

# SPL \* FORMS 触发器的应用与设计

林 郁 (福建省龙岩卷烟厂计算中心)

**摘要:**触发器(*TRIGGER*)设计是 *SQL \* FORMS* 进行开发应用的关键及难点所在,本文描述了触发器的构成及执行机制,并用某 *FORM* 中触发器实例详细说明它的应用及设计。

## 一、引言

1. ORACLE 是当前颇为流行的关系数据库管理系统。SQL \* FORMS 是 ORACLE 的产品之一。是一个很具特色的应用生成器,是 ORACLE 提供的一种自动生成和运行交互式应用的工具软件,是基于表格的应用快速开发工具,是适合支持当前企业 MIS 原型开发的一个强有力的工具。

SQL \* FORMS 提供了强有力的触发器(Trigger)机制。为维护数据的完整性,数据的检验等提供了重要的工具。

触发器的设计是 SQL \* FORMS 的精华体现,但又是很复杂的,要求较高,可以说是难点所在。充分利用触发器不仅可提高效率,而且可设计出功能强且灵活、用户操作方便的应用。

利用触发器可以完成以下任务:

(1) 执行各种基本操作前或后进行预处理或后处理,包括输入数据的验证、插入、删除、修改记录时,自动进行计算,自动生成某些数据;

(2) 重定义功能键;

(3) 调用高级语言编写的用户程序(User Esit)。

2. 触发器在 SQL \* FORMS 中分为三级:

(1) 域级:触发器对应于一个指定的域;

(2) 块级:触发器对应于一个指定的块;

(3) FORM 级:触发器对应于整个 FORM。

触发器分成与级有关的触发器与级无关的触发器两种,与级无关的触发器又分成功能键触发器和用户名触发器。

## 二、触发器的构成及执行控制

1. 在 SQL \* FORMS 中触发器是指一条命令或一

组命令,在运行 FORM 时,它们因被某些事件触发而执行。一个触发器是由一个或多个步组成的,每步触发器步又由一条触发器命令及控制机制组成,其中触发器命令完成这一步所要完成的操作,而控制机制则决定了这一步执行完了该如何处理。

SQL \* FORMS 中可以使用三种命令编写触发器步。

A. SQL 命令, B. SQL \* FORMS 命令, C. User Exit(用户子程序)。

每一触发器的属性包括以下属性:

- Abort trigger when step fail: 在该步失败时,中止触发器的执行(此为缺省项);

- Reverse return code 把成功 / 失败的标准颠倒过来。

- Return success when aborting trigger.

此项仅在第一项被选择时有意义,意为触发器中止时返回成功。

- Separate cursor data area: 决定是否为每一步单独分配空间(此项仅为用 SQL 语句写的触发器步时有意义)。

- Success label: 成功转向的步标号。

- Failure label: 失败转向的步标号。

每一触发器的执行结果只有成功、失败两种,而每一个触发器步执行的结果有三种:成功、失败、致命错。

2. 每一个触发器步的控制成份应包括如下内容:

(1) 在顺序执行时,决定各步次序的步顺序号 Sep#;

(2) 作为其它步执行条件转移到这一步的标识的步标号 Label;

(3) 指明这一步执行成功或失败后转移去向的标号,成功标号和失败标号。(这是在其它步中定义的步标号)

(4) 表明在这步执行失败后,是否停止执行的标志。

(5)表明在这步执行失败并停止执行其它步时,触发器本身认为是成功还是失败的标号。

(6)另外个一个标志是当前步命令执行成功失败的标准是什么。

### 3. 触发器执行结果及原因

其中命令类型		结果		
		成功	失败	致命错误
SQL 命令	SELECT INSERT UPDATE DELETE	数据库中至少有一行操作对象	一行操作对象也没有	有句法错或表不存在
其它		没有句法错	不能发生	
SQL * FORMS 命令		能够正确执行	由于任何原因(包括句法错)而不能执行	内存不够
用户子程序		取决于程序的内在逻辑(但必须返回结果)		被调用触发器不存在

==== 甲班产品入库登记 ====

车间		年 月 日 (箱)	
编号	牌号	甲班成品	甲班次品
备注			

车间主任      仓管员      车间统计员

### 三. 触发器设计举例

笔者在参加开发“龙岩卷烟厂管理信息系统”(采用 ORACLE5.1B) 工程中利用 SQL \* FORMS 开发其中的“成品仓库管理子系统”,在该子系统的开发中用户充分参与,开发人员对原型逐步求精,取得了很好的效果,用户使用后较为满意。下面拟用在构造该子系统的—个 FORM 中所涉及的一些较为复杂的触发器例子来说明

触发器执行机制及独特作用。

1. 设计如下的 FORM,目标—当用户输入车间名和年月日后,要达到输入,修改,查询功能。

(1)为了构造此 FORM 作了如下的规划:

设计成由 B0,B1,B2,B3 四个块组成,其中:

B0 块由 CJMING (车间名), NYRI00 (年月日), X1 (年), X2 (月), X3 (日), X4 (中间变量) 共 6 个域组成;其中 NYRI00, X4 为隐含域。

B1 块由 BZHU00(备注), CJZREN (车间主任), CGYUAN (仓管员), CJTJY0 (车间统计员), CJMING (车间名), NYRI00 (年月日) 共 6 个域组成,这 6 个域对应于备注表 (SJD311) 的 6 个相应列,其中 NYRI00, CJMING 为隐含域 B2 块由 NYRI00(年月日), PHAO00(牌号), JBCPRK (甲班次品入库) JBCPRK(甲班次品入库), CJMING(车间名), X1 (中间变量) 6 个域组成,前 5 个域对应于入库表 (SJD312) 的 5 个相应的列,其中 NYRI00, CJMING 为隐含域, B3 块由 PHMING(牌号名), XHAO00 (序号) 2 个域组成,对应于牌号特性表 (SJ6010) 的相应列。

各块的作用及关系: B0 块是控制块,提供该 FORM 与用户的界面, B3 块是该厂牌号特性管理信息,供用户输入牌号时提示用, B1 块是各车间每天的入库管理信息; B2 块是 B1 块所对应各牌号的具体入库数; B1 块与 B2 块对应的基表分别为 SJD311, SJD312, 它们之间关系是 1:N, 并以 NYRI00 (年月日), CJMING (车间名) 为关键字。

(2) 此 FORM 中设计的触发器类型有 KEY- OTHERS, KEY- EXIT, KEY- NXTFLD, KEY- DELREC, KEY- RPBVLK, KEY- UP, KEY- DOWN, POST- CHANGE, POST- UPDATE 等,下面列举了几个触发器例子。

• 对整个 FORM 设计 1 个触发器(即 FORM 级触发器)

类型: PRE-FORM

步 1: 标号 L1

```
SELECT 'X' FROM DUAL
WHERE TO-NUMBER (TO-CHAR)
SYSDATE, 'HH24') BETWEEN 7 AND 17
```

出错信息: 非法上机时间!!!

属性: \* ABORT TRIGGER WHEN STEP FALL

此触发器的作用:限制了用户的上机时间

• 在 B0 块的域 CJMING 的 ATTRIBUTE 中选择 AUTOMATIC HELP 项,在 VALIDATION 中填写的提示内容为:“1——咀三 2——咀二 3——普二”

并在此域上设计如下一个域级触发器

类型:KEY-NXTFLD

步 1 标号 L1

```
SELECT'咀 三 ' INTO:BO. CJMING FROM
DUAL
```

```
WHERE SUBSTR (: BO.CJMING,1,1)='1'
```

属性:SUCCESS LABEL L5

FAILURE LABEL L2

步 2 标号 L2

```
SELECT'咀 二 ' INTO:BO.CJMING FROM
DUAL
```

```
WHERE SUBSTR (:BO.CJMING,1,1)='2'
```

属性:SUCCESS LABEL L5

FAILURE LABEL L3

步 3 标号 L3

```
SELECT'普 二 ' INTO: BO.CJMING FROM
DUAL
```

```
WHERE SUBSTR (:BO. CJMING,1,1)='3'
```

属性:SUCCESS LABEL L5

FAILURE LABEL L4

步 4 标号 L4

```
SELECT'X'FROM DUAL WHERE SUBSTR
(:BO. CJMING, 1, 2)= '咀'OR
```

```
SUBSTR(:BO.CJMING,1,2)='普'
```

出错信息:输入错,请重输!!!

属性:\* ABORT TRIGGER WHEN STEP FAIL

步 5 标号 L5

```
#ESEMACHO NULL;
```

属性:\* ABORT TRIGGER WHEN STEP FAIL

此触发器的作用:限制了用户对车间名的输入范围。

• 在 B0 块的域 X1,X2,X3 上为了限制用户输入的年份、月份、日期的合法范围,在这 3 个域上各设计了 POST-CHANGE 触发器,以在 X2 域上设计的触发器为例:

类型:POST-CHANGE

步 1 标号 L1

```
SELECT'X'FROM DUAL WHERE :BO.X1 > =
'01'AND :BO.X1 < = '12'
```

出错信息:月份输入错,请重输!!!

属性:\* ABORT TRIGGER WHEN STEP FAIL

此触发器的作用:限制了用户对月份的输入范围。

在 X3 域上还定义如下两个触发器:

1.类型。KEY-OTHERS

步 1 标号 L1

```
#EXEMACHO NULL;
```

属性:\* ABORT TRIGGER WHEN STIP FAIL.

此触发器的作用:屏蔽其它功能键

2.类型。KEY-EXIT

步 1 标号 L1

```
#EXEMACHO NULL;
```

属性:\* ABORT TRIGGER WHEN STIP FAIL.

此触发器的作用:按下 <EXC> 键时退出

• 在 B1 块一设计 1 个域级触发器

类型:KEY-DELREC

步 1 标号 L1

```
SELECT'X'FROM SJD312 WHERE:B1.
NYRI00 = SJD312. NYRI00
```

属性:SUCCESS LABEL L2

FAILURE LABEL L4

步 2 标号 L2

```
SELECT'X'FROM DUAL WHERE 1 > 2
```

出错信息:删此信息时,应先用 <SHIFT> + <F6> 键把下面相关信息全删掉!!

属性:SUCCESS LABEL L3

FAILURE LABEL L3

步 3 标号 L3

```
#EXEMACHO GOBLK B2
```

属性:SUCCESS LABEL L7

FAILURE LABEL L7

步 4 标号 L4

```
#EXEMACHO DELREC;
```

属性:SUCCESS LABEL L5

FAILURE LABEL L5

步 5 标号 L5

```
#EXEMACHO COMMIT;
```

属性:SUCCESS LABEL L6

FAILURE LABEL L6

步 6 标号 L6

```
#EXEMACHO GOBLK B0
```

属性:\* ABORT TRIGGER WHEN STEP FAIL

步 7 标号 L7

```
#EXEMACHO NULL;
```

属性:\* ABORT TRIGGER WHEN STEP FAIL

(下转第 54 页)

(上接第 25 页)

此触发器的设计目的及作用:当用户在做 B1 块中记录的删除时,为了确保库中数据的一致性,避免发生只删除 B1 块中的记录,而不删 B2 块中的相应记录,使 B2 块中的相应记录因 B1 块主记录的删除而丢失(B2 块中相应记录实际上未删除)情况,一般采取在删 B1 块中记录的同时,把 B2 块中相应记录也一起删掉的方法。在实现中由于屏幕大小所限,B2 块只能同时看到六条相应的记录,而生产的牌号总数在 10 个左右,为了避免发生删除 B1 块中的记录时,把 B2 块中相关的不必删的记录一起删掉,采用了在删 B1 块中记录时,先判 B2 块中是否有相关的记录,若有则转 B2 块中去做 B2 块相应记录的删除;若无,则把 B1 块中的该记录删掉。

• 在 B3 块上定义块级触发器 3 个:

1.类型。KEY-DOWN

步 1 标号 L1

```
#EXEMACRO SCRKOWN;
```

属性: \* ABORT TRIGGER WHEN STEP FAIL

此触发器的作用:当按下 <↓> 键时屏幕向下翻一屏

2.类型。KEY-UP

步 1 标号 L1

```
#EXEMACRO SCRUP;
```

属性: \* ABORT TRIGGER WHEN STEP FAIL

此触发器的作用:当按下 <↑> 键时屏幕向上翻一屏

3.类型。KEY-OTHERS

步 1 标号 L1

```
#EXEMACRO PRVBLK;
```

属性: \* ABORT TRIGGER WHEN STEP FAIL

此触发器的作用:当按下 <ENTER> 键时返回输入状态

#### 四、结束语

触发器的设计是应用 SQL \* FORMS 进行设计系统原型的重点、难点所在,要熟练掌握 SQL \* FORMS 这一工具,就必须掌握好触发器的设计要领。限于篇幅,文中不可能列出该 FORM 中所有的触发器。

\*\* 本文的写作得到航空航天部 204 所陆庭凡同志的指导及帮助,谨此致谢!

#### 参考文献:

[1]杜小勇等 <<ORACLE5.10 版相关数据库系统教程>>——希望高级电脑技术公司 1990,11 北京