

SAP ABAP on HANA

Interview Questions and Answers



Learn more in detail

Enrol on the course

[SAP ABAP on HANA Training for Beginners](#)

On Udemy.com

1Q: What is SAP HANA?

A: SAP HANA is an in-memory, column-oriented, relational database management system developed by SAP. It is designed to process large amounts of data in real-time, enabling businesses to make faster and more informed decisions.

2Q: What is SAP ABAP on HANA?

A: SAP ABAP on HANA is a programming language used to develop applications in the SAP HANA database environment. ABAP on HANA provides enhanced performance and improved processing speed by leveraging the in-memory computing capabilities of the HANA database.

It provides improved performance, real-time analytics, simplified development, and enhanced scalability.

3Q: What is difference between SAP HANA and S/4 HANA?

A: SAP HANA is an in-memory database platform that enables real-time data processing and analysis. It can be used as a standalone database platform or as the underlying technology for other SAP applications.

SAP S/4HANA, on the other hand, is a suite of enterprise resource planning (ERP) applications that are built on top of the SAP HANA platform. S/4HANA is a next-generation business suite that is designed to help businesses run more efficiently by providing real-time insights and streamlined processes.

In summary, SAP HANA is a technology platform, while SAP S/4HANA is an ERP suite that is built on top of the SAP HANA platform.

4Q: What is the difference between traditional ABAP and ABAP on HANA?

A: The main difference between traditional ABAP and ABAP on HANA is that ABAP on HANA uses the HANA database technology to process data faster and more efficiently. This means that ABAP on HANA applications can leverage the power of HANA to provide real-time data processing and analysis.

SAP ABAP on HANA

Interview Questions and Answers



Learn more in detail

Enrol on the course

[SAP ABAP on HANA Training for Beginners](#)

On Udemy.com

5Q: What are the main features of SAP HANA?

A:

- **In-memory Database:**

With SAP HANA, you can now store the complete database in memory. This means that disk movement is not needed and swapping can be eliminated. Hence increase in the speed of read-write access.

- **Multicore CPU and parallel processing:**

To get the best performance from new advanced hardware, SAP HANA makes use of parallelism by using all the cores of a CPU, and several CPUs.

Column store tables are automatically processed in parallel.

Even the same column can be split up and processed by different cores at the same time.

- **Column and Row storage:**

SAP HANA supports both col and row store tables.

By accessing data in column-store order, you benefit immensely from simplified table scan and data pre-caching. This can make all the difference in performance.

With column store, SAP HANA scans columns of data so quickly that additional indexes are usually not required.

It is easy to alter column store tables without dropping and reloading data.

Column store tables are optimal for parallel processing, as each core is able to work on a different column.

- **Data Compression**

SAP HANA uses a dictionary per column Operates directly on compressed data using integers.

It reduces the amount of memory required.

It speeds up operations on columns because the columns can be loaded into the CPU caches faster and with fewer loading cycles.

SAP ABAP on HANA

Interview Questions and Answers



Learn more in detail

Enrol on the course

[SAP ABAP on HANA Training for Beginners](#)

On Udemy.com

- **Aggregates and Indexes.**

Using the power of SAP HANA, you can aggregate on the fly from any line-item table. You do not need prebuilt aggregates. SAP HANA organizes data using column stores, which means that indexes are not needed. They can still be created but offer little improvement.

6Q: What is the in-memory database?

A: Storing data in main memory rather than on disk provides faster data access, faster querying and processing. SAP HANA uses Dynamic tiering, frequently accessed "hot" data is stored on main memory and less frequently accessed "warm" data is stored on disk.

7Q: What is the difference between a traditional database and SAP HANA?

A: The main difference between a traditional database and SAP HANA is that SAP HANA is an in-memory database that stores data in columns rather than rows. This enables faster data processing and real-time analytics capabilities.

8Q: Advantages of column store over row store?

A: you benefit immensely from simplified table scan and data pre-caching. only the required columns are loaded to memory, so you avoid using up memory with columns that will never be used the data is arranged efficiently with all values of a column appearing one after another. This continuous sequencing of the column values is preferred by the CPU. It is easy to alter column store tables without dropping and reloading data.

9Q: Guidelines to increase ABAP code performance on SAP HANA database?

A:

- Keep result set small.
- Minimize the number of data transfer.
- Minimize number of database accesses.
- Minimize search overhead.

SAP ABAP on HANA

Interview Questions and Answers



Learn more in detail

Enrol on the course

[SAP ABAP on HANA Training for Beginners](#)

On Udemy.com

- Keep unnecessary load away from database.

10Q: Can we create secondary index in SAP HANA?

A: Yes, A secondary index in SAP HANA is an additional index that can be created on a table to improve the performance of specific queries.

11Q: What are the benefits of data compression in SAP HANA?

A: The benefits of data compression in SAP HANA include:

- Reduced memory and storage requirements
- Improved query performance due to reduced data volume
- Lower costs for hardware and storage
- Faster data backups and restores.

12Q: What are the benefits of parallel processing in SAP HANA?

A: The benefits of parallel processing in SAP HANA include:

- Faster query performance due to the ability to process tasks concurrently.
- Improved scalability for larger workloads
- Ability to leverage distributed computing resources for improved processing power.
- Reduced hardware costs compared to traditional single-node architectures.

13Q: What are the benefits of SAP HANA multi-CPU architecture?

A: SAP ABAP HANA multi-CPU architecture offers several benefits, including increased processing power, improved data processing speed, and high availability. By distributing the workload across multiple CPUs, the system can handle larger data volumes and execute complex queries faster. Additionally, the system can continue to operate even if one or more CPUs fail, ensuring high availability.

SAP ABAP on HANA

Interview Questions and Answers



Learn more in detail

Enrol on the course

[SAP ABAP on HANA Training for Beginners](#)

On Udemy.com

14Q: How is code push down implemented in SAP ABAP HANA?

A: Code push down in SAP ABAP HANA is implemented through the use of new ABAP language constructs, such as inline declarations and expressions, which allow developers to perform complex database operations directly in the database layer. Additionally, SAP provides tools such as ABAP Managed Database Procedures (AMDPs) and Core Data Services (CDS) to help developers implement code push down techniques in their applications.

15Q: What are the benefits of code push down in SAP ABAP HANA?

A: The benefits of code push down in SAP ABAP HANA include improved application performance, reduced network traffic and memory usage, and improved scalability. By pushing processing down to the database layer, the application can take advantage of the in-memory computing capabilities of the HANA database, resulting in faster data access and processing.

16Q: What is SAP ADT?

A: SAP ADT stands for ABAP Development Tools. It is an integrated development environment (IDE) used by developers to develop, debug, and deploy ABAP applications in the SAP system.

SAP ABAP on HANA

Interview Questions and Answers



Learn more in detail

Enrol on the course

[SAP ABAP on HANA Training for Beginners](#)

On Udemy.com

17Q: Advantages of using ADT over ABAP workbench?

A:

ABAP Development Tool (ADT)	ABAP Workbench (SE80)
The functions and features of Eclipse and ADT lend themselves particularly well to object-oriented programming,	ABAP Workbench is better suited — in particular working with module pools and screens.
It is possible to open many objects in editor, no limit of 6 sessions.	It is possible to open 6 objects in editor
Many dialogs can be flexible arranged on one screen	-
It also enables cross-platform development by integrating ABAP and non-ABAP development tools in a single, powerful integrated development environment (IDE).	Support only ABAP environment
The architecture of ABAP Development Tools for SAP NetWeaver is designed to allow developers to use a single Eclipse client that connects to several ABAP backend systems of different releases. This allows developers to control all activities in a central location and helps avoid time-consuming context switches.	-

18Q: How do you create a new ABAP project in SAP ADT?

A: To create a new ABAP project in SAP ADT, follow these steps:

1. Open the SAP Development perspective in Eclipse.
2. Right-click on the ABAP Development node in the Project Explorer and select New > ABAP Project.
3. Enter a name and description for the project, and select the system and client where the project will be created.
4. Select the package where the project will be saved and specify any additional project settings.
5. Click Finish to create the project.

SAP ABAP on HANA

Interview Questions and Answers



Learn more in detail

Enrol on the course

[SAP ABAP on HANA Training for Beginners](#)

On Udemy.com

19Q: How do you debug an ABAP program in SAP ADT?

A: To debug an ABAP program in SAP ADT, follow these steps:

1. Open the ABAP Development perspective in Eclipse.
2. Navigate to the program you want to debug in the Object Navigator.
3. Set a breakpoint in the program by double-clicking on the line of code where you want to pause the program.
4. Execute the program in debug mode by right-clicking on the program and selecting Debug As > ABAP Application.
5. Step through the program line by line using the debug toolbar and inspect variables and objects in the Debug Perspective.

20Q: What is the CDS view?

A: To take advantage of SAP HANA for application development, SAP introduced a new data modelling infrastructure known as core data services.

CDS is an enhancement of SQL which provides a Data Definition Language (DDL) for defining semantically rich database tables/views (CDS entities) and user-defined types in the database. Some of the enhancements are:

- Expressions used for calculations and queries in the data model
- Associations on a conceptual level, replacing joins with simple path expressions in queries
- Annotations to enrich the data models with additional (domain-specific) metadata

21Q: Explain CDS-related Repository Objects?

A:

- **Data Definition:**

Also referred to as DDL Source (for Data Definition Language, named after the DDL part of SQL).

Contains the definition of either a CDS View or a CDS Table function.

Display only in ABAP workbench.

Editing requires the use of the ABAP Development Tool (ADT in Eclipse).

SAP ABAP on HANA

Interview Questions and Answers



Learn more in detail

Enrol on the course

[SAP ABAP on HANA Training for Beginners](#)

On Udemy.com

- **Access Control:**

Also referred to as DCL Source (for Data Control Language, named after the DCL part of SQL).

Contains the definition of authorization rules that are automatically checked when a program accesses a certain CDS View or CDS table function.

Display only in ABAP workbench.

Editing requires the use of the ABAP Development Tool (ADT in Eclipse).

22Q: What are the advantages of using CDS views?

A: CDS Views offer several benefits, including:

- Improved performance: CDS Views are optimized for SAP HANA, so they provide fast data access and processing.
- Reusability: CDS Views can be reused across different applications, reducing the need for redundant code and data models.
- Simplified data modeling: CDS Views provide a simplified and standardized way to define data models, making it easier to develop and maintain applications.
- Integration with other SAP technologies: CDS Views can be integrated with other SAP technologies, such as SAP Fiori and SAP BW, to provide a seamless user experience.

SAP ABAP on HANA

Interview Questions and Answers



Learn more in detail
Enrol on the course
[SAP ABAP on HANA Training for Beginners](#)
On Udemy.com

23Q: Difference between ABAP dictionary view and ABAP CDS view?

A:

	ABAP Dictionary View	ABAP CDS view
Support on all DBMS	Yes	Yes
Support combining queries	Inner join	Inner join, Outer Joins, Union
Calculation	No	Aggregation, grouping, Calculation expression
Nested Views (View-on-View)	No	Yes

24Q: Difference between ABAP CDS and HANA CDS?

A:

ABAP CDS	HANA CDS
Support all DBs	Support only HANA DBs
ABAP CDS is located on the Application Server ABAP	HANA CDS is located on the SAP HANA database directly
Support the implementation of ABAP applications – independent of the database system.	Support the development of SAP HANA native applications (SAP HANA XS).
Initial focus on View Building	Initial focus on building models from scratch

SAP ABAP on HANA

Interview Questions and Answers



Learn more in detail

Enrol on the course

[SAP ABAP on HANA Training for Beginners](#)

On Udemy.com

25Q: How do you define a CDS View in SAP?

A: A CDS View is defined using a DDL (Data Definition Language) statement in the ABAP Development Tools (ADT) or SAP HANA Studio. The statement defines the view name, data fields, and other attributes, such as filtering and sorting criteria.

26Q: What is a CDS extension?

A: CDS extensions are used to enhance existing CDS views by adding additional fields or annotations. They allow developers to modify the behaviour of CDS views without having to create a new view from scratch.

27Q: What is Associations? How it is different from Join?

A: An association defines relationship entities. An association associates the current CDS view as a source data source with the target data source target using an ON condition.

Although associations and joins look different, there is no difference on the database level. Eventually, any association is translated into an ordinary join. But in the case of so-called exposed associations, it depends on the way a view is consumed. The join is only executed if the consumer requests data from the associated data source. This can have a positive effect on the performance and sometimes is referred to as "JOIN on Demand".

Associations may contain additional semantic information such as cardinality.

28Q: What are the types of associations?

A:

- Ad-hoc association
- Exposed association
- Filtered association.

SAP ABAP on HANA

Interview Questions and Answers



Learn more in detail

Enrol on the course

[SAP ABAP on HANA Training for Beginners](#)

On Udemy.com

29Q: What is Union and union all?

A: UNION joins the result sets of two queries.

The rows of the result set of the query after UNION are inserted into the result set of the query before UNION.

If the addition ALL is not specified, all duplicate entries are removed from the results set. They are not removed if ALL is specified.

30Q: How CDS view and SQL view are related?

A: A CDS View is defined in a DDL Source, which is a new type of repository object.

Upon activation of a DDL Source, two objects are created: the SQL View and the CDS View.

The SQL View is visible as an object in the ABAP Dictionary where it cannot be edited and only reveals a fraction of the information available in the DDL source. It serves as a representative of the database object.

The CDS View carries more semantics than its SQL view. It is not created on the Database and it is not visible in the ABAP Dictionary. It can, however, be consumed via open SQL.

31Q: What is the use of Annotation?

A: Annotations enrich the CDS definition with metadata.

It starts with character @.

The annotation specifies the properties and semantics of an entity and its behaviour when it is consumed.

32Q: What is the cardinality in an association?

A: Cardinality in an association represents the number of occurrences of an entity that can be associated with another entity. The possible values for cardinality are:

- **[0..1]**: zero or one occurrence of the entity can be associated with the current entity.
- **[1]**: exactly one occurrence of the entity can be associated with the current entity.
- **[0..n]**: zero to many occurrences of the entity can be associated with the current entity.
- **[1..n]**: one to many occurrences of the entity can be associated with the current entity

SAP ABAP on HANA

Interview Questions and Answers



Learn more in detail

Enrol on the course

[SAP ABAP on HANA Training for Beginners](#)

On Udemy.com

33Q: What is AMDP?

A: Allows developers to create and execute database procedures in an ABAP environment using ABAP method. AMDP is the top-down approach of using HANA Database Procedures in ABAP. Allow the execution of complex calculations inside the HANA database. AMDP only requires ABAP Application Server (AS) for developing, managing and calling database procedures.

34Q: What makes AMDP class and ABAP method different from normal class and method?

A:

AMDP class

- Can contain both regular methods and AMDP methods.
- One or more AMDP methods can be present in AMDP class.
- Can only be created using ADT.
- class with AMDPs must implement interface IF_AMDP_MARKER_HDB

AMDP Methods

- AMDP methods can be defined in the public, protected, or private visibility section of the class.
- Although you can define AMDP methods as instance methods, they are always executed like static methods.
- All AMDP method parameters have to be passed by value, and must be of either table or scalar types. Pass by reference is not permitted
- Exporting, importing and changing parameters are allowed. Returning parameters are not allowed
- AMDP Method is specified with addition BY DATABASE PROCEDURE in the method implementation part, followed by the database system (for example, HDB) and the implementation language (for example, SQLScript).
- For parameters of table types, the line types have to consist of elementary components, because nested tables are not supported. ABAP Dictionary structure types are not allowed.

SAP ABAP on HANA

Interview Questions and Answers



Learn more in detail
Enrol on the course
[SAP ABAP on HANA Training for Beginners](#)
On Udemy.com

35Q: Comparison between ABAP CDS view and AMDP

A:

ABAP CDS views	AMDP
Work on all DBs supported by SAP	Work only on SAP HANA DB
Use it for reusable queries	Use it for complex queries which are not handled by ABAP CDS views
Only 1 result set can be returned	Multiple result set can be returned
-	Use database specific programming language, such as SAP HANA SQL script
-	Features of native SQL functions can be leveraged.

36Q: What are features of new OPEN SQL ?

A. New Open SQL (>= ABAP 7.40 SP05)

- Support arithmetic expressions with operators + - * /
- Arithmetic functions FLOOR, CEIL, MOD, ABS, DIV
- Support CASE expression
- Support right outer join
- Support UNION and UNION ALL
- The maximum number of tables supported in Open SQL JOIN clauses has been increased to 50.
- The maximum number of sub-queries supported in an Open SQL statement has been increased from 9 to 50.
- Access to SAP HANA built-in functions, for example, DAYS_BETWEEN()
- Access to SAP HANA views and procedures

SAP ABAP on HANA

Interview Questions and Answers



Learn more in detail

Enrol on the course

[SAP ABAP on HANA Training for Beginners](#)

On Udemy.com

37Q: What is CDS Table Function?

A: The AMDP framework supports AMDP functions alongside the existing AMDP procedures.

AMDP functions use the new addition BY DATABASE FUNCTION of the METHOD statement in AMDP classes.

AMDP functions are functional methods of global classes which define functions stored and executed on the database. It is not allowed to call functional methods that define AMDP functions directly in ABAP

ABAP CDS introduced CDS table functions to make AMDP functions available as data sources of SELECT statements.

38Q: What is the difference between AMDP procedure and AMDP function?

A:

AMDP Procedure	AMDP Function
AMDP procedure implementation can be consumed by ABAP program directly.	AMDP function implementations cannot be called by an ABAP program directly. But they can be called from other AMDP functions or procedures.
AMDP procedure cannot be consumed by the CDS view.	AMDP function can be consumed by the other CDS view via the CDS table function.
AMDP Procedure uses the addition BY DATABASE PROCEDURE of the METHOD statement in AMDP classes.	AMDP functions use the addition BY DATABASE FUNCTION of the METHOD statement in AMDP classes.
AMDP procedure supports importing and exporting parameters	Unlike AMDP procedure implementations, AMDP function implementations have no export parameters. They have exactly one tabular return value (returning parameter) and besides that only import parameter.
AMDP procedure can return multiple result sets.	AMDP function can return single result set
Possible use case:	CDS table functions are implemented using AMDP functions
let's say you want to create an ABAP report where you want a code push-down approach to fetch data based on some complex logic that you think can't be handled by the CDS view or needs multiple results set in return, in such case you can use the AMDP procedure. AMDP procedures can't be used as a data source for other CDS views.	Possible use case: In case you want to create a consumption CDS view build upon multiple CDS views for FIORI application, let's say in one of the underlying CDS views you want to use any native HANA function feature in such case you can use the AMDP function wrapped in CDS table function and it can be used as a data source for other CDS views or ABAP reports.

SAP ABAP on HANA

Interview Questions and Answers



Learn more in detail

Enrol on the course

[SAP ABAP on HANA Training for Beginners](#)

On Udemy.com

39Q: How to use Select-Options in CDS view?

A:

SELECT OPTIONS is not an SQL feature so

1. Conversion of the selection tables into an SQL WHERE clause using method
CL_SHDB_SELTAB=>COMBINE_SEL TABS()
2. Handling of dynamic WHERE clauses within the AMDP function method using the function APPLY_FILTER

40Q: What is SAP ADBC?

A: SAP ADBC stands for ABAP Database Connectivity, and it is a programming interface that allows ABAP programs to access relational databases using SQL statements. ADBC provides a way to connect to databases and execute SQL statements without needing to use Open SQL, which is limited to accessing the database schema defined in the ABAP Dictionary.

41Q: How to create ADBC?

A: Below steps are involved in creations of ADBC

- Call method get_connection() of class CL_SQL_CONNECTION to get database connection only when accessing secondary DB
- Create a statement object: Instantiation of class CL_SQL_STATEMENT
- Fill string variable with SQL syntax
- Call method execute_query() of class CL_SQL_STATEMENT to issue native SQL call
- Call method set_param() or set_param_table() of class CL_SQL_RESULT_SET to assign target variable for result set:
- Call method next_package() of class CL_SQL_RESULT_SET to retrieve result set:
- Call Method close() of class CL_SQL_RESULT_SET to close result and release resources

SAP ABAP on HANA

Interview Questions and Answers



Learn more in detail

Enrol on the course

[SAP ABAP on HANA Training for Beginners](#)

On Udemy.com

42Q: **When to use ABAP SQL, CDS views, AMDP and ADBC?**

A:

ABAP SQL, CDS views, and AMDP (ABAP Managed Database Procedures) are all options for accessing and manipulating data in SAP systems. The choice of which to use depends on various factors such as performance, complexity, and data structure.

- **ABAP SQL** should be used when simple data retrieval or manipulation is required, and the underlying database tables or views are well-structured. ABAP SQL can be used for basic SELECT, INSERT, UPDATE, and DELETE statements, and it is suitable for simple queries that do not require complex data processing.
- **CDS views** should be used when a more complex view of data is required, and when the data needs to be accessed from multiple sources. CDS views can be used to define complex joins, unions, and aggregations, and they are optimized for performance. They are also suitable for creating reports and analytical applications.
- **AMDP** should be used when complex data processing is required, and when the data manipulation cannot be done efficiently using ABAP SQL or CDS views alone. AMDP allows developers to write database procedures in ABAP that can be executed on the database server, rather than in the application server, which can result in improved performance. AMDP is suitable for complex data transformations and calculations that require significant processing power.
- **ABAP Database Connectivity (ADBC)** is used in SAP ABAP on HANA systems when a direct connection to the database is required for data access and manipulation. ADBC is particularly useful when accessing and manipulating large volumes of data or when executing complex SQL statements that cannot be easily handled using ABAP Open SQL or CDS Views. It is useful when working with legacy database code that cannot be easily migrated to modern data access technologies such as CDS views or AMDP.

SAP ABAP on HANA

Interview Questions and Answers

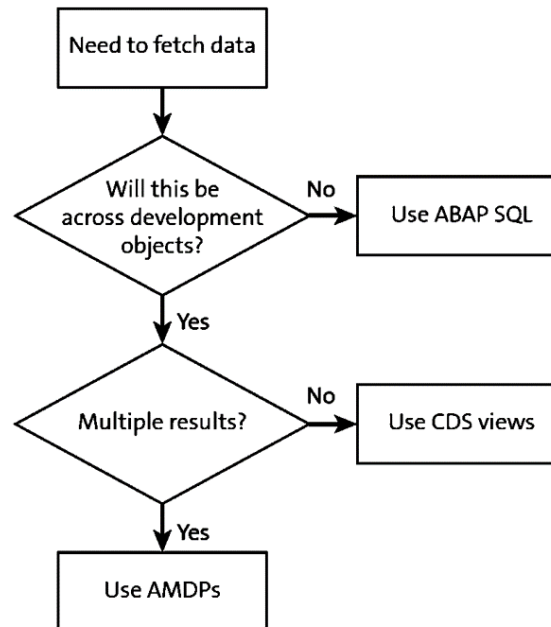


Learn more in detail

Enrol on the course

[SAP ABAP on HANA Training for Beginners](#)

On Udemy.com



43Q: **What is ALV IDA?**

A:

The SAP List Viewer with Integrated Data Access (ALV with IDA) offers ABAP developers the option to take advantage of SAP HANA, without having to present the user with a new or different interface.

SAP ABAP on HANA

Interview Questions and Answers



Learn more in detail

Enrol on the course

[SAP ABAP on HANA Training for Beginners](#)

On Udemy.com

44Q: What is difference between classical ALV and ALV IDA?

A:

Feature	Classic ALV	ALV IDA	Limitations for ALV IDA
Multi-db support	Yes	Yes	
Sorting	Yes	Yes	Without currency/unit consideration (must first sort on currency/UOM)
Selection Screen	Dealt with by application	Yes - select-options and parameters to be passed to the ALV API	
Filtering	Yes	Yes	
Aggregation	Yes	Yes	No unit/currency split if aggregating amounts/currency values
Grouping	No	Yes	
Find	Yes	Yes	Only text fields No fields with <u>conversion</u> exits
Text and Fuzzy Search	No	Yes	
Authorization Checks	Dealt with by application	Yes	
Export to Excel	Yes	Yes	Restricted to 10,000 records
Printing	Yes	Yes	Restricted to 10,000 records
Personalization	Yes	Yes	
Cells with links, buttons, icons	Yes	Yes	
Fixed Columns	Yes	Yes	
Double-click	Yes	Yes	
Graphics	Yes	No	
Editable	Yes	No	

SAP ABAP on HANA

Interview Questions and Answers



Learn more in detail

Enrol on the course

[SAP ABAP on HANA Training for Beginners](#)

On Udemy.com

45Q: What are the benefits of using ALV IDA?

A: ALV IDA provides several benefits over traditional ALV grids, including:

- The ability to display data from multiple tables in a single grid, without the need for complex joins or nested SELECT statements.
- Improved performance, since ALV IDA retrieves only the data that is needed to display the grid, rather than retrieving all the data from each table and performing joins in memory.
- Flexibility, since ALV IDA can be used to display data from any combination of tables or views, regardless of the underlying data model.
- Customizability, since ALV IDA provides a wide range of options for customizing the layout and behaviour of the grid

46Q: What is a Virtual Data Model (VDM) in SAP CDS views?

A: A Virtual Data Model (VDM) is a data model that defines a logical view of data based on one or more underlying tables, views, or other data sources. It provides a unified and standardized way to define and consume data models in SAP CDS views.

47Q: What are types of CDS views in Virtual Data Model (VDM)?

A:

1. **Basic Views:** Basic Views are the simplest type of CDS view, which define a single database table or view. They provide a simple way to access the data and can be used as a building block for more complex views.
2. **Composite Views:** Composite Views combine multiple Basic Views to create a more complex view of the data. They can include joins, unions, or projections to combine the data from multiple tables or views. Composite views can also be used to provide a simplified view of complex data structures.

SAP ABAP on HANA

Interview Questions and Answers



Learn more in detail

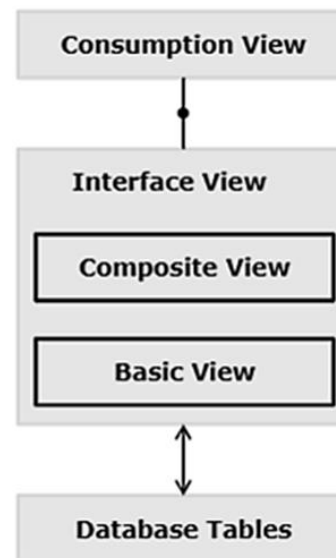
Enrol on the course

[SAP ABAP on HANA Training for Beginners](#)

On Udemy.com

- 3. Consumption Views:** Consumption Views are used to provide a business-oriented view of the data, which is optimized for reporting and analysis. They are built on top of Basic and Composite Views, and can include calculated fields, filters, and aggregations to provide a summarized view of the data. Consumption Views can be used to define key performance indicators (KPIs) and other metrics for business analysis.

These three types of CDS views can be organized into a layered architecture in the VDM to create a scalable and flexible data model that meets the needs of the business.



48Q: Which is mandatory annotation in CDS view?

A: The annotation `@AbapCatalog.sqlViewName` is mandatory

SAP ABAP on HANA

Interview Questions and Answers



Learn more in detail

Enrol on the course

[SAP ABAP on HANA Training for Beginners](#)

On Udemy.com

49Q. Difference between SAP HANA Native SQL, Classic Open SQL and Enhanced Open SQL

A:

	SAP HANA Native SQL	Classic Open SQL	Enhanced Open SQL
Element List Separator	Comma (,)	Blank ()	Comma (,)
Position of Element List	Before FROM clause	Before FROM clause	Before or after FROM clause(key word FIELDS)
Separator between table and field name	Period (.)	Tilde (~)	Tilde (~)
Schema handling	Implicit default schema (user schema) or schema name before table name	Only default schema	Only default schema
Client Handling	No implicit client handling. Client needs to be specified in WHERE-clause and join conditions.	Implicit client handling by DB interface. Explicit client via addition CLIENT SPECIFIED.	Implicit client handling by DB interface. Explicit client via addition USING CLIENT.
Delimitate number of rows	TOP ...	UP TO ... ROWS	UP TO ... ROWS
Access to CDS Views	Only SQL view visible on database	Only access to SQL view supported.	Access to CDS view possible.
CORRESPONDING FIELDS	Not supported	Supported	Supported
FOR ALL ENTRIES	Not supported	Supported	Supported

50Q: What conditions have to be fulfilled to perform an SAP HANA full-text search on column COL of database table TAB. Which of the following?

A:

- Table TAB is located in column store.
- A full text index exists for column COL of table TAB.