

I'm not a robot   
reCAPTCHA

Continue

## Sas certification prep guide 9.4

Authentication is always a hot topic on social media channels. The SAS Certification Community site offers a lot of advice on basic programming using SAS 9.4 certification: SAS Certified Professionals, for candidates preparing for the most popular exams. Here's a summary of the top recommendations to help you prepare for the exam. Do you have any additional advice to add to the list? Leave a comment below. Also, if you're not sure which SAS programming credentials are right for you, we'll use a comparison chart to guide you. (Editor's note: Originally published in 2013, this post was updated in April 2020 to include information about basic programming expert certification and other new resources.) Tag Certification is a sas programming sas lab (created by) priced at \$114.95 publisher SAS Labs on Publish Date October 18, 2019 page 430D 8.27 X 11.02 x 0.87 inches | The 2.12 pound language English type paperback EAN/UPC 9781642954678 SAS has the perfect choice of books and electronics to help you get the most out of your SAS software. Books and products cover a variety of topics, all developed and reviewed by technical experts. SAS meets the information needs of new and experienced users through a variety of transfer methods including PDF, HTML, electronic and hard copybooks. Partnerships with authors, other publishers and distributors allow us to use a wide range of products from a variety of sources to meet the needs of users around the world. Employee Favorites List (10 books) The correct pre-statement citation for this manual is as follows: SAS Labs Inc. 2019. SAS® Certified Professional Preparation Guide: Advanced Programming with SAS® 9.4 Cary, NC: SAS Institute Inc. SAS® Certified Professional Preparation Guide: SAS® 9.4 Copyright © 2019, SAS Institute, Cary, NC, US ISBN 978-1-64295-467-8 (paperback) ISBN 978-1-64295-469-2 (Epub) ISBN 978-1-64295-470-8 (Moby) produced in the United States. For hard copy books: No part of this publication may be copied, stored or transmitted to the search system in any form or in any way, electronic, mechanical, copy or other way without the prior written permission of the publisher SAS Institute Inc. For web downloads or e-books: The use of this publication is subject to the terms and conditions set by the supplier when acquiring this publication. The scanning, uploading and distribution of this book via the Internet or other means without the publisher's permission is illegal and punishable by law. Buy only authorized electronic editions Do not participate in or encourage electronic piracy of copyrighted material. Your support for the rights of others is appreciated. U.S. government license rights: Limit The software and its documentation are commercial computer software developed at a personal cost and are provided to the U.S. government with limited rights. If the U.S. government uses, replicates, or discloses the Software, if applicable, FAR 12.212, DFAR 227.7202-3(a), DFAR 227.7202-3(a), DFAR 227.7202-4, To the extent required under U.S. federal law, the minimum limiting authority is set by FAR 52.227-19 (December 207). If FAR 52.227-19 applies, this section shall act as a notice under that section (c) and does not require any other notice to be attached to the software or documents. The government's rights in the Software and Documents are only the rights set forth in this Agreement. SAS Institute Inc., SAS Campus Drive, Cary, NC 27513-2414 October 2019 SAS® and all other SAS Institute Inc. The name of the product or service is a registered trademark or trademark of SAS Institute Inc. in the United States and other countries. © U.S. registration. Other brand and product names are trademarks of their respective companies. P1:Certification Last Updated: October 16, 2019 SAS® Certified Professional Preparation Guide: SAS® 9.4 Title Page Using The Sample Data Accessibility Feature in the Readiness Guide for Advanced Programming Exam Parts using Copyright 1: SAS Chapter SQL Processing 1: PROC SQL Basic PROC SQL Basic Provisions in Clause Group BY TO Clause Query Quiz End Note Chapter 2 : Create table statements using table creation and management AS keywords using like clauses and insert management statements pre-table chapter Quiz Chapter 3: Combine tables using PROC SQL understanding create Cartesian products using internal joins using SQL join, compare SQL joins, and merge data step matching quiz chapter 4: SET Operators understanding set operators using setup operators external union set operator quiz chapter 5 : Advanced SQL Technology FedSQL Procedure Quiz 2: SAS Macro Language Handling Chapter 7: Using macro variables to introduce macro variables Using macro variables Using MACROS VARIABLES Using PROC SQL Quiz Chapter 6 to create and manage views, sub-query sub-queries using sub-queries, use sas macro functions to store and process character processing text using SAS macro features to store and process character processing text using SAS variables Manipulate strings. The Macros feature provides the ability to mask special characters that create macro variables during the execution of proc SQL steps that create macro variables during the execution of data steps that indirectly reference macro variables: Macro program actions that use parameters to control macros to define macro forwarding information as macros, and invoke macro program actions 10: Understanding external files Macro definition processing Quiz Chapter 10 : Advanced macro technology macro functions that store macro definitions in external files as macro-understanding macro calls to collect macro-calling data 3 Part 11: Advanced SAS Programming Technology Chapter 11: A one-dimensional array that defines and references an array that extends and references the use of one-dimensional arrays: Using hash objects that define hash objects to find hash objects, use hash objects to create hash objects using Quiz Chapter 3: Using sas utility procedures to create photo formats in a format that creates a format with PROC FCMP quiz chapter 14 . : SAS utility procedure to create picture format using advanced features using various advanced features to perform pattern matching with Perl regular Expression Quiz Part 4: Practice programming scenario difference between workbook and certification exam scenario 1 scenario 2 scenario 3 scenario 4 scenario 5 scenario difference 6 scenario 7 scenario 8 scenario 9 scenario 10 part 5: solution chapter quiz answer key chapter 1: PROC SQL basic chapter 2: Table Creation and Management Chapter 3: PROC SQL Chapter 4: SET Operators Chapter 5: Asubqueries Chapter 5: Using Advanced SQL Technology Chapter 7: Creating and Using Macro Variable Chapter 9: Storing and Processing Text Chapter 9: Working with Macro Programs Chapter 10: Advanced Macro Technology Chapter 11: Defining and Processing Arrays Chapter 12: Hash Object Chapter Usage Data Processing Chapter 13: Using SAS Utility Procedure Chapter 14: Using Advanced Features Chapter 17: Programming Scenario 1 Scenario 2 Scenario 4 Scenario 5 Scenario 8 Scenario 9 Scenario 10 We recommend that you use the recommended read index additional resources and set up practice data for SAS on demand: navigation. Click the Professional Preparation Guide directory. Open the cre8data.sas program and click Raw. Right-click anywhere on the screen and save the Cre8data.sas program to select a location where you can access sas. Open the cre8premdata.sas program and click Raw. Right-click anywhere on the screen and select Save to save the cre8premdata.sas program the same. With cre8data.sas, Log on to SAS On Demand for Academics and open SAS Studio. Right-click the File (Home) folder, and then select New -> Folder. Specify a name for the new folder ceradv, and then click Save. Right-click the ceradv folder and select Upload File. Go to Select File (click File and go to the ceradv folder within the Practice Data folder on your local computer). Select all program files and click Open. All available programs are listed under File Selection. Click Upload. Open and edit the cre8data.sas program. In the path macro variable, replace /folder/myfolders with a path to the ceradv folder. In SAS Studio, right-click the ceradv folder, and then click Properties. Copy the path of the location box and paste it into the %LET statement. Save, and then run the cre8data.sas program. The walkthruing data is now created and ready to use. Note: When you end a SAS session, the path macro variables in the cre8data.sas program are reset. If you do not want to run cre8data.sas again each time, run the libname.sas program in the sample data folder to restore the library. To complete the example in this book, you must have access to the SAS window environment, SAS Enterprise Guide, or SAS Studio (or SAS University Edition) by setting up practice data in a different SAS environment. To access the sample data file and create walkthruing data, the sample. Click the Professional Preparation Guide directory. Open the cre8data.sas program and click Raw. Right-click anywhere on the screen and save the Cre8data.sas program to select a location where you can access sas. We recommend that you create a new folder named Ceradv in a location where you can access the SAS and save the cre8data.sas program in the Ceradv folder. The Libref associated with this book use seradv as the Libref name. Open the cre8premdata.sas program and click Raw. Right-click on everything on the screen and select Save to save the cre8premdata.sas program in the same location as cre8data.sas. Open the cre8data.sas program in the selected SAS environment. In the SAS window environment: File ->, and then go to the Ceradv folder in the Walkthruing Data folder. In SAS Studio: Navigation Pane, expand Files and Folders, and then navigate to the Ceradv folder within the Practice Data folder. In THE SAS ENTERPRISE GUIDE: IN THE LIST OF SERVERS, EXPAND → SERVER, → FILE, AND THEN GO TO THE CERDADV FOLDER IN THE WALKTHRUING DATA FOLDER. In the path macro variable, save the /folder/myfolders practice data and replace the program %let path=/folders/myfolders/mymy-folder-name with a path to the folder that runs it. Important: The location specified for the path macro variable and the location of the downloaded SAS program must be the same location. Otherwise, the cre8data.sas program will not be able to create a walkthruing. The walkthruing data is now created and ready to use. Note: When you end a SAS session, the path macro variables in the cre8data.sas program are reset. If you do not want to run cre8data.sas again each time, run the libname.sas program in the sample data folder to restore the library. Last Update: October 16, 2019 Preparation Guide Overview SAS Certified Professional Preparation Guide: Advanced programming using SAS 9.4 is a test preparation document that uses the following environments and products: SAS Window Environment SAS Enterprise Guide or SAS University Edition Accessibility Documentation Help. The following table lists accessibility information for products listed: Document Format Contact accessibility@sas.com. Final update: To complete the examples in this book for exam requirements and details requirements on October 16, 2019, you must have access to the SAS window environment, SAS Enterprise Guide, SAS Studio, or SAS University Edition. Exam objectives and updates for this book Current exam objectives and a list of updates to this book www.sas.com/certify available in the following articles: Exam objectives are subject to change. Sas and Pearson VUE allow you to purchase practice exam exams. For more information about practice exams, see exam registration ww.sas.com/certify SAS global certification website to enroll in SAS 9.4 advanced programming performance-based exams. In this book, sas syntax, an additional resource for learning SAS programming syntax rules, is the following: Data Output-SAS Dataset (DROP=Variable)= KEEP=Variable(s); SAS dataset settings; BY variable(s); Run; The following rules are used in the example: DATA, DROP=, KEEP=, SET, BY, and RUN are bold capital letters because they must be spelled as shown. Output-SAS datasets, variables, SAS datasets, and options are in italics because they represent the values they each provide. Because it is an optional syntax, it is enclosed in angular brackets. DROP= and KEEP= are vertical bars () to indicate that it is mutually all-time. The example syntax shown in this book contains only what you need to know to prepare for the certification exam. For the full syntax, see the appropriate SAS reference guide. Last Updated: October 16, 2019 Chapter 1: PROC SQL Fundamentals PROC SQL Basic Provisions in Group BY Clause Retention Clause BY Clause WHAT Section Section WHAT Section BY Options Verification Query Syntax Quiz End Notes Last Updated: October 16, 2019 WHAT IS PROC SQL Basics? PROC SQL is a SAS implementation of a structured query language (SQL). SQL is A language that is widely used to retrieve and update data in a table and in a view based on that table. The following table compares the terms used for data processing, SAS, and SQL. This book use all these terms. PROC SQL can often be used as an alternative to other SAS procedures or DATA steps. Use PROC SQL for the following actions: When you retrieve data from a SAS table and manipulate it to add, modify, or delete data values from the table, the table and the view join multiple tables (if they contain columns with the same name), and the report PROC SQL Syntax APROC SQL statement starts the SQL procedure. You can use multiple statements in the PROC SQL step. Each statement defines a process and runs immediately. Each statement must end with a semicolon. The SQL procedure ends with the QUIT statement. Last Updated: October 16, 2019 PROC SQL SELECT Statement Brief Overview Select statement retrieves and displays data. It consists of a SELECT clause and several optional clauses that are available within the SELECT statement. Each clause begins with a keyword followed by one or more components. Optional clauses name input datasets, subsets, groups, or data sorts. A PROC SQL step that contains one or more SELECT statements is called a PROC SQL query. A SELECT statement is just one of several statements available in PROC SQL. The Select Statement Syntax Selection statement is the default tool for PROC SQL. You can use select statements to identify, manipulate, and retrieve columns of data from one or more tables and views. The SELECT statement must contain the SELECT clause and the FROM clause, both of which are required for PROC SQL queries. When you create a SELECT statement, you must specify the clauses in the following order: The FROM clause selects one or more source tables or views. The WHERE clause allows you to filter the data. The GROUP BY clause allows you to process data as a group. The HAVING clause works with the GROUP BY clause to filter the star-by-group results by group. The ORDER BY clause specifies a book of orders. Example: To select a column, create a SELECT clause to specify which columns to display in the query. After the keyword SELECT, list one or more column names and separate the column names into commas. The SELECT clause allows you to specify an existing column and create a column. Existing columns are already stored in the table. The following SELECT clauses specify column EmpID, work code, salary, and bonus. Column EmpID, work code, and payroll are existing columns. A column called Bonus is a new column. The column alias appears as a title in the output and matches the case used in the SELECT clause. proc SQL; Select non-mpid, job code, salary, salary \*.06 as an bonus. Where salary<32000 orders by work code; Termination; Output 1.1 PROC SQL query result example: Use an asterisk (\*) in the SELECT \* SELECT clause to display all columns in the order stored in the table. All rows are displayed by default unless restricted or sorted. The following SELECT statement displays all columns and rows in the table Cerdadv.Staffchange. Cerdadv.Staffchange lists all employees of the company who have changed their employment status. proc SQL; \* Selection from ceradv.staffchange; Termination; Output 1.2 SELECT SQL Query Results \* Example: When specifying SELECT\*, you can use the Feedback option to write an expanded list of columns to the SAS log using the Feedback option in the PROC SQL statement. For example, the PROC SQL query shown below contains feedback options. \* Selection from ceradv.staffchange; Termination; The following is written to the SAS log: Log 1.1 SAS Log Notes: Statements are converted as follows: Empid, staff changes. Castle, staff change. Name, staff change. City, staff change. Status, staff changes. CERDADV's phone number. Employee changes; The Feedback option is a debugging tool that allows you to see exactly what is being submitted to the SQL processor. In the SAS log, the resulting message extends the asterisk (\*) to a column list. It also solves macro variables and places parentheses around expressions to display the evaluation order. Example: When you create a new column, you can create a new column that contains text or calculations. The new column appears in the output along with the existing column selected. The new column exists only for the duration of the query, unless you create a table or view. To create a new column, include a valid SAS expression in the SELECT clause column list. You can assign column aliases and names to new columns using the AS keyword and the name you want to use. Note: Column aliases must follow the rules for SAS names. The following example uses an expression to calculate a new column bonus: The value in the new column is multiplied by the salary of 0.06. The AS keyword is used to assign column alias bonuses to new columns. proc SQL; Payroll<32000 Order:empid, staffchange, salary, salary\*.06 as the bonus for payroll<32000;payrollmasters based on this job code. Termination; Output 1.3 PROC SQL query results: New column – Bonus column A alias is useful because it can reference columns elsewhere in the query. The column alias appears as a title in the output and matches the case used in the SELECT clause. In the SELECT clause, you can label an existing column or a new column. If you assign both a label and a column alias to a new column, the label The column in output[1]. If only column aliases are specified, specify exactly what you want the column alias to appear in the output. Example: If you remove duplicate rows from the output, you can use separate keywords to remove duplicate rows. Distinct keywords apply only to all columns listed in the SELECT clause and their columns. proc SQL; Select the flight number and destination from ceradv.internationalflights. Termination; Note: Distinct keywords are the same as UNIQUE. Unique keywords are the same as distinctions, but not an ANSI standard. Output 1.4 PROC SQL Query Results: Unique Value Last Updated: After you create the SELECT clause on October 16, 2019, write the clause syntax for the FROM clause, and then use the FROM clause to specify the table or view you want to contact. Type a keyword , and then type a name for the table. Example: If you query a single table using the FROM clause, you can query a persistent SAS table called a payroll master that is stored in the Ceradv library. The FROM clause specifies that the payrollmaster be queried. proc SQL; Payroll<32000 Orders By this job code,<32000 Ordercertadv.As a bonus to the Payroll Master, select Nonmpid, Work Code, Salary, Salary\*.06. Termination; Last updated: October 16, 2019 WHERE clause Brief overview The WHERE clause allows you to sub-set data based on the conditions that each row in the table must meet. The PROC SQL output contains only rows that meet the criteria. The WHERE clause is used within the SELECT statement in the PROC SQL step. The expression in the WHERE clause can be a valid SAS expression. In the WHERE clause, you can specify the columns in the base table specified in the FROM clause. The columns specified in the WHERE clause do not need to be assigned to the SELECT clause. Where clause syntax selection and FROM clauses after where clauses must be made. Example: In the following PROC SQL query, using the WHERE clause, the WHERE clause selects rows with a payroll value of less than \$32,000. proc SQL; Payroll<32000 Orders By this job code,<32000 Ordercertadv.As a bonus to the Payroll Master, select Nonmpid, Work Code, Salary, Salary\*.06. Termination; Output 1.5 PROC SQL query results: Using calculated values that understand how to understand proc SQL process calculated columns, a subset of payrolls with salaries below \$32,000 showed how to define a new column by using the SELECT clause and performing calculations. The following PROC SQL query adds three existing column values: Board, Transport, and Non-Revenue to create a new column total. proc SQL outobs=10; Flight numbers, dates, destinations, boarding + transfers + transfers + non-revenues by total from stadv.marchlights; Termination; You can also sub-set rows using columns calculated in the WHERE clause. However Of the way SQL queries are handled, you cannot assign only column aliases to the WHERE clause. To see what happens, take the preceding PROC SQL query and add a WHERE clause to the SELECT statement to reference the calculated column total. proc SQL outobs=10; Total number of flights, dates, destinations, boarding + transfers + total 100 &; non-revenue from certadv.marchlights to guns; Termination; When this query is run, the SAS log displays the following error message: ERROR: The following columns were not found in the attribution table: This error message is generated because the WHERE clause is processed before the SELECT clause in the SQL query. The SQL processor finds a table for each column named in the WHERE clause. Sas generates an error message because the table Cerdadv.Marchlights does not contain a column called Total. Note: To avoid error messages, you should use calculated keywords. Example: Use column aliases and calculated keywords in the WHERE clause to refer to values calculated using values calculated in the WHERE clause. The calculated keyword tells PROC SQL that the value is calculated within the query, proc SQL outobs=10; In ceradv.marchlights, calculated & total 100, select The number of flights, date, destination, boarding + transfer + non-revenue + non-revenue. Termination; Output 1.6 PROC SQL query results: Use calculated keywords Note: Repeat the calculation in the WHERE clause instead of using calculated keywords. However, this method is inefficient because PROC SQL must perform the calculation twice. An ANSI method that calculates the WHERE clause again. In the preceding query, the alternate WHERE statement will be: Boarding + Transfer + Non-Revenue &:100; Example: You can also use calculated keywords in other parts of a query by using the calculated values in the SELECT clause. proc SQL outobs=10; Flight number, date, destination, boarding + transfer + non-revenue by total, total calculated in half at ceradv.marchlights 12; Termination; This query generates the following output: Output 1.7 PROC SQL query results: You can use a subset of calculated keywords using conditional operator overview to specify valid SAS expressions to sub-set or limit the data displayed in the output by specifying a valid SAS expression. Expressions can include different types of operators, such as: In PROC SQL comparisons, operator usage, logic, and connection operators are used in PROC SQL in the same way that they are used by other SAS. The following conditional operators are also available in PROC SQL queries: All of these operators can be used for other SAS procedures. Tip To create negative conditions, you can have the NOT operator before these conditional operators. To select within a range, use the BETWEEN-operator to select rows based on a range of numeric or character values from the BETWEEN-AND operator in the WHERE clause. BETWEEN-and operators are included. In addition to the values that occur between limits, the query results include values that you specify as limits for a range of values. Note: When you specify a limit on a range of values, you do not need to specify a smaller value first. Here are some examples of where clauses: To select a string from an embedded operator, the Include or Question Mark (?) operator is typically used to select a row that contains a specific string in a character column. These operators are interchangeable. Note: The Include or Question Mark (?) operator is not part of the ANSI standard. SAS enhancement. The following PROC SQL query uses Embedding to select a row that contains the string ER in the Name column. As the output shows, the row containing all the ER within the Name column is displayed. proc SQL; Select the name of the commercial customer that contains 'ER' in the name. Termination; Output 1.8 PROC SQL query results: To select a value from the IN operator, use the IN operator (List) to select only rows that match one of the values in the fixed value list of numbers or characters. Here's an example of the WHERE clause that contains the IN operator: Operator.

