

```

1  USE MEI_TRAB
2
3
4  if exists (select * from sys.triggers where Name = 'TR_Encomenda_I_U')
5  Begin
6      DROP TRIGGER TR_Encomenda_I_U
7  End
8  GO
9
10 -- Trigger associado a Insert e Update. O Delete de encomendas é tratado à parte.
11
12 CREATE TRIGGER TR_Encomenda_I_U
13 ON Encomenda
14 FOR INSERT, UPDATE
15 AS
16 DECLARE @NumReg int
17 DECLARE @EventType Char(1)
18
19 DECLARE @DELETEDCOUNT INT
20 DECLARE @INSERTEDCOUNT INT
21
22 DECLARE @EncId int
23 DECLARE @Objecto varchar(30)
24 DECLARE @Valor varchar(100)
25 DECLARE @Referencia varchar(100)
26
27 SELECT @DELETEDCOUNT = COUNT(*) FROM deleted
28 SELECT @INSERTEDCOUNT = COUNT(*) FROM inserted
29
30 -- Event type
31 IF (@DELETEDCOUNT > 0) AND (@INSERTEDCOUNT > 0 )
32     Set @EventType = 'U'
33 ELSE IF (@INSERTEDCOUNT > 0 )
34     Set @EventType = 'I'
35
36 Set @Objecto = 'Encomenda' -- Tabela encomenda
37 Set @Referencia = NULL -- Para estas operações não é usada a referência
38
39 DECLARE fh_cursor CURSOR FOR
40 SELECT EncId FROM inserted ORDER by EncId
41
42 OPEN fh_cursor
43
44 FETCH NEXT FROM fh_cursor INTO @EncId
45
46 WHILE (@@fetch_status = 0)
47 BEGIN
48
49     Set @Valor = CAST(@EncId as nvarchar(100)) -- Convert int to varchar
50
51     INSERT INTO LogOperations (EventType, Objecto, Valor, Referencia)
52     Values (@EventType, @Objecto, @Valor, @Referencia)
53
54
55     FETCH NEXT FROM fh_cursor INTO @EncId
56 END
57
58 IF (@@fetch_status = -1)
59 Begin
60     CLOSE fh_cursor
61     DEALLOCATE fh_cursor
62     RETURN
63 End
64
65 on_error:
66 CLOSE fh_cursor
67 DEALLOCATE fh_cursor
68
69
70 RAISERROR('Can't insert the record',16,1 )
71 ROLLBACK TRANSACTION

```

```

72 GO
73
74
75
76 if exists (select * from sys.triggers where Name = 'TR_Encomenda_D')
77 Begin
78     DROP TRIGGER TR_Encomenda_D
79 End
80 GO
81
82
83 CREATE TRIGGER TR_Encomenda_D
84 ON Encomenda
85 FOR DELETE
86 AS
87 DECLARE @NumReg int
88 DECLARE @EventType Char(1)
89
90 DECLARE @EncId int
91 DECLARE @ClienteID int
92 DECLARE @Nome nvarchar(30)
93 DECLARE @Morada nvarchar(30)
94
95 DECLARE @Objecto varchar(30)
96 DECLARE @Valor varchar(100)
97 DECLARE @Referencia varchar(100)
98
99 Set @EventType = 'D'
100
101 Set @Objecto = 'Encomenda' -- Tabela encomenda
102 Set @Referencia = NULL -- Para estas operações não é usada a referência
103
104 --
105 -- with a cursor we can iterate the deleted rows
106 --
107
108
109 DECLARE fh_cursor CURSOR FOR
110 SELECT EncId, ClienteId, Nome, Morada FROM Deleted ORDER by EncId
111
112 OPEN fh_cursor
113
114 FETCH NEXT FROM fh_cursor INTO @EncId, @ClienteId, @Nome, @Morada
115
116 WHILE (@@fetch_status = 0)
117 BEGIN
118     Set @Valor = CAST(@EncId as nvarchar(10)) + '|' + @Nome + '|' + @Morada
119
120     INSERT INTO LogOperations (EventType, Objecto, Valor, Referencia)
121     Values (@EventType, @Objecto, @Valor, @Referencia)
122
123     FETCH NEXT FROM fh_cursor INTO @EncId, @ClienteId, @Nome, @Morada
124 END
125
126 IF (@@fetch_status = -1)
127 Begin
128     CLOSE fh_cursor
129     DEALLOCATE fh_cursor
130     RETURN
131 End
132
133 on_error:
134     CLOSE fh_cursor
135     DEALLOCATE fh_cursor
136
137
138     RAISERROR('Can't insert the record',16,1 )
139     ROLLBACK TRANSACTION
140 GO
141
142 if exists (select * from sys.triggers where Name = 'TR_EncLinha_I_U')

```

```

143 Begin
144     DROP TRIGGER TR_EncLinha_I_U
145 End
146 GO
147
148
149 CREATE TRIGGER TR_EncLinha_I_U
150 ON EncLinha
151 FOR INSERT, UPDATE
152 AS
153 DECLARE @EventType Char(1)
154
155 DECLARE @EncId int
156 DECLARE @ProdutoID int
157
158 DECLARE @Objecto varchar(30)
159 DECLARE @Valor varchar(100)
160 DECLARE @Referencia varchar(100)
161
162 DECLARE @DELETEDCOUNT INT
163 DECLARE @INSERTEDCOUNT INT
164
165 SELECT @DELETEDCOUNT = COUNT(*) FROM deleted
166 SELECT @INSERTEDCOUNT = COUNT(*) FROM inserted
167
168 -- Event type
169 IF (@DELETEDCOUNT > 0) AND (@INSERTEDCOUNT > 0 )
170     Set @EventType = 'U'
171 ELSE IF (@INSERTEDCOUNT > 0 )
172     Set @EventType = 'I'
173
174 Set @Objecto = 'EncLinha' -- Tabela encomenda
175
176
177 DECLARE fh_cursor CURSOR FOR
178 SELECT EncId, ProdutoId FROM inserted ORDER by EncId
179
180 OPEN fh_cursor
181
182 FETCH NEXT FROM fh_cursor INTO @EncId, @ProdutoId
183
184 WHILE (@@fetch_status = 0)
185 BEGIN
186
187
188     Set @Valor = CAST(@EncId as varchar(100)) -- Convert int to varchar
189     Set @Referencia = CAST(@ProdutoId as varchar(100)) -- Convert int to varchar
190
191     INSERT INTO LogOperations (EventType, Objecto, Valor, Referencia)
192     Values (@EventType, @Objecto, @Valor, @Referencia)
193
194
195     FETCH NEXT FROM fh_cursor INTO @EncId, @ProdutoId
196 END
197
198 IF (@@fetch_status = -1)
199     Begin
200         CLOSE fh_cursor
201         DEALLOCATE fh_cursor
202         RETURN
203     End
204
205 on_error:
206     CLOSE fh_cursor
207     DEALLOCATE fh_cursor
208
209
210     RAISERROR('Can't insert the record',16,1 )
211     ROLLBACK TRANSACTION
212 GO
213

```

```

214
215
216 if exists (select * from sys.triggers where Name = 'TR_EncLinha_D')
217 Begin
218     DROP TRIGGER TR_EncLinha_D
219 End
220 GO
221
222
223 CREATE TRIGGER TR_EncLinha_D
224 ON EncLinha
225 FOR Delete
226 AS
227 DECLARE @EventType Char(1)
228
229 DECLARE @DELETEDCOUNT INT
230 DECLARE @INSERTEDCOUNT INT
231
232 DECLARE @EncId int
233 DECLARE @ProdutoID int
234 DECLARE @Designacao nvarchar(50)
235 DECLARE @Preco Decimal(10,2)
236 DECLARE @Qtd Decimal(10,2)
237
238 DECLARE @Objecto varchar(30)
239 DECLARE @Valor varchar(100)
240 DECLARE @Referencia varchar(100)
241
242 Set @EventType = 'D'
243
244 Set @Objecto = 'EncLinha' -- Tabela encomenda
245 Set @Referencia = NULL -- Para estas operações não é usada a referência
246
247
248 DECLARE fh_cursor CURSOR FOR
249 SELECT EncId, ProdutoId, Designacao, Preco, Qtd FROM Deleted ORDER by EncId
250
251 OPEN fh_cursor
252
253 FETCH NEXT FROM fh_cursor INTO @EncId, @ProdutoId, @Designacao, @Preco, @Qtd
254
255 WHILE (@@fetch_status = 0)
256 BEGIN
257     Set @Valor = CAST(@EncId as nvarchar(10)) + '|' + CAST(@ProdutoId as
    nvarchar(10))
258     Set @Referencia = @Designacao + '|' + CAST(@Preco as nvarchar(10)) + '|' +
    CAST(@Qtd as nvarchar(10))
259
260     INSERT INTO LogOperations (EventType, Objecto, Valor, Referencia)
261     Values (@EventType, @Objecto, @Valor, @Referencia)
262
263
264     FETCH NEXT FROM fh_cursor INTO @EncId, @ProdutoId, @Designacao, @Preco, @Qtd
265 END
266
267 IF (@@fetch_status = -1)
268 Begin
269     CLOSE fh_cursor
270     DEALLOCATE fh_cursor
271     RETURN
272 End
273
274 on_error:
275 CLOSE fh_cursor
276 DEALLOCATE fh_cursor
277
278
279 RAISERROR('Can't insert the record',16,1 )
280 ROLLBACK TRANSACTION
281 GO
282

```