

**Free Questions for C\_ABAPD\_2309**

**Shared by Cole on 04-10-2024**

**For More Free Questions and Preparation Resources**

**Check the Links on Last Page**

# Question 1

---

**Question Type:** MultipleChoice

---

Exhibit:

```
target_itab = VALUE # ( FOR row IN source_itab(  
  field1 = row-field1  
  field2 = row-field2  
  fieldn = row-fieldn )  
);
```

Which of the following statements are correct? Note: There are 2 correct answers to this question.

**Options:**

---

**A-** FOR defines a loop that runs over the content of source\_itab

- B-** source\_itab is only visible within the loop.
- C-** row is a predefined name and cannot be chosen arbitrarily.
- D-** row is only visible within the loop.

**Answer:**

---

A, D

**Explanation:**

---

The code snippet in the image is an example of using the FOR statement to create an internal table with a constructor expression. The FOR statement introduces an iteration expression that runs over the content of source\_itab and assigns each row to the variable row. The variable row is then used to populate the fields of target\_itab<sup>12</sup>. Some of the correct statements about the code snippet are:

FOR defines a loop that runs over the content of source\_itab: This is true. The FOR statement iterates over the rows of source\_itab and assigns each row to the variable row. The iteration expression can also specify a range or a condition for the loop<sup>12</sup>.

row is only visible within the loop: This is true. The variable row is a local variable that is only visible within the scope of the iteration expression. It cannot be accessed outside the loop<sup>12</sup>.

You cannot do any of the following:

source\_itab is only visible within the loop: This is false. The variable source\_itab is not a local variable that is defined by the FOR statement. It is an existing internal table that is used as the data source for the iteration expression. It can be accessed outside the loop<sup>12</sup>.

row is a predefined name and cannot be chosen arbitrarily: This is false. The variable row is not a predefined name that is reserved by the FOR statement. It is a user-defined name that can be chosen arbitrarily. However, it must not conflict with any existing names in the program12.

## Question 2

---

**Question Type:** MultipleChoice

---

What are some of the reasons that Core Data Services are preferable to the classical approach to data modeling? Note: There are 2 correct answers to this question.

### Options:

---

- A- They implement code pushdown.
- B- They avoid data transfer completely.
- C- They transfer computational results to the application server.
- D- They compute results on the application server.

### Answer:

---

A, C

### **Explanation:**

---

Core Data Services (CDS) are preferable to the classical approach to data modeling for several reasons, but two of them are:

They implement code pushdown. Code pushdown is the principle of moving data-intensive logic from the application server to the database server, where the data resides. This reduces the data transfer between the application server and the database server, which improves the performance and scalability of the application. CDS enable code pushdown by allowing the definition of semantic data models and business logic in the database layer, using SQL and SQL-based expressions<sup>1</sup>.

They transfer computational results to the application server. CDS allow the application server to access the data and the logic defined in the database layer by using Open SQL statements. Open SQL is a standardized and simplified subset of SQL that can be used across different database platforms. Open SQL statements are translated into native SQL statements by the ABAP runtime environment and executed on the database server. The results of the computation are then transferred to the application server, where they can be further processed or displayed<sup>2</sup>.

## **Question 3**

---

**Question Type:** MultipleChoice

---

In an Access Control Object, which clauses are used? Note: There are 3 correct answers to this question.

## Options:

---

- A- Where (to specify the access conditions)
- B- Crant (to identify the data source)
- C- Return code (to assign the return code of the authority check)
- D- Define role (to specify the role name)
- E- Revoke (to remove access to the data source)

## Answer:

---

A, D, E

## Explanation:

---

An Access Control Object (ACO) is a CDS annotation that defines the access control rules for a CDS view entity. An ACO consists of one or more clauses that specify the role name, the data source, the access conditions, and the return code of the authority check<sup>12</sup>. Some of the clauses that are used in an ACO are:

Where (to specify the access conditions): This clause is used to define the logical expression that determines whether a user has access to the data source or not. The expression can use the fields of the data source, the parameters of the CDS view entity, or the predefined variables \$user and \$session. The expression can also use the functions check\_authorization and check\_role to perform additional authority checks<sup>12</sup>.

Define role (to specify the role name): This clause is used to assign a name to the role that is defined by the ACO. The role name must be unique within the namespace of the CDS view entity and must not contain any special characters. The role name can be used to reference the ACO in other annotations, such as @AccessControl.authorizationCheck or @AccessControl.grant12.

Revoke (to remove access to the data source): This clause is used to explicitly deny access to the data source for a user who meets the conditions of the where clause. The revoke clause overrides any grant clause that might grant access to the same user. The revoke clause can be used to implement the principle of least privilege or to enforce data segregation12.

You cannot do any of the following:

Grant (to identify the data source): This is not a valid clause in an ACO. The grant clause is a separate annotation that is used to grant access to a CDS view entity or a data source for a user who has a specific role. The grant clause can reference an ACO by its role name to apply the access conditions defined by the ACO12.

Return code (to assign the return code of the authority check): This is not a valid clause in an ACO. The return code of the authority check is a predefined variable that is set by the system after performing the access control check. The return code can be used in the where clause of the ACO to specify different access conditions based on the outcome of the check12.

## Question 4

---

**Question Type:** MultipleChoice

---

Class super has subclass sub. Which rules are valid for the sub constructor? Note: There are 2 correct answers to this question.

### Options:

---

- A- The method signature can be changed.
- B- Import parameters can only be evaluated after calling the constructor of super.
- C- The constructor of super must be called before using any components of your own instance.
- D- Events of your own instance cannot be raised before the registration of a handler in super.

### Answer:

---

A, C

### Explanation:

---

The sub constructor is the instance constructor of the subclass sub that inherits from the superclass super. The sub constructor has some rules that it must follow when it is defined and implemented<sup>12</sup>. Some of the valid rules are:

The method signature can be changed: This is true. The sub constructor can have a different method signature than the super constructor, which means that it can have different input parameters, output parameters, or exceptions. However, the sub constructor must still call the super constructor with appropriate actual parameters that match its interface<sup>12</sup>.

The constructor of super must be called before using any components of your own instance: This is true. The sub constructor must ensure that the super constructor is called explicitly using `super->constructor` before accessing any instance components of its own class, such as attributes or methods. This is because the super constructor initializes the inherited components of the subclass and sets



the self-reference me-> to the current instance12.

You cannot do any of the following:

Import parameters can only be evaluated after calling the constructor of super: This is false. The sub constructor can evaluate its own import parameters before calling the constructor of super, as long as it does not access any instance components of its own class. For example, the sub constructor can use its import parameters to calculate some values or check some conditions that are needed for calling the super constructor12.

Events of your own instance cannot be raised before the registration of a handler in super: This is false. The sub constructor can raise events of its own instance before calling the constructor of super, as long as it does not access any instance components of its own class. For example, the sub constructor can raise an event to notify the consumers of the subclass about some status or error that occurred during the initialization of the subclass12.

## Question 5

---

**Question Type:** MultipleChoice

---

When does SAP recommend to use a sorted or a hashed table respectively? Note: There are 2 correct answers to this question.

**Options:**

---

- A-** A hashed table, when you read a single record and specify the complete key.
- B-** A hashed table, when you read a subset in a loop and specify a part of the key from the left without gaps.
- C-** A sorted table, when you read a subset in a loop and specify a part of the key from the left ^ without gaps.
- D-** A sorted table, when you read a single record and specify non key fields.

**Answer:**

---

A, B

## Question 6

---

**Question Type:** MultipleChoice

---

What is the purpose of a foreign key relationship between two tables in the ABAP Dictionary?

**Options:**

---

- A-** To document the relationship between the two tables
- B-** To ensure the integrity of data in the corresponding database tables

**C-** To create a corresponding foreign key relationship in the database

**Answer:**

---

B

**Explanation:**

---

The purpose of a foreign key relationship between two tables in the ABAP Dictionary is to ensure the integrity of data in the corresponding database tables. A foreign key relationship defines a logical link between a foreign key table and a check table, where the foreign key fields of the former are assigned to the primary key fields of the latter. This means that the values entered in the foreign key fields must exist in the check table, otherwise the system will reject the entry. This way, the foreign key relationship prevents the insertion of invalid or inconsistent data in the database tables.

A foreign key relationship also serves to document the relationship between the two tables in the ABAP Dictionary, but this is not its primary purpose. A foreign key relationship does not necessarily create a corresponding foreign key relationship in the database, as this depends on the database system and the settings of the ABAP Dictionary. Some database systems do not support foreign keys at all, while others require additional steps to activate them. Therefore, the foreign key relationship in the ABAP Dictionary is mainly a logical concept that is enforced by the ABAP runtime environment.

[https://help.sap.com/doc/saphelp\\_snc70/7.0/en-US/cf/21ea77446011d189700000e8322d00/content.htm](https://help.sap.com/doc/saphelp_snc70/7.0/en-US/cf/21ea77446011d189700000e8322d00/content.htm)

## Question 7

---

**Question Type: MultipleChoice**

---

Which of the following are features of Core Data Services? Note: There are 3 correct answers to this question.

**Options:**

---

- A- Inheritance
- B- Associations
- C- Annotations
- D- Delegation
- E- Structured Query Language (SQL)

**Answer:**

---

B, C, E

**Explanation:**

---

Core Data Services (CDS) is a framework for defining and consuming semantically rich data models in SAP HANA. CDS supports various features that enhance the capabilities of SQL and enable developers to create data models that are optimized for performance, readability, and extensibility<sup>12</sup>. Some of the features of CDS are:

Associations: Associations are a way of defining relationships between CDS entities, such as tables or views. Associations enable navigation and path expressions in CDS queries, which allow accessing data from related entities without explicit joins. Associations also support cardinality, referential constraints, and cascading options<sup>34</sup>.

Annotations: Annotations are a way of adding metadata to CDS entities or their elements, such as fields or parameters. Annotations provide additional information or instructions for the CDS compiler, the database, or the consumers of the CDS views. Annotations can be used for various purposes, such as defining access control, UI rendering, OData exposure, or search capabilities<sup>5</sup>.

Structured Query Language (SQL): SQL is the standard language for querying and manipulating data in relational databases. CDS is based on SQL and extends it with additional features and syntax. CDS supports SQL features such as joins, aggregations, filters, expressions, functions, and subqueries. CDS also supports SQL Script, which is a scripting language for stored procedures and functions in SAP HANA .

You cannot do any of the following:

Inheritance: Inheritance is not a feature of CDS. Inheritance is a concept in object-oriented programming that allows a class to inherit the properties and methods of another class. CDS does not support object-oriented programming or classes.

Delegation: Delegation is not a feature of CDS. Delegation is a concept in object-oriented programming that allows an object to delegate some of its responsibilities to another object. CDS does not support object-oriented programming or objects.

## Question 8

---

**Question Type:** MultipleChoice

---

The class `zcl_demo_class` is in a software component with the language version set to "Standard ABAP". The function module "ZF11" is in a software component with the language version set to "ABAP Cloud". Both the class and function module are customer created. Regarding line #6, which of the following is a valid statement?

### Options:

---

- A- 'ZF1' can be called whether it has been released or not for cloud development.
- B- 'ZF1' can be called via a wrapper that itself has been released for cloud development.
- C- 'ZF1' can be called via a wrapper that itself has not been released for cloud development.
- D- 'ZF1' must be released for cloud development to be called.

### Answer:

---

B

### Explanation:

---

The function module ZF1 is in a software component with the language version set to "ABAP Cloud". This means that it follows the ABAP Cloud Development Model, which requires the usage of public SAP APIs and extension points to access SAP functionality and data. These APIs and extension points are released by SAP and documented in the SAP API Business Hub1. Customer-created function modules are not part of the public SAP APIs and are not released for cloud development. Therefore, calling a function module directly from a class with the language version set to "Standard ABAP" is not allowed and will result in a syntax error. However, there is a

possible way to call a function module indirectly from a class with the language version set to "Standard ABAP":

Create a wrapper class or interface for the function module and release it for cloud development. A wrapper is a class or interface that encapsulates the function module and exposes its functionality through public methods or attributes. The wrapper must be created in a software component with the language version set to "ABAP Cloud" and must be marked as released for cloud development using the annotation `@EndUserText.label`. The wrapper can then be called from a class with the language version set to "Standard ABAP" using the public methods or attributes.

For example, the following code snippet shows how to create a wrapper class for the function module ZF1 and call it from the class `zcl_demo_class`:

```
@EndUserText.label: 'Wrapper for ZF1' CLASS zcl_wrapper_zf1 DEFINITION PUBLIC FINAL CREATE PUBLIC. PUBLIC SECTION.  
CLASS-METHODS: call_zf1 IMPORTING iv_a TYPE i iv_b TYPE i EXPORTING ev_result TYPE i. ENDCLASS.
```

```
CLASS zcl_wrapper_zf1 IMPLEMENTATION. METHOD call_zf1. CALL FUNCTION 'ZF1' EXPORTING a = iv_a b = iv_b IMPORTING  
result = ev_result. ENDMETHOD. ENDCLASS.
```

```
CLASS zcl_demo_class DEFINITION. METHODS: m1. ENDCLASS.
```

```
CLASS zcl_demo_class IMPLEMENTATION. METHOD m1. DATA(lv_result) = zcl_wrapper_zf1=>call_zf1( iv_a = 2 iv_b = 3 ). WRITE: /  
lv_result. ENDMETHOD. ENDCLASS.
```

The output of this code is:

## Question 9

---

**Question Type:** MultipleChoice

---

Which of the following are parts of the definition of a new database table?Note: There are 2 correct answers to this question.

### Options:

---

- A- Partitioning attribues
- B- Extension
- C- Semantic table attributes
- D- Field list

### Answer:

---

C, D

## Question 10

---

**Question Type:** MultipleChoice

---



Which patterns raise an exception? Note: There are 3 correct answers to this question.

### Options:

---

- A-** DATA: gv\_target TYPE p DECIMALS 2. CONSTANTS: go intl TYPE i VALUE 3. gv\_target -U EXACT (2 gcojntl).
- B-** DATA: gv\_target TYPE string. CONSTANTS: gco\_string TYPE LENGTH 16 VALUE 0123456789ABCDEF\*. gv\_target = EXACT # gco\_string+5 (5) ).
- C-** DATA: gv\_target TYPE c LENGTH 5. V CONSTANTS: ECO string TYPE string VALUE 0123456789ABCDEF'. gv\_target - EXACT (gco\_string + 5 (6) ).
- D-** DATA: Ev target TYPE p DECIMALS 3. CONSTANTS: gcojntl TYPE i VALUE 2. Ev\_target -U EXACT #2 / gcojntl ).
- E-** DATA: gv\_target TYPE d. s/ CONSTANTS: gco\_date TYPE d VALUE '20331233\*. gv\_target EXACT ( geo\_date).

### Answer:

---

A, C, E

### Explanation:

---

The patterns that raise an exception are those that use the constructor operator EXACT to perform a lossless assignment or calculation, but the result cannot be converted to the target data type without data loss. The following are the explanations for each pattern:

A: This pattern raises the exception `CX_SY_CONVERSION_LOST` because the result of the calculation  $2 * 3$  is 6, which cannot be assigned to a packed number with two decimal places without losing the integer part. The operator `-U` is used to perform a lossless calculation with the calculation type `decfloat34`.

B: This pattern does not raise an exception because the result of the substring expression `gco_string+5(5)` is '6789A', which can be assigned to a string without data loss. The operator `EXACT #` is used to perform a lossless assignment with the data type of the argument.

C: This pattern raises the exception `CX_SY_CONVERSION_LOST` because the result of the substring expression `gco_string+5(6)` is '6789AB', which cannot be assigned to a character field with length 5 without losing the last character. The operator `EXACT` is used to perform a lossless assignment with the data type of the target field.

D: This pattern does not raise an exception because the result of the calculation  $2 / 2$  is 1, which can be assigned to a packed number with three decimal places without data loss. The operator `-U` is used to perform a lossless calculation with the calculation type `decfloat34`.

E: This pattern raises the exception `CX_SY_CONVERSION_ERROR` because the constant `gco_date` contains an invalid value '20331233' for a date data type, which cannot be converted to a valid date. The operator `EXACT` is used to perform a lossless assignment with the data type of the target field.

**To Get Premium Files for C\_ABAPD\_2309 Visit**

**[https://www.p2pexams.com/products/c\\_abapd\\_2309](https://www.p2pexams.com/products/c_abapd_2309)**

**For More Free Questions Visit**

**<https://www.p2pexams.com/sap/pdf/c-abapd-2309>**

