Number: C_ABAPD_2309
Passing Score: 800
Time Limit: 120
File Version: 3.0

Exam Code: C_ABAPD_2309

Exam Name: SAP Certified Associate - Back-End Developer - ABAP Cloud

Exam A

QUESTION 1
When does SAP recommend to use a sorted or a hashed table respectively? Note: There are 2 correct answers to this question.

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.

Correct Answer: A, B
Section:

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

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.

Correct Answer: A, C
Section:
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 implemented12. 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 interface12.
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 3
In an Access Control Object, which clauses are used? Note: There are 3 correct answers to this question.

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)

Correct Answer: A, D, E
Section:
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 check12. 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 checks12.
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
You want to provide a short description of the data definition for developers that will be attached to the database view
Which of the following annotations would do this if you inserted it on line #27

A. @UI headerinto description label
B. @UI.badge.title.label
C. @EndUserText.quickInfo
D. @EndUserText label

Correct Answer: D
Section:
Explanation:
The annotation that can be used to provide a short description of the data definition for developers that will be attached to the database view is the @EndUserText.label annotation. This annotation is used to specify a text label for the data definition that can be displayed in the development tools or in the documentation.The annotation can be inserted on line #27 in the code snippet provided in the question12. For example:
The following code snippet uses the @EndUserText.label annotation to provide a short description of the data definition for the CDS view ZCDS_VIEW:
@AbapCatalog.sqlViewName: 'ZCDS_VIEW' @AbapCatalog.compiler.compareFilter: true @AbapCatalog.preserveKey: true @AccessControl.authorizationCheck: #CHECK @EndUserText.label: 'CDS view for flight data' 'short description for developers define view ZCDS_VIEW as select from sflight { key carrid, key connid, key fldate, seatsmax, seatsocc }
You cannot do any of the following:
@UI.headerInfo.description.label: This annotation is used to specify a text label for the description field of the header information of a UI element.This annotation is not relevant for the data definition of a database view12.
@UI.badge.title.label: This annotation is used to specify a text label for the title field of a badge UI element.This annotation is not relevant for the data definition of a database view12.
@EndUserText.quickInfo: This annotation is used to specify a quick information text for the data definition that can be displayed as a tooltip in the development tools or in the documentation.This annotation is not the same as a short description or a label for the data definition12.

QUESTION 5
Which statement can you use to change the contents of a row of data in an internal table?

A. Append table
B. Modify table
C. Insert table
D. Update table

Correct Answer: B

Section:
Explanation:
The statement that can be used to change the contents of a row of data in an internal table is MODIFY table. The MODIFY table statement can be used to change the contents of one or more rows of an internal table, either by specifying the table index, the table key, or a condition. The MODIFY table statement can also be used to change the contents of a database table, by specifying the table name and a work area or an internal table. The MODIFY table statement can use the TRANSPORTING addition to specify which fields should be changed, and the WHERE addition to specify which rows should be changed.
The other statements are not suitable for changing the contents of a row of data in an internal table, as they have different purposes and effects. These statements are:
APPEND table: This statement can be used to add a new row of data to the end of an internal table, either by specifying a work area or an inline declaration. The APPEND table statement does not change the existing rows of the internal table, but only increases the number of rows by one.
INSERT table: This statement can be used to insert a new row of data into an internal table, either by specifying the table index, the table key, or a sorted position. The INSERT table statement does not change the existing rows of the internal table, but only shifts them to make room for the new row. The INSERT table statement can also be used to insert a new row of data into a database table, by specifying the table name and a work area or an inline declaration.
UPDATE table: This statement can be used to update the contents of a database table, by specifying the table name and a work area or an internal table. The UPDATE table statement can use the SET addition to specify which fields should be updated, and the WHERE addition to specify which rows should be updated. The UPDATE table statement does not affect the internal table, but only the corresponding database table.

QUESTION 6
As a consultant you are posed the following question from a client who is using SAP S/4HANA Cloud, public edition and also SAP BTP, ABAP environment.
'We are currently using an SAP Fiori app based on SAP Fiori elements that analyzes open orders. We have determined that it should be extended via a new button on the UI which will perform an on-the-fly calculation and display the result in a quick popup for the enduser. We have been informed by
SAP that all underlying stack layers for the SAP Fiori app have been extensibility enabled.'
Based on this which of the following extension types would you recommend to the customer to add the new button?

A. RAP BO Behavior Extension
B. SAP HANA database table extension
C. RAP BO Node Extension
D. Business Service Extension

Correct Answer: C
Section:

QUESTION 7
Which function call returns 0?

A. Count_any_of (val - 'ABAP ABAP abap' sub 'AB')
B. Count (val - 'ABAP ABAP abap' sub - 'AB')
C. find_any_of (val = 'ABAP ABAP abap' sub = 'AB')
D. find_any_not_of(val 'ABAP ABAP abap' sub = 'AB')

Correct Answer: D
Section:
Explanation:
The function find_any_not_of returns the position of the first character in the string val that is not contained in the string sub. If no such character is found, the function returns 0. In this case, the string val contains only the characters A, B, and a, which are all contained in the string sub, so the function returns 0. The other functions return positive values, as follows:
Count_any_of returns the number of occurrences of any character in the string sub within the string val. In this case, it returns 8, since there are 8 A's and B's in val.
Count returns the number of occurrences of the string sub within the string val. In this case, it returns 2, since there are 2 AB's in val.
find_any_of returns the position of the first character in the string val that is contained in the string sub. In this case, it returns 1, since the first character A is in sub.Reference:String Functions - ABAP Keyword Documentation,Examples of String Functions - ABAP Keyword Documentation

QUESTION 8
In which products must you use the ABAP Cloud Development Model? Note: There are 2 correct answers to this question.

A. SAP S/4HANA Cloud, private edition
B. SAP BTP, ABAP environment
C. SAP S/4HANA on premise
D. SAP S/4HANA Cloud, public edition

Correct Answer: A, B
Section:
Explanation:
The ABAP Cloud Development Model is the ABAP development model to build cloud-ready business apps, services, and extensions. It comes with SAP BTP and SAP S/4HANA.It works with public or private cloud, and even on-premise1.However, the complete ABAP Cloud Development Model, including the cloud-optimized ABAP language and public local SAP APIs and extension points, is available only in SAP BTP ABAP Environment and in the 2208/2022 versions of the SAP S/4HANA editions1. Therefore, you must use the ABAP Cloud Development Model in SAP BTP, ABAP environment and SAP S/4HANA Cloud, private edition. You can also use it in SAP S/4HANA on premise, but it is not mandatory.You cannot use it in SAP S/4HANA Cloud, public edition, because it does not allow custom ABAP code2.Reference:1: ABAP Cloud | SAP Blogs2: SAP S/4HANA Cloud Extensibility -- Overview and Comparison | SAP Blogs

QUESTION 9
Which RESTful Application Programming object can be used to organize the display of fields in an app?

A. Data model view
B. Metadata extension
C. Service definition
D. Projection view

Correct Answer: B
Section:
Explanation:
A metadata extension is a RESTful Application Programming object that can be used to organize the display of fields in an app. A metadata extension is a CDS view that annotates another CDS view with UI annotations, such as labels, icons, or facets. These annotations define how the data should be presented in the app, such as which fields should be shown on the object page, which fields should be editable, or which fields should be used for filtering or sorting.A metadata extension can also be used to add custom actions or validations to the app12.Reference:1:Refine the Object Page with Annotations | SAP Tutorials2:ABAP RAP : Enabling custom actions with a dialog for additional input fields | SAP Blogs

QUESTION 10
Refer to the exhibit.

```
DATA: go_super TYPE REF TO lcl_super,
      go_sub   TYPE REF TO lcl_sub.
go_sub = NEW #( ... ).
go_super = go_sub.
```

with lcl_super being superclass of lcl_sub.

When accessing the subclass instance through go_super, what can you do? Note: There are 2 correct answers to this question.

A. Access the inherited private components.
B. Access the inherited public components.
C. Call a subclass specific public method
D. Call inherited public redefined methods.

Correct Answer: A, B
Section:
Explanation:
When accessing the subclass instance through go_super, you can do both of the following:
Access the inherited private components: A subclass inherits all the private attributes and methods of its superclass, unless they are explicitly overridden by the subclass.Therefore, you can access the inherited private components of the superclass through go_super, as long as they are not hidden by other attributes or methods in the subclass12.
Access the inherited public components: A subclass inherits all the public attributes and methods of its superclass, unless they are explicitly overridden by the subclass.Therefore, you can access the inherited public components of the superclass through go_super, as long as they are not hidden by other attributes or methods in the subclass12.
You cannot do any of the following:
Call a subclass specific public method: A subclass does not have any public methods that are not inherited from its superclass.Therefore, you cannot call a subclass specific public method through go_super12.
Call inherited public redefined methods: A subclass does not have any public methods that are redefined from its superclass.Therefore, you cannot call inherited public redefined methods through go_super12.

QUESTION 11
Given the following code in an SAP S/4HANA Cloud private edition tenant:

```
1 CLASS zcl_demo_class DEFINITION.
2 METHODS: m1.
3 ENDCLASS..
4 CLASS zcl_demo_class_Implementation.
5 METHOD m1.
6 CALL FUNCTION 'ZF1'.
7 ENDMETHOD
8 ENDCLASS.
```

The class zcl_demo_class is in a software component with the language version set to 'ABAP Cloud'. The function module ZF1' is in a different software component with the language version set to 'Standard ABAP'. Both the class and function module are customer created.
Regarding line #6, which of the following are valid statements? Note: There are 2 correct answers to this question.

A. ZF1' can be called only if it is released for cloud development.
B. 'ZF1' can be called if a wrapper is created for it and the wrapper itself is released for cloud development.
C. 'ZF1' can be called whether it is released or not for cloud development
D. ZF1' can be called if a wrapper is created for it but the wrapper itself is not released for cloud development.

Correct Answer: A, B
Section:
Explanation:
The ABAP Cloud Development Model requires that only public SAP APIs and extension points are used 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 an ABAP Cloud class is not allowed and will result in a syntax error. However, there are two possible ways to call a function module indirectly from an ABAP Cloud class:
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 ''Standard ABAP'' and must be marked as released for cloud development using the annotation @EndUserText.label.The wrapper can then be called from an ABAP Cloud class using the public methods or attributes2.
Use the ABAP Cloud Connector to call the function module as a remote function call (RFC) from an ABAP Cloud class. The ABAP Cloud Connector is a service that enables the secure and reliable communication between SAP BTP, ABAP environment and on-premise systems. The function module must be exposed as an RFC-enabled function module in the on-premise system and must be registered in the ABAP Cloud Connector. The ABAP Cloud class can then use the class cl_rfc_destination_service to get the destination name and the class cl_abap_system to create a proxy object for the function module.The proxy object can then be used to call the function module3.

QUESTION 12
Which restrictions exist for ABAP SQL arithmetic expressions? Note: There are 2 correct answers to this question.

A. Floating point types and integer types can NOT be used in the same expression.
B. The operator/is allowed only in floating point expressions.

C. Decimal types and integer types can NOT be used in the same expression.
D. The operator is allowed only in floating point expressions.

Correct Answer: B, D
Section:
Explanation:
ABAP SQL arithmetic expressions have different restrictions depending on the data type of the operands. The following are some of the restrictions:
Floating point types and integer types can be used in the same expression, as long as the integer types are cast to floating point types using the cast function. For example,CAST (num1 AS FLTP) / CAST (num2 AS FLTP)is a valid expression, where num1 and num2 are integer types.
The operator / is allowed only in floating point expressions, where both operands have the type FLTP or f. For example,num1 / num2is a valid expression, where num1 and num2 are floating point types. If the operator / is used in an integer expression or a decimal expression, a syntax error occurs.
Decimal types and integer types can be used in the same expression, as long as the expression is a decimal expression. A decimal expression has at least one operand with the type DEC, CURR, or QUAN or p with decimal places. For example,num1 + num2is a valid expression, where num1 is a decimal type and num2 is an integer type.
The operator ** is allowed only in floating point expressions, where both operands have the type FLTP or f. For example,num1 ** num2is a valid expression, where num1 and num2 are floating point types. If the operator ** is used in an integer expression or a decimal expression, a syntax error occurs.

QUESTION 13
What are valid statements? Note: There are 2 correct answers to this question.

A. ##NEEDED is checked by the syntax checker.
B. The pragma is not checked by the syntax checker.
C. #EC_NEEDED is not checked by the syntax checker.
D. The pseudo-comment is checked by the syntax checker

Correct Answer: A, B
Section:
Explanation:
Both statements are valid in ABAP, but they have different effects on the program.
##NEEDED is a pragma that can be used to hide warnings from the ABAP compiler syntax check. It tells the check tools that a variable or a parameter is needed for further processing, even if it is not used in the current statement.For example, if you declare a variable without assigning any value to it, you can use ##NEEDED to suppress the warning about unused variables12.
The pragma is not checked by the syntax checker means that you can use any pragma to hide any warning from the ABAP compiler syntax check, regardless of its effect on the program logic or performance.For example, if you use ##SHADOW to hide a warning about an obscured function, you can also use it to hide a warning about an invalid character in a string12.
You cannot do any of the following:
#EC_NEEDED is not checked by the syntax checker: This is not a valid statement in ABAP.There is no pseudo-comment with #EC_NEEDED in ABAP3.
The pseudo-comment is checked by the syntax checker: This is false. Pseudo-comments are obsolete and should no longer be used in ABAP.They were replaced by pragmas since SAP NW 7.0 EhP2 (Enhancement Package)4.

QUESTION 14
Which internal table type allows unique and non-unique keys?

A. Sorted
B. Hashed
C. Standard

Correct Answer: C
Section:
Explanation:
The internal table type that allows both unique and non-unique keys is the standard table. A standard table has an internal linear index that can be used to access the table entries. The key of a standard table is always non-unique, which means that the table can contain duplicate entries. However, the system does not check the uniqueness of the key when inserting new entries, so the programmer can ensure that the key is unique by using appropriate logic. A standard table can be accessed either by using the table index or the key, but the response time for key access is proportional to the table size.

The other two internal table types, sorted and hashed, do not allow non-unique keys. A sorted table is filled in sorted order according to the defined table key, which must be unique. A sorted table can be accessed either by using the table index or the key, but the response time for key access is logarithmically proportional to the table size. A hashed table can only be accessed by using a unique key, which must be specified when declaring the table. A hashed table has no index, and the response time for key access is constant, regardless of the table size.

QUESTION 15
In this nested join below in which way is the join evaluated?

```
SELECT FROM t_a AS a
    LEFT OUTER JOIN t_b AS b
    LEFT OUTER JOIN t_c AS c
    ON c~f1 = b~f1 AND c~f2 = b~f2
    ON b~f1 = a~f1
    WHERE ....
```

A. From the left to the right in the order of the tables: 1. a is joined with b 2. b is joined with c
B. From the right to the left in the order of the tables: 1. b is joined with c. 2. b is joined with a.
C. From the top to the bottom in the order of the on conditions 1. b is joined with c 2. a is joined with b
D. From the bottom to the top in the order of the on conditions: 1. a is joined with b 2. b is joined with c

Correct Answer: C
Section:
Explanation:
The nested join is evaluated from the top to the bottom in the order of the ON conditions. This means that the join expression is formed by assigning each ON condition to the directly preceding JOIN from left to right. The join expression can be parenthesized implicitly or explicitly to show the order of evaluation. In this case, the implicit parentheses are as follows:
SELECT * FROM (a INNER JOIN (b INNER JOIN c ON b~c = c~c) ON a~b = b~b)
This means that the first join expression is b INNER JOIN c ON b~c = c~c, which joins the columns of tables b and c based on the condition that b~c equals c~c. The second join expression is a INNER JOIN (b INNER JOIN c ON b~c = c~c) ON a~b = b~b, which joins the columns of table a and the result of the first join expression based on the condition that a~b equals b~b. The final result set contains all combinations of rows from tables a, b, and c that satisfy both join conditions.

QUESTION 16
What are some features of a unique secondary key? Note: There are 2 correct answers to this question.

A. It is created when a table is filled.
B. It is updated when the modified table is read again.
C. It is created with the first read access of a table.
D. It is updated when the table is modified.

Correct Answer: C, D
Section:
Explanation:
A unique secondary key is a type of secondary key that ensures that the key combination of all the rows in a table is unique.A unique secondary key has two purposes: firstly, to speed up access to the table, and secondly, to enforce data integrity1.
It is created with the first read access of a table: This is true. A unique secondary key is created when an internal table is filled for the first time using the statement READ TABLE or a similar statement.The system assigns a name and an index to each row of the table based on the key fields23.
It is updated when the modified table is read again: This is false. A unique secondary key does not need to be updated when the internal table content changes, because it already ensures data uniqueness.The system uses a lazy update strategy for non-unique secondary keys, which means that it delays updating them until they are actually accessed23.
You cannot do any of the following:
It is created when a table is filled: This is false.As explained above, a unique secondary key is created only with the first read access of a table23.
It is updated when the modified table is read again: This is false.As explained above, a unique secondary key does not need to be updated when the internal table content changes23.

QUESTION 17
Refer to the exhibit.

Given the following Core Data Services View Entity Data Definition,

```
1 @AccessControl.authorizationCheck: #NOT_REQUIRED
2 DEFINE VIEW ENTITY demo_cds_data_source_matrix
3  AS SELECT FROM
4  <source>
5  {
6    KEY field_1,
7    field_2,
8    field_3
9  }
```

Which of the following types are permitted to be used for <source> on line #4? Note: There are 2 correct answers to this question.

A. A database table from the ABAP Dictionary
B. A CDS DDIC-based view
C. An external view from the ABAP Dictionary
D. A database view from the ABAP Dictionary

Correct Answer: A, B
Section:
Explanation:
The <source> clause in the CDS View Entity Data Definition can be used to specify the data source for the view entity.The <source> clause can accept different types of data sources, depending on the type of the view entity1.
A database table from the ABAP Dictionary: This is a valid type of data source for a CDS View Entity Data Definition. A database table from the ABAP Dictionary is a table that is defined in the ABAP Dictionary using the keyword TABLE or TABLE OF.The name of the database table must be unique within its namespace and must not contain any special characters2.
A CDS DDIC-based view: This is also a valid type of data source for a CDS View Entity Data Definition. A CDS DDIC-based view is a view that is defined in the Core Data Services using the keyword DEFINE VIEW ENTITY.The name of the CDS DDIC-based view must be unique within its namespace and must not contain any special characters3.
You cannot do any of the following:
An external view from the ABAP Dictionary: This is not a valid type of data source for a CDS View Entity Data Definition. An external view from the ABAP Dictionary is a view that is defined in an external application using any language supported by SAP, such as SQL, PL/SQL, or Java.The name of the external view must be unique within its namespace and must not contain any special characters4.
A database view from the ABAP Dictionary: This is not a valid type of data source for a CDS View Entity Data Definition. A database view from the ABAP Dictionary is a view that is defined in an external application using any language supported by SAP, such as SQL, PL/SQL, or Java.The name of the database view must be unique within its namespace and must not contain any special characters4.

QUESTION 18
When processing a loop with the statement DO... ENDDO, what system variable contains the implicit loop counter?

A. sy-linno
B. sy-labix
C. sy-subrc
D. sy-index

Correct Answer: D
Section:

Explanation:
When processing a loop with the statement DO... ENDDO, the system variable that contains the implicit loop counter is sy-index. The loop counter is a numeric value that indicates how many times the loop has been executed. The loop counter is initialized to 1 before the first execution of the loop and is incremented by 1 after each execution. The loop counter can be used to control the number of loop iterations or to access the loop elements by index.The loop counter can also be accessed or modified within the loop body, but this is not recommended as it may cause unexpected results or errors1.
For example, the following code snippet uses the loop counter sy-index to display the numbers from 1 to 10:
DO 10 TIMES. WRITE: / sy-index. ENDDO.
The output of this code is:
1 2 3 4 5 6 7 8 9 10

QUESTION 19
Which patterns raise an exception? Note: There are 3 correct answers to this question.

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).

Correct Answer: A, C, E
Section:
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.

QUESTION 20
Which of the following are parts of the definition of a new database table?Note: There are 2 correct answers to this question.

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

Correct Answer: C, D
Section:

QUESTION 21
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?

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.

Correct Answer: B
Section:
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 attributes2.
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:
5

QUESTION 22
Which of the following are features of Core Data Services? Note: There are 3 correct answers to this question.

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

Correct Answer: B, C, E
Section:
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 extensibility12. 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 options34.
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 capabilities5.
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 23
What is the purpose of a foreign key relationship between two tables in the ABAP Dictionary?

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

Correct Answer: B
Section:
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

QUESTION 24
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.

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.

Correct Answer: A, C
Section:
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 expressions1.
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 displayed2.

QUESTION 25
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.

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.

Correct Answer: A, D
Section:
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_itab12. 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 loop12.
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 loop12.
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 loop12.
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 26
Which extensibility type does SAP recommend you use to enhance the existing UI for an SAP Fiori app?

A. Key user
B. Classic
C. Side-by-side
D. Developer

Correct Answer: D
Section:
Explanation:
According to the SAP clean core extensibility and ABAP cloud topic, SAP recommends using developer extensibility to enhance the existing UI for an SAP Fiori app. Developer extensibility allows you to use the UI adaptation editor in SAP Web IDE to modify the UI layout, add or remove fields, and bind them to the data model. You can also use the SAPUI5 framework to create custom controls, views, and controllers. Developer extensibility is based on the in-app extensibility concept, which means that the extensions are part of the same application and are deployed together with the app. Developer extensibility requires developer skills and access to the source

code of the app.Reference: SAP Learning Hub, SAP S/4HANA Cloud Extensibility - In-App Extensibility,SAP Fiori: Extensibility

QUESTION 27
Image:

In the following ABAP SQL code, what are valid case distinctions? Note: There are 2 correct answers to this question.
A)

```
SELECT FROM dbtab1 FIELDS F1,
CASE
WHEN F2 = '1' THEN 'Value 1
WHEN f2'2' THEN 'Value 2' ELSE "Value for the rest' END AS f case
INTO TABLE @et t1.
```

B)

```
SELECT FROM dbtab1 FIELDS f1,
CASE f2
WHEN `1’ THEN 'Value 1'
WHEN '2' THEN 'Value 2'
ELSE "Value for the rest' END AS f_case
INTO TABLE @gt_t1.
```

C)

```
SELECT FROM dbtab1 FIELDS F1,
CASE
WHEN F2 = '1' THEN "Value 1' WHEN f2 < f3 AND f2 = `2’ THEN "Value 2'
WHEN OTHERS 'Value for the rest' ENDCASE AS f_case
INTO TABLE @gt t1.
```

A. Option A

B. Option B
C. Option C

Correct Answer: A, B
Section: