



**Free Questions for C\_HANADEV\_18 by dumpshq**

**Shared by Dale on 29-01-2024**

**For More Free Questions and Preparation Resources**

**Check the Links on Last Page**

# Question 1

---

**Question Type:** MultipleChoice

---

How do you specify the target currency for currency conversion in calculation views? There are 3 correct answers to this question.

## Options:

---

- A- In a column
- B- In a table function
- C- With an input parameter
- D- With a fixed value
- E- In a procedure

## Answer:

---

A, C, D

## Explanation:

---

To specify the target currency for currency conversion in calculation views, you can use one of the following methods:

In a column: You can associate the measure with an attribute column that contains the target currency code. For example, if you have a table that stores the sales amount and the currency code for each transaction, you can use the currency code column as the target currency for the sales amount measure.

With an input parameter: You can create an input parameter that allows the user to select or enter the target currency code at runtime. For example, if you want to display the sales amount in different currencies based on the user's preference, you can use an input parameter as the target currency for the sales amount measure.

With a fixed value: You can associate the measure with a fixed currency code that is available in the currency table. For example, if you want to display the sales amount in a single currency, such as USD, you can use a fixed value as the target currency for the sales amount measure. [Reference: Associate Measures with Currency, How to perform Currency Conversion in SAP HANA, Implementing Currency Conversion.](#)

## Question 2

---

**Question Type: MultipleChoice**

---

Which OData capacities are supported in SAP HANA extended application services, advanced model (XSA)? There are 3 correct answers to this question.

**Options:**

---

- A- Union
- B- Aggregation
- C- Join
- D- Projection
- E- Association

**Answer:**

---

B, C, E

**Explanation:**

---

OData is a protocol for exposing and consuming data over the web using RESTful APIs. OData supports various capacities, such as query options, operations, and annotations, to enable flexible and powerful data access and manipulation. SAP HANA extended application services, advanced model (XSA) is a framework for developing and deploying cloud-native applications on SAP HANA. XSA supports OData v4 as one of the service types that can be defined and exposed by the SAP Cloud Application Programming Model (CAP). XSA supports the following OData capacities in CAP:

**Aggregation:** This capacity allows you to perform aggregation functions, such as sum, count, min, max, and average, on the data returned by an OData service. You can use the \$apply query option to specify the aggregation expressions and groupings. XSA supports the OData Aggregation Extension for applying aggregations on entity sets and collections.

**Join:** This capacity allows you to perform join operations on the data returned by an OData service. You can use the \$expand query option to include related entities or properties in the response. XSA supports the OData Association and Navigation concepts for defining

and accessing the relationships between entities.

Association: This capacity allows you to define and use associations between entities in an OData service. Associations are semantic links that describe the cardinality and referential constraints of the relationships. You can use the OData Association and Referential Constraint annotations to specify the association details in the service definition or the metadata document.

The other options are not correct because:

A) Union: This capacity is not supported by OData or XSA. Union is a set operation that combines the results of two or more queries into a single result set. OData does not provide a query option or an extension for performing union operations on the data returned by an OData service.

D) Projection: This capacity is supported by OData, but not by XSA. Projection is a query option that allows you to select a subset of properties or entities from an OData service. You can use the \$select query option to specify the properties or entities you want to include in the response. However, XSA does not support the \$select query option in CAP, as it relies on the projection capabilities of the underlying data source, such as SAP HANA or PostgreSQL.

[SAP Cloud Application Programming Model, OData v4 Support](#)

[SAP Cloud Application Programming Model, Annotations](#)

[OData Version 4.0 Part 1: Protocol](#)

[OData Version 4.0 Part 2: URL Conventions](#)

[OData Aggregation Extension for Data Services Version 4.0](#)

## Question 3

---

**Question Type:** MultipleChoice

---

A Node.js module is executed for the first time in SAP HANA extended application services, advanced model (XS advanced). Which of the following activities are performed automatically? There are 2 correct answers to this question.

### Options:

---

- A- The source code is compiled to create an executable binary file.
- B- A new SAP HANA Deployment Infrastructure (HDI) container is created.
- C- The required modules are downloaded based on module dependencies.
- D- A new service is executed on the application server.

### Answer:

---

B, C

### Explanation:

---

When a Node.js module is executed for the first time in SAP HANA extended application services, advanced model (XS advanced), the following activities are performed automatically<sup>12</sup>:

A new SAP HANA Deployment Infrastructure (HDI) container is created: An HDI container is a logical database schema that contains the database objects and data that are required by the Node.js module. An HDI container is created based on the configuration and definition files of the Node.js module, such as the package.json, the mta.yaml, and the hdi-config.json. The HDI container is bound to the Node.js module as a service, and it can be accessed using the HDI client library or the SQL client library.

The required modules are downloaded based on module dependencies: The Node.js module may depend on other modules or libraries that provide additional functionality or services, such as express, hdb, or passport. These dependencies are specified in the package.json file of the Node.js module, and they are downloaded and installed automatically from the npm registry or the SAP npm registry when the Node.js module is executed for the first time.

The other options are not correct because they are not activities that are performed automatically when a Node.js module is executed for the first time in XS advanced. The source code is not compiled to create an executable binary file, but rather interpreted and executed by the Node.js runtime environment. A new service is not executed on the application server, but rather the existing Node.js module is executed as a service on the application server. Reference:

[SAP HANA Platform, Developing Applications with SAP HANA Cloud Platform, Developing Multi-Target Applications, Developing Node.js Modules](#)

[SAP HANA Platform, SAP HANA Extended Application Services, Advanced Model, Developing and Deploying Applications, Developing Node.js Applications](#)

## Question 4

---

### Question Type: MultipleChoice

---

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.

#### Options:

---

- A- Scalar variables
- B- Virtual tables
- C- Database tables
- D- Intermediate table variables

#### Answer:

---

C, D

#### Explanation:

---

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 effects<sup>1</sup>.

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 function<sup>2</sup>. Reference: [1:APPLY\\_FILTER | SAP Help Portal](#)<sup>2:Virtual Tables | SAP Help Portal</sup>

## Question 5

---

**Question Type:** MultipleChoice

---

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.

**Options:**

---

A- Security

B- Language

C- Default schema

D- Authorization

### Answer:

---

B, C

### Explanation:

---

You can specify the language and the default schema as metadata declarations in a SQL Script procedure when you use the SAP HANA Deployment Infrastructure (HDI). The language declaration defines the language mode of the procedure, which can be SQLSCRIPT or SQL. The default schema declaration defines the schema that is used to resolve unqualified object names in the procedure body. These declarations are optional and can be specified after the procedure name and before the parameter list. For example:

```
CREATE PROCEDURE MY_PROCEDURE LANGUAGE SQLSCRIPT DEFAULT SCHEMA MY_SCHEMA ( ... )
```

The other options are incorrect because security and authorization are not valid metadata declarations for SQL Script procedures. Security and authorization are handled by the HDI framework, which automatically grants the necessary privileges to the technical users and roles that access the HDI container objects. You do not need to specify any security or authorization clauses in the SQL Script procedure definition. Reference:

[Procedure Metadata- SAP Help Portal](#)

[CREATE PROCEDURE Statement \(Procedural\)- SAP Help Portal](#)

## Question 6

---

**Question Type:** MultipleChoice

---

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.

### Options:

---

- A- Push
- B- Rebase
- C- Pull
- D- Reset

### Answer:

---

C

### **Explanation:**

---

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.

## **Question 7**

---

**Question Type:** MultipleChoice

---

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.

### Options:

---

- A- A file with extension .hdbtable data
- B- A file with extension .hdbtable
- C- A file with extension .hdbdd
- D- A file with extension .hdbti

### Answer:

---

D

### Explanation:

---

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. Reference:

[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](#)

## Question 8

---

**Question Type: MultipleChoice**

---

Which functionality is provided by SAP WebIDE for SAP HANA? There are 2 correct answers to this question.

### Options:

---

- A- Syntax-aware editors for code
- B- User management
- C- Build and deploy applications
- D- Organization and space management

## **Answer:**

---

A, C

## **Explanation:**

---

SAP Web IDE for SAP HANA is a browser-based integrated development environment (IDE) that allows you to create and manage multi-target applications (MTAs) that use the SAP HANA Deployment Infrastructure (HDI). SAP Web IDE for SAP HANA provides a comprehensive suite of tools that enable the development of complex applications comprising web-based or mobile user interfaces (UIs), business logic, and extensive SAP HANA data models. SAP Web IDE for SAP HANA is tightly integrated with the SAP HANA runtime tools, the SAP HANA deployment infrastructure (HDI) for XS advanced, the tools used for application life-cycle management (ALM), and the XS advanced run-time platform. Some of the functionality that is provided by SAP Web IDE for SAP HANA are:

**Syntax-aware editors for code:** SAP Web IDE for SAP HANA provides syntax-aware editors for various code and SAP HANA artifacts, such as SQLScript, CDS, calculation views, HTML5, JavaScript, Java, and more. The syntax-aware editors offer features such as syntax highlighting, code completion, code formatting, code validation, code navigation, code templates, and code refactoring. The syntax-aware editors help you to write and edit code more efficiently and accurately.

**Build and deploy applications:** SAP Web IDE for SAP HANA provides build and deploy tools that allow you to compile, package, and deploy your MTA project to the XS advanced environment. The build and deploy tools use the `mta.yaml` file, which is the deployment descriptor file that specifies the metadata and dependencies for your MTA project. The build and deploy tools also use the HDI service, which is the service layer that enables you to deploy database artifacts to HDI containers. The build and deploy tools support various commands, such as `build`, `run`, `deploy`, `undeploy`, and `redploy`.

The following functionality is not provided by SAP Web IDE for SAP HANA, but by other tools or services:

User management: User management is the process of creating and managing users and their roles and privileges in the XS advanced environment. User management is not performed by SAP Web IDE for SAP HANA, but by the User Account and Authentication (UAA) service, which is a shared service that provides authorization and trust management for XS advanced applications. The UAA service uses the `xs-security.json` file, which is the application security descriptor file that specifies the security configuration for your application. The UAA service also uses the SAP HANA cockpit, which is a web-based administration tool that allows you to manage users and roles in the SAP HANA database.

Organization and space management: Organization and space management is the process of creating and managing the organizational units of the XS advanced environment, such as organizations and spaces. Organizations and spaces are used to isolate and group the resources and applications of different users or teams. Organization and space management is not performed by SAP Web IDE for SAP HANA, but by the Organization and Space Management tool, which is a web-based tool that allows you to create and manage organizations and spaces and their quotas and roles. The Organization and Space Management tool also uses the Cloud Foundry Command Line Interface (CF CLI), which is a command-line tool that allows you to perform various operations on the XS advanced environment and the applications and services.

[SAP HANA Deployment Infrastructure Reference], Chapter 5: HDI with XS Advanced, Section 5.1: Developing with the SAP Web IDE for SAP HANA, pp. 101-106.

[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, pp. 17-19.

## Question 9

---

**Question Type:** MultipleChoice

---



You create a multi-target application in SAP Web IDE for SAP HANA

### **Options:**

---

**A-** 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

### **Explanation:**

---

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. Reference: [Developing Multi-Target Applications, Git Integration, SAP HANA Deployment Infrastructure](#).

## Question 10

---

**Question Type:** MultipleChoice

---

What are the main SAP HANA engines responsible for executing calculation views? There are 3 correct answers to this question.

### Options:

---

- A- Java Engine
- B- XS Engine
- C- OLAP Engine
- D- Calculation Engine
- E- Join Engine

### Answer:

---

C, D, E

### Explanation:

---

Calculation views are a type of virtual data models that allow you to define complex calculations and aggregations on data from various sources, such as tables, views, or other calculation views. Calculation views are executed by different SAP HANA engines, depending on the type and complexity of the calculation view and the query that is submitted to it. The main SAP HANA engines responsible for executing calculation views are:

**OLAP Engine:** This engine is responsible for executing analytical queries on calculation views that are based on the star schema, which consists of a central fact table and multiple dimension tables. The OLAP engine supports multidimensional analysis, such as slicing, dicing, filtering, sorting, ranking, and drill-down operations. The OLAP engine also optimizes the query execution by applying techniques such as push-down, pruning, and caching.

**Calculation Engine:** This engine is responsible for executing complex calculations and transformations on calculation views that are based on the graph model, which consists of nodes and edges that represent data sources and operations. The calculation engine supports advanced features, such as hierarchies, unions, projections, aggregations, joins, and scripting. The calculation engine also leverages the parallel processing capabilities of SAP HANA to optimize the performance of data-intensive operations.

**Join Engine:** This engine is responsible for executing join operations on calculation views that involve multiple data sources. The join engine supports different types of joins, such as inner join, left outer join, right outer join, full outer join, referential join, and text join. The join engine also applies various join algorithms, such as hash join, nested loop join, and merge join, depending on the characteristics of the data sources and the join conditions.

The other options are not correct because:

A) **Java Engine:** This engine is not responsible for executing calculation views, but for executing Java applications that run on the SAP HANA XS Advanced Model. The Java engine provides the core services and libraries for building and running Java applications, such as data access, authentication, authorization, localization, logging, and testing.

B) XS Engine: This engine is not responsible for executing calculation views, but for executing XSJS and XSODATA applications that run on the SAP HANA XS Classic Model. The XS engine provides the runtime environment and the web server for executing XSJS and XSODATA applications, which can access SAP HANA database features using JavaScript syntax.

[SAP HANA Developer Guide for SAP HANA Web IDE, Chapter 6, Section 6.1](#)

[SAP HANA Developer Guide for SAP HANA Web IDE, Chapter 6, Section 6.2](#)

[SAP HANA Developer Guide for SAP HANA Web IDE, Chapter 6, Section 6.3](#)

[SAP HANA Developer Guide for SAP HANA Web IDE, Chapter 6, Section 6.4](#)

**To Get Premium Files for C\_HANADEV\_18 Visit**

**[https://www.p2pexams.com/products/c\\_hanadev\\_18](https://www.p2pexams.com/products/c_hanadev_18)**

**For More Free Questions Visit**

**<https://www.p2pexams.com/sap/pdf/c-hanadev-18>**

