

C_ABAPD_2507試験の準備方法 | 権威のある C_ABAPD_2507資格トレーニング試験 | 高品質なSAP Certified Associate - Back-End Developer - ABAP Cloud 最新問題

C_ABAPD_2507	
Number of Questions:	80
Question Types:	Multiple-choice and multiple-response
Duration:	180 minutes
Cut Score:	65%
Available Languages:	English (others may be available)

無料でクラウドストレージから最新のPass4Test C_ABAPD_2507 PDFダンプをダウンロードする：<https://drive.google.com/open?id=1GmuTJkQzSFcMxpB12ym-b3Ntgx5yrst>

一般的には、IT技術会社ではSAP C_ABAPD_2507資格認定を持つ職員の給料は持たない職員の給料に比べ、15%より高いです。これなので、IT技術職員としてのあなたはPass4TestのSAP C_ABAPD_2507問題集デモを参考し、試験の準備に速く行動しましょう。我々社はあなたがSAP C_ABAPD_2507試験に一発的に合格するために、最新版の備考資料を提供します。

我々はC_ABAPD_2507試験に失敗したら全額で返金するという承諾をしています。お客様は我々の商品を利用したら、試験の出題率は100%とはいきませんが、85%程度は出題されました、もし不幸であなたはC_ABAPD_2507試験に失敗したら、あなたは失敗した成績書のスキャンを我々のメールアドレスに送って、我々は失敗の原因を問わず、あなたの支払ったC_ABAPD_2507問題集の金額を全額であなたに返してあなたの経済損失を減少します。

>> C_ABAPD_2507資格トレーニング <<

SAP C_ABAPD_2507最新問題 & C_ABAPD_2507独学書籍

Pass4Testは、受験者向けのC_ABAPD_2507試験資料を作成するための専門的なプラットフォームです。C_ABAPD_2507試験に合格し、関連する認定をより効率的で簡単な方法で取得できるようお手伝いします。当社のC_ABAPD_2507試験材料の優れた品質とリーズナブルな価格により、当社のC_ABAPD_2507試験トレントは、国際分野の他のメーカーよりも価格が優れているだけでなく、多くの点で明らかに優れています。C_ABAPD_2507試験問題集の合格率は99%~100%であり、これは市場で独特です。

SAP C_ABAPD_2507 認定試験の出題範囲：

トピック	出題範囲
トピック 1	ABAP RESTful Application Programming Model: This section of the exam measures skills of SAP Application Programmers and covers the fundamentals of the ABAP RESTful Application Programming Model (RAP). It includes topics such as behavior definitions, service binding, and the use of managed and unmanaged scenarios. The focus is on building modern, scalable, and cloud-ready applications using RAP.

トピック 2	ABAP Core Data Services and Data Modeling: This section of the exam measures skills of SAP ABAP Developers and covers the creation, definition, and use of Core Data Services (CDS) views for data modeling within SAP environments. Candidates are expected to understand annotations, data definitions, and the role of CDS in enabling advanced data processing and integration across SAP systems.
トピック 3	Core ABAP Programming: This section of the exam measures skills of SAP Application Programmers and covers foundational ABAP programming knowledge. Topics include modularization techniques, internal tables, control structures, and classical report programming. Mastery of these concepts is essential for building efficient ABAP applications.
トピック 4	SAP Clean Core Extensibility and ABAP Cloud: This section of the exam measures skills of SAP Application Programmers and covers the clean core principles and extensibility options within SAP BTP. It also includes cloud-native ABAP development practices, emphasizing the creation of upgrade-stable and maintainable extensions aligned with SAP's cloud strategy.
トピック 5	ABAP SQL and Code Pushdown: This section of the exam measures skills of SAP ABAP Developers and covers the use of advanced SQL techniques within ABAP. It includes code pushdown strategies that leverage database-level processing to enhance application performance. Key areas include Open SQL enhancements and integrating logic closer to the database.

SAP Certified Associate - Back-End Developer - ABAP Cloud 認定 C_ABAPD_2507 試験問題 (Q72-Q77):

質問 # 72

You have a superclass `super1` and a subclass `sub1` of `super1`. Each class has an instance constructor and a static constructor. The first statement of your program creates an instance of `sub1`.

In which sequence will the constructors be executed?

- A. Class constructor of `sub1` # Instance constructor of `super1` # Instance constructor of `sub1` # Class constructor of `super1`
- B. Class constructor of `super1` # Class constructor of `sub1` # Instance constructor of `super1` # Instance constructor of `sub1`
- C. Instance constructor of `sub1` # Instance constructor of `super1` # Class constructor of `super1` # Class constructor of `sub1`
- D. Instance constructor of `super1` # Class constructor of `super1` # Class constructor of `sub1` # Instance constructor of `sub1`

正解: B

解説:

Comprehensive and Detailed Explanation from Exact Extract:

Execution order when creating an instance of a subclass:

- * Class constructor of the superclass (`super1`) executes first.
- * Class constructor of the subclass (`sub1`) executes second.
- * Then the instance constructor of the superclass (`super1`) executes.
- * Finally, the instance constructor of the subclass (`sub1`) executes.

This sequence guarantees that both the static (class-level) and instance-level initializations of the superclass are complete before the subclass is constructed.

Verified Study Guide Reference: ABAP Objects Programming Guide - Class and Instance Constructor Execution Order.

質問 # 73

You want to join two database tables, `T_CARRIER` and `T_CONNECTIONS`, to retrieve all carriers, whether they have corresponding connections or not.

Which statements would achieve this?

Note: There are 2 correct answers to this question.

- A. `SELECT FROM t_carrier`
`LEFT INNER JOIN t_connections`
`ON ...`
- B. `SELECT FROM t_connections`
`RIGHT OUTER JOIN t_carrier`

- ON ...
- C. SELECT FROM t_carrier
LEFT OUTER JOIN t_connections
ON ...
- D. SELECT FROM t_carrier
INNER JOIN t_connections
ON ...

正解: B、C

解説:

Comprehensive and Detailed Explanation From Exact Extract:

The requirement is:

* Retrieve all carriers from T_CARRIER

* Include them even if no corresponding connections exist in T_CONNECTIONS.

Evaluation of each join type:

* A. INNER JOIN: Only returns rows where a carrier has at least one matching connection. # Incorrect, since carriers without connections would be excluded.

* B. LEFT OUTER JOIN (correct): Returns all rows from the left table (T_CARRIER), and connections if they exist. Missing connections are represented with NULL. # Correct answer.

* C. LEFT INNER JOIN: This is syntactically invalid in ABAP SQL. INNER JOIN and LEFT OUTER JOIN are separate join types, not combined. # Incorrect.

* D. RIGHT OUTER JOIN (correct): Equivalent to LEFT OUTER JOIN when the tables are reversed.

Returns all rows from T_CARRIER, whether or not connections exist. # Correct answer.

Thus, LEFT OUTER JOIN and RIGHT OUTER JOIN are the valid solutions to retrieve all carriers regardless of connections.

Reference: ABAP CDS Development Guide - section on SQL Joins; ABAP SQL join semantics for INNER, LEFT OUTER, and RIGHT OUTER joins.

質問 # 74

What are some characteristics of secondary keys for internal tables? Note: There are 3 correct answers to this question.

- A. Secondary keys must be chosen explicitly when you actually read from an internal table.
- B. Sorted secondary keys do NOT have to be unique.
- C. Secondary keys can only be created for standard tables.
- D. Hashed secondary keys do NOT have to be unique.
- E. Multiple secondary keys are allowed for any kind of internal table.

正解: A、B、E

解説:

Secondary keys are additional keys that can be defined for internal tables to optimize the access to the table using fields that are not part of the primary key. Secondary keys can be either sorted or hashed, depending on the table type and the uniqueness of the key. Secondary keys have the following characteristics1:

A . Secondary keys must be chosen explicitly when you actually read from an internal table. This means that when you use a READ TABLE or a LOOP AT statement to access an internal table, you have to specify the secondary key that you want to use with the USING KEY addition. For example, the following statement reads an internal table itab using a secondary key sec_key:

```
READ TABLE itab USING KEY sec_key INTO DATA(wa).
```

If you do not specify the secondary key, the system will use the primary key by default2.

B . Multiple secondary keys are allowed for any kind of internal table. This means that you can define more than one secondary key for an internal table, regardless of the table type. For example, the following statement defines an internal table itab with two secondary keys sec_key_1 and sec_key_2:

```
DATA itab TYPE SORTED TABLE OF ty_itab WITH NON-UNIQUE KEY sec_key_1 COMPONENTS field1 field2  
sec_key_2 COMPONENTS field3 field4.
```

You can then choose which secondary key to use when you access the internal table1.

D . Sorted secondary keys do NOT have to be unique. This means that you can define a sorted secondary key for an internal table that allows duplicate values for the key fields. A sorted secondary key maintains a predefined sorting order for the internal table, which is defined by the key fields in the order in which they are specified. For example, the following statement defines a sorted secondary key sec_key for an internal table itab that sorts the table by field1 in ascending order and field2 in descending order:

```
DATA itab TYPE STANDARD TABLE OF ty_itab WITH NON-UNIQUE SORTED KEY sec_key COMPONENTS field1  
ASCENDING field2 DESCENDING.
```

You can then access the internal table using the sorted secondary key with a binary search algorithm, which is faster than a linear search³.

The following are not characteristics of secondary keys for internal tables, because:

C . Hashed secondary keys do NOT have to be unique. This is false because hashed secondary keys must be unique. This means that you can only define a hashed secondary key for an internal table that does not allow duplicate values for the key fields. A hashed secondary key does not have a predefined sorting order for the internal table, but uses a hash algorithm to store and access the table rows. For example, the following statement defines a hashed secondary key `sec_key` for an internal table `itab` that hashes the table by `field1` and `field2`:

```
DATA itab TYPE STANDARD TABLE OF ty_itab WITH UNIQUE HASHED KEY sec_key COMPONENTS field1 field2.
```

You can then access the internal table using the hashed secondary key with a direct access algorithm, which is very fast.

E . Secondary keys can only be created for standard tables. This is false because secondary keys can be created for any kind of internal table, such as standard tables, sorted tables, and hashed tables. However, the type of the secondary key depends on the type of the internal table. For example, a standard table can have sorted or hashed secondary keys, a sorted table can have sorted secondary keys, and a hashed table can have hashed secondary keys¹.

質問 # 75

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.

- A. The name of a type defined privately in class `ZCL_CLASS_A`
- B. The name of a type defined privately in another class
- C. The name of a domain from the ABAP Dictionary
- D. The name of a data element from the ABAP Dictionary

正解: C、D

解説:

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.

質問 # 76

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

- A. Sorted
- B. Hashed
- C. Standard

正解: A

質問 # 77

.....

- P.S. Pass4TestがGoogle Driveで共有している無料かつ新しいC_ABAPD_2507ダンプ: <https://drive.google.com/open?id=1GmuTJkQzSFcMxpB12ym-b3Ntgx5vrst>