

# Apêndice Complementar - WEI'2018

## DDL do Modelo Bancário

A Listagem 1 apresenta a DDL do modelo bancário gerada automaticamente pela extensão. É possível constatar que a DDL é fiel ao modelo bancário ilustrado na Figura 3 do artigo, pois o Algoritmo 1 utiliza cada regra de mapeamento ilustrado na Tabela de Mapeamento (Tabelas 2, 3 e 4). Comentários foram adicionados ao *script* SQL para demonstrar quais são as regras utilizadas para gerar cada elemento ER. Por exemplo, as linhas 1 a 33 apresentam a DDL gerada para criação de tabelas na qual utiliza a *Regra de Mapeamento #1* presente na Tabela de Mapeamento 6.

```
1  -- Início da Regra de Mapeamento #1
2  CREATE TABLE GERENTE (
3  id NUMBER NOT NULL,
4  nome VARCHAR2(128)
5  );
6
7  CREATE TABLE AGENCIA (
8  codigo NUMBER NOT NULL,
9  nome VARCHAR2(128)
10 );
11
12 CREATE TABLE CONTA (
13 saldo NUMBER,
14 numero NUMBER NOT NULL
15 );
16
17 CREATE TABLE CORRENTE (
18 tx_manutencao NUMBER
19 );
20
21 CREATE TABLE POUPANCA (
22 dia_aniversario NUMBER
23 );
24
25 CREATE TABLE CORRENTISTA (
26 nome VARCHAR2(128),
27 cpf NUMBER NOT NULL
28 );
29
30 CREATE TABLE CHEQUE (
31 num_registro NUMBER NOT NULL
32 );
33 -- Fim da Regra de Mapeamento #1
34
35 -- Início da Regra de Mapeamento #2
36 ALTER TABLE GERENTE ADD CONSTRAINT PK_GERENTE PRIMARY KEY (id);
37
38 ALTER TABLE AGENCIA ADD CONSTRAINT PK_AGENCIA PRIMARY KEY (codigo);
39
40 ALTER TABLE CONTA ADD CONSTRAINT PK_CONTA PRIMARY KEY (numero);
41
42 ALTER TABLE CORRENTISTA ADD CONSTRAINT PK_CORRENTISTA PRIMARY KEY (cpf);
43 -- Fim da Regra de Mapeamento #2
44
45 -- Início da Regra de Mapeamento #4
46 ALTER TABLE CHEQUE ADD numero_conta NUMBER NOT NULL;
47 ALTER TABLE CHEQUE ADD CONSTRAINT PK_CHEQUE PRIMARY KEY (num_registro, numero_conta);
48 ALTER TABLE CHEQUE ADD CONSTRAINT FK_CHEQUE FOREIGN KEY (numero_conta) REFERENCES CONTA (numero);
49 -- Fim da Regra de Mapeamento #4
50
51 -- Início da Regra de Mapeamento #18
52 ALTER TABLE CORRENTE ADD numero_conta NUMBER NOT NULL;
53 ALTER TABLE CORRENTE ADD CONSTRAINT PK_CORRENTE PRIMARY KEY (numero_conta) REFERENCES CONTA (numero);
54 ALTER TABLE CORRENTE ADD CONSTRAINT FK_CORRENTE FOREIGN KEY (numero_conta) REFERENCES CONTA (numero) ON DELETE CASCADE;
55 ALTER TABLE POUPANCA ADD numero_conta NUMBER NOT NULL;
56 ALTER TABLE POUPANCA ADD CONSTRAINT PK_POUPANCA PRIMARY KEY (numero_conta) REFERENCES CONTA (numero);
57 ALTER TABLE POUPANCA ADD CONSTRAINT FK_POUPANCA FOREIGN KEY (numero_conta) REFERENCES CONTA (numero) ON DELETE CASCADE;
58
59 CREATE OR REPLACE TRIGGER genspecTrigger_CONTA AFTER INSERT OR DELETE OR UPDATE ON CONTA
60 REFERENCING NEW AS n OLD as o FOR EACH ROW
61 DECLARE X0 number; X1 number;
62 BEGIN
63 IF INSERTING THEN
64 SELECT COUNT(*) INTO X0 FROM CORRENTE c WHERE c.numero_conta = :n.numero;
65 SELECT COUNT(*) INTO X1 FROM POUPANCA c WHERE c.numero_conta = :n.numero;
66 IF (X0 + X1 < 1) THEN RAISE_APPLICATION_ERROR(-20000, 'Violação detectada!'); END IF;
67 END IF;
68 END;
69
70 CREATE OR REPLACE TRIGGER genspecTrigger_CORRENTE AFTER INSERT OR DELETE OR UPDATE ON CORRENTE
71 REFERENCING NEW AS n OLD as o FOR EACH ROW
72 DECLARE X0 number; X1 number;
73 BEGIN
```

```

74 IF INSERTING THEN
75     SELECT COUNT(*) INTO X1 FROM POUPANCA c WHERE c.numero_conta = :n.numero_conta;
76     IF (X1 != 0) THEN RAISE_APPLICATION_ERROR(-20000, 'Violacao detectada!'); END IF;
77 END IF;
78 IF DELETING THEN
79     SELECT COUNT(*) INTO X0 FROM CONTA c WHERE c.numero = :o.numero_conta;
80     IF (X0 != 0) THEN RAISE_APPLICATION_ERROR(-20001, 'Violacao detectada!'); END IF;
81 END IF;
82 IF UPDATING THEN
83     IF (:n.numero_conta != :o.numero_conta) THEN
84         SELECT COUNT(*) INTO X0 FROM CONTA c WHERE c.numero = :o.numero_conta;
85         IF (X0 != 0) THEN RAISE_APPLICATION_ERROR(-20002, 'Violacao detectada!'); END IF;
86         SELECT COUNT(*) INTO X1 FROM POUPANCA c WHERE c.numero_conta = :n.numero_conta;
87         IF (X1 != 0) THEN RAISE_APPLICATION_ERROR(-20003, 'Violacao detectada!'); END IF;
88     END IF;
89 END IF;
90 END;
91
92 CREATE OR REPLACE TRIGGER genspecTrigger_POUPANCA AFTER INSERT OR DELETE OR UPDATE ON POUPANCA
93 REFERENCING NEW AS n OLD as o FOR EACH ROW
94 DECLARE X0 number; X1 number;
95 BEGIN
96     IF INSERTING THEN
97         SELECT COUNT(*) INTO X1 FROM CORRENTE c WHERE c.numero_conta = :n.numero_conta;
98         IF (X1 != 0) THEN RAISE_APPLICATION_ERROR(-20004, 'Violacao detectada!'); END IF;
99     END IF;
100     IF DELETING THEN
101         SELECT COUNT(*) INTO X0 FROM CONTA c WHERE c.numero = :o.numero_conta;
102         IF (X0 != 0) THEN RAISE_APPLICATION_ERROR(-20005, 'Violacao detectada!'); END IF;
103     END IF;
104     IF UPDATING THEN
105         IF (:n.numero_conta != :o.numero_conta) THEN
106             SELECT COUNT(*) INTO X0 FROM CONTA c WHERE c.numero = :o.numero_conta;
107             IF (X0 != 0) THEN RAISE_APPLICATION_ERROR(-20006, 'Violacao detectada!'); END IF;
108             SELECT COUNT(*) INTO X1 FROM CORRENTE c WHERE c.numero_conta = :n.numero_conta;
109             IF (X1 != 0) THEN RAISE_APPLICATION_ERROR(-20007, 'Violacao detectada!'); END IF;
110         END IF;
111     END IF;
112 END;
113 -- Fim da Regra de Mapeamento #18
114
115 -- Início da Regra de Mapeamento #6
116 ALTER TABLE AGENCIA ADD id_gerente NUMBER NOT NULL;
117 ALTER TABLE AGENCIA ADD CONSTRAINT FK_AGENCIA FOREIGN KEY (id_gerente) REFERENCES GERENTE (id);
118 ALTER TABLE AGENCIA ADD CONSTRAINT UQ_AGENCIA UNIQUE (id_gerente);
119 -- Fim da Regra de Mapeamento #6
120
121 -- Início da Regra de Mapeamento #8
122 ALTER TABLE CONTA ADD codigo_agencia NUMBER;
123 ALTER TABLE CONTA ADD CONSTRAINT FK_CONTA FOREIGN KEY (codigo_agencia) REFERENCES AGENCIA (codigo);
124 -- Fim da Regra de Mapeamento #8
125
126 -- Início da Regra de Mapeamento #16
127 CREATE TABLE CARTAO (cpf_correntista NUMBER NOT NULL, numero_conta NUMBER NOT NULL);
128 ALTER TABLE CARTAO ADD CONSTRAINT PK_CARTAO PRIMARY KEY (cpf_correntista, numero_conta);
129 ALTER TABLE CARTAO ADD CONSTRAINT FK_CARTAO FOREIGN KEY (cpf_correntista) REFERENCES CORRENTISTA (cpf);
130 ALTER TABLE CARTAO ADD CONSTRAINT FK2_CARTAO FOREIGN KEY (numero_conta) REFERENCES CONTA (numero);
131 ALTER TABLE CARTAO ADD numero NUMBER NOT NULL;
132 ALTER TABLE CARTAO ADD senha NUMBER NOT NULL;
133 -- Fim da Regra de Mapeamento #16
134
135 -- Início da Regra de Mapeamento #21
136 CREATE TABLE CONTA_MOVIMENTACOES (
137     numero_conta NUMBER NOT NULL,
138     pk_movimentacoes NUMBER NOT NULL,
139     movimentacoes VARCHAR2(128) NOT NULL
140 );
141
142 ALTER TABLE CONTA_MOVIMENTACOES ADD CONSTRAINT PK_CONTA_MOVIMENTACOES PRIMARY KEY (pk_movimentacoes);
143 ALTER TABLE CONTA_MOVIMENTACOES ADD CONSTRAINT FK_CONTA_MOVIMENTACOES FOREIGN KEY (numero_conta) REFERENCES CONTA (
144     numero);
145 -- Fim da Regra de Mapeamento #21
146
147 -- Início da Regra de Mapeamento #22
148 CREATE OR REPLACE VIEW VW_AGENCIA AS (
149     SELECT a.codigo, COUNT (c.numero) AS qtde_contas FROM AGENCIA a, CONTA c WHERE c.codigo_agencia = a.codigo GROUP BY
150     a.codigo
151 );
152 -- Fim da Regra de Mapeamento #22

```

## Listagem 1. DDL do Modelo Bancário