

第七章 表控制 Table Control 设计

表控制 (Table Control) 是 SAP 应用中最重要对象之一, 业务单据的输入、基础数据的表格输入等都是使用 Table Control 控件, 本章主要介绍:

- (1) 在屏幕设计中通过向导来制作表控制对象, 由系统生成程序;
- (2) 非向导设计 Table Control 控件;
- (3) 通过系统生成数据表维护程序, 并在程序中调用;
- (4) 表格设计技巧;
- (5) 通过表格维护变更内表数据, 并将内表数据提交数据库。

7. 1 使用向导制作 Table Control

本节先建立一个内表, 通过向导建立一个表格控件使用该内表。

7. 1. 1 建立程序

先建立程序, 定义内表, 屏幕有退出按钮, 代码如下。
逻辑程序:

*PBO 显示屏幕前的处理

PROCESS BEFORE OUTPUT.

MODULE STATUS_0100.

*PAI 用户输入后的处理

PROCESS AFTER INPUT.

MODULE USER_COMMAND_0100.

主程序:

REPORT YTEST020.

*功能码返回值

DATA:OK_CODE TYPE SY-UCOMM,
SAVE_OK LIKE OK_CODE.

*定义内表, 注意, 两种写法一样

*DATA SP1 LIKE SPFLI OCCURS 0 WITH HEADER LINE.

DATA SP1 LIKE TABLE OF SPFLI WITH HEADER LINE.

*增加内表数据

SELECT * INTO CORRESPONDING FIELDS OF TABLE SP1 FROM SPFLI.

*测试输出数据

*LOOP AT SP1.

* WRITE: / SP1-CARRID,SP1-CITYFROM.

*ENDLOOP.

*直接调用窗口

CALL SCREEN 100.

*用户交互

MODULE USER_COMMAND_0100 INPUT.

SAVE_OK = OK_CODE.

CLEAR OK_CODE.

*退出按钮时退出程序

CASE SAVE_OK.

WHEN 'EXIT'.

LEAVE PROGRAM.

ENDCASE.

ENDMODULE.

*定义状态条、包括菜单、工具条按钮、系统按钮等

MODULE STATUS_0100 OUTPUT.

SET PF-STATUS 'STATUS1'.

ENDMODULE.

7.1.2 使用向导建立 Table Control 控件

启动屏幕编辑器,选“表控制”按钮()，在屏幕上建立该对象后,系统启动导航界面,如图 7-1 所示。

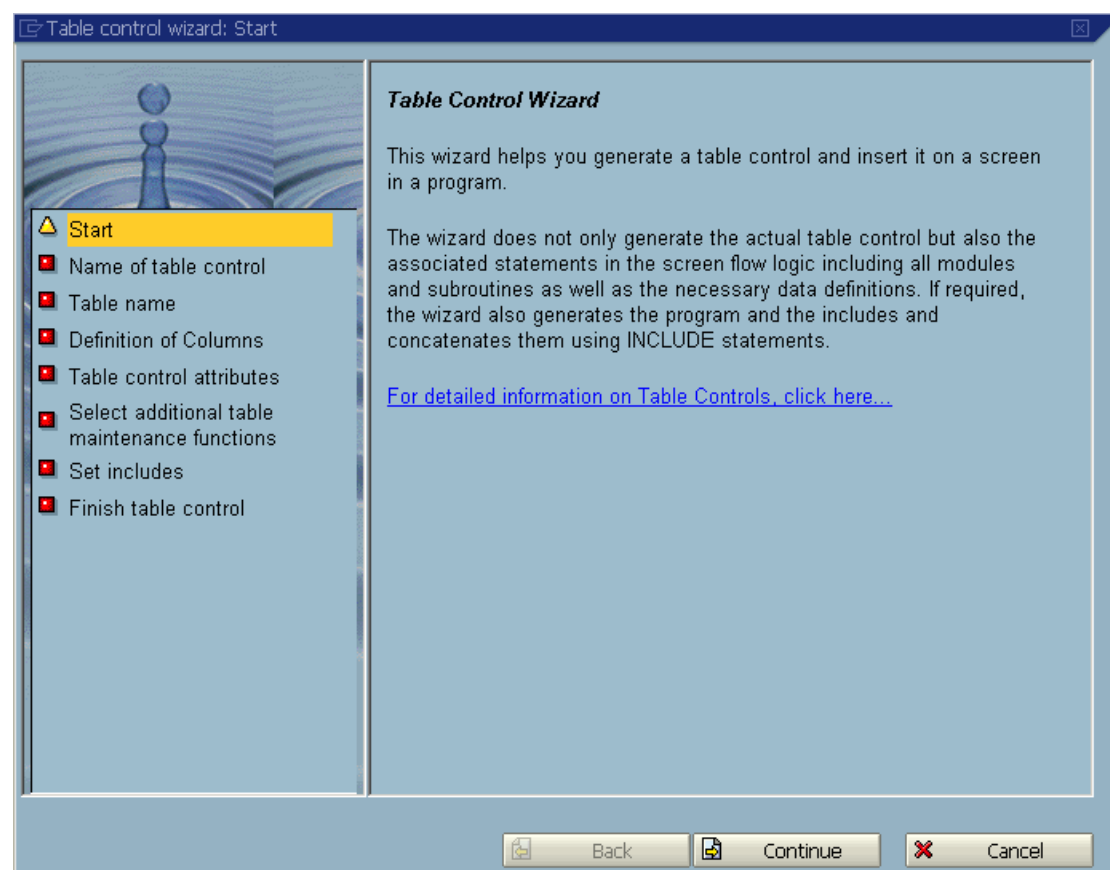


图 7-1

输入 Table Control 的名称，继续，如图 7-2 所示。

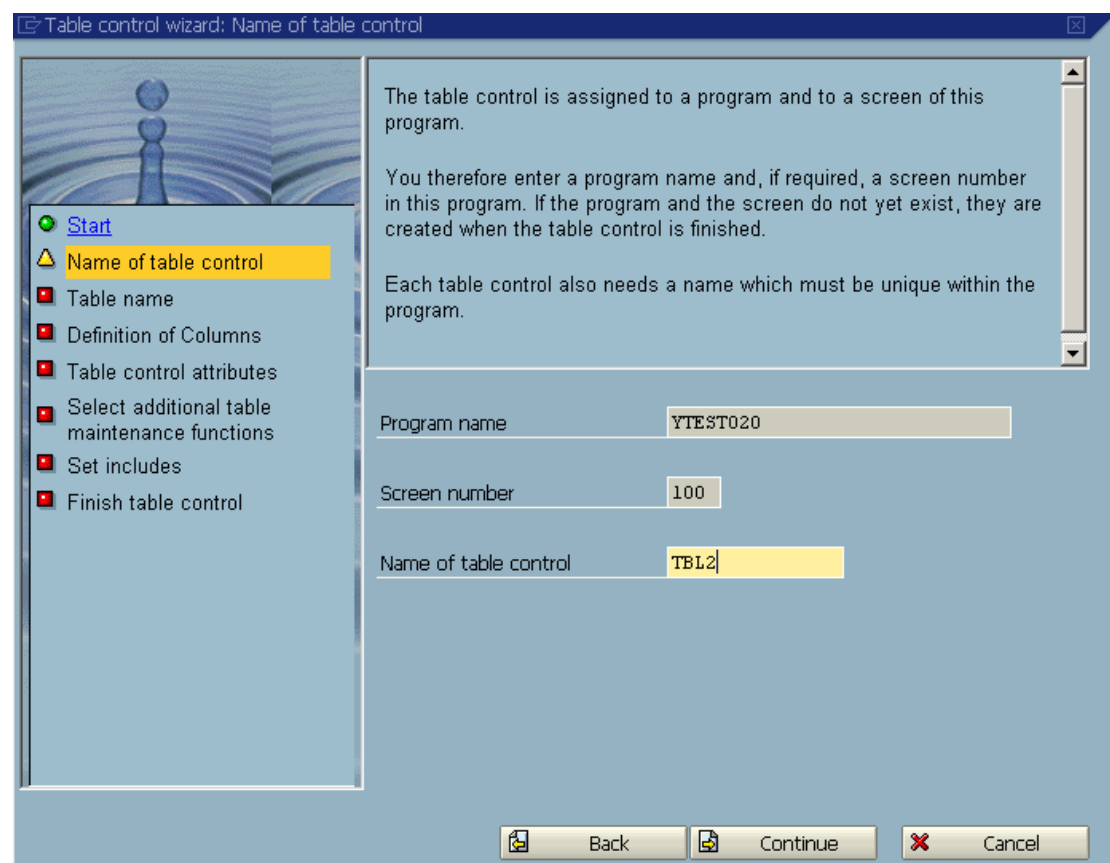


图 7-2

选择从内表建立，输入内表名称 SP1（在主程序中已建立该内表），如图 7-3 所示。

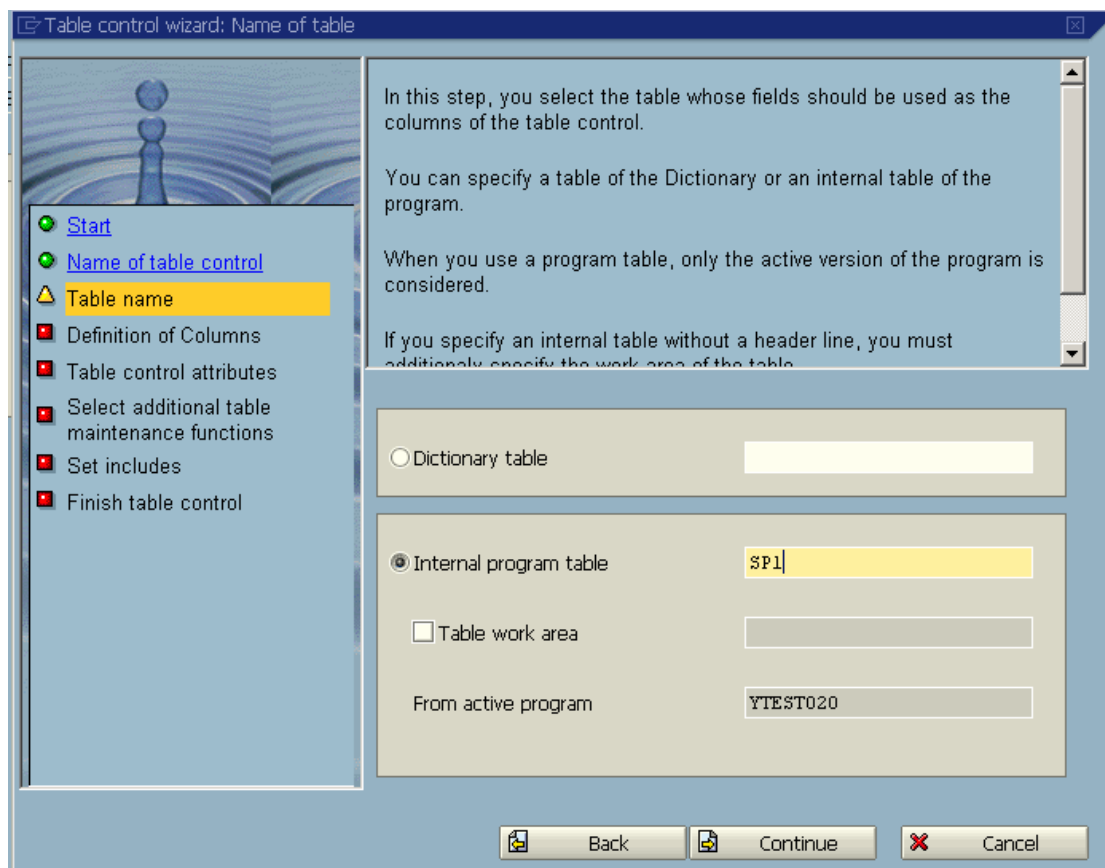


图 7-3

如果程序没有激活，会有如图 7-4 所示提示，再打开一个 SESSION 激活程序。

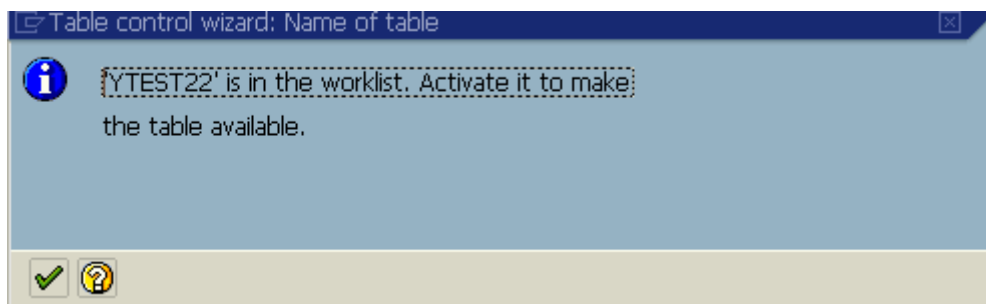


图 7-4

选择表字段，继续，如图 7-5 所示。

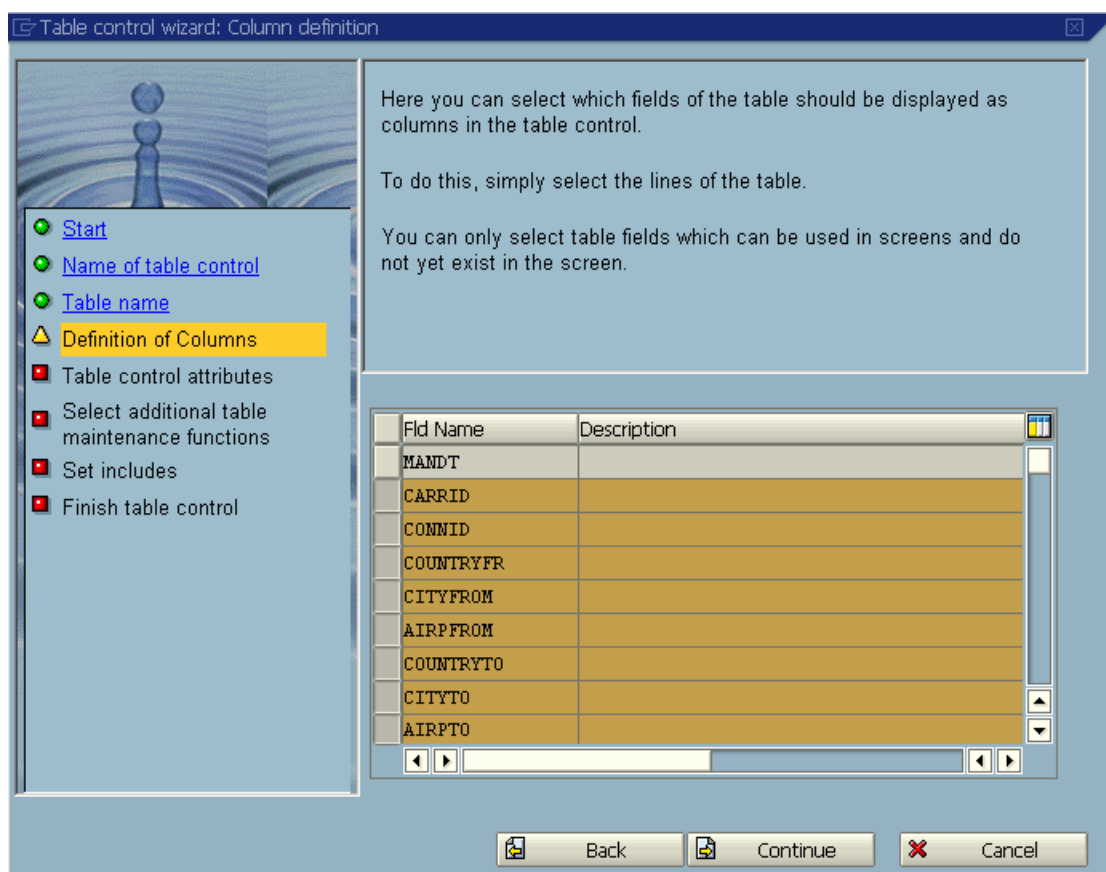


图 7-5

选择表格控件能输入（Input control）、有表头（With column headers）、单选纪录(Single), 如图 7-6 所示。

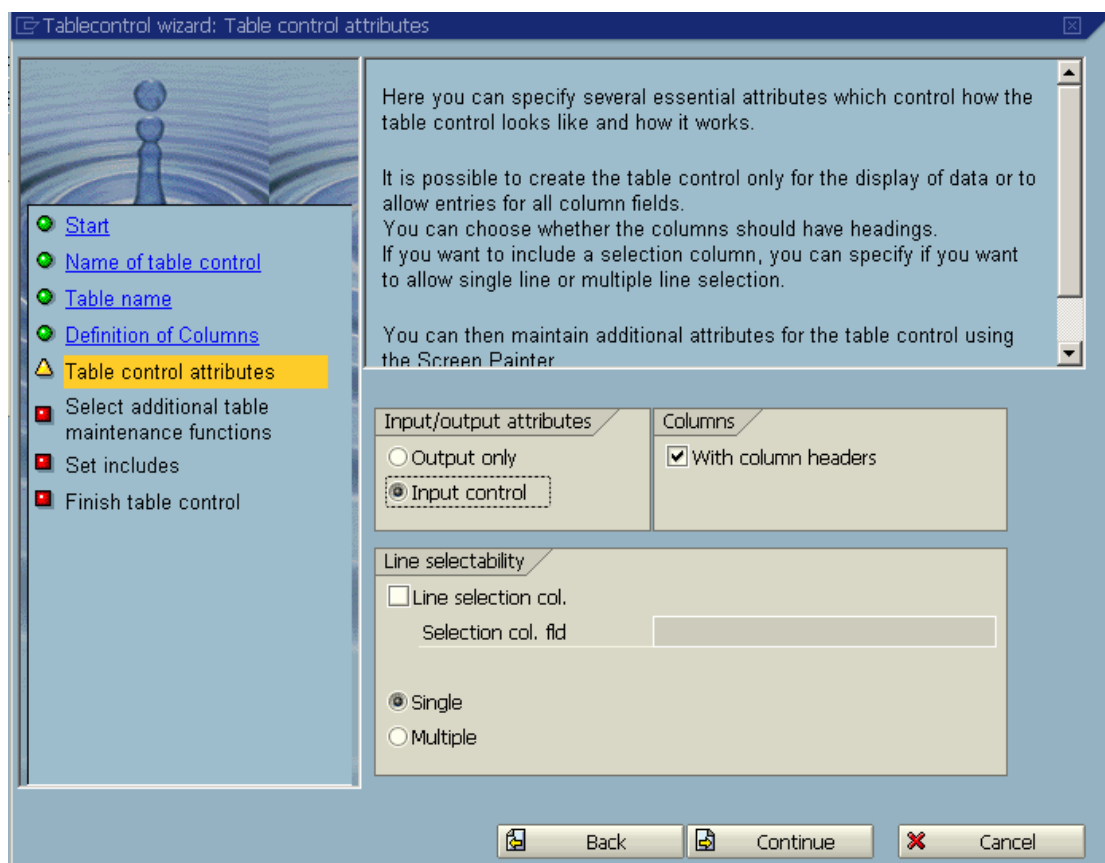


图 7-6

设定属性无滚动条，继续，如图 7-7 所示。

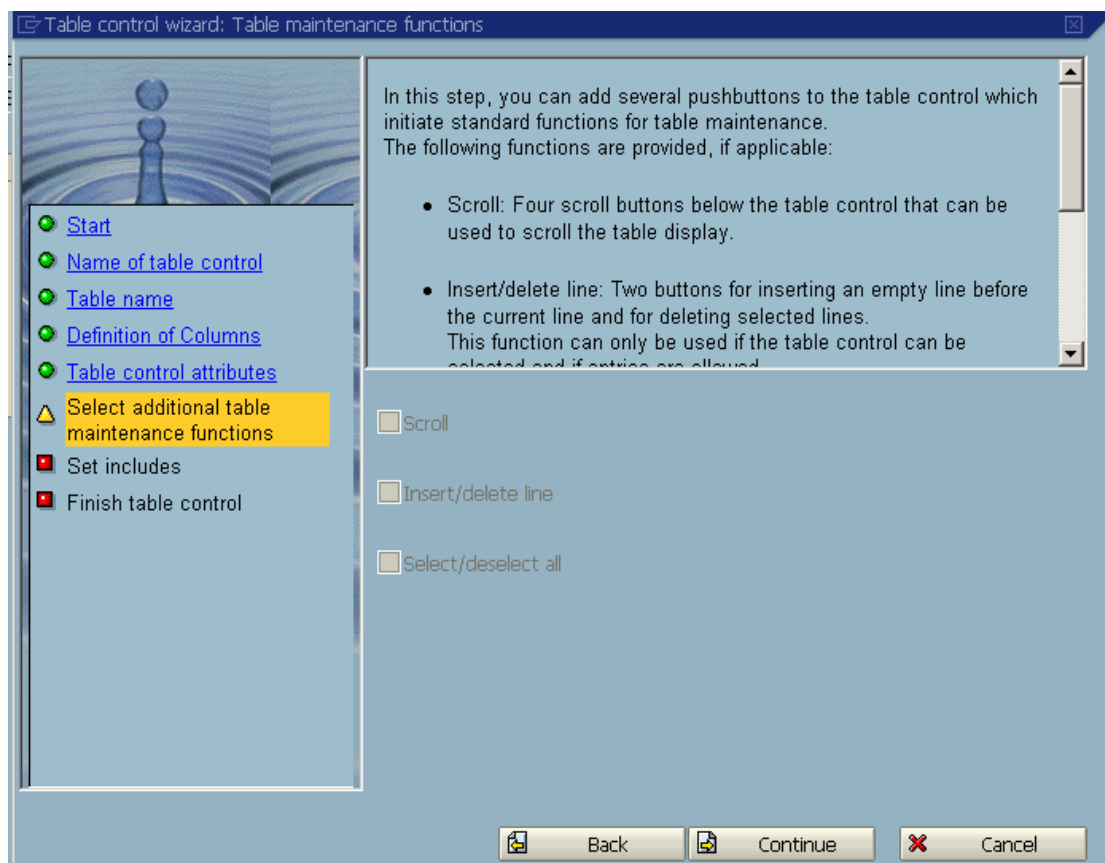


图 7-7

维护生成程序的名称，使用默认的名称，如图 7-8 所示。

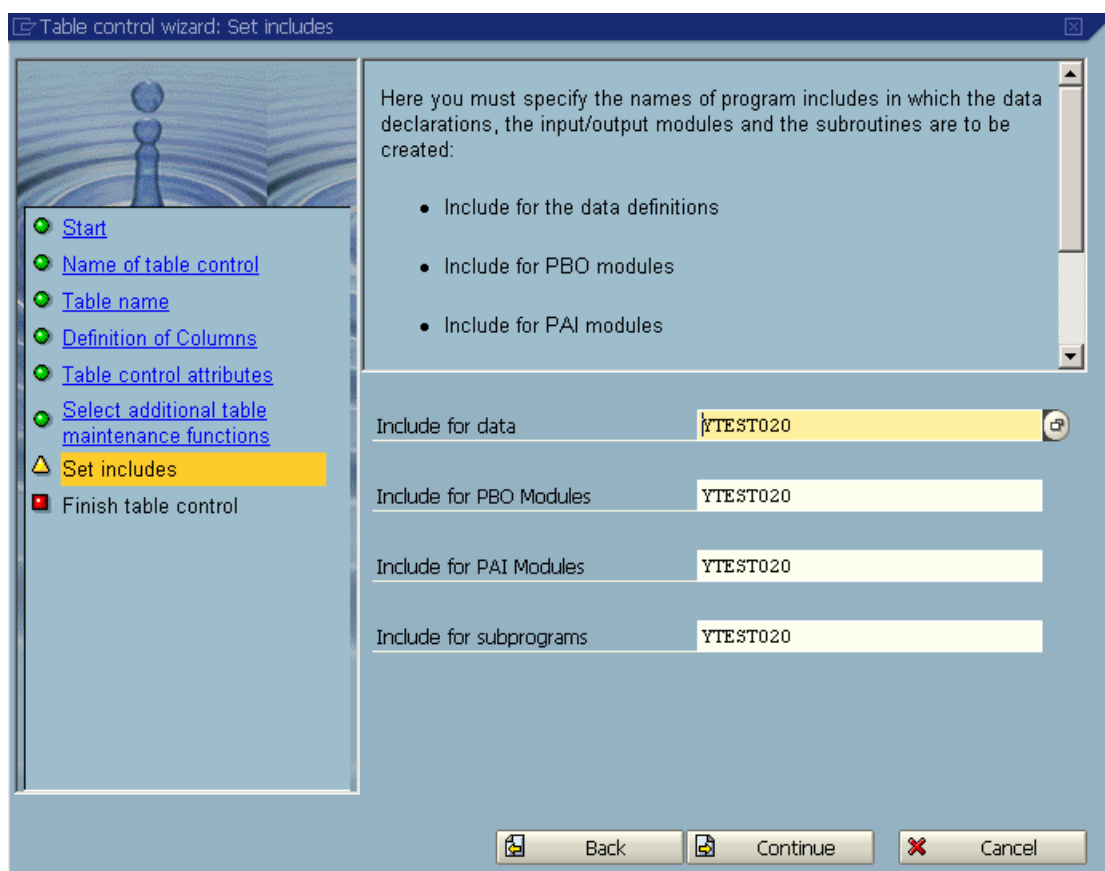


图 7-8

单击“完全”按钮，完成设计，如图 7-9 所示。

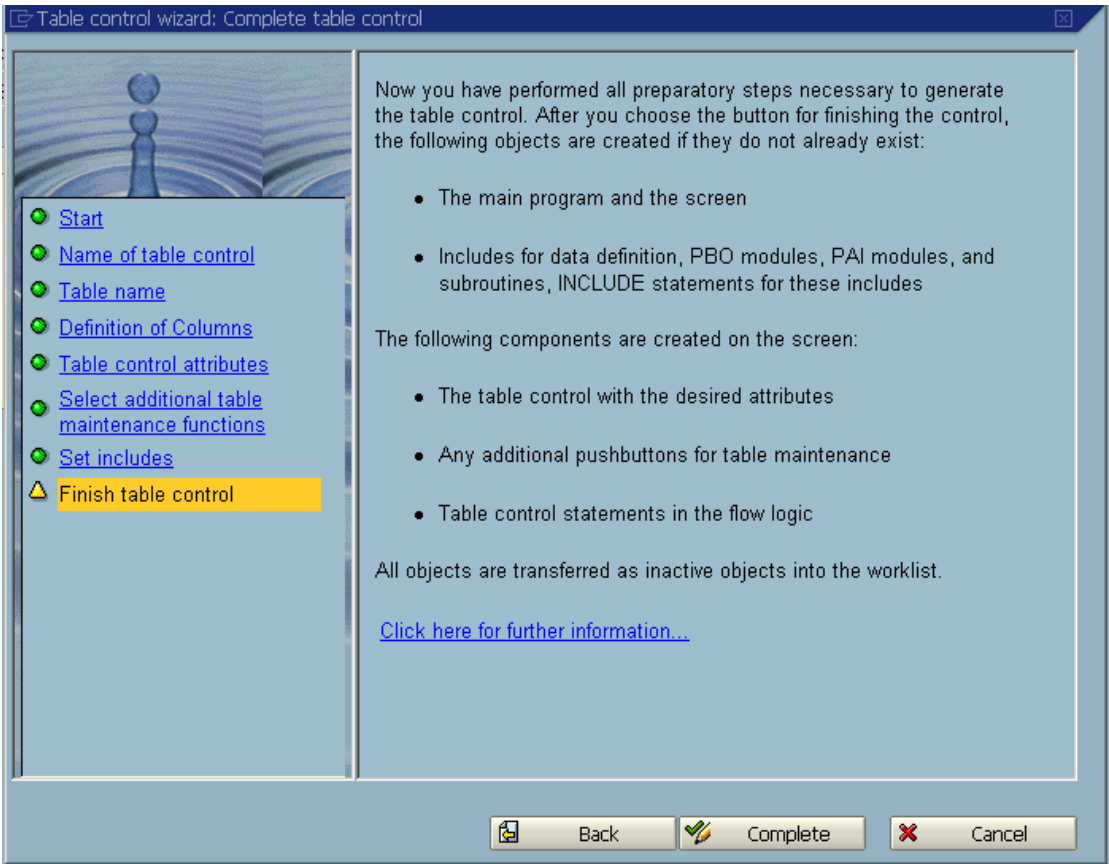
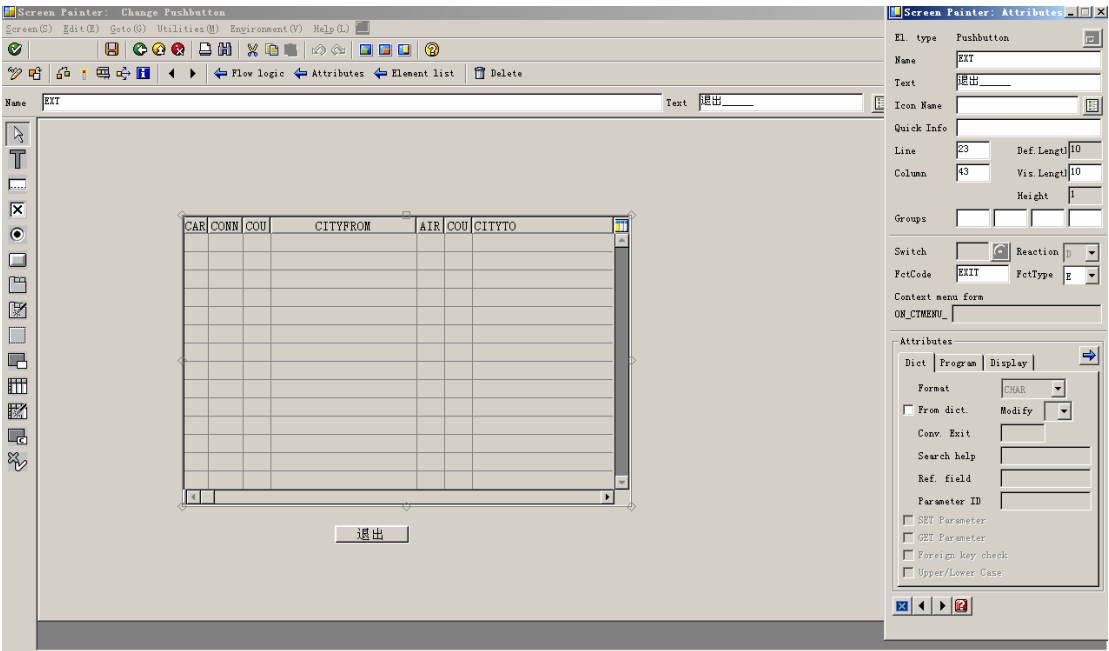


图 7-9

完成后设计界面如图 7-10 所示。



选择主程序并执行,系统输出如图 7-11 所示。

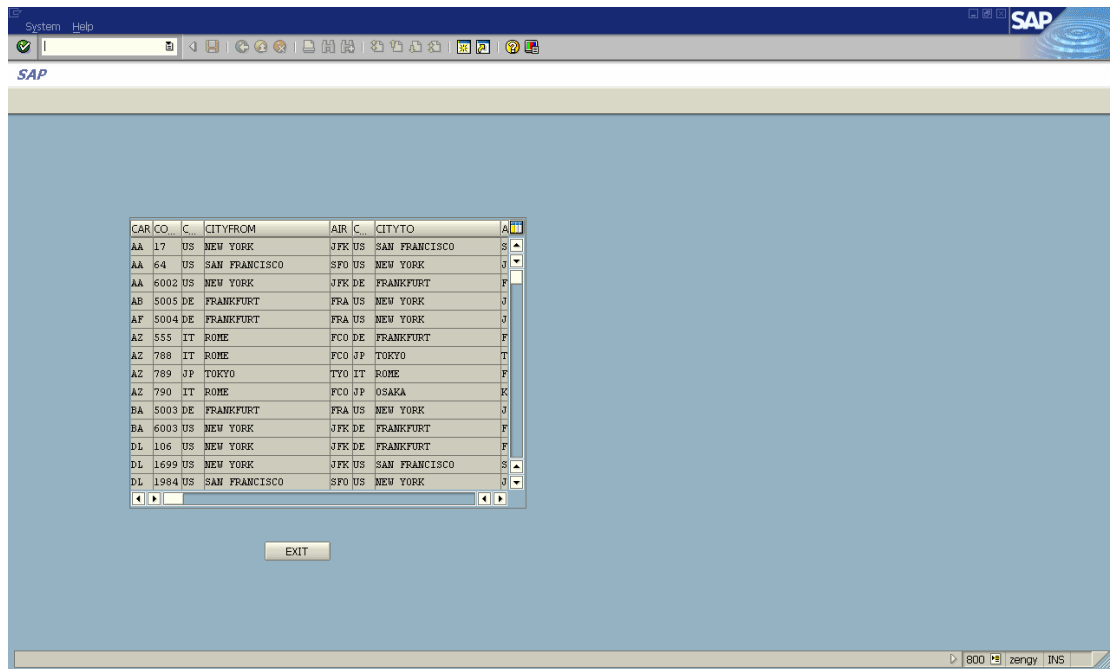


图 7-11

向导生成的 Table Control 对象已经能够正常地游览数据。

分析生成代码：

逻辑程序：

*PBO 显示屏幕前的处理

PROCESS BEFORE OUTPUT.

*PBO FLOW LOGIC FOR TABLECONTROL “TBL2”

*MODULE TBL2_CHANGE_TC_ATTR.

*MODULE TBL2_CHANGE_COL_ATTR.

*循环取内表逐行写入表控制当前行

LOOP AT SP1

WITH CONTROL TBL2

CURSOR TBL2-CURRENT_LINE.

*MODULE TBL2_CHANGE_FIELD_ATTR

ENDLOOP.

MODULE STATUS_0100.

*PAI 用户输入后的处理

PROCESS AFTER INPUT.

*PAI FLOW LOGIC FOR TABLECONTROL “TBL2”.

*当有数据改变时更新内表

PROCESS AFTER INPUT.

LOOP AT SP1.

CHAIN.

FIELD SP1-CARRID.

FIELD SP1-CONNID.

FIELD SP1-COUNTRYFR.

```

FIELD SP1-CITYFROM.
FIELD SP1-AIRPFROM.
FIELD SP1-COUNTRYTO.
FIELD SP1-CITYTO.
FIELD SP1-AIRPTO.
FIELD SP1-FLTIME.
FIELD SP1-DEPTIME.
FIELD SP1-ARRTIME.
FIELD SP1-DISTANCE.
FIELD SP1-DISTID.
FIELD SP1-FLTYPE.
FIELD SP1-PERIOD.
MODULE TBL2_MODIFY ON CHAIN-REQUEST.
ENDCHAIN.
ENDLOOP.
*MODULE TBL2_CHANGE_TC_ATTR.
*MODULE TBL2_CHANGE_COL_ATTR.

MODULE USER_COMMAND_0100.

主程序:
REPORT YTEST020.
*功能码返回值
DATA:OK_CODE TYPE SY-UCOMM,
      SAVE_OK LIKE OK_CODE.

*定义内表,注意,两种写法一样
*DATA SP1 LIKE SPFLI OCCURS 0 WITH HEADER LINE.
DATA SP1 LIKE TABLE OF SPFLI WITH HEADER LINE.
*增加内表数据
SELECT * INTO CORRESPONDING FIELDS OF TABLE SP1 FROM SPFLI.

*测试输出数据
*LOOP AT SP1.
*   WRITE: / SP1-CARRID,SP1-CITYFROM.
*ENDLOOP.

*直接调用窗口
CALL SCREEN 100.

*用户交互
MODULE USER_COMMAND_0100 INPUT.
      SAVE_OK = OK_CODE.
      CLEAR OK_CODE.

```

***退出时退出程序**

```
CASE SAVE_OK.
    WHEN 'EXIT'.
        LEAVE PROGRAM.
    when others.
ENDCASE.
ENDMODULE.                                "USER_COMMAND_0100 INPUT
```

***定义状态条, 包括菜单、工具条按钮、系统按钮等**

```
MODULE STATUS_0100 OUTPUT.
    SET PF-STATUS 'STATUS1'.
ENDMODULE.                                "STATUS_0100 OUTPUT
```

***以下是自动产生的程序**

***定义表控制对象**


```
CONTROLS: TBL2 TYPE TABLEVIEW USING SCREEN 0100.
```

```
MODULE TBL2_CHANGE_TC_ATTR OUTPUT.
    DESCRIBE TABLE SP1 LINES TBL2-lines.
ENDMODULE.
```

***当数据改变时更新内表**

```
MODULE TBL2_MODIFY INPUT.
    MODIFY SP1
    INDEX TBL2-CURRENT_LINE.
ENDMODULE.
```

7. 2 手工制作 Table Control

先建立一个有退出按钮的程序, 进入屏幕编辑器, 选中控制对象() , 并在屏幕上创建, 如图 7-12 所示。

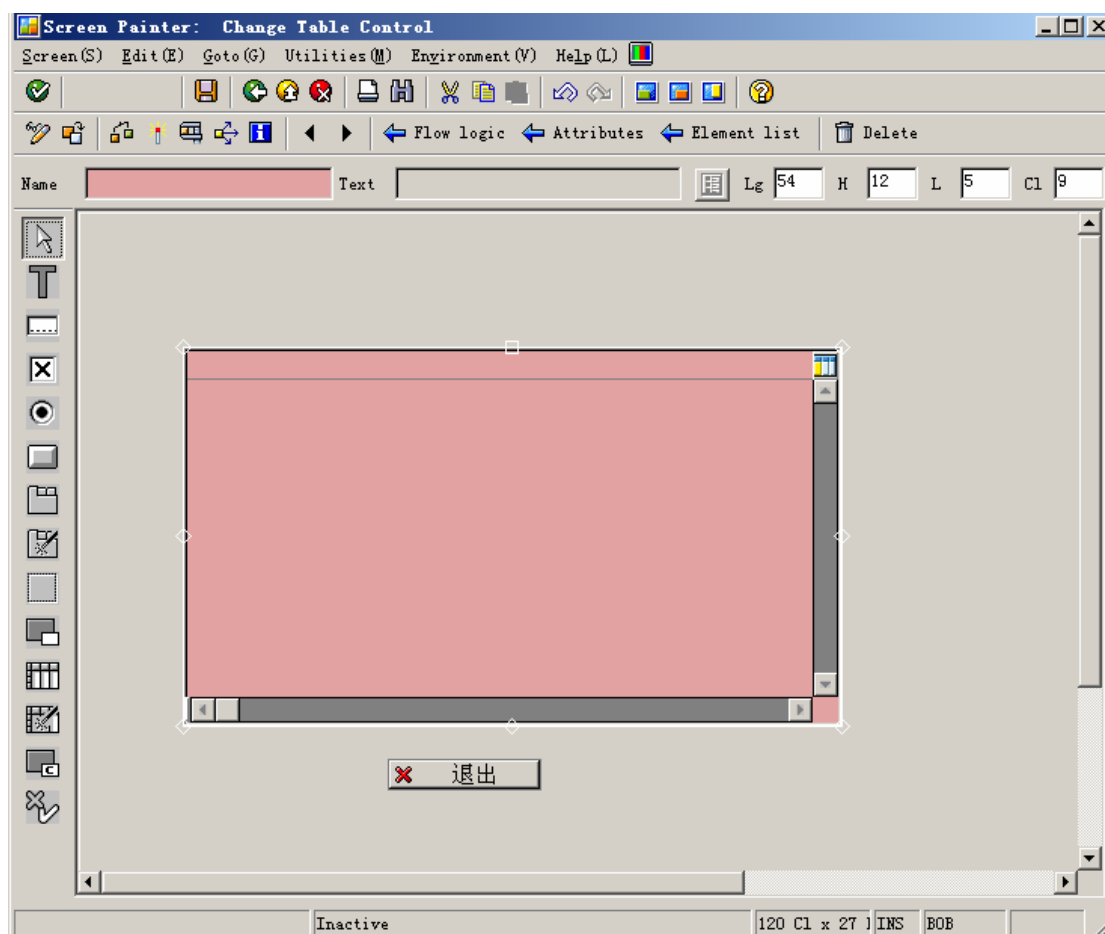



图 7-12

建立的对象显示粉红色，表示属性未设置正确。输入名称属性后，表头会转为灰色。表格的创建与常规编程软件不同，需要选中一个输入输出字段，然后在表身创建，如图 7-13 所示。

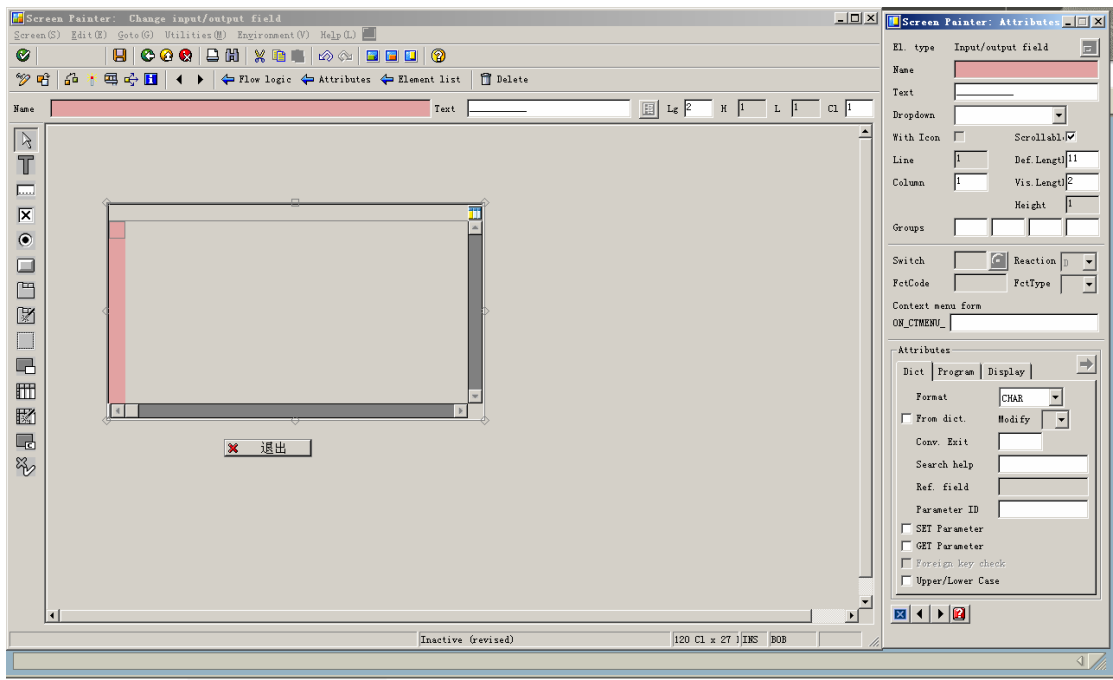


图 7-13

输入名称后整个表格都转为灰色（表示已设置正确）。调整输入字段的属性，如图 7-14 所示。

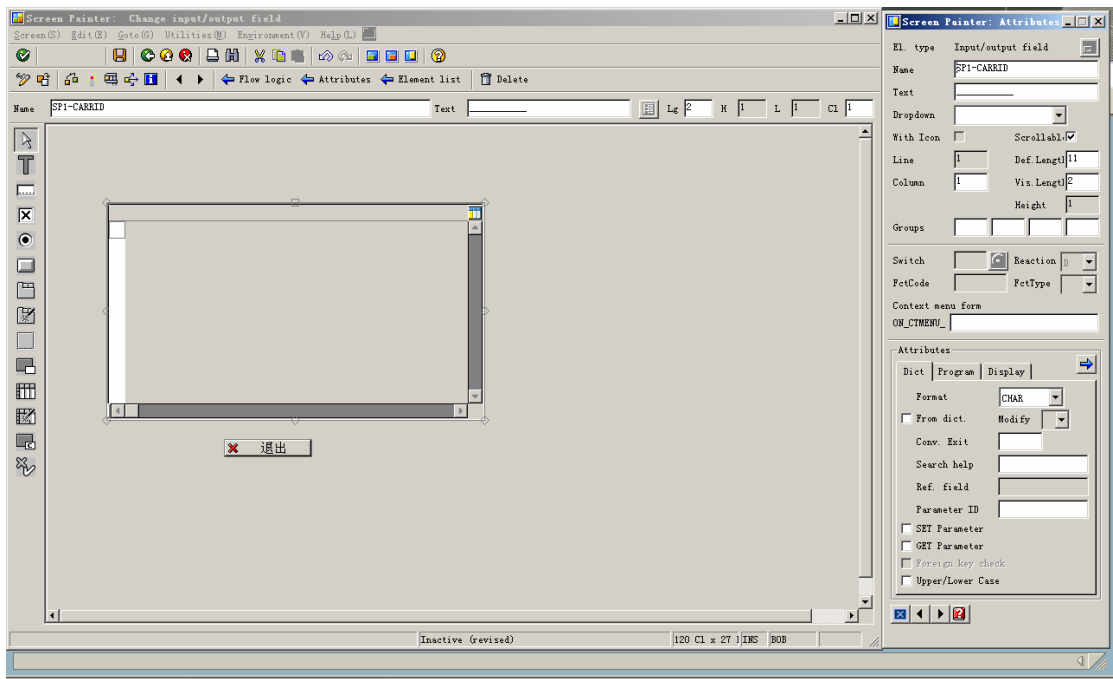


图 7-14

手工调整字段名称和内表对应的名称一致后，第一个字段建立完成，然后为该字段加上表头，制作方式和输入字段一样，选中一个文本字段，创建在该字段上方的表头，如图 7-15 所示。

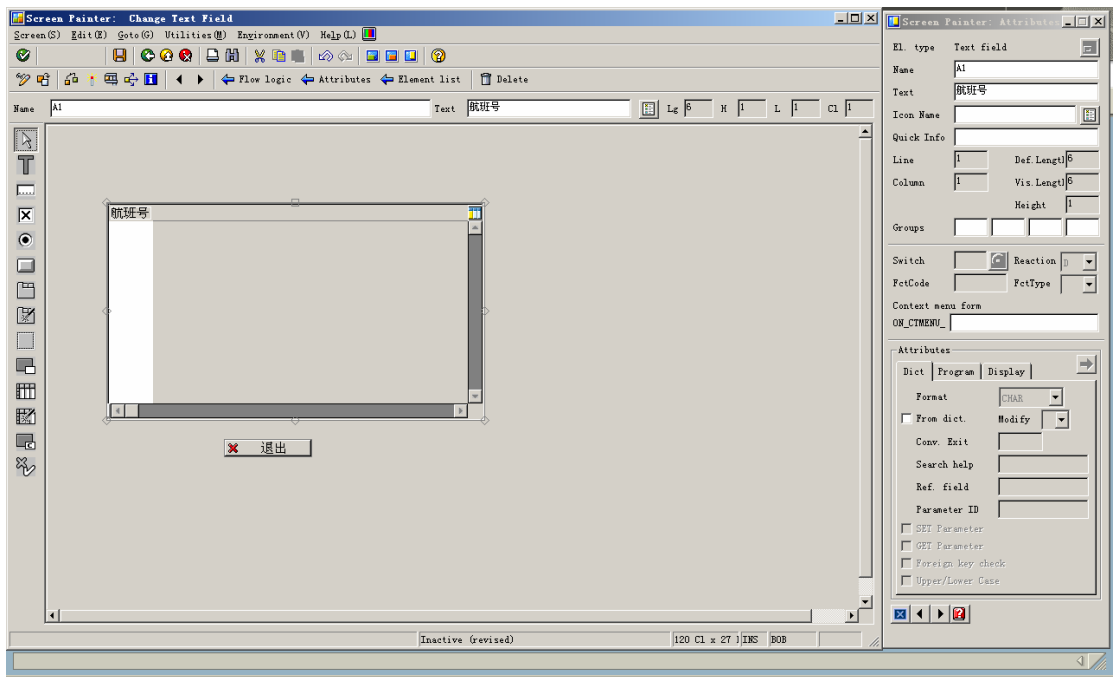


图 7-15

接着创建航班号、起飞城市等信息，并添加垂直水平分格符，完成后如图 7-16 所示。

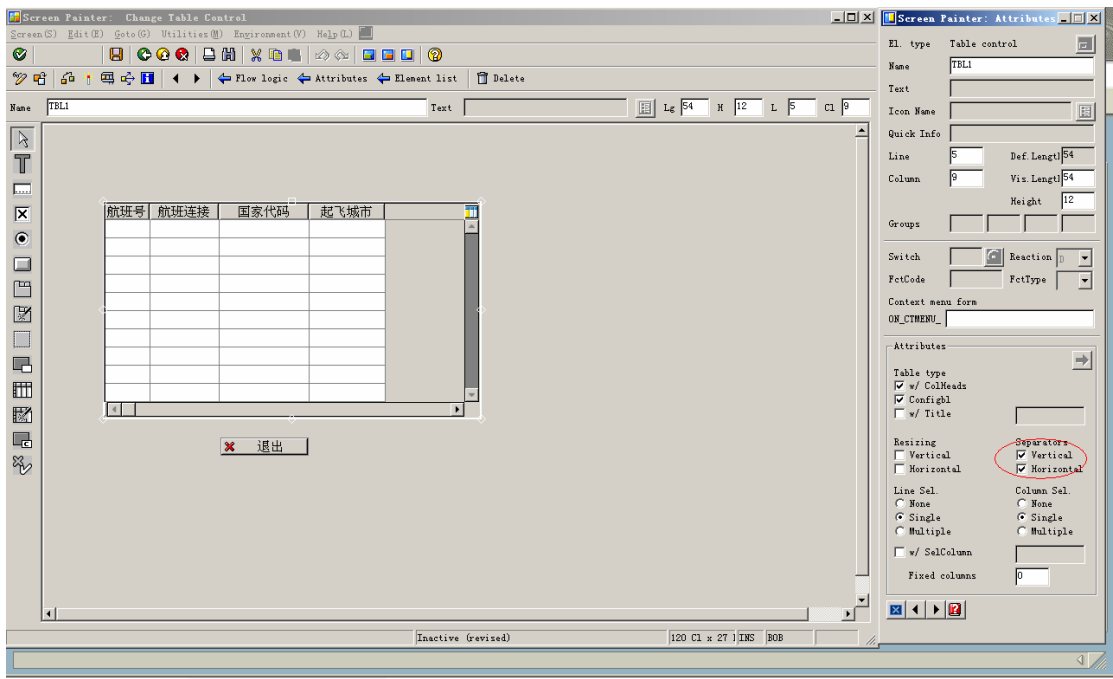


图 7-16

界面设计完成。

设计逻辑程序：

*PB0显示屏幕前的处理

PROCESS BEFORE OUTPUT.

MODULE STATUS_0100.

*处理LOOP从内表读到表控制

LOOP WITH CONTROL TBL1.

```
MODULE FILLTBL1.
ENDLOOP.
```

PROCESS AFTER INPUT.

*处理LOOP从表控制更新内表

```
LOOP WITH CONTROL TBL1.
MODULE READTBL1.
ENDLOOP.
MODULE USER_COMMAND_0100.
```

主程序:

REPORT YTEST21.

```
DATA: OK_CODE TYPE SY-UCOMM,
      SAVE_OK TYPE SY-UCOMM.
```

*定义内表，注意，两种写法一样

*DATA SP2 LIKE SPFLI OCCURS 0 WITH HEADER LINE.

DATA SP2 LIKE TABLE OF SPFLI WITH HEADER LINE.

*定义单结构记录

DATA SP1 TYPE SPFLI.

*增加内表数据

SELECT * INTO CORRESPONDING FIELDS OF TABLE SP2 FROM SPFLI.

*直接调用窗口

CALL SCREEN 100.

*定义表控制对象

CONTROLS TBL1 TYPE TABLEVIEW USING SCREEN 100.

*输出数据，看内表有无改变

LOOP AT SP2.

WRITE: / SP2-CARRID, SP2-CONNID, SP2-COUNTRYFR, SP2-CITYFROM.

ENDLOOP.

```
*&-----*
```

```
*&      Module  STATUS_0100  OUTPUT
```

```
*&-----*
```

```
*      text
```

```
*-----*
```

MODULE STATUS_0100 OUTPUT.

SET PF-STATUS 'STATUS1'.

ENDMODULE. " STATUS_0100 OUTPUT

```
*&-----*
```

```
*&      Module  USER_COMMAND_0100  INPUT
```

```
*&-----*
```

```
*      text
```

```

*-----*
MODULE USER_COMMAND_0100 INPUT.
  SAVE_OK = OK_CODE.
  CLEAR OK_CODE.
  CASE SAVE_OK.
    WHEN 'EXIT' .
      LEAVE TO SCREEN 0.
  ENDCASE.
ENDMODULE.                                " USER_COMMAND_0100  INPUT
*&-----*
*&      Module  FILL  OUTPUT
*&-----*
* 逐行从内表填写表控制
*-----*
MODULE FILLTBL1 OUTPUT.
  READ TABLE SP2 INTO SP1 INDEX TBL1-CURRENT_LINE.
ENDMODULE.                                " FILL  OUTPUT
*&-----*
*&      Module  READ  INPUT
*&-----*
* 逐行从内表控制更新内表
*-----*
MODULE READTBL1 INPUT.
  MODIFY SP2 FROM SP1 INDEX TBL1-CURRENT_LINE.
ENDMODULE.                                " READ  INPUT

```

程序输出结果如图 7-17 所示。

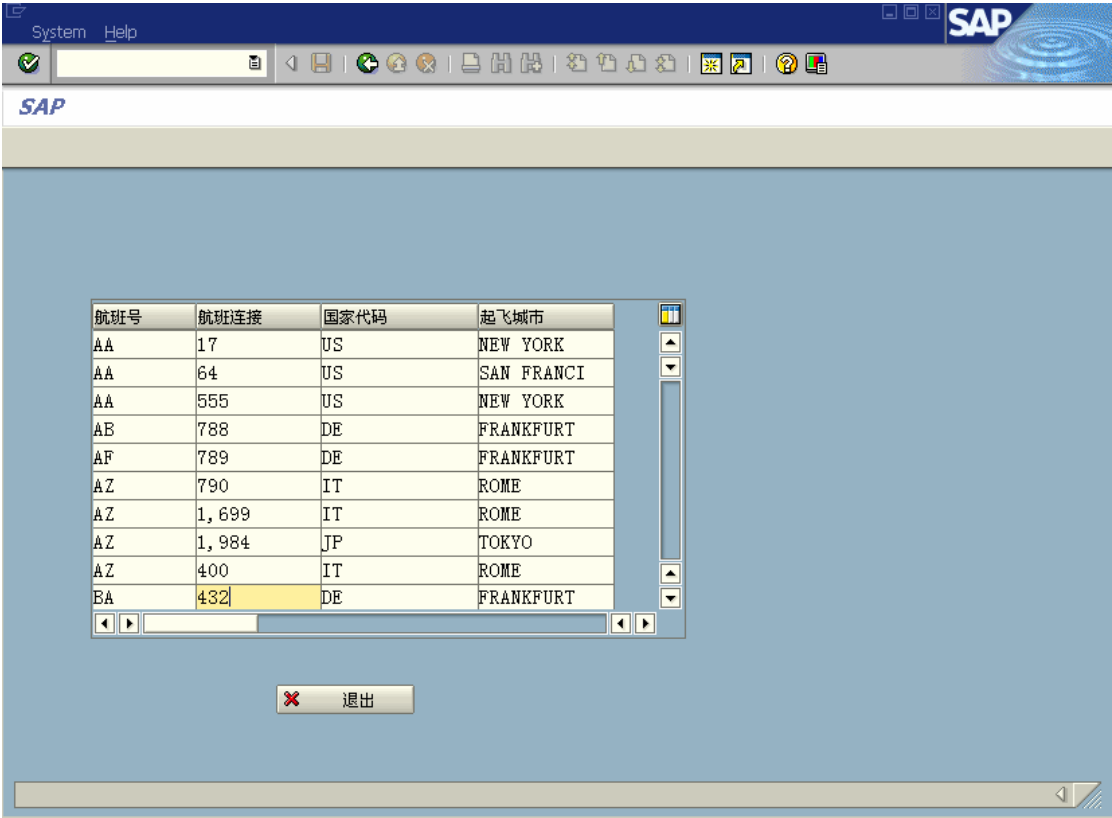


图 7-17

调整前 2 行数据，如图 7-18 所示。

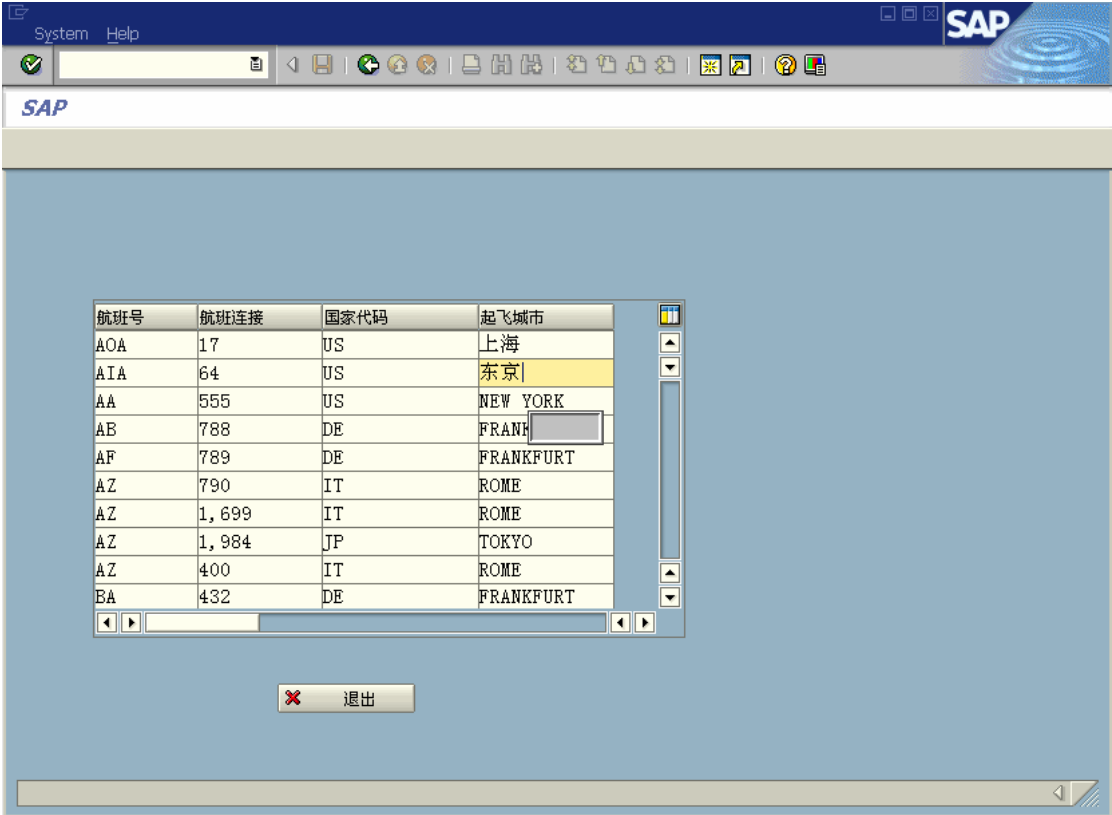


图 7-18

退出编辑屏幕，程序输出内表数据，屏幕显示数据已更改，如图 7-19 所示。

AOA	0017	US	上海
AIA	0064	US	东京

图 7-19

上述表格控件是没有垂直滚动条的, 原因是没有定义表格的记录数, 因而在 PBO 中还需要添加以下代码:

```
DATA NUMS TYPE I.
MODULE STATUS_0100 OUTPUT.
  SET PF-STATUS 'STATUS1'.
  IF NUMS = 0.
    DESCRIBE TABLE SP2 LINES NUMS.
    TBL1-LINES = NUMS.
  ENDIF.
ENDMODULE.
```

7.3 通过系统生成数据表维护程序

通过系统生成数据表维护程序 TCODE: SE55, 通过系统数据维护功能使用生成程序 (TCODE: SM30), 过程如下: 建立一个数据表 (SE11), 生成维护程序 (SE55), 数据维护 (SM30), 设计代码调用生成程序。

7.3.1 新建数据表

新建一个数据表 YMANA, 详细步骤请参阅“数据字典”, 表结构如图 7-20 所示。

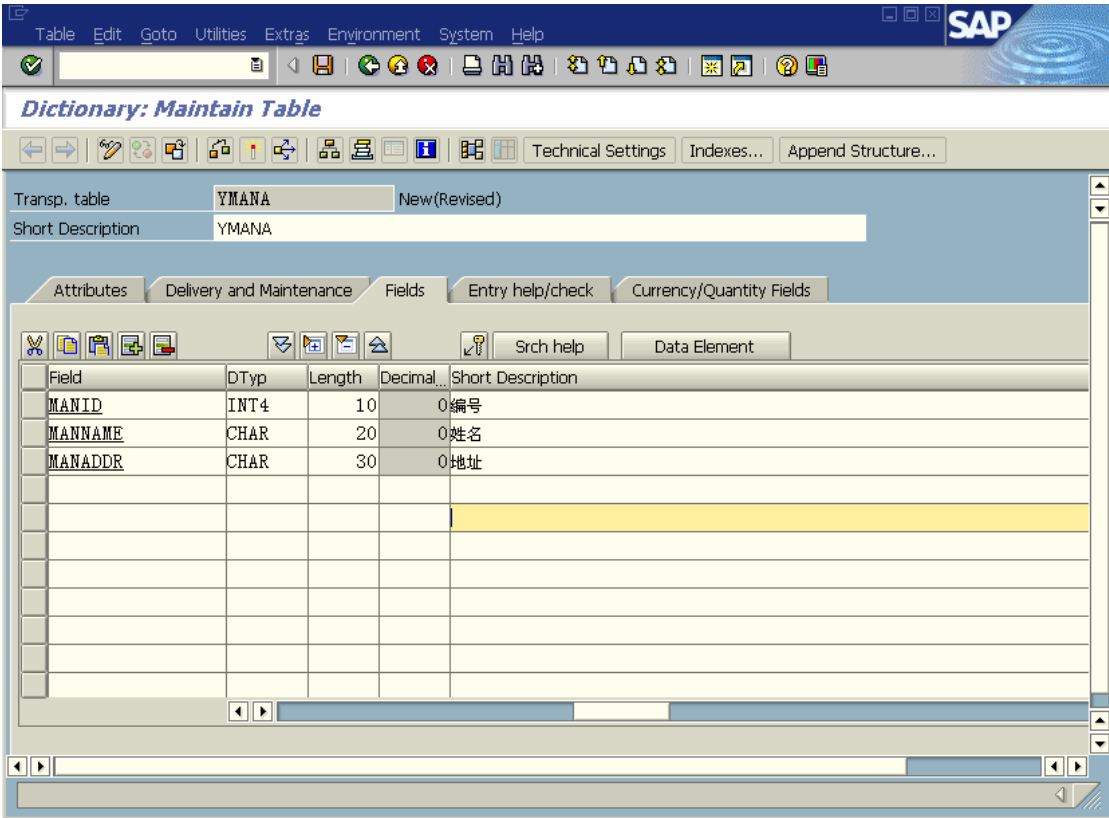


图 7-20

7.3.2 生成维护程序

输入 TCODE: SE55，输入函数组名称和屏幕号，选择权限组，如图 7-21 所示。

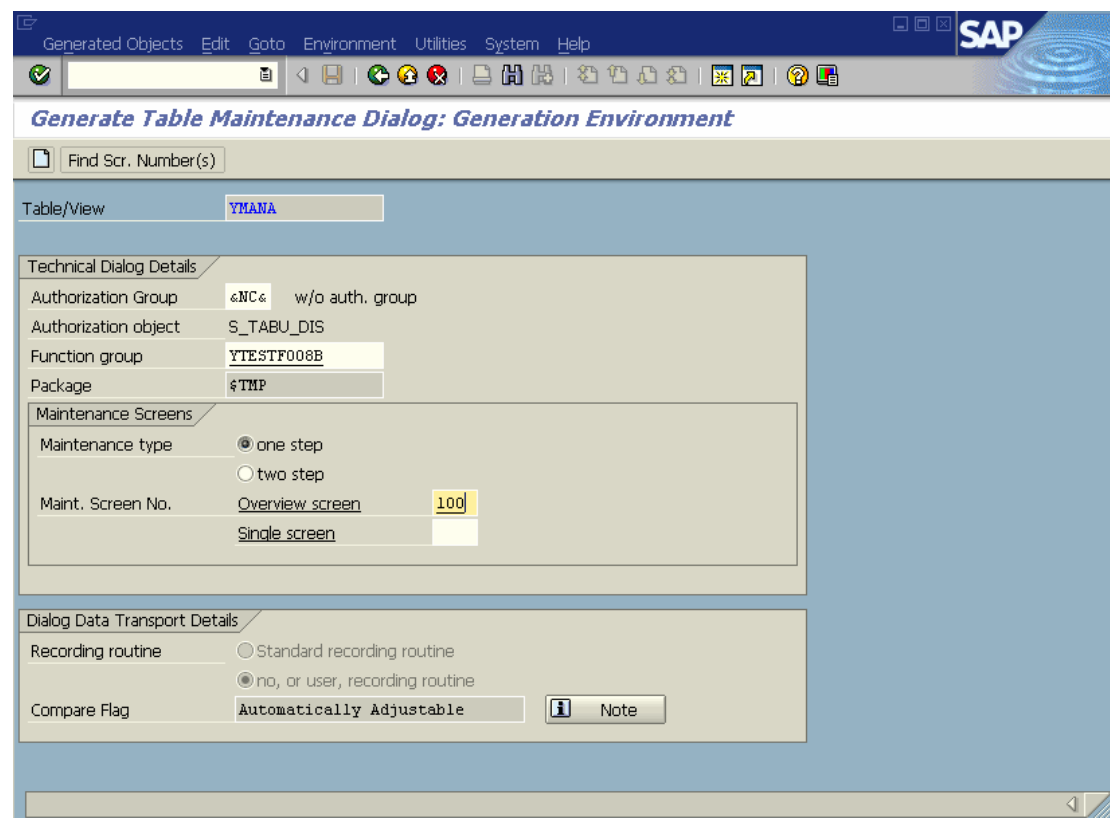


图 7-21

单击新建按钮，选择“本地对象”，生成维护程序。

7.3.3 数据维护

输入 TCODE: SM30，输入表名后单击维护按钮，如图 7-22 所示。

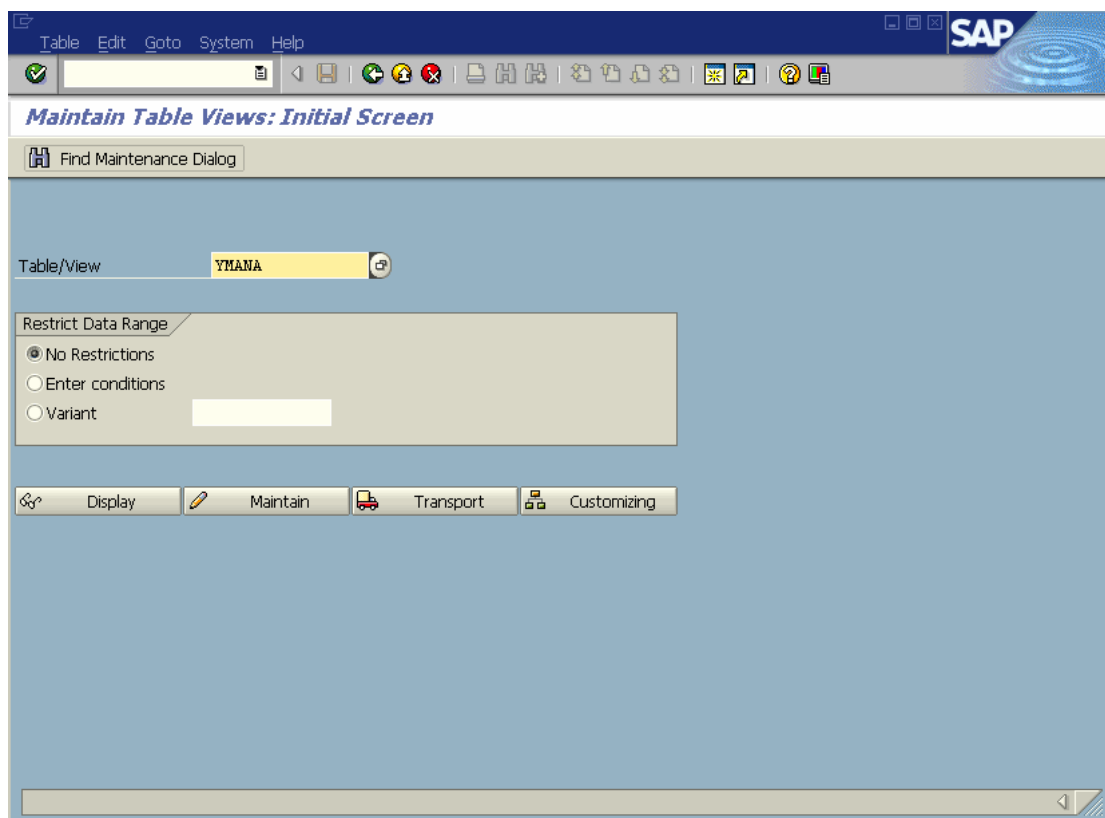


图 7-22

系统进入数据维护界面，如图 7-23 所示。

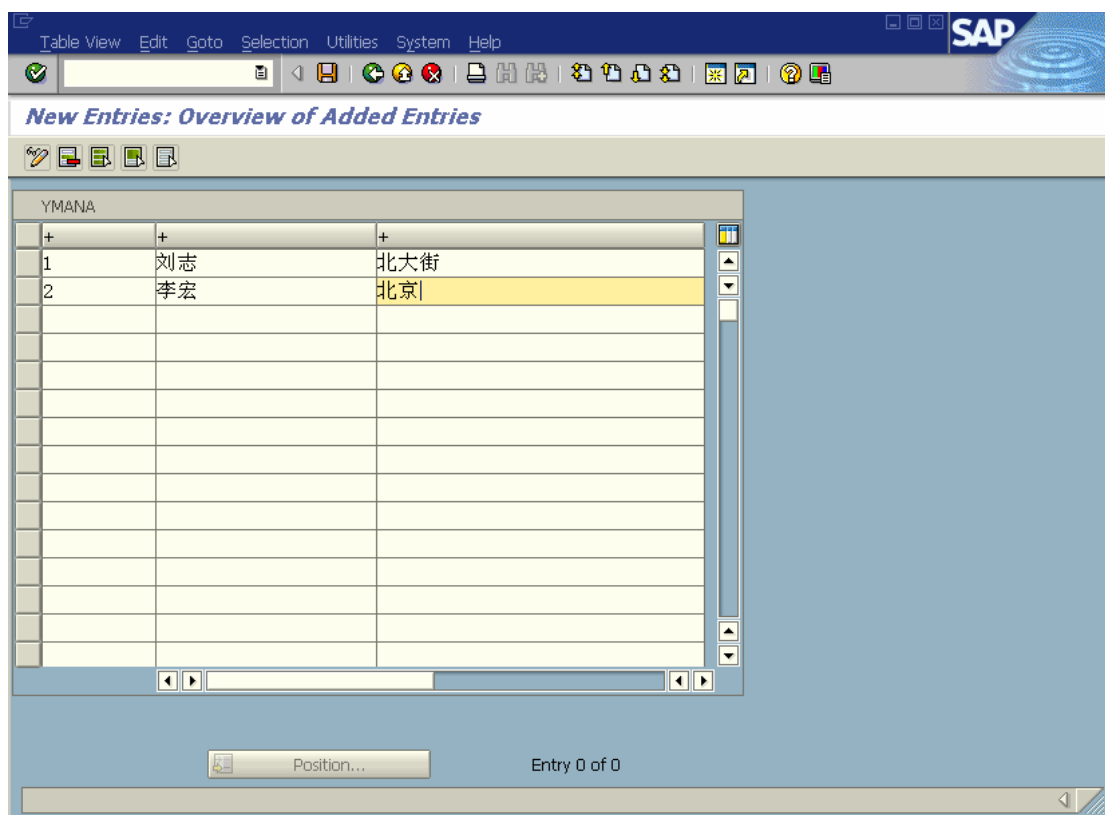


图 7-22

单击右下角的状态条，屏幕显示程序名为“SAPLYTESTF008B”如图 7-24 所示。

System	BOB (3) 000
✓ Client	000
User	
Program	SAPLYTESTF008B
Transaction	SM30
Response Time	47 ms
Interpretation Time	31 ms
Round Trips/Flushes	2/1

图 7-24

生成程序的表头等信息都可以调整，输入 TCODE: SE51 进入屏幕编辑器，输入程序名“SAPLYTESTF008B”，找到编号为 100 的屏幕，进入设计界面后，可以调整屏幕对象，如图 7-25 所示。

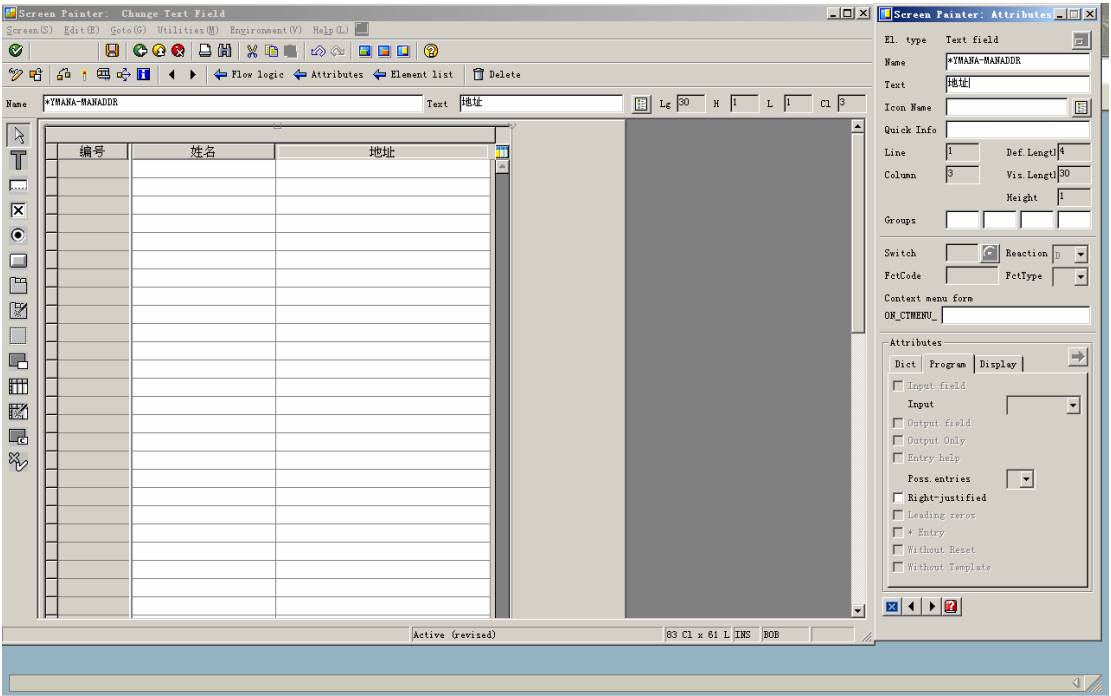


图 7-25

保存激活后，使用界面也相应改变，如图 7-26 所示。

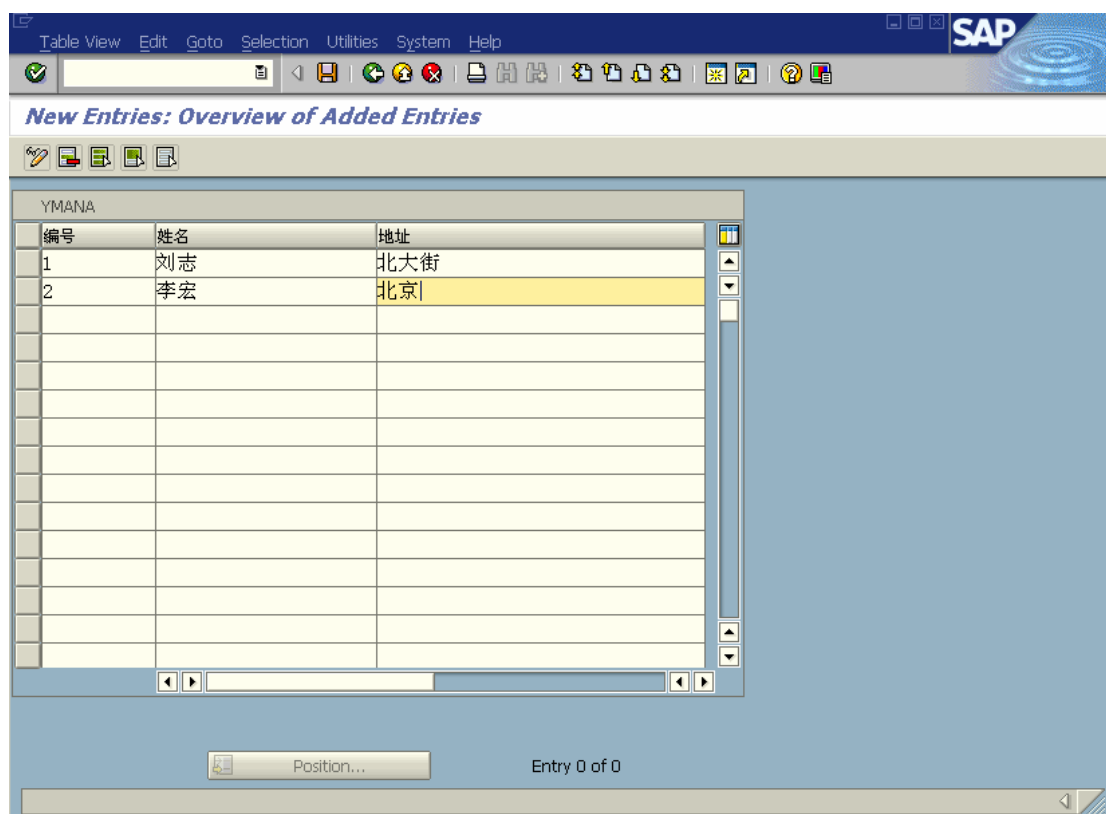


图 7-26

7.3.4 从程序调用生成的维护窗口

通过函数调用生成程序，程序如下：

```
REPORT YTEST23 .
CALL FUNCTION 'VIEW_MAINTENANCE_CALL'
  EXPORTING
    ACTION                = 'U'
    VIEW_NAME              = 'YMANA'
  EXCEPTIONS
    FOREIGN_LOCK           = 2
    NO_TVDIR_ENTRY         = 8
    .
IF SY-SUBRC <> 0.
* MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO
*          WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4.
ENDIF.
```

7.4 表格相关技巧

7.4.1 双击启动事件

定义状态条中的 F2 功能键，在程序中处理返回功能码，如图 7-27 所示。

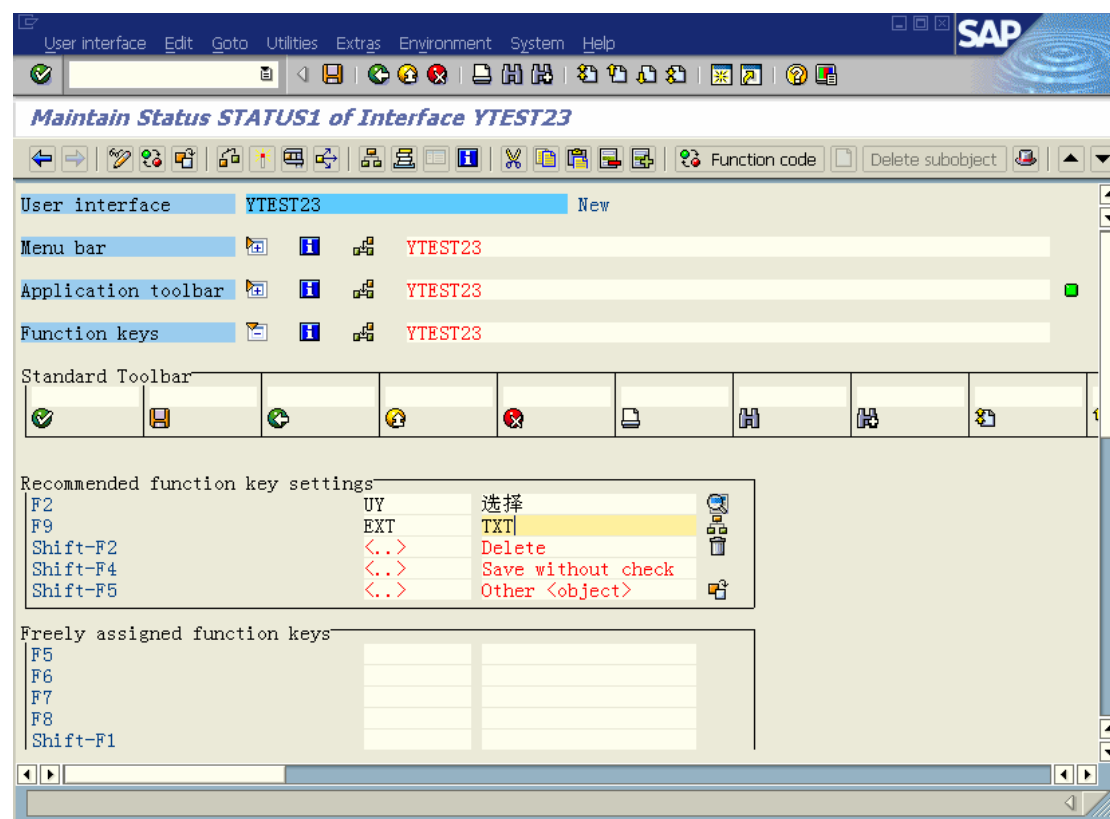


图 7-27

7.4.2 取得行号

程序:

*取得当前屏幕和行号

```
GET CURSOR LINE LINEA.
```

```
CHECK SY-SUBRC = 0.
```

*当前行号 = 当前屏首行序号 + 当前屏行号

```
LINEA = TBLA-TOP_LINE - 1.
```

*用取得当前行号读取内表

```
READ TABLE WA_FPFLI INDEX LINEA.
```

*在状态条显示

```
MESSAGE S005(YMESS) WITH LINEA.
```


7.4.3 定义表格的读写属性

程序:

WHEN 'READWR'.

*当单击按钮时，第一列只读，其他列读写切换

LOOP AT TBL1-COLS INTO ACOL WHERE INDEX GT 0.

IF SY-TABIX = 1.

ACOL-SCREEN-INPUT = '0'.

ELSE.

IF ACOL-SCREEN-INPUT = '0'.

ACOL-SCREEN-INPUT = '1'.

ELSEIF ACOL-SCREEN-INPUT = '1'.

ACOL-SCREEN-INPUT = '0'.

ENIF.

ENDIF.

ENDLOOP.

MODIFY TBL1-COLS FROM ACOL INDEX SY-TABIX.

输出界面如图所示。

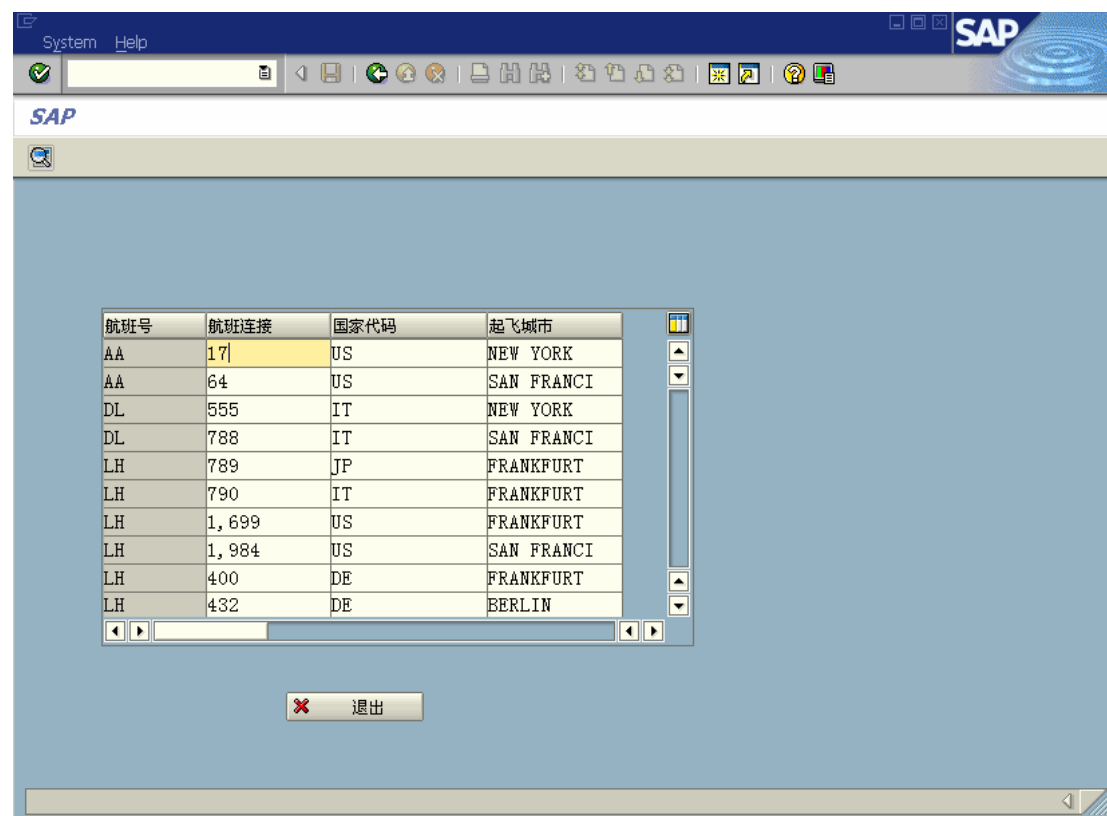


图7-28

7.5 通过表格维护变更内表数据后提交数据库

结合内表，对表格控件数据增加、修改、删除后提交到数据库，最终界面如图7-29所示。

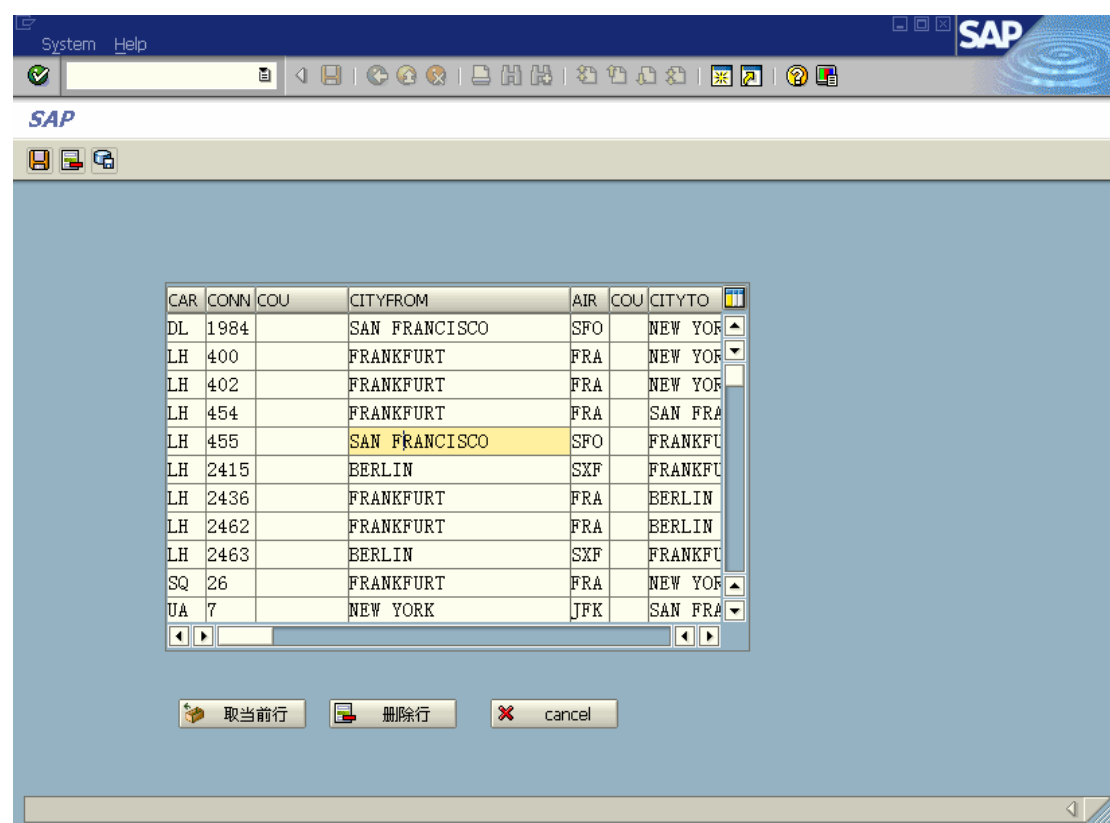


图7-29

步骤:

- (1) 用向导建立表格
- (2) 建立按钮“取当前行”，在事件中取表格控件当前行，并在状态条上显示当前行数据。
- (3) 建立按钮“删除行”，在事件中删除表格上当前选择的行。
- (4) 增加系统按钮“保存”，将内表数据提交数据库。

逻辑程序:

PROCESS BEFORE OUTPUT.

*&SPWIZARD: PBO FLOW LOGIC FOR TABLECONTROL 'TBL1'

MODULE TBL1_CHANGE_TC_ATTR.

*&SPWIZARD: MODULE TBL1_CHANGE_COL_ATTR.

LOOP AT SP1

WITH CONTROL TBL1

CURSOR TBL1-CURRENT_LINE.

*&SPWIZARD: MODULE TBL1_CHANGE_FIELD_ATTR

ENDLOOP.

* MODULE STATUS_0100.

*

PROCESS AFTER INPUT.

*&SPWIZARD: PAI FLOW LOGIC FOR TABLECONTROL 'TBL1'

LOOP AT SP1.

CHAIN.

```

        FIELD SP1-CARRID.
        FIELD SP1-CONNID.
        FIELD SP1-COUNTRYFR.
        FIELD SP1-CITYFROM.
        FIELD SP1-AIRPFROM.
        FIELD SP1-COUNTRYTO.
        FIELD SP1-CITYTO.
        FIELD SP1-AIRPTO.
        MODULE TBL1_MODIFY ON CHAIN-REQUEST.
    ENDCHAIN.
ENDLOOP.
*&SPWIZARD: MODULE TBL1_CHANGE_TC_ATTR.
*&SPWIZARD: MODULE TBL1_CHANGE_COL_ATTR.
    MODULE CANCEL AT EXIT-COMMAND.
    MODULE USER_COMMAND_0100.

主程序:
REPORT YTEST25 .
DATA: OK_CODE TYPE SY-UCOMM,
      SAVE_OK TYPE SY-UCOMM.
DATA: SP1 LIKE TABLE OF SPFLI WITH HEADER LINE,
      DELA_SPFLI LIKE TABLE OF SPFLI WITH HEADER LINE.
DATA LINEA TYPE I.
*&SPWIZARD: DECLARATION OF TABLECONTROL 'TBL1' ITSELF
CONTROLS: TBL1 TYPE TABLEVIEW USING SCREEN 0100.
DATA WA1 LIKE SPFLI.
SELECT * FROM SPFLI INTO CORRESPONDING FIELDS OF TABLE SP1.
CALL SCREEN 100.
*&SPWIZARD: OUTPUT MODULE FOR TC 'TBL1'. DO NOT CHANGE THIS LINE!
*&SPWIZARD: UPDATE LINES FOR EQUIVALENT SCROLLBAR
MODULE TBL1_CHANGE_TC_ATTR OUTPUT.
    DESCRIBE TABLE SP1 LINES TBL1-LINES.
ENDMODULE.                                "TBL1_CHANGE_TC_ATTR OUTPUT

*&SPWIZARD: INPUT MODULE FOR TC 'TBL1'. DO NOT CHANGE THIS LINE!
*&SPWIZARD: MODIFY TABLE
MODULE TBL1_MODIFY INPUT.
    MODIFY SP1
        INDEX TBL1-CURRENT_LINE.
ENDMODULE.                                "TBL1_MODIFY INPUT
*&-----*
*&      Module  STATUS_0100  OUTPUT
*&-----*
*      text

```

```

*-----*
MODULE STATUS_0100 OUTPUT.
    SET PF-STATUS 'STATUS1'.

ENDMODULE.                " STATUS_0100  OUTPUT
*&-----*
*&      Module  CANCEL  INPUT
*&-----*
*      text
*-----*
MODULE CANCEL INPUT.

ENDMODULE.                " CANCEL  INPUT
*&-----*
*&      Module  USER_COMMAND_0100  INPUT
*&-----*
*      text
*-----*
MODULE USER_COMMAND_0100 INPUT.
    SAVE_OK = OK_CODE.
    CLEAR OK_CODE.
    CASE SAVE_OK.
        WHEN 'EXIT'.
            LEAVE TO SCREEN 0.
        WHEN 'SAVE'.
            MODIFY SPFLI FROM TABLE SP1.
            IF SY-SUBRC NE 0.
                MESSAGE I005(YMESS) WITH '更新数据错误!'.
                EXIT.
            ELSE.
                MESSAGE I005(YESS) WITH '更新数据OK!'.
            ENDIF.
            DELETE SPFLI FROM TABLE DELA_SPFLI.
            IF SY-SUBRC NE 0.
                MESSAGE I005(YMESS) WITH '更新数据错误!'.
            ELSE.
                MESSAGE I005(YESS) WITH '更新数据OK!'.
            ENDIF.
        WHEN 'DELA'.
            GET CURSOR LINE LINEA.
            CHECK SY-SUBRC = 0.
            LINEA = TBL1-TOP_LINE + LINEA - 1.
            READ TABLE SP1 INDEX LINEA.
            APPEND SP1 TO DELA_SPFLI.
    ENDCASE.
ENDMODULE.

```

```

        DELETE SP1 INDEX : LINEA.
    WHEN 'GETDATA'.
        GET CURSOR LINE LINEA.
        CHECK SY-SUBRC = 0.
        LINEA = TBL1-TOP_LINE - 1.
        READ TABLE SP1 INDEX LINEA.
        MESSAGE S006(YMESS) WITH SP1-CARRID SP1-CITYFROM.
    ENDCASE.
ENDMODULE.                                " USER_COMMAND_0100  INPUT

```