



eSolutions

---

ICT Volume 3 : Application Standards

## ICT 3.2.1.2-2014 Oracle PL/SQL and SQL Coding Standards

---

### Abstract

This document defines the standards governing Oracle PL/SQL coding

*All rights reserved. No part of this work covered by Deakin University's copyright may be reproduced or copied in any form or by any means (graphic, electronic or mechanical, including photocopying, recording, taping or information retrieval systems) without the written permission of Deakin University.*

### Document Control

<b>Document Title</b>	ICT 3.2.1.2-2014 Oracle PL/SQL and SQL Coding Standards
<b>Version</b>	2014
<b>Controlled Copy (Electronic Reference)</b>	

### Document History

Ver.	Primary Author(s)	Description of Version	Date Completed
0.01	Steven George	Initial draft	26-11-2008
0.02	Shane Wansink	Review / Information security additions	09-01-2008
0.02	Steven George	Standards compliance	25-02-2009
0.03	Steven George	Standards compliance	18-03-2009
0.04	Steven George	Correct numbering error	07-04-2009
0.05	Brett McBride	Changed naming convention for tables and views after feedback	09-04-2009
0.06	Brett McBride	Added ROWNUM usage to SQL section	26-08-2009
0.07	Dan Wickramasinghe	Modified the SQL section and CURSORS section to expand on the use of sql's returning a single row.	03-09-2009
0.08	Brett McBride	Added _mv_src and l_ to naming conventions section	03-11-2010
1.0	Brett McBride	Slight updates to s2, s4.1. Review and release.	09-12-2011
1.1	Michael Heley	Standards Compliance Review	11-09-2014

---

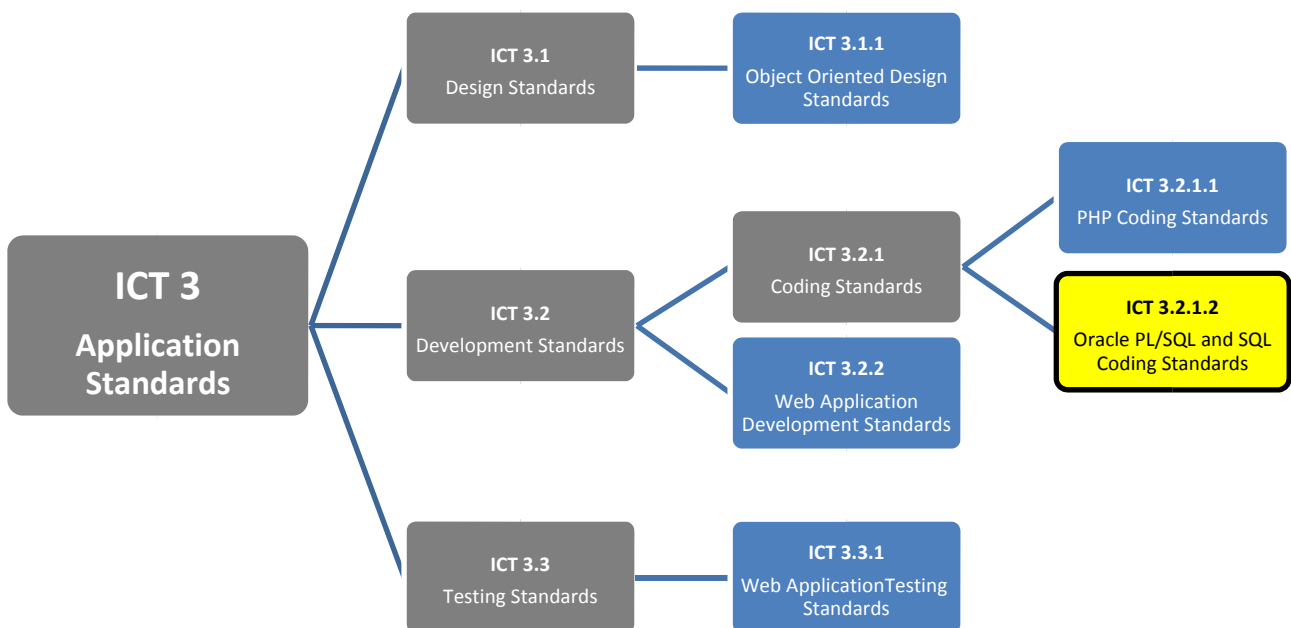
## Table of Contents

<b>1 PL/SQL</b> .....	<b>7</b>
1.1 Packages, procedures and functions .....	7
1.1.1 Packages shall be used where there is a logical grouping of functions and procedures .....	7
1.1.2 Sub-programs shall be used to modularise procedural code.....	7
1.1.3 Sub-programs and cursors shall communicate via parameters and return.....	7
values.....	7
1.1.4 Parameters and variables based on columns or rows shall be declared with.....	7
%TYPE or %ROWTYPE .....	7
1.1.5 GOTO statements are not to be used .....	7
1.1.6 Program units shall only COMMIT data if they represent a controlling transaction. ....	7
1.1.7 Code shall be indented to aide readability. ....	8
1.1.8 Oracle reserved words shall appear in uppercase.....	8
1.1.9 Program units shall be technically code reviewed. ....	8
1.1.10 Data input shall be validated to ensure data is correct and appropriate. ....	8
1.1.11 Internal processing shall be validated to detect and prevent any corruption of information through processing errors .....	8
1.1.12 Data output shall be validated to ensure data is correct and appropriate. This can be done by Unit Tests or otherwise. ....	8
1.1.13 All code shall be wrapped on UAT and Production databases .....	8
1.2 Design .....	8
1.2.1 An audit trail shall be kept on each table .....	8
1.2.2 Database accounts connecting to the database shall do so as a user with the least privileges required to perform the operations required. ....	8
1.2.3 No application shall connect to the database as a schema owner. ....	8
sufficient information to see all versions of each row.....	8
1.2.5 Referential integrity shall be maintained via database constraints .....	8
1.2.6 Database designs shall achieve at least Third Normal Form.....	8
1.2.7 Entity-Relationship diagrams shall be created and maintained for all applications.....	8
1.2.8 Customisations to third party products shall be maintained in a separate schema. ....	8
1.2.9 Tables and indexes shall exist in separate tablespaces. 1.2.10 Logging shall be to a sufficient level to attribute key events within an .....	9
application to a date/time and user, and ascertain success/failure.....	<b>Error! Bookmark not defined.</b>
1.2.11 Errors shall be logged. ....	<b>Error! Bookmark not defined.</b>
1.2.12 All PL/SQL program units shall be stored in source control.....	9
1.2.13 All PL/SQL program units shall contain version information and version history. ....	9

---

1.2.14 Shared code shall be used where it exists.....	9
1.3 SQL.....	9
1.3.1 ANSI-style SQL syntax shall be used .....	9
1.3.2 Column ordering shall be via explicit row names, and not by column position. ....	9
1.3.3 ROWNUM shall not be used in conjunction with equality checking .....	9
1.3.4 SELECT ... INTO can be used if query will return only 1 row at most .....	9
1.4 Naming conventions.....	9
1.4.1 Object naming conventions.....	9
1.4.2 Program unit naming conventions.....	10
<b>2 Cursors .....</b>	<b>10</b>
2.1 Cursor for loops shall not be used to retrieve a single row from a .....	10
query which returns multiple rows .....	10
2.2 Cursors will be fetched into cursor records, and never into multiple variables.....	10
2.3 Select for update shall be used when one or more of the selected rows will be updated .	10
<b>3 Unit testing .....</b>	<b>10</b>
3.1 All program units shall be unit tested .....	10
<b>4 Documentation.....</b>	<b>11</b>
4.1 All PL/SQL program units shall contain a header section. ....	11
<b>5 Error handling .....</b>	<b>11</b>
5.1 All program units shall raise and handle exceptions as necessary. ....	11

## ICT Volume 3 : Application Standards



## **Standards Brief**

This document serves to outline standards that shall apply within Deakin University.

## **Standard Document Access**

All Deakin University staff and authorised/approved contracted personnel are provided access to this document.

## **Policy**

These standards must be used in conjunction with all other referenced standards, and when considered in isolation from the referenced standards may not constitute adequate conformance.

## **Conflict of Information or Clarification**

Whenever a conflict of information occurs or clarification of instruction is required all queries shall be made to Deakin University eSolutions (DeS).

# 1 PL/SQL

The following standards relate to PL/SQL program units.

## 1.1 Packages, procedures and functions

### 1.1.1 Packages shall be used where there is a logical grouping of functions and procedures

### 1.1.2 Sub-programs shall be used to modularise procedural code

Sub-programs allow local re-use of functionality and can make code more readable.

### 1.1.3 Sub-programs and cursors shall communicate via parameters and return values

While it is possible for sub-programs and cursors to reference global variables, this is not the preferred method.

### 1.1.4 Parameters and variables based on columns or rows shall be declared with %TYPE or %ROWTYPE

The use of %TYPE and %ROWTYPE means that program units will remain aligned with changing data definitions.

### 1.1.5 GOTO statements are not to be used

The use of GOTO statements contributes to spaghetti code, and is rarely the best way to write code.

### 1.1.6 Program units shall only COMMIT data if they represent a controlling transaction.

The ability to rollback allows re-testing of a process. A program unit should not contain a commit unless:

- It is part of an autonomous transaction; or
- The commit is an explicitly known and required side-effect of the call

An example of this would be a user-callable function, which you would expect to commit on successful completion – and therefore should explicitly commit on successful completion.

Such a user-callable function may call other utility functions to perform various updates/calculations et cetera, and a commit in such a function may unintentionally commit part of the calling transaction/program unit.

- 1.1.7 Code shall be indented to aide readability.**
- 1.1.8 Oracle reserved words shall appear in uppercase.**
- 1.1.9 Program units shall be technically code reviewed.**
- 1.1.10 Data input shall be validated to ensure data is correct and appropriate.**
- 1.1.11 Internal processing shall be validated to detect and prevent any corruption of information through processing errors**
- 1.1.12 Data output shall be validated to ensure data is correct and appropriate. This can be done by Unit Tests or otherwise.**
- 1.1.13 All code shall be wrapped on UAT and Production databases**

## **1.2 Design**

### **1.2.1 An audit trail shall be kept on each table**

As a minimum, UPDATE\_WHO and UPDATE\_ON shall be used to record the most recent update to each row in a table. The audit details shall be updated via a database trigger.

### **1.2.2 Database accounts connecting to the database shall do so as a user with the least privileges required to perform the operations required.**

### **1.2.3 No application shall connect to the database as a schema owner.**

### **1.2.4 History tables shall be maintained for critical data, and shall contain sufficient information to see all versions of each row.**

This will contain either the entire previous row contents, or changed values as a minimum, as well as the effective dates and times that row was current for, who changed it and what type of operation was performed.

### **1.2.5 Referential integrity shall be maintained via database constraints**

### **1.2.6 Database designs shall achieve at least Third Normal Form**

### **1.2.7 Entity-Relationship diagrams shall be created and maintained for all applications.**

### **1.2.8 Customisations to third party products shall be maintained in a separate schema.**

Where this is impractical then a naming convention shall be used to distinguish customised objects.



### **1.2.9 Tables and indexes shall exist in separate tablespaces.**

### **1.2.10 Logging shall conform to ICT 3.2.1.1-2014 Logging Standards.**

### **1.2.11 All PL/SQL program units shall be stored in source control.**

This includes packages, functions, procedures, triggers.

### **1.2.12 All PL/SQL program units shall contain version information and version history.**

This includes the current version number and an indication that the program unit is locally developed which must appear at the top of the program unit (between “create or replace” and the object name) so that it is not wrapped.

### **1.2.13 Shared code shall be used where it exists.**

## **1.3 SQL**

### **1.3.1 ANSI-style SQL syntax shall be used**

ANSI SQL is a standard which all major databases conform to.

### **1.3.2 Column ordering shall be via explicit row names, and not by column position.**

This ensures that the addition, removal or positional changing of columns does not affect the result ordering.

### **1.3.3 ROWNUM shall not be used in conjunction with equality checking**

ROWNUM is a pseudo-column that is evaluated when all rows are returned. Equality checking (ie “WHERE ROWNUM = n”) may not perform as you expect, or may behave differently in future Oracle versions. Equivalent functionality can be achieved with other operators, eg “WHERE ROWNUM < 2”, “WHERE ROWNUM BETWEEN 5 AND 10”.

### **1.3.4 SELECT ... INTO can be used if query will return only 1 row at most**

If the query can only return one row due to constraints, it may be written as a direct SELECT statement rather than using a cursor. The NO\_DATA\_FOUND and TOO\_MANY\_ROWS exceptions should be catered for always.

## **1.4 Naming conventions**

### **1.4.1 Object naming conventions**

The following naming conventions shall apply:

- “p\_” prefixed to all parameters
- “v\_” prefixed to all variables
- “l\_” prefixed to local variables, functions and procedures within a package/function/procedure
- “gv\_” prefixed to all global variables
- “cst\_” prefixed to all constants

- “r\_” prefixed to all pl/sql records
- “t\_” prefixed to all types
- “c\_” prefixed to all cursors
- “cp\_” prefixed to all cursor parameters
- “e\_” prefixed to all exceptions
- “sp\_” prefixed to all savepoints
- “\_mv” suffixed to all materialized view synonyms
- “\_mv\_src” suffixed to all materialized view sources
- “\_v” suffixed for views
- “\_Q” suffixed for queues
- “\_QT” suffixed for queue tables

#### 1.4.2 Program unit naming conventions

Program units shall use the following naming:

<subsystem><program unit type><descriptive name>

Subsystem shall be a short (3-4 letter) code representing the system or subsystem the program unit belongs to

Program unit type shall be one of P, F, K representing procedures, functions and packages respectively.

Description name shall be short and representative of what the program unit does.

## 2 Cursors

### 2.1 Cursor for loops shall not be used to retrieve a single row from a query which returns multiple rows.

Cursor for loops will only be used when every row in the cursor will be processed. A SELECT ... INTO can be used when only one row is needed.

### 2.2 Cursors will be fetched into cursor records, and never into multiple variables.

### 2.3 Select for update shall be used when one or more of the selected rows will be updated.

## 3 Unit testing

### 3.1 All program units shall be unit tested.

## **4 Documentation**

### **4.1 All PL/SQL program units shall contain a header section.**

This will display purpose, usage information and source control keywords. It need not contain version history, which can be found in source control.

## **5 Error handling**

### **5.1 All program units shall raise and handle exceptions as necessary.**

### **5.2 All user-defined error codes (-20000 to -20999) shall always be unique per program unit.**