Proc SQL & Macros (Bundle) – the Language of SAS

Please read our [Terms and Conditions](#) and our [Privacy Policy](#)

Duration: 30 – 32 hours (plus additional time required for exercises and study)

Learning Overview:

Our Proc SQL and Macros courses have been bundled together to help maximise the Learner's time and minimise costs.   It is an ideal course for those looking to add to their professional knowledge of the Language of SAS.

The learning is delivered using blended learning techniques via our digital learning platform.

There are mini quizzes and comprehensive exercises throughout to help assess and reinforce your learning.

Learning Outcomes:

By the end of this course you will be able to:

- Describe the language elements of SQL
- Write SQL queries to summarise data table, generate reports and new data tables
- Manage tables and indexes using SQL
- Utilise macro variables and functions to make your programs smarter
- Create and store macro programs to simplify repetitive tasks
- Implement conditional iterative macro processing to help you generate dynamic, data driven code in the Language of SAS

Delivery Schedule:

On day 1 we invite join a short online welcome to meet your instructor, introduce you to our Digital Learning platform and get you started on the PROC SQL eLearning modules of this course.

You will need 2 hours to complete the 3 eLearning modules with additional time required for exercises.

We ask you to complete these modules before joining us for the online masterclass on the afternoon of day 1. This is followed by 3 x live online classes to complete your PROC SQL learning.

In the afternoon of day 3 we introduce you to the MACROS eLearning modules. You will need 2.5 hours to complete the 3 eLearning modules with additional time required for exercises.

We ask you to complete these modules before joining us for the online masterclass on day 4. This is followed by 3 x live online classes to complete your MACROs learning.

Each day finishes with an opportunity for you to join an open Q&A session with the instructor if you need any additional support.

	09:00 - 12:00	13:00 - 17:00
Day 1	Welcome and Introduction to PROC-SQL eLearning modules	Live Online Masterclass to consolidate PROC SQL elearning
Day 2	PROC SQL (Live Online Class 1)	PROC SQL (Live Online Class 2)
Day 3	PROC SQL (Live Online Class 3)	MACROS eLearning modules
Day 4	Live Online Masterclass to consolidate MACROS elearning	MACROS (Live Online Class 1)
Day 5	MACROS (Online Class 2)	MACROS (Live Online Class 3)

Pre-requisites:

To get the most out of this course, it is expected that you should have attended our Fundamentals and Intermediate courses in the language of SAS (or comparable courses) or have completed nine months of developing code in the language of SAS including topics:

- Use of Global statements e.g. LIBNAME, FILENAME and OPTIONS
- Use of functions to transform character data (e.g. SCAN, FIND, SUBSTR)
- Use of functions to transform numeric data (e.g. CEIL, INT, FLOOR, ROUND)
- Use of functions to transform dates (e.g. DAY, MONTH, YEAR, INTCK)
- KEEP and DROP statements and dataset options to select variables
- IF and WHERE statements to select observations
- Conditional Processing with IF-THEN-ELSE
- Iterative DO Loops
- Conditional Do Loops – DO WHILE and DO UNTIL
- Compile-Time statements (e.g. LENGTH, RETAIN)
- Combining Data Sets Vertically and Horizontally
- Summarising data with the FREQ, MEANS and TABULATE procedures

For the hands-on practice activities in the course, you will need access to an environment that runs the programming Language of SAS. On our courses, we signpost you to some of the free tools available.

Check out the link below to review system requirements:

- [SAS® OnDemand for Academics](#)

Learning Modules:

PROC SQL

Introduction to SQL Language Elements (eLearning)

Learning Objective: Explain how to implement SQL within the language of SAS;

- Origins of Structured Query Language (SQL)
- Implementing SQL within the Language of SAS
- SQL Language Elements

Selecting COLUMNS & ROWS (eLearning)

Learning Objective: Describe how to subset data to select columns and rows.

- Selecting Columns
- The DESCRIBE and FEEDBACK options
- Create Columns
- The CASE Expression
- Selecting Rows
- Special WHERE Operators

Summarising Data (eLearning)

Learning Objective: Explain how to summarise and classify data

- Producing Summary Statistics
- Columns created by a query
- Classifying Data – GROUP BY
- Summary Statistics by GroupSelecting Summarised Data - HAVING

Report Presentation (live online class)

Learning Objective: Describe ways to improve the presentation of data.

- Formats and Labels
- Ordering Data
- Summarising Data

Sub Queries (live online class)

Learning Objective: Describe the different types of sub-queries used in the programming Language of SAS

- Non-Correlated Subquery
- Correlated Subquery
- Substituting Values
- Returning Multiple Values

Creating Data Tables (live online class)

Learning Objective: Explain PROC-SQL techniques used to create a Data Table

- Create a Data Table form the Results of a Query
- Create a Data Table from a Definition
- Create a Data View from the Results of a Query
- Data Views versus Data Tables

Joining Data Tables Horizontally (live online class)

Learning Objective: Explain PROC-SQL techniques used to join Data Tables horizontally

- Inner Joins
- Outer Joins
- Full Outer Joins
- The COALESCE Statement
- Table Aliases
- In-Line Views

Joining Data Tables Vertically (live online class)

Learning Objective: Explain the PRCO SQL techniques used to join Data Tables vertically

- Methods of Combing Data
- Modifiers
- Except Rule
- Intersect Rule
- Union Rule
- Outer-Union Rule

Managing Tables (live online class)

Learning Objective: Describe how to manage tables using PROC SQL

- Add, Update and Delete data rows
- Add, Modify and Delete data columns
- Delete a Table, View or Index
- Explain what an Index is
- Create and use an Index

Macros

Introduction to Macro processing (eLearning)

Learning Objective: Explain basic Macro Processing in the Language of SAS.

- Compilation and Execution phases
- Tokenisation
- Macro Triggers
- The %PUT Statement
- Displaying macro variable values

Introduction to Macro variables (eLearning)

Learning Objective: Describe six different features of Macro Variables.

- Macro Variable Storage
- Macro Variable Resolution
- User-Defined Macro Variables
- The %LET Statement
- The SYMBOLGEN Option
- Delimiting Macro Variables

Macro Functions (eLearning)

Learning Objective: Identify three different types of Macro Functions.

- Character Functions – review
- Using Macro Functions
- Macro Functions for Numbers
- The %SYSFUNC and other functions including: %INDEX, %SCAN, %SUBSTR, %UPCASE, %BQUOTE, %STR, %NRBQUOTE, %NRSTR, %EVAL and % SYSEVALF.

Macro Programs (live online class)

Learning Objective: *Create and store macro programs to simplify repetitive tasks*

- Compilation
- Macro Calls
- Generating Code
- The MPRINT Option
- Storage
- Autocall Macros
- Best Practices
- Macro Program Parameters

Global and Local Symbol Tables (live online class)

Learning Objective: Utilise Macro Variables on the Global and Local Symbol Tables

- The Global Symbol Table
- The Local Symbol Table
- Nested Macros
- Creating and Updating Macro Variables
- Resolving Macro Variables

DATA Step and PROC SQL (live online class)

Learning Objective: Creating Macro Variables in DATA Step and Proc SQL

- Creating Macro Variables in DATA Step
- Compare CALL SYMPUTX and %LET
- Indirect Macro Variable References
- Create Macro Variables in PROC SQL
- Store multiple values in a Macro Variable

Conditional Processing (live online class)

Learning Objective: Implement Conditional Macro Processing

- %IF -%THEN - %ELSE
- The MLOGIC Option
- Conditionally Returning a Program Step
- Conditionally Returning a Statement
- Conditionally Returning a Partial Statement
- %DO - %END
- The &INCLUDE Statement

Iterative Processing (live online class)

*Learning Objective: G*enerate dynamic, data driven code - using Conditional Iterative Processing

- Iterative Processing in a Macro Program
- Generate Program Code Iteratively – Dynamic Code
- Perform Conditional Iterative Processing