SAP S/4 HANA

Categorical Data

Categorical Data

All programming languages distinguish between various types of data with various uses, such as type data for storing or displaying values and numerical data for calculations.

how data is stored in the repository, and how the ABAP statements work with the data.

There are three categories of data types.

- Elementary types
- Complex types
- Reference types

- Categorical Data
  - Elementary types
  - Complex types
  - Reference types

Elementary

They are predefined data types.

They are single data types. They are not composed of other data types.

Elementary data types are of 2 types:

Fixed length data types - C(character) , N(numeric) , I(Integer) , P(packed number) , F(floating point) , D(date), T(time) , X(hexadecimal).

Variable length data types - String , Xstring.

Complex Type

There is no pre-defined complex data type in ABAP.

They are the combination of elementary data types.

There are 2 types of complex data types.

Structure type

Table type

Reference data type

There is no pre-defined reference data type.

It describes data objects that contain references to other objects.

They are of 2 types of reference data type.

Data reference

Object reference

Reference data type

Example : DATA lo_object TYPE REF TO zclass.

In the above syntax - DATA = keyword , lo_object = name of data object, TYPE REF TO = keyword , zclass = name of already existing class.

Type of Data Object

The Data objects are of 2 types.

Literals(unnamed data objects)

Named data objects

Literals(Unnamed Data Objects)

Literals don’t have any name that’s why they are called as unnamed data objects.

The are fully defined by their value.

There are 2 types of literals.

Numeric literals - Numeric literals have sequence of numbers. Examples - 123 , -4567 etc.

Character literals - Character literals are sequences of alphanumeric characters in single quotation marks. Examples - ‘Test 123’ , ‘SAP ABAP’ etc.

Named Data Object

Data objects that have a name are called as named data objects.

The various types of named data objects are as follows :

Variables

Constants

Text symbols

Variable

Variables are data objects whose contents can be changed.

Variables are declared using the DATA, CLASS-DATA, STATICS, PARAMETERS, SELECT-OPTIONS, and RANGES keyword.

Example - DATA lv_empid(20) TYPE n.

```
        lv_empid = 10.
        lv_empid = 20.
```

Constant

Constants are data objects whose contents can not be changed.

Constants are declared using the CONSTANTS keyword.

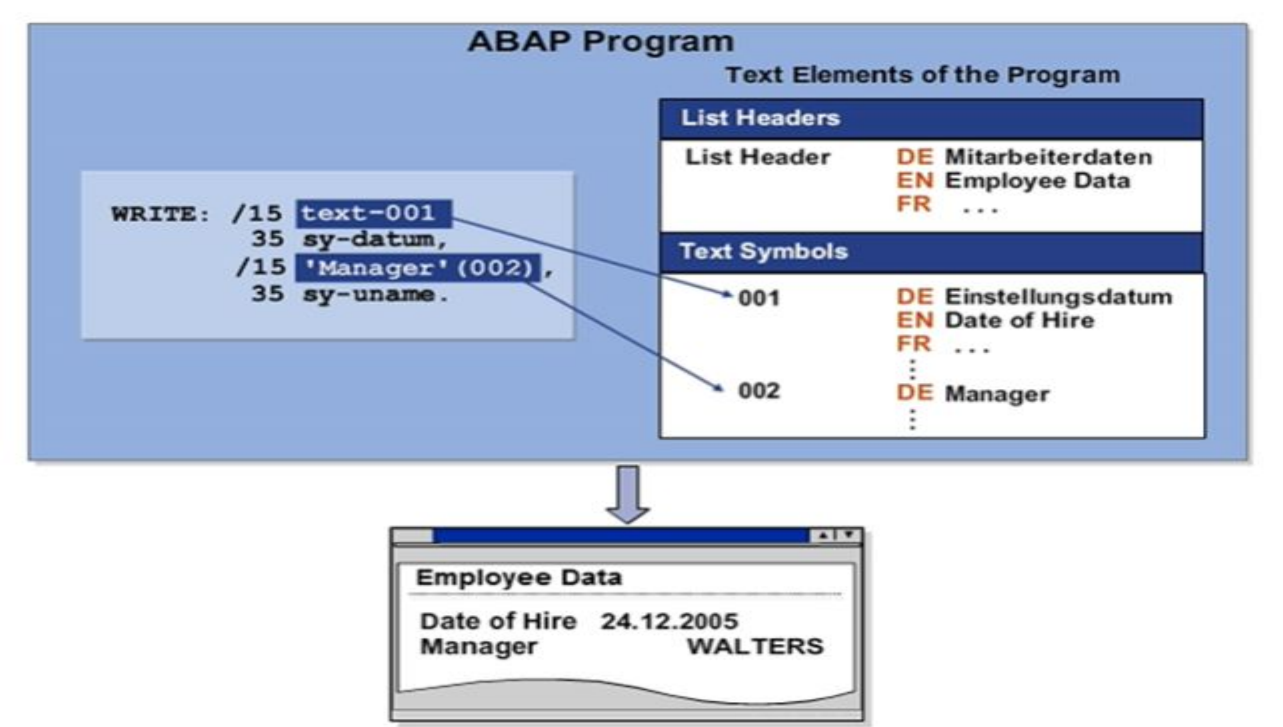
Example - CONSTANTS lc_pi TYPE P DECIMALS 3 VALUE ‘3.141’.

Text Symbol

Each text symbol is used in the particular program to which it belongs. These symbols are stored outside the source code in their own Repository object, the text pool for the program.

Text symbols can be translated into various languages and each of them is stored with a language indicator in the text pool,

Text Symbol Example



```
WRITE: /15 text-001
        35 sy-datum,
       /15 'Manager'(002),
        35 sy-uname.
```

```
Employee Data

Date of Hire   24.12.2005
Manager             WALTERS
```

Local and Global Data Type

Local data types can only be used in the program in which they are defined. To mention any local variable use lv_data variable

Global data types, in contrast, can be used throughout the entire system. To mention any global variable use gv_data variable

Global data types can be used throughout the system, which increases the system's consistency. The fact that they can be reused reduces the amount of maintenance efforts required.

Global data types can have a where-used list generated in the ABAP Dictionary. The where-used list includes repository objects that use the data type in question.

Elementary data objects with values

You are to create a simple ABAP program for basic calculation types. You must be able to enter values and the arithmetic operator. For this reason, you need to know:

How to fill elementary data objects with values

How to perform calculations in ABAP

Calculations and Arithmetic Expressions

n ABAP, you can program arithmetic expressions nested to any depth.

+ (Addition)
- (Subtraction)
* (Multiplication)
/ (Division)
** (Exponentiation)
DIV (Integral division without remainder)
MOD (Remainder after integral division)