



C_ABAPD_2309^{Q&As}

SAP Certified Associate - Back-End Developer - ABAP Cloud

**Pass SAP C_ABAPD_2309 Exam with 100%
Guarantee**

Free Download Real Questions & Answers **PDF** and **VCE** file from:

https://www.geekcert.com/c_abapd_2309.html

100% Passing Guarantee
100% Money Back Assurance

Following Questions and Answers are all new published by SAP Official
Exam Center

-  **Instant Download** After Purchase
-  **100% Money Back** Guarantee
-  **365 Days** Free Update
-  **800,000+** Satisfied Customers





QUESTION 1

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: AB

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. References: 1: ABAP Cloud | SAP Blogs 2: SAP S/4HANA Cloud Extensibility ?Overview and Comparison | SAP Blogs

QUESTION 2

In a subclass `subl` you want to redefine a component of a superclass `superl`. How do you achieve this? Note: There are 2 correct answers to this question.

- A. You add the clause `REDEFINITION` to the component in `subl`.
- B. You implement the redefined component for a second time in `superl`.
- C. You implement the redefined component in `subl`.
- D. You add the clause `REDEFINITION` to the component in `superl`.

Correct Answer: AC

To redefine a component of a superclass in a subclass, you need to do the following¹²:

You add the clause `REDEFINITION` to the component declaration in the subclass. This indicates that the component is inherited from the superclass and needs to be reimplemented in the subclass. The redefinition must happen in the same

visibility section as the component declaration in the superclass. For example, if the superclass has a public method `m1`, the subclass must also declare the redefined method `m1` as public with the `REDEFINITION` clause.

You implement the redefined component in the subclass. This means that you provide the new logic or behavior for the component that is specific to the subclass. The redefined component in the subclass will override the original component



in the superclass when the subclass object is used. For example, if the superclass has a method m1 that returns `Hello\`, the subclass can redefine the method m1 to return `Hi\` instead.

You cannot do any of the following:

You implement the redefined component for a second time in the superclass. This is not possible, because the superclass already has an implementation for the component that is inherited by the subclass. The subclass is responsible for

providing the new implementation for the redefined component, not the superclass. You add the clause REDEFINITION to the component in the superclass. This is not necessary, because the superclass does not need to indicate that the

component can be redefined by the subclass. The subclass is the one that needs to indicate that the component is redefined by adding the REDEFINITION clause to the component declaration in the subclass.

References:1:METHODS - REDEFINITION - ABAP Keyword Documentation - SAP Online Help2:Redefining Methods - ABAP Keyword Documentation - SAP Online Help

QUESTION 3

What RESTful Application Programming feature is used to ensure the uniqueness of a semantic key?

- A. Validation
- B. Action
- C. Determination

Correct Answer: C

Explanation: The RESTful Application Programming feature that is used to ensure the uniqueness of a semantic key is determination. A determination is a type of behavior implementation that defines a logic that is executed automatically when certain events occur, such as create, update, delete, or activate. A determination can be used to calculate or derive values for certain fields, such as semantic keys, based on other fields or external sources. A determination can also be used to check the uniqueness of a semantic key by comparing it with the existing values in the database or the transaction buffer. A determination can use the ABAP SQL or the EML syntax to access and manipulate data. A determination can be defined using the DETERMINE action clause in the behavior definition of a CDS view entity or a projection view. A determination can also be annotated with the @ObjectModel.determination annotation to specify the event, the timing, and the scope of the determination¹² The other RESTful Application Programming features are not used to ensure the uniqueness of a semantic key, but have different purposes and effects. These features are:

Validation: A validation is a type of behavior implementation that defines a logic that is executed automatically when certain events occur, such as create, update, delete, or activate. A validation can be used to check the consistency and correctness of the data, such as mandatory fields, data types, value ranges, or business rules. A validation can use the ABAP SQL or the EML syntax to access and manipulate data. A validation can be defined using the VALIDATE action clause in the behavior definition of a CDS view entity or a projection view. A validation can also be annotated with the @ObjectModel.validation annotation to specify the event, the timing, and the scope of the validation¹²

Action: An action is a type of behavior implementation that defines a logic that is executed explicitly by the user or the application. An action can be used to perform a specific business operation, such as creating, updating, deleting, or activating an entity instance, or triggering a workflow or a notification. An action can use the ABAP SQL or the EML syntax to access and manipulate data. An action can be defined using the ACTION clause in the behavior definition of a CDS view entity or a projection view. An action can also be annotated with the @ObjectModel.action annotation to specify the name, the description, the parameters, and the visibility of the action¹²

References: Behavior Implementation - ABAP Keyword Documentation, Behavior Definition

-ABAP Keyword Documentation



QUESTION 4

You want to provide a short description of the data definition for developers that will be attached to the database view (Sorry, we do not have a more clear image. If we have a better resource for the image, we will update this one immediately.)

```
1 @AccessControl.authorizationCheck: #CHECK
2
3
4 SELECT FROM sflight AS sflight
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

Which of the following annotations would do this if you inserted it on line #27

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

Correct Answer: D

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.

References: 1: ABAP CDS - SAP Annotations - ABAP Keyword Documentation - SAP Online Help 2: ABAP CDS - Data Definitions - ABAP Keyword Documentation - SAP Online Help

QUESTION 5

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

- A. SOAP consumption
- B. CDS Views
- C. Business Add-ins (BADls)
- D. Business Events
- E. OData services

Correct Answer: ADE

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. References: Consuming SOAP Services - ABAP Keyword Documentation, Business Events - ABAP Keyword Documentation, OData Services - ABAP Keyword Documentation, CDS Data Model Views - ABAP Keyword Documentation, [Business Add- Ins (BADls) - ABAP Keyword Documentation]



[Latest C_ABAPD_2309 Dumps](#)

[C_ABAPD_2309 VCE Dumps](#)

[C_ABAPD_2309 Study Guide](#)