* CDS views exist in 2 flavours, namely, **ABAP CDS views and HANA CDS views**.
* Adobe Forms **reads the associated master and transaction data from the SAP system**. The application then displays the data in the desired and predefined form, for example, as a print form. It is also possible to enter data in forms. SAP Interactive Forms by Adobe is the successor to SAPscript and SmartForms.
* A matchcode is a means of finding data records stored in the system.
* **Structure : It is a Data type used to define a stuctured data object like Work area.**
* **Table Type:It is a Data type used to define a Internal data object**.
* **PDF** stands for **Portable Document Format.**  
  **Definition:**  
  It is a Universal file format developed by Adobe that preserves all the fonts, formatting, graphics, and color of any source document,  
  regardless of the application and platform used to create it. PDF files are compact and can be shared, viewed, navigated, and printed exactly as intended by anyone with Adobe Reader software.

**Why PDF Forms?**  
 1.It's an open standard  
 2.An Ideal document format  
 3.It provides security options like -  
  a.Digital signature validation  
  b.Disable save option  
  c.Disable select option  
  d.Disable print option.

**Benefits Adobe Provides for SAP -**  
 1.Use of PDF format ensures that the appearance of the form remains same, irrespective of the environment it is being used in.  
 2.It helps in optimization of business process by automating the creation of data for SAP systems. Thus saving time and costs (in case of interactive forms).  
 3.Integrate business processes with more users.  
 4.Overcome limitations of paper based forms like -manual data entry, error prone and easily outdated, traceability, high costs in production and storage.

* A form interface that sends the application data to the form.
* **Context - mapping parameters from interface.**  
   A form context that contains the form logic. This logic controls the dynamic formatting of the form. In the context (also known as the form context), you  
   specify which data is copied from the interface to the form.

Move corresponding-

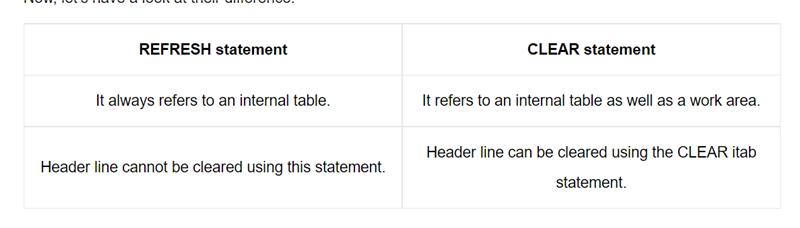
If there are components with the same name, the target table **itab2** is deleted without the addition **KEEPING TARGET LINES** and the same number of initial rows are inserted as exist in the source table **itab1**. The rows of the source table are then extracted sequentially (in the same order as in the statement [**LOOP**](javascript:call_link('abaploop_at_itab.htm'))) and the content of each row is assigned to the corresponding row in the target table in accordance with the rules for [**MOVE-CORRESPONDING *[*EXACT*]*** for structures](javascript:call_link('abapmove-corresponding_structure.htm')). Finally, the [table keys](javascript:call_link('abenitab_key.htm')) and associated [table indexes](javascript:call_link('abentable_index_glosry.htm')) are updated (if necessary) in the target table in accordance with the rules [insertions in internal tables](javascript:call_link('abapinsert_itab.htm')). The relevant exceptions are raised if uniqueness is violated.

If there are no components with the same name, no assignment is made and the target table is left unchanged.

**KEEPING TARGET LINES**

**Effect**

This addition stops the target table **itab2** from being deleted. Instead, it appends the same number of initial rows as exist in the source table **itab1**.

* 'sy-subrc' is **a return code, set by the following ABAP statements**. As a rule, if SY-SUBRC = 0, the statement was executed successfully. ASSIGN sets SY-SUBRC to 0 if the field symbol assignment was possible, otherwise to 4.
* Within the Loop command, AT NEW statement detects when the VBAP structure has a different VBELN (document number) then the previous one.  
  When a new VBELN Sales Order number is detected in the loop, the order number is printed. And a bracket is opened for the item positions of that sales order.  
    
  LOOP AT END statement also detects the different vbeln number and enables ABAP developers to do work with the previous structure in the loop.  
  In this sample ABAP program, I used the Loop At End for closing the bracket containing the list of sales order item positions.
* 
* CLEAR : It will clear only Header of the internal Table.
* Refresh : It will clear the Data in that internal table,but allocated memory will
* remain.
* Free : It will clear the data as well as allocated memory for that internal table.
* **LY\_FILTER( ) function can be used on internal table or DB tables to get all the items that match the filter condition.**
* ADBC can always be used **when access to a database using the Native SQL interface instead of the ABAP SQL interface is necessary**.
* AMDPs **allow you as an ABAP developer to write database procedures directly in ABAP**.
* ADBC can always be used **when access to a database using the Native SQL interface instead of the ABAP SQL interface is necessary**.
* A CDS view serves **to define the structure of an SQL view and represents a projection onto one or several Dictionary tables or Dictionary views**
* The data of an application is distributed across several database tables. Using ABAP CDS views, you can rearrange the table fields according to application-specific needs from the ABAP source code of your implementation.
* **CDS views can be created to read and process data at DB layer.** **Whereas AMDP can be created to process and modify data at DB layer**.
* \*CL\_SALV\_GUI\_TABLE\_IDA class has method create which returns RO\_ALV\_GUI\_TABLE\_IDA.  
  \*RO\_ALV\_GUI\_TABLE\_IDA is of type IF\_SALV\_GUI\_TABLE\_IDA.  
  \*IF\_SALV\_GUI\_TABLE\_IDA has method FULLSCREEN.  
  \*IF\_SALV\_GUI\_TABLE\_IDA has method SET\_SELECT\_OPTIONS.  
  \*&---------------------------------------------------------------------\*  
    
  \* SET\_SELECT\_OPTIONS has parameter IT\_RANGES  
  \* IT\_RANGES has an IMPORTING PARAMETER of TYPE IF\_SALV\_SERVICE\_TYPES=>YT\_NAMED\_RANGES  
  \* IF\_SALV\_SERVICE\_TYPES IS AN INTERFACE WHIUCH HAS THE TYPE YT\_NAMED\_RANGES  
    
  \*CL\_SALV\_RANGE\_TAB\_COLLECTOR is a class which has the method ADD\_RANGES\_FOR\_NAME  
  \* ADD\_RANGES\_FOR\_NAME has an Importing parameter IV\_NAME  
  \* IV\_NAME is of type string  
  \* ADD\_RANGES\_FOR\_NAME has an Importing parameter IT\_RANGES  
  \* IT\_RANGES is an Internal table (ANY TABLE) type  
    
  \*CL\_SALV\_RANGE\_TAB\_COLLECTOR is a class which has the method GET\_COLLECTED\_RANGES  
  \*GET\_COLLECTED\_RANGES has Exporting parameter with name ET\_NAMED\_RANGES  
  \*ET\_NAMED\_RANGES is of type IF\_SALV\_SERVICE\_TYPES=>YT\_NAMED\_RANGES  
  \*'Z7674\_CDSVW1'
* The operator **NEW** works in the same way as the statement [**CREATE DATA dref TYPE dtype**](javascript:call_link('abapcreate_data_existing.htm')), where **dref** stands for the result that points to the new anonymous data object. The result is a data reference variable of the [static type](javascript:call_link('abenstatic_type_glosry.htm')) **dtype**. A constructor expression of this type cannot be continued using a [component selector](javascript:call_link('abencomponent_selector_glosry.htm')).
* A class **class**.
* ALV with IDA (SAP List Viewer with Integrated Data Access) **helps tables that contain very large quantities of data to be displayed on the UI**.
* An SAP HANA system is composed of three main components: **the host, the system, and the instance**. A host is the operating environment in which the SAP HANA database runs. The host provides all the resources and services (CPU, memory, network, and operating system) that the SAP HANA database requires.
* Unlike any other databases which use to store data on the hard drive, **SAP HANA stores it in-memory**. That means when you need the data; the system can access it directly from the in-memory storage rather than calling it up from the hard drive
* Sap hana vs database server
* The fundamental difference between the two is the database technology. The SQL version is based on disk storage technology whereas the HANA is on in-memory technology. For HANA, the data is loaded on the memory of the server and as there are no physical moving parts to retrieve the data, it supports applications and functions that require a lot of processing power.
* SAP HANA (High-performance ANalytic Appliance) is a multi-model database that stores data in its memory instead of keeping it on a disk. This results in data processing that is magnitudes faster than that of disk-based data systems, allowing for advanced, real-time [analytics](https://www.ibm.com/in-en/analytics/business-analytics).
* First event -
* Initialization : triggered when the report is loaded in memory.
* At selection-screen output : triggered when the selection screen is loaded in memory before being displayed.
* At selection-screen : before leaving the selection screen.
* start-of-selection : the first event for displaying the report.
* This event keyword defines an event block whose event is triggered by the ABAP runtime environment
* when calling the executable program selection screen processing of a selection screen.
* In an executable program, all statements that are not declarations,
* and are listed before the first explicit processing block, are assigned to this event block.
* If the program does not contain an explicitly defined event block START-OF-SELECTION,
* these statements form the complete event block START-OF-SELECTION.
* If a program contains an explicitly defined event block START-OF-SELECTION,
* these statements are added to the beginning of the event block.
* If the program contains no explicitly defined event blocks,
* these statements form the entire event block START-OF-SELECTION.
* end-of-selection : after the start-of-selection is completed.
* classiscal report events.
* top-of-page : every time a new page is started in the list.
* end-of-page : every time the list data reaches the footer region of the page.
* interactive report events.
* top of page during line selection : top of page event for secondary list.
* at line-selection : evey time user dbl-clicks(F2) on the list data.
* at pF<key> : function key from F5 to F12 to perform interactive action on the list.
* at user-command
* You use Smart Forms **to create and maintain forms for mass printing in SAP systems**. Besides using the printer for standard output you can also select the Internet (by using a generated HTML output), a fax, or e-mail as the output medium. I
* **the primary index contains the key fields of the table and a pointer to the non-key fields of the table**. The system creates the primary index automatically when the table is created in the database.
* A secondary index is **a data structure that contains a subset of attributes from a table, along with an alternate key to support Query operations**.
* Receiving - typed value returned by the method - maximum one of these
* Importing - value returned by the method - many possible, all optional
* Exporting - value sent to the method - many possible, some optional
* Changing - value sent to the method , changed within the method, sent back from method - many possible some optional
* For DEFINING a method
* Returning - typed value returned to the caller- maximum one of these
* Exporting - value sent to the caller - many possible, all optional
* Importing - value sent by the caller - many possible, some optional
* Changing - value sent by the caller, changed within the method, sent back to caller - many possible some optional
* Note - only for returning/receiving must the parameter have a specific type.

SAP Smart Forms tool can be used **to print and send documents**. This tool is useful in developing forms, PDF files, e-mails and documents for the Internet. The tool provides an interface to build and maintain the layout and logic of a form.

* Forms are for data entry - one record at a time. Reports are for output - all relevant records.
* A report is **a presentation of data in an organized structure**.
* A calculation view is **a flexible information view that you can use to define more advanced slices on the data available in the SAP HANA database**. Calculation views are simple and yet powerful because they mirror the functionality found in both attribute views and analytic views, and also other analytic capabilities.