New 2024 Realistic Free SAP C-HANADEV-18 Exam Dump Questions & Answer [Q37-Q55
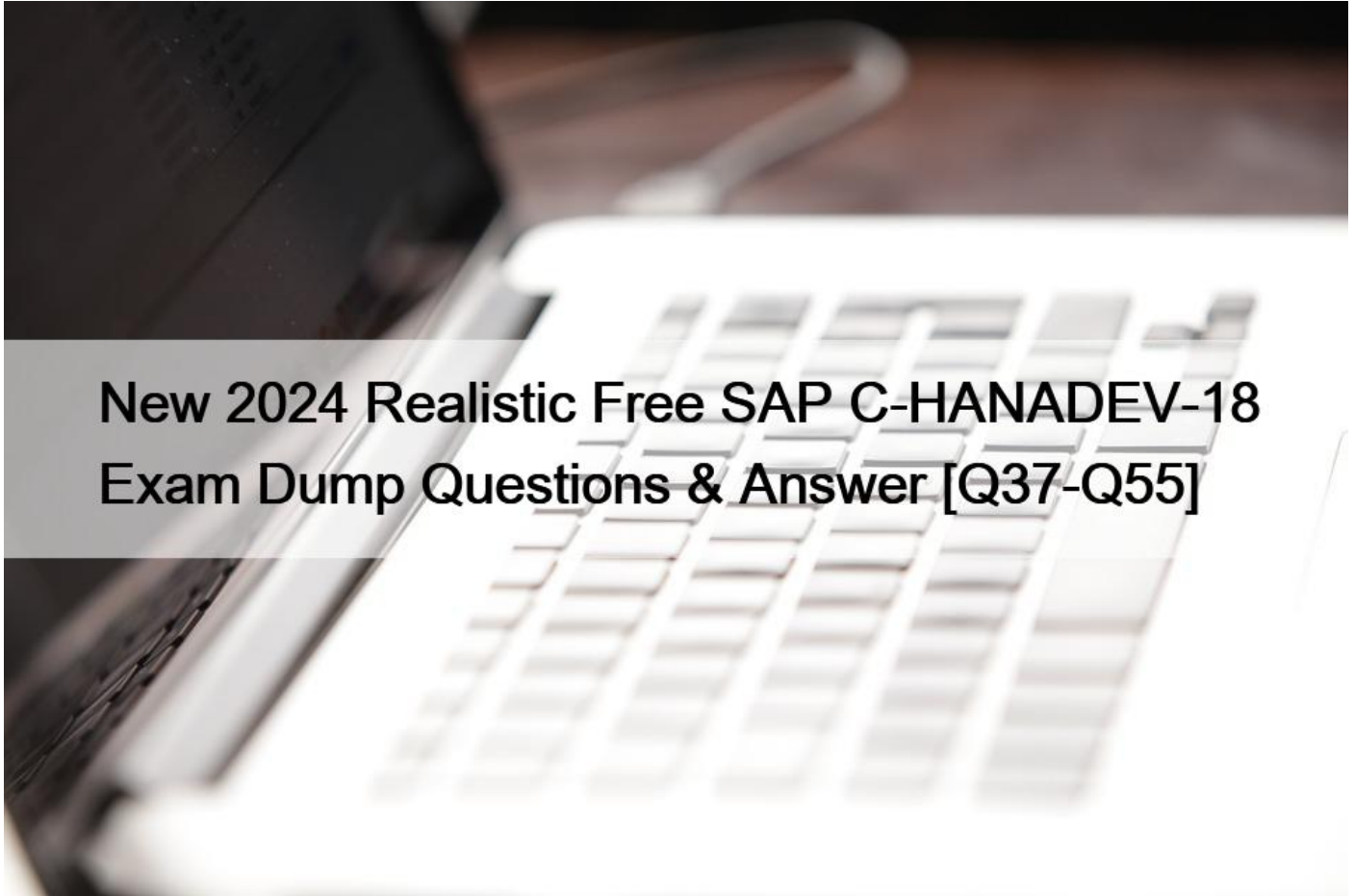

New 2024 Realistic Free SAP C-HANADEV-18 Exam Dump Questions and Answer

C-HANADEV-18 Practice Test Engine: Try These 95 Exam Questions

To be eligible for the SAP C-HANADEV-18 certification exam, candidates must have a solid understanding of SAP HANA 2.0 SPS06 technology and have experience in developing and deploying applications using the technology. Candidates can prepare for the exam by attending SAP training courses, reviewing SAP HANA documentation, and practicing their skills in SAP HANA development. Passing the exam can enhance their career prospects by demonstrating their expertise in SAP HANA 2.0 SPS06 technology and increasing their value to potential employers.

SAP C-HANADEV-18 certification exam is a valuable credential for developers who work with SAP HANA. It is designed to test their understanding of SAP HANA architecture, data modeling, SQL and SQL Script, data provisioning, and application development. Passing C-HANADEV-18 exam demonstrates to potential employers that a developer has the skills and knowledge needed to develop and deploy applications on the SAP HANA platform.

NO.37 Which parameters does the APPLY_FILTER function support? There are 2 correct answers to this question.
* Filter condition
* Join condition
* Entity set
* Table variable

NO.38 You need to install SAP HANA 2.0, express edition to develop a native SAP HANA application. Which of the following deployment options do you have?

There are 2 correct answers to this question.
* Installation on Windows Server
* Installation on Mac OS
* Installation on Linux OS
* Usage of virtual machine on Microsoft Windows
SAP HANA 2.0, express edition is a streamlined version of SAP HANA that can run on laptops and other resource-constrained hosts. It supports native SAP HANA application development and can be installed on Linux OS or used as a virtual machine on Microsoft Windows. Installation on Windows Server or Mac OS is not supported by SAP HANA 2.0, express edition. References:

* SAP HANA 2.0 SPS06 – Application Development for SAP HANA1, Section 1.1, p. 5

* SAP HANA, express edition – Installation Guide, Section 1.1, p. 7

* SAP HANA, express edition – FAQ, Question 1

NO.39 What are the key characteristics of the calculation view’s Input Parameter? There are 3 correct answers to this question.
* It is passed using a WHERE clause.
* It can NOT be used to filter measure values.
* It can be used in a conditional expression.
* It is passed via reserved word PLACEHOLDER.
* It can be used to pass values to table functions.

NO.40 You developed a multi-target application that contains only a database module. Which environment are the runtime objects created in? Please choose the correct answer.
* Java Runtime Environment (JRE)
* SAP HANA Runtime Tools (HRTT)
* SAP Web IDE for SAP HANA
* HANA Deployment Infrastructure Container
A multi-target application (MTA) is a single application that consists of multiple modules that are developed using different technologies and designed to run on different target runtime environments. A database module is a module that contains database artifacts, such as tables, views, procedures, or functions, that are deployed to a SAP HANA database. A HANA Deployment Infrastructure (HDI) container is a logical grouping of database objects that are isolated from other containers and schemas in the same database. An HDI container has its own technical user, roles, and privileges, and can be accessed only through a service binding. When a database module is built, the SAP Web IDE for SAP HANA or the SAP Business Application Studio automatically creates an HDI container and binds it as a resource to the database module. It also creates the runtime objects, such as physical tables or views, in the schema associated with the HDI container. Therefore, the runtime objects of a database module are created in the HDI container, which is the correct answer. The other options are incorrect because they are not the environments where the runtime objects of a database module are created. The Java Runtime Environment (JRE) is a software environment that provides the minimum requirements for executing a Java application. The SAP HANA Runtime Tools (HRTT) are a set of tools that enable developers to create, run, and debug SAP HANA native applications in Eclipse. The SAP Web IDE for SAP HANA is a web-based

development environment that supports the development of MTA projects and modules. References:

* SAP HANA Platform 2.0 SPS06: Developing Multitarget Applications, Section 1.1

* SAP HANA Platform 2.0 SPS06: SAP HANA Deployment Infrastructure, Section 1

* SAP HANA Platform 2.0 SPS06: SAP HANA Database Application Development, Section 2.1

* SAP HANA Platform 2.0 SPS06: SAP HANA Database Application Development, Section 2.2

* SAP HANA Platform 2.0 SPS06: SAP HANA Database Application Development, Section 2.3

* SAP Business Application Studio Multitarget Application (MTA) development toolkit, Section 2

NO.41 In Node.js, which file do you use to define external module dependencies? Please choose the correct answer.
* mta.yaml
* pom.xml
* package.json
* server.js

In Node.js, the package.json file is used to define external module dependencies. A module is a piece of code that can be reused and shared by other applications. A dependency is a module that is required by another module to function properly. The package.json file contains a list of dependencies with their names and versions, as well as other metadata about the application, such as name, description, version, scripts, and license. The package.json file allows the Node.js application to install and manage the dependencies using the npm command-line tool. The other files are not used to define external module dependencies in Node.js. The mta.yaml file is used to define the structure and dependencies of a multi-target application. The pom.xml file is used to define the configuration and dependencies of a Maven project. The server.js file is used to define the main entry point and logic of a Node.js application. References:

* SAP HANA 2.0 SPS06 – Application Development for SAP HANA1, Section 5.2.1, p. 135-136

* SAP HANA 2.0 SPS06 – Node.js Development2, Section 2.1, p. 15-16

* SAP HANA 2.0 SPS06 – Node.js Development2, Section 2.2, p. 17-18

* SAP HANA 2.0 SPS06 – Node.js Development2, Section 2.3, p. 19-20

* SAP HANA 2.0 SPS06 – Node.js Development2, Section 2.4, p. 21-22

* SAP HANA 2.0 SPS06 – Node.js Development2, Section 2.5, p. 23-24

* SAP HANA 2.0 SPS06 – Node.js Development2, Section 2.6, p. 25-26

* SAP HANA 2.0 SPS06 – Node.js Development2, Section 2.7, p. 27-28

* SAP HANA 2.0 SPS06 – Node.js Development2, Section 2.8, p. 29-30

* SAP HANA 2.0 SPS06 – Node.js Development2, Section 2.9, p. 31-32

* SAP HANA 2.0 SPS06 – Node.js Development2, Section 2.10, p. 33-34

* SAP HANA 2.0 SPS06 – Node.js Development2, Section 2.11, p. 35-36

* SAP HANA 2.0 SPS06 – Node.js Development2, Section 2.12, p. 37-38

NO.42 To which SAP HANA authorization entities can you grant a role? There are 2 correct answers to this question.
* Object
* Role
* Privilege
* User

NO.43 Which Git commands are available with the SAP Web IDE for SAP HANA? There are 3 correct answers to this question.
* Test changes
* Push code
* Clone a repository
* Redeploy the code
* Merge changes

NO.44 Your multi-target application will use XSJS. Which module type do you create in your application project?

Please choose the correct answer.
* Java
* SAP HANA database
* Node.js
* HTML5
To create a multi-target application that will use XSJS, you need to create a Node.js module type in your application project. XSJS is a scripting language that allows you to implement server-side logic and access SAP HANA database features using JavaScript syntax. XSJS is supported by the Node.js runtime of the SAP HANA XS Advanced Model, which provides the necessary libraries and services to execute XSJS code. You can use the SAP Web IDE for SAP HANA to create a Node.js module and add XSJS files to it. The other options are not correct because:

* A. Java is not the module type you need to create for XSJS, as XSJS is not a Java-based language. Java is a module type that allows you to implement Java applications using the Java runtime of the SAP HANA XS Advanced Model.

* B. SAP HANA database is not the module type you need to create for XSJS, as XSJS is not a database object. SAP HANA database is a module type that allows you to create database artifacts, such as tables, views, procedures, and functions, using the SAP HANA deployment infrastructure (HDI).

* D. HTML5 is not the module type you need to create for XSJS, as XSJS is not a web page. HTML5 is a module type that allows you to create static web content, such as HTML, CSS, and JavaScript files, using the HTML5 application repository service.

References:

* SAP HANA Developer Guide for SAP HANA Web IDE, Chapter 5, Section 5.6

* SAP HANA Developer Guide for SAP HANA Web IDE, Chapter 5, Section 5.7

* SAP HANA Developer Guide for SAP HANA Web IDE, Chapter 5, Section 5.8

* SAP HANA Developer Guide for SAP HANA Web IDE, Chapter 5, Section 5.9

NO.45 Which Git command do you use to consolidate all changes from one branch with another branch using a single commit? Please choose the correct answer.

* Commit
* Rebase
* Merge
* Push

The Git command that you use to consolidate all changes from one branch with another branch using a single commit is merge. The merge command is used to integrate changes from another branch into the current branch. The target of this integration (i.e. the branch that receives changes) is always the currently checked out HEAD branch. The merge command can create a merge commit, which is a special commit that has two or more parent commits and records the result of the merge. Alternatively, the merge command can also perform a fast-forward merge, which is a simple update of the HEAD pointer without creating a new commit, if the current branch is an ancestor of the other branch.

For example, suppose you have two branches, master and feature, and you want to consolidate all changes from feature into master using a single commit. You can use the following commands:

git checkout master # switch to the master branch git merge feature # merge the feature branch into the master branch This will create a merge commit on the master branch that has two parents: the previous tip of the master branch and the tip of the feature branch. The merge commit will contain all the changes from the feature branch as well as the changes from the master branch.

The following Git commands are not used to consolidate all changes from one branch with another branch using a single commit, but for other purposes:

* Commit: The commit command is used to create a new commit on the current branch that records the changes made in the working tree and the index. The commit command does not integrate changes from another branch, but only from the local repository.

* Rebase: The rebase command is used to reapply a series of commits from one branch on top of another branch. The rebase command does not create a single commit, but rather modifies the history of the current branch by rewriting the commits and changing their parent commits. The rebase command can be used to achieve a linear history, but it can also cause conflicts and inconsistencies if used on public branches.

* Push: The push command is used to transfer commits from the local repository to a remote repository.

The push command does not integrate changes from another branch, but only from the local repository to the remote repository. The push command can also update the remote branch pointers to reflect the

* transferred commits.

References:

* [Git Tower], Git Merge – Integrating changes from another branch,

https://www.git-tower.com/learn/git/commands/git-merge/.

* [Git Documentation], git-merge – Join two or more development histories together,

https://git-scm.com/docs/git-merge.

NO.46 What types of user-defined functions are supported by SAP HANA? Note: There are 2 correct answers to this question.

* Table
* Scalar
* Aggregate
* Hierarchy

NO.47 Which OData capacities are supported in SAP HANA extended application services, advanced model (XSA)? There are 3 correct answers to this question.
* Union
* Aggregation
* Join
* Projection
* Association

NO.48 How do you debug a Node.js module in SAP Web IDE for SAP HANA? Please choose the correct answer.
* Set the enabled parameter to true in the section debugger of the xsengine.ini file.
* Add the sap.hana .xs.debugger::Debugger role to the HDI Container’s #RT User.
* Start the debugger from the XS command line interface and run the program in SAP Web IDE for SAP HANA.
* Attach the debugger to the application in the SAP Web IDE for SAP HANA.
According to the SAP Web IDE for SAP HANA Developer Guide, you can debug a Node.js module in SAP Web IDE for SAP HANA by attaching the debugger to the application in the SAP Web IDE for SAP HANA.

To do this, you need to open the debugger panel, attach your application, and choose your multi-target application and select a debug target. Then, you can perform the regular debugging tasks, such as setting breakpoints, stepping through the code, examining the variables, and so on. The other options are incorrect, because:

* Setting the enabled parameter to true in the section debugger of the xsengine.ini file is not a way to debug a Node.js module in SAP Web IDE for SAP HANA, but a way to enable the XS JavaScript debugger for XSJS applications. This is not relevant for Node.js modules, which use a different runtime and debugger.

* Adding the sap.hana.xs.debugger::Debugger role to the HDI Container’s #RT User is not a way to debug a Node.js module in SAP Web IDE for SAP HANA, but a way to grant the XS JavaScript debugger privileges to the runtime user of the HDI container. This is not relevant for Node.js modules, which use a different runtime and debugger.

* Starting the debugger from the XS command line interface and running the program in SAP Web IDE for SAP HANA is not a way to debug a Node.js module in SAP Web IDE for SAP HANA, but a way to debug a Node.js module using command-line tools. This is an alternative option for debugging Node.js modules, but it does not use the SAP Web IDE for SAP HANA.

References: SAP Web IDE for SAP HANA Developer Guide, Chapter 6, Section 6.4.2, page 2111.

NO.49 What are the nodes where filter expressions can be used in a calculation view? There are 2 correct answers to this question.
* Aggregation
* Star join
* Union
* Rank
Filter expressions can be used in a calculation view to restrict or modify the data that is displayed or processed by the view. Filter expressions can be used in the following nodes of a calculation view12:

* Aggregation: An aggregation node is a node that applies aggregation functions, such as sum, count, or average, to the data that is passed from the previous node. Filter expressions can be used in an aggregation node to filter the data before or after the aggregation, or to define the aggregation level or the measure attributes. For example, you can use a filter expression to show only

the sales data for a specific region or product category, or to calculate the average revenue per customer.

* Union: A union node is a node that combines the data from two or more nodes that have the same structure and data types. Filter expressions can be used in a union node to filter the data from each input node, or to filter the data from the output node. For example, you can use a filter expression to exclude the duplicate rows from the union result, or to show only the data that matches a certain condition from each input node.

The other options are not correct because filter expressions cannot be used in these nodes of a calculation view. A star join node is a node that joins a fact table with one or more dimension tables, based on the common key attributes. A star join node does not support filter expressions, but it supports input parameters, which are variables that can be used to filter the data at runtime. A rank node is a node that ranks the data according to a specified order and criteria. A rank node does not support filter expressions, but it supports rank filters, which are conditions that can be used to limit the number or percentage of rows in the rank result.

References:

* SAP HANA Platform, SAP HANA Modeling Guide for SAP HANA Web Workbench, Calculation Views

* SAP HANA Platform, SAP HANA Developer Guide for SAP HANA Web IDE, Developing Database Modules, Developing Calculation Views, Using Filter Expressions

NO.50 You created several database tables in a multi-target application and need to keep their names as short as possible. Which parameter of the .hdinamespace file do you set to ignore? Please choose the correct answer.
* name
* hdbtable
* mixinTypes
* subfolder

According to the SAP HANA Performance Guide for Developers1, the .hdinamespace file is a JSON resource that defines the naming rules for the run-time objects in a multi-target application. The name parameter specifies the common name-space prefix for the objects, while the subfolder parameter determines whether the name of the sub-folder is added or ignored in the run-time name space. If you want to keep the names of the objects as short as possible, you should set the subfolder parameter to ignore, so that the name of the sub-folder is not appended to the name-space prefix and the object name. For example, if you have a table named CUSTOMERS in a sub-folder named SALES, and you set the subfolder parameter to ignore, the run-time name of the table will be com.sap.hana.example::CUSTOMERS, where com.sap.hana.example is the name-space prefix defined by the name parameter. If you set the subfolder parameter to append, the run-time name of the table will be com.sap.hana.example.SALES::CUSTOMERS, which is longer and more complex. For more information on the .hdinamespace file and the naming rules, see [The HDI Name-Space Configuration File]2 and [Run-Time Name Spaces in SAP HDI]3.

References: 2: The HDI Name-Space Configuration File 3: Run-Time Name Spaces in SAP HDI 1: SAP HANA Performance Guide for Developers

NO.51 To perform a specific task of an XS advanced application, what does a user need? Please choose the correct answer.
* To have directly assigned a Role Collection
* To have directly assigned a Scope
* To be assigned to an Organization
* To be assigned to a Space

NO.52 You need to create a native SAP HANA application that fully leverages the SAP HANA platform. How do you implement data-intensive calculations?

Please choose the correct answer.

* Push the calculations onto the application layer.
* Push the calculations onto the database layer
* Push the calculations onto the presentation layer.
* Distribute calculations between application layer and presentation layer.

To create a native SAP HANA application that fully leverages the SAP HANA platform, you should implement data-intensive calculations on the database layer, using SQLScript or calculation views. SQLScript is a scripting language that allows you to write stored procedures, functions, and triggers that perform complex calculations and data transformations on the SAP HANA database. Calculation views are graphical or scripted views that define data models based on tables, views, or other calculation views, and apply filters, joins, aggregations, and other operations on the data. By pushing the calculations onto the database layer, you can take advantage of the in-memory processing, parallelization, and optimization capabilities of SAP HANA, and reduce the data transfer and network latency between the application layer and the database layer123.

The other options are not correct because they do not fully leverage the SAP HANA platform, and they may result in poor performance, high resource consumption, and increased complexity. Pushing the calculations onto the application layer means that you use a programming language, such as Java or Node.js, to perform the calculations on the application server, which may not be as efficient or scalable as the database server. Pushing the calculations onto the presentation layer means that you use a UI framework, such as SAPUI5 or SAP Fiori, to perform the calculations on the client device, such as a browser or a mobile device, which may not have enough processing power or memory to handle large or complex data sets. Distributing the calculations between the application layer and the presentation layer means that you split the calculations into different parts and execute them on different layers, which may introduce inconsistency, redundancy, and dependency issues. References:

* SAP HANA Platform, SAP HANA SQL and System Views Reference, SQLScript Guide

* SAP HANA Platform, SAP HANA Modeling Guide for SAP HANA Web Workbench, Calculation Views

* SAP HANA Platform, Developing Applications with SAP HANA Cloud Platform, Developing Multi-Target Applications, Developing Database Modules

NO.53 Which join type is NOT supported by join optimization (pruning)? Please choose the correct answer.
* Outer Join
* Text Join
* Referential Join
* Inner Join

Join optimization (pruning) is a technique that allows you to omit join fields from the aggregation if they are not requested by the query and they do not affect the result set. Join optimization (pruning) can improve the query performance by reducing the data volume and the number of join operations. However, join optimization (pruning) is only supported for certain types of joins, such as outer join, text join, and referential join. These types of joins preserve the number of records from one of the join partners, regardless of the join condition. Therefore, the join execution does not influence the result set, and the join fields can be safely omitted12.

Inner join is not a type of join that is supported by join optimization (pruning). Inner join is a type of join that only returns the records that have a matching partner in both join partners, based on the join condition. Inner join can add or delete records from the result set, depending on the data and the join condition. Therefore, the join execution does influence the result set, and the join fields cannot be omitted without changing the query semantics12. References: 1: Optimize Join Execution | SAP Help Portal 2: Prerequisites for Pruning Join Columns | SAP Help Portal

NO.54 You use APPLY_FILTER to add a dynamic WHERE clause on SELECT statements. On which objects can you apply the filter? There are 2 correct answers to this question.
* Scalar variables
* Virtual tables
* Database tables

* Intermediate table variables

You can use APPLY_FILTER to add a dynamic WHERE clause on SELECT statements that query database tables or intermediate table variables. Database tables are persistent tables that store data in the SAP HANA database. Intermediate table variables are variables that store temporary results of SQL queries in a tabular format. You can use APPLY_FILTER to define data transformations and data flows without using imperative statements or side effects1.

The other two options, scalar variables and virtual tables, are not objects that can be applied with the filter.

Scalar variables are variables that store single values of a specific data type. You cannot use APPLY_FILTER on scalar variables, as they are not tabular data sources. Virtual tables are tables that provide access to remote data sources, such as other SAP HANA databases or external systems. You cannot use APPLY_FILTER on virtual tables, as they are not supported by the function2.
References: 1: APPLY_FILTER | SAP Help Portal 2: Virtual Tables | SAP Help Portal

NO.55 Which new components can you use to develop native SAP HANA applications in SAP HANA extended application services, advanced model (XS advanced)?

There are 2 correct answers to this question.
* SAP HANA repository
* SAP HANA studio
* Git repository
* SAP Web IDE for SAP HANA

Guaranteed Success in SAP Certified Development Associate C-HANADEV-18 Exam Dumps: