

## SQA Advanced Unit specification

### General information for centres

**Unit title:** Software Development: Programming in PL/SQL

**Unit code:** HT0G 48

**Unit purpose:** On completion of the unit the candidate should be able to:

- 1 understand and implement variables, datatypes and advanced datatypes within PL/SQL program blocks.
- 2 understand program structure, using cursors and exceptions handling in PL/SQL.
- 3 create procedures and functions in PL/SQL.
- 4 create packages and triggers in PL/SQL.
- 5 apply and implement advanced data types and dependencies in PL/SQL programs.

This unit is designed to develop a broad knowledge of the concepts, principles, boundaries and scope of software development in a database programming environment using PL/SQL. These will be reinforced by developing practical skills using the structures and features of PL/SQL in the creation of solutions to problems. It forms part of an SQA Advanced Computing Group Award programme, although it can also be used as a stand-alone unit by candidates wishing to acquire and develop programming skills using a database programming language.

**Credit points and level:** 2 SQA Credits at SCQF level 8: (16 SCQF credit points at SCQF level 8\*)

*\*SCQF credit points are used to allocate credit to qualifications in the Scottish Credit and Qualifications Framework (SCQF). Each qualification in the Framework is allocated a number of SCQF credit points at an SCQF level. There are 12 SCQF levels, ranging from National 1 to Doctorates.*

**Recommended prior knowledge and skills:** Access to this unit will be at the discretion of the Centre, however it is recommended that candidates should have achieved the Core Skill of Problem Solving at SCQF level 6 before taking this unit. It is also recommended that candidates have prior experience of using computer systems or studied Database Design and SQL programming language at SQA Advanced level. This may be evidenced by possession of SQA Advanced Unit *SQL: Introduction* (HP2E 47). Alternatively, candidates should have considerable practical work experience and a full appreciation of the role of database design, implementation and programming.

---

## SQA Advanced Unit Specification

**Core Skills:** There are opportunities to develop the Core Skill of *Problem Solving* (SCQF level 6) in this unit, although there is no automatic certification of Core Skills or Core Skills components.

**Context for delivery:** If this unit is delivered as part of a group award, it is recommended that it should be taught and assessed within the subject area of the group award to which it contributes.

**Assessment:** The unit should be assessed by means of a practical exercise and multiple-choice questions to assess practical skills and knowledge and understanding.

The practical exercise will assess the implementation of skills gained throughout the unit. Due to the nature and complexity of the practical exercise it is recommended that it be divided into two parts, the first covering Outcomes 1 to 3 with the second modifying the code produced from part one to assess the knowledge gained from Outcomes 4 and 5, the practical exercise should be open-book and students should be given access to course notes and on line help facilities.

The multiple-choice questions will cover the knowledge and understanding from Outcomes 1 and 2 with a threshold set at 60% and may also be sub divided into two parts if applicable.

## SQA Advanced Unit Specification

### SQA Advanced Unit specification: statement of standards

**Unit title:** Software Development: Programming in PL/SQL

**Unit code:** HT0G 48

The sections of the unit stating the outcomes, knowledge and/or skills, and evidence requirements are mandatory.

Where evidence for outcomes is assessed on a sample basis, the whole of the content listed in the knowledge and/or skills section must be taught and available for assessment. Candidates should not know in advance the items on which they will be assessed and different items should be sampled on each assessment occasion.

Please refer to knowledge and/or skills for the unit and evidence requirements for the unit after the outcomes.

#### Outcome 1

Understand and implement Variables, Datatypes and Advanced Datatypes within PL/SQL program blocks

##### Knowledge and/or Skills

- ◆ Using variables in PL/SQL
- ◆ Using lexical units in PL/SQL
- ◆ Recognising data types
- ◆ Using scalar data types in PL/SQL
- ◆ SQL joins, group functions and subqueries
- ◆ Writing PL/SQL executable statements
- ◆ Nested blocks and variable scope

#### Outcome 2

Understand program structure, using cursors and exceptions handling in PL/SQL

##### Knowledge and/or Skills

- ◆ Retrieving data in PL/SQL
- ◆ Manipulating data in PL/SQL
- ◆ Conditional control: IF statements, case statements
- ◆ Iterative control: basic loops, while and for loops, nested loops
- ◆ Using cursors and exceptions handling
- ◆ Explicit cursors, cursor attributes, cursor FOR loops
- ◆ Cursors with parameters, using cursors for update, using multiple cursors

## **SQA Advanced Unit Specification**

### **Outcome 3**

Create procedures and functions in PL/SQL

#### **Knowledge and/or Skills**

- ◆ Handling exceptions, Oracle server exceptions, user-defined exceptions
- ◆ Creating procedures in PL/SQL
- ◆ Using parameters, passing parameters in PL/SQL
- ◆ Creating and using functions in PL/SQL
- ◆ Using the data dictionary
- ◆ Managing procedures and functions in PL/SQL
- ◆ Understanding object privileges in PL/SQL
- ◆ Understanding user rights in PL/SQL

### **Outcome 4**

Create packages and triggers in PL/SQL

#### **Knowledge and/or Skills**

- ◆ Creating and managing packages
- ◆ Advanced package concepts and package variables
- ◆ Using supplied packages and creating dynamic SQL
- ◆ Creating and managing DML triggers
- ◆ Creating and managing DDL and database event triggers

### **Outcome 5**

Apply and implement advanced data types and dependencies in PL/SQL programs

#### **Knowledge and/or Skills**

- ◆ Using large object data types
- ◆ Managing Bfiles
- ◆ User-defined records
- ◆ Indexing tables of records
- ◆ Understanding dependencies

## SQA Advanced Unit Specification

### Knowledge and/or Skills, and Evidence Requirements for the Unit

The unit should be assessed by means of a practical exercise and multiple-choice questions to assess practical skills and knowledge and understanding.

The practical exercise will assess the implementation of skills gained throughout the unit. Due to the nature and complexity of the practical exercise it is recommended that it be divided into two parts, the first covering Outcomes 1 to 3 with the second modifying the code produced from part one to assess the knowledge gained from Outcomes 4 and 5, the practical exercise should be open-book and students should be given access to course notes and on line help facilities.

The multiple-choice questions will cover the knowledge and understanding from Outcomes 1 and 2 with a threshold set at 60% and may also be sub divided into two parts and delivered on individual occasions if applicable.

Constructed Response — multiple-choice questions Outcomes 1 and 2.

The knowledge and understanding should be measured using multiple-choice questions for Outcomes 1 and 2. Due to the complex nature of these outcomes it is recommended that the assessment of Outcomes 1 and 2 are carried out on individual occasions and not combined.

Breakdown of questions for the constructed response choice should be as follows:

Outcome 1 — Understand and implement variables, data types and advanced data types within PL/SQL program blocks

Using variables in PL/SQL, using lexical units in PL/SQL	<b>5 questions</b>
Recognising data types, using scalar data types in PL/SQL	<b>5 questions</b>
SQL joins, group functions and subqueries	<b>5 questions</b>
Nested Blocks and Variable Scope	<b>5 questions</b>

Outcome 2 — Understand program structure, using cursors and exceptions handling in PL/SQL

Retrieving data in PL/SQL	<b>5 questions</b>
Manipulating data in PL/SQL	<b>5 questions</b>
Understand program structure	<b>5 questions</b>
Using cursors and exceptions handling in PL/SQL	<b>5 questions</b>

### Practical exercise

The achievement in the case study should be measured in two parts covering specific milestones, the first covering Outcomes 1 to 3 with the second modifying the code produced from part one to assess the knowledge gained from Outcomes 4 and 5. Achievement in the case study will be evidenced by the production of a series of structured and commented code modules to a given specification with appropriate datasets produced to show code modules operate to given design specification. The coding must meet the given design specification and use a sample of the features listed below to give a complete business solution, the content items in brackets is suggested examples of content for questions.

### SQA Advanced Unit specification: statement of standards (cont)

**Unit title:** Software Development: Programming in PL/SQL

#### Part one — Outcomes 1 to 3

Create PL/SQL blocks to:

- ◆ insert records to a table (INSERT, VALUES, COMMIT)
- ◆ drop records from a table (DELETE, IF)
  - retrieve data from a table using date functions (CURSOR, SYSDATE, WHERE, AND, BETWEEN, BEGIN, END)
  - insert records to a table with parameters (DECLARE, NVL, MAX, SELECT, FOR, VALUES, COMMIT)
  - insert records to a table with scalar variables (DECLARE, IS, FOR, CURSOR, WHERE, COMMIT)
- ◆ retrieve records to a table with scalar variables (DECLARE, IS, IF, FOR, CURSOR, WHERE, SELECT)
  - insert records to a table with functions (AVG, DECLARE, IS, FOR, CURSOR, WHERE, COMMIT)
  - insert records to a table with a sequence (NEXTVAL, DECLARE, IS, FOR, CURSOR, WHERE, COMMIT)
  - retrieve data from a table with parameters (DECLARE, IS, FOR, CURSOR, WHERE, COMMIT)
- ◆ retrieve data from multiple tables (JOIN, SELECT)
  - convert data types of records in a table (IF, CASE)
  - update records in a table with parameters (DECLARE, INSERT, INTO, VALUES, COMMIT)
  - retrieve data from a table using numerical functions eg (AVG, COUNT, BETWEEN, NULL)

Create procedures/functions to:

- ◆ insert records to a table (INSERT, COMMIT)
- ◆ delete records from a table (DELETE, ROWCOUNT, COMMIT)
- ◆ retrieve data from a table using date functions (SYSDATE, ADD\_MONTHS)
- ◆ insert multiple records to a table depending on the value of a parameter (INSERT, COMMIT, FOR)
- ◆ retrieve data from multiple tables (JOIN, SELECT)
- ◆ covert data types of records in a table depending on a set of conditions (IF,CASE)
- ◆ retrieve data from a table using scalar variables (RETURN, INTEGER, COUNT)
  - retrieve data from a table with parameters (CURSOR, FOR)
  - update records in a table (INSERT, INTO, VALUES, COMMIT)
- ◆ retrieve data from a table using numerical functions eg (AVG, COUNT)

#### Part two — Outcomes 4 to 5

Convert and rewrite PL/SQL blocks created in part one to create a series of procedures/functions. All procedures/functions are to include exception handling where appropriate.

Create packages using pre-written procedures and functions incorporating a number of tasks.

Create a series of Database Triggers.

## SQA Advanced Unit Specification

### Administrative Information

<b>Unit code:</b>	HT0G 48
<b>Unit title:</b>	Software Development: Programming in PL/SQL
<b>Superclass category:</b>	CB
<b>Original date of publication:</b>	August 2017
<b>Version:</b>	01

#### History of changes:

Version	Description of change	Date

**Source:** SQA

© Scottish Qualifications Authority 2008, 2017

This publication may be reproduced in whole or in part for educational purposes provided that no profit is derived from reproduction and that, if reproduced in part, the source is acknowledged.

SQA acknowledges the valuable contribution that Scotland's colleges have made to the development of SQA Advanced qualifications.

For further information, please call SQA's Customer Contact Centre on 44 (0) 141 500 5030 or 0345 279 1000. Alternatively, complete our Centre Feedback Form.

## SQA Advanced Unit Specification

### SQA Advanced Unit specification: support notes

#### Unit title: Software Development: Programming in PL/SQL

This part of the unit specification is offered as guidance. The support notes are not mandatory.

While the exact time allocated to this unit is at the discretion of the centre, the notional design length is 80 hours.

#### Guidance on the content and context for this unit

This unit is intended as an introduction to the principles and procedures involved in developing robust, reliable, efficient and maintainable PL/SQL as an extension to the Oracle DBMS environment. Its aim is to acquire competence in the development of software solutions using the program language syntax and constructs required to reinforce software development. The concepts of good practice, such as internal documentation, should be stressed throughout. The unit should provide a foundation of good PL/SQL programming skills.

It forms part of an SQA Advanced Computing Group Award programme and should be delivered within the context of the Group Award. It can also be delivered as a stand-alone unit by candidates wishing to acquire, and develop, programming skills using an object oriented database programming language.

As a unit in PL/SQL programming on an SQA Advanced Diploma, many of the component parts of the context and examples used should be provided to the candidate, eg tables, synonyms, sequences as deemed appropriate, this would permit the candidate to concentrate more on the programming techniques involved, rather than designing and implementing the initial structure of the problems used.

By the end of the unit, the candidate should have achieved a good foundation in the skills required for developing reliable, robust and efficient program designs solve business problems using an object oriented database programming language.

It is likely that the unit will be delivered in the second year of a full-time SQA Advanced Computing Diploma but where the candidates have the recommended prior knowledge and skills could be delivered at any stage of a course.

#### Guidance on the delivery and assessment of this Unit

For assessment purposes, the candidate will be issued with a design for implementation. The assessor must ensure that candidate evidence is being produced at SCQF level 8. The implementation of the design requires the candidate to consider and select **in each** of the following areas:

- ◆ create PL/SQL blocks
- ◆ convert and rewrite PL/SQL blocks to procedures/functions all procedures/functions are to include exception handling where appropriate
- ◆ create packages using pre-written procedures and functions incorporating a number of tasks
- ◆ create a series of Database Triggers

The principles of good practice, such as internal documentation and code indentation, should be included throughout.



## SQA Advanced Unit Specification

After introducing the steps involved in developing a solution to typical problems, candidates should be presented with a series of practical exercises to illustrate the features of the programming language as they are introduced.

Regardless of delivery mode, the use of either a single and coherent case study approach divided into two sections or a two case study approach along with a multiple-choice test to cover knowledge and understanding is recommended for assessment, with the model broken down into identifiable and assessable stages allowing the tutor the opportunity for regular monitoring of the candidate's progress and allowing time for intermittent remediation.

Testing should involve the candidate comparing the expected and the actual results, evaluating the differences and amending the code as necessary. For testing purposes the bank of test data may be supplied by the assessor.

During the learning process, part-completed code modules may be offered to the candidate for completion of the key syntax and language features.

### *Opportunities for developing Core Skills*

There are opportunities to develop the Core Skill of *Problem Solving* (SCQF level 6) in this unit, although there is no automatic certification of Core Skills or Core Skills components. The complex nature of the project will give scope for candidates to develop their problem solving skills.

## Open learning

If this unit is delivered by open or distance learning methods, additional planning and resources may be required for candidate support, assessment and quality assurance.

## Equality and inclusion

This unit specification has been designed to ensure that there are no unnecessary barriers to learning or assessment. The individual needs of learners should be taken into account when planning learning experiences, selecting assessment methods or considering alternative evidence.

Further advice can be found on our website [www.sqa.org.uk/assessmentarrangements](http://www.sqa.org.uk/assessmentarrangements).

### General information for candidates

#### Unit title: Software Development: Programming in PL/SQL

This unit is designed to enable you to develop a broad knowledge of the concepts, principles, boundaries and scope of software development using PL/SQL as an extension to Oracle DBMS. These will be reinforced by developing the practical skills required in using the structures and features of PL/SQL in the creation of solutions to problems. The unit is primarily intended for people who propose to follow careers in database administration and programming where an intensive knowledge of databases is required.

On completion of the unit you should be able to:

- ◆ understand and implement Variables, Datatypes and Advanced Datatypes within PL/SQL program blocks.
- ◆ understand Program Structure, Using Cursors and Exceptions Handling in PL/SQL.
- ◆ create procedures and functions in PL/SQL.
- ◆ create Packages and Triggers in PL/SQL.
- ◆ apply and implement advanced data types and dependencies in PL/SQL programs.

Outcome 1 requires you to develop solutions in PL/SQL in order to understand and use Variables, Datatypes and Advanced Datatypes in PL/SQL. The topics covered include PL/SQL Blocks, SELECT Statements and Single-Row Functions, Variables and Lexical Units, Data Types and creating PL/SQL group functions and Subqueries.

Outcome 2 introduces Program Structure in PL/SQL and covers the use of Cursors and Exceptions Handling in PL/SQL. More problem solving skills are developed. The topics covered include Data Manipulation Language, Retrieving Data in PL/SQL, Manipulating Data in PL/SQL, Using Transaction Control Statements. You will be introduced to Conditional Control using IF Statements and Case Statements, Iterative Control using While, For Loops and Nested Loops.

Outcome 3 develops your PL/SQL skills and introduces exception handling including Server Exceptions and User-Defined Exceptions. You will create Procedures in PL/SQL, Use Parameters and Pass Parameters. Functions and Using Functions in SQL will be introduced with a series of practical exercises along with developing object privileges and user right issues.

In Outcome 4 you will develop advanced PL/SQL Packages and Triggers using already defined statements. In this Outcome the topics covered will include Creating and Managing Packages, Creating and Managing DML Triggers and Creating and Managing DDL and Database Event Triggers.

Outcome 5 further develops your skills in PL/SQL and introduces advanced topics such as advanced data types and dependencies in PL/SQL. Practical examples will be used to Managing Bfiles, develop User-Defined Records, Index Tables of Records and you will develop dependencies.

The unit should be assessed by means of a case study and multiple-choice exams to assess practical skills and knowledge and understanding.

## **SQA Advanced Unit Specification**

The case study will assess the implementation of practical skills gained throughout the unit. Due to the nature and complexity of the case study it is recommended that it be divided into two parts, the first covering Outcomes 1 to 3 with the second modifying the code produced from part one to assess the knowledge gained from Outcomes 4 and 5, the case study should be open-book and students should be given access to course notes and on line help facilities.

The multiple-choice exam will cover the knowledge and understanding learned in Outcomes 1 and 2 with a threshold set at 60% and may also be sub divided into two parts if applicable.

You will be allowed access to your course notes and online help facilities.

Your assessor will agree a date for submission of the project.