SAP.C-HANADEV-18.v2024-04-30.q66

Exam Code:	C-HANADEV-18
Exam Name:	Certified Development Associate - SAP HANA 2.0 SPS06
Certification Provider:	SAP
Free Question Number:	66
Version:	v2024-04-30
# of views:	262
# of Questions views:	660
https://www.freeqas.com/qa/SAP/C-HANADEV-18/SAP.C-HANADEV-18.v2024-04-30.q66.html	

NEW QUESTION: 1
You need to combine data from different database tables within your OData service. Which capability do you use for this relationship? Please choose the correct answer.
A. Aggregations
B. Projections
C. Associations
D. Entity sets
Answer: C (LEAVE A REPLY)
An OData service is a web service that exposes data from a data source, such as a SAP HANA database, in a standardized and interoperable format. An OData service consists of one or more entity sets, which are collections of entities of the same type. An entity is a data object that has properties and a unique key. An OData service can also define associations between entity sets, which are relationships that specify how entities are related to each other. For example, an association can define a one-to-many or a many-to-many relationship between two entity sets. Associations enable an OData service to combine data from different database tables within the same service. For example, an OData service can define an association between a Products entity set and a Categories entity set, where each product belongs to one category and each category can have multiple products. This association can be used to query or update data from both entity sets in a single request.
The other options are incorrect because they are not capabilities that can be used to combine data from different database tables within an OData service. Aggregations are operations that summarize or group data, such as count, sum, average, or min/max. Projections are operations that select or filter data, such as select, expand, or filter. Entity sets are collections of entities of the same type, but they do not define relationships between entities. References:

* SAP HANA Platform 2.0 SPS06: OData V4 Developer Guide, Section 2.1
* SAP HANA Platform 2.0 SPS06: OData V4 Developer Guide, Section 2.3
* SAP HANA Platform 2.0 SPS06: OData V4 Developer Guide, Section 2.4
* SAP HANA Platform 2.0 SPS06: OData V4 Developer Guide, Section 2.5
* SAP HANA Platform 2.0 SPS06: OData V4 Developer Guide, Section 2.6

NEW QUESTION: 2
How do you debug a Node.js module in SAP Web IDE for SAP HANA? Please choose the correct answer.
A. Set the enabled parameter to true in the section debugger of the xsengine.ini file.
B. Add the sap.hana .xs.debugger::Debugger role to the HDI Container's #RT User.
C. Start the debugger from the XS command line interface and run the program in SAP Web IDE for SAP HANA.
D. Attach the debugger to the application in the SAP Web IDE for SAP HANA.
Answer: (SHOW ANSWER)
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.

NEW QUESTION: 3
You want to exclusively use declarative logic in an SQL Script procedure. Which of the following operators can you use? There are 2 correct answers to this question.

A. COMM IT statement
B. IF condition
C. TABLE variable
D. MAP_REDUCE function
Answer: (SHOW ANSWER)
To exclusively use declarative logic in an SQL Script procedure, you can use TABLE variables and MAP_REDUCE functions. TABLE variables are variables that store intermediate results of SQL queries in a tabular format. You can use TABLE variables to define data transformations and data flows without using imperative statements or side effects1. MAP_REDUCE functions are functions that apply a map function to each row of a table and then aggregate the results using a reduce function. You can use MAP_REDUCE functions to implement complex algorithms and calculations in a declarative way2.
The other two options, COMMIT statement and IF condition, are not operators that can be used to exclusively use declarative logic in an SQL Script procedure. COMMIT statement is a statement that ends a transaction and makes all changes permanent in the database. You cannot use COMMIT statement in a declarative logic, as it is an imperative statement that has a side effect3. IF condition is a condition that executes a block of code based on a boolean expression. You cannot use IF condition in a declarative logic, as it is an imperative construct that introduces control flow4. References: 1: TABLE Variables | SAP Help Portal 2: MAP_REDUCE Function | SAP Help Portal 3: COMMIT Statement | SAP Help Portal 4: IF Statement | SAP Help Portal

NEW QUESTION: 4
What are the nodes where filter expressions can be used in a calculation view? There are 2 correct answers to this question.
A. Aggregation
B. Star join
C. Union
D. Rank
Answer: A,C (LEAVE A REPLY)
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

NEW QUESTION: 5
To perform a specific task of an XS advanced application, what does a user need? Please choose the correct answer.
A. To be assigned to an Organization
B. To have directly assigned a Scope
C. To have directly assigned a Role Collection
D. To be assigned to a Space
Answer: (SHOW ANSWER)

NEW QUESTION: 6
Which SAP HANA system views contain information about the object ownership? There are 2 correct answers to this question.
A. OBJECTS
B. SCHEMAS
C. TABLES
D. ROLES
Answer: B,C (LEAVE A REPLY)

NEW QUESTION: 7
Which rules apply to declarative logic in SQL Script? There are 2 correct answers to this question.
A. It declares the flow of data.
B. It supports the full set of SQL Script.
C. It defines how data is processed.

D. It is free of side effects.
Answer: A,D (LEAVE A REPLY)

NEW QUESTION: 8
In a calculation view, what are the different types of input mapping available in Table Function? There are 2 correct answers to this question.
A. Variable
B. Constant
C. Formula
D. Data Source Column
Answer: B,D (LEAVE A REPLY)
In a calculation view, the different types of input mapping available in Table Function are Constant and Data Source Column. Constant allows you to map a fixed value to the input parameter of the Table Function. Data Source Column allows you to map a column from another node or data source to the input parameter of the Table Function12.
The other two options, Variable and Formula, are not types of input mapping available in Table Function. Variable is a type of input mapping available in Calculation Scenario, which allows you to map a variable from the calculation view to the input parameter of the Calculation Scenario3. Formula is a type of input mapping available in Scripted Calculation View, which allows you to map a formula expression to the input parameter of the Scripted Calculation View4. References: 1: Use Table Functions in Calculation Views | SAP Help Portal 2: Map Input Parameters | SAP Help Portal 3: Map Input Parameters or Variables | SAP Help Portal 4: Map Input Parameters | SAP Help Portal

NEW QUESTION: 9
Which elements can you specify with the SAP WebIDE for SAP HANA graphical editor for Core Data Services data models? There are 3 correct answers to this question.
A. Entity
B. Synonym
C. Procedure
D. Context
E. Association
Answer: (SHOW ANSWER)
The SAP Web IDE for SAP HANA graphical editor for Core Data Services (CDS) data models allows you to specify the following elements12:
* Entity: An entity is a CDS artifact that defines a data structure with attributes, keys, and associations. An entity can be mapped to a database table or view, or it can be used as a source for another entity or a calculation view. You can create an entity by dragging the entity icon from the palette to the canvas, and then defining its properties and attributes in the properties panel.

* Context: A context is a CDS artifact that defines a namespace for other CDS artifacts, such as entities, views, types, or associations. A context can be used to organize and group related CDS artifacts, and to avoid name conflicts. You can create a context by dragging the context icon from the palette to the canvas, and then defining its name and description in the properties panel. You can also add other CDS artifacts to the context by dragging them from the project explorer or the palette to the context node in the canvas.
* Association: An association is a CDS artifact that defines a relationship between two entities, based on a join condition. An association can be used to navigate from one entity to another, and to filter, aggregate, or project data from the target entity. You can create an association by dragging the association icon from the palette to the canvas, and then connecting it to the source and target entities.
You can also define the cardinality, join condition, and exposed attributes of the association in the properties panel.
The other options are not correct because they are not elements that can be specified with the SAP Web IDE for SAP HANA graphical editor for CDS data models. A synonym is a database object that provides an alternative name for another database object, such as a table, view, or procedure. A synonym is not a CDS artifact, and it cannot be created or edited with the graphical editor. A procedure is a database object that contains a set of SQL statements that perform a specific task. A procedure can be created or edited with the SQL editor, but not with the graphical editor. A procedure can also be exposed as a CDS artifact, but it cannot be specified with the graphical editor. References:
* SAP HANA Platform, SAP HANA Modeling Guide for SAP HANA Web Workbench, Core Data Services
* SAP HANA Platform, SAP HANA Developer Guide for SAP HANA Web IDE, Developing Core Data Services Models, Using the Graphical Editor

NEW QUESTION: 10
You develop a Node.js application for the SAP HANA platform that executes a SQL statement in the SAP HANA database. Which modules do you use?
There are 2 correct answers to this question.
A. @sap/instance-manager
B. @sap/xsenv
C. @sap/hdbext
D. @sap/node-vsi
Answer: B,C (LEAVE A REPLY)
To develop a Node.js application for the SAP HANA platform that executes a SQL statement in the SAP HANA database, you need to use the following modules:
* @sap/xsenv: This module provides functions to load service configuration and credentials from the environment variables of the SAP Cloud Platform or the SAP HANA XS Advanced Model. You can use this module to access the SAP HANA database service information, such as host, port, user, password, and schema.

* @sap/hdbext: This module provides a wrapper for the SAP HANA client for Node.js, which enables you to connect to the SAP HANA database and execute SQL statements. You can use this module to create a database connection pool, execute queries, fetch results, and handle errors.
The other options are not correct because:
* A. @sap/instance-manager: This module provides functions to manage the lifecycle of Node.js applications running on the SAP Cloud Platform or the SAP HANA XS Advanced Model. You can use this module to start, stop, restart, scale, and monitor your applications, but not to execute SQL statements in the SAP HANA database.
* D. @sap/node-vsi: This module provides functions to interact with the SAP HANA Virtual Schema Integrator (VSI), which enables you to access data from remote sources using virtual tables. You can use this module to create, update, delete, and query virtual tables, but not to execute SQL statements in the SAP HANA database.
References:
* SAP HANA Developer Guide for SAP HANA Web IDE, Chapter 5, Section 5.2
* SAP HANA Developer Guide for SAP HANA Web IDE, Chapter 5, Section 5.3
* SAP HANA Developer Guide for SAP HANA Web IDE, Chapter 5, Section 5.4
* SAP HANA Developer Guide for SAP HANA Web IDE, Chapter 5, Section 5.5

NEW QUESTION: 11
What do you use to set up unit testing for your SQL Script procedure? Please choose the correct answer.
A. SQL Script logging procedures
B. A library with language SQLSCRIPT TEST
C. The SQL Script debugger
D. The SQL Script Code Analyzer
Answer: (SHOW ANSWER)
To set up unit testing for your SQL Script procedure, you use a library with language SQLSCRIPT TEST. A library with language SQLSCRIPT TEST is a special type of library that contains SQL Script test cases for testing the behavior and performance of SQL Script procedures. A SQL Script test case is a SQL Script procedure that has the prefix TEST_ and that uses the ASSERT and EXPECT statements to verify the expected results and outputs of the procedure under test. A library with language SQLSCRIPT TEST can be created in the SAP Web IDE for SAP HANA or in the SAP HANA Studio, and can be executed by using the SQL Script Test Framework, which is a tool that runs the test cases and generates a test report.
The following options are not used to set up unit testing for your SQL Script procedure, but for other purposes:
* SQL Script logging procedures: SQL Script logging procedures are procedures that use the APPLY_FILTER and APPLY_AGGREGATION statements to perform logging operations on table variables or cursors. SQL Script logging procedures can be used to

monitor and analyze the execution of SQL Script procedures, but not to test their correctness or functionality.
* The SQL Script debugger: The SQL Script debugger is a tool that allows you to debug SQL Script procedures by setting breakpoints, inspecting variables, and stepping through the code. The SQL Script debugger can be used to identify and fix errors or bugs in SQL Script procedures, but not to verify their expected results or outputs.
* The SQL Script Code Analyzer: The SQL Script Code Analyzer is a tool that analyzes SQL Script code and provides suggestions and recommendations for improving its quality, performance, and maintainability. The SQL Script Code Analyzer can be used to optimize and refactor SQL Script code, but not to test its behavior or functionality.
References:
* [SAP HANA Platform Documentation], SAP HANA SQLScript Reference, Chapter 6: SQLScript Test Framework, pp. 153-158.
* [SAP HANA Platform Documentation], SAP HANA Developer Guide for SAP HANA XS Advanced Model, Chapter 7: Developing SQLScript Procedures, Section 7.4: Testing SQLScript Procedures, pp.
137-140.

NEW QUESTION: 12
You implement a native SAP HANA application using SAP HANA extended application services, advanced model (XS advanced) and SAPUI5. Where is the UI rendering executed? Please choose the correct answer.
A. On the SAPUI5 HTML5 module
B. On the SAP Fiori front-end server
C. On the front-end client device
D. On the XS advanced application server
Answer: C (LEAVE A REPLY)
The UI rendering is executed on the front-end client device, such as a browser or a mobile device, that accesses the SAP HANA application. SAPUI5 is a JavaScript-based UI framework that enables the development of responsive and user-friendly web applications. SAPUI5 applications run in the browser and communicate with the back-end server via RESTful services, such as OData. The SAPUI5 HTML5 module is a component of the multi-target application (MTA) that contains the UI logic and resources, such as views, controllers, models, and libraries. The SAPUI5 HTML5 module is deployed to the XS advanced application server, which serves the static UI files to the front-end client device. The SAP Fiori front-end server is a separate component that provides the SAP Fiori launchpad, a single entry point for SAP Fiori apps, and the SAP Fiori UI components, such as controls, themes, and icons. The SAP Fiori front-end server is not required for SAP HANA native applications, but it can be used to integrate them with other SAP Fiori apps123.
References:

* SAP HANA Platform, Developing Applications with SAP HANA Cloud Platform, Developing Multi-Target Applications, Developing HTML5 Modules
* SAP HANA Platform, SAP HANA Extended Application Services, Advanced Model, Developing and
* Deploying Applications, Developing HTML5 Applications
* SAP HANA Platform, SAP HANA Extended Application Services, Advanced Model, Developing and Deploying Applications, Developing SAP Fiori Applications

NEW QUESTION: 13
Which join type is NOT supported by join optimization (pruning)? Please choose the correct answer.
A. Outer Join
B. Text Join
C. Referential Join
D. Inner Join
Answer: D (LEAVE A REPLY)
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

NEW QUESTION: 14
You need to consume a Node.js service in an SAPUI5 application. In which section of the mta.xml file do you define the variable for the exposed service?
Please choose the correct answer.
A. In the path section of the Node.js module
B. In the provides section of the Node.js module
C. In the properties sect on of the MTA resources
D. In the requires sect on of the HTML5 module
Answer: (SHOW ANSWER)

To consume a Node.js service in an SAPUI5 application, you need to define the variable for the exposed service in the requires section of the HTML5 module in the mta.xml file. This way, you can specify the dependency of the SAPUI5 application on the Node.js service and bind them together. The requires section allows you to reference a resource or a module that provides a service or a capability that your module needs.
You can also define additional properties and parameters for the required element, such as the name of the variable that holds the service URL. References: The MTA Development Descriptor, The MTA Deployment Descriptor, Developing Multi-Target Applications

NEW QUESTION: 15
You create a multi-target application in SAP Web IDE for SAP HANA. Which source code management option do you use? Please choose the correct answer.
A. SVC-based repository
B. Git-based repository
C. SAP HANA repository workspace
D. SAP HANA repository
Answer: B (LEAVE A REPLY)
When you create a multi-target application in SAP Web IDE for SAP HANA, you use a Git-based repository as the source code management option. A Git-based repository allows you to store, manage, and synchronize your code across multiple branches and versions. You can also collaborate with other developers and integrate your changes with the SAP HANA Deployment Infrastructure (HDI). A Git-based repository is the recommended option for developing multi-target applications in SAP Web IDE for SAP HANA. References: Developing Multi-Target Applications, Git Integration, SAP HANA Deployment Infrastructure.

NEW QUESTION: 16
You are asked to produce sales value for the month using sales order line items (sales price per item x quantity sold) what do you use to achieve this in a calculated column of a calculation view? There are 2 correct answers to this question.
A. Keep flag
B. Dynamic join
C. Transparent filter
D. Non-equijoin
Answer: C,D (LEAVE A REPLY)
To produce sales value for the month using sales order line items (sales price per item x quantity sold), you can use a calculated column of a calculation view that performs the multiplication of the two columns and applies a transparent filter and a non-equijoin to restrict the data to the desired month. A transparent filter is a filter condition that is applied to the data source of a view node and is propagated to the subsequent nodes. A non-equijoin is a join condition that uses operators other than equality, such as greater than,

less than, or between. For example, you can use a transparent filter to filter the sales order line items by the month of the order date, and then use a non-equijoin to join the sales order line items with the sales price table by the order date and the valid from and valid to dates of the sales price. References: Transparent Filters, Non-Equijoins, Create Calculated Columns.

NEW QUESTION: 17
You implement a native SAP HANA application using SAP HANA extended application services, advanced model (XS advanced) and SAPUI5. Where is the UI rendering executed? Please choose the correct answer.
A. On the XS advanced application server
B. On the SAPUI5 HTML5 module
C. On the SAP Fiori front-end server
D. On the front-end client device
Answer: (SHOW ANSWER)

NEW QUESTION: 18
What are the characteristics of the calculation view of type "SQL Access Only"? There are 2 correct answers to this question.
A. Can be directly used for multidimensional analysis
B. Can be directly used as modelled data source for another view
C. Can be directly exposed to all client tools
D. Can be direct y accessed via SQL
Answer: (SHOW ANSWER)

NEW QUESTION: 19
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.
A. Rebase
B. Merge
C. Commit
D. Push

Answer: B (LEAVE A REPLY)

NEW QUESTION: 20
Which SAP HANA system views contain information about the object ownership? There are 2 correct answers to this question.
A. SCHEMAS
B. TABLES
C. OBJECTS
D. ROLES
Answer: A,D (LEAVE A REPLY)
The SAP HANA system views that contain information about the object ownership are SCHEMAS and ROLES. The SCHEMAS view provides information about the schemas in the database, including the owner of each schema1. The ROLES view provides information about the roles in the database, including the owner of each role2. The object ownership indicates who has the right to grant or revoke privileges on the object, or to alter or drop the object3.
The other two options, TABLES and OBJECTS, do not contain information about the object ownership. The TABLES view provides information about the tables in the database, but not the owner of each table4. The OBJECTS view provides information about the objects in the database, such as views, procedures, functions, etc., but not the owner of each object5. References: 1: SCHEMAS System View | SAP Help Portal 2: ROLES System View | SAP Help Portal 3: Object Ownership | SAP Help Portal 4: TABLES System View | SAP Help Portal 5: OBJECTS System View | SAP Help Portal

NEW QUESTION: 21
Which of the following are characteristics of database procedures?
A. Database procedures can have both input and output parameters; however, a parameter CANNOT be both input and output.
B. Database procedures can have several output parameters, and a mix of both scalar and table types is possible.
C. If, in the database procedure header, you use the READS SQL DATA option, then INSERT statements are prohibited; however, dynamic SQL is allowed.
D. If, in the database procedure header, you use the SQL SECURITY INVOKER option, then only the owner of the procedure can start it.
Answer: A,B (LEAVE A REPLY)
According to the SAP HANA SQLScript Reference1, database procedures are subroutines that can be called from SQL statements or other database procedures. Database procedures can have the following characteristics:
* Database procedures can have both input and output parameters; however, a parameter CANNOT be both input and output. Input parameters are used to pass values to the procedure, while output parameters are used to return values from the procedure. A

parameter can be either scalar or table type, depending on the data type and cardinality. A parameter cannot be both input and output, because this would create ambiguity and inconsistency in the parameter passing mechanism. For example, you cannot declare a parameter as IN OUT, or assign a value to an input parameter, or read a value from an output parameter. For more information on database procedure parameters, see [Parameters of Database Procedures]2.

* Database procedures can have several output parameters, and a mix of both scalar and table types is possible. Output parameters are used to return values from the procedure to the caller. A procedure can have zero or more output parameters, depending on the purpose and logic of the procedure. Output parameters can be either scalar or table type, depending on the data type and cardinality. A procedure can have a mix of both scalar and table output parameters, as long as they are compatible with the caller's expectations and syntax. For example, you can use a SELECT statement with INTO clause to assign values to scalar output parameters, or use a SELECT statement with RESULT clause to return a table output parameter. For more information on database procedure output parameters, see [Output Parameters of Database Procedures]3.
* If, in the database procedure header, you use the READS SQL DATA option, then INSERT statements are prohibited; however, dynamic SQL is allowed. The READS SQL DATA option is used to indicate that the procedure only reads data from the database, and does not modify or write any data. This option is useful for performance optimization and security enforcement, as it allows the database to apply certain optimizations and checks to the procedure. However, this option also imposes some restrictions on the procedure, such as prohibiting any INSERT, UPDATE, DELETE, or MERGE statements, or any other statements that modify the database state. Dynamic SQL, which is SQL code that is constructed
* and executed at run time, is still allowed, as long as it does not violate the READS SQL DATA option. For more information on the READS SQL DATA option, see [Procedure Header Options]4.
* If, in the database procedure header, you use the SQL SECURITY INVOKER option, then only the owner of the procedure can start it. The SQL SECURITY INVOKER option is used to indicate that the procedure is executed with the privileges of the user who invokes the procedure, rather than the privileges of the user who created the procedure. This option is useful for security enforcement and access control, as it allows the database to apply the appropriate authorization checks and restrictions to the procedure. However, this option does not affect who can start the procedure, as this is determined by the GRANT EXECUTE statement, which grants the execute privilege on the procedure to a specific user or role. The SQL SECURITY INVOKER option only affects how the procedure is executed, not who can execute it. For more information on the SQL SECURITY INVOKER option, see [Procedure Header Options]4.

References: 1: SAP HANA SQLScript Reference 2: Parameters of Database Procedures 3: Output Parameters of Database Procedures 4: Procedure Header Options

NEW QUESTION: 22
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.
A. Installation on Windows Server
B. Installation on Mac OS
C. Installation on Linux OS
D. Usage of virtual machine on Microsoft Windows
Answer: C,D (LEAVE A REPLY)
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

NEW QUESTION: 23
You need to consume a Node.js service in an SAPUI5 application. In which section of the mta.xml file do you define the variable for the exposed service?
Please choose the correct answer.
A. In the path section of the Node.js module
B. In the provides section of the Node.js module
C. In the requires sect on of the HTML5 module
D. In the properties sect on of the MTA resources
Answer: C (LEAVE A REPLY)

NEW QUESTION: 24
What are some of the advantages of SQLScript compared to plain SQL queries? There are 2 correct answers to this question.
A. It enables modular parameterized programing.
B. It increases performance by parallel execution of complex SQL statements.
C. It retrieves a high data volume from a database.
D. It contains a well-defined ANSI standard.
Answer: A,B (LEAVE A REPLY)

NEW QUESTION: 25
You develop an OData service using XSODATA. Which HTTP commands can you use to define the operations on the resources? There are 3 correct answers to this question.

A. DPUT
B. READ
C. INSERT
D. POST
E. DGET
Answer: A,D,E (LEAVE A REPLY)
According to the SAP HANA Developer Guide, XSODATA supports the following HTTP commands to define the operations on the resources:
* DGET: Retrieves a single entity or a collection of entities from the data source.
* DPUT: Updates an existing entity in the data source.
* POST: Creates a new entity in the data source.
* DELETE: Deletes an existing entity from the data source.
The READ and INSERT commands are not valid HTTP commands for XSODATA. They are keywords that can be used in the service definition file (.xsodata) to specify the access mode for each entity set. References:
SAP HANA Developer Guide, Chapter 6, Section 6.4.2, page 2111

NEW QUESTION: 26
Your multi-target application will use XSJS. Which module type do you create in your application project? Please choose the correct answer.
A. SAP HANA database
B. HTML5
C. Java
D. Node.js
Answer: D (LEAVE A REPLY)

NEW QUESTION: 27
You need to initially load data from a .csv file into a Core Data Services table in SAP HANA extended application services, advanced model (XS advanced). Which file type do you create? Please choose the correct answer.
A. A file with extension .hdbtable data
B. A fie with extension .hdbtable
C. A file with extension .hdbdd
D. A file with extension .hdbti
Answer: (SHOW ANSWER)
To initially load data from a .csv file into a Core Data Services (CDS) table in SAP HANA extended application services, advanced model (XS advanced), you need to create a file with extension .hdbti, which stands for HDB Table Import. This file defines the configuration and mapping for importing data from a local or remote file into a table in the HDI container. You can specify the source file name, location, format, delimiter, header, encoding, etc., as well as the target table name, schema, and column mapping. You can

also specify whether to truncate the target table before importing, or to append the data to the existing table. You can use the SAP Web IDE for SAP HANA to create and deploy the .hdbti file, or use the HDI deployer CLI tool. The other options are incorrect because they are not the correct file types for importing data into a CDS table. A file with extension .hdbtabledata is a file that contains the data for a table in a JSON format, but it is not used for importing data from a .csv file. A file with extension .hdbtable is a file that defines the structure and properties of a table, but it does not contain any data. A file with extension .hdbdd is a file that defines the CDS entities, such as contexts, types, views, associations, etc., but it does not import any data. References:
* Importing Data into Tables - SAP Help Portal
* Creating Table Import Configuration Files - SAP Help Portal
* HDI Deployer CLI Tool - SAP Help Portal
* [HA300 - SAP HANA 2.0 SPS06 Modeling] - SAP Training

NEW QUESTION: 28
You need to synchronize all unsynchronized changes of the remote Git branch with a local Git branch. Which operation do you use? Please choose the correct answer.
A. Push
B. Rebase
C. Pull
D. Reset
Answer: (SHOW ANSWER)
According to the SAP HANA Developer Guide, you use the pull operation to synchronize all unsynchronized changes of the remote Git branch with a local Git branch. The pull operation fetches the changes from the remote branch and merges them with the local branch. This way, you can update your local branch with the latest changes from the remote branch. The other options are incorrect, because:
* Push is the opposite of pull. It sends the changes from the local branch to the remote branch, not the other way around.
* Rebase is an alternative to merge. It rewrites the history of the local branch by applying the changes from the remote branch as new commits on top of the local branch. This creates a linear history, but it may cause conflicts or lose some information.
* Reset is a way to undo changes in the local branch. It moves the branch pointer to a different commit, and optionally discards the changes in the working directory or the staging area.
References: SAP HANA Developer Guide, Chapter 6, Section 6.4.2, page 2111.

NEW QUESTION: 29
Which of the following joins can be defined only with multi-column join in calculation views? Please choose the correct answer.
A. Dynamic join

B. Text join
C. Temporal join
D. Multi join
Answer: (SHOW ANSWER)
According to the SAP HANA Developer Guide, a multi join is a join that can be defined only with multi-column join in calculation views. A multi join allows you to join multiple data sources using multiple join conditions. You can specify the join type, the join cardinality, and the join columns for each pair of data sources. A multi join is useful when you need to join data sources that have complex relationships or different granularities. The other options are incorrect, because:
* A dynamic join is a join that can be defined with either single-column join or multi-column join in calculation views. A dynamic join allows you to join two data sources using a join condition that is evaluated at run time based on the user input. You can specify the join type, the join cardinality, and the join columns as parameters or variables.
* A text join is a join that can be defined with either single-column join or multi-column join in calculation views. A text join allows you to join a data source with a text table that contains language-dependent descriptions of the data. You can specify the join type, the join cardinality, and the join columns as well as the language column and the language code.
* A temporal join is a join that can be defined with either single-column join or multi-column join in calculation views. A temporal join allows you to join a data source with a history table that contains time-dependent versions of the data. You can specify the join type, the join cardinality, and the join columns as well as the validity columns and the validity period.
References: SAP HANA Developer Guide, Chapter 6, Section 6.4.2, page 2111.

NEW QUESTION: 30
What are the issue categories that SQL Script Code Analyser scans for? There are 3 correct answers to this question.
A. Performance
B. Privileges
C. Security
D. Business logic
E. Consistency
Answer: (SHOW ANSWER)
The SQL Script Code Analyser is a tool that scans SQL Script code for potential issues and provides suggestions for improvement. The tool can be accessed from the SAP Web IDE for SAP HANA or from the SAP HANA database explorer. The tool scans for issues in the following categories:
* Performance: This category covers issues that may affect the execution time or resource consumption of the SQL Script code, such as inefficient joins, missing indexes,

unnecessary data transfers, etc. The tool provides recommendations for optimizing the code, such as using parallel processing, avoiding nested loops, using table variables, etc.
* Security: This category covers issues that may expose the SQL Script code to security risks, such as SQL injection, privilege escalation, data leakage, etc. The tool provides recommendations for securing the code, such as using parameterized queries, escaping literals, applying proper authorization checks, etc.
* Consistency: This category covers issues that may affect the readability, maintainability, or portability of the SQL Script code, such as coding style, naming conventions, documentation, compatibility, etc.
* The tool provides recommendations for improving the code quality, such as using consistent indentation, avoiding hard-coded values, adding comments, following best practices, etc.
The other options are incorrect because privileges and business logic are not issue categories that the SQL Script Code Analyser scans for. Privileges are handled by the SAP HANA Deployment Infrastructure framework, which automatically grants the necessary privileges to the technical users and roles that access the HDI container objects. Business logic is the domain-specific logic that defines the functionality and behavior of the SQL Script code, and it is not within the scope of the tool to verify or validate it. References:
* SQL Script Code Analyzer - SAP Help Portal
* SQL Script Code Analyzer in SAP Web IDE for SAP HANA - SAP Help Portal
* SQL Script Code Analyzer in SAP HANA Database Explorer - SAP Help Portal
* HA150 - SAP HANA 2.0 SPS06 SQLScript for SAP HANA - SAP Training

NEW QUESTION: 31
What is the package descriptor package.json used for? There are 2 correct answers to this question.
A. To define the routes
B. To set the router version
C. To list the package dependencies
D. To define back-end destinations
Answer: (SHOW ANSWER)

NEW QUESTION: 32
What happens if you define a database object in the persistence model and deploying using the SAP HANA Deployment Infrastructure (HDI)?
Please choose the correct answer.
A. A descriptive representation of the object is created in Hot during runtime.
B. The HDI container is created when the design-time object is saved.
C. The corresponding runtime object is created in the HDI container on deployment
D. The corresponding runtime object is created in the database module of the multi-target application during runtime.
Answer: B (LEAVE A REPLY)

NEW QUESTION: 33
You create a multi-target application in SAP Web IDE for SAP HANA. Which source code management option do you use? Please choose the correct answer.
A. Git-based repository
B. SAP HANA repository
C. SAP HANA repository workspace
D. SVC-based repository
Answer: A (LEAVE A REPLY)

NEW QUESTION: 34
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.
A. Push the calculations onto the application layer.
B. Push the calculations onto the database layer
C. Push the calculations onto the presentation layer.
D. Distribute calculations between application layer and presentation layer.
Answer: B (LEAVE A REPLY)
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

NEW QUESTION: 35
You work on a multi-target application that uses the SAP HANA Deployment Infrastructure (HDI). Which operations can you perform in the SAP Web IDE for SAP HANA?
There are 2 correct answers to this question.
A. Deploy HTML5 modules to the SAP HANA database.
B. Create objects in the SAP HANA repository.
C. Configure the artifact namespaces.
D. Choose the supported SAP HANA versions.
Answer: A,B (LEAVE A REPLY)

NEW QUESTION: 36
You create an SAP HANA extended application services, advanced model (XS advanced) multi-target application. What does the mta.xml file contain?
Please choose the correct answer.
A. The Node.js package definition with dependencies and initial run configurations
B. The plugin name and version for every supported file suffix
C. The application global elements, modules, resources, properties, and parameters
D. The application router description with the authentication methods and access routes
Answer: C (LEAVE A REPLY)

NEW QUESTION: 37
You need to edit a database procedure. You are concerned that the version of the database procedure file you have in your local Git repository may not be synchronized with the latest version in the remote Git repository, but at this time, you do NOT want to sync

the file. You only want to see the commits applied to the file in the remote repository. Which command must you use?

A. Git merge

B. Git pull

C. Git stash

D. Git fetch

Answer: C (LEAVE A REPLY)

NEW QUESTION: 38

What is the package descriptor package.json used for? There are 2 correct answers to this question.

A. To define back-end destinations

B. To set the router version

C. To list the package dependencies

D. To define the routes

Answer: (SHOW ANSWER)

The package descriptor package.json is a JSON file that defines the build, deployment, and runtime dependencies of a JavaScript application in SAP HANA XS Advanced. The package.json file is mandatory for JavaScript applications and it is located in the general section of the project. As well as the application name and version, dependencies to other Node.js modules, the Node.js version, run scripts, and the main program are configured.

The package descriptor package.json is used for the following purposes:

* To list the package dependencies: The package.json file contains a dependencies property that lists the names and versions of the Node.js modules that the application depends on. These modules are installed by the npm install command during the build process and are available in the node_modules folder of the application. The dependencies property can also specify the scope and type of the dependencies, such as devDependencies for development-only modules, or peerDependencies for modules that are required by other modules.

* To define the routes: The package.json file contains a sap.cloud.service property that defines the name of the service that the application provides or consumes. This name is used to generate the routes for the application in the XS advanced environment. The routes are defined in the xs-app.json file, which is located in the same folder as the package.json file. The routes specify the rules for forwarding requests to the back-end microservices or destinations.

The following purposes are not achieved by the package descriptor package.json, but by other files or tools:

* To define back-end destinations: The back-end destinations are defined in the mta.yaml file, which is the deployment descriptor file that specifies the metadata and dependencies for the multi-target application (MTA) project. The mta.yaml file is located in the root folder of the MTA project and is used by the Cloud Foundry environment to deploy the

application. The back-end destinations are declared as resources of type org.cloudfoundry.existing-service or org.cloudfoundry.managed-service, and are bound to the application modules by the requires property.
* To set the router version: The router version is set by the @sap/approuter module, which is a Node.js module that provides the application router service for the XS advanced environment. The application router service is responsible for routing requests to the appropriate destinations and for authenticating users. The @sap/approuter module is installed by the npm install command during the build process and is available in the node_modules folder of the application. The version of the @sap/approuter module is specified in the dependencies property of the package.json file.
References:
* [SAP HANA Deployment Infrastructure Reference], Chapter 5: HDI with XS Advanced, Section 5.1:
Developing with the SAP Web IDE for SAP HANA, Subsection 5.1.2: Configure Application Routing (xs-app.json), pp. 101-104.
* [SAP HANA Platform Documentation], SAP HANA Developer Guide for SAP HANA XS Advanced Model, Chapter 4: Developing HTML5 Applications, Section 4.1: Developing HTML5 Applications Using SAP Web IDE for SAP HANA, Subsection 4.1.3: Configure Application Routing (xs-app.json), pp. 77-80.

NEW QUESTION: 39
What types of user-defined functions are supported by SAP HANA? Note: There are 2 correct answers to this question.
A. Table
B. Scalar
C. Aggregate
D. Hierarchy
Answer: (SHOW ANSWER)
SAP HANA supports two types of user-defined functions: table and scalar. User-defined functions are functions that are created by users to perform specific tasks or calculations on data. User-defined functions can be used in SQL expressions, such as in the SELECT, WHERE, or GROUP BY clauses of SQL statements.
User-defined functions can also be used in SQLScript, which is a scripting language for SAP HANA that extends SQL with procedural and analytical features.
Table functions are user-defined functions that return a table as the output. Table functions can be used to implement complex logic or transformations on data, such as pivoting, unpivoting, aggregation, or ranking.
Table functions can also be used to create calculation views, which are virtual data models that expose data from SAP HANA tables or views. Table functions are created by using the CREATE FUNCTION statement with the RETURNS <return_table_type> clause, where

<return_table_type> specifies the table type of the output. Table functions can be called in the FROM clause of SQL statements, or in the node expression of calculation views.
Scalar functions are user-defined functions that return a scalar value as the output. Scalar functions can be used to perform simple calculations or validations on data, such as converting units, checking conditions, or generating IDs. Scalar functions are created by using the CREATE FUNCTION statement with the RETURNS <return_parameter_list> clause, where <return_parameter_list> specifies the scalar type of the output. Scalar functions can be called in the SELECT or WHERE clauses of SQL statements, or in the expression of calculated columns.
The following types of functions are not supported by SAP HANA as user-defined functions, but as built-in functions or features:
* Aggregate functions: Aggregate functions are functions that perform calculations on a set of values and return a single value. For example, SUM, COUNT, MIN, MAX, or AVG are aggregate functions.
Aggregate functions are built-in functions that are provided by SAP HANA and can be used in SQL expressions, such as in the SELECT or GROUP BY clauses of SQL statements, or in the expression of aggregated columns. Aggregate functions cannot be created by users as user-defined functions.
* Hierarchy functions: Hierarchy functions are functions that operate on hierarchical data, such as parent-child or level-based hierarchies. For example, HIERARCHY_ANCESTORS, HIERARCHY_DESCENDANTS, or HIERARCHY_LEVEL are hierarchy functions. Hierarchy functions are built-in functions that are provided by SAP HANA and can be used in SQL expressions, such as in the SELECT or WHERE clauses of SQL statements, or in the expression of hierarchy columns. Hierarchy functions cannot be created by users as user-defined functions.
References:
* [SAP HANA Platform Documentation], SAP HANA SQL and System Views Reference, Chapter 3:
SQL Expressions, Section 3.5: Function Expressions, pp. 59-60.
* [SAP HANA Platform Documentation], SAP HANA SQLScript Reference, Chapter 6: SQLScript Test Framework, Section 6.1: User-Defined Functions, pp. 153-154.

NEW QUESTION: 40
What characterizes the Node.js environment? There are 2 correct answers to this question.
A. It is built on Google's V8 JavaScript engine.
B. It is single-threaded.
C. It is a client-side design-time environment for JavaScript.
D. It uses a synchronous programming model.
Answer: A,B (LEAVE A REPLY)

NEW QUESTION: 41
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.
A. Java Runtime Environment (JRE)
B. SAP HANA Runtime Tools (HRTT)
C. SAP Web IDE for SAP HANA
D. HANA Deployment Infrastructure Container
Answer: D (LEAVE A REPLY)
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

NEW QUESTION: 42
Why do you use associations in your persistence model for SAP HANA extended application services, advanced model (XS advanced)? Please choose the correct answer.

A. To enforce database-level constraints
B. To define the relationships between contexts and views
C. To define multiplicity and key of an entity relation
D. To set relationships between multiple contexts
Answer: (SHOW ANSWER)
Associations are used in the persistence model for SAP HANA extended application services, advanced model (XS advanced) to define the multiplicity and key of an entity relation. An entity relation is a logical connection between two entities, such as one-to-one, one-to-many, or many-to-many. Multiplicity specifies how many instances of one entity can be related to one instance of another entity. Key specifies which attributes of the related entities are used to identify and join the instances. Associations are defined using the Core Data Services (CDS) language, which is a declarative language for modeling data structures and services. Associations are part of the CDS entity definition, and they can be used to navigate from one entity to another, and to filter, aggregate, or project data from the target entity12.
The other options are not correct because they are not the purposes of using associations in the persistence model for XS advanced. Associations do not enforce database-level constraints, such as primary keys, foreign keys, or check constraints. These constraints are defined separately using the CDS annotations or the SQL DDL statements. Associations do not define the relationships between contexts and views, which are different types of CDS artifacts. A context is a namespace for grouping related CDS artifacts, such as entities, views, types, or associations. A view is a CDS artifact that defines a data model based on one or more entities or other views, and applies filters, joins, aggregations, or other operations on the data. Associations do not set relationships between multiple contexts, which are independent from each other and do not have any logical connection. References:
* SAP HANA Platform, SAP HANA Modeling Guide for SAP HANA Web Workbench, Core Data Services
* SAP HANA Platform, SAP HANA Developer Guide for SAP HANA Web IDE, Developing Database Modules, Developing Core Data Services Models, Defining Associations

NEW QUESTION: 43
Which of the following elements can be part of the UI5 application's index.html file? There are 2 correct answers to this question
A. Backend Connection
B. Framework Reference
C. UI-Area
D. Bootstrap
Answer: (SHOW ANSWER)

NEW QUESTION: 44

You create an SAP HANA extended application services, advanced model (XS advanced) multi-target application. What does the mta.xml file contain?
Please choose the correct answer.
A. The plugin name and version for every supported file suffix
B. The application router description with the authentication methods and access routes
C. The Node.js package definition with dependencies and initial run configurations
D. The application global elements, modules, resources, properties, and parameters
Answer: D (LEAVE A REPLY)
The mta.xml file is the main descriptor file for a multi-target application (MTA). It defines the application global elements, such as ID, version, provider, and schema version. It also defines the modules, which are the components of the application that provide specific functionality, such as a database module, a Node.js module, or an HTML5 module. Each module has a name, a type, a path, and a list of required dependencies.
The mta.xml file also defines the resources, which are the shared services or configurations that the modules can consume, such as a user-provided service, an SAP HANA database, or an SAP HANA HDI container.
Each resource has a name, a type, and optional parameters. The mta.xml file also defines the properties and parameters, which are the key-value pairs that can be used to configure the modules and resources, such as the host, port, or credentials. The properties and parameters can be defined globally, per module, or per resource12.
The other options are not correct because they are not part of the mta.xml file, but rather of other files or components of the MTA. The plugin name and version for every supported file suffix are defined in the .che file, which is the configuration file for the SAP Web IDE3.
The application router description with the authentication methods and access routes are defined in the xs-app.json file, which is the configuration file for the application router, a Node.js module that provides a single entry point and routing mechanism for the MTA. The Node.js package definition with dependencies and initial run configurations are defined in the package.json file, which is the configuration file for the Node.js module that contains the application logic and business services. References:
* SAP HANA Platform, Developing Applications with SAP HANA Cloud Platform, Developing Multi-Target Applications
* SAP HANA Platform, SAP HANA Extended Application Services, Advanced Model, Developing and Deploying Applications, Descriptor Files, mta.xml
* SAP HANA Platform, Developing Applications with SAP HANA Cloud Platform, Developing Multi-Target Applications, Descriptor Files, .che
* [SAP HANA Platform], SAP HANA Extended Application Services, Advanced Model, Developing and Deploying Applications, Descriptor Files, xs-app.json
* [SAP HANA Platform], SAP HANA Extended Application Services, Advanced Model, Developing and Deploying Applications, Descriptor Files, package.json

NEW QUESTION: 45

What are the different types of graphical calculation views that can handle measures? There are 2 correct answers to this question.

A. SQL Access Only

B. Dimension

C. Cube with star join

D. Cube

Answer: (SHOW ANSWER)

NEW QUESTION: 46

What characterizes the Node.js environment? There are 2 correct answers to this question.

A. It uses a synchronous programming model.

B. It is a client-side design-time environment for JavaScript.

C. It is single-threaded.

D. It is built on Google's V8 JavaScript engine.

Answer: C,D (LEAVE A REPLY)

According to the SAP Web IDE for SAP HANA Developer Guide, the Node.js environment is characterized by the following features:

* It is single-threaded: Node.js uses a single thread to handle multiple concurrent requests, instead of creating a new thread for each request. This reduces the memory and CPU overhead, and enables high scalability and performance. However, it also means that any blocking or long-running operation can affect the responsiveness of the entire application. Therefore, Node.js relies on asynchronous and non-blocking I/O operations, callbacks, promises, and events to handle concurrency and avoid blocking the main thread.

* It is built on Google's V8 JavaScript engine: Node.js uses the V8 engine to execute JavaScript code.

The V8 engine is a fast and powerful engine that compiles JavaScript code into native machine code, and supports the latest ECMAScript standards and features. The V8 engine also provides access to low-level system resources, such as files, network, and processes, through the Node.js API.

The other options are incorrect, because:

* Node.js does not use a synchronous programming model, but an asynchronous and event-driven programming model. A synchronous programming model means that each operation blocks the execution until it is completed, and the next operation can only start after the previous one is finished.

An asynchronous programming model means that each operation can start without waiting for the previous one to finish, and the execution can continue with other tasks while the operation is in progress.

An event-driven programming model means that each operation can trigger an event when it is completed, and the event can invoke a callback function that handles the result or the error of the operation.

* Node.js is not a client-side design-time environment for JavaScript, but a server-side run-time environment for JavaScript. A client-side design-time environment for JavaScript means that the JavaScript code is written and executed in the browser, and it can manipulate the HTML document and interact with the user interface. A server-side run-time environment for JavaScript means that the JavaScript code is written and executed on the server, and it can handle HTTP requests and responses, communicate with databases, and perform business logic.
References: SAP Web IDE for SAP HANA Developer Guide, Chapter 6, Section 6.4.2, page 2111.

NEW QUESTION: 47
You develop a Node.js application for the SAP HANA platform that executes a SQL statement in the SAP HANA database. Which modules do you use?
There are 2 correct answers to this question.
A. @sap/xsenv
B. @sap/instance-manager
C. @sap/hdbext
D. @sap/node-vsi
Answer: A,C (LEAVE A REPLY)

NEW QUESTION: 48
Which of the following information must you specify when you create a new project in SAP Web IDE for SAP HANA? There are 2 correct answers to this question.
A. Template
B. SAP HANA database version
C. Namespace
D. Project name
Answer: C,D (LEAVE A REPLY)

NEW QUESTION: 49
To which SAP HANA authorization entities can you grant a role? There are 2 correct answers to this question.

A. Role
B. User
C. Object
D. Privilege
Answer: A,B (LEAVE A REPLY)

NEW QUESTION: 50
Which SAP HANA extended application services, advanced model (XS advanced) tools can you use to retrieve the URL for the SAP WebIDE for SAP HANA?
There are 2 correct answers to this question.
A. Runtime tool
B. XS advanced Controller Start Screen
C. XS Advanced command-line client tool
D. Organization and Space Management tool
Answer: B,C (LEAVE A REPLY)
The SAP Web IDE for SAP HANA is a web-based development tool that allows you to create and manage multi-target applications (MTAs) that use the SAP HANA Deployment Infrastructure (HDI). The SAP Web IDE for SAP HANA is itself an XS advanced application that runs on the XS advanced runtime and can be accessed via a URL. To retrieve the URL for the SAP Web IDE for SAP HANA, you can use the following XS advanced tools:
* XS advanced Controller Start Screen: This is a web-based tool that provides an overview of the XS advanced environment and the available applications and services. You can access the XS advanced Controller Start Screen by opening the URL https://<host>:3<instance_number>30/ in a browser, where
<host> is the host name of your SAP HANA system and <instance_number> is the instance number of your SAP HANA system. From the XS advanced Controller Start Screen, you can see the list of applications and their URLs under the Applications tab. You can find the SAP Web IDE for SAP HANA application by its name (webide) and click on its URL to open it in a new tab.
* XS Advanced command-line client tool: This is a command-line tool that allows you to perform various operations on the XS advanced environment and the applications and services. You can access the XS Advanced command-line client tool by opening a terminal and entering the command xs. To retrieve the URL for the SAP Web IDE for SAP HANA, you need to first log in to the XS advanced environment by entering the command xs login and providing your credentials. Then, you need to switch to the SAP space by entering the command xs target -s SAP, where the SAP Web IDE for SAP HANA application is deployed. Finally, you need to list the applications and their URLs by entering the command xs apps.
You can find the SAP Web IDE for SAP HANA application by its name (webide) and copy its URL to open it in a browser.

The following tools are not used to retrieve the URL for the SAP Web IDE for SAP HANA, but for other purposes:
* Runtime tool: This is a web-based tool that allows you to monitor and manage the runtime components of the XS advanced environment, such as applications, services, processes, and logs. You can access the Runtime tool by opening the URL https://<host>:3<instance_number>32/ in a browser, where <host> is the host name of your SAP HANA system and <instance_number> is the instance number of your SAP HANA system. From the Runtime tool, you can see the status and details of the applications and services, but not their URLs.
* Organization and Space Management tool: This is a web-based tool that allows you to create and manage the organizational units of the XS advanced environment, such as organizations and spaces.
You can access the Organization and Space Management tool by opening the URL https://<host>:3<instance_number>31/ in a browser, where <host> is the host name of your SAP HANA system and <instance_number> is the instance number of your SAP HANA system. From the Organization and Space Management tool, you can see the list of organizations and spaces and their quotas and roles, but not the applications and their URLs.
References:
* [SAP HANA Deployment Infrastructure Reference], Chapter 5: HDI with XS Advanced, Section 5.1:
Developing with the SAP Web IDE for SAP HANA, Subsection 5.1.1: Create a Project, p. 101.
* [SAP HANA Platform Documentation], SAP HANA Developer Guide for SAP HANA XS Advanced Model, Chapter 2: Getting Started with SAP Web IDE for SAP HANA, Section 2.1: Creating a Project, p. 17.

NEW QUESTION: 51
You configure an OData service for a transactional SAPUI5 application. What is the default data access behavior of the OData service? Please choose the correct answer.
A. Read: Yes Create: Yes Update: Yes Delete: Yes
B. Read: Yes Create: Yes Update: Yes Delete: No
C. Read: Yes, Create: No Update: Yes Delete: No
Answer: (SHOW ANSWER)
Read: Yes, Create: No Update: No Delete: No

NEW QUESTION: 52
To perform a specific task of an XS advanced application, what does a user need? Please choose the correct answer.
A. To have directly assigned a Role Collection
B. To have directly assigned a Scope

C. To be assigned to an Organization
D. To be assigned to a Space
Answer: (SHOW ANSWER)
According to the SAP HANA Developer Guide, to perform a specific task of an XS advanced application, a user needs to have directly assigned a role collection. A role collection is a set of roles that grant the user the necessary privileges and authorizations to access and use the application. A role collection can be assigned to a user either directly by an administrator, or indirectly by a workflow or a self-service. The other options are incorrect, because:
* To have directly assigned a scope is not a way to perform a specific task of an XS advanced application, but a way to limit the access of a user to a subset of resources within an application. A scope is a qualifier that can be added to a role to restrict the user's access to a specific resource, such as a space, an organization, or a service instance. A scope does not grant any privileges or authorizations by itself, but only modifies the existing ones.
* To be assigned to an organization is not a way to perform a specific task of an XS advanced application,
* but a way to group users and resources in a logical unit. An organization is a container that holds spaces, users, applications, and services. An organization can have one or more spaces, which are isolated environments for developing, deploying, and running applications. A user can belong to one or more organizations, but can only perform tasks in the spaces that they are assigned to.
* To be assigned to a space is not a way to perform a specific task of an XS advanced application, but a way to access a specific environment for developing, deploying, and running applications. A space is a sub-container within an organization that holds applications and services. A space can have one or more users, who can have different roles and permissions depending on their tasks. A user can belong to one or more spaces within an organization, but can only perform tasks in the spaces that they are assigned to.
References: SAP HANA Developer Guide, Chapter 6, Section 6.4.2, page 2111.

NEW QUESTION: 53
You implement an asynchronous Node.js service that sends requests to the database.
How do you handle the database responses? There are 2 correct answers to this question.
A. Process the callback methods in the order in which the requests are returned from the database.
B. Process the callback methods in the order in which the requests are sent by the service.
C. Call a method directly after every send request.
D. Register a callback method for every sent request.
Answer: A,D (LEAVE A REPLY)

NEW QUESTION: 54

You are asked to produce sales value for the month using sales order line items (sales price per item x quantity sold) what do you use to achieve this in a calculated column of a calculation view? There are 2 correct answers to this question.
A. Non-equijoin
B. Transparent filter
C. Keep flag
D. Dynamic join
Answer: B,D (LEAVE A REPLY)

NEW QUESTION: 55
You need to combine data from different database tables within your OData service. Which capability do you use for this relationship? Please choose the correct answer.
A. Entity sets
B. Associations
C. Aggregations
D. Projections
Answer: B (LEAVE A REPLY)

NEW QUESTION: 56
You create a simple data model to store user information. The model contains two tables: one for the user information, and the other for user address details.
The user can only have one address. You also need a view to show the complete information of the user including the address.
Which actions do you perform if you use CDS artifacts only? Please choose the correct answer.
A. Define a User Details view using select on the user entity and using the association to read the address information.
Create two entities for the user and address.
Extend the user entity with address columns.
B. Create a view named User Details using select on the user entity.
x-s Create two entities for the user and address.
Create a User Details calculation view and join the user and address information.
C. Create two entities for the user and address.
Create an association in the user entity to the address entity.
Define a User Details view using select on the address entity and using the association to read the user information.
D. Create two entities for the user and address.
Create an association in the user entity to the address entity.
Answer: D (LEAVE A REPLY)

NEW QUESTION: 57

Which join type is NOT supported by join optimization (pruning)? Please choose the correct answer.
A. Inner Join
B. Text Join
C. Outer Join
D. Referential Join
Answer: A (LEAVE A REPLY)

NEW QUESTION: 58
Which OData capacity do you use when you need to restrict the number or selection of exposed columns?
Please choose the correct answer.
A. Parameter entity sets
B. Aggregation
C. Key specification
D. Property projection
Answer: D (LEAVE A REPLY)
Property projection is an OData capability that allows you to restrict the number or selection of exposed columns in an OData service. Property projection is achieved by using the $select query option, which specifies a subset of properties to be included in the response. The $select query option can be applied to a single entity, a collection of entities, or a complex type. Property projection can be used to reduce the payload size and improve the performance of the OData service.
For example, suppose you have an OData service that exposes a Products entity set with the following properties: ID, Name, Category, Price, and Description. If you want to restrict the number or selection of exposed columns to only ID and Name, you can use the $select query option as follows:
GET /Products?$select=ID,Name
The result is: { "@odata.context": "$metadata#Products(ID,Name)", "value": [{ "ID": 1, "Name": "Laptop" }, { "ID": 2, "Name": "Mouse" }, { "ID": 3, "Name": "Keyboard" }] }
The following OData capabilities are not used to restrict the number or selection of exposed columns, but for other purposes:
* Parameter entity sets: Parameter entity sets are a way to define entity sets that require one or more parameters to be specified in the request. Parameter entity sets can be used to implement function imports or actions that return a collection of entities. Parameter entity sets can also be used to filter or sort the results based on the parameters.
* Aggregation: Aggregation is a way to apply aggregate functions, such as sum, count, min, max, or average, to the properties of an entity set or a complex type. Aggregation can be used to perform calculations or analysis on the data. Aggregation is achieved by using the $apply query option, which specifies a transformation pipeline with various operators, such as groupby, aggregate, filter, or order by.

* Key specification: Key specification is a way to define the key properties of an entity type, which uniquely identify an entity instance within an entity set. Key specification is part of the entity type definition in the metadata document of the OData service. Key specification can be used to retrieve a single entity by its key values.
References:
* [OData Version 4.0 Part 2: URL Conventions], Section 5.1.1: System Query Option $select, pp. 10-11.
* [OData Version 4.0 Part 3: Common Schema Definition Language (CSDL)], Section 13: Entity Model, Subsection 13.2: Entity Sets, pp. 88-89.

NEW QUESTION: 59
You create a simple data model to store user information. The model contains two tables: one for the user information, and the other for user address details.
The user can only have one address. You also need a view to show the complete information of the user including the address.
Which actions do you perform if you use CDS artifacts only? Please choose the correct answer.
A. Create two entities for the user and address.
Create an association in the user entity to the address entity.
Define a User Details view using select on the address entity and using the association to read the user information.
B. Create two entities for the user and address.
Create an association in the user entity to the address entity.
C. Define a User Details view using select on the user entity and using the association to read the address information.
Create two entities for the user and address.
Extend the user entity with address columns.
D. Create a view named User Details using select on the user entity.
x-s Create two entities for the user and address.
Create a User Details calculation view and join the user and address information.
Answer: C (LEAVE A REPLY)
The correct action is to define a User Details view using select on the user entity and using the association to read the address information. This way, you can leverage the power of CDS associations to create a join between the user and address entities without explicitly specifying the join condition. You can also use the association path to access the address fields in the view definition. This approach simplifies the data model and avoids redundancy. Option A is incorrect because it selects from the address entity instead of the user entity, which is not the desired output. Option B is incorrect because it extends the user entity with address columns, which is unnecessary and violates the normalization principle. Option D is incorrect because it creates a calculation view instead of a CDS view, which is not the requirement. Calculation views are graphical or scripted views that can be

used for complex calculations and data transformations, but they are not CDS artifacts.
References:
* SAP HANA 2.0 SPS06 - Application Development for SAP HANA1, Section 2.2.2, p. 32-33
* SAP HANA 2.0 SPS06 - Core Data Services for SAP HANA2, Section 2.3, p. 25-26
* SAP HANA 2.0 SPS06 - Calculation View Modeling3, Section 1.1, p. 7

NEW QUESTION: 60
Which application runtimes are currently supported by the SAP Cloud Application Programming Model? There are 2 correct answers to this question.
A. Python
B. Node.js
C. Java
D. Go
Answer: B,C (LEAVE A REPLY)

NEW QUESTION: 61
Which metadata declarations can you specify in a SQL Script procedure when you use the SAP HANA Deployment Infrastructure? There are 2 correct answers to this question.
A. Default schema
B. Security
C. Language
D. Authorization
Answer: B,C (LEAVE A REPLY)

NEW QUESTION: 62
From which tool can you run the XS command line? Please choose the correct answer.
A. SAP HANA studio
B. SAP WebIDE for SAP HANA
C. Local operating system shell
D. SAP HANA cockpit
Answer: C (LEAVE A REPLY)

NEW QUESTION: 63
Which artifacts do you define in the persistence model to access objects in an external SAP HANA Deployment Infrastructure container? Please choose the correct answer.
A. Views
B. Imports
C. Procedures
D. Synonyms
Answer: D (LEAVE A REPLY)

NEW QUESTION: 64
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.
A. SAP HANA repository
B. SAP HANA studio
C. Git repository
D. SAP Web IDE for SAP HANA
Answer: C,D (LEAVE A REPLY)
SAP HANA extended application services, advanced model (XS advanced) provides a comprehensive platform for the development and execution of micro-service oriented applications, taking advantage of SAP HANA's in-memory architecture and parallel execution capabilities1. XS advanced supports several programming languages and execution environments, such as Java and Node.js, and also provides compatibility with XS classic model2. To develop native SAP HANA applications in XS advanced, you can use the following new components:
* Git repository: XS advanced integrates with Git, a distributed version control system, to manage the source code of your applications. You can use Git to clone, create, push, pull, merge, and branch your projects, and also to collaborate with other developers3.
* SAP Web IDE for SAP HANA: XS advanced provides a web-based integrated development environment (IDE) that allows you to create, edit, test, debug, and deploy your applications. SAP Web IDE for SAP HANA supports various development tools and features, such as editors, wizards, templates, code completion, syntax highlighting, code analysis, debugging, testing, and deployment3.
The other options are not correct because:
* SAP HANA repository: This is a component of XS classic model, not XS advanced. The SAP HANA repository stores the design-time artifacts of your applications, such as views, procedures, and scripts. In XS advanced, the design-time artifacts are stored in Git repositories instead4.
* SAP HANA studio: This is a desktop-based IDE that can be used to develop applications for XS classic model, but not for XS advanced. SAP HANA studio provides various

perspectives and tools to access and manage the SAP HANA database, such as the administration console, the catalog, the content, and the security4.
References: 2: SAP HANA Extended Application Services, Advanced Model 4: SAP HANA Extended Application Services 1: SAP HANA extended application services, advanced model 3: SAP HANA Developer Guide for SAP HANA XS Advanced Model (SAP Web IDE)

NEW QUESTION: 65
You need to access a remote database table object within your HDB module in a Multi-target application project.
In SAP HANA extended application services, advanced model (XS advanced) a user-provided service was created to access the external schema. The service is already declared in the Multi-target application project YAML file.
Which activities do you perform?
A. There are 3 correct answers to this question.
B. Create a view that refers to a synonym. n Create a synonym that refers to the view.
C. Create/update the .hdbgrants file providing authorizations for the remote schema.
D. Create a synonym for the remote table object.
E. Create a view that refers to the table.
Answer: (SHOW ANSWER)
To access a remote database table object within your HDB module in a Multi-target application project, you need to perform the following activities:
* Create/update the .hdbgrants file providing authorizations for the remote schema. This file defines the privileges that are granted to the HDI container technical users for accessing the remote schema. You need to specify the user-provided service name, the schema name, and the privileges, such as SELECT, INSERT, UPDATE, DELETE, EXECUTE, etc. You can also use the wildcard character (*) to grant privileges on all objects in the remote schema.
* Create a synonym for the remote table object. A synonym is a database object that provides an alternative name for another database object, such as a table, view, or procedure. You can use synonyms to simplify the access to remote objects by avoiding the need to specify the full qualified name. You can create a synonym in your HDB module by using the .hdbsynonym file, where you specify the target object name, the target object schema, and the user-provided service name.
* Create a view that refers to the table. A view is a database object that defines a logical table based on the result set of a SQL query. You can use views to access data from one or more tables, including remote tables, by applying filters, joins, aggregations, calculations, etc. You can create a view in your HDB module by using the .hdbview file, where you specify the SQL query that references the synonym of the remote table object.

The other options are incorrect because you do not need to create a view that refers to a synonym, nor create a synonym that refers to a view. These are redundant steps that do not add any value to the access of the remote table object. References:
* Accessing Remote Data - SAP Help Portal
* Granting Privileges on Remote Schemas - SAP Help Portal
* Creating Synonyms - SAP Help Portal
* Creating Views - SAP Help Portal

NEW QUESTION: 66
What are the key characteristics of the calculation view's Input Parameter? There are 3 correct answers to this question.
A. It is passed using a WHERE clause.
B. It can NOT be used to filter measure values.
C. It can be used in a conditional expression.
D. It is passed via reserved word PLACEHOLDER.
E. It can be used to pass values to table functions.
Answer: C,D,E (LEAVE A REPLY)
A calculation view is a view that combines data from multiple sources, such as tables, views, or functions, using graphical or SQLScript logic. A calculation view can define input parameters, which are variables that allow the user to influence the query execution with values that are entered at runtime. Input parameters can be used for various purposes, such as filtering, currency conversion, or dynamic calculations.
Some of the key characteristics of the calculation view's input parameter are:
* It can be used in a conditional expression. A conditional expression is an expression that evaluates to a value based on a condition. For example, an input parameter can be used to determine which column to use for aggregation, or which table function to call, based on the user's input. A conditional expression can be written using the CASE or IF syntax in SQLScript, or using the graphical expression editor in the calculation view.
* It is passed via reserved word PLACEHOLDER. The reserved word PLACEHOLDER is used to pass the input parameter value to the calculation view when calling it from SQL. The syntax is PLACEHOLDER." inputparametername
" => 'value'. For example, if the calculation view has an input parameter named CURRENCY, then the SQL statement to call the view with the value 'USD' would be SELECT * FROM "CALC_VIEW" (PLACEHOLDER." CURRENCY
" => 'USD').
* It can be used to pass values to table functions. A table function is a function that returns a table as its output. A table function can be used as a data source in a calculation view, and it can accept input parameters as arguments. For example, a table function can perform currency conversion based on the input parameter value, and return the converted data to the calculation view.

The other options are incorrect because they are not characteristics of the calculation view's input parameter. It is not passed using a WHERE clause, as the WHERE clause is used to filter data based on a condition, not to pass values to a view. It can be used to filter measure values, as measure values are numeric values that can be aggregated, such as sales amount or quantity. An input parameter can be used to filter measure values based on a range, a comparison, or a calculation. References:
* SAP HANA Platform 2.0 SPS06: SAP HANA SQL and System Views Reference, Section 2.1.5.3
* SAP HANA Platform 2.0 SPS06: SAP HANA SQLScript Reference, Section 2.1.2.1
* SAP HANA Platform 2.0 SPS06: SAP HANA SQLScript Reference, Section 2.1.2.2
* SAP HANA Platform 2.0 SPS06: SAP HANA SQLScript Reference, Section 2.1.2.3
* SAP HANA Platform 2.0 SPS06: SAP HANA SQLScript Reference, Section 2.1.2.4
* SAP HANA Platform 2.0 SPS06: SAP HANA SQLScript Reference, Section 2.1.2.5
* SAP HANA Platform 2.0 SPS06: SAP HANA SQLScript Reference, Section 2.1.2.6
* SAP HANA Platform 2.0 SPS06: SAP HANA SQLScript Reference, Section 2.1.2.7
* SAP HANA Platform 2.0 SPS06: SAP HANA SQLScript Reference, Section 2.1.2.8
* SAP HANA Platform 2.0 SPS06: SAP HANA SQLScript Reference, Section 2.1.2.9
* SAP HANA Platform 2.0 SPS06: SAP HANA SQLScript Reference, Section 2.1.2.10
* SAP HANA Platform 2.0 SPS06: SAP HANA SQLScript Reference, Section 2.1.2.11
* SAP HANA Platform 2.0 SPS06: SAP HANA SQLScript Reference, Section 2.1.2.12
* SAP HANA Platform 2.0 SPS06: SAP HANA SQLScript Reference, Section 2.1.2.13
* SAP HANA Platform 2.0 SPS06: SAP HANA SQLScript Reference, Section 2.1.2.14
* SAP HANA Platform 2.0 SPS06: SAP HANA SQLScript Reference, Section 2.1.2.15
* SAP HANA Platform 2.0 SPS06: SAP HANA SQLScript Reference, Section 2.1.2.16
* SAP HANA Platform 2.0 SPS06: SAP HANA SQLScript Reference, Section 2.1.2.17
* SAP HANA Platform 2.0 SPS06: SAP HANA SQLScript Reference, Section 2.1.2.18
* SAP HANA Platform 2.0 SPS06: SAP HANA SQLScript Reference, Section 2.1.2.19
* SAP HANA Platform 2.0 SPS06: SAP HANA SQLScript Reference, Section 2.1.2.20
* SAP HANA Platform 2.0 SPS06: SAP HANA SQLScript Reference, Section 2.1.2.21
* SAP HANA Platform 2.0 SPS06: SAP HANA SQLScript Reference, Section 2.1.2.22
* SAP HANA Platform 2.0 SPS06: SAP HANA SQLScript Reference, Section 2.1.2.23
* SAP HANA Platform 2.0 SPS06: SAP HANA SQLScript Reference, Section 2.1.2.24
* SAP HANA Platform 2.0 SPS06: SAP HANA SQLScript Reference, Section 2.1.2.25
* SAP HANA Platform 2.0 SPS06: SAP HANA SQLScript Reference, Section 2.1.2.26
* SAP HANA Platform 2.0 SPS06: SAP HANA SQLScript Reference, Section 2.1.2.27
* SAP HANA Platform 2.0 SPS06: SAP HANA SQLScript Reference, Section 2.1.2.28
* SAP HANA Platform 2.0 SPS06: SAP HANA SQLScript Reference, Section 2.1.2.29
* SAP HANA Platform 2.0 SPS06: SAP HANA SQLScript Reference, Section 2.1.2.30
* SAP HANA Platform 2.0 SPS06: SAP HANA SQLScript Reference, Section 2.1.2.31
* SAP HANA Platform 2.0 SPS06: SAP HANA SQLScript Reference, Section 2.1.2.32
* SAP HANA Platform 2.0 SPS06: SAP HANA SQLScript Reference, Section 2.1.