## ABAP 编程基础

### 基本语法特点

1 不区分大小写

2语句分隔符 ‘.’

### 基本数据类型

#### 基本介绍

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **数据类型** | **默认大小** | **有效大小** | **初始值** | **说明** | **示例** |
| C | 1 | 1-65535 | SPACE | 文本字符(串) | ‘name’ |
| N | 1 | 1-65535 | ’00…0’ | 数字文本 | ‘0123’ |
| T | 6 | 6 | ‘000000’ | 时间(HHMMSS) | ‘123010’ |
| D | 8 | 8 | ‘00000000’ | 日期(YYYYMMDD) | ‘20080101’ |
| I | 4 | 4 | 0 | 整型(整数) | 99 |
| F | 8 | 8 | 0 | 浮点数 | 5E+04 |
| P | 8 | 1-16 | 0 | 压缩数 | 12.91 |
| X | 1 | 1-65535X | ‘0’ | 十六进制数 | ‘3A’ |
| String | 1 | 无限定 | SPACE | 字符串(长度可峦) | ‘A String’ |
| XString | 1 | 无限定 | SPACE | 十六进制字符串 | ‘A3FF’ |

#### 细节注意

- N 为数字文本字符类型，如果定义的位数为4位，但是数据只有两位时（外部显示），会补上前置零，例如：0020（内部显示）

-P类型，默认为8字节，最大允许16字节，其数据对象占据内存字节数和数值范围取定义时指定的整个数据大小和小数点后位数，如果不指定小数位，则将视为I类型。其有效数字位大小可以是从1~31位数字（小数点与正负号占用一个位置，半个字节），小数点后最多允许14个数字。

EX:

DATA: VP\_A(16) TYPE P DECIMALS 14 VALUE

‘-12345678901234567.12345678901234’.

- D 和 T 类型在显示时，字符间中间不带任何格式符，

例如，日期：20140224（内部显示），时间：120535（内部显示）

### 变量声明方式

变量声明关键字：DATA

类型说明关键字：TYPE、LIKE

|  |
| --- |
|  |
| 类型声明： |
| TYPES: BEGIN OF TY\_STU,          NAME TYPE C LENGTH 6,          ID(8)  TYPE N,          BDATE TYPE D,        END OF TY\_STU. |
| 变量声明： |
| DATA: NAME(5) TYPE C VALUE 'NYYDD'， "设置默认值时必须为常量  AGE TYPE I. |
| 结构体声明： |
| DATA: BEGIN OF STRUC.          INCLUDE TYPE TY\_STU.          INCLUDE STRUCTURE SFLIGHT.        DATA: END OF STRUC. |
| 内表声明1： |
| DATA: ITAB LIKE TABLE OF STRUC [WITH HEADER LINE]. |
| 内表声明2： |
| DATA: BEGIN OF ITAB OCCURS 0,        ...       END OF ITAB. |
| 常量声明： |
| CONSTANTS: DATE LIKE SY-DATUM VALUE '19960926'. |

### 系统预置变量

#### 常用的系统变量

|  |  |
| --- | --- |
| 变量名 | 说明 |
| SY-DATUM | 当前系统日期 |
| SY-UZEIT | 当前系统时间 |
| SY-UNAME | 当前登入的账号名 |
| SY-REPID | 当前程序名称 |
| SY-DYNNR | 当前Screen号 |
| SY-MANDT | 当前系统（集团）号 |
| SY-SUBRC | 系统执行某指令后,表示执行成功与否的变量 |
| SY-INDEX | 当前LOOP循环过的次数 |
| SY-TABIX | 当前处理的是internal table 的第几条 |
| SY-TMAXL | Internal table的总条目数 |
| SY-DBLNT | 数据库操作时被处理过的记录的数量 |
| SY-UCOMM | PAI所出发的功能代码 |
| SY-LOOPC | 当前表格控件（Table Control）在屏幕中的总行数 |
| SY-ULINE | 水平线 |
| SY-VLINE | 垂直线 |

### 字符串的处理

#### 常用字符串处理

|  |  |
| --- | --- |
| 操作 | 语法与说明 |
| 字符串连接 | CONCATENATE s1 s2 […sn] | LINES OF ITAB INTO sd [SEPARATED BY sep] [RESPECTING BLANKS]  1 只能连接C、D、N、T、STRING类型；  2 默认C、D、N、T类型前导空格会保留，尾部空格会被去掉（RESPECTING BLANKS 选项可保留），String类型所有空格都会保留；  3 SEPARATED BY sep 选项为设置连接时插入分隔符 |
| 字符串去空格 | **CONDENSE <c> [NO-GAPS]**  **1 字符串中间的多个连续的空格使用一个空格替换；**  **2 如果是C类型还会去掉字符串前面的空格，如果是String类型，则后面空格也会被去掉；**  **3 NO-GAPS选项：字符串中间的所有空格都也都会去除**  **4空格去掉后会左对齐** |
| 字符串拆分 | SPLIT SD AT sep INTO s1 s2 [...sn] | TABLE itab  1必须指定足够目标字段；  2 当分隔符位于字符串首时，该分隔符左边会拆分出一个空的字符串；当分隔符位于字符串末尾时，该分隔符右边不会拆分出空字符串；  3 若SD为T、D类型，则把数据都存到第一个子串s1中（T、D类型字符间中间不带任何格式符） |
| 字符串替换 | REPLACE |
| 字符串截断（移除） | SHIFT |
| ... |  |

### 基本语法结构

|  |
| --- |
| **IF <> .**  **…**  **ELSEIF <>.**  **…**  **ELSE.**  **…**  **ENDIF.** |
| **CASE <>.**  **WHEN <>.**  **…**  **WHEN<>.**  **…**  **OTHER.**  **…**  **ENDCASE.** |
| DO times TIMES.  …  ENDDO. |
| WHILE <>.  …  ENDWHILE. |
|  |

### 内表基本操作

插入数据：

INSERT [LINE | INITIAL LINE ] INTO TABLE ITAB [INDEX N].  
INSERT  LINES OF ITAB [FROM N1] [TO N2] INTO [TABLE] ITAB [INDEX IDX]  
  
APPEND [LINE | INITIAL LINE ] INTO TABLE ITAB  
APPEND INSERT LINES OF ITAB [FROM N1] [TO N2] INTO [TABLE] ITAB  
COLLECT LINE INTO ITAB.  
读取数据：  
READ TABLE ITAB [INTO WA] INDEX IDX.  
READ TABLE ITAB FROM KEY [INTO WA].  
READ TABLE ITAB WITH KEY K1 = F1 ... KN = FN [INTO WA].  
更改数据：  
MODIFY ITAB [ FROM WA | INDEX IDX] [TRANSPORTING F1 ... FN] [WHERE COND]  
删除数据：  
DELETE ITAB [INDEX IDX]  
DELETE TABLE ITAB FROM WA  
DELETE TABLE ITAB WITH KEY K1 = F1 ...  
DELETE ITAB [FROM N1] [ TO N2] [WHERE COND]  
遍历：  
LOOP AT ITAB [INTO WA | ASSIGNING <FS> | TRANSPORTING NO FIELDS] [FROM N1] [TO N2] [WHERE COND]  
排序：  
SORT ITAB [ASCENDING|DESCENDING] BY FIELD1 ...  
  
控制级操作AT LEVEL/ENDAT:  
AT FIRST/ENDAT  
AT LAST/ENDAT  
AT NEW F/ENDAT  
AT END OF F/ENDAT  
  
SUM.  
清除：  
CLEAR/REFRESH/FREE

### 事件块

#### 报表事件块

INITIALIZATION. "初始化事件

START-OF-SELECTION. "选择开始事件

END-OF-SELECTION. "选择结束事件

#### 选择屏幕事件块

AT SELECTION-SCREEN OUTPUT. "选择屏幕PBO事件，显示选择屏幕之前触发

AT SELECTION-SCREEN. "选择屏幕PAI事件，用户在选择屏幕中执行某些功能后被触发

#### **逻辑数据库事件块**

GET node … . "选择逻辑数据库中当前级别数据中的数据  
GET node LATE. "选择逻辑数据库中上一级别数据源中的数据

#### 列表事件块

TOP-OF-PAGE." 页眉控制，在每个新页面开始时触发  
END-OF-PAGE." 页脚控制，在当前页面结束时触发  
AT LINE-SELECTION." 行选择控制，用户在选择某列表行时触发  
AT USER-COMMAND." 用户交互控制，用户选择某个GUI功能时触发

### 常用语句

#### 获取当前程序运行的时间

GET RUN TIME FIELD time（TYPE I）

## 数据库操作

### 内连接INNER JOIN 与左外连接 LEFT OUT JOIN的区别

INNER JOIN

基本规则：主表与结合表都必须满足ON中所指定的条件

PS: 有点像交集（根据连接条件）

LEFT OUT JOIN

基本规则：

对于主数据表中的数据即使在数据表中结合的条件字段不存在，也要将该行取出，结合表中不存在的字段保持空白。

### FOR ALL ENTRIES 的作用与用法

作用：从数据表中选取符合已存在内表中所有满足条件的数据值

SELECT field1 ... FROM DBTAB INTO ITAB1

FOR ALL ENTRIES IN ITAB2 WHEARE field1 = ITAB2-field1.

（PS:LIKE:

SELECT field1 ... FROM DBTAB INTO ITAB1 WHEARE field1 IN (ITAB2-field1).）

### 动态指定数据表

DATA: dbname TYPE STRING VALUE 'TB'. (变量内容（数据库名）必须大写)

...

SELECT \* FROM (DBNAME) INTO TABLE ITAB.

### 指定数据区域（集团）

一帮情况下，系统会自动进行集团处理，

对于集团相关的表，不能在OPEN SQL 语句下指定MANDT字段（出现语法或运行时错误）

也不能设置当前集团外的其他值（若出现，系统会自动替换并使用当前集团）

CLIENT SPECIFIED 取消系统自动处理

SELECT|INSERT... DBTAB CLIENT SPECIFIED(必须紧跟在表之后) ... WHERE MANDT ...

### 限定选择行数

SELECT ... DBTAB UP TO n ROWS.

### 光标

DATA: cur TYPE CURSOR.

OPEN CURSOR [WITH HOLD] cur FOR SELECT ...

FETCH NEXT CURSOR cur INTO <target>. (没有读取任何行 SY-SUBRC = 4 ELSE 0)

CLOSE CURSOR. (没有 WITH HOLD时 数据库提交系统会自动关闭)

### 更新数据：INSERT\UPDATA\MODIFY\DELETE

SY-SUBRC = 0 操作成功 SY-DBCNT 实际操作的数据行数

根据SQL语句效率原则，更新语句因尽可能整体更新多行数据

#### INSERT

INSERT INTO DBTAB VALUES WA.

INSERT INTO DBTAB FROM WA.

INSERT DBTAB FROM TABLE ITAB [ACCEPTING DUPLICATE KEYS].

ACCEPTING DUPLICATE KEYS: 如果出现关键字相同的条目，系统将Sy-SUBRC返回4 并跳过该条目，并继续更新其他条目

#### UPDATE

UPDATE DBTAB SET f1 = v1... WHERE <FIX\_KEY>|<condition>

UPDATE DBTAB FROM TABLE ITAB.

#### MODIFY

MODIFY = INSERT(不存在时) + UPDATE

#### DELETE

DELETE FROM DBTAB WHERE CONDITION.

DELETE DBTAB FROM WA|ITAB

### 数据的一致性

#### SAP LUW 与 DB LUW

-- 两个数据一致状态种的时间间隔 （更新了一个数据，另一个数据必须被更新）

-- 事务

#### SAP LOCKED 与 DB LOCKED

-- 多任务系统 同一时间不同的进程的操作