# *Database Management II (420-D20-HR)*

# *Lab 1 – SQL Review and Introduction to PL/SQL*

Date assigned: Tuesday, January 24, 2017

Date due: **Tuesday, January 24, 2017**

**Objectives:**

At the end of this lab you will be able to:

* use the ANSI/ISO standard joins in SQL queries (review)
* use variables, constants, data types and declarations in an anonymous PL/SQL block;
* use the PL/SQL assignment statement and arithmetic operators;
* write a simple PL/SQL block;

**References:**

Class notes and samples

[Variables](http://www.plsqltutorial.com/plsql-variables/)

[Block structure](http://www.plsqltutorial.com/plsql-block-structure/)

**Preparation:**

Your professor will walk through a sample PL/SQL program that demonstrates:

1. How to code and run PL/SQL in Oracle SQL developer
2. The basic structure of PL/SQL code blocks
3. Basic input/output in PL/SQL
4. Variables in PL/SQL

**Marking and Time management:**

|  |  |  |  |
| --- | --- | --- | --- |
| Section | Question | Mark | Effort (minutes) |
| 1. Review | Quiz | 8 | 0 |
|  | SQL Joins -i | 6 | 10 |
|  | SQL Joins –ii | 8 | 5 |
|  | SQL Joins -iiii | 8 | 5 |
| 1. PL/SQL Intro | 1 | 8 | 10 |
|  | 2 | 10 | 10 |
|  | 3 | 12 | 10 |
| 1. User-defined identifiers |  | 10 | 5 |
| English and handed in properly |  | 5 | 0 |
| Totals |  | 75 | 0 |

**To Start:**

1. Rename this document to ***username*\_D20\_L01\_PL\_SQL\_Intro.docx**
2. Start **SQL Developer** and connect to your Oracle account.

**To be handed in:**

1. ***username\_*D20\_L01\_PL\_SQL\_Intro.docx** containing your responses created for this lab should uploaded to **Moodle**. Please remember to fill in the Effort section of the Marking Rubric.
2. The **Lab 1 Review Quiz** should be completed in **Moodle**.

**For each of the blocks created for the following problems, include a comment block at the top of the block with the question number, your name and a brief description of what the block does. Insert comments throughout to explain the steps.**

**Use ISO/ANSI standards joins for all joins.**

**Name all identifiers according to the naming standards shown below.**

**Format all blocks using the SQL Developer Formatter**

**Naming Standards:**

|  |  |
| --- | --- |
| **Identifier type** | **Prefix** |
| local variables | lv\_ |
| local constant | lc\_ |

# SQL Review

## Complete the Lab 1 Review Quiz in Moodle.

## ISO/ANSI Standard Joins

*Purpose:* Review SQL queries and how to write inner and outer joins in Oracle using ISO/ANSI standard notation.

*To Do:*

### Install the Shah table database (see Moodle for the “Recreate Shah Table” sql script. You should have already done this as part of the in-class exercises)

### Provide the SQL statement for the following queries. They MUST be ISO/ANSI standard joins syntax. You must also provide your sample output.

#### List all classrooms in use on Mondays at 9:00 in the SP03 term:

**SQL:**

**SELECT cs.roomid**

**FROM iu\_crssection cs**

**INNER JOIN iu\_location l**

**on l.ROOMID = cs.ROOMID**

**WHERE day like 'M%'**

**and termid = 'SP03'**

**and starttime = '09:00'**

**and l.ROOMTYPE = 'C'**

**;**

**Sample output:**



#### List all the course sections and the name of faculty member teaching each. Include any course sections that have not been assigned a teacher.

**SQL:**

**SELECT cs.csid, nvl(f.name, 'unassigned') AS "Teacher"**

**FROM iu\_crssection cs**

**LEFT JOIN iu\_faculty f**

**on cs.facultyid = f.FACULTYID**

**Sample output:**

****

#### Write a query to list the NamanNavan supervisors and the employees that they supervise. Include employees with no supervisor.

**SQL:**

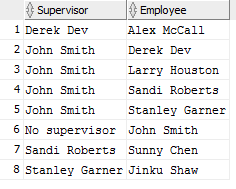
**SELECT REGEXP\_REPLACE(e2.fname||' '||e2.lname, '^ $', 'No supervisor') AS "Supervisor",**

**e.fname||' '||e.lname AS "Employee"**

**FROM nn\_employee e**

**LEFT JOIN NN\_EMPLOYEE e2**

**on e.supervisor = e2.employeeidSample output:**



***Expected Output:***

|  |  |
| --- | --- |
| **Supervisor** | **Employee** |
| **No supervisor** | **John Smith** |
| **John Smith** | **Larry Houston** |
| **John Smith** | **Sandi Roberts** |
| **Derek Dev** | **Alex McCall** |
| **John Smith** | **Derek Dev** |
| **Stanley Garner** | **Jinku Shaw** |
| **John Smith** | **Stanley Garner** |
| **Sandi Roberts** | **Sunny Chen** |

# PL/SQL Intro

***Objectives:***

* Learn to code a simple PL/SQL block
* Learn to display messages using the PUT\_LINE procedure of the DBMS\_OUTPUT package
* Learn to use substitution variables to input data to a PL/SQL block
* Select into scalar variables

***To Do:***

## Write a PL/SQL block to find the square, cube and double of a number inputted with a substitution variable, and print the results using the built-in package DBMS\_OUTPUT. Once your PL/SQL block is working, format it and copy it below. Include a sample of the output as well.

If 7 was entered as the number, the output should look similar to:

**The square of 7 is 49**

**The cube of 7 is 343**

**The double of 7 is 14**

**PL/SQL:**

**SET SERVEROUTPUT ON;**

**DECLARE**

**num1 number;**

**BEGIN**

**num1 := &Enter\_a\_number;**

**DBMS\_OUTPUT.PUT\_LINE('The cube of '|| num1 || ' is ' || num1\*num1\*num1);**

**DBMS\_OUTPUT.PUT\_LINE('The square of '|| num1 || ' is ' || num1\*num1);**

**DBMS\_OUTPUT.PUT\_LINE('Double '|| num1 || ' is ' || num1\*2);**

**END;**

**Sample output:**

**The cube of 7 is 343**

**The square of 7 is 49**

**Double 7 is 14**

## Write a PL/SQL program to input hours and rate. Determine gross pay and net pay with a tax rate of 28%. Print your results. (No need to calculate overtime). Note the following:

* + The tax rate should be declared as a constant.
  + Use %TYPE to declare the gross pay and net pay to have the same data type as the pay rate.
  + The gross pay, the tax and the net pay should be displayed in currency format.

If 40 hours were worked at $15 an hour, the output would look similar to:

**Hours: 40**

**Pay rate: $15.00**

**Gross Pay: $600.00**

**Tax: $168.00**

**Net Pay: $432.00**

**PL/SQL:**

**DECLARE**

**lv\_hours number;**

**lv\_rate number;**

**lc\_tax CONSTANT NUMBER (3, 2) := 0.28;**

**lv\_gross\_pay lv\_rate % TYPE;**

**lv\_net\_pay lv\_rate % TYPE;**

**BEGIN**

**lv\_hours := &hours;**

**lv\_rate := &rate;**

**lv\_gross\_pay := lv\_hours \* lv\_rate;**

**lv\_net\_pay := lv\_gross\_pay - (lv\_gross\_pay \* lc\_tax);**

**DBMS\_OUTPUT.PUT\_LINE('Hours: ' || lv\_hours);**

**DBMS\_OUTPUT.PUT\_LINE('Pay Rate: ' || to\_char(lv\_rate, '$99.99'));**

**DBMS\_OUTPUT.PUT\_LINE('Gross Pay: ' || to\_char(lv\_gross\_pay, '$9,999.99'));**

**DBMS\_OUTPUT.PUT\_LINE('Tax: ' || to\_char(lv\_gross\_pay \* lc\_tax, '$9,999.99'));**

**DBMS\_OUTPUT.PUT\_LINE('Net Pay: ' || to\_char(lv\_net\_pay, '$9,999.99'));**

**END;Sample output:**

**Hours: 40**

**Pay Rate: $15.00**

**Gross Pay: $600.00**

**Tax: $168.00**

**Net Pay: 432.000**

## Using the Shah student database, find the final mark for student 100 in csid 1104. Display the student's first and last names, the course title and the final mark. ***Output:***

Jose Diaz got A in Accounting

**PL/SQL:**

**DECLARE**

**lv\_mark iu\_registration.FINAL%TYPE;**

**lv\_fname iu\_student.first%TYPE;**

**lv\_lname iu\_student.last%TYPE;**

**lv\_course iu\_course.title%TYPE;**

**BEGIN**

**SELECT r.FINAL, s.first, s.last, c.title**

**into lv\_mark, lv\_fname, lv\_lname, lv\_course**

**FROM iu\_registration r, iu\_student s, iu\_course c, IU\_CRSSECTION cs**

**WHERE r.studentid = 100**

**AND r.csid = 1104**

**AND r.csid = cs.csid**

**AND cs.courseid = c.courseid**

**AND r.studentid = s.studentid;**

**DBMS\_OUTPUT.PUT\_LINE('Student name: ' || lv\_fname ||' '|| lv\_lname);**

**DBMS\_OUTPUT.PUT\_LINE('Class: ' || lv\_course);**

**DBMS\_OUTPUT.PUT\_LINE('Mark: ' || lv\_mark);**

**END;**

**Sample output:**

**Student name: Jose Diaz**

**Class: Accounting**

**Mark: A**

# PL/SQL User-defined Identifiers

***Objectives:***

* Learn the constraints on valid user-defined identifiers in PL/SQL

***To Do:***

1. Read references or research valid Oracle PL/SQL User-defined identifiers
2. Complete the following table:

|  |  |  |
| --- | --- | --- |
| User-Defined Identifier | Valid/Invalid (V/I) | Reason (if invalid) |
| Rate\_of\_pay | V |  |
| 2Number | I | Begins with number |
| END | I | PL/SQL keyword |
| Department number | I | Has a space |
| Dollars$\_and\_cents | V |  |
| SS# | V |  |
| Largest\_yearly\_salary\_paid\_to\_employees | I | Variables must be less than 30 characters |
| TaxRate% | I | PL/SQL operator |
| SS# | V |  |