Free Questions for C_ABAPD_2309 by actualtestdumps

Shared by Sears on 24-05-2024

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

Which of the following integration frameworks have been released for ABAP cloud development? Note: There are 3 correct answers to this question.

Options:

A- SOAP consumption

B- CDS Views

C- Business Add-ins (BAdls)

D- Business Events

E- OData services

Answer:

A, D, E

Explanation:

The following are the integration frameworks that have been released for ABAP cloud development:

SOAP consumption: This framework allows you to consume SOAP web services from ABAP cloud applications. You can use the ABAP Development Tools in Eclipse to create a service consumption model based on a WSDL file or URL. The service consumption model generates the required ABAP artifacts, such as proxy classes, data types, and constants, to access the web service.You can then use the proxy classes to call the web service operations from your ABAP code1

Business Events: This framework allows you to publish and subscribe to business events from ABAP cloud applications. Business events are messages that represent a change in the state of a business object or process. You can use the ABAP Development Tools in Eclipse to create a business event definition based on a CDS view entity or a projection view. The business event definition specifies the event key, the event payload, and the event metadata.You can then use the ABAP Messaging Channel (AMC) framework to publish and subscribe to business events using the AMC API2

OData services: This framework allows you to expose and consume OData services from ABAP cloud applications. OData is a standardized protocol for creating and consuming RESTful APIs. You can use the ABAP RESTful Application Programming Model (RAP) to create OData services based on CDS view entities or projection views. The RAP framework generates the required OData metadata and runtime artifacts, such as service definitions, service bindings, and service implementations. You can then use the SAP Gateway framework to register and activate your OData services.You can also use the ABAP Development Tools in Eclipse to consume OData services from other sources using the service consumption model3

The other integration frameworks are not released for ABAP cloud development, as they are either not supported or not recommended for cloud scenarios. These frameworks are:

CDS Views: CDS views are not an integration framework, but a data modeling framework. CDS views are used to define data models based on database tables or other CDS view entities. CDS views can have associations, aggregations, filters, parameters, and annotations.CDS views can also be used as the basis for other integration frameworks, such as OData services or business events4

Business Add-ins (BAdls): BAdls are not supported for ABAP cloud development, as they are part of the classic ABAP enhancement framework. BAdls are used to implement custom logic in predefined enhancement spots in the standard SAP code. BAdls are not compatible with the cloud strategy and the clean core paradigm, as they modify the SAP code and can cause upgrade and maintenance issues. For ABAP cloud development, SAP recommends using the key user extensibility tools or the side-by-side extensibility approach instead of BAdls.

Question 2

Question Type: MultipleChoice

In class ZCL_CLASS_A, you use the statement DATA var TYPE ***

What may stand in place of ***? Note: There are 2 correct answers to this question.

Options:

A- The name of a type defined privately in class ZCL_CLASS_A

B- The name of a data element from the ABAP Dictionary

C- The name of a type defined privately in another class

D- The name of a domain from the ABAP Dictionary

Answer:

B, D

Explanation:

In class ZCL_CLASS_A, you use the statement DATA var TYPE *** to declare a data object named var with a data type specified by ***.The data type can be any of the following1:

A predefined ABAP type, such as i, f, c, string, xstring, and so on.

A data element from the ABAP Dictionary, such as matnr, carrid, bukrs, and so on.A data element defines the semantic and technical attributes of a data field, such as the domain, the length, the data type, the description, and the value range2.

A domain from the ABAP Dictionary, such as matnr_d, carrid_d, bukrs_d, and so on.A domain defines the technical attributes of a data field, such as the data type, the length, the output length, the number of decimal places, and the value range3.

A type defined globally in a class, an interface, or a type pool, such as zcl_class_b=>type_a, zif_interface_c=>type_b, ztype_pool_d=>type_c, and so on.A global type is a type that is defined in a global repository object and can be used in any program or class4.

A type defined locally in the current class, such as type_a, type_b, type_c, and so on.A local type is a type that is defined in the declaration part of a class and can only be used within the class5.

Therefore, the possible values for *** are B. the name of a data element from the ABAP Dictionary and D. the name of a domain from the ABAP Dictionary. The other options are not valid because:

A) The name of a type defined privately in class ZCL_CLASS_A is a local type and cannot be used with the DATA statement.A local type can only be used with the TYPES statement5.

C) The name of a type defined privately in another class is a private type and cannot be accessed from outside the class. A private type can only be used within the class that defines it.

Question 3

Question Type: MultipleChoice

Which ABAP SQL clause allows the use of inline declarations?

Options:

A- FROM

B- INTO CORRESPONDING FIELDS OF

C- INTO

D- FIELDS

Answer:

C

Explanation:

The ABAP SQL clause that allows the use of inline declarations is the INTO clause. The INTO clause is used to specify the target variable or field symbol where the result of the SQL query is stored. The INTO clause can use inline declarations to declare the target variable or field symbol at the same position where it is used, without using a separate DATA or FIELD-SYMBOLS statement.The inline declaration is performed using the DATA or @DATA operators in the declaration expression12. For example:

The following code snippet uses the INTO clause with an inline declaration to declare a local variable itab and store the result of the SELECT query into it:

SELECT * FROM scarr INTO TABLE @DATA (itab).

The following code snippet uses the INTO clause with an inline declaration to declare a field symbol <fs> and store the result of the SELECT query into it:

SELECT SINGLE * FROM scarr INTO @<fs>.

You cannot do any of the following:

FROM: The FROM clause is used to specify the data source of the SQL query, such as a table, a view, or a join expression.The FROM clause does not allow the use of inline declarations12.

INTO CORRESPONDING FIELDS OF: The INTO CORRESPONDING FIELDS OF clause is used to specify the target structure or table where the result of the SQL query is stored. The INTO CORRESPONDING FIELDS OF clause does not allow the use of inline declarations.The target structure or table must be declared beforehand using a DATA or FIELD-SYMBOLS statement12.

FIELDS: The FIELDS clause is used to specify the columns or expressions that are selected from the data source of the SQL query. The FIELDS clause does not allow the use of inline declarations.The FIELDS clause must be followed by an INTO clause that specifies the target variable or field symbol where the result is stored12.

Question 4

Question Type: MultipleChoice

Why would you use Access Controls with CDS Views? Note: There are 2 correct answers to this question.

Options:

A- Only the data corresponding to the user's authorization is transferred from the database to the application layer.

B- The system field sy-subrc is set, giving you the result of the authorization check

C- You do not have to remember to implement AUTHORITY CHECK statements.

D- All of the data from the data sources is loaded into your application automatically and filtered there according to the user's authorization.

Answer:

A, C

Explanation:

You would use Access Controls with CDS Views for the following reasons:

A) Only the data corresponding to the user's authorization is transferred from the database to the application layer. This is true because Access Controls allow you to define CDS roles that specify the authorization conditions for accessing a CDS view. The CDS roles are evaluated for every user at runtime and the system automatically adds the restrictions to the selection conditions of the CDS view. This ensures that only the data that the user is authorized to see is read from the database and transferred to the application layer.This improves the security and the performance of the data access1.

C) You do not have to remember to implement AUTHORITY CHECK statements. This is true because Access Controls provide a declarative and centralized way of defining the authorization logic for a CDS view. You do not have to write any procedural code or use the AUTHORITY CHECK statement to check the user's authorization for each data source or field.The system handles the authorization check automatically and transparently for you2.

The following reasons are not valid for using Access Controls with CDS Views:

B) The system field sy-subrc is set, giving you the result of the authorization check. This is false because the system field sy-subrc is not used by Access Controls. The sy-subrc field is used by the AUTHORITY CHECK statement to indicate the result of the authorization check, but Access Controls do not use this statement.Instead, Access Controls use CDS roles to filter the data according to the user's authorization2.

D) All of the data from the data sources is loaded into your application automatically and filtered there according to the user's authorization. This is false because Access Controls do not load all the data from the data sources into the application layer. Access Controls filter the data at the database layer, where the data resides, and only transfer the data that the user is authorized to see to the application layer.This reduces the data transfer and the memory consumption of the application layer1.

Question 5

Question Type: MultipleChoice

Which type of legacy code does SAP recommend you eliminate when you review modifications as part of an SAP S/4HANA system conversion? Note: There are 2 correct answers to this question.

Options:

A- Code that supports a critical business process

B- Code that now is identical to a standard SAP object

C- Code that has less than 10% usage according to usage statistics

D- Code that can be redesigned as a key user extension

Answer:

B, D

Explanation:

SAP recommends that you eliminate the following types of legacy code when you review modifications as part of an SAP S/4HANA system conversion:

Code that now is identical to a standard SAP object. This type of code is redundant and unnecessary, as it does not provide any additional functionality or customization. It can also cause conflicts or errors during the system conversion, as the standard SAP object may have changed or been replaced in SAP S/4HANA. Therefore, you should delete this type of code and use the standard SAP object instead.

Code that can be redesigned as a key user extension. This type of code is usually related to UI or business logic adaptations that can be achieved using the in-app tools provided by SAP S/4HANA. By redesigning this type of code as a key user extension, you can simplify and standardize your code base, reduce maintenance efforts, and avoid compatibility issues during the system conversion. Therefore, you should migrate this type of code to the key user extensibility framework and delete the original code.

The other types of legacy code are not recommended to be eliminated, as they may still be relevant or necessary for your business processes. However, you should still review and adjust them according to the SAP S/4HANA simplification items and best practices.

These types of code are:

Code that supports a critical business process. This type of code is essential for your business operations and cannot be easily replaced or removed. However, you should check if this type of code is compatible with SAP S/4HANA, and if not, you should adapt it accordingly. You should also consider if this type of code can be optimized or enhanced using the new features and capabilities of SAP S/4HANA.

Code that has less than 10% usage according to usage statistics. This type of code is rarely used and may not be worth maintaining or converting. However, you should not delete this type of code without verifying its relevance and impact on your business processes. You should also consider if this type of code can be replaced or consolidated with other code that has higher usage or better performance.

Question 6

Question Type: MultipleChoice

What are some properties of database tables? Note: There are 2 correct answers to this question.

Options:

A- They store information in two dimensions.

B- They may have key fields.

C- They can have any number of key fields.

D- They can have relationships to other tables.

Answer:

A, D

Explanation:

Database tables are data structures that store information in two dimensions, using rows and columns. Each row represents a record or an entity, and each column represents an attribute or a field. Database tables may have key fields, which are columns that uniquely identify each row or a subset of rows. Key fields can be used to enforce data integrity, perform efficient searches, and establish relationships to other tables. Database tables can have relationships to other tables, which are associations or links between the key fields of two or more tables.Relationships can be used to model the logical connections between different entities, join data from multiple tables, and enforce referential integrity12.

Question 7

Question Type: MultipleChoice

Which of the following are parts of answers to this question.

Options:

A- Partitioning attributes

B- Extension

C- Field list

D- Semantic table attributes

Answer:

B, C

Explanation:

A CDS view is a data definition that defines a data structure and a data selection from one or more data sources. A CDS view consists of several parts, but two of them are:

Extension: An extension is an optional clause that allows a CDS view to extend another CDS view by adding new elements, annotations, or associations. The extension clause has the syntax EXTEND VIEW view_name WITH view_name.The first view_name is the name of the CDS view that is being extended, and the second view_name is the name of the CDS view that is doing the extension1.

Field list: A field list is a mandatory clause that specifies the elements of the CDS view. The field list has the syntax SELECT FROM data_source { element_list }. The data_source is the name of the data source that the CDS view selects data from, and the element_list is a comma-separated list of elements that the CDS view exposes.The elements can be fields of the data source, expressions, associations, or annotations2.

The following example shows a CDS view that extends another CDS view and defines a field list:

```
@AbapCatalog.sqlViewName: 'ZCDS_EXT' define view Z_CDS_Extension extend view Z_CDS_Base with Z_CDS_Extension as select
from ztable { // field list key ztable.id as ID, ztable.name as Name, ztable.age as Age, // extension @Semantics.currencyCode: true
ztable.currency as Currency }
```

The other options are not parts of a CDS view, but rather related concepts:

Partitioning attributes: Partitioning attributes are attributes that are used to partition a table into smaller subsets of data. Partitioning attributes are defined in the ABAP Dictionary for transparent tables and can improve the performance and scalability of data access.Partitioning attributes are not part of the CDS view definition, but rather the underlying table definition3.

Semantic table attributes: Semantic table attributes are attributes that provide additional information about the meaning and usage of a table. Semantic table attributes are defined in the ABAP Dictionary for transparent tables and can be used to enhance the data modeling and consumption of the table.Semantic table attributes are not part of the CDS view definition, but rather the underlying table definition4.