**Structures:**  Structures are basically data objects consisting of various components or fields of any data types. It differs from the tables in a way that it simulates the format of the table **and is used to hold one row of data whereas table has multiple rows of data.**

A structure is a sequence of any elementary types, reference types, or complex data types.

You use structures in ABAP programs to group work areas that logically belong together. Since the elements of a structure can have any data type, structures can have a large range of uses. For example, you can use a structure with elementary data types to display lines from a database table within a program. You can also use structures containing aggregated elements to include all of the attributes of a screen or control in a single data object.

The following terms are important when we talk about structures:

**Nested and non-nested structures**

**Flat and deep structures**

A nested structure is a structure that contains one or more other structures as components.

Flat structures contain only elementary data types with a fixed length (no internal tables, reference types, or strings).

The term deep structure can apply regardless of whether the structure is nested or not. Nested structures are flat so long as none of the above types is contained in any nesting level.

Any structure that contains at least one internal table, reference type, or string as a component (regardless of nesting) is a deep structure.

Accordingly, internal tables, references, and strings are also known as deep data types. The technical difference between deep structures and all others is as follows. When you create a deep structure, the system creates a pointer in memory that points to the real field contents or other administrative information. When you create a flat data type, the actual field contents are stored with the type in memory. Since the field contents are not stored with the field descriptions in the case of deep structures, assignments, offset and length specifications and other operations are handled differently from flat structures.

**Structures**

* Data type in DDIC, composed of Data Element, table type and tables.(can have structures inside structures too.)
* Does not contain primary key otherwise same like table.
* Reusable component and can be used in multiple programs
* Define same work area in multiple programs.

2 ways of including structures in tables.

1. Include Structure (Customized --- Z/Y)---- FLAT structures.
2. Append Structure.

|  |  |
| --- | --- |
| Include | Append |
| Can be used with custom tables only | With standard tables |
| Reusable by multiple tables | Not reusable |
| Just give field name as “.include” and data element as Structure name | Click on append structure button to create structure for a field. |