell Me / Show Me – LIKE
Clarify that the results shown in the example are a partial list from the actual data. It is probably best to demonstrate this by executing the statements in Oracle Application Express so that students can see all the data.

Slide 16: Tell Me / Show Me – Terminology
Concatenation – to connect or link together in a series
DISTINCT – Keyword used to eliminate duplicate rows from the output of a SQL statement.
BETWEEN…AND – Operator used to select and display rows based on a range of values.
IN – Condition used to test whether a value is in a specified set of values.
LIKE– Condition allowing you to select rows that match either literal strings or number patterns.

Copyright © 2009, Oracle. All rights reserved.

Slide 17: Summary
No instructor notes for this slide

Slide 18: Try It / Solve It
No instructor notes for this slide

Copyright © 2009, Oracle. All rights reserved.

SECTION 1 LESSON 5 – Review of SQL Single-Row Functions

Slide 1: Review of SQL Single-Row Functions
No instructor notes for this slide.

Slide 2: What Will I Learn?
These are some of the more commonly used single-row functions, but is not a complete list.

Slide 3: Why Learn It?
No instructor notes for this slide

Slide 4: Tell Me / Show Me – Case Manipulation Functions
The three example statements return the same row. Remind students that case manipulation functions are especially useful when one is not sure of what case (upper, lower, initial caps) the data is stored in. The functions help avoid a mismatch between the case of the query and the database case storage.

Slide 5: Tell Me / Show Me – Case Manipulation Functions (continued)
No instructor notes for this slide

Slide 6: Tell Me / Show Me – Character Manipulation Functions
Walk the class through other examples that use literals. This will be a good way to review the DUAL table.

```
SELECT CONCAT('Hello',' World') FROM dual;
  returns 'Hello World'
SELECT SUBSTR ('Hello World',7,5) FROM dual;
  returns 'World'
SELECT LENGTH ('Hello World') FROM dual;
  returns 11
```

Slide 7: Tell Me / Show Me – Character Manipulation Functions
Note that if a column is defined as datatype CHAR instead of VARCHAR2, the LENGTH function will include trailing spaces.

Slide 8: Tell Me / Show Me – Number Functions
No instructor note for this slide

Slide 9: Tell Me / Show Me – Number Functions (continued)
Point out that one of the more common uses of MOD is to determine whether a number is odd or even. Even numbers will have no remainder when divided by 2.

Slide 10: Tell Me / Show Me – Conversion Functions
No instructor notes for this slide

Copyright © 2009, Oracle. All rights reserved.

Slide 11: Tell Me / Show Me – Conversion Functions (continued)
No instructor notes for this slide

Slide 12: Tell Me / Show Me – Conversion Functions (continued)
No instructor notes for this slide

Slide 13: Tell Me / Show Me – Conversion Functions (continued)
No instructor notes for this slide

Slide 14: Tell Me / Show Me – Date Functions
Remind students that the Oracle database's default unit of date arithmetic is one day: "+5" means "add 5 days". To add 5 hours, we would use "+5/24".
The slide example assumes that today's date is 30 November 2006.

Slide 15: Tell Me / Show Me – Date Functions (continued)
Remind students that the DATE data type always stores year information as a four-digit number internally: two digits for the century and two digits for the year. For example, the Oracle database stores the year as 1996 or 2005, not just as 96 or 05. The century component is not displayed by default.
In the example, the value returned by the MONTHS_BETWEEN function is formatted using a TO_CHAR function.

Slide 16: Tell Me / Show Me – Date Functions (continued)
Ask students: why do we need the MONTHS_BETWEEN and ADD_MONTHS functions?

Answer: Because not all months contain the same number of days. To add one month to SYSDATE, we cannot use SYSDATE+31. What if today is in February?
(The slide example assumes that today's date is 30 November 2006.)

Slide 17: Tell Me / Show Me – General Functions
Review the meaning of NULL with the class. NULL is a value that is unknown. It isn't equal to zero nor a blank space.

Slide 18: Tell Me / Show Me – General Functions (continued)
No instructor notes for this slide

Slide 19: Tell Me / Show Me – General Functions (continued)
Walk through the NULLIF example. Point out that it nests two functions, NULLIF and TO_CHAR. The date_of_independence column is a date, but the national_holiday_date column is of character data type. Therefore, we converted the date_of_independence to a character string so we could compare the two columns.

Slide 20: Summary
These are some of the more commonly used single-row functions, but is not an exhaustive list.

Copyright © 2009, Oracle. All rights reserved.

Slide 21: Try It / Solve It
No instructor notes for this slide

Copyright © 2009, Oracle. All rights reserved.

PRACTICE SOLUTIONS

SECTION 1 LESSON 1 - Introduction to PL/SQL

Terminology
Directions: Identify the vocabulary word for each definition below:

1. _______**PL/SQL**___________ Oracle Corporations standard procedural language for relational databases which allows basic program logic and control flow to be combined with SQL statements.

Try It/Solve It Questions

1. Circle the programming language meeting the criteria

			Answer
3GL	PL/SQL	SQL	**PL/SQL**
4GL	PL/SQL	SQL	**SQL**
Is proprietary to Oracle Corporation	PL/SQL	SQL	**PL/SQL**
Nonprocedural	PL/SQL	SQL	**SQL**
Procedural	PL/SQL	SQL	**PL/SQL**
Is ANSI-compliant	PL/SQL	SQL	**SQL**

2. In your own words, describe why a procedural language like PL/SQL is needed.

It allows logic and control to be combined with basic SQL statements, making it possible to create more useful programs.

3. What is a procedural construct?

A logical use of a programming concept in a programming language.

4. List some examples of procedural constructs in PL/SQL.

Variables, constants, conditional statements, loops, cursors, reusable program units.

Copyright © 2009, Oracle. All rights reserved.

5. In the following code, identify and circle examples of these procedural constructs: variable, conditional control, SQL statement

```
DECLARE
     v_first_name varchar2(40);
     v_last_name varchar2(40);
     v_first_letter varchar2(1);
BEGIN
     SELECT first_name, last_name into v_first_name, v_last_name
               FROM students
               WHERE student_id=105;

     v_first_letter := get_first_letter(last_name);
     IF 'N' > 'v_first_letter' THEN
DBMS_OUTPUT.PUT_LINE('The last name for:   '||first_name||' '||last_name||' is
between A and M');
     ELSE
DBMS_OUTPUT.PUT_LINE('The last name for: '||first_name||' '||last_name||' is
between N and Z');
     END IF;
END;
```

Variables: v_first_name, v_last_name, v_first_letter.
Conditional control: IF … ELSE … END IF;
SQL statement: SELECT …. WHERE student_id = 105;

Copyright © 2009, Oracle. All rights reserved.

SECTION 1 LESSON 2 - Benefits of PL/SQL

Terminology

Directions: Identify the vocabulary word for each definition below:

1. **______Portability________** The ability for PL/SQL programs to run anywhere an Oracle server runs.
2. **______Blocks___________** The basic unit of PL/SQL programs- also known as modules.
3. **______Exceptions________** An error that occurs in the database or in a user's program during runtime.

Try It/Solve It Questions

1. Why is it more efficient to combine SQL statements into PL/SQL blocks?

An application is more efficient when created in PL/SQL blocks; there is less network traffic, resulting in a faster application.

2. Why is it beneficial to use PL/SQL with an Oracle database? List at least three reasons.

1) It allows the programmer to combine procedural constructs with SQL.
2) The basic unit of a PL/SQL program is a block, and blocks allow the programmer to create code that is easier to read and maintain.
3) PL/SQL allows the programmer to combine multiple SQL statements into a single PL/SQL program. Sending a single PL/SQL program to the database server for processing is more efficient than sending multiple SQL statements.
4) PL/SQL is integrated into several Oracle tools such as Forms, Reports, and Oracle Application Express.
5) You can develop PL/SQL code on one operating system and deploy it on another operating system because PL/SQL code is independent of the operating system and the platform.
6) PL/SQL allows the programmer to prepare for errors by writing exception handling logic into code; PL/SQL easily handles data-oriented exceptions such as no data found and too many rows.

3. How is PL/SQL different from C and Java? List three differences.

1) PL/SQL requires a database or Oracle tool..
2) PL/SQL allows for some object-oriented programming techniques, but is not as extensive as Java.
3) PL/SQL is the most efficient language to use with an Oracle database.
4) PL/SQL is portable to other operating systems, as long as it is running a compatible Oracle database (that is, a database with the same version).
5) PL/SQL is easier to learn than C and Java.

Copyright © 2009, Oracle. All rights reserved.

4. List three examples of what you can build with PL/SQL code.

Using PL/SQL, you can:

1) **Create Web and other applications**
2) **Manage application data (write programs for DDL or DML)**
3) **Manage the Oracle database (write programs for security, for managing database jobs)**
4) **Create custom reports**

Copyright © 2009, Oracle. All rights reserved.

SECTION 1 LESSON 3 - Creating PL/SQL Blocks

Terminology

1. **_Anonymous PL/SQL block__** Unnamed blocks of code not stored in the database and do not exist after they are executed.
2. **_Functions_________________** A program that computes and returns a value.
3. **_Subprograms______________** Named PL/SQL blocks that are stored in the database and can be declared as procedures or functions.
4. **__Complier________________** Software that checks and translates programs written in high-level programming languages into binary code to execute.
5. **_ Procedure________________** A program that performs an action and does not have to return a value.

Try It/Solve It Questions

1. Complete the following chart defining the syntactical requirements for a PL/SQL block:

	Optional or Mandatory?	What is included in this section?
DECLARE	**Optional**	**Variables, cursors, user-defined exceptions**
BEGIN	**Mandatory**	**SQL statements** **PL/SQL statements**
EXCEPTION	**Optional**	**Actions to perform when errors occur**
END;	**Mandatory**	**End; (do not forget the semicolon!)**

2. Which of the following PL/SQL blocks execute successfully? For the blocks that fail, explain why they fail

A.
```
BEGIN
END;
```

B.
```
DECLARE
   amount INTEGER(10);
END;
```

C.
```
DECLARE
BEGIN
END;
```

Copyright © 2009, Oracle. All rights reserved.

D.
```
DECLARE
  amount NUMBER(10);
BEGIN
  DBMS_OUTPUT.PUT_LINE(amount);
END;
```

A. Fails because the executable section must contain at least one statement.
B. Fails because there is no executable section (BEGIN is missing).
C. Fails because the executable section must contain at least one statement.
D. Succeeds.

3. Fill in the blanks:

   A. PL/SQL blocks that have no names are called **___anonymous blocks_____**.

   B. **__Procedures____** and **___Functions______** are named blocks and are stored in the database.

4. In Application Express, create and execute a simple anonymous block that outputs "Hello World."

```
BEGIN
  DBMS_OUTPUT.PUT_LINE ('Hello World');
END;
```

Extension Exercise

1. Create and execute a simple anonymous block that does the following:
   - Declares a variable of datatype DATE and populates it with the date that is six months from today
   - Outputs "In six months, the date will be: <insert date>."

```
DECLARE
 v_timestamp DATE;
BEGIN
 SELECT ADD_MONTHS(SYSDATE,6) INTO v_timestamp FROM dual;
 DBMS_OUTPUT.PUT_LINE('In six months, the date will be: '||v_timestamp);
END;
```

Copyright © 2009, Oracle. All rights reserved.

SECTION 1 LESSON 4 - Review of SQL SELECT Statements

Terminology

1. **__BETWEEN...AND_______** Operator used to select and display rows based on a range of values.
2. **__Concatenation___________** to connect or link together in a series
3. **__LIKE__________________** A condition allowing you to select rows that match either literal strings or number patterns.
4. **__DISTINCT______________** Keyword used to eliminate duplicate rows from the output of a SQL statement.
5. **___IN____________________** Condition used to test whether a value is in a specified set of values.

Try It/Solve It Questions

1. Display a list of all country names and their lowest and highest elevations from the wf_countries table. Display the highest elevation as HIGH, and the lowest elevation as "low point." Sort results by country name.

```
SELECT country_name, lowest_elevation "low point", highest_elevation high
   FROM wf_countries
   ORDER BY country_name;
```

2. Which country has a lowest_elevation > 1000 and a highest_elevation < 5000?

```
SELECT country_name, lowest_elevation "low point", highest_elevation high
   FROM wf_countries
   WHERE lowest_elevation > 1000
   AND highest_elevation < 5000;
```

Kingdom of Lesotho

3. Calculate the projected population growth for each country in the wf_countries table by multiplying population by population growth rate. Display the calculated value as PROJECTION.

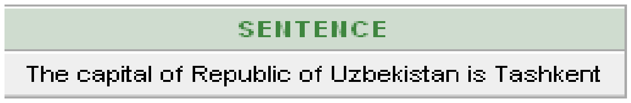
```
SELECT country_name, population, population_growth_rate,
           population * population_growth_rate projection
  FROM wf_countries;
```

4. Display the list of national holiday names from the wf_countries table. Eliminate duplicate names. Display the output in ascending alphabetic sequence.

```
SELECT DISTINCT national_holiday_name
   FROM wf_countries
   ORDER BY national_holiday_name;
```

Copyright © 2009, Oracle. All rights reserved.

5. What SQL statement will return the following? (Use the country_id in the WHERE clause.)



SENTENCE
The capital of Republic of Uzbekistan is Tashkent

```
SELECT 'The capital of ' || country_name || ' is '|| capitol as sentence
   FROM wf_countries
  WHERE country_id = 998;  -- (or WHERE country_name LIKE  ' %Uzbek%';)
```

6. Display the countries that have an “s” as the third letter of their names.

```
SELECT country_name
   FROM wf_countries
  WHERE country_name LIKE '__s%';
-- (or WHERE  SUBSTR(country_name,3,1) = 's';)
```

7. Write a SQL statement that will give you the same results as:

```
SELECT country_name, area
  FROM wf_countries
  WHERE area BETWEEN 100 AND 200;
```

```
SELECT country_name, area
   FROM wf_countries
   WHERE area >= 100 AND area <=200;
```

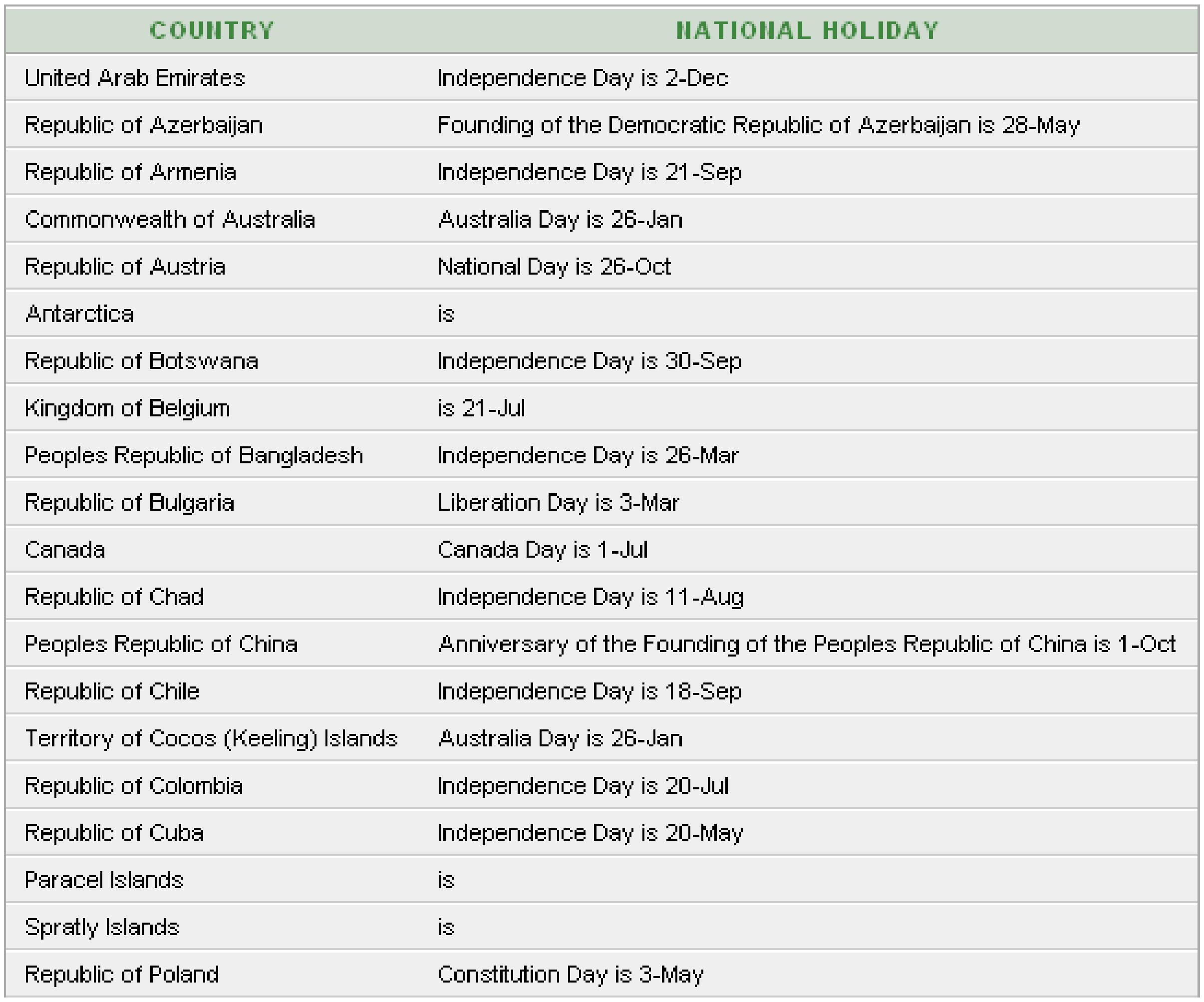
8. Write a SQL statement that will give you the same results as:

```
SELECT region_id, country_name
  FROM wf_countries
  WHERE region_id = 1 OR region_id = 7 OR region_id = 10;
```

```
SELECT region_id, country_name
   FROM wf_countries
  WHERE region_id IN (1, 7,10)
```

Copyright © 2009, Oracle. All rights reserved.

9. Write a SQL statement that will return data that looks like this:



COUNTRY	NATIONAL HOLIDAY
United Arab Emirates	Independence Day is 2-Dec
Republic of Azerbaijan	Founding of the Democratic Republic of Azerbaijan is 28-May
Republic of Armenia	Independence Day is 21-Sep
Commonwealth of Australia	Australia Day is 26-Jan
Republic of Austria	National Day is 26-Oct
Antarctica	is
Republic of Botswana	Independence Day is 30-Sep
Kingdom of Belgium	is 21-Jul
Peoples Republic of Bangladesh	Independence Day is 26-Mar
Republic of Bulgaria	Liberation Day is 3-Mar
Canada	Canada Day is 1-Jul
Republic of Chad	Independence Day is 11-Aug
Peoples Republic of China	Anniversary of the Founding of the Peoples Republic of China is 1-Oct
Republic of Chile	Independence Day is 18-Sep
Territory of Cocos (Keeling) Islands	Australia Day is 26-Jan
Republic of Colombia	Independence Day is 20-Jul
Republic of Cuba	Independence Day is 20-May
Paracel Islands	is
Spratly Islands	is
Republic of Poland	Constitution Day is 3-May

```
SELECT country_name COUNTRY,
        national_holiday_name ||' is '|| national_holiday_date AS "NATIONAL
HOLIDAY"
  FROM wf_countries;
```

10. Due to a successful global health program, the life expectancy at birth will increase by 7 years for countries whose median ages are below 20. Display the current and increased life expectancy for these countries in the wf_countries table. Name the calculated column “Improved Expectancy.”

```
SELECT country_id, country_name,  life_expect_at_birth,
        life_expect_at_birth + 7 as "Improved Expectancy"
  FROM wf_countries
 WHERE median_age < 20;
```

Copyright © 2009, Oracle. All rights reserved.

SECTION 1 LESSON 5 - Review of SQL Single-Row Functions

Terminology
No new vocabulary for this lesson.

Try It/Solve It Questions

1. Write a SQL statement that lists the country names in alphabetical order. Display the results in uppercase and give the column a heading of "NAME".

```
SELECT UPPER(country_name) AS name
   FROM wf_countries
   ORDER BY country_name;
```

2. Display all the languages in the wf_languages table that start with "f." Use the lowercase "f" in your SQL statement.

```
SELECT *
  FROM wf_languages
  WHERE LOWER(language_name) like 'f%';
```

3. From the wf_world_regions table, display the ID, name, and an abbreviation that is the first three characters of the region name.

```
SELECT region_id, region_name, SUBSTR(region_name,1,3) AS abbreviation
   FROM wf_world_regions;
```

4. Modify your SQL statement so that the abbreviation is the first three characters of the region name followed by the length of the region name. For example: Western Asia would be Wes12.

```
SELECT region_id, region_name,
   CONCAT(SUBSTR(region_name,1,3), LENGTH(region_name)) AS abbreviation
   FROM wf_world_regions;
```

5. Display all country names from the wf_countries table, along with the life expectancy, rounded to the nearest whole number.

```
SELECT country_name, ROUND(life_expect_at_birth, 0)
   FROM wf_countries;
```

Copyright © 2009, Oracle. All rights reserved.

6. Display only those countries from the wf_countries table whose median age, when rounded, is an even number.

SELECT country_name, median_age, ROUND(median_age,0)
FROM wf_countries
WHERE MOD(ROUND(median_age, 0), 2) = 0;

7. Display the date 100 months from today. Choose your own format model.

SELECT TO_CHAR(ADD_MONTHS(SYSDATE, 100), 'fmddth Month, RRRR')
as "100 months from today"
FROM dual;

Note: the format may vary depending on what the student chooses.

8. List the country name and coastline from wf_countries. Format it so that a coastline of 1000 will display as 1,000.

SELECT country_name, TO_CHAR(coastline,'999,999')
FROM wf_countries;

9. Write a SQL statement to list country names and capitals. If the capital is null, display it as “none listed.”

SELECT country_name, NVL(capitol,'none listed')
FROM wf_countries;

10. List the language ids and associated comments from wf_spoken_languages. If comments are null, display it as “No comment.” Give your column an alias.

SELECT language_id, NVL(comments,'No comment') AS comments
FROM wf_spoken_languages;

Note: The alias may vary depending on what the student chooses.

Copyright © 2009, Oracle. All rights reserved.