



# A00-221

---

SAS BIG DATA PROGRAMMING AND LOADING

[ANALYTICSEXAM.COM](http://ANALYTICSEXAM.COM)

Exam Summary – Syllabus – Questions

## Table of Contents

<b>Introduction to A00-221 Exam on SAS Big Data Programming and Loading.....</b>	<b>2</b>
<b>SAS A00-221 Certification Details: .....</b>	<b>2</b>
<b>SAS A00-221 Exam Syllabus:.....</b>	<b>2</b>
<b>A00-221 Sample Questions: .....</b>	<b>10</b>
<b>Answers to A00-221 Exam Questions:.....</b>	<b>13</b>

# Introduction to A00-221 Exam on SAS Big Data Programming and Loading

This page is a one-stop solution for any information you may require for SAS Big Data Programming and Loading (A00-221) Certification exam. The SAS A00-221 Exam Summary, Syllabus Topics and Sample Questions provide the base for the actual SAS Certified Big Data Professional Using SAS 9 exam preparation, we have designed these resources to help you get ready to take your dream exam.

The SAS Big Data Programming and Loading credential is globally recognized for validating SAS Big Data Professional knowledge. With the SAS Certified Big Data Professional Using SAS 9 Certification credential, you stand out in a crowd and prove that you have the SAS Big Data Professional knowledge to make a difference within your organization. The SAS Big Data Programming and Loading Certification (A00-221) exam will test the candidate's knowledge on following areas.

## SAS A00-221 Certification Details:

Exam Name	SAS Big Data Programming and Loading
Exam Code	A00-221
Exam Duration	105 minutes
Exam Questions	60 to 65 Multiple choice questions
Passing Score	68%
Exam Price	\$180 (USD)
Training	<ol style="list-style-type: none"> <li><a href="#">SAS Academy for Data Science: Big Data</a></li> <li><a href="#">DS2 Programming Essentials with Hadoop</a></li> <li><a href="#">Introduction to SAS and Hadoop</a></li> <li><a href="#">Getting Started with SAS In-Memory Statistics</a></li> </ol>
Exam Registration	<a href="#">Pearson VUE</a>
Sample Questions	<a href="#">SAS Big Data Professional Certification Sample Question</a>
Practice Exam	<a href="#">SAS Big Data Professional Certification Practice Exam</a>

## SAS A00-221 Exam Syllabus:

Objective	Details
<p><b>SAS and Hadoop - 30%</b></p>	<p><b>Describe the baseline requirements for interacting with Hadoop</b></p> <ul style="list-style-type: none"> <li>• SAS_HADOOP_JAR_PATH and SAS_HADOOP_CONFIG_PATH environment variables</li> <li>• JAR file requirements</li> <li>• Hadoop XML configuration file contents</li> <li>• JAR files and XML files must be accessible to the SAS Server</li> <li>• Understand the precedence of settings for Hadoop XML configuration files</li> <li>• Identify the components of a SAS and Hadoop solution</li> <li>• List the communication paths between the components of a SAS and Hadoop solution</li> </ul> <p><b>Use the HADOOP procedure and the Hadoop FILENAME statement to interact with Hadoop from a SAS session</b></p> <ul style="list-style-type: none"> <li>• Know which HDFS commands are available through the Hadoop procedure</li> <li>• Submit HDFS file system commands (DELETE, MKDIR, RENAME, CHMOD, LS, CAT)</li> <li>• Copy files between SAS and Hadoop via COPYFROMLOCAL and COPYTOLOCAL statement</li> <li>• Submit MapReduce programs with the MAPREDUCE statement</li> <li>• Understand best practice considerations when using the FILENAME statement</li> <li>• use the FILENAME statement to read data from and write data to the Hadoop file system in a SAS DATA step</li> <li>• Execute Pig code with the PIG statement in the HADOOP procedure</li> </ul> <p><b>Query and manage Hive tables stored in Hadoop using explicit SQL pass-through</b></p> <ul style="list-style-type: none"> <li>• Manage connections to Hive with the CONNECT/DISCONNECT statements (schema, server, username, password, etc)</li> <li>• Access Hive metadata via SHOW and DESCRIBE statements</li> <li>• Select data from tables with HiveQL (select, from, where clauses)</li> </ul>

	<ul style="list-style-type: none"> <li>• Join tables with HiveQL</li> <li>• Use both HiveQL and SAS SQL features (ORDER BY, functions, labels) in the same SQL procedure SELECT statement</li> <li>• Create SAS data sets and views from Hive results</li> <li>• String dates vs. SAS dates</li> <li>• Using the CAST function to control data type in explicit queries (32k string lengths)</li> <li>• Create Hive table definitions</li> <li>• Load data into Hive table definitions from local data</li> <li>• Load data into Hive table definitions from HDFS data</li> <li>• Control length of character variables created in Hive tables</li> <li>• Work with Hive string types in SAS</li> <li>• Control Hive table properties with TBLPROPERTIES statement (SASFMT or with data set option DBSASTYPE= option)</li> <li>• Compare managed and external Hive tables</li> <li>• Compare different Hive file types (textfile, sequencefile). Use of SERDES</li> <li>• Use data set options to define specific HDFS file types</li> </ul> <p><b>Work with Hadoop files using the SAS/ACCESS LIBNAME statement</b></p> <ul style="list-style-type: none"> <li>• Write a LIBNAME statement to access Hive tables</li> <li>• Access Hive metadata using LIBNAME statement and the CONTENTS procedure</li> <li>• Understand that the SAS/ACCESS engine writes database-specific SQL code when using implicit pass-through</li> <li>• Maximize the use of HiveQL by optimizing implicit pass-through (summarization, subsetting, joins)</li> <li>• Use system options to determine where processing occurs (SASTRACE, NOSTSUFFIX, SASTRACELOC)</li> <li>• Evaluate SAS logs to determine the amount of implicit pass-through performed by a SAS program</li> <li>• Identify SAS data set options that can be implicitly passed to Hive</li> <li>• Identify SAS functions that can be passed to Hive</li> <li>• Convert date formats with the SASDATEFMT= data set option</li> <li>• Embed LIBNAME statements in SQL View definitions</li> </ul>
--	---

	<ul style="list-style-type: none"> <li>• Identify best practices when combining tables to maximize Hive usage</li> <li>• Methods to combine/join tables</li> <li>• Copy data sets to Hive using the COPY procedure</li> <li>• Identify advantages of using SAS/ACCESS LIBNAME method</li> <li>• Identify disadvantages of using SAS/ACCESS LIBNAME method</li> <li>• Maximize performance when using the LIBNAME statement</li> <li>• Efficient methods for BY GROUP processing with in-database procedures</li> <li>• Managing data types for computed columns.</li> <li>• Partition and cluster Hive tables</li> <li>• Create Hive external tables</li> </ul>
<p><b>SAS DS2 Programming</b> - 30%</p>	<p><b>Write DS2 programs</b></p> <ul style="list-style-type: none"> <li>• Utilize run group processing</li> <li>• Use DATA, ENDDATA, and RUN statements properly</li> <li>• Use system methods, INIT(), RUN(), TERM()</li> <li>• Build user defined methods</li> <li>• Pass arguments to user defined methods</li> <li>• Explain the use of the INIT(), RUN(), TERM() system methods</li> <li>• Use the OVERWRITE option</li> <li>• Understand how DS2 handles reserved keywords</li> <li>• Recognize components of traditional DATA Step programming that are or are not supported in DS2</li> </ul> <p><b>Read data using DS2</b></p> <ul style="list-style-type: none"> <li>• read data with a SET statement</li> <li>• write FedSQL code within SET statements to read data</li> <li>• Use FedSQL SELECT statements to extract specific variables from input data sets</li> <li>• Use FedSQL Join statements to merge data from multiple input data sets</li> <li>• Use FedSQL WHERE statements to extract specific observations from input data sets</li> <li>• Use the MERGE statement to join data.</li> <li>• Subset data using subsetting IF statements</li> <li>• Read table data with a BY statement, without pre-sorting the data</li> </ul>

	<ul style="list-style-type: none"> <li>• Use a FedSQL query with an ORDER BY clause to provide sorted data to the SET statement for BY group processing</li> </ul> <p><b>Work with variables, arrays, and ANSI SQL data types</b></p> <ul style="list-style-type: none"> <li>• Define and use local and global variables (understand scope, what goes into PDV, output data sets)</li> <li>• Declare variables with the DCL statement</li> <li>• Use fractional, integer, character and Date &amp; Time ANSI SQL data types</li> <li>• Use CHAR, NCHAR, VARCHAR, NVARCHAR character data types</li> <li>• Use DECIMAL, DOUBLE, FLOAT, REAL fractional numeric data types</li> <li>• Use BIGINT, INTEGER, SMALLINT, TINYINT interger numeric data types</li> <li>• Use BINARY, VARBINARY binary data types</li> <li>• Use DATE, TIME, TIMESTAMP date and time data types</li> <li>• Identify coercible and non-coercible data types</li> <li>• Understand autoconversion of DS2 data types when DS2 variables are output to SAS data sets</li> <li>• Select variables with KEEP and DROP statements and KEEP= and DROP= options</li> <li>• Understand how SAS will perform automatic type conversions</li> <li>• DS2SCOND option statement</li> <li>• Use ANSI quoting standards in variable assignment statements</li> <li>• Use macro variables within ANSI quoted variable assignment statements (%TSLIT macro)</li> <li>• set variable attributes (Length, format, informat) within variable declaration statements</li> <li>• Use the VARARRAY statement to declare arrays</li> <li>• Use the DCL staement to declare temporary arrays</li> <li>• Assign values to array variables</li> <li>• Understand the difference between SAS MISSING and ANSI NULL data values</li> <li>• Invoke ANSI data processing mode for NULL values with the ANSIMODE option</li> </ul> <p><b>Use expressions and functions in DS2 programs</b></p>
--	---

	<ul style="list-style-type: none"> <li>• Use the DS2 IF expression in place of IF/THEN conditional statements</li> <li>• Use the DS2 LIKE expression to compare character values to specific patterns</li> <li>• Convert SAS datetime variables to DS2 ANSI TIMESTAMP variables with the TO_TIMESTAMP function</li> <li>• Convert SAS date variables to DS2 ANSI DATE variables with the TO_DATE function</li> <li>• Convert SAS time variables to DS2 ANSI TIME variables with the TO_TIME function</li> <li>• Convert DS2 ANSI DATE, TIME, and TIMESTAMP variables to SAS date, time and datetime variables with the TO_DOUBLE function</li> <li>• Increment date and time values with the INTDT and INTTS functions</li> <li>• Execute FedSQL statements with the SQLEXEC function</li> </ul> <p><b>Work with Methods, Packages, and Threads</b></p> <ul style="list-style-type: none"> <li>• Create methods that modify parameters at the site by using IN_OUT variables</li> <li>• Create methods that return a value using a RETURN statement</li> <li>• Overload methods by creating methods with multiple signatures</li> <li>• Create user defined packages with the PACKAGE statement</li> <li>• Understand the capabilities of predefined DS2 packages (such as FCMP, SQLSTMT, HASH, JSON)</li> <li>• Intantiate DS2 packages with the DECLARE statement</li> <li>• Use threading to alleviate CPU bound processes</li> <li>• Create threads using the THREAD statement</li> <li>• Declare instances of threads in a DS2 program</li> <li>• Call threads using a SET FROM statement</li> <li>• Specify the number of threads using a THREADS= option</li> <li>• How to run threads inside parallel databases</li> <li>• DS2ACCEL = YES option</li> <li>• Requirements to execute DS2 code in-database</li> </ul>
<p><b>Hadoop Programming - 15%</b></p>	<p><b>Describe the Hadoop architecture</b></p>

	<ul style="list-style-type: none"> <li>• Identify Hadoop elements such as Name Nodes, Data Nodes, Job Trackers, Task Trackers, YARN</li> <li>• Explain Hadoop concepts such as distributed storage &amp; processing, splits, replication, MapReduce</li> <li>• Describe components of the Hadoop ecosystem (Hive, Pig, Sqoop)</li> <li>• Describe attributes of big data</li> <li>• Identify use cases for Hadoop</li> </ul> <p><b>Manipulate and load data files using command line tools</b></p> <ul style="list-style-type: none"> <li>• Use Linux shell commands (ls -ltr/pwd/mkdir/cd)</li> <li>• Load and manipulate data into Hadoop using Linux commands (hdfs dfs - mkdir/put/copyFromLocal/ls/cat)</li> <li>• Use Sqoop to move data from a RDBMS into Hadoop</li> </ul> <p><b>Write Hive programs to create, join, and query data tables</b></p> <ul style="list-style-type: none"> <li>• Create databases and tables in Hive</li> <li>• Understand the difference between external and internal tables</li> <li>• Work with Hive variable types</li> <li>• Load data into Hive table definitions with LOAD and INSERT statements</li> <li>• Recognize challenges when importing data (embedded delimiter characters, header values)</li> <li>• Limit values returned by a Hive query with the SELECT.....LIMIT keyword</li> <li>• Sort Hive query results with the SELECT...ORDER BY keyword</li> <li>• Group Hive query results with the GROUP BY keyword</li> <li>• Choose which values to select from a data table using the SELECT WHERE keyword</li> <li>• Retrieve unique values with the SELECT DISTINCT</li> <li>• Join tables (Inner, Outer, Left, Right)</li> <li>• Use functions in Hive queries (sum, count, avg, max, min, round, floor, ceil, rand, concat, substr, upper, ucase, lower, lcase, trim)</li> <li>• Use relational and arithmetic operators in Hive queries</li> </ul>
--	---

	<p><b>Write Pig programs to perform ETL tasks and to analyze large data sets</b></p> <ul style="list-style-type: none"> <li>• Identify Pig data types</li> <li>• Build Pig programs with LOAD, FOREACH/GENERATE, FILTER, SPLT, LIMIT, UNION, DISTINCT, ORDER, GROUP, STORE, DUMP keywords</li> <li>• Use name and positional references in Pig programs</li> <li>• Identify valid identifiers (start with letter, then letters, digits, underscores)</li> <li>• Use Arithmetic, String, and Boolean Expressions</li> <li>• Use the CAST operator to change variable types</li> <li>• Increase parallel processing with the PARALLEL keyword</li> <li>• Combine data from multiple tables with INNER, LEFT, RIGHT, and OUTER JOIN keywords</li> <li>• Combine data using special join types: REPLICATED, SKEWED, MERGE</li> <li>• Use parameters in a Pig program</li> <li>• Use Diagnostic operators: DESCRIBE, EXPLAIN, DUMP, ILLUSTRATE</li> <li>• Use functions in Pig programs</li> <li>• EVAL Functions: AVG, SUM, CONCAT, COUNT, COUNT_STAR, IsEmpty, MIN, MAX, SIZE, SUBTRACT, TOKENIZE</li> <li>• DATE Functions: CurrentTime, DaysBetween, HoursBetween, GetDay, GetHour (etc.), AddDuration, ToUnixTime, ToDate, ToMilliseconds, ToString</li> <li>• String Functions: STARTSWITH, ENDSWITH, REGEX_EXTRACT, REPLACE, TRIM, LTRIM, RTIM, INDEXOF, LAST_INDEX_OF, LOWER, UPPER, LCFIRST, UCFIRST SUBSTRING, EqualsIgnoreCase</li> <li>• Math Functions: ABS, ACOS, ATAN (etc) SQRT, CBRT, Exp, CEIL, FLOOR, LOG, LOG10, RANDOM, ROUND</li> <li>• Tuple, Bag, Map functions: TOTUPLE, TOBAG, TOMAP, TOP</li> <li>• Register and use User Defined Functions</li> </ul>
<p><b>Data manipulation with the IMSTAT procedure - 25%</b></p>	<p><b>Execute IMSTAT procedures</b></p> <ul style="list-style-type: none"> <li>• Define a SASIOLA library to access in-memory data in a LASR Analytic Server</li> <li>• Describe the key functionality of the IMSTAT procedure</li> </ul>

	<ul style="list-style-type: none"> <li>• Perform one-dimensional numerical exploration with IMSTAT procedure statements SUMMARY and FREQUENCY</li> <li>• Perform two-dimensional numerical exploration using the CROSSTAB or GROUPBY=option</li> </ul> <p><b>Perform actions required to produce graphs with PROC IMSTAT</b></p> <ul style="list-style-type: none"> <li>• Use PROC IMSTAT statements and options that calculate summary statistics for graphing</li> <li>• Transfer the summary statistics tables to the SAS server</li> </ul> <p><b>Manipulate In-Memory Data</b></p> <ul style="list-style-type: none"> <li>• Define WHERE clauses to explore subsets of an in-memory table</li> <li>• Create permanent columns using the COMPUTE statement</li> <li>• Create temporary columns using temporary expressions of computed columns</li> <li>• Work with SAS formats in the IMSTAT procedures</li> <li>• Use the Fetch statement to retrieve data from an in-memory table</li> <li>• Join in-memory tables</li> </ul> <p><b>Use High-Performance procedures with the SAS LASR Analytic Server</b></p> <ul style="list-style-type: none"> <li>• Compare the SAS High-Performance procedures and SAS IN-Memory Statistics</li> <li>• Use the HPIMPUTE procedure to add imputed columns to an in-memory table</li> </ul>
--	---

## A00-221 Sample Questions:

**Q1: In the example below, the input data set is a Hive table accessed using a SAS/ACCESS to Hadoop LIBNAME statement.**

```
proc freq data=hivelib.myhivetable;
  class month;
  by year;
run;
```

**Which statement is true about this program?**

**Options:**

A. BY statements do not require the data be received by the procedure in any specific order.

- B. The procedure will fail unless the table HIVELIB.MYHIVETABLE is already stored ordered by YEAR.
- C. BY statements are not supported for Hive tables because it is not possible to order data that is distributed on different nodes of the Hadoop cluster.
- D. SAS will generate a HiveQL query to return the data to SAS ordered by YEAR so that the procedure receives the data ordered by YEAR as required.

**Q2: Many temporary tables may be created in the LASR server by PROC IMSTAT analysis actions. What happens to temporary tables when a PROC IMSTAT session is terminated?**

**Options:**

- A. All temporary tables are saved to the SAS server WORK library.
- B. The last temporary table created is saved to the SAS WORK library.
- C. All temporary tables are purged from the LASR server.
- D. The last temporary table created is saved to storage in the HDFS.

**Q3: Refer to the log message shown below:**

```
proc ds2;
  data test;
  dcl double date;
  method run();
  set work.one;
  mm=2;
  end;
  enddata;
run;
```

ERROR: Compilation error.

ERROR: Parse encountered type when expecting identifier.

ERROR: Parse failed on line 60: dcl double >>> date <<< ;

NOTE: PROC DS2 has set option NOEXEC and will continue to prepare statements.

```
quit;
```

**Which of the following changes will fix the errors shown in the log?**

**Options:**

- A. Replace line 60 with dcl double "date";
- B. Replace line 60 with dcl string date;
- C. Replace line 60 with dcl double 'date'n;
- D. Replace line 60 with dcl double 'date';

**Q4: The following SAS program is submitted:**

```
proc ds2;
  data work.sasorders;
  dcl timestamp order_timestamp;
  dcl double order_datetime;
  method run();
  set orders;
  order_timestamp = to_timestamp(order_datetime);
  end;
enddata;
```

```
run;
quit;
```

**What happens when the program is executed?**

**Options:**

- A.
  - The variable order\_timestamp is created and processed as an ANSI timestamp value in the DS2 program.
  - The order\_timestamp value is converted to a SAS datetime when it is written to the output SAS data set.
- B.
  - The variable order\_timestamp is created and processed as an ANSI timestamp value in the DS2 program.
  - The output data set stores the values as a SAS timestamp value.
- C.
  - The program does not execute because order\_datetime is a SAS datetime value.
- D.
  - The variable order\_timestamp is converted to a SAS time value.
  - The output data set stores this as the number of seconds since midnight.

**Q5: This question will ask you to provide a line of missing code.**

**Which line of code would you insert to get the mean and standard deviation of INCOME and AGE, calculated separately for GENDER variable values F (female) and M (male)?**

```
proc imstat;
  table mylasr.bankdata(tag="&TagString");
  <insert code here>
run;
```

**Option:**

- A. crosstab income\*gender age\*gender;
- B. summary income age / groupby=gender;
- C. univariate income age / by=gender;
- D. summary income age / by=gender;

**Q6: Web server logs are written in an HDFS directory. The following lines indicate the format and an example of the comma-separated values for one line in the log file.**

```
# IP address, timestamp, request, status, size
192.168.12.41,24/Nov/2015:10:09:58 -0500, "GET /services/config.xml
HTTP/1.1",200,816
```

Which CREATE TABLE statement enables a Hive query to access each of the fields?

**Options:**

- A. create external table weblogs (ip string, dt string, req string, status int, sz string) fields delimited fields by ',' location '/data/weblogs';
- B. create external table weblogs (ip string, dt string, req string, status int, sz string) fields terminated by ',' location '/data/weblogs';

- C. create external table weblogs (ip string, rest string) row format delimited fields terminated by ',' location '/data/weblogs';
- D. create external table weblogs (ip string, dt string, req string, status int, sz string) row format delimited fields terminated by ',' location '/data/weblogs';

**Q7: What is an advantage of using a LIBNAME statement to interact with your Hadoop cluster?**

**Options:**

- A. It enables some SAS procedures to push processing into Hive.
- B. It ensures that Hive will handle all processing.
- C. It enables you to submit user-written HiveQL code to Hive.
- D. The GENERATE\_PIG\_CODE= option enables you to bypass Hive and generate Pig Latin code.

**Q8: When working with data stored in Hadoop, which SAS function is NOT passed to Hive by default?**

**Options:**

- A. DATE
- B. HOUR
- C. VAR
- D. UPCASE

**Q9: Which operator is NOT a diagnostic operator for testing a Pig program?**

**Options:**

- A. EXPLAIN
- B. ILLUSTRATE
- C. SPLIT
- D. DUMP

**Q10: Which statement creates a temporary array within DS2?**

**Options:**

- A. dcl double a[2];
- B. vararray double a(2);
- C. array a(2) s1-s2;
- D. vararray double a[2];

**Answers to A00-221 Exam Questions:**

Question: 1	Answer: D	Question: 2	Answer: C
Question: 3	Answer: A	Question: 4	Answer: A
Question: 5	Answer: B	Question: 6	Answer: D
Question: 7	Answer: A	Question: 8	Answer: A
Question: 9	Answer: C	Question: 10	Answer: A

Note: If you find any typo or data entry error in these sample questions, we request you to update us by commenting on this page or write an email on [feedback@analyticsexam.com](mailto:feedback@analyticsexam.com)