

```
CREATE TABLE tb_cliente
```

```
(
    id_cliente      integer IDENTITY ( 1,1 ),
    nome_fantasia    varchar(60) NULL ,
    pessoa_fisica_juridica char(1) NULL ,
    razao_social     varchar(60) NULL ,
    cpf              integer NULL ,
    cnpj              integer NULL ,
    inscricao_estadual varchar(30) NULL ,
    telefone         varchar(13) NULL ,
    email            varchar(50) NULL ,
    rua_endereco     varchar(50) NULL ,
    num_endereco     char(10) NULL ,
    bairro_endereco  char(40) NULL ,
    cidade_endereco  varchar(50) NULL ,
    estado_endereco  char(2) NULL ,
    cep_endereco     char(8) NULL ,
    compl_endereco   varchar(30) NULL ,
    instagram        varchar(20) NULL ,
    site             varchar(100) NULL ,
    dt_abertura      date NULL ,
    status_cadastro  char(10) NULL ,
    des_banco        varchar(30) NULL ,
    agencia_banco    varchar(15) NULL ,
    conta_banco      varchar(15) NULL ,
    chave_pix        varchar(50) NULL ,
    flg_consentimento boolean NULL ,
    dt_hr_inclusao   timestamp NULL ,
    dt_hr_alteracao  timestamp NULL
)
go
```

```
ALTER TABLE tb_cliente
```

```
ADD CONSTRAINT XPKtb_cliente PRIMARY KEY CLUSTERED (id_cliente ASC)
go
```

```
CREATE TABLE tb_cliente_atacado
```

```
(
    id_cliente      integer NOT NULL ,
    dt_hr_inclusao   timestamp NULL ,
    dt_hr_alteracao  timestamp NULL ,
    dtassinatura     date NULL ,
    des_termos_contrato varchar(20) NULL ,
    seg_funcionamento boolean NULL ,
    ter_funcionamento boolean NULL ,
    qua_funcionamento boolean NULL ,
    qui_funcionamento boolean NULL ,
    sex_funcionamento boolean NULL ,
    sab_funcionamento boolean NULL ,
    dom_funcionamento boolean NULL ,
    seg_hr_abertura  time NULL ,
    ter_hr_abertura  time NULL ,
    qua_hr_abertura  time NULL ,
    qui_hr_abertura  time NULL ,
    sex_hr_abertura  time NULL ,
    sab_hr_abertura  time NULL ,
    dom_hr_abertura  time NULL ,
    seg_hr_fechamento time NULL ,
    ter_hr_fechamento time NULL ,
    qua_hr_fechamento time NULL ,
    qui_hr_fechamento time NULL ,
    sex_hr_fechamento time NULL ,
    sab_hr_fechamento time NULL ,
    dom_hr_fechamento time NULL
)
)
```

go

```
ALTER TABLE tb_cliente_atacado
    ADD CONSTRAINT XPKtb_cliente_atacado PRIMARYKEY CLUSTERED (id_cliente ASC)
```

go

```
CREATE TABLE tb_cliente_consentimento
(
    id_cliente      integer NOT NULL ,
    dt_consentimento date NOT NULL ,
    meio_consentimento varchar(30) NULL ,
    dt_hr_inclusao   timestamp NULL ,
    dt_hr_alteracao   timestamp NULL
)
```

go

```
ALTER TABLE tb_cliente_consentimento
    ADD CONSTRAINT XPKtb_cliente_consentimento PRIMARYKEY CLUSTERED (id_cliente ASC,dt_consentimento ASC)
```

go

```
CREATE TABLE tb_cliente_data_comemorativa
(
    dt_comemorativa date NULL ,
    des_data_comemorativa varchar(30) NULL ,
    id_cliente      integer NOT NULL ,
    id_data_comemorativa integer NOT NULL ,
    dt_hr_inclusao   timestamp NULL ,
    dt_hr_alteracao   timestamp NULL
)
```

go

```
ALTER TABLE tb_cliente_data_comemorativa
    ADD CONSTRAINT XPKtb_cliente_data_comemorativa PRIMARYKEY CLUSTERED (id_cliente ASC,id_data_comemorativa ASC)
```

go

```
CREATE TABLE tb_cliente_empresa
(
    id_cliente      integer NOT NULL ,
    id_empresa      integer NOT NULL ,
    dt_hr_inclusao   timestamp NULL ,
    dt_hr_alteracao   timestamp NULL
)
```

go

```
ALTER TABLE tb_cliente_empresa
    ADD CONSTRAINT XPKtb_cliente_empresa PRIMARYKEY CLUSTERED (id_cliente ASC,id_empresa ASC)
```

go

```
CREATE TABLE tb_cliente_representante
(
    id_cliente      integer NOT NULL ,
    cidade_atuacao   varchar(20) NOT NULL ,
    bairro_atuacao   varchar(20) NOT NULL ,
    dt_hr_inclusao   timestamp NULL ,
```

```
        dt_hr_alteracao    timestamp NULL ,
        dt_inicio_vigencia date NOT NULL ,
        dt_fim_vigencia    date NULL ,
        des_termos_contrato text NULL ,
        dtassinatura       date NULL
    )
go
```

```
ALTER TABLE tb_cliente_representante
    ADD CONSTRAINT XPKtb_cliente_representante PRIMARY KEY CLUSTERED (id_cliente ASC, cidade_atuacao ASC, bairro_atuacao ASC, dt_inicio_vigencia ASC)
go
```

```
CREATE TABLE tb_condicao_pagamento
(
    id_condicao      integer IDENTITY ( 1,1 ) ,
    cod_condicao      varchar(20) NULL ,
    des_condicao      varchar(255) NULL ,
    qt_parcelas      integer NULL ,
    tp_pagamento     varchar(20) NULL ,
    status           boolean NULL ,
    dt_hr_inclusao    timestamp NULL ,
    dt_hr_alteracao   timestamp NULL
)
go
```

```
ALTER TABLE tb_condicao_pagamento
    ADD CONSTRAINT XPKtb_condicao_pagamento PRIMARY KEY CLUSTERED (id_condicao ASC)
go
```

```
ALTER TABLE tb_condicao_pagamento
    ADD CONSTRAINT XAK1tb_condicao_pagamento UNIQUE (cod_condicao ASC)
go
```

```
CREATE TABLE tb_condicao_pagamento_parcela
(
    id_parcela       integer IDENTITY ( 1,1 ) ,
    id_condicao        integer NOT NULL ,
    num_parcela       integer NULL ,
    vlr_percentual     decimal(5,2) NULL ,
    dias_vencimento    integer NULL ,
    dt_hr_inclusao     timestamp NULL ,
    dt_hr_alteracao    timestamp NULL
)
go
```

```
ALTER TABLE tb_condicao_pagamento_parcela
    ADD CONSTRAINT XPKtb_condicao_pagamento_parcela PRIMARY KEY CLUSTERED (id_parcela ASC, id_condicao ASC)
go
```

```
CREATE TABLE tb_data_comemorativa
(
    id_data_comemorativa integer IDENTITY ( 1,1 ) ,
    des_data              varchar(40) NULL ,
    dt_hr_inclusao        timestamp NULL ,
    dt_hr_alteracao       timestamp NULL ,
    dt_comemorativa       date NULL
)
```

```
)  
go
```

```
ALTER TABLE tb_data_comemorativa  
    ADD CONSTRAINT XPKtb_data_comemorativa PRIMARY KEY CLUSTERED (id_data_comemorativa ASC)  
go
```

```
CREATE TABLE tb_data_especial_empresa  
(  
    id_data_especial integer NOT NULL ,  
    id_empresa integer NOT NULL ,  
    dt_hr_inclusao timestamp NULL ,  
    dt_hr_alteracao timestamp NULL  
)  
go
```

```
ALTER TABLE tb_data_especial_empresa  
    ADD CONSTRAINT XPKtb_data_especial_empresa PRIMARY KEY CLUSTERED (id_data_especial ASC,id_empresa ASC)  
go
```

```
CREATE TABLE tb_datas_especiais  
(  
    id_data_especial integer IDENTITY ( 1,1 ) ,  
    des_data_especial varchar(50) NULL ,  
    dt_hr_inclusao timestamp NULL ,  
    dt_hr_alteracao timestamp NULL  
)  
go
```

```
ALTER TABLE tb_datas_especiais  
    ADD CONSTRAINT XPKtb_datas_especiais PRIMARY KEY CLUSTERED (id_data_especial ASC)  
go
```

```
CREATE TABLE tb_empresa  
(  
    id_empresa integer IDENTITY ( 1,1 ) ,  
    nome_fantasia varchar(60) NULL ,  
    razao_social varchar(60) NULL ,  
    cnpj integer NULL ,  
    inscricao_estadual varchar(30) NULL ,  
    regime_tributario varchar(20) NULL ,  
    telefone varchar(13) NULL ,  
    email varchar(50) NULL ,  
    instagram varchar(20) NULL ,  
    site varchar(100) NULL ,  
    dt_abertura date NULL ,  
    status varchar(10) NULL ,  
    dt_hr_inclusao timestamp NULL ,  
    dt_hr_alteracao timestamp NULL  
)  
go
```

```
ALTER TABLE tb_empresa  
    ADD CONSTRAINT XPKtb_empresa PRIMARY KEY CLUSTERED (id_empresa ASC)  
go
```

```
CREATE TABLE tb_estrutura_dados_sensiveis
(
    id_estrutura_dados integer IDENTITY ( 1,1 ) ,
    dt_hr_inclusao timestamp NULL ,
    dt_hr_alteracao timestamp NULL ,
    des_tabela varchar(20) NULL
)
go
```

```
ALTER TABLE tb_estrutura_dados_sensiveis
    ADD CONSTRAINT XPk1tb_estrutura_dados_sensiveis PRIMARYKEY CLUSTERED (id_estrutura_dados ASC)
go
```

```
ALTER TABLE tb_estrutura_dados_sensiveis
    ADD CONSTRAINT XAK1tb_estrutura_dados_sensiveis UNIQUE (des_tabela ASC)
go
```

```
CREATE TABLE tb_estrutura_dados_sensiveis_campos
(
    id_estrutura_campos integer IDENTITY ( 1,1 ) ,
    id_estrutura_dados integer NOT NULL ,
    des_campo varchar(20) NULL ,
    dt_hr_inclusao timestamp NULL ,
    dt_hr_alteracao timestamp NULL
)
go
```

```
ALTER TABLE tb_estrutura_dados_sensiveis_campos
    ADD CONSTRAINT XPk1tb_estrutura_dados_sensiveis_campos PRIMARY KEY CLUSTERED (id_estrutura_campos ASC,id_estrutura_dados ASC)
go
```

```
CREATE TABLE tb_estrutura_sistema
(
    id_estrutura_sistema integer IDENTITY ( 1,1 ) ,
    cod_tela varchar(10) NULL ,
    des_tela varchar(50) NULL ,
    dt_hr_inclusao timestamp NULL ,
    dt_hr_alteracao timestamp NULL ,
    id_modulo integer NOT NULL
)
go
```

```
ALTER TABLE tb_estrutura_sistema
    ADD CONSTRAINT XPk1tb_estrutura_sistema PRIMARYKEY CLUSTERED (id_estrutura_sistema ASC,id_modulo ASC)
go
```

```
ALTER TABLE tb_estrutura_sistema
    ADD CONSTRAINT XAK1tb_estrutura_sistema UNIQUE (cod_tela ASC)
go
```

```
CREATE TABLE tb_filial
(
```

```

id_filial      char(18) NOT NULL ,
id_empresa    integer NOT NULL ,
nome_fantasia  varchar(60) NULL ,
razao_social   varchar(60) NULL ,
cnpj          integer NULL ,
inscricao_estadual varchar(30) NULL ,
rua_endereco   varchar(50) NULL ,
num_endereco   char(10) NULL ,
bairro_endereco char(40) NULL ,
cidade_endereco varchar(50) NULL ,
estado_endereco char(2) NULL ,
cep_endereco   char(8) NULL ,
compl_endereco varchar(30) NULL ,
telefone       varchar(13) NULL ,
nome_responsavel varchar(50) NULL ,
num_telefone_responsavel char(13) NULL ,
seg_hr_abertura time NULL ,
seg_hr_fechamento time NULL ,
status         varchar(10) NULL ,
dt_abertura    date NULL ,
seg_funcionamento boolean NULL ,
ter_funcionamento boolean NULL ,
qua_funcionamento boolean NULL ,
qui_funcionamento boolean NULL ,
sex_funcionamento boolean NULL ,
sab_funcionamento boolean NULL ,
dom_funcionamento boolean NULL ,
ter_hr_abertura time NULL ,
qua_hr_abertura time NULL ,
qui_hr_abertura time NULL ,
sex_hr_abertura time NULL ,
sab_hr_abertura time NULL ,
dom_hr_abertura time NULL ,
ter_hr_fechamento time NULL ,
qua_hr_fechamento time NULL ,
qui_hr_fechamento time NULL ,
sex_hr_fechamento time NULL ,
sab_hr_fechamento time NULL ,
dom_hr_fechamento time NULL ,
email         varchar(50) NULL ,
email_responsavel varchar(50) NULL ,
dt_hr_inclusao timestamp NULL ,
dt_hr_alteracao timestamp NULL
)
go

```

```

ALTER TABLE tb_filial
ADD CONSTRAINT XPKtb_filial PRIMARY KEY CLUSTERED (id_filial ASC,id_empresa ASC)
go

```

```

CREATE TABLE tb_log_auditoria
(
    id_log      integer IDENTITY ( 1,1 ) ,
    tp_acao     varchar(10) NULL ,
    nome_tabela varchar(20) NULL ,
    dados_anteriores varchar(60) NULL ,
    dados_novos  varchar(60) NULL ,
    dt_hr_acao   datetime NULL ,
    ip_origem    varchar(15) NULL ,
    des_justificativa varchar(100) NULL ,
    campo_afetado  varchar(50) NULL ,
    chave_registro varchar(30) NULL ,
    dt_hr_inclusao timestamp NULL ,
    dt_hr_alteracao timestamp NULL ,
    id_usuario    integer NULL
)

```

go

```
ALTER TABLE tb_log_auditoria
    ADD CONSTRAINT XPKtb_log_auditoria PRIMARY KEY CLUSTERED (id_log ASC)
```

go

```
CREATE TABLE tb_meio_condicao_pagamento
(
    id_condicao      integer NOT NULL ,
    id_meio_pagamento integer NOT NULL ,
    dt_hr_inclusao   timestamp NULL ,
    dt_hr_alteracao  timestamp NULL ,
    id_tipo_cliente  integer NOT NULL
)
```

go

```
ALTER TABLE tb_meio_condicao_pagamento
    ADD CONSTRAINT XPKtb_meio_condicao_pagamento PRIMARY KEY CLUSTERED (id_condicao ASC,id_meio_pagamento ASC,id_tipo_cliente ASC)
```

go

```
CREATE TABLE tb_meio_pagamento
(
    id_meio_pagamento integer IDENTITY ( 1,1 ) ,
    des_meio_pagamento varchar(50) NULL ,
    dt_hr_inclusao      timestamp NULL ,
    dt_hr_alteracao     timestamp NULL
)
```

go

```
ALTER TABLE tb_meio_pagamento
    ADD CONSTRAINT XPKtb_meio_pagamento PRIMARY KEY CLUSTERED (id_meio_pagamento ASC)
```

go

```
CREATE TABLE tb_modulos
(
    id_modulo      integer IDENTITY ( 1,1 ) ,
    des_modulo      varchar(30) NULL ,
    dt_hr_inclusao   timestamp NULL ,
    dt_hr_alteracao  timestamp NULL
)
```

go

```
ALTER TABLE tb_modulos
    ADD CONSTRAINT XPKtb_modulos PRIMARY KEY CLUSTERED (id_modulo ASC)
```

go

```
CREATE TABLE tb_produto
(
    id_produto      integer IDENTITY ( 1,1 ) ,
    des_produto      varchar(60) NULL ,
    vlr_peso         integer NULL ,
    num_ean          char(14) NULL ,
    status           varchar(10) NULL ,
    vlr_validade_dias_ambiente integer NULL ,
```

```
        vlr_validade_dias_refrigerado integer NULL ,
        vlr_validade_dias_congelador integer NULL ,
        dt_hr_inclusao    timestamp NULL ,
        dt_hr_alteracao   timestamp NULL ,
        id_tipo_produto   char(18) NOT NULL ,
        id_unidade_medida integer NULL
    )
go
```

```
ALTER TABLE tb_produto
    ADD CONSTRAINT XPKtb_produto PRIMARY KEY CLUSTERED (id_produto ASC)
go
```

```
CREATE TABLE tb_produto_acabado_estoque
(
    id_poduto_acabado_estoque integer IDENTITY ( 1,1 ) ,
    qt_estoque                integer NULL ,
    id_produto_acabado        integer NULL ,
    id_unidade_medida         integer NULL ,
    dt_hr_inclusao            timestamp NULL ,
    dt_hr_alteracao           timestamp NULL
)
go
```

```
ALTER TABLE tb_produto_acabado_estoque
    ADD CONSTRAINT XPKtb_produto_acabado_estoque PRIMARY KEY CLUSTERED (id_poduto_acabado_estoque ASC)
go
```

```
CREATE TABLE tb_produto_acabado_movimentacao_estoque
(
    id_movimentacao    integer IDENTITY ( 1,1 ) ,
    tp_movimentacao    varchar(10) NULL ,
    qt_movimentacao    integer NULL ,
    id_poduto_acabado_estoque integer NOT NULL ,
    id_tipo_movimentacao_estoque integer NULL ,
    dt_movimentacao    date NULL ,
    dt_hr_inclusao     datetime NULL ,
    dt_hr_alteracao    timestamp NULL
)
go
```

```
ALTER TABLE tb_produto_acabado_movimentacao_estoque
    ADD CONSTRAINT XPKtb_produto_acabado_movimentacao_estoque PRIMARY KEY CLUSTERED (id_movimentacao ASC)
go
```

```
CREATE TABLE tb_produto_ficha_tecnica
(
    id_ficha_tecnica    integer IDENTITY ( 1,1 ) ,
    qt_insumo           decimal(10,3) NULL ,
    dt_hr_inclusao      timestamp NULL ,
    dt_hr_alteracao     timestamp NULL ,
    id_produto_acabado  integer NULL ,
    id_produto_insumo   integer NULL ,
    id_unidade_medida   integer NULL
)
go
```



```
ALTER TABLE tb_produto_ficha_tecnica
    ADD CONSTRAINT XPKtb_produto_ficha_tecnica PRIMARY KEY CLUSTERED (id_ficha_tecnica ASC)
go
```

```
CREATE TABLE tb_produto_insumo_estoque
(
    id_produto_insumo_estoque char(18) NOT NULL ,
    id_produto_insumo integer NULL ,
    id_unidade_medida integer NULL ,
    qt_estoque decimal(10,2) NULL ,
    dt_hr_inclusao timestamp NULL ,
    dt_hr_alteracao timestamp NULL
)
go
```

```
ALTER TABLE tb_produto_insumo_estoque
    ADD CONSTRAINT XPKtb_produto_insumo_estoque PRIMARY KEY CLUSTERED (id_produto_insumo_estoque ASC)
go
```

```
CREATE TABLE tb_produto_insumo_movimentacao_estoque
(
    id_movimentacao integer IDENTITY ( 1,1 ) ,
    id_produto_insumo_estoque char(18) NULL ,
    id_tipo_movimentacao_estoque integer NULL ,
    tp_movimentacao varchar(10) NULL ,
    qt_movimentacao decimal(10,2) NULL ,
    dt_movimentacao date NULL ,
    dt_hr_inclusao timestamp NULL ,
    dt_hr_alteracao timestamp NULL
)
go
```

```
ALTER TABLE tb_produto_insumo_movimentacao_estoque
    ADD CONSTRAINT XPKtb_produto_insumo_movimentacao_estoque PRIMARY KEY CLUSTERED (id_movimentacao ASC)
go
```

```
CREATE TABLE tb_produto_roteiro_producao
(
    id_roteiro integer IDENTITY ( 1,1 ) ,
    nr_sequencia integer NULL ,
    des_etapa varchar(255) NULL ,
    tempo_estimado integer NULL ,
    dt_hr_inclusao timestamp NULL ,
    dt_hr_alteracao timestamp NULL ,
    id_produto_acabado integer NULL
)
go
```

```
ALTER TABLE tb_produto_roteiro_producao
    ADD CONSTRAINT XPKtb_produto_roteiro_producao PRIMARY KEY CLUSTERED (id_roteiro ASC)
go
```

```
CREATE TABLE tb_produto_tabela_preco
(
    id_produto_acabado integer NOT NULL ,
```

```
        id_tabela_preco integer NOT NULL ,
        id_tipo_cliente integer NOT NULL ,
        dt_hr_inclusao timestamp NULL ,
        dt_hr_alteracao timestamp NULL ,
        vlr_preco decimal(10,2) NULL
    )
go
```

```
ALTER TABLE tb_produto_tabela_preco
    ADD CONSTRAINT XPKtb_produto_tabela_preco PRIMARYKEY CLUSTERED (id_produto_acabado ASC,id_tabela_preco ASC,id_tipo_cliente ASC)
go
```

```
CREATE TABLE tb_tabela_preco
(
    id_tabela_preco integer IDENTITY ( 1,1 ) ,
    des_tabela_preco varchar(60) NULL ,
    id_tipo_cliente integer NOT NULL ,
    dt_inicio_vigencia date NULL ,
    dt_fim_vigencia date NULL ,
    dt_hr_inclusao timestamp NULL ,
    dt_hr_alteracao timestamp NULL ,
    status varchar(10) NULL
)
go
```

```
ALTER TABLE tb_tabela_preco
    ADD CONSTRAINT XPKtb_tabela_preco PRIMARYKEY CLUSTERED (id_tabela_preco ASC,id_tipo_cliente ASC)
go
```

```
CREATE TABLE tb_tabela_preco_cliente
(
    id_tabela_preco integer NOT NULL ,
    id_tipo_cliente integer NOT NULL ,
    id_cliente integer NOT NULL ,
    dt_hr_inclusao timestamp NULL ,
    dt_hr_alteracao timestamp NULL
)
go
```

```
ALTER TABLE tb_tabela_preco_cliente
    ADD CONSTRAINT XPKtb_tabela_preco_cliente PRIMARYKEY CLUSTERED (id_tabela_preco ASC,id_tipo_cliente ASC,id_cliente ASC)
go
```

```
CREATE TABLE tb_tipo_cliente
(
    id_tipo_cliente integer IDENTITY ( 1,1 ) ,
    des_tipo_cliente varchar(20) NULL ,
    dt_hr_inclusao timestamp NULL ,
    dt_hr_alteracao timestamp NULL
)
go
```

```
ALTER TABLE tb_tipo_cliente
    ADD CONSTRAINT XPKtb_tipo_cliente PRIMARYKEY CLUSTERED (id_tipo_cliente ASC)
go
```

```
CREATE TABLE tb_tipo_cliente_cliente
(
    id_tipo_cliente integer NOT NULL,
    id_cliente       integer NOT NULL,
    dt_hr_inclusao   timestamp NULL,
    dt_hr_alteracao   timestamp NULL
)
go
```

```
ALTER TABLE tb_tipo_cliente_cliente
    ADD CONSTRAINT XPKtb_tipo_cliente_cliente PRIMARYKEY CLUSTERED (id_tipo_cliente ASC,id_cliente ASC)
go
```

```
CREATE TABLE tb_tipo_cliente_condicao
(
    id_tipo_cliente integer NOT NULL,
    id_condicao       integer NOT NULL,
    dt_hr_inclusao   timestamp NULL,
    dt_hr_alteracao   timestamp NULL
)
go
```

```
ALTER TABLE tb_tipo_cliente_condicao
    ADD CONSTRAINT XPKtb_tipo_cliente_condicao PRIMARYKEY CLUSTERED (id_tipo_cliente ASC,id_condicao ASC)
go
```

```
CREATE TABLE tb_tipo_cliente_meio_condicao
(
    id_tipo_cliente integer NOT NULL,
    id_cliente       integer NOT NULL,
    id_condicao       integer NOT NULL,
    id_meio_pagamento integer NOT NULL,
    dt_hr_inclusao   timestamp NULL,
    dt_hr_alteracao   timestamp NULL
)
go
```

```
ALTER TABLE tb_tipo_cliente_meio_condicao
    ADD CONSTRAINT XPKtb_tipo_cliente_meio_condicao PRIMARY KEY CLUSTERED (id_tipo_cliente ASC,id_cliente ASC,id_condicao ASC,id_meio_pagamento ASC)
go
```

```
CREATE TABLE tb_tipo_movimentacao_estoque
(
    id_tipo_movimentacao_estoque integer IDENTITY ( 1,1 ),
    des_tipo_movimentacao_estoque varchar(60) NULL,
    dt_hr_inclusao   timestamp NULL,
    dt_hr_alteracao   timestamp NULL
)
go
```

```
ALTER TABLE tb_tipo_movimentacao_estoque
    ADD CONSTRAINT XPKtb_tipo_movimentacao_estoque PRIMARYKEY CLUSTERED (id_tipo_movimentacao_estoque ASC)
go
```

```
CREATE TABLE tb_tipo_produto
(
    id_tipo_produto char(18) NOT NULL,
    des_tipo_produto char(18) NULL,
    dt_hr_inclusao timestamp NULL,
    dt_hr_alteracao timestamp NULL
)
go
```

```
ALTER TABLE tb_tipo_produto
    ADD CONSTRAINT XPKtb_tipo_produto PRIMARY KEY CLUSTERED (id_tipo_produto ASC)
go
```

```
CREATE TABLE tb_unidade_medida
(
    id_unidade_medida integer IDENTITY ( 1,1 ),
    des_unidade_medida varchar(30) NULL,
    dt_hr_inclusao timestamp NULL,
    dt_hr_alteracao timestamp NULL
)
go
```

```
ALTER TABLE tb_unidade_medida
    ADD CONSTRAINT XPKtb_unidade_medida PRIMARY KEY CLUSTERED (id_unidade_medida ASC)
go
```

```
CREATE TABLE tb_usuario
(
    id_usuario integer IDENTITY ( 1,1 ),
    nome varchar(60) NULL,
    email varchar(50) NULL,
    telefone varchar(13) NULL,
    senha_hash text NULL,
    status varchar(10) NULL,
    dt_hr_inclusao timestamp NULL,
    dt_hr_alteracao timestamp NULL,
    dt_validade_senha date NULL,
    dt_hr_ultimo_acesso timestamp NULL,
    usuario varchar(20) NULL
)
go
```

```
ALTER TABLE tb_usuario
    ADD CONSTRAINT XPKtb_usuario PRIMARY KEY CLUSTERED (id_usuario ASC)
go
```

```
CREATE TABLE tb_usuario_estrutura_dados
(
    id_estrutura_campos integer NOT NULL,
    id_estrutura_dados integer NOT NULL,
    id_usuario integer NOT NULL,
    fig_nanomizado boolean NULL,
    fig_alterar boolean NULL,
    fig_incluir boolean NULL,
    fig_excluir boolean NULL,
    fig_visualizar boolean NULL,

```

```
        dt_hr_inclusao    timestamp NULL ,
        dt_hr_alteracao   timestamp NULL ,
        flg_administrador boolean NULL
    )
go
```

```
ALTER TABLE tb_usuario_estrutura_dados
    ADD CONSTRAINT XPKtb_usuario_estrutura_dados PRIMARY KEY CLUSTERED (id_estrutura_campos ASC,id_estrutura_dados ASC,id_usuario ASC)
go
```

```
CREATE TABLE tb_usuario_estrutura_sistema
(
    id_estrutura_sistema integer NOT NULL ,
    id_modulo             integer NOT NULL ,
    id_usuario            integer NOT NULL ,
    flg_vizualizar        boolean NULL ,
    flg_alterar           boolean NULL ,
    flg_incluir           boolean NULL ,
    flg_excluir           boolean NULL ,
    flg_processar         boolean NULL ,
    dt_hr_inclusao        timestamp NULL ,
    dt_hr_alteracao       timestamp NULL ,
    flg_administrador     boolean NULL
)
go
```

```
ALTER TABLE tb_usuario_estrutura_sistema
    ADD CONSTRAINT XPKtb_usuario_estrutura_sistema PRIMARY KEY CLUSTERED (id_estrutura_sistema ASC,id_modulo ASC,id_usuario ASC)
go
```

```
ALTER TABLE tb_cliente_atacado
    ADD CONSTRAINT R_19 FOREIGN KEY (id_cliente) REFERENCES tb_cliente(id_cliente)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION
go
```

```
ALTER TABLE tb_cliente_consentimento
    ADD CONSTRAINT R_7 FOREIGN KEY (id_cliente) REFERENCES tb_cliente(id_cliente)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION
go
```

```
ALTER TABLE tb_cliente_data_comemorativa
    ADD CONSTRAINT R_5 FOREIGN KEY (id_cliente) REFERENCES tb_cliente(id_cliente)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION
go
```

```
ALTER TABLE tb_cliente_data_comemorativa
    ADD CONSTRAINT R_6 FOREIGN KEY (id_data_comemorativa) REFERENCES tb_data_comemorativa(id_data_comemorativa)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION
```

go

```
ALTER TABLE tb_cliente_empresa
    ADD CONSTRAINT R_3 FOREIGN KEY (id_cliente) REFERENCES tb_cliente(id_cliente)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
```

go

```
ALTER TABLE tb_cliente_empresa
    ADD CONSTRAINT R_4 FOREIGN KEY (id_empresa) REFERENCES tb_empresa(id_empresa)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
```

go

```
ALTER TABLE tb_cliente_representante
    ADD CONSTRAINT R_18 FOREIGN KEY (id_cliente) REFERENCES tb_cliente(id_cliente)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
```

go

```
ALTER TABLE tb_condicao_pagamento_parcela
    ADD CONSTRAINT R_24 FOREIGN KEY (id_condicao) REFERENCES tb_condicao_pagamento(id_condicao)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
```

go

```
ALTER TABLE tb_data_especial_empresa
    ADD CONSTRAINT R_20 FOREIGN KEY (id_data_especial) REFERENCES tb_dados_especiais(id_data_especial)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
```

go

```
ALTER TABLE tb_data_especial_empresa
    ADD CONSTRAINT R_21 FOREIGN KEY (id_empresa) REFERENCES tb_empresa(id_empresa)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
```

go

```
ALTER TABLE tb_estrutura_dados_sensíveis_campos
    ADD CONSTRAINT R_9 FOREIGN KEY (id_estrutura_dados) REFERENCES tb_estrutura_dados_sensíveis(id_estrutura_dados)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
```

go

```
ALTER TABLE tb_estrutura_sistema
    ADD CONSTRAINT R_8 FOREIGN KEY (id_modulo) REFERENCES tb_modulos(id_modulo)
```

```
ON DELETE NO ACTION
ON UPDATE NO ACTION
```

go

```
ALTER TABLE tb_filial
ADD CONSTRAINT R_2 FOREIGN KEY (id_empresa) REFERENCES tb_empresa(id_empresa)
ON DELETE NO ACTION
ON UPDATE NO ACTION
```

go

```
ALTER TABLE tb_log_auditoria
ADD CONSTRAINT R_13 FOREIGN KEY (id_usuario) REFERENCES tb_usuario(id_usuario)
ON DELETE NO ACTION
ON UPDATE NO ACTION
```

go

```
ALTER TABLE tb_meio_condicao_pagamento
ADD CONSTRAINT R_25 FOREIGN KEY (id_condicao) REFERENCES tb_condicao_pagamento(id_condicao)
ON DELETE NO ACTION
ON UPDATE NO ACTION
```

go

```
ALTER TABLE tb_meio_condicao_pagamento
ADD CONSTRAINT R_26 FOREIGN KEY (id_meio_pagamento) REFERENCES tb_meio_pagamento(id_meio_pagamento)
ON DELETE NO ACTION
ON UPDATE NO ACTION
```

go

```
ALTER TABLE tb_meio_condicao_pagamento
ADD CONSTRAINT R_27 FOREIGN KEY (id_tipo_cliente) REFERENCES tb_tipo_cliente(id_tipo_cliente)
ON DELETE NO ACTION
ON UPDATE NO ACTION
```

go

```
ALTER TABLE tb_produto
ADD CONSTRAINT R_30 FOREIGN KEY (id_tipo_produto) REFERENCES tb_tipo_produto(id_tipo_produto)
ON DELETE NO ACTION
ON UPDATE NO ACTION
```

go

```
ALTER TABLE tb_produto
ADD CONSTRAINT R_31 FOREIGN KEY (id_unidade_medida) REFERENCES tb_unidade_medida(id_unidade_medida)
ON DELETE NO ACTION
ON UPDATE NO ACTION
```

go

```
ALTER TABLE tb_produto_acabado_estoque
    ADD CONSTRAINT R_41 FOREIGN KEY (id_produto_acabado) REFERENCES tb_produto(id_produto)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
```

go

```
ALTER TABLE tb_produto_acabado_estoque
    ADD CONSTRAINT R_42 FOREIGN KEY (id_unidade_medida) REFERENCES tb_unidade_medida(id_unidade_medida)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
```

go

```
ALTER TABLE tb_produto_acabado_movimentacao_estoque
    ADD CONSTRAINT R_43 FOREIGN KEY (id_produto_acabado_estoque) REFERENCES tb_produto_acabado_estoque(id_produto_acabado_estoque)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
```

go

```
ALTER TABLE tb_produto_acabado_movimentacao_estoque
    ADD CONSTRAINT R_44 FOREIGN KEY (id_tipo_movimentacao_estoque) REFERENCES tb_tipo_movimentacao_estoque(id_tipo_movimentacao_estoque)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
```

go

```
ALTER TABLE tb_produto_ficha_tecnica
    ADD CONSTRAINT R_32 FOREIGN KEY (id_produto_acabado) REFERENCES tb_produto(id_produto)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
```

go

```
ALTER TABLE tb_produto_ficha_tecnica
    ADD CONSTRAINT R_33 FOREIGN KEY (id_produto_insumo) REFERENCES tb_produto(id_produto)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
```

go

```
ALTER TABLE tb_produto_ficha_tecnica
    ADD CONSTRAINT R_34 FOREIGN KEY (id_unidade_medida) REFERENCES tb_unidade_medida(id_unidade_medida)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
```

go

```
ALTER TABLE tb_produto_insumo_estoque
    ADD CONSTRAINT R_45 FOREIGN KEY (id_produto_insumo) REFERENCES tb_produto(id_produto)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
```

go



```
ALTER TABLE tb_produto_insumo_estoque
    ADD CONSTRAINT R_46 FOREIGN KEY (id_unidade_medida) REFERENCES tb_unidade_medida(id_unidade_medida)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
go
```

```
ALTER TABLE tb_produto_insumo_movimentacao_estoque
    ADD CONSTRAINT R_47 FOREIGN KEY (id_produto_insumo_estoque) REFERENCES tb_produto_insumo_estoque(id_produto_insumo_estoque)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
go
```

```
ALTER TABLE tb_produto_insumo_movimentacao_estoque
    ADD CONSTRAINT R_48 FOREIGN KEY (id_tipo_movimentacao_estoque) REFERENCES tb_tipo_movimentacao_estoque(id_tipo_movimentacao_estoque)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
go
```

```
ALTER TABLE tb_produto_rotreiro_producao
    ADD CONSTRAINT R_35 FOREIGN KEY (id_produto_acabado) REFERENCES tb_produto(id_produto)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
go
```

```
ALTER TABLE tb_produto_tabela_preco
    ADD CONSTRAINT R_39 FOREIGN KEY (id_produto_acabado) REFERENCES tb_produto(id_produto)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
go
```

```
ALTER TABLE tb_produto_tabela_preco
    ADD CONSTRAINT R_40 FOREIGN KEY (id_tabela_preco,id_tipo_cliente) REFERENCES tb_tabela_preco(id_tabela_preco,id_tipo_cliente)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
go
```

```
ALTER TABLE tb_tabela_preco
    ADD CONSTRAINT R_36 FOREIGN KEY (id_tipo_cliente) REFERENCES tb_tipo_cliente(id_tipo_cliente)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
go
```

```
ALTER TABLE tb_tabela_preco_cliente
    ADD CONSTRAINT R_37 FOREIGN KEY (id_tabela_preco,id_tipo_cliente) REFERENCES tb_tabela_preco(id_tabela_preco,id_tipo_cliente)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
go
```

```
ALTER TABLE tb_tabela_preco_cliente
    ADD CONSTRAINT R_38 FOREIGN KEY (id_tipo_cliente,id_cliente) REFERENCES tb_tipo_cliente_cliente(id_tipo_cliente,id_cliente)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION
```

go

```
ALTER TABLE tb_tipo_cliente_cliente
    ADD CONSTRAINT R_16 FOREIGN KEY (id_tipo_cliente) REFERENCES tb_tipo_cliente(id_tipo_cliente)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION
```

go

```
ALTER TABLE tb_tipo_cliente_cliente
    ADD CONSTRAINT R_17 FOREIGN KEY (id_cliente) REFERENCES tb_cliente(id_cliente)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION
```

go

```
ALTER TABLE tb_tipo_cliente_condicao
    ADD CONSTRAINT R_22 FOREIGN KEY (id_tipo_cliente) REFERENCES tb_tipo_cliente(id_tipo_cliente)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION
```

go

```
ALTER TABLE tb_tipo_cliente_condicao
    ADD CONSTRAINT R_23 FOREIGN KEY (id_condicao) REFERENCES tb_condicao_pagamento(id_condicao)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION
```

go

```
ALTER TABLE tb_tipo_cliente_meio_condicao
    ADD CONSTRAINT R_28 FOREIGN KEY (id_tipo_cliente,id_cliente) REFERENCES tb_tipo_cliente_cliente(id_tipo_cliente,id_cliente)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION
```

go

```
ALTER TABLE tb_tipo_cliente_meio_condicao
    ADD CONSTRAINT R_29 FOREIGN KEY (id_condicao,id_meio_pagamento,id_tipo_cliente) REFERENCES tb_meio_condicao_pagamento(id_condicao,id_meio_pagamento,id_tipo_cliente)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION
```

go

```
ALTER TABLE tb_usuario_estrutura_dados
    ADD CONSTRAINT R_14 FOREIGN KEY (id_estrutura_campos,id_estrutura_dados) REFERENCES tb_estrutura_dados_sensíveis_campos(id_estrutura_campos,id_estrutura_dados)
        ON DELETE NO ACTION
```

ON UPDATE NO ACTION

go

```
ALTER TABLE tb_usuario_estrutura_dados
    ADD CONSTRAINT R_15 FOREIGN KEY (id_usuario) REFERENCES tb_usuario(id_usuario)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
```

go

```
ALTER TABLE tb_usuario_estrutura_sistema
    ADD CONSTRAINT R_10 FOREIGN KEY (id_estrutura_sistema,id_modulo) REFERENCES tb_estrutura_sistema(id_estrutura_sistema,id_modulo)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
```

go

```
ALTER TABLE tb_usuario_estrutura_sistema
    ADD CONSTRAINT R_11 FOREIGN KEY (id_usuario) REFERENCES tb_usuario(id_usuario)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
```

go

CREATE TRIGGER tD\_tb\_cliente ON tb\_cliente FOR DELETE AS

/\* ERwin Builtin Trigger \*/

/\* DELETE trigger on tb\_cliente \*/

BEGIN

DECLARE @errno int,

@errmsg varchar(255)

/\* ERwin Builtin Trigger \*/

/\* tb\_cliente tb\_cliente\_atacado on parent delete no action \*/

/\* ERWIN\_RELATION:CHECKSUM="00067f40", PARENT\_OWNER="", PARENT\_TABLE="tb\_cliente"

CHILD\_OWNER="", CHILD\_TABLE="tb\_cliente\_atacado"

P2C\_VERB\_PHRASE="", C2P\_VERB\_PHRASE="",

FK\_CONSTRAINT="R\_19", FK\_COLUMNS="id\_cliente" \*/

IF EXISTS (

SELECT \* FROM deleted,tb\_cliente\_atacado

WHERE

/\* %JoinFKPK(tb\_cliente\_atacado,deleted," = ","AND") \*/

tb\_cliente\_atacado.id\_cliente = deleted.id\_cliente

)

BEGIN

SELECT @errno = 30001,

@errmsg = 'Cannot delete tb\_cliente because tb\_cliente\_atacado exists.'

GOTO ERROR

END

/\* ERwin Builtin Trigger \*/

/\* tb\_cliente tb\_cliente\_representante on parent delete no action \*/

/\* ERWIN\_RELATION:CHECKSUM="00000000", PARENT\_OWNER="", PARENT\_TABLE="tb\_cliente"

CHILD\_OWNER="", CHILD\_TABLE="tb\_cliente\_representante"

P2C\_VERB\_PHRASE="", C2P\_VERB\_PHRASE="",

FK\_CONSTRAINT="R\_18", FK\_COLUMNS="id\_cliente" \*/

IF EXISTS (

SELECT \* FROM deleted,tb\_cliente\_representante

WHERE

/\* %JoinFKPK(tb\_cliente\_representante,deleted," = ","AND") \*/

tb\_cliente\_representante.id\_cliente = deleted.id\_cliente

)

BEGIN

```

SELECT @errno = 30001,
       @errmsg = 'Cannot delete tb_cliente because tb_cliente_representante exists.'
GOTO ERROR
END

```

```

/* ERwin Builtin Trigger */
/* tb_cliente tb_tipo_cliente_cliente on parent delete no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_cliente"
CHILD_OWNER="", CHILD_TABLE="tb_tipo_cliente_cliente"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_17", FK_COLUMNS="id_cliente" */
IF EXISTS (
  SELECT * FROM deleted,tb_tipo_cliente_cliente
  WHERE
    /* %JoinFKPK(tb_tipo_cliente_cliente,deleted,"=","AND") */
    tb_tipo_cliente_cliente.id_cliente = deleted.id_cliente
)
BEGIN
  SELECT @errno = 30001,
       @errmsg = 'Cannot delete tb_cliente because tb_tipo_cliente_cliente exists.'
  GOTO ERROR
END

```

```

/* ERwin Builtin Trigger */
/* tb_cliente tb_cliente_consentimento on parent delete no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_cliente"
CHILD_OWNER="", CHILD_TABLE="tb_cliente_consentimento"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_7", FK_COLUMNS="id_cliente" */
IF EXISTS (
  SELECT * FROM deleted,tb_cliente_consentimento
  WHERE
    /* %JoinFKPK(tb_cliente_consentimento,deleted,"=","AND") */
    tb_cliente_consentimento.id_cliente = deleted.id_cliente
)
BEGIN
  SELECT @errno = 30001,
       @errmsg = 'Cannot delete tb_cliente because tb_cliente_consentimento exists.'
  GOTO ERROR
END

```

```

/* ERwin Builtin Trigger */
/* tb_cliente tb_cliente_data_comemorativa on parent delete no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_cliente"
CHILD_OWNER="", CHILD_TABLE="tb_cliente_data_comemorativa"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_5", FK_COLUMNS="id_cliente" */
IF EXISTS (
  SELECT * FROM deleted,tb_cliente_data_comemorativa
  WHERE
    /* %JoinFKPK(tb_cliente_data_comemorativa,deleted,"=","AND") */
    tb_cliente_data_comemorativa.id_cliente = deleted.id_cliente
)
BEGIN
  SELECT @errno = 30001,
       @errmsg = 'Cannot delete tb_cliente because tb_cliente_data_comemorativa exists.'
  GOTO ERROR
END

```

```

/* ERwin Builtin Trigger */
/* tb_cliente tb_cliente_empresa on parent delete no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_cliente"
CHILD_OWNER="", CHILD_TABLE="tb_cliente_empresa"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_3", FK_COLUMNS="id_cliente" */
IF EXISTS (
  SELECT * FROM deleted,tb_cliente_empresa
  WHERE
    /* %JoinFKPK(tb_cliente_empresa,deleted,"=","AND") */
    tb_cliente_empresa.id_cliente = deleted.id_cliente
)

```

```

)
BEGIN
    SELECT @errno = 30001,
           @errmsg = 'Cannot delete tb_cliente because tb_cliente_empresa exists.'
    GOTO ERROR
END

```

```

/* ERwin Builtin Trigger */
RETURN
ERROR:
    raiserror @errno @errmsg
    rollback transaction
END

```

go

```

CREATE TRIGGER tU_tb_cliente ON tb_cliente FOR UPDATE AS

```

```

/* ERwin Builtin Trigger */
/* UPDATE trigger on tb_cliente */

```

```

BEGIN
    DECLARE @NUMROWS int,
            @nullcnt int,
            @validcnt int,
            @insid_cliente integer,
            @errno int,
            @errmsg varchar(255)

```

```

    SELECT @NUMROWS = @@rowcount

```

```

/* ERwin Builtin Trigger */
/* tb_cliente tb_cliente_atacado on parent update no action */
/* ERWIN_RELATION:CHECKSUM="000733fb", PARENT_OWNER="", PARENT_TABLE="tb_cliente"
   CHILD_OWNER="", CHILD_TABLE="tb_cliente_atacado"
   P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
   FK_CONSTRAINT="R_19", FK_COLUMNS="id_cliente" */
IF

```

```

    /* %ParentPK(" OR",UPDATE) */
    UPDATE(id_cliente)

```

```

BEGIN
    IF EXISTS (
        SELECT * FROM deleted,tb_cliente_atacado
        WHERE
            /* %JoinFKPK(tb_cliente_atacado,deleted," = ","AND") */
            tb_cliente_atacado.id_cliente = deleted.id_cliente
    )

```

```

BEGIN
    SELECT @errno = 30005,
           @errmsg = 'Cannot update tb_cliente because tb_cliente_atacado exists.'
    GOTO ERROR
END
END

```

```

/* ERwin Builtin Trigger */
/* tb_cliente tb_cliente_representante on parent update no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_cliente"
   CHILD_OWNER="", CHILD_TABLE="tb_cliente_representante"
   P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
   FK_CONSTRAINT="R_18", FK_COLUMNS="id_cliente" */
IF

```

```

    /* %ParentPK(" OR",UPDATE) */
    UPDATE(id_cliente)

```

```

BEGIN
    IF EXISTS (
        SELECT * FROM deleted,tb_cliente_representante
        WHERE
            /* %JoinFKPK(tb_cliente_representante,deleted," = ","AND") */
            tb_cliente_representante.id_cliente = deleted.id_cliente
    )

```

```

    BEGIN

```

```

SELECT @errno = 30005,
       @errmsg = 'Cannot update tb_cliente because tb_cliente_representante exists.'
GOTO ERROR
END
END

```

```

/* ERwin Builtin Trigger */
/* tb_cliente tb_tipo_cliente_cliente on parent update no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_cliente"
CHILD_OWNER="", CHILD_TABLE="tb_tipo_cliente_cliente"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_17", FK_COLUMNS="id_cliente" */
IF
/* %ParentPK(" OR",UPDATE) */
UPDATE(id_cliente)
BEGIN
IF EXISTS (
SELECT * FROM deleted,tb_tipo_cliente_cliente
WHERE
/* %JoinFKPK(tb_tipo_cliente_cliente,deleted," = "," AND") */
tb_tipo_cliente_cliente.id_cliente = deleted.id_cliente
)
BEGIN
SELECT @errno = 30005,
       @errmsg = 'Cannot update tb_cliente because tb_tipo_cliente_cliente exists.'
GOTO ERROR
END
END

```

```

/* ERwin Builtin Trigger */
/* tb_cliente tb_cliente_consentimento on parent update no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_cliente"
CHILD_OWNER="", CHILD_TABLE="tb_cliente_consentimento"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_7", FK_COLUMNS="id_cliente" */
IF
/* %ParentPK(" OR",UPDATE) */
UPDATE(id_cliente)
BEGIN
IF EXISTS (
SELECT * FROM deleted,tb_cliente_consentimento
WHERE
/* %JoinFKPK(tb_cliente_consentimento,deleted," = "," AND") */
tb_cliente_consentimento.id_cliente = deleted.id_cliente
)
BEGIN
SELECT @errno = 30005,
       @errmsg = 'Cannot update tb_cliente because tb_cliente_consentimento exists.'
GOTO ERROR
END
END

```

```

/* ERwin Builtin Trigger */
/* tb_cliente tb_cliente_data_comemorativa on parent update no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_cliente"
CHILD_OWNER="", CHILD_TABLE="tb_cliente_data_comemorativa"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_5", FK_COLUMNS="id_cliente" */
IF
/* %ParentPK(" OR",UPDATE) */
UPDATE(id_cliente)
BEGIN
IF EXISTS (
SELECT * FROM deleted,tb_cliente_data_comemorativa
WHERE
/* %JoinFKPK(tb_cliente_data_comemorativa,deleted," = "," AND") */
tb_cliente_data_comemorativa.id_cliente = deleted.id_cliente
)
BEGIN
SELECT @errno = 30005,

```

```

        @errmsg = 'Cannot update tb_cliente because tb_cliente_data_comemorativa exists.'
    GOTO ERROR
END
END

```

```

/* ERwin Builtin Trigger */
/* tb_cliente tb_cliente_empresa on parent update no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_cliente"
  CHILD_OWNER="", CHILD_TABLE="tb_cliente_empresa"
  P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
  FK_CONSTRAINT="R_3", FK_COLUMNS="id_cliente" */
IF
/* %ParentPK(" OR",UPDATE) */
UPDATE(id_cliente)
BEGIN
IF EXISTS (
  SELECT * FROM deleted,tb_cliente_empresa
  WHERE
/* %JoinFKPK(tb_cliente_empresa,deleted," = "," AND") */
  tb_cliente_empresa.id_cliente = deleted.id_cliente
)
BEGIN
  SELECT @errno = 30005,
    @errmsg = 'Cannot update tb_cliente because tb_cliente_empresa exists.'
  GOTO ERROR
END
END

```

```

/* ERwin Builtin Trigger */
RETURN
ERROR:
  raiserror @errno @errmsg
  rollback transaction
END

```

go

```

CREATE TRIGGER tD_tb_cliente_atacado ON tb_cliente_atacado FOR DELETE AS
/* ERwin Builtin Trigger */
/* DELETE trigger on tb_cliente_atacado */
BEGIN
  DECLARE @errno int,
    @errmsg varchar(255)
  /* ERwin Builtin Trigger */
  /* tb_cliente tb_cliente_atacado on child delete no action */
  /* ERWIN_RELATION:CHECKSUM="0001464a", PARENT_OWNER="", PARENT_TABLE="tb_cliente"
    CHILD_OWNER="", CHILD_TABLE="tb_cliente_atacado"
    P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
    FK_CONSTRAINT="R_19", FK_COLUMNS="id_cliente" */
  IF EXISTS (SELECT * FROM deleted,tb_cliente
    WHERE
/* %JoinFKPK(deleted,tb_cliente," = "," AND") */
    deleted.id_cliente = tb_cliente.id_cliente AND
    NOT EXISTS (
      SELECT * FROM tb_cliente_atacado
      WHERE
/* %JoinFKPK(tb_cliente_atacado,tb_cliente," = "," AND") */
        tb_cliente_atacado.id_cliente = tb_cliente.id_cliente
    )
  )
BEGIN
  SELECT @errno = 30010,
    @errmsg = 'Cannot delete last tb_cliente_atacado because tb_cliente exists.'
  GOTO ERROR
END

```

```

/* ERwin Builtin Trigger */
RETURN
ERROR:
    raiserror @errno @errmsg
    rollback transaction
END

```

go

```

CREATE TRIGGER tU_tb_cliente_atacado ON tb_cliente_atacado FOR UPDATE AS
/* ERwin Builtin Trigger */
/* UPDATE trigger on tb_cliente_atacado */
BEGIN
    DECLARE @NUMROWS int,
            @nullcnt int,
            @validcnt int,
            @insid_cliente integer,
            @errno int,
            @errmsg varchar(255)

    SELECT @NUMROWS = @@rowcount
    /* ERwin Builtin Trigger */
    /* tb_cliente tb_cliente_atacado on child update no action */
    /* ERWIN_RELATION:CHECKSUM="000160c2", PARENT_OWNER="", PARENT_TABLE="tb_cliente"
    CHILD_OWNER="", CHILD_TABLE="tb_cliente_atacado"
    P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
    FK_CONSTRAINT="R_19", FK_COLUMNS="id_cliente" */
    IF
        /* %ChildFK(" OR",UPDATE) */
        UPDATE(id_cliente)
    BEGIN
        SELECT @nullcnt = 0
        SELECT @validcnt = count(*)
            FROM inserted,tb_cliente
            WHERE
                /* %JoinFKPK(inserted,tb_cliente) */
                inserted.id_cliente = tb_cliente.id_cliente
        /* %NotNullFK(inserted," IS NULL","select @nullcnt = count(*) from inserted where"," AND") */

        IF @validcnt + @nullcnt != @NUMROWS
        BEGIN
            SELECT @errno = 30007,
                @errmsg = 'Cannot update tb_cliente_atacado because tb_cliente does not exist.'
            GOTO ERROR
        END
    END
END

```

```

/* ERwin Builtin Trigger */
RETURN
ERROR:
    raiserror @errno @errmsg
    rollback transaction
END

```

go

```

CREATE TRIGGER tD_tb_cliente_consentimento ON tb_cliente_consentimento FOR DELETE AS
/* ERwin Builtin Trigger */
/* DELETE trigger on tb_cliente_consentimento */
BEGIN
    DECLARE @errno int,
            @errmsg varchar(255)
    /* ERwin Builtin Trigger */
    /* tb_cliente tb_cliente_consentimento on child delete no action */

```



```

/* ERWIN_RELATION:CHECKSUM="000154f7", PARENT_OWNER="", PARENT_TABLE="tb_cliente"
CHILD_OWNER="", CHILD_TABLE="tb_cliente_consentimento"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_7", FK_COLUMNS="id_cliente" */
IF EXISTS (SELECT * FROM deleted,tb_cliente
WHERE
/* %JoinFKPK(deleted,tb_cliente,"=","AND") */
deleted.id_cliente = tb_cliente.id_cliente AND
NOT EXISTS (
SELECT * FROM tb_cliente_consentimento
WHERE
/* %JoinFKPK(tb_cliente_consentimento,tb_cliente,"=","AND") */
tb_cliente_consentimento.id_cliente = tb_cliente.id_cliente
)
)
BEGIN
SELECT @errno = 30010,
       @errmsg = 'Cannot delete last tb_cliente_consentimento because tb_cliente exists.'
GOTO ERROR
END

/* ERwin Builtin Trigger */
RETURN
ERROR:
raiserror @errno @errmsg
rollback transaction
END

go

CREATE TRIGGER tU_tb_cliente_consentimento ON tb_cliente_consentimento FOR UPDATE AS
/* ERwin Builtin Trigger */
/* UPDATE trigger on tb_cliente_consentimento */
BEGIN
DECLARE @NUMROWS int,
        @nullcnt int,
        @validcnt int,
        @insid_cliente integer,
        @insdt_consentimento date,
        @errno int,
        @errmsg varchar(255)

SELECT @NUMROWS = @@rowcount
/* ERwin Builtin Trigger */
/* tb_cliente tb_cliente_consentimento on child update no action */
/* ERWIN_RELATION:CHECKSUM="00016a64", PARENT_OWNER="", PARENT_TABLE="tb_cliente"
CHILD_OWNER="", CHILD_TABLE="tb_cliente_consentimento"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_7", FK_COLUMNS="id_cliente" */
IF
/* %ChildFK(" OR",UPDATE) */
UPDATE(id_cliente)
BEGIN
SELECT @nullcnt = 0
SELECT @validcnt = count(*)
FROM inserted,tb_cliente
WHERE
/* %JoinFKPK(inserted,tb_cliente) */
inserted.id_cliente = tb_cliente.id_cliente
/* %NotNullFK(inserted,"IS NULL","select @nullcnt = count(*) from inserted where","AND") */

IF @validcnt + @nullcnt != @NUMROWS
BEGIN
SELECT @errno = 30007,
       @errmsg = 'Cannot update tb_cliente_consentimento because tb_cliente does not exist.'
GOTO ERROR
END
END

```

```

/* ERwin Builtin Trigger */
RETURN
ERROR:
    raiserror @errno @errmsg
    rollback transaction
END

```

go

```

CREATE TRIGGER tD_tb_cliente_data_comemorativa ON tb_cliente_data_comemorativa FOR DELETE AS
/* ERwin Builtin Trigger */
/* DELETE trigger on tb_cliente_data_comemorativa */
BEGIN
    DECLARE @errno int,
            @errmsg varchar(255)
    /* ERwin Builtin Trigger */
    /* tb_data_comemorativa tb_cliente_data_comemorativa on child delete no action */
    /* ERWIN_RELATION:CHECKSUM="00030db7", PARENT_OWNER="", PARENT_TABLE="tb_data_comemorativa"
    CHILD_OWNER="", CHILD_TABLE="tb_cliente_data_comemorativa"
    P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
    FK_CONSTRAINT="R_6", FK_COLUMNS="id_data_comemorativa" */
    IF EXISTS (SELECT * FROM deleted, tb_data_comemorativa
    WHERE
        /* %JoinFKPK(deleted, tb_data_comemorativa, " = ", " AND") */
        deleted.id_data_comemorativa = tb_data_comemorativa.id_data_comemorativa AND
        NOT EXISTS (
            SELECT * FROM tb_cliente_data_comemorativa
            WHERE
                /* %JoinFKPK(tb_cliente_data_comemorativa, tb_data_comemorativa, " = ", " AND") */
                tb_cliente_data_comemorativa.id_data_comemorativa = tb_data_comemorativa.id_data_comemorativa
        )
    )
    BEGIN
        SELECT @errno = 30010,
               @errmsg = 'Cannot delete last tb_cliente_data_comemorativa because tb_data_comemorativa exists.'
        GOTO ERROR
    END

/* ERwin Builtin Trigger */
/* tb_cliente tb_cliente_data_comemorativa on child delete no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_cliente"
CHILD_OWNER="", CHILD_TABLE="tb_cliente_data_comemorativa"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_5", FK_COLUMNS="id_cliente" */
IF EXISTS (SELECT * FROM deleted, tb_cliente
WHERE
    /* %JoinFKPK(deleted, tb_cliente, " = ", " AND") */
    deleted.id_cliente = tb_cliente.id_cliente AND
    NOT EXISTS (
        SELECT * FROM tb_cliente_data_comemorativa
        WHERE
            /* %JoinFKPK(tb_cliente_data_comemorativa, tb_cliente, " = ", " AND") */
            tb_cliente_data_comemorativa.id_cliente = tb_cliente.id_cliente
        )
    )
    BEGIN
        SELECT @errno = 30010,
               @errmsg = 'Cannot delete last tb_cliente_data_comemorativa because tb_cliente exists.'
        GOTO ERROR
    END

```

```

/* ERwin Builtin Trigger */
RETURN
ERROR:

```

```
        raiserror @errno @errmsg
        rollback transaction
END
```

```
go
```

```
CREATE TRIGGER tU_tb_cliente_data_comemorativa ON tb_cliente_data_comemorativa FOR UPDATE AS
/* ERwin Builtin Trigger */
/* UPDATE trigger on tb_cliente_data_comemorativa */
BEGIN
    DECLARE @NUMROWS int,
            @nullcnt int,
            @validcnt int,
            @insid_cliente integer,
            @insid_data_comemorativa integer,
            @errno int,
            @errmsg varchar(255)

    SELECT @NUMROWS = @@rowcount
    /* ERwin Builtin Trigger */
    /* tb_data_comemorativa tb_cliente_data_comemorativa on child update no action */
    /* ERWIN_RELATION:CHECKSUM="0002f13c", PARENT_OWNER="", PARENT_TABLE="tb_data_comemorativa"
    CHILD_OWNER="", CHILD_TABLE="tb_cliente_data_comemorativa"
    P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
    FK_CONSTRAINT="R_6", FK_COLUMNS="id_data_comemorativa" */
    IF
        /* %ChildFK(" OR",UPDATE) */
        UPDATE(id_data_comemorativa)
    BEGIN
        SELECT @nullcnt = 0
        SELECT @validcnt = count(*)
            FROM inserted,tb_data_comemorativa
            WHERE
                /* %JoinFKPK(inserted,tb_data_comemorativa) */
                inserted.id_data_comemorativa = tb_data_comemorativa.id_data_comemorativa
        /* %NotNullFK(inserted," IS NULL","select @nullcnt = count(*) from inserted where"," AND") */

        IF @validcnt + @nullcnt != @NUMROWS
        BEGIN
            SELECT @errno = 30007,
                   @errmsg = 'Cannot update tb_cliente_data_comemorativa because tb_data_comemorativa does not exist.'
            GOTO ERROR
        END
    END

    /* ERwin Builtin Trigger */
    /* tb_cliente tb_cliente_data_comemorativa on child update no action */
    /* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_cliente"
    CHILD_OWNER="", CHILD_TABLE="tb_cliente_data_comemorativa"
    P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
    FK_CONSTRAINT="R_5", FK_COLUMNS="id_cliente" */
    IF
        /* %ChildFK(" OR",UPDATE) */
        UPDATE(id_cliente)
    BEGIN
        SELECT @nullcnt = 0
        SELECT @validcnt = count(*)
            FROM inserted,tb_cliente
            WHERE
                /* %JoinFKPK(inserted,tb_cliente) */
                inserted.id_cliente = tb_cliente.id_cliente
        /* %NotNullFK(inserted," IS NULL","select @nullcnt = count(*) from inserted where"," AND") */

        IF @validcnt + @nullcnt != @NUMROWS
        BEGIN
            SELECT @errno = 30007,
                   @errmsg = 'Cannot update tb_cliente_data_comemorativa because tb_cliente does not exist.'
            GOTO ERROR
        END
    END
```

END

```
/* ERwin Builtin Trigger */
RETURN
ERROR:
    raiserror @errno @errmsg
    rollback transaction
END
```

go

```
CREATE TRIGGER tD_tb_cliente_empresa ON tb_cliente_empresa FOR DELETE AS
/* ERwin Builtin Trigger */
/* DELETE trigger on tb_cliente_empresa */
BEGIN
    DECLARE @errno int,
            @errmsg varchar(255)
    /* ERwin Builtin Trigger */
    /* tb_empresa tb_cliente_empresa on child delete no action */
    /* ERWIN_RELATION:CHECKSUM="0002913e", PARENT_OWNER="", PARENT_TABLE="tb_empresa"
    CHILD_OWNER="", CHILD_TABLE="tb_cliente_empresa"
    P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
    FK_CONSTRAINT="R_4", FK_COLUMNS="id_empresa" */
    IF EXISTS (SELECT * FROM deleted, tb_empresa
    WHERE
        /* %JoinFKPK(deleted, tb_empresa, " = ", " AND") */
        deleted.id_empresa = tb_empresa.id_empresa AND
        NOT EXISTS (
            SELECT * FROM tb_cliente_empresa
            WHERE
                /* %JoinFKPK(tb_cliente_empresa, tb_empresa, " = ", " AND") */
                tb_cliente_empresa.id_empresa = tb_empresa.id_empresa
        )
    )
    BEGIN
        SELECT @errno = 30010,
               @errmsg = 'Cannot delete last tb_cliente_empresa because tb_empresa exists.'
        GOTO ERROR
    END

    /* ERwin Builtin Trigger */
    /* tb_cliente tb_cliente_empresa on child delete no action */
    /* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_cliente"
    CHILD_OWNER="", CHILD_TABLE="tb_cliente_empresa"
    P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
    FK_CONSTRAINT="R_3", FK_COLUMNS="id_cliente" */
    IF EXISTS (SELECT * FROM deleted, tb_cliente
    WHERE
        /* %JoinFKPK(deleted, tb_cliente, " = ", " AND") */
        deleted.id_cliente = tb_cliente.id_cliente AND
        NOT EXISTS (
            SELECT * FROM tb_cliente_empresa
            WHERE
                /* %JoinFKPK(tb_cliente_empresa, tb_cliente, " = ", " AND") */
                tb_cliente_empresa.id_cliente = tb_cliente.id_cliente
        )
    )
    BEGIN
        SELECT @errno = 30010,
               @errmsg = 'Cannot delete last tb_cliente_empresa because tb_cliente exists.'
        GOTO ERROR
    END
```

```
/* ERwin Builtin Trigger */
RETURN
```

```
ERROR:
  raiserror @errno @errmsg
  rollback transaction
END
```

```
go
```

```
CREATE TRIGGER tU_tb_cliente_empresa ON tb_cliente_empresa FOR UPDATE AS
/* ERwin Builtin Trigger */
/* UPDATE trigger on tb_cliente_empresa */
BEGIN
  DECLARE @NUMROWS int,
          @nullcnt int,
          @validcnt int,
          @insid_cliente integer,
          @insid_empresa integer,
          @errno int,
          @errmsg varchar(255)

  SELECT @NUMROWS = @@rowcount
  /* ERwin Builtin Trigger */
  /* tb_empresa tb_cliente_empresa on child update no action */
  /* ERWIN_RELATION:CHECKSUM="0002b4c9", PARENT_OWNER="", PARENT_TABLE="tb_empresa"
  CHILD_OWNER="", CHILD_TABLE="tb_cliente_empresa"
  P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
  FK_CONSTRAINT="R_4", FK_COLUMNS="id_empresa" */
  IF
    /* %ChildFK(" OR",UPDATE) */
    UPDATE(id_empresa)
  BEGIN
    SELECT @nullcnt = 0
    SELECT @validcnt = count(*)
      FROM inserted,tb_empresa
      WHERE
        /* %JoinFKPK(inserted,tb_empresa) */
        inserted.id_empresa = tb_empresa.id_empresa
    /* %NotNullFK(inserted," IS NULL","select @nullcnt = count(*) from inserted where"," AND") */

    IF @validcnt + @nullcnt != @NUMROWS
    BEGIN
      SELECT @errno = 30007,
             @errmsg = 'Cannot update tb_cliente_empresa because tb_empresa does not exist.'
      GOTO ERROR
    END
  END

  /* ERwin Builtin Trigger */
  /* tb_cliente tb_cliente_empresa on child update no action */
  /* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_cliente"
  CHILD_OWNER="", CHILD_TABLE="tb_cliente_empresa"
  P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
  FK_CONSTRAINT="R_3", FK_COLUMNS="id_cliente" */
  IF
    /* %ChildFK(" OR",UPDATE) */
    UPDATE(id_cliente)
  BEGIN
    SELECT @nullcnt = 0
    SELECT @validcnt = count(*)
      FROM inserted,tb_cliente
      WHERE
        /* %JoinFKPK(inserted,tb_cliente) */
        inserted.id_cliente = tb_cliente.id_cliente
    /* %NotNullFK(inserted," IS NULL","select @nullcnt = count(*) from inserted where"," AND") */

    IF @validcnt + @nullcnt != @NUMROWS
    BEGIN
      SELECT @errno = 30007,
             @errmsg = 'Cannot update tb_cliente_empresa because tb_cliente does not exist.'
      GOTO ERROR
    END
  END
```

```
END
END
```

```
/* ERwin Builtin Trigger */
RETURN
ERROR:
    raiserror @errno @errmsg
    rollback transaction
END
```

go

```
CREATE TRIGGER tD_tb_cliente_representante ON tb_cliente_representante FOR DELETE AS
/* ERwin Builtin Trigger */
/* DELETE trigger on tb_cliente_representante */
BEGIN
    DECLARE @errno int,
            @errmsg varchar(255)
    /* ERwin Builtin Trigger */
    /* tb_cliente tb_cliente_representante on child delete no action */
    /* ERWIN_RELATION:CHECKSUM="00015197", PARENT_OWNER="", PARENT_TABLE="tb_cliente"
    CHILD_OWNER="", CHILD_TABLE="tb_cliente_representante"
    P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
    FK_CONSTRAINT="R_18", FK_COLUMNS="id_cliente" */
    IF EXISTS (SELECT * FROM deleted, tb_cliente
    WHERE
        /* %JoinFKPK(deleted, tb_cliente, " = ", " AND") */
        deleted.id_cliente = tb_cliente.id_cliente AND
        NOT EXISTS (
            SELECT * FROM tb_cliente_representante
            WHERE
                /* %JoinFKPK(tb_cliente_representante, tb_cliente, " = ", " AND") */
                tb_cliente_representante.id_cliente = tb_cliente.id_cliente
        )
    )
    BEGIN
        SELECT @errno = 30010,
               @errmsg = 'Cannot delete last tb_cliente_representante because tb_cliente exists.'
        GOTO ERROR
    END
```

```
/* ERwin Builtin Trigger */
RETURN
ERROR:
    raiserror @errno @errmsg
    rollback transaction
END
```

go

```
CREATE TRIGGER tU_tb_cliente_representante ON tb_cliente_representante FOR UPDATE AS
/* ERwin Builtin Trigger */
/* UPDATE trigger on tb_cliente_representante */
BEGIN
    DECLARE @NUMROWS int,
            @nullcnt int,
            @validcnt int,
            @insid_cliente integer,
            @inscidade_atuacao varchar(20),
            @insbairro_atuacao varchar(20),
            @insdt_inicio_vigencia date,
            @errno int,
            @errmsg varchar(255)
```

```

SELECT @NUMROWS = @@rowcount
/* ERwin Builtin Trigger */
/* tb_cliente tb_cliente_representante on child update no action */
/* ERWIN_RELATION:CHECKSUM="000164ad", PARENT_OWNER="", PARENT_TABLE="tb_cliente"
CHILD_OWNER="", CHILD_TABLE="tb_cliente_representante"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_18", FK_COLUMNS="id_cliente" */
IF
/* %ChildFK(" OR",UPDATE) */
UPDATE(id_cliente)
BEGIN
SELECT @nullcnt = 0
SELECT @validcnt = count(*)
FROM inserted,tb_cliente
WHERE
/* %JoinFKPK(inserted,tb_cliente) */
inserted.id_cliente = tb_cliente.id_cliente
/* %NotNullFK(inserted," IS NULL","select @nullcnt = count(*) from inserted where"," AND") */

IF @validcnt + @nullcnt != @NUMROWS
BEGIN
SELECT @errno = 30007,
@errmsg = 'Cannot update tb_cliente_representante because tb_cliente does not exist.'
GOTO ERROR
END
END

/* ERwin Builtin Trigger */
RETURN
ERROR:
raiserror @errno @errmsg
rollback transaction
END

```

go

```

CREATE TRIGGER tD_tb_condicao_pagamento ON tb_condicao_pagamento FOR DELETE AS
/* ERwin Builtin Trigger */
/* DELETE trigger on tb_condicao_pagamento */
BEGIN
DECLARE @errno int,
@errmsg varchar(255)
/* ERwin Builtin Trigger */
/* tb_condicao_pagamento tb_meio_condicao_pagamento on parent delete no action */
/* ERWIN_RELATION:CHECKSUM="0003a342", PARENT_OWNER="", PARENT_TABLE="tb_condicao_pagamento"
CHILD_OWNER="", CHILD_TABLE="tb_meio_condicao_pagamento"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_25", FK_COLUMNS="id_condicao" */
IF EXISTS (
SELECT * FROM deleted,tb_meio_condicao_pagamento
WHERE
/* %JoinFKPK(tb_meio_condicao_pagamento,deleted," = "," AND") */
tb_meio_condicao_pagamento.id_condicao = deleted.id_condicao
)
BEGIN
SELECT @errno = 30001,
@errmsg = 'Cannot delete tb_condicao_pagamento because tb_meio_condicao_pagamento exists.'
GOTO ERROR
END

/* ERwin Builtin Trigger */
/* tb_condicao_pagamento tb_condicao_pagamento_parcela on parent delete no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_condicao_pagamento"
CHILD_OWNER="", CHILD_TABLE="tb_condicao_pagamento_parcela"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_24", FK_COLUMNS="id_condicao" */

```

```

IF EXISTS (
  SELECT * FROM deleted,tb_condicao_pagamento_parcela
  WHERE
    /* %JoinFKPK(tb_condicao_pagamento_parcela,deleted,"=","AND") */
    tb_condicao_pagamento_parcela.id_condicao = deleted.id_condicao
)
BEGIN
  SELECT @errno = 30001,
    @errmsg = 'Cannot delete tb_condicao_pagamento because tb_condicao_pagamento_parcela exists.'
  GOTO ERROR
END

/* ERwin Builtin Trigger */
/* tb_condicao_pagamento tb_tipo_cliente_condicao on parent delete no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_condicao_pagamento"
  CHILD_OWNER="", CHILD_TABLE="tb_tipo_cliente_condicao"
  P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
  FK_CONSTRAINT="R_23", FK_COLUMNS="id_condicao" */
IF EXISTS (
  SELECT * FROM deleted,tb_tipo_cliente_condicao
  WHERE
    /* %JoinFKPK(tb_tipo_cliente_condicao,deleted,"=","AND") */
    tb_tipo_cliente_condicao.id_condicao = deleted.id_condicao
)
BEGIN
  SELECT @errno = 30001,
    @errmsg = 'Cannot delete tb_condicao_pagamento because tb_tipo_cliente_condicao exists.'
  GOTO ERROR
END

/* ERwin Builtin Trigger */
RETURN
ERROR:
  raiserror @errno @errmsg
  rollback transaction
END

```

go

```

CREATE TRIGGER tU_tb_condicao_pagamento ON tb_condicao_pagamento FOR UPDATE AS
/* ERwin Builtin Trigger */
/* UPDATE trigger on tb_condicao_pagamento */
BEGIN
  DECLARE @NUMROWS int,
    @nullcnt int,
    @validcnt int,
    @insid_condicao integer,
    @errno int,
    @errmsg varchar(255)

  SELECT @NUMROWS = @@rowcount
  /* ERwin Builtin Trigger */
  /* tb_condicao_pagamento tb_meio_condicao_pagamento on parent update no action */
  /* ERWIN_RELATION:CHECKSUM="0003f8e8", PARENT_OWNER="", PARENT_TABLE="tb_condicao_pagamento"
    CHILD_OWNER="", CHILD_TABLE="tb_meio_condicao_pagamento"
    P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
    FK_CONSTRAINT="R_25", FK_COLUMNS="id_condicao" */
  IF
    /* %ParentPK(" OR",UPDATE) */
    UPDATE(id_condicao)
  BEGIN
    IF EXISTS (
      SELECT * FROM deleted,tb_meio_condicao_pagamento
      WHERE
        /* %JoinFKPK(tb_meio_condicao_pagamento,deleted,"=","AND") */
        tb_meio_condicao_pagamento.id_condicao = deleted.id_condicao
    )
    BEGIN

```



```

SELECT @errno = 30005,
       @errmsg = 'Cannot update tb_condicao_pagamento because tb_meio_condicao_pagamento exists.'
GOTO ERROR
END
END

/* ERwin Builtin Trigger */
/* tb_condicao_pagamento tb_condicao_pagamento_parcela on parent update no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_condicao_pagamento"
   CHILD_OWNER="", CHILD_TABLE="tb_condicao_pagamento_parcela"
   P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
   FK_CONSTRAINT="R_24", FK_COLUMNS="id_condicao" */
IF
/* %ParentPK(" OR",UPDATE) */
UPDATE(id_condicao)
BEGIN
IF EXISTS (
  SELECT * FROM deleted,tb_condicao_pagamento_parcela
  WHERE
    /* %JoinFKPK(tb_condicao_pagamento_parcela,deleted," = "," AND") */
    tb_condicao_pagamento_parcela.id_condicao = deleted.id_condicao
)
BEGIN
  SELECT @errno = 30005,
         @errmsg = 'Cannot update tb_condicao_pagamento because tb_condicao_pagamento_parcela exists.'
  GOTO ERROR
END
END

/* ERwin Builtin Trigger */
/* tb_condicao_pagamento tb_tipo_cliente_condicao on parent update no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_condicao_pagamento"
   CHILD_OWNER="", CHILD_TABLE="tb_tipo_cliente_condicao"
   P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
   FK_CONSTRAINT="R_23", FK_COLUMNS="id_condicao" */
IF
/* %ParentPK(" OR",UPDATE) */
UPDATE(id_condicao)
BEGIN
IF EXISTS (
  SELECT * FROM deleted,tb_tipo_cliente_condicao
  WHERE
    /* %JoinFKPK(tb_tipo_cliente_condicao,deleted," = "," AND") */
    tb_tipo_cliente_condicao.id_condicao = deleted.id_condicao
)
BEGIN
  SELECT @errno = 30005,
         @errmsg = 'Cannot update tb_condicao_pagamento because tb_tipo_cliente_condicao exists.'
  GOTO ERROR
END
END

/* ERwin Builtin Trigger */
RETURN
ERROR:
  raiserror @errno @errmsg
  rollback transaction
END

go

```

```

CREATE TRIGGER tD_tb_condicao_pagamento_parcela ON tb_condicao_pagamento_parcela FOR DELETE AS
/* ERwin Builtin Trigger */
/* DELETE trigger on tb_condicao_pagamento_parcela */
BEGIN
  DECLARE @errno int,

```

```

        @errmsg varchar(255)
/* ERwin Builtin Trigger */
/* tb_condicao_pagamento tb_condicao_pagamento_parcela on child delete no action */
/* ERWIN_RELATION:CHECKSUM="00018933", PARENT_OWNER="", PARENT_TABLE="tb_condicao_pagamento"
CHILD_OWNER="", CHILD_TABLE="tb_condicao_pagamento_parcela"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_24", FK_COLUMNS="id_condicao" */
IF EXISTS (SELECT * FROM deleted, tb_condicao_pagamento
WHERE
/* %JoinFKPK(deleted, tb_condicao_pagamento, " = "," AND") */
deleted.id_condicao = tb_condicao_pagamento.id_condicao AND
NOT EXISTS (
SELECT * FROM tb_condicao_pagamento_parcela
WHERE
/* %JoinFKPK(tb_condicao_pagamento_parcela, tb_condicao_pagamento, " = "," AND") */
tb_condicao_pagamento_parcela.id_condicao = tb_condicao_pagamento.id_condicao
)
)
BEGIN
SELECT @errno = 30010,
        @errmsg = 'Cannot delete last tb_condicao_pagamento_parcela because tb_condicao_pagamento exists.'
GOTO ERROR
END

/* ERwin Builtin Trigger */
RETURN
ERROR:
raiserror @errno @errmsg
rollback transaction
END

go

```

```

CREATE TRIGGER tU_tb_condicao_pagamento_parcela ON tb_condicao_pagamento_parcela FOR UPDATE AS
/* ERwin Builtin Trigger */
/* UPDATE trigger on tb_condicao_pagamento_parcela */
BEGIN
DECLARE @NUMROWS int,
        @nullcnt int,
        @validcnt int,
        @insid_parcela integer,
        @insid_condicao integer,
        @errno int,
        @errmsg varchar(255)

SELECT @NUMROWS = @@rowcount
/* ERwin Builtin Trigger */
/* tb_condicao_pagamento tb_condicao_pagamento_parcela on child update no action */
/* ERWIN_RELATION:CHECKSUM="00017449", PARENT_OWNER="", PARENT_TABLE="tb_condicao_pagamento"
CHILD_OWNER="", CHILD_TABLE="tb_condicao_pagamento_parcela"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_24", FK_COLUMNS="id_condicao" */
IF
/* %ChildFK(" OR", UPDATE) */
UPDATE(id_condicao)
BEGIN
SELECT @nullcnt = 0
SELECT @validcnt = count(*)
FROM inserted, tb_condicao_pagamento
WHERE
/* %JoinFKPK(inserted, tb_condicao_pagamento) */
inserted.id_condicao = tb_condicao_pagamento.id_condicao
/* %NotNullFK(inserted, " IS NULL", "select @nullcnt = count(*) from inserted where", " AND") */

IF @validcnt + @nullcnt != @NUMROWS
BEGIN
SELECT @errno = 30007,
        @errmsg = 'Cannot update tb_condicao_pagamento_parcela because tb_condicao_pagamento does not exist.'

```

```
GOTO ERROR
END
END
```

```
/* ERwin Builtin Trigger */
RETURN
ERROR:
    raiserror @errno @errmsg
    rollback transaction
END
```

go

```
CREATE TRIGGER tD_tb_data_comemorativa ON tb_data_comemorativa FOR DELETE AS
/* ERwin Builtin Trigger */
/* DELETE trigger on tb_data_comemorativa */
BEGIN
    DECLARE @errno int,
            @errmsg varchar(255)
    /* ERwin Builtin Trigger */
    /* tb_data_comemorativa tb_cliente_data_comemorativa on parent delete no action */
    /* ERWIN_RELATION:CHECKSUM="000141c8", PARENT_OWNER="", PARENT_TABLE="tb_data_comemorativa"
    CHILD_OWNER="", CHILD_TABLE="tb_cliente_data_comemorativa"
    P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
    FK_CONSTRAINT="R_6", FK_COLUMNS="id_data_comemorativa" */
    IF EXISTS (
        SELECT * FROM deleted, tb_cliente_data_comemorativa
        WHERE
            /* %JoinFKPK(tb_cliente_data_comemorativa,deleted,"=","AND") */
            tb_cliente_data_comemorativa.id_data_comemorativa = deleted.id_data_comemorativa
    )
    BEGIN
        SELECT @errno = 30001,
               @errmsg = 'Cannot delete tb_data_comemorativa because tb_cliente_data_comemorativa exists.'
        GOTO ERROR
    END
```

```
/* ERwin Builtin Trigger */
RETURN
ERROR:
    raiserror @errno @errmsg
    rollback transaction
END
```

go

```
CREATE TRIGGER tU_tb_data_comemorativa ON tb_data_comemorativa FOR UPDATE AS
/* ERwin Builtin Trigger */
/* UPDATE trigger on tb_data_comemorativa */
BEGIN
    DECLARE @NUMROWS int,
            @nullcnt int,
            @validcnt int,
            @insid_data_comemorativa integer,
            @errno int,
            @errmsg varchar(255)

    SELECT @NUMROWS = @@rowcount
    /* ERwin Builtin Trigger */
    /* tb_data_comemorativa tb_cliente_data_comemorativa on parent update no action */
    /* ERWIN_RELATION:CHECKSUM="00015bde", PARENT_OWNER="", PARENT_TABLE="tb_data_comemorativa"
    CHILD_OWNER="", CHILD_TABLE="tb_cliente_data_comemorativa"
    P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
    FK_CONSTRAINT="R_6", FK_COLUMNS="id_data_comemorativa" */
```

```

IF
/* %ParentPK(" OR",UPDATE) */
UPDATE(id_data_comemorativa)
BEGIN
IF EXISTS (
SELECT * FROM deleted,tb_cliente_data_comemorativa
WHERE
/* %JoinFKPK(tb_cliente_data_comemorativa,deleted," = ","AND") */
tb_cliente_data_comemorativa.id_data_comemorativa = deleted.id_data_comemorativa
)
BEGIN
SELECT @errno = 30005,
@errmsg = 'Cannot update tb_data_comemorativa because tb_cliente_data_comemorativa exists.'
GOTO ERROR
END
END

/* ERwin Builtin Trigger */
RETURN
ERROR:
raiserror @errno @errmsg
rollback transaction
END

go

```

```

CREATE TRIGGER tD_tb_data_especial_empresa ON tb_data_especial_empresa FOR DELETE AS
/* ERwin Builtin Trigger */
/* DELETE trigger on tb_data_especial_empresa */
BEGIN
DECLARE @errno int,
@errmsg varchar(255)
/* ERwin Builtin Trigger */
/* tb_empresa tb_data_especial_empresa on child delete no action */
/* ERWIN_RELATION:CHECKSUM="0002debe", PARENT_OWNER="", PARENT_TABLE="tb_empresa"
CHILD_OWNER="", CHILD_TABLE="tb_data_especial_empresa"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_21", FK_COLUMNS="id_empresa" */
IF EXISTS (SELECT * FROM deleted,tb_empresa
WHERE
/* %JoinFKPK(deleted,tb_empresa," = ","AND") */
deleted.id_empresa = tb_empresa.id_empresa AND
NOT EXISTS (
SELECT * FROM tb_data_especial_empresa
WHERE
/* %JoinFKPK(tb_data_especial_empresa,tb_empresa," = ","AND") */
tb_data_especial_empresa.id_empresa = tb_empresa.id_empresa
)
)
BEGIN
SELECT @errno = 30010,
@errmsg = 'Cannot delete last tb_data_especial_empresa because tb_empresa exists.'
GOTO ERROR
END

/* ERwin Builtin Trigger */
/* tb_datas_especiais tb_data_especial_empresa on child delete no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_datas_especiais"
CHILD_OWNER="", CHILD_TABLE="tb_data_especial_empresa"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_20", FK_COLUMNS="id_data_especial" */
IF EXISTS (SELECT * FROM deleted,tb_datas_especiais
WHERE
/* %JoinFKPK(deleted,tb_datas_especiais," = ","AND") */
deleted.id_data_especial = tb_datas_especiais.id_data_especial AND
NOT EXISTS (

```

```

SELECT * FROM tb_data_especial_empresa
WHERE
/* %JoinFKPK(tb_data_especial_empresa,tb_datas_especiais,"=","AND") */
tb_data_especial_empresa.id_data_especial = tb_datas_especiais.id_data_especial
)
)
BEGIN
SELECT @errno = 30010,
       @errmsg = 'Cannot delete last tb_data_especial_empresa because tb_datas_especiais exists.'
GOTO ERROR
END

```

```

/* ERwin Builtin Trigger */
RETURN
ERROR:
raiserror @errno @errmsg
rollback transaction
END

```

go

```

CREATE TRIGGER tU_tb_data_especial_empresa ON tb_data_especial_empresa FOR UPDATE AS
/* ERwin Builtin Trigger */
/* UPDATE trigger on tb_data_especial_empresa */
BEGIN
DECLARE @NUMROWS int,
        @nullcnt int,
        @validcnt int,
        @insid_data_especial integer,
        @insid_empresa integer,
        @errno int,
        @errmsg varchar(255)

SELECT @NUMROWS = @@rowcount
/* ERwin Builtin Trigger */
/* tb_empresa tb_data_especial_empresa on child update no action */
/* ERWIN_RELATION:CHECKSUM="0002d30b", PARENT_OWNER="", PARENT_TABLE="tb_empresa"
CHILD_OWNER="", CHILD_TABLE="tb_data_especial_empresa"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_21", FK_COLUMNS="id_empresa" */
IF
/* %ChildFK("OR",UPDATE) */
UPDATE(id_empresa)
BEGIN
SELECT @nullcnt = 0
SELECT @validcnt = count(*)
FROM inserted,tb_empresa
WHERE
/* %JoinFKPK(inserted,tb_empresa) */
inserted.id_empresa = tb_empresa.id_empresa
/* %NotNullFK(inserted,"IS NULL","select @nullcnt = count(*) from inserted where","AND") */

IF @validcnt + @nullcnt != @NUMROWS
BEGIN
SELECT @errno = 30007,
       @errmsg = 'Cannot update tb_data_especial_empresa because tb_empresa does not exist.'
GOTO ERROR
END
END

```

```

/* ERwin Builtin Trigger */
/* tb_datas_especiais tb_data_especial_empresa on child update no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_datas_especiais"
CHILD_OWNER="", CHILD_TABLE="tb_data_especial_empresa"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_20", FK_COLUMNS="id_data_especial" */
IF
/* %ChildFK("OR",UPDATE) */

```

```

UPDATE(id_data_especial)
BEGIN
SELECT @nullcnt = 0
SELECT @validcnt = count(*)
  FROM inserted,tb_datas_especiais
  WHERE
    /* %JoinFKPK(inserted,tb_datas_especiais) */
    inserted.id_data_especial = tb_datas_especiais.id_data_especial
/* %NotNullFK(inserted," IS NULL","select @nullcnt = count(*) from inserted where"," AND") */

IF @validcnt + @nullcnt != @NUMROWS
BEGIN
  SELECT @errno = 30007,
    @errmsg = 'Cannot update tb_data_especial_empresa because tb_datas_especiais does not exist.'
  GOTO ERROR
END
END

```

```

/* ERwin Builtin Trigger */
RETURN
ERROR:
  raiserror @errno @errmsg
  rollback transaction
END

```

go

```

CREATE TRIGGER tD_tb_datas_especiais ON tb_datas_especiais FOR DELETE AS
/* ERwin Builtin Trigger */
/* DELETE trigger on tb_datas_especiais */
BEGIN
  DECLARE @errno int,
    @errmsg varchar(255)
  /* ERwin Builtin Trigger */
  /* tb_datas_especiais tb_data_especial_empresa on parent delete no action */
  /* ERWIN_RELATION:CHECKSUM="000130b6", PARENT_OWNER="", PARENT_TABLE="tb_datas_especiais"
  CHILD_OWNER="", CHILD_TABLE="tb_data_especial_empresa"
  P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
  FK_CONSTRAINT="R_20", FK_COLUMNS="id_data_especial" */
  IF EXISTS (
    SELECT * FROM deleted,tb_data_especial_empresa
    WHERE
      /* %JoinFKPK(tb_data_especial_empresa,deleted," = "," AND") */
      tb_data_especial_empresa.id_data_especial = deleted.id_data_especial
  )
  BEGIN
    SELECT @errno = 30001,
      @errmsg = 'Cannot delete tb_datas_especiais because tb_data_especial_empresa exists.'
    GOTO ERROR
  END

```

```

/* ERwin Builtin Trigger */
RETURN
ERROR:
  raiserror @errno @errmsg
  rollback transaction
END

```

go

```

CREATE TRIGGER tU_tb_datas_especiais ON tb_datas_especiais FOR UPDATE AS
/* ERwin Builtin Trigger */
/* UPDATE trigger on tb_datas_especiais */
BEGIN

```

```

DECLARE @NUMROWS int,
        @nullcnt int,
        @validcnt int,
        @insid_data_especial integer,
        @errno int,
        @errmsg varchar(255)

SELECT @NUMROWS = @@rowcount
/* ERwin Builtin Trigger */
/* tb_datas_especiais tb_data_especial_empresa on parent update no action */
/* ERWIN_RELATION:CHECKSUM="000147dd", PARENT_OWNER="", PARENT_TABLE="tb_datas_especiais"
  CHILD_OWNER="", CHILD_TABLE="tb_data_especial_empresa"
  P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
  FK_CONSTRAINT="R_20", FK_COLUMNS="id_data_especial" */
IF
/* %ParentPK(" OR",UPDATE) */
UPDATE(id_data_especial)
BEGIN
IF EXISTS (
  SELECT * FROM deleted,tb_data_especial_empresa
  WHERE
  /* %JoinFKPK(tb_data_especial_empresa,deleted,"="," AND") */
  tb_data_especial_empresa.id_data_especial = deleted.id_data_especial
)
BEGIN
  SELECT @errno = 30005,
         @errmsg = 'Cannot update tb_datas_especiais because tb_data_especial_empresa exists.'
  GOTO ERROR
END
END

/* ERwin Builtin Trigger */
RETURN
ERROR:
  raiserror @errno @errmsg
  rollback transaction
END

go

```

```

CREATE TRIGGER tD_tb_empresa ON tb_empresa FOR DELETE AS
/* ERwin Builtin Trigger */
/* DELETE trigger on tb_empresa */
BEGIN
DECLARE @errno int,
        @errmsg varchar(255)
/* ERwin Builtin Trigger */
/* tb_empresa tb_data_especial_empresa on parent delete no action */
/* ERWIN_RELATION:CHECKSUM="000303ca", PARENT_OWNER="", PARENT_TABLE="tb_empresa"
  CHILD_OWNER="", CHILD_TABLE="tb_data_especial_empresa"
  P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
  FK_CONSTRAINT="R_21", FK_COLUMNS="id_empresa" */
IF EXISTS (
  SELECT * FROM deleted,tb_data_especial_empresa
  WHERE
  /* %JoinFKPK(tb_data_especial_empresa,deleted,"="," AND") */
  tb_data_especial_empresa.id_empresa = deleted.id_empresa
)
BEGIN
  SELECT @errno = 30001,
         @errmsg = 'Cannot delete tb_empresa because tb_data_especial_empresa exists.'
  GOTO ERROR
END

/* ERwin Builtin Trigger */
/* tb_empresa tb_cliente_empresa on parent delete no action */

```

```

/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_empresa"
CHILD_OWNER="", CHILD_TABLE="tb_cliente_empresa"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_4", FK_COLUMNS="id_empresa" */
IF EXISTS (
  SELECT * FROM deleted,tb_cliente_empresa
WHERE
  /* %JoinFKPK(tb_cliente_empresa,deleted," = "," AND") */
  tb_cliente_empresa.id_empresa = deleted.id_empresa
)
BEGIN
  SELECT @errno = 30001,
         @errmsg = 'Cannot delete tb_empresa because tb_cliente_empresa exists.'
  GOTO ERROR
END

/* ERwin Builtin Trigger */
/* tb_empresa tb_filial on parent delete no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_empresa"
CHILD_OWNER="", CHILD_TABLE="tb_filial"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_2", FK_COLUMNS="id_empresa" */
IF EXISTS (
  SELECT * FROM deleted,tb_filial
WHERE
  /* %JoinFKPK(tb_filial,deleted," = "," AND") */
  tb_filial.id_empresa = deleted.id_empresa
)
BEGIN
  SELECT @errno = 30001,
         @errmsg = 'Cannot delete tb_empresa because tb_filial exists.'
  GOTO ERROR
END

/* ERwin Builtin Trigger */
RETURN
ERROR:
  raiserror @errno @errmsg
  rollback transaction
END

```

go

```

CREATE TRIGGER tU_tb_empresa ON tb_empresa FOR UPDATE AS
/* ERwin Builtin Trigger */
/* UPDATE trigger on tb_empresa */
BEGIN
  DECLARE @NUMROWS int,
          @nullcnt int,
          @validcnt int,
          @insid_empresa integer,
          @errno int,
          @errmsg varchar(255)

  SELECT @NUMROWS = @@rowcount
  /* ERwin Builtin Trigger */
  /* tb_empresa tb_data_especial_empresa on parent update no action */
  /* ERWIN_RELATION:CHECKSUM="00035153", PARENT_OWNER="", PARENT_TABLE="tb_empresa"
  CHILD_OWNER="", CHILD_TABLE="tb_data_especial_empresa"
  P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
  FK_CONSTRAINT="R_21", FK_COLUMNS="id_empresa" */
  IF
    /* %ParentPK(" OR",UPDATE) */
    UPDATE(id_empresa)
  BEGIN
    IF EXISTS (
      SELECT * FROM deleted,tb_data_especial_empresa
      WHERE

```



```

/* %JoinFKPK(tb_data_especial_empresa,deleted,"=","AND") */
tb_data_especial_empresa.id_empresa = deleted.id_empresa
)
BEGIN
  SELECT @errno = 30005,
         @errmsg = 'Cannot update tb_empresa because tb_data_especial_empresa exists.'
  GOTO ERROR
END
END

/* ERwin Builtin Trigger */
/* tb_empresa tb_cliente_empresa on parent update no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_empresa"
  CHILD_OWNER="", CHILD_TABLE="tb_cliente_empresa"
  P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
  FK_CONSTRAINT="R_4", FK_COLUMNS="id_empresa" */
IF
  /* %ParentPK("OR",UPDATE) */
  UPDATE(id_empresa)
BEGIN
  IF EXISTS (
    SELECT * FROM deleted,tb_cliente_empresa
    WHERE
      /* %JoinFKPK(tb_cliente_empresa,deleted,"=","AND") */
      tb_cliente_empresa.id_empresa = deleted.id_empresa
  )
  BEGIN
    SELECT @errno = 30005,
           @errmsg = 'Cannot update tb_empresa because tb_cliente_empresa exists.'
    GOTO ERROR
  END
END

/* ERwin Builtin Trigger */
/* tb_empresa tb_filial on parent update no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_empresa"
  CHILD_OWNER="", CHILD_TABLE="tb_filial"
  P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
  FK_CONSTRAINT="R_2", FK_COLUMNS="id_empresa" */
IF
  /* %ParentPK("OR",UPDATE) */
  UPDATE(id_empresa)
BEGIN
  IF EXISTS (
    SELECT * FROM deleted,tb_filial
    WHERE
      /* %JoinFKPK(tb_filial,deleted,"=","AND") */
      tb_filial.id_empresa = deleted.id_empresa
  )
  BEGIN
    SELECT @errno = 30005,
           @errmsg = 'Cannot update tb_empresa because tb_filial exists.'
    GOTO ERROR
  END
END

/* ERwin Builtin Trigger */
RETURN
ERROR:
  raiserror @errno @errmsg
  rollback transaction
END

go

```

CREATE TRIGGER tD\_tb\_estrutura\_dados\_sensíveis ON tb\_estrutura\_dados\_sensíveis FOR DELETE AS

```

/* ERwin Builtin Trigger */
/* DELETE trigger on tb_estrutura_dados_sensiveis */
BEGIN
DECLARE @erno int,
        @errmsg varchar(255)
/* ERwin Builtin Trigger */
/* tb_estrutura_dados_sensiveis tb_estrutura_dados_sensiveis_campos on parent delete no action */
/* ERWIN_RELATION:CHECKSUM="000154c8", PARENT_OWNER="", PARENT_TABLE="tb_estrutura_dados_sensiveis"
CHILD_OWNER="", CHILD_TABLE="tb_estrutura_dados_sensiveis_campos"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_9", FK_COLUMNS="id_estrutura_dados" */
IF EXISTS (
    SELECT * FROM deleted,tb_estrutura_dados_sensiveis_campos
    WHERE
        /* %JoinFKPK(tb_estrutura_dados_sensiveis_campos,deleted,"=","AND") */
        tb_estrutura_dados_sensiveis_campos.id_estrutura_dados = deleted.id_estrutura_dados
    )
BEGIN
    SELECT @erno = 30001,
           @errmsg = 'Cannot delete tb_estrutura_dados_sensiveis because tb_estrutura_dados_sensiveis_campos exists.'
    GOTO ERROR
END

```

```

/* ERwin Builtin Trigger */
RETURN
ERROR:
    raiserror @erno @errmsg
    rollback transaction
END

```

go

```

CREATE TRIGGER tU_tb_estrutura_dados_sensiveis ON tb_estrutura_dados_sensiveis FOR UPDATE AS
/* ERwin Builtin Trigger */
/* UPDATE trigger on tb_estrutura_dados_sensiveis */
BEGIN
DECLARE @NUMROWS int,
        @nullcnt int,
        @validcnt int,
        @insid_estrutura_dados integer,
        @erno int,
        @errmsg varchar(255)

SELECT @NUMROWS = @@rowcount
/* ERwin Builtin Trigger */
/* tb_estrutura_dados_sensiveis tb_estrutura_dados_sensiveis_campos on parent update no action */
/* ERWIN_RELATION:CHECKSUM="000170f1", PARENT_OWNER="", PARENT_TABLE="tb_estrutura_dados_sensiveis"
CHILD_OWNER="", CHILD_TABLE="tb_estrutura_dados_sensiveis_campos"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_9", FK_COLUMNS="id_estrutura_dados" */
IF
    /* %ParentPK(" OR",UPDATE) */
    UPDATE(id_estrutura_dados)
BEGIN
    IF EXISTS (
        SELECT * FROM deleted,tb_estrutura_dados_sensiveis_campos
        WHERE
            /* %JoinFKPK(tb_estrutura_dados_sensiveis_campos,deleted,"=","AND") */
            tb_estrutura_dados_sensiveis_campos.id_estrutura_dados = deleted.id_estrutura_dados
        )
    BEGIN
        SELECT @erno = 30005,
               @errmsg = 'Cannot update tb_estrutura_dados_sensiveis because tb_estrutura_dados_sensiveis_campos exists.'
        GOTO ERROR
    END
END

```

```

/* ERwin Builtin Trigger */
RETURN
ERROR:
    raiserror @errno @errmsg
    rollback transaction
END

```

go

```

CREATE TRIGGER tD_tb_estrutura_dados_sensiveis_campos ON tb_estrutura_dados_sensiveis_campos FOR DELETE AS
/* ERwin Builtin Trigger */
/* DELETE trigger on tb_estrutura_dados_sensiveis_campos */
BEGIN
    DECLARE @errno int,
            @errmsg varchar(255)
    /* ERwin Builtin Trigger */
    /* tb_estrutura_dados_sensiveis_campos tb_usuario_estrutura_dados on parent delete no action */
    /* ERWIN_RELATION:CHECKSUM="0003494c", PARENT_OWNER="", PARENT_TABLE="tb_estrutura_dados_sensiveis_campos"
    CHILD_OWNER="", CHILD_TABLE="tb_usuario_estrutura_dados"
    P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
    FK_CONSTRAINT="R_14", FK_COLUMNS="id_estrutura_campos""id_estrutura_dados" */
    IF EXISTS (
        SELECT * FROM deleted,tb_usuario_estrutura_dados
        WHERE
            /* %JoinFKPK(tb_usuario_estrutura_dados,deleted,"=","AND") */
            tb_usuario_estrutura_dados.id_estrutura_campos = deleted.id_estrutura_campos AND
            tb_usuario_estrutura_dados.id_estrutura_dados = deleted.id_estrutura_dados
        )
    BEGIN
        SELECT @errno = 30001,
               @errmsg = 'Cannot delete tb_estrutura_dados_sensiveis_campos because tb_usuario_estrutura_dados exists.'
        GOTO ERROR
    END

```

```

/* ERwin Builtin Trigger */
/* tb_estrutura_dados_sensiveis tb_estrutura_dados_sensiveis_campos on child delete no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_estrutura_dados_sensiveis"
CHILD_OWNER="", CHILD_TABLE="tb_estrutura_dados_sensiveis_campos"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_9", FK_COLUMNS="id_estrutura_dados" */
IF EXISTS (SELECT * FROM deleted,tb_estrutura_dados_sensiveis
WHERE
    /* %JoinFKPK(deleted,tb_estrutura_dados_sensiveis,"=","AND") */
    deleted.id_estrutura_dados = tb_estrutura_dados_sensiveis.id_estrutura_dados AND
    NOT EXISTS (
        SELECT * FROM tb_estrutura_dados_sensiveis_campos
        WHERE
            /* %JoinFKPK(tb_estrutura_dados_sensiveis_campos,tb_estrutura_dados_sensiveis,"=","AND") */
            tb_estrutura_dados_sensiveis_campos.id_estrutura_dados = tb_estrutura_dados_sensiveis.id_estrutura_dados
        )
    )
BEGIN
    SELECT @errno = 30010,
           @errmsg = 'Cannot delete last tb_estrutura_dados_sensiveis_campos because tb_estrutura_dados_sensiveis exists.'
    GOTO ERROR
END

```

```

/* ERwin Builtin Trigger */
RETURN
ERROR:
    raiserror @errno @errmsg
    rollback transaction
END

```

go

```

CREATE TRIGGER tU_tb_estrutura_dados_sensiveis_campos ON tb_estrutura_dados_sensiveis_campos FOR UPDATE AS
/* ERwin Builtin Trigger */
/* UPDATE trigger on tb_estrutura_dados_sensiveis_campos */
BEGIN
  DECLARE @NUMROWS int,
          @nullcnt int,
          @validcnt int,
          @insid_estrutura_campos integer,
          @insid_estrutura_dados integer,
          @errno int,
          @errmsg varchar(255)

  SELECT @NUMROWS = @@rowcount
  /* ERwin Builtin Trigger */
  /* tb_estrutura_dados_sensiveis_campos tb_usuario_estrutura_dados on parent update no action */
  /* ERWIN_RELATION:CHECKSUM="00034728", PARENT_OWNER="", PARENT_TABLE="tb_estrutura_dados_sensiveis_campos"
  CHILD_OWNER="", CHILD_TABLE="tb_usuario_estrutura_dados"
  P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
  FK_CONSTRAINT="R_14", FK_COLUMNS="id_estrutura_campos" id_estrutura_dados" */
  IF
    /* %ParentPK(" OR",UPDATE) */
    UPDATE(id_estrutura_campos) OR
    UPDATE(id_estrutura_dados)
  BEGIN
    IF EXISTS (
      SELECT * FROM deleted, tb_usuario_estrutura_dados
      WHERE
        /* %JoinFKPK(tb_usuario_estrutura_dados,deleted," = "," AND") */
        tb_usuario_estrutura_dados.id_estrutura_campos = deleted.id_estrutura_campos AND
        tb_usuario_estrutura_dados.id_estrutura_dados = deleted.id_estrutura_dados
    )
    BEGIN
      SELECT @errno = 30005,
             @errmsg = 'Cannot update tb_estrutura_dados_sensiveis_campos because tb_usuario_estrutura_dados exists.'
      GOTO ERROR
    END
  END

  /* ERwin Builtin Trigger */
  /* tb_estrutura_dados_sensiveis tb_estrutura_dados_sensiveis_campos on child update no action */
  /* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_estrutura_dados_sensiveis"
  CHILD_OWNER="", CHILD_TABLE="tb_estrutura_dados_sensiveis_campos"
  P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
  FK_CONSTRAINT="R_9", FK_COLUMNS="id_estrutura_dados" */
  IF
    /* %ChildFK(" OR",UPDATE) */
    UPDATE(id_estrutura_dados)
  BEGIN
    SELECT @nullcnt = 0
    SELECT @validcnt = count(*)
      FROM inserted, tb_estrutura_dados_sensiveis
      WHERE
        /* %JoinFKPK(inserted, tb_estrutura_dados_sensiveis) */
        inserted.id_estrutura_dados = tb_estrutura_dados_sensiveis.id_estrutura_dados
    /* %NotNullFK(inserted," IS NULL", "select @nullcnt = count(*) from inserted where", " AND") */

    IF @validcnt + @nullcnt != @NUMROWS
    BEGIN
      SELECT @errno = 30007,
             @errmsg = 'Cannot update tb_estrutura_dados_sensiveis_campos because tb_estrutura_dados_sensiveis does not exist.'
      GOTO ERROR
    END
  END

  /* ERwin Builtin Trigger */
  RETURN
ERROR:
  raiserror @errno @errmsg

```

```
rollback transaction
END
```

go

```
CREATE TRIGGER tD_tb_estrutura_sistema ON tb_estrutura_sistema FOR DELETE AS
/* ERwin Builtin Trigger */
/* DELETE trigger on tb_estrutura_sistema */
BEGIN
  DECLARE @ermo int,
          @errmsg varchar(255)
  /* ERwin Builtin Trigger */
  /* tb_estrutura_sistema tb_usuario_estrutura_sistema on parent delete no action */
  /* ERWIN_RELATION:CHECKSUM="0002b432", PARENT_OWNER="", PARENT_TABLE="tb_estrutura_sistema"
  CHILD_OWNER="", CHILD_TABLE="tb_usuario_estrutura_sistema"
  P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
  FK_CONSTRAINT="R_10", FK_COLUMNS="id_estrutura_sistema""id_modulo" */
  IF EXISTS (
    SELECT * FROM deleted, tb_usuario_estrutura_sistema
    WHERE
      /* %JoinFKPK(tb_usuario_estrutura_sistema,deleted,"=","AND") */
      tb_usuario_estrutura_sistema.id_estrutura_sistema = deleted.id_estrutura_sistema AND
      tb_usuario_estrutura_sistema.id_modulo = deleted.id_modulo
  )
  BEGIN
    SELECT @ermo = 30001,
           @errmsg = 'Cannot delete tb_estrutura_sistema because tb_usuario_estrutura_sistema exists.'
    GOTO ERROR
  END

  /* ERwin Builtin Trigger */
  /* tb_modulos tb_estrutura_sistema on child delete no action */
  /* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_modulos"
  CHILD_OWNER="", CHILD_TABLE="tb_estrutura_sistema"
  P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
  FK_CONSTRAINT="R_8", FK_COLUMNS="id_modulo" */
  IF EXISTS (SELECT * FROM deleted, tb_modulos
    WHERE
      /* %JoinFKPK(deleted,tb_modulos,"=","AND") */
      deleted.id_modulo = tb_modulos.id_modulo AND
      NOT EXISTS (
        SELECT * FROM tb_estrutura_sistema
        WHERE
          /* %JoinFKPK(tb_estrutura_sistema,tb_modulos,"=","AND") */
          tb_estrutura_sistema.id_modulo = tb_modulos.id_modulo
      )
  )
  BEGIN
    SELECT @ermo = 30010,
           @errmsg = 'Cannot delete last tb_estrutura_sistema because tb_modulos exists.'
    GOTO ERROR
  END

  /* ERwin Builtin Trigger */
  RETURN
ERROR:
  raiserror @ermo @errmsg
  rollback transaction
END
```

go

```
CREATE TRIGGER tU_tb_estrutura_sistema ON tb_estrutura_sistema FOR UPDATE AS
/* ERwin Builtin Trigger */
/* UPDATE trigger on tb_estrutura_sistema */
```

```

BEGIN
DECLARE @NUMROWS int,
        @nullcnt int,
        @validcnt int,
        @insid_estrutura_sistema integer,
        @insid_modulo integer,
        @errno int,
        @errmsg varchar(255)

SELECT @NUMROWS = @@rowcount
/* ERwin Builtin Trigger */
/* tb_estrutura_sistema tb_usuario_estrutura_sistema on parent update no action */
/* ERWIN_RELATION:CHECKSUM="0002e120", PARENT_OWNER="", PARENT_TABLE="tb_estrutura_sistema"
  CHILD_OWNER="", CHILD_TABLE="tb_usuario_estrutura_sistema"
  P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
  FK_CONSTRAINT="R_10", FK_COLUMNS="id_estrutura_sistema""id_modulo" */
IF
/* %ParentPK(" OR",UPDATE) */
UPDATE(id_estrutura_sistema) OR
UPDATE(id_modulo)
BEGIN
IF EXISTS (
  SELECT * FROM deleted,tb_usuario_estrutura_sistema
  WHERE
  /* %JoinFKPK(tb_usuario_estrutura_sistema,deleted," = "," AND") */
  tb_usuario_estrutura_sistema.id_estrutura_sistema = deleted.id_estrutura_sistema AND
  tb_usuario_estrutura_sistema.id_modulo = deleted.id_modulo
)
BEGIN
  SELECT @errno = 30005,
         @errmsg = 'Cannot update tb_estrutura_sistema because tb_usuario_estrutura_sistema exists.'
  GOTO ERROR
END
END

/* ERwin Builtin Trigger */
/* tb_modulos tb_estrutura_sistema on child update no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_modulos"
  CHILD_OWNER="", CHILD_TABLE="tb_estrutura_sistema"
  P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
  FK_CONSTRAINT="R_8", FK_COLUMNS="id_modulo" */
IF
/* %ChildFK(" OR",UPDATE) */
UPDATE(id_modulo)
BEGIN
SELECT @nullcnt = 0
SELECT @validcnt = count(*)
  FROM inserted,tb_modulos
  WHERE
  /* %JoinFKPK(inserted,tb_modulos) */
  inserted.id_modulo = tb_modulos.id_modulo
/* %NotNullFK(inserted," IS NULL","select @nullcnt = count(*) from inserted where"," AND") */

IF @validcnt + @nullcnt != @NUMROWS
BEGIN
  SELECT @errno = 30007,
         @errmsg = 'Cannot update tb_estrutura_sistema because tb_modulos does not exist.'
  GOTO ERROR
END
END

/* ERwin Builtin Trigger */
RETURN
ERROR:
  raiserror @errno @errmsg
  rollback transaction
END

```

go

```

CREATE TRIGGER tD_tb_filial ON tb_filial FOR DELETE AS
/* ERwin Builtin Trigger */
/* DELETE trigger on tb_filial */
BEGIN
  DECLARE @errno int,
          @errmsg varchar(255)
/* ERwin Builtin Trigger */
/* tb_empresa tb_filial on child delete no action */
/* ERWIN_RELATION:CHECKSUM="0001313e", PARENT_OWNER="", PARENT_TABLE="tb_empresa"
CHILD_OWNER="", CHILD_TABLE="tb_filial"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_2", FK_COLUMNS="id_empresa" */
IF EXISTS (SELECT * FROM deleted, tb_empresa
WHERE
/* %JoinFKPK(deleted, tb_empresa, " = ", " AND") */
deleted.id_empresa = tb_empresa.id_empresa AND
NOT EXISTS (
  SELECT * FROM tb_filial
  WHERE
/* %JoinFKPK(tb_filial, tb_empresa, " = ", " AND") */
tb_filial.id_empresa = tb_empresa.id_empresa
)
)
BEGIN
  SELECT @errno = 30010,
         @errmsg = 'Cannot delete last tb_filial because tb_empresa exists.'
  GOTO ERROR
END

/* ERwin Builtin Trigger */
RETURN
ERROR:
  raiserror @errno @errmsg
  rollback transaction
END

go

```

```

CREATE TRIGGER tU_tb_filial ON tb_filial FOR UPDATE AS
/* ERwin Builtin Trigger */
/* UPDATE trigger on tb_filial */
BEGIN
  DECLARE @NUMROWS int,
          @nullcnt int,
          @validcnt int,
          @insid_filial char(18),
          @insid_empresa integer,
          @errno int,
          @errmsg varchar(255)

  SELECT @NUMROWS = @@rowcount
/* ERwin Builtin Trigger */
/* tb_empresa tb_filial on child update no action */
/* ERWIN_RELATION:CHECKSUM="00014a17", PARENT_OWNER="", PARENT_TABLE="tb_empresa"
CHILD_OWNER="", CHILD_TABLE="tb_filial"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_2", FK_COLUMNS="id_empresa" */
IF
/* %ChildFK(" OR", UPDATE) */
UPDATE(id_empresa)
BEGIN
  SELECT @nullcnt = 0
  SELECT @validcnt = count(*)
  FROM inserted, tb_empresa

```

```

WHERE
    /* %JoinFKPK(inserted,tb_empresa) */
    inserted.id_empresa = tb_empresa.id_empresa
/* %NotNullFK(inserted," IS NULL","select @nullcnt = count(*) from inserted where"," AND") */

IF @validcnt + @nullcnt != @NUMROWS
BEGIN
    SELECT @errno = 30007,
           @errmsg = 'Cannot update tb_filial because tb_empresa does not exist.'
    GOTO ERROR
END
END

```

```

/* ERwin Builtin Trigger */
RETURN
ERROR:
    raiserror @errno @errmsg
    rollback transaction
END

```

go

```

CREATE TRIGGER tD_tb_log_auditoria ON tb_log_auditoria FOR DELETE AS
/* ERwin Builtin Trigger */
/* DELETE trigger on tb_log_auditoria */
BEGIN
    DECLARE @errno int,
            @errmsg varchar(255)
    /* ERwin Builtin Trigger */
    /* tb_usuario tb_log_auditoria on child delete no action */
    /* ERWIN_RELATION:CHECKSUM="00014aa9", PARENT_OWNER="", PARENT_TABLE="tb_usuario"
    CHILD_OWNER="", CHILD_TABLE="tb_log_auditoria"
    P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
    FK_CONSTRAINT="R_13", FK_COLUMNS="id_usuario" */
    IF EXISTS (SELECT * FROM deleted,tb_usuario
    WHERE
        /* %JoinFKPK(deleted,tb_usuario," = "," AND") */
        deleted.id_usuario = tb_usuario.id_usuario AND
        NOT EXISTS (
            SELECT * FROM tb_log_auditoria
            WHERE
                /* %JoinFKPK(tb_log_auditoria,tb_usuario," = "," AND") */
                tb_log_auditoria.id_usuario = tb_usuario.id_usuario
        )
    )
    BEGIN
        SELECT @errno = 30010,
               @errmsg = 'Cannot delete last tb_log_auditoria because tb_usuario exists.'
        GOTO ERROR
    END

```

```

/* ERwin Builtin Trigger */
RETURN
ERROR:
    raiserror @errno @errmsg
    rollback transaction
END

```

go

```

CREATE TRIGGER tU_tb_log_auditoria ON tb_log_auditoria FOR UPDATE AS
/* ERwin Builtin Trigger */
/* UPDATE trigger on tb_log_auditoria */
BEGIN

```



```

DECLARE @NUMROWS int,
        @nullcnt int,
        @validcnt int,
        @insid_log integer,
        @errno int,
        @errmsg varchar(255)

SELECT @NUMROWS = @@rowcount
/* ERwin Builtin Trigger */
/* tb_usuario tb_log_auditoria on child update no action */
/* ERWIN_RELATION:CHECKSUM="00017ec4", PARENT_OWNER="", PARENT_TABLE="tb_usuario"
  CHILD_OWNER="", CHILD_TABLE="tb_log_auditoria"
  P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
  FK_CONSTRAINT="R_13", FK_COLUMNS="id_usuario" */
IF
/* %ChildFK(" OR",UPDATE) */
UPDATE(id_usuario)
BEGIN
SELECT @nullcnt = 0
SELECT @validcnt = count(*)
FROM inserted,tb_usuario
WHERE
/* %JoinFKPK(inserted,tb_usuario) */
inserted.id_usuario = tb_usuario.id_usuario
/* %NotNullFK(inserted," IS NULL","select @nullcnt = count(*) from inserted where"," AND") */
select @nullcnt = count(*) from inserted where
inserted.id_usuario IS NULL
IF @validcnt + @nullcnt != @NUMROWS
BEGIN
SELECT @errno = 30007,
       @errmsg = 'Cannot update tb_log_auditoria because tb_usuario does not exist.'
GOTO ERROR
END
END

/* ERwin Builtin Trigger */
RETURN
ERROR:
raiserror @errno @errmsg
rollback transaction
END

go

```

```

CREATE TRIGGER td_tb_meio_condicao_pagamento ON tb_meio_condicao_pagamento FOR DELETE AS
/* ERwin Builtin Trigger */
/* DELETE trigger on tb_meio_condicao_pagamento */
BEGIN
DECLARE @errno int,
        @errmsg varchar(255)
/* ERwin Builtin Trigger */
/* tb_meio_condicao_pagamento tb_tipo_cliente_meio_condicao on parent delete no action */
/* ERWIN_RELATION:CHECKSUM="000645d6", PARENT_OWNER="", PARENT_TABLE="tb_meio_condicao_pagamento"
  CHILD_OWNER="", CHILD_TABLE="tb_tipo_cliente_meio_condicao"
  P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
  FK_CONSTRAINT="R_29", FK_COLUMNS="id_condicao""id_meio_pagamento""id_tipo_cliente" */
IF EXISTS (
SELECT * FROM deleted,tb_tipo_cliente_meio_condicao
WHERE
/* %JoinFKPK(tb_tipo_cliente_meio_condicao,deleted," = "," AND") */
tb_tipo_cliente_meio_condicao.id_condicao = deleted.id_condicao AND
tb_tipo_cliente_meio_condicao.id_meio_pagamento = deleted.id_meio_pagamento AND
tb_tipo_cliente_meio_condicao.id_tipo_cliente = deleted.id_tipo_cliente
)
BEGIN
SELECT @errno = 30001,

```

```

    @errmsg = 'Cannot delete tb_meio_condicao_pagamento because tb_tipo_cliente_meio_condicao exists.'
GOTO ERROR
END

```

```

/* ERwin Builtin Trigger */
/* tb_tipo_cliente tb_meio_condicao_pagamento on child delete no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_tipo_cliente"
CHILD_OWNER="", CHILD_TABLE="tb_meio_condicao_pagamento"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_27", FK_COLUMNS="id_tipo_cliente" */
IF EXISTS (SELECT * FROM deleted, tb_tipo_cliente
WHERE
/* %JoinFKPK(deleted, tb_tipo_cliente, " = ", " AND") */
deleted.id_tipo_cliente = tb_tipo_cliente.id_tipo_cliente AND
NOT EXISTS (
SELECT * FROM tb_meio_condicao_pagamento
WHERE
/* %JoinFKPK(tb_meio_condicao_pagamento, tb_tipo_cliente, " = ", " AND") */
tb_meio_condicao_pagamento.id_tipo_cliente = tb_tipo_cliente.id_tipo_cliente
)
)
BEGIN
SELECT @ermo = 30010,
    @errmsg = 'Cannot delete last tb_meio_condicao_pagamento because tb_tipo_cliente exists.'
GOTO ERROR
END

```

```

/* ERwin Builtin Trigger */
/* tb_meio_pagamento tb_meio_condicao_pagamento on child delete no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_meio_pagamento"
CHILD_OWNER="", CHILD_TABLE="tb_meio_condicao_pagamento"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_26", FK_COLUMNS="id_meio_pagamento" */
IF EXISTS (SELECT * FROM deleted, tb_meio_pagamento
WHERE
/* %JoinFKPK(deleted, tb_meio_pagamento, " = ", " AND") */
deleted.id_meio_pagamento = tb_meio_pagamento.id_meio_pagamento AND
NOT EXISTS (
SELECT * FROM tb_meio_condicao_pagamento
WHERE
/* %JoinFKPK(tb_meio_condicao_pagamento, tb_meio_pagamento, " = ", " AND") */
tb_meio_condicao_pagamento.id_meio_pagamento = tb_meio_pagamento.id_meio_pagamento
)
)
BEGIN
SELECT @ermo = 30010,
    @errmsg = 'Cannot delete last tb_meio_condicao_pagamento because tb_meio_pagamento exists.'
GOTO ERROR
END

```

```

/* ERwin Builtin Trigger */
/* tb_condicao_pagamento tb_meio_condicao_pagamento on child delete no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_condicao_pagamento"
CHILD_OWNER="", CHILD_TABLE="tb_meio_condicao_pagamento"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_25", FK_COLUMNS="id_condicao" */
IF EXISTS (SELECT * FROM deleted, tb_condicao_pagamento
WHERE
/* %JoinFKPK(deleted, tb_condicao_pagamento, " = ", " AND") */
deleted.id_condicao = tb_condicao_pagamento.id_condicao AND
NOT EXISTS (
SELECT * FROM tb_meio_condicao_pagamento
WHERE
/* %JoinFKPK(tb_meio_condicao_pagamento, tb_condicao_pagamento, " = ", " AND") */
tb_meio_condicao_pagamento.id_condicao = tb_condicao_pagamento.id_condicao
)
)
BEGIN
SELECT @ermo = 30010,
    @errmsg = 'Cannot delete last tb_meio_condicao_pagamento because tb_condicao_pagamento exists.'

```

```
GOTO ERROR
END
```

```
/* ERwin Builtin Trigger */
RETURN
ERROR:
  raiserror @errno @errmsg
  rollback transaction
END
```

go

```
CREATE TRIGGER tU_tb_meio_condicao_pagamento ON tb_meio_condicao_pagamento FOR UPDATE AS
/* ERwin Builtin Trigger */
/* UPDATE trigger on tb_meio_condicao_pagamento */
BEGIN
  DECLARE @NUMROWS int,
          @nullcnt int,
          @validcnt int,
          @insid_condicao integer,
          @insid_meio_pagamento integer,
          @insid_tipo_cliente integer,
          @errno int,
          @errmsg varchar(255)

  SELECT @NUMROWS = @@rowcount
  /* ERwin Builtin Trigger */
  /* tb_meio_condicao_pagamento tb_tipo_cliente_meio_condicao on parent update no action */
  /* ERWIN_RELATION:CHECKSUM="00065768", PARENT_OWNER="", PARENT_TABLE="tb_meio_condicao_pagamento"
  CHILD_OWNER="", CHILD_TABLE="tb_tipo_cliente_meio_condicao"
  P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
  FK_CONSTRAINT="R_29", FK_COLUMNS="id_condicao""id_meio_pagamento""id_tipo_cliente" */
  IF
    /* %ParentPK(" OR",UPDATE) */
    UPDATE(id_condicao) OR
    UPDATE(id_meio_pagamento) OR
    UPDATE(id_tipo_cliente)
  BEGIN
    IF EXISTS (
      SELECT * FROM deleted,tb_tipo_cliente_meio_condicao
      WHERE
        /* %JoinFKPK(tb_tipo_cliente_meio_condicao,deleted," = "," AND") */
        tb_tipo_cliente_meio_condicao.id_condicao = deleted.id_condicao AND
        tb_tipo_cliente_meio_condicao.id_meio_pagamento = deleted.id_meio_pagamento AND
        tb_tipo_cliente_meio_condicao.id_tipo_cliente = deleted.id_tipo_cliente
    )
    BEGIN
      SELECT @errno = 30005,
             @errmsg = 'Cannot update tb_meio_condicao_pagamento because tb_tipo_cliente_meio_condicao exists.'
      GOTO ERROR
    END
  END

  /* ERwin Builtin Trigger */
  /* tb_tipo_cliente tb_meio_condicao_pagamento on child update no action */
  /* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_tipo_cliente"
  CHILD_OWNER="", CHILD_TABLE="tb_meio_condicao_pagamento"
  P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
  FK_CONSTRAINT="R_27", FK_COLUMNS="id_tipo_cliente" */
  IF
    /* %ChildFK(" OR",UPDATE) */
    UPDATE(id_tipo_cliente)
  BEGIN
    SELECT @nullcnt = 0
    SELECT @validcnt = count(*)
    FROM inserted,tb_tipo_cliente
    WHERE
      /* %JoinFKPK(inserted,tb_tipo_cliente) */
```

```

        inserted.id_tipo_cliente = tb_tipo_cliente.id_tipo_cliente
/* %NotNullFK(inserted," IS NULL","select @nullcnt = count(*) from inserted where"," AND") */

IF @validcnt + @nullcnt != @NUMROWS
BEGIN
    SELECT @errno = 30007,
           @errmsg = 'Cannot update tb_meio_condicao_pagamento because tb_tipo_cliente does not exist.'
    GOTO ERROR
END
END

/* ERwin Builtin Trigger */
/* tb_meio_pagamento tb_meio_condicao_pagamento on child update no action */
/* ERWIN_RELATION:CHECKSUM='00000000', PARENT_OWNER='', PARENT_TABLE='tb_meio_pagamento'
  CHILD_OWNER='', CHILD_TABLE='tb_meio_condicao_pagamento'
  P2C_VERB_PHRASE='', C2P_VERB_PHRASE='',
  FK_CONSTRAINT='R_26', FK_COLUMNS='id_meio_pagamento' */
IF
/* %ChildFK(" OR",UPDATE) */
UPDATE(id_meio_pagamento)
BEGIN
    SELECT @nullcnt = 0
    SELECT @validcnt = count(*)
    FROM inserted,tb_meio_pagamento
    WHERE
/* %JoinFKPK(inserted,tb_meio_pagamento) */
        inserted.id_meio_pagamento = tb_meio_pagamento.id_meio_pagamento
/* %NotNullFK(inserted," IS NULL","select @nullcnt = count(*) from inserted where"," AND") */

IF @validcnt + @nullcnt != @NUMROWS
BEGIN
    SELECT @errno = 30007,
           @errmsg = 'Cannot update tb_meio_condicao_pagamento because tb_meio_pagamento does not exist.'
    GOTO ERROR
END
END

/* ERwin Builtin Trigger */
/* tb_condicao_pagamento tb_meio_condicao_pagamento on child update no action */
/* ERWIN_RELATION:CHECKSUM='00000000', PARENT_OWNER='', PARENT_TABLE='tb_condicao_pagamento'
  CHILD_OWNER='', CHILD_TABLE='tb_meio_condicao_pagamento'
  P2C_VERB_PHRASE='', C2P_VERB_PHRASE='',
  FK_CONSTRAINT='R_25', FK_COLUMNS='id_condicao' */
IF
/* %ChildFK(" OR",UPDATE) */
UPDATE(id_condicao)
BEGIN
    SELECT @nullcnt = 0
    SELECT @validcnt = count(*)
    FROM inserted,tb_condicao_pagamento
    WHERE
/* %JoinFKPK(inserted,tb_condicao_pagamento) */
        inserted.id_condicao = tb_condicao_pagamento.id_condicao
/* %NotNullFK(inserted," IS NULL","select @nullcnt = count(*) from inserted where"," AND") */

IF @validcnt + @nullcnt != @NUMROWS
BEGIN
    SELECT @errno = 30007,
           @errmsg = 'Cannot update tb_meio_condicao_pagamento because tb_condicao_pagamento does not exist.'
    GOTO ERROR
END
END

/* ERwin Builtin Trigger */
RETURN
ERROR:
    raiserror @errno @errmsg
    rollback transaction
END

```

go

```
CREATE TRIGGER tD_tb_meio_pagamento ON tb_meio_pagamento FOR DELETE AS
/* ERwin Builtin Trigger */
/* DELETE trigger on tb_meio_pagamento */
BEGIN
  DECLARE @ermo int,
          @errmsg varchar(255)
  /* ERwin Builtin Trigger */
  /* tb_meio_pagamento tb_meio_condicao_pagamento on parent delete no action */
  /* ERWIN_RELATION:CHECKSUM="00012e07", PARENT_OWNER="", PARENT_TABLE="tb_meio_pagamento"
  CHILD_OWNER="", CHILD_TABLE="tb_meio_condicao_pagamento"
  P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
  FK_CONSTRAINT="R_26", FK_COLUMNS="id_meio_pagamento" */
  IF EXISTS (
    SELECT * FROM deleted, tb_meio_condicao_pagamento
    WHERE
      /* %JoinFKPK(tb_meio_condicao_pagamento,deleted,"="," AND") */
      tb_meio_condicao_pagamento.id_meio_pagamento = deleted.id_meio_pagamento
  )
  BEGIN
    SELECT @ermo = 30001,
           @errmsg = 'Cannot delete tb_meio_pagamento because tb_meio_condicao_pagamento exists.'
    GOTO ERROR
  END

  /* ERwin Builtin Trigger */
  RETURN
ERROR:
  raiserror @ermo @errmsg
  rollback transaction
END
```

go

```
CREATE TRIGGER tU_tb_meio_pagamento ON tb_meio_pagamento FOR UPDATE AS
/* ERwin Builtin Trigger */
/* UPDATE trigger on tb_meio_pagamento */
BEGIN
  DECLARE @NUMROWS int,
          @nullcnt int,
          @validcnt int,
          @insid_meio_pagamento integer,
          @ermo int,
          @errmsg varchar(255)

  SELECT @NUMROWS = @@rowcount
  /* ERwin Builtin Trigger */
  /* tb_meio_pagamento tb_meio_condicao_pagamento on parent update no action */
  /* ERWIN_RELATION:CHECKSUM="00014024", PARENT_OWNER="", PARENT_TABLE="tb_meio_pagamento"
  CHILD_OWNER="", CHILD_TABLE="tb_meio_condicao_pagamento"
  P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
  FK_CONSTRAINT="R_26", FK_COLUMNS="id_meio_pagamento" */
  IF
    /* %ParentPK(" OR",UPDATE) */
    UPDATE(id_meio_pagamento)
  BEGIN
    IF EXISTS (
      SELECT * FROM deleted, tb_meio_condicao_pagamento
      WHERE
        /* %JoinFKPK(tb_meio_condicao_pagamento,deleted,"="," AND") */
        tb_meio_condicao_pagamento.id_meio_pagamento = deleted.id_meio_pagamento
    )
  BEGIN
```

```

SELECT @errno = 30005,
       @errmsg = 'Cannot update tb_meio_pagamento because tb_meio_condicao_pagamento exists.'
GOTO ERROR
END
END

```

```

/* ERwin Builtin Trigger */
RETURN
ERROR:
    raiserror @errno @errmsg
    rollback transaction
END

```

go

```

CREATE TRIGGER tD_tb_modulos ON tb_modulos FOR DELETE AS
/* ERwin Builtin Trigger */
/* DELETE trigger on tb_modulos */
BEGIN
    DECLARE @errno int,
            @errmsg varchar(255)
    /* ERwin Builtin Trigger */
    /* tb_modulos tb_estrutura_sistema on parent delete no action */
    /* ERWIN_RELATION:CHECKSUM="00011762", PARENT_OWNER="", PARENT_TABLE="tb_modulos"
    CHILD_OWNER="", CHILD_TABLE="tb_estrutura_sistema"
    P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
    FK_CONSTRAINT="R_8", FK_COLUMNS="id_modulo" */
    IF EXISTS (
        SELECT * FROM deleted, tb_estrutura_sistema
        WHERE
            /* %JoinFKPK(tb_estrutura_sistema,deleted," = ","AND") */
            tb_estrutura_sistema.id_modulo = deleted.id_modulo
    )
    BEGIN
        SELECT @errno = 30001,
               @errmsg = 'Cannot delete tb_modulos because tb_estrutura_sistema exists.'
        GOTO ERROR
    END

```

```

/* ERwin Builtin Trigger */
RETURN
ERROR:
    raiserror @errno @errmsg
    rollback transaction
END

```

go

```

CREATE TRIGGER tU_tb_modulos ON tb_modulos FOR UPDATE AS
/* ERwin Builtin Trigger */
/* UPDATE trigger on tb_modulos */
BEGIN
    DECLARE @NUMROWS int,
            @nullcnt int,
            @validcnt int,
            @insid_modulo integer,
            @errno int,
            @errmsg varchar(255)

    SELECT @NUMROWS = @@rowcount
    /* ERwin Builtin Trigger */
    /* tb_modulos tb_estrutura_sistema on parent update no action */
    /* ERWIN_RELATION:CHECKSUM="0001230b", PARENT_OWNER="", PARENT_TABLE="tb_modulos"
    CHILD_OWNER="", CHILD_TABLE="tb_estrutura_sistema"

```

```

P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_8", FK_COLUMNS="id_modulo" */
IF
/* %ParentPK(" OR",UPDATE) */
UPDATE(id_modulo)
BEGIN
IF EXISTS (
SELECT * FROM deleted,tb_estrutura_sistema
WHERE
/* %JoinFKPK(tb_estrutura_sistema,deleted,"=","AND") */
tb_estrutura_sistema.id_modulo = deleted.id_modulo
)
BEGIN
SELECT @errno = 30005,
@errmsg = 'Cannot update tb_modulos because tb_estrutura_sistema exists.'
GOTO ERROR
END
END

```

```

/* ERwin Builtin Trigger */
RETURN
ERROR:
raiserror @errno @errmsg
rollback transaction
END

```

go

```

CREATE TRIGGER tD_tb_produto ON tb_produto FOR DELETE AS
/* ERwin Builtin Trigger */
/* DELETE trigger on tb_produto */
BEGIN
DECLARE @errno int,
@errmsg varchar(255)
/* ERwin Builtin Trigger */
/* tb_produto tb_produto_insumo_estoque on parent delete no action */
/* ERWIN_RELATION:CHECKSUM="0009bf12", PARENT_OWNER="", PARENT_TABLE="tb_produto"
CHILD_OWNER="", CHILD_TABLE="tb_produto_insumo_estoque"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_45", FK_COLUMNS="id_produto_insumo" */
IF EXISTS (
SELECT * FROM deleted,tb_produto_insumo_estoque
WHERE
/* %JoinFKPK(tb_produto_insumo_estoque,deleted,"=","AND") */
tb_produto_insumo_estoque.id_produto_insumo = deleted.id_produto
)
BEGIN
SELECT @errno = 30001,
@errmsg = 'Cannot delete tb_produto because tb_produto_insumo_estoque exists.'
GOTO ERROR
END

/* ERwin Builtin Trigger */
/* tb_produto tb_produto_acabado_estoque on parent delete no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_produto"
CHILD_OWNER="", CHILD_TABLE="tb_produto_acabado_estoque"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_41", FK_COLUMNS="id_produto_acabado" */
IF EXISTS (
SELECT * FROM deleted,tb_produto_acabado_estoque
WHERE
/* %JoinFKPK(tb_produto_acabado_estoque,deleted,"=","AND") */
tb_produto_acabado_estoque.id_produto_acabado = deleted.id_produto
)
BEGIN
SELECT @errno = 30001,

```

```

        @errmsg = 'Cannot delete tb_produto because tb_produto_acabado_estoque exists.'
    GOTO ERROR
END

```

```

/* ERwin Builtin Trigger */
/* tb_produto tb_produto_tabela_preco on parent delete no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_produto"
CHILD_OWNER="", CHILD_TABLE="tb_produto_tabela_preco"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_39", FK_COLUMNS="id_produto_acabado" */
IF EXISTS (
    SELECT * FROM deleted,tb_produto_tabela_preco
    WHERE
        /* %JoinFKPK(tb_produto_tabela_preco,deleted," = ","AND") */
        tb_produto_tabela_preco.id_produto_acabado = deleted.id_produto
)
BEGIN
    SELECT @errno = 30001,
        @errmsg = 'Cannot delete tb_produto because tb_produto_tabela_preco exists.'
    GOTO ERROR
END

```

```

/* ERwin Builtin Trigger */
/* tb_produto tb_produto_roteiro_producao on parent delete no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_produto"
CHILD_OWNER="", CHILD_TABLE="tb_produto_roteiro_producao"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_35", FK_COLUMNS="id_produto_acabado" */
IF EXISTS (
    SELECT * FROM deleted,tb_produto_roteiro_producao
    WHERE
        /* %JoinFKPK(tb_produto_roteiro_producao,deleted," = ","AND") */
        tb_produto_roteiro_producao.id_produto_acabado = deleted.id_produto
)
BEGIN
    SELECT @errno = 30001,
        @errmsg = 'Cannot delete tb_produto because tb_produto_roteiro_producao exists.'
    GOTO ERROR
END

```

```

/* ERwin Builtin Trigger */
/* tb_produto tb_produto_ficha_tecnica on parent delete no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_produto"
CHILD_OWNER="", CHILD_TABLE="tb_produto_ficha_tecnica"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_33", FK_COLUMNS="id_produto_insumo" */
IF EXISTS (
    SELECT * FROM deleted,tb_produto_ficha_tecnica
    WHERE
        /* %JoinFKPK(tb_produto_ficha_tecnica,deleted," = ","AND") */
        tb_produto_ficha_tecnica.id_produto_insumo = deleted.id_produto
)
BEGIN
    SELECT @errno = 30001,
        @errmsg = 'Cannot delete tb_produto because tb_produto_ficha_tecnica exists.'
    GOTO ERROR
END

```

```

/* ERwin Builtin Trigger */
/* tb_produto tb_produto_ficha_tecnica on parent delete no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_produto"
CHILD_OWNER="", CHILD_TABLE="tb_produto_ficha_tecnica"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_32", FK_COLUMNS="id_produto_acabado" */
IF EXISTS (
    SELECT * FROM deleted,tb_produto_ficha_tecnica
    WHERE
        /* %JoinFKPK(tb_produto_ficha_tecnica,deleted," = ","AND") */
        tb_produto_ficha_tecnica.id_produto_acabado = deleted.id_produto
)

```



```

BEGIN
  SELECT @errno = 30001,
         @errmsg = 'Cannot delete tb_produto because tb_produto_ficha_tecnica exists.'
  GOTO ERROR
END

/* ERwin Builtin Trigger */
/* tb_unidade_medida tb_produto on child delete no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_unidade_medida"
CHILD_OWNER="", CHILD_TABLE="tb_produto"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_31", FK_COLUMNS="id_unidade_medida" */
IF EXISTS (SELECT * FROM deleted,tb_unidade_medida
WHERE
/* %JoinFKPK(deleted,tb_unidade_medida,"=","AND") */
deleted.id_unidade_medida = tb_unidade_medida.id_unidade_medida AND
NOT EXISTS (
  SELECT * FROM tb_produto
  WHERE
/* %JoinFKPK(tb_produto,tb_unidade_medida,"=","AND") */
tb_produto.id_unidade_medida = tb_unidade_medida.id_unidade_medida
)
)
BEGIN
  SELECT @errno = 30010,
         @errmsg = 'Cannot delete last tb_produto because tb_unidade_medida exists.'
  GOTO ERROR
END

/* ERwin Builtin Trigger */
/* tb_tipo_produto tb_produto on child delete no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_tipo_produto"
CHILD_OWNER="", CHILD_TABLE="tb_produto"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_30", FK_COLUMNS="id_tipo_produto" */
IF EXISTS (SELECT * FROM deleted,tb_tipo_produto
WHERE
/* %JoinFKPK(deleted,tb_tipo_produto,"=","AND") */
deleted.id_tipo_produto = tb_tipo_produto.id_tipo_produto AND
NOT EXISTS (
  SELECT * FROM tb_produto
  WHERE
/* %JoinFKPK(tb_produto,tb_tipo_produto,"=","AND") */
tb_produto.id_tipo_produto = tb_tipo_produto.id_tipo_produto
)
)
BEGIN
  SELECT @errno = 30010,
         @errmsg = 'Cannot delete last tb_produto because tb_tipo_produto exists.'
  GOTO ERROR
END

/* ERwin Builtin Trigger */
RETURN
ERROR:
  raiserror @errno @errmsg
  rollback transaction
END

```

go

```

CREATE TRIGGER tU_tb_produto ON tb_produto FOR UPDATE AS
/* ERwin Builtin Trigger */
/* UPDATE trigger on tb_produto */
BEGIN
  DECLARE @NUMROWS int,
          @nullcnt int,
          @validcnt int,

```

```
@insid_produto integer,  
@errno int,  
@errmsg varchar(255)
```

```
SELECT @NUMROWS = @@rowcount
```

```
/* ERwin Builtin Trigger */
```

```
/* tb_produto tb_produto_insumo_estoque on parent update no action */
```

```
/* ERWIN_RELATION:CHECKSUM="000a9bad", PARENT_OWNER="", PARENT_TABLE="tb_produto"
```

```
CHILD_OWNER="", CHILD_TABLE="tb_produto_insumo_estoque"
```

```
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
```

```
FK_CONSTRAINT="R_45", FK_COLUMNS="id_produto_insumo" */
```

```
IF
```

```
/* %ParentPK(" OR",UPDATE) */
```

```
UPDATE(id_produto)
```

```
BEGIN
```

```
IF EXISTS (
```

```
SELECT * FROM deleted,tb_produto_insumo_estoque
```

```
WHERE
```

```
/* %JoinFKPK(tb_produto_insumo_estoque,deleted," = ","AND") */
```

```
tb_produto_insumo_estoque.id_produto_insumo = deleted.id_produto
```

```
)
```

```
BEGIN
```

```
SELECT @errno = 30005,
```

```
@errmsg = 'Cannot update tb_produto because tb_produto_insumo_estoque exists.'
```

```
GOTO ERROR
```

```
END
```

```
END
```

```
/* ERwin Builtin Trigger */
```

```
/* tb_produto tb_produto_acabado_estoque on parent update no action */
```

```
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_produto"
```

```
CHILD_OWNER="", CHILD_TABLE="tb_produto_acabado_estoque"
```

```
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
```

```
FK_CONSTRAINT="R_41", FK_COLUMNS="id_produto_acabado" */
```

```
IF
```

```
/* %ParentPK(" OR",UPDATE) */
```

```
UPDATE(id_produto)
```

```
BEGIN
```

```
IF EXISTS (
```

```
SELECT * FROM deleted,tb_produto_acabado_estoque
```

```
WHERE
```

```
/* %JoinFKPK(tb_produto_acabado_estoque,deleted," = ","AND") */
```

```
tb_produto_acabado_estoque.id_produto_acabado = deleted.id_produto
```

```
)
```

```
BEGIN
```

```
SELECT @errno = 30005,
```

```
@errmsg = 'Cannot update tb_produto because tb_produto_acabado_estoque exists.'
```

```
GOTO ERROR
```

```
END
```

```
END
```

```
/* ERwin Builtin Trigger */
```

```
/* tb_produto tb_produto_tabela_preco on parent update no action */
```

```
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_produto"
```

```
CHILD_OWNER="", CHILD_TABLE="tb_produto_tabela_preco"
```

```
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
```

```
FK_CONSTRAINT="R_39", FK_COLUMNS="id_produto_acabado" */
```

```
IF
```

```
/* %ParentPK(" OR",UPDATE) */
```

```
UPDATE(id_produto)
```

```
BEGIN
```

```
IF EXISTS (
```

```
SELECT * FROM deleted,tb_produto_tabela_preco
```

```
WHERE
```

```
/* %JoinFKPK(tb_produto_tabela_preco,deleted," = ","AND") */
```

```
tb_produto_tabela_preco.id_produto_acabado = deleted.id_produto
```

```
)
```

```
BEGIN
```

```
SELECT @errno = 30005,
```

```
@errmsg = 'Cannot update tb_produto because tb_produto_tabela_preco exists.'
```

```
GOTO ERROR
END
END
```

```
/* ERwin Builtin Trigger */
/* tb_produto tb_produtoroteiro_producao on parent update no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_produto"
CHILD_OWNER="", CHILD_TABLE="tb_produtoroteiro_producao"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_35", FK_COLUMNS="id_produto_acabado" */
IF
/* %ParentPK(" OR",UPDATE) */
UPDATE(id_produto)
BEGIN
IF EXISTS (
SELECT * FROM deleted,tb_produtoroteiro_producao
WHERE
/* %JoinFKPK(tb_produtoroteiro_producao,deleted," = "," AND") */
tb_produtoroteiro_producao.id_produto_acabado = deleted.id_produto
)
BEGIN
SELECT @errno = 30005,
@errmsg = 'Cannot update tb_produto because tb_produtoroteiro_producao exists.'
GOTO ERROR
END
END
```

```
/* ERwin Builtin Trigger */
/* tb_produto tb_produto_ficha_tecnica on parent update no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_produto"
CHILD_OWNER="", CHILD_TABLE="tb_produto_ficha_tecnica"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_33", FK_COLUMNS="id_produto_insumo" */
IF
/* %ParentPK(" OR",UPDATE) */
UPDATE(id_produto)
BEGIN
IF EXISTS (
SELECT * FROM deleted,tb_produto_ficha_tecnica
WHERE
/* %JoinFKPK(tb_produto_ficha_tecnica,deleted," = "," AND") */
tb_produto_ficha_tecnica.id_produto_insumo = deleted.id_produto
)
BEGIN
SELECT @errno = 30005,
@errmsg = 'Cannot update tb_produto because tb_produto_ficha_tecnica exists.'
GOTO ERROR
END
END
```

```
/* ERwin Builtin Trigger */
/* tb_produto tb_produto_ficha_tecnica on parent update no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_produto"
CHILD_OWNER="", CHILD_TABLE="tb_produto_ficha_tecnica"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_32", FK_COLUMNS="id_produto_acabado" */
IF
/* %ParentPK(" OR",UPDATE) */
UPDATE(id_produto)
BEGIN
IF EXISTS (
SELECT * FROM deleted,tb_produto_ficha_tecnica
WHERE
/* %JoinFKPK(tb_produto_ficha_tecnica,deleted," = "," AND") */
tb_produto_ficha_tecnica.id_produto_acabado = deleted.id_produto
)
BEGIN
SELECT @errno = 30005,
@errmsg = 'Cannot update tb_produto because tb_produto_ficha_tecnica exists.'
GOTO ERROR
```

```

END
END

/* ERwin Builtin Trigger */
/* tb_unidade_medida tb_produto on child update no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_unidade_medida"
CHILD_OWNER="", CHILD_TABLE="tb_produto"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_31", FK_COLUMNS="id_unidade_medida" */
IF
/* %ChildFK(" OR",UPDATE) */
UPDATE(id_unidade_medida)
BEGIN
SELECT @nullcnt = 0
SELECT @validcnt = count(*)
FROM inserted,tb_unidade_medida
WHERE
/* %JoinFKPK(inserted,tb_unidade_medida) */
inserted.id_unidade_medida = tb_unidade_medida.id_unidade_medida
/* %NotNullFK(inserted," IS NULL","select @nullcnt = count(*) from inserted where"," AND") */
select @nullcnt = count(*) from inserted where
inserted.id_unidade_medida IS NULL
IF @validcnt + @nullcnt != @NUMROWS
BEGIN
SELECT @errno = 30007,
@errmsg = 'Cannot update tb_produto because tb_unidade_medida does not exist.'
GOTO ERROR
END
END

/* ERwin Builtin Trigger */
/* tb_tipo_produto tb_produto on child update no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_tipo_produto"
CHILD_OWNER="", CHILD_TABLE="tb_produto"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_30", FK_COLUMNS="id_tipo_produto" */
IF
/* %ChildFK(" OR",UPDATE) */
UPDATE(id_tipo_produto)
BEGIN
SELECT @nullcnt = 0
SELECT @validcnt = count(*)
FROM inserted,tb_tipo_produto
WHERE
/* %JoinFKPK(inserted,tb_tipo_produto) */
inserted.id_tipo_produto = tb_tipo_produto.id_tipo_produto
/* %NotNullFK(inserted," IS NULL","select @nullcnt = count(*) from inserted where"," AND") */
IF @validcnt + @nullcnt != @NUMROWS
BEGIN
SELECT @errno = 30007,
@errmsg = 'Cannot update tb_produto because tb_tipo_produto does not exist.'
GOTO ERROR
END
END

/* ERwin Builtin Trigger */
RETURN
ERROR:
raiserror @errno @errmsg
rollback transaction
END

```

go

CREATE TRIGGER tD\_tb\_produto\_acabado\_estoque ON tb\_produto\_acabado\_estoque FOR DELETE AS

```

/* ERwin Builtin Trigger */
/* DELETE trigger on tb_produto_acabado_estoque */
BEGIN
  DECLARE @ermo int,
          @errmsg varchar(255)
/* ERwin Builtin Trigger */
/* tb_produto_acabado_estoque tb_produto_acabado_movimentacao_estoque on parent delete no action */
/* ERWIN_RELATION:CHECKSUM="00044ddd", PARENT_OWNER="", PARENT_TABLE="tb_produto_acabado_estoque"
CHILD_OWNER="", CHILD_TABLE="tb_produto_acabado_movimentacao_estoque"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_43", FK_COLUMNS="id_poduto_acabado_estoque" */
IF EXISTS (
  SELECT * FROM deleted,tb_produto_acabado_movimentacao_estoque
  WHERE
    /* %JoinFKPK(tb_produto_acabado_movimentacao_estoque,deleted,"=","AND") */
    tb_produto_acabado_movimentacao_estoque.id_poduto_acabado_estoque = deleted.id_poduto_acabado_estoque
)
BEGIN
  SELECT @ermo = 30001,
         @errmsg = 'Cannot delete tb_produto_acabado_estoque because tb_produto_acabado_movimentacao_estoque exists.'
  GOTO ERROR
END

/* ERwin Builtin Trigger */
/* tb_unidade_medida tb_produto_acabado_estoque on child delete no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_unidade_medida"
CHILD_OWNER="", CHILD_TABLE="tb_produto_acabado_estoque"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_42", FK_COLUMNS="id_unidade_medida" */
IF EXISTS (SELECT * FROM deleted,tb_unidade_medida
  WHERE
    /* %JoinFKPK(deleted,tb_unidade_medida,"=","AND") */
    deleted.id_unidade_medida = tb_unidade_medida.id_unidade_medida AND
  NOT EXISTS (
    SELECT * FROM tb_produto_acabado_estoque
    WHERE
      /* %JoinFKPK(tb_produto_acabado_estoque,tb_unidade_medida,"=","AND") */
      tb_produto_acabado_estoque.id_unidade_medida = tb_unidade_medida.id_unidade_medida
    )
)
BEGIN
  SELECT @ermo = 30010,
         @errmsg = 'Cannot delete last tb_produto_acabado_estoque because tb_unidade_medida exists.'
  GOTO ERROR
END

/* ERwin Builtin Trigger */
/* tb_produto tb_produto_acabado_estoque on child delete no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_produto"
CHILD_OWNER="", CHILD_TABLE="tb_produto_acabado_estoque"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_41", FK_COLUMNS="id_produto_acabado" */
IF EXISTS (SELECT * FROM deleted,tb_produto
  WHERE
    /* %JoinFKPK(deleted,tb_produto,"=","AND") */
    deleted.id_produto_acabado = tb_produto.id_produto AND
  NOT EXISTS (
    SELECT * FROM tb_produto_acabado_estoque
    WHERE
      /* %JoinFKPK(tb_produto_acabado_estoque,tb_produto,"=","AND") */
      tb_produto_acabado_estoque.id_produto_acabado = tb_produto.id_produto
    )
)
BEGIN
  SELECT @ermo = 30010,
         @errmsg = 'Cannot delete last tb_produto_acabado_estoque because tb_produto exists.'
  GOTO ERROR
END

```

```

/* ERwin Builtin Trigger */
RETURN
ERROR:
    raiserror @errno @errmsg
    rollback transaction
END

```

go

```

CREATE TRIGGER tU_tb_produto_acabado_estoque ON tb_produto_acabado_estoque FOR UPDATE AS
/* ERwin Builtin Trigger */
/* UPDATE trigger on tb_produto_acabado_estoque */
BEGIN
    DECLARE @NUMROWS int,
            @nullcnt int,
            @validcnt int,
            @insid_poduto_acabado_estoque integer,
            @errno int,
            @errmsg varchar(255)

    SELECT @NUMROWS = @@rowcount
    /* ERwin Builtin Trigger */
    /* tb_produto_acabado_estoque tb_produto_acabado_movimentacao_estoque on parent update no action */
    /* ERWIN_RELATION:CHECKSUM="0004c882", PARENT_OWNER="", PARENT_TABLE="tb_produto_acabado_estoque"
    CHILD_OWNER="", CHILD_TABLE="tb_produto_acabado_movimentacao_estoque"
    P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
    FK_CONSTRAINT="R_43", FK_COLUMNS="id_poduto_acabado_estoque" */
    IF
        /* %ParentPK(" OR",UPDATE) */
        UPDATE(id_poduto_acabado_estoque)
    BEGIN
        IF EXISTS (
            SELECT * FROM deleted,tb_produto_acabado_movimentacao_estoque
            WHERE
                /* %JoinFKPK(tb_produto_acabado_movimentacao_estoque,deleted,"=","AND") */
                tb_produto_acabado_movimentacao_estoque.id_poduto_acabado_estoque = deleted.id_poduto_acabado_estoque
        )
        BEGIN
            SELECT @errno = 30005,
                   @errmsg = 'Cannot update tb_produto_acabado_estoque because tb_produto_acabado_movimentacao_estoque exists.'
            GOTO ERROR
        END
    END
END

/* ERwin Builtin Trigger */
/* tb_unidade_medida tb_produto_acabado_estoque on child update no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_unidade_medida"
    CHILD_OWNER="", CHILD_TABLE="tb_produto_acabado_estoque"
    P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
    FK_CONSTRAINT="R_42", FK_COLUMNS="id_unidade_medida" */
    IF
        /* %ChildFK(" OR",UPDATE) */
        UPDATE(id_unidade_medida)
    BEGIN
        SELECT @nullcnt = 0
        SELECT @validcnt = count(*)
            FROM inserted,tb_unidade_medida
            WHERE
                /* %JoinFKPK(inserted,tb_unidade_medida) */
                inserted.id_unidade_medida = tb_unidade_medida.id_unidade_medida
        /* %NotNullFK(inserted,"IS NULL","select @nullcnt = count(*) from inserted where","AND") */
        select @nullcnt = count(*) from inserted where
            inserted.id_unidade_medida IS NULL
        IF @validcnt + @nullcnt != @NUMROWS
        BEGIN
            SELECT @errno = 30007,
                   @errmsg = 'Cannot update tb_produto_acabado_estoque because tb_unidade_medida does not exist.'
            GOTO ERROR
        END
    END

```

END

```
/* ERwin Builtin Trigger */
/* tb_produto tb_produto_acabado_estoque on child update no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_produto"
  CHILD_OWNER="", CHILD_TABLE="tb_produto_acabado_estoque"
  P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
  FK_CONSTRAINT="R_41", FK_COLUMNS="id_produto_acabado" */
IF
/* %ChildFK(" OR",UPDATE) */
UPDATE(id_produto_acabado)
BEGIN
SELECT @nullcnt = 0
SELECT @validcnt = count(*)
FROM inserted,tb_produto
WHERE
/* %JoinFKPK(inserted,tb_produto) */
inserted.id_produto_acabado = tb_produto.id_produto
/* %NotNullFK(inserted," IS NULL","select @nullcnt = count(*) from inserted where"," AND") */
select @nullcnt = count(*) from inserted where
inserted.id_produto_acabado IS NULL
IF @validcnt + @nullcnt != @NUMROWS
BEGIN
SELECT @errno = 30007,
@errmsg = 'Cannot update tb_produto_acabado_estoque because tb_produto does not exist.'
GOTO ERROR
END
END
```

```
/* ERwin Builtin Trigger */
RETURN
ERROR:
raiserror @errno @errmsg
rollback transaction
END
```

go

CREATE TRIGGER td\_tb\_produto\_acabado\_movimentacao\_estoque ON tb\_produto\_acabado\_movimentacao\_estoque FOR DELETE AS

```
/* ERwin Builtin Trigger */
/* DELETE trigger on tb_produto_acabado_movimentacao_estoque */
BEGIN
DECLARE @errno int,
@errmsg varchar(255)
/* ERwin Builtin Trigger */
/* tb_tipo_movimentacao_estoque tb_produto_acabado_movimentacao_estoque on child delete no action */
/* ERWIN_RELATION:CHECKSUM="0003eed7", PARENT_OWNER="", PARENT_TABLE="tb_tipo_movimentacao_estoque"
  CHILD_OWNER="", CHILD_TABLE="tb_produto_acabado_movimentacao_estoque"
  P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
  FK_CONSTRAINT="R_44", FK_COLUMNS="id_tipo_movimentacao_estoque" */
IF EXISTS (SELECT * FROM deleted,tb_tipo_movimentacao_estoque
WHERE
/* %JoinFKPK(deleted,tb_tipo_movimentacao_estoque," = "," AND") */
deleted.id_tipo_movimentacao_estoque = tb_tipo_movimentacao_estoque.id_tipo_movimentacao_estoque AND
NOT EXISTS (
SELECT * FROM tb_produto_acabado_movimentacao_estoque
WHERE
/* %JoinFKPK(tb_produto_acabado_movimentacao_estoque,tb_tipo_movimentacao_estoque," = "," AND") */
tb_produto_acabado_movimentacao_estoque.id_tipo_movimentacao_estoque = tb_tipo_movimentacao_estoque.id_tipo_movimentacao_estoque
)
)
BEGIN
SELECT @errno = 30010,
@errmsg = 'Cannot delete last tb_produto_acabado_movimentacao_estoque because tb_tipo_movimentacao_estoque exists.'
GOTO ERROR
END
```

```

/* ERwin Builtin Trigger */
/* tb_produto_acabado_estoque tb_produto_acabado_movimentacao_estoque on child delete no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_produto_acabado_estoque"
CHILD_OWNER="", CHILD_TABLE="tb_produto_acabado_movimentacao_estoque"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_43", FK_COLUMNS="id_poduto_acabado_estoque" */
IF EXISTS (SELECT * FROM deleted,tb_produto_acabado_estoque
WHERE
/* %JoinFKPK(deleted,tb_produto_acabado_estoque,"=","AND") */
deleted.id_poduto_acabado_estoque = tb_produto_acabado_estoque.id_poduto_acabado_estoque AND
NOT EXISTS (
SELECT * FROM tb_produto_acabado_movimentacao_estoque
WHERE
/* %JoinFKPK(tb_produto_acabado_movimentacao_estoque,tb_produto_acabado_estoque,"=","AND") */
tb_produto_acabado_movimentacao_estoque.id_poduto_acabado_estoque = tb_produto_acabado_estoque.id_poduto_acabado_estoque
)
)
BEGIN
SELECT @errno = 30010,
       @errmsg = 'Cannot delete last tb_produto_acabado_movimentacao_estoque because tb_produto_acabado_estoque exists.'
GOTO ERROR
END

/* ERwin Builtin Trigger */
RETURN
ERROR:
raiserror @errno @errmsg
rollback transaction
END

go

```

CREATE TRIGGER tU\_tb\_produto\_acabado\_movimentacao\_estoque ON tb\_produto\_acabado\_movimentacao\_estoque FOR UPDATE AS

```

/* ERwin Builtin Trigger */
/* UPDATE trigger on tb_produto_acabado_movimentacao_estoque */
BEGIN
DECLARE @NUMROWS int,
        @nullcnt int,
        @validcnt int,
        @insid_movimentacao integer,
        @errno int,
        @errmsg varchar(255)

SELECT @NUMROWS = @@rowcount
/* ERwin Builtin Trigger */
/* tb_tipo_movimentacao_estoque tb_produto_acabado_movimentacao_estoque on child update no action */
/* ERWIN_RELATION:CHECKSUM="0003af89", PARENT_OWNER="", PARENT_TABLE="tb_tipo_movimentacao_estoque"
CHILD_OWNER="", CHILD_TABLE="tb_produto_acabado_movimentacao_estoque"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_44", FK_COLUMNS="id_tipo_movimentacao_estoque" */
IF
/* %ChildFK("OR",UPDATE) */
UPDATE(id_tipo_movimentacao_estoque)
BEGIN
SELECT @nullcnt = 0
SELECT @validcnt = count(*)
FROM inserted,tb_tipo_movimentacao_estoque
WHERE
/* %JoinFKPK(inserted,tb_tipo_movimentacao_estoque) */
inserted.id_tipo_movimentacao_estoque = tb_tipo_movimentacao_estoque.id_tipo_movimentacao_estoque
/* %NotNullFK(inserted,"IS NULL","select @nullcnt = count(*) from inserted where","AND") */
select @nullcnt = count(*) from inserted where
inserted.id_tipo_movimentacao_estoque IS NULL
IF @validcnt + @nullcnt != @NUMROWS
BEGIN
SELECT @errno = 30007,
       @errmsg = 'Cannot update tb_produto_acabado_movimentacao_estoque because tb_tipo_movimentacao_estoque does not exist.'

```



```

GOTO ERROR
END
END

/* ERwin Builtin Trigger */
/* tb_produto_acabado_estoque tb_produto_acabado_movimentacao_estoque on child update no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_produto_acabado_estoque"
CHILD_OWNER="", CHILD_TABLE="tb_produto_acabado_movimentacao_estoque"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_43", FK_COLUMNS="id_poduto_acabado_estoque" */
IF
/* %ChildFK(" OR",UPDATE) */
UPDATE(id_poduto_acabado_estoque)
BEGIN
SELECT @nullcnt = 0
SELECT @validcnt = count(*)
FROM inserted,tb_produto_acabado_estoque
WHERE
/* %JoinFKPK(inserted,tb_produto_acabado_estoque) */
inserted.id_poduto_acabado_estoque = tb_produto_acabado_estoque.id_poduto_acabado_estoque
/* %NotnullFK(inserted," IS NULL","select @nullcnt = count(*) from inserted where","AND") */

IF @validcnt + @nullcnt != @NUMROWS
BEGIN
SELECT @errno = 30007,
@errmsg = 'Cannot update tb_produto_acabado_movimentacao_estoque because tb_produto_acabado_estoque does not exist.'
GOTO ERROR
END
END

/* ERwin Builtin Trigger */
RETURN
ERROR:
raiserror @errno @errmsg
rollback transaction
END

go

```

```

CREATE TRIGGER tD_tb_produto_ficha_tecnica ON tb_produto_ficha_tecnica FOR DELETE AS
/* ERwin Builtin Trigger */
/* DELETE trigger on tb_produto_ficha_tecnica */
BEGIN
DECLARE @errno int,
@errmsg varchar(255)
/* ERwin Builtin Trigger */
/* tb_unidade_medida tb_produto_ficha_tecnica on child delete no action */
/* ERWIN_RELATION:CHECKSUM="00045bde", PARENT_OWNER="", PARENT_TABLE="tb_unidade_medida"
CHILD_OWNER="", CHILD_TABLE="tb_produto_ficha_tecnica"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_34", FK_COLUMNS="id_unidade_medida" */
IF EXISTS (SELECT * FROM deleted,tb_unidade_medida
WHERE
/* %JoinFKPK(deleted,tb_unidade_medida," = ","AND") */
deleted.id_unidade_medida = tb_unidade_medida.id_unidade_medida AND
NOT EXISTS (
SELECT * FROM tb_produto_ficha_tecnica
WHERE
/* %JoinFKPK(tb_produto_ficha_tecnica,tb_unidade_medida," = ","AND") */
tb_produto_ficha_tecnica.id_unidade_medida = tb_unidade_medida.id_unidade_medida
)
)
BEGIN
SELECT @errno = 30010,
@errmsg = 'Cannot delete last tb_produto_ficha_tecnica because tb_unidade_medida exists.'
GOTO ERROR

```

```

END

/* ERwin Builtin Trigger */
/* tb_producto tb_producto_ficha_tecnica on child delete no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_producto"
CHILD_OWNER="", CHILD_TABLE="tb_producto_ficha_tecnica"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_33", FK_COLUMNS="id_producto_insumo" */
IF EXISTS (SELECT * FROM deleted,tb_producto
WHERE
/* %JoinFKPK(deleted,tb_producto,"=","AND") */
deleted.id_producto_insumo = tb_producto.id_producto AND
NOT EXISTS (
SELECT * FROM tb_producto_ficha_tecnica
WHERE
/* %JoinFKPK(tb_producto_ficha_tecnica,tb_producto,"=","AND") */
tb_producto_ficha_tecnica.id_producto_insumo = tb_producto.id_producto
)
)
BEGIN
SELECT @errno = 30010,
@errmsg = 'Cannot delete last tb_producto_ficha_tecnica because tb_producto exists.'
GOTO ERROR
END

/* ERwin Builtin Trigger */
/* tb_producto tb_producto_ficha_tecnica on child delete no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_producto"
CHILD_OWNER="", CHILD_TABLE="tb_producto_ficha_tecnica"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_32", FK_COLUMNS="id_producto_acabado" */
IF EXISTS (SELECT * FROM deleted,tb_producto
WHERE
/* %JoinFKPK(deleted,tb_producto,"=","AND") */
deleted.id_producto_acabado = tb_producto.id_producto AND
NOT EXISTS (
SELECT * FROM tb_producto_ficha_tecnica
WHERE
/* %JoinFKPK(tb_producto_ficha_tecnica,tb_producto,"=","AND") */
tb_producto_ficha_tecnica.id_producto_acabado = tb_producto.id_producto
)
)
BEGIN
SELECT @errno = 30010,
@errmsg = 'Cannot delete last tb_producto_ficha_tecnica because tb_producto exists.'
GOTO ERROR
END

/* ERwin Builtin Trigger */
RETURN
ERROR:
raiserror @errno @errmsg
rollback transaction
END

```

```

go

```

```

CREATE TRIGGER tU_tb_producto_ficha_tecnica ON tb_producto_ficha_tecnica FOR UPDATE AS
/* ERwin Builtin Trigger */
/* UPDATE trigger on tb_producto_ficha_tecnica */
BEGIN
DECLARE @NUMROWS int,
@nullcnt int,
@validcnt int,
@insid_ficha_tecnica integer,
@errno int,
@errmsg varchar(255)

```

```

SELECT @NUMROWS = @@rowcount
/* ERwin Builtin Trigger */
/* tb_unidade_medida tb_produto_ficha_tecnica on child update no action */
/* ERWIN_RELATION:CHECKSUM="0004db29", PARENT_OWNER="", PARENT_TABLE="tb_unidade_medida"
  CHILD_OWNER="", CHILD_TABLE="tb_produto_ficha_tecnica"
  P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
  FK_CONSTRAINT="R_34", FK_COLUMNS="id_unidade_medida" */
IF
/* %ChildFK(" OR",UPDATE) */
UPDATE(id_unidade_medida)
BEGIN
SELECT @nullcnt = 0
SELECT @validcnt = count(*)
  FROM inserted,tb_unidade_medida
  WHERE
    /* %JoinFKPK(inserted,tb_unidade_medida) */
    inserted.id_unidade_medida = tb_unidade_medida.id_unidade_medida
/* %NotNullFK(inserted,"IS NULL","select @nullcnt = count(*) from inserted where"," AND") */
select @nullcnt = count(*) from inserted where
  inserted.id_unidade_medida IS NULL
IF @validcnt + @nullcnt != @NUMROWS
BEGIN
  SELECT @errno = 30007,
    @errmsg = 'Cannot update tb_produto_ficha_tecnica because tb_unidade_medida does not exist.'
  GOTO ERROR
END
END

```

```

/* ERwin Builtin Trigger */
/* tb_produto tb_produto_ficha_tecnica on child update no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_produto"
  CHILD_OWNER="", CHILD_TABLE="tb_produto_ficha_tecnica"
  P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
  FK_CONSTRAINT="R_33", FK_COLUMNS="id_produto_insumo" */
IF
/* %ChildFK(" OR",UPDATE) */
UPDATE(id_produto_insumo)
BEGIN
SELECT @nullcnt = 0
SELECT @validcnt = count(*)
  FROM inserted,tb_produto
  WHERE
    /* %JoinFKPK(inserted,tb_produto) */
    inserted.id_produto_insumo = tb_produto.id_produto
/* %NotNullFK(inserted,"IS NULL","select @nullcnt = count(*) from inserted where"," AND") */
select @nullcnt = count(*) from inserted where
  inserted.id_produto_insumo IS NULL
IF @validcnt + @nullcnt != @NUMROWS
BEGIN
  SELECT @errno = 30007,
    @errmsg = 'Cannot update tb_produto_ficha_tecnica because tb_produto does not exist.'
  GOTO ERROR
END
END

```

```

/* ERwin Builtin Trigger */
/* tb_produto tb_produto_ficha_tecnica on child update no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_produto"
  CHILD_OWNER="", CHILD_TABLE="tb_produto_ficha_tecnica"
  P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
  FK_CONSTRAINT="R_32", FK_COLUMNS="id_produto_acabado" */
IF
/* %ChildFK(" OR",UPDATE) */
UPDATE(id_produto_acabado)
BEGIN
SELECT @nullcnt = 0
SELECT @validcnt = count(*)
  FROM inserted,tb_produto
  WHERE
    /* %JoinFKPK(inserted,tb_produto) */

```

```

        inserted.id_produto_acabado = tb_produto.id_produto
/* %NotNullFK(inserted,"IS NULL","select @nullcnt = count(*) from inserted where","AND") */
select @nullcnt = count(*) from inserted where
    inserted.id_produto_acabado IS NULL
IF @validcnt + @nullcnt != @NUMROWS
BEGIN
    SELECT @errno = 30007,
           @errmsg = 'Cannot update tb_produto_ficha_tecnica because tb_produto does not exist.'
    GOTO ERROR
END
END

```

```

/* ERwin Builtin Trigger */
RETURN
ERROR:
    raiserror @errno @errmsg
rollback transaction
END

```

go

```

CREATE TRIGGER tD_tb_produto_insumo_estoque ON tb_produto_insumo_estoque FOR DELETE AS
/* ERwin Builtin Trigger */
/* DELETE trigger on tb_produto_insumo_estoque */
BEGIN
    DECLARE @errno int,
            @errmsg varchar(255)
/* ERwin Builtin Trigger */
/* tb_produto_insumo_estoque tb_produto_insumo_movimentacao_estoque on parent delete no action */
/* ERWIN_RELATION:CHECKSUM="00046a42", PARENT_OWNER="", PARENT_TABLE="tb_produto_insumo_estoque"
CHILD_OWNER="", CHILD_TABLE="tb_produto_insumo_movimentacao_estoque"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_47", FK_COLUMNS="id_produto_insumo_estoque" */
IF EXISTS (
    SELECT * FROM deleted,tb_produto_insumo_movimentacao_estoque
    WHERE
        /* %JoinFKPK(tb_produto_insumo_movimentacao_estoque,deleted,"=","AND") */
        tb_produto_insumo_movimentacao_estoque.id_produto_insumo_estoque = deleted.id_produto_insumo_estoque
)
BEGIN
    SELECT @errno = 30001,
           @errmsg = 'Cannot delete tb_produto_insumo_estoque because tb_produto_insumo_movimentacao_estoque exists.'
    GOTO ERROR
END

/* ERwin Builtin Trigger */
/* tb_unidade_medida tb_produto_insumo_estoque on child delete no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_unidade_medida"
CHILD_OWNER="", CHILD_TABLE="tb_produto_insumo_estoque"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_46", FK_COLUMNS="id_unidade_medida" */
IF EXISTS (SELECT * FROM deleted,tb_unidade_medida
    WHERE
        /* %JoinFKPK(deleted,tb_unidade_medida,"=","AND") */
        deleted.id_unidade_medida = tb_unidade_medida.id_unidade_medida AND
        NOT EXISTS (
            SELECT * FROM tb_produto_insumo_estoque
            WHERE
                /* %JoinFKPK(tb_produto_insumo_estoque,tb_unidade_medida,"=","AND") */
                tb_produto_insumo_estoque.id_unidade_medida = tb_unidade_medida.id_unidade_medida
        )
)
BEGIN
    SELECT @errno = 30010,
           @errmsg = 'Cannot delete last tb_produto_insumo_estoque because tb_unidade_medida exists.'
    GOTO ERROR

```

```

END

/* ERwin Builtin Trigger */
/* tb_produto tb_produto_insumo_estoque on child delete no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_produto"
CHILD_OWNER="", CHILD_TABLE="tb_produto_insumo_estoque"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_45", FK_COLUMNS="id_produto_insumo" */
IF EXISTS (SELECT * FROM deleted, tb_produto
WHERE
/* %JoinFKPK(deleted, tb_produto, " = ", " AND") */
deleted.id_produto_insumo = tb_produto.id_produto AND
NOT EXISTS (
SELECT * FROM tb_produto_insumo_estoque
WHERE
/* %JoinFKPK(tb_produto_insumo_estoque, tb_produto, " = ", " AND") */
tb_produto_insumo_estoque.id_produto_insumo = tb_produto.id_produto
)
)
BEGIN
SELECT @errno = 30010,
@errmsg = 'Cannot delete last tb_produto_insumo_estoque because tb_produto exists.'
GOTO ERROR
END

/* ERwin Builtin Trigger */
RETURN
ERROR:
raiserror @errno @errmsg
rollback transaction
END

go

CREATE TRIGGER tU_tb_produto_insumo_estoque ON tb_produto_insumo_estoque FOR UPDATE AS
/* ERwin Builtin Trigger */
/* UPDATE trigger on tb_produto_insumo_estoque */
BEGIN
DECLARE @NUMROWS int,
@nullcnt int,
@validcnt int,
@insid_produto_insumo_estoque char(18),
@errno int,
@errmsg varchar(255)

SELECT @NUMROWS = @@rowcount
/* ERwin Builtin Trigger */
/* tb_produto_insumo_estoque tb_produto_insumo_movimentacao_estoque on parent update no action */
/* ERWIN_RELATION:CHECKSUM="0004ae83", PARENT_OWNER="", PARENT_TABLE="tb_produto_insumo_estoque"
CHILD_OWNER="", CHILD_TABLE="tb_produto_insumo_movimentacao_estoque"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_47", FK_COLUMNS="id_produto_insumo_estoque" */
IF
/* %ParentPK(" OR", UPDATE) */
UPDATE(id_produto_insumo_estoque)
BEGIN
IF EXISTS (
SELECT * FROM deleted, tb_produto_insumo_movimentacao_estoque
WHERE
/* %JoinFKPK(tb_produto_insumo_movimentacao_estoque, deleted, " = ", " AND") */
tb_produto_insumo_movimentacao_estoque.id_produto_insumo_estoque = deleted.id_produto_insumo_estoque
)
BEGIN
SELECT @errno = 30005,
@errmsg = 'Cannot update tb_produto_insumo_estoque because tb_produto_insumo_movimentacao_estoque exists.'
GOTO ERROR
END
END

```

```

/* ERwin Builtin Trigger */
/* tb_unidade_medida tb_produto_insumo_estoque on child update no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_unidade_medida"
  CHILD_OWNER="", CHILD_TABLE="tb_produto_insumo_estoque"
  P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
  FK_CONSTRAINT="R_46", FK_COLUMNS="id_unidade_medida" */
IF
  /* %ChildFK(" OR",UPDATE) */
  UPDATE(id_unidade_medida)
BEGIN
  SELECT @nullcnt = 0
  SELECT @validcnt = count(*)
    FROM inserted,tb_unidade_medida
    WHERE
      /* %JoinFKPK(inserted,tb_unidade_medida) */
      inserted.id_unidade_medida = tb_unidade_medida.id_unidade_medida
  /* %NotNullFK(inserted,"IS NULL","select @nullcnt = count(*) from inserted where"," AND") */
  select @nullcnt = count(*) from inserted where
    inserted.id_unidade_medida IS NULL
  IF @validcnt + @nullcnt != @NUMROWS
  BEGIN
    SELECT @errno = 30007,
      @errmsg = 'Cannot update tb_produto_insumo_estoque because tb_unidade_medida does not exist.'
    GOTO ERROR
  END
END

/* ERwin Builtin Trigger */
/* tb_produto tb_produto_insumo_estoque on child update no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_produto"
  CHILD_OWNER="", CHILD_TABLE="tb_produto_insumo_estoque"
  P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
  FK_CONSTRAINT="R_45", FK_COLUMNS="id_produto_insumo" */
IF
  /* %ChildFK(" OR",UPDATE) */
  UPDATE(id_produto_insumo)
BEGIN
  SELECT @nullcnt = 0
  SELECT @validcnt = count(*)
    FROM inserted,tb_produto
    WHERE
      /* %JoinFKPK(inserted,tb_produto) */
      inserted.id_produto_insumo = tb_produto.id_produto
  /* %NotNullFK(inserted,"IS NULL","select @nullcnt = count(*) from inserted where"," AND") */
  select @nullcnt = count(*) from inserted where
    inserted.id_produto_insumo IS NULL
  IF @validcnt + @nullcnt != @NUMROWS
  BEGIN
    SELECT @errno = 30007,
      @errmsg = 'Cannot update tb_produto_insumo_estoque because tb_produto does not exist.'
    GOTO ERROR
  END
END

/* ERwin Builtin Trigger */
RETURN
ERROR:
  raiserror @errno @errmsg
  rollback transaction
END

go

```

```

CREATE TRIGGER tD_tb_produto_insumo_movimentacao_estoque ON tb_produto_insumo_movimentacao_estoque FOR DELETE AS
/* ERwin Builtin Trigger */

```

```

/* DELETE trigger on tb_produto_insumo_movimentacao_estoque */
BEGIN
  DECLARE @erro int,
           @errmsg varchar(255)
  /* ERwin Builtin Trigger */
  /* tb_tipo_movimentacao_estoque tb_produto_insumo_movimentacao_estoque on child delete no action */
  /* ERWIN_RELATION:CHECKSUM="0003e0df", PARENT_OWNER="", PARENT_TABLE="tb_tipo_movimentacao_estoque"
  CHILD_OWNER="", CHILD_TABLE="tb_produto_insumo_movimentacao_estoque"
  P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
  FK_CONSTRAINT="R_48", FK_COLUMNS="id_tipo_movimentacao_estoque" */
  IF EXISTS (SELECT * FROM deleted, tb_tipo_movimentacao_estoque
             WHERE
              /* %JoinFKPK(deleted, tb_tipo_movimentacao_estoque, " = " AND") */
              deleted.id_tipo_movimentacao_estoque = tb_tipo_movimentacao_estoque.id_tipo_movimentacao_estoque AND
              NOT EXISTS (
                SELECT * FROM tb_produto_insumo_movimentacao_estoque
                WHERE
                  /* %JoinFKPK(tb_produto_insumo_movimentacao_estoque, tb_tipo_movimentacao_estoque, " = " AND") */
                  tb_produto_insumo_movimentacao_estoque.id_tipo_movimentacao_estoque = tb_tipo_movimentacao_estoque.id_tipo_movimentacao_estoque
              )
            )
  BEGIN
    SELECT @erro = 30010,
           @errmsg = 'Cannot delete last tb_produto_insumo_movimentacao_estoque because tb_tipo_movimentacao_estoque exists.'
    GOTO ERROR
  END

```

```

/* ERwin Builtin Trigger */
/* tb_produto_insumo_estoque tb_produto_insumo_movimentacao_estoque on child delete no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_produto_insumo_estoque"
  CHILD_OWNER="", CHILD_TABLE="tb_produto_insumo_movimentacao_estoque"
  P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
  FK_CONSTRAINT="R_47", FK_COLUMNS="id_produto_insumo_estoque" */
  IF EXISTS (SELECT * FROM deleted, tb_produto_insumo_estoque
             WHERE
              /* %JoinFKPK(deleted, tb_produto_insumo_estoque, " = " AND") */
              deleted.id_produto_insumo_estoque = tb_produto_insumo_estoque.id_produto_insumo_estoque AND
              NOT EXISTS (
                SELECT * FROM tb_produto_insumo_movimentacao_estoque
                WHERE
                  /* %JoinFKPK(tb_produto_insumo_movimentacao_estoque, tb_produto_insumo_estoque, " = " AND") */
                  tb_produto_insumo_movimentacao_estoque.id_produto_insumo_estoque = tb_produto_insumo_estoque.id_produto_insumo_estoque
              )
            )
  BEGIN
    SELECT @erro = 30010,
           @errmsg = 'Cannot delete last tb_produto_insumo_movimentacao_estoque because tb_produto_insumo_estoque exists.'
    GOTO ERROR
  END

```

```

/* ERwin Builtin Trigger */
RETURN
ERROR:
  raiserror @erro @errmsg
  rollback transaction
END

```

go

```

CREATE TRIGGER tU_tb_produto_insumo_movimentacao_estoque ON tb_produto_insumo_movimentacao_estoque FOR UPDATE AS
/* ERwin Builtin Trigger */
/* UPDATE trigger on tb_produto_insumo_movimentacao_estoque */
BEGIN
  DECLARE @NUMROWS int,
           @nullcnt int,
           @validcnt int,
           @insid_movimentacao integer,
           @erro int,

```

```

@errmsg varchar(255)

SELECT @NUMROWS = @@rowcount
/* ERwin Builtin Trigger */
/* tb_tipo_movimentacao_estoque tb_produto_insumo_movimentacao_estoque on child update no action */
/* ERWIN_RELATION:CHECKSUM="0003edd3", PARENT_OWNER="", PARENT_TABLE="tb_tipo_movimentacao_estoque"
CHILD_OWNER="", CHILD_TABLE="tb_produto_insumo_movimentacao_estoque"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_48", FK_COLUMNS="id_tipo_movimentacao_estoque" */
IF
/* %ChildFK(" OR",UPDATE) */
UPDATE(id_tipo_movimentacao_estoque)
BEGIN
SELECT @nullcnt = 0
SELECT @validcnt = count(*)
FROM inserted,tb_tipo_movimentacao_estoque
WHERE
/* %JoinFKPK(inserted,tb_tipo_movimentacao_estoque) */
inserted.id_tipo_movimentacao_estoque = tb_tipo_movimentacao_estoque.id_tipo_movimentacao_estoque
/* %NotNullFK(inserted," IS NULL","select @nullcnt = count(*) from inserted where"," AND") */
select @nullcnt = count(*) from inserted where
inserted.id_tipo_movimentacao_estoque IS NULL
IF @validcnt + @nullcnt != @NUMROWS
BEGIN
SELECT @errno = 30007,
@errmsg = 'Cannot update tb_produto_insumo_movimentacao_estoque because tb_tipo_movimentacao_estoque does not exist.'
GOTO ERROR
END
END

/* ERwin Builtin Trigger */
/* tb_produto_insumo_estoque tb_produto_insumo_movimentacao_estoque on child update no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_produto_insumo_estoque"
CHILD_OWNER="", CHILD_TABLE="tb_produto_insumo_movimentacao_estoque"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_47", FK_COLUMNS="id_produto_insumo_estoque" */
IF
/* %ChildFK(" OR",UPDATE) */
UPDATE(id_produto_insumo_estoque)
BEGIN
SELECT @nullcnt = 0
SELECT @validcnt = count(*)
FROM inserted,tb_produto_insumo_estoque
WHERE
/* %JoinFKPK(inserted,tb_produto_insumo_estoque) */
inserted.id_produto_insumo_estoque = tb_produto_insumo_estoque.id_produto_insumo_estoque
/* %NotNullFK(inserted," IS NULL","select @nullcnt = count(*) from inserted where"," AND") */
select @nullcnt = count(*) from inserted where
inserted.id_produto_insumo_estoque IS NULL
IF @validcnt + @nullcnt != @NUMROWS
BEGIN
SELECT @errno = 30007,
@errmsg = 'Cannot update tb_produto_insumo_movimentacao_estoque because tb_produto_insumo_estoque does not exist.'
GOTO ERROR
END
END

/* ERwin Builtin Trigger */
RETURN
ERROR:
raiserror @errno @errmsg
rollback transaction
END

```



```

CREATE TRIGGER tD_tb_produtoroteiro_producao ON tb_produtoroteiro_producao FOR DELETE AS
/* ERwin Bultin Trigger */
/* DELETE trigger on tb_produtoroteiro_producao */
BEGIN
DECLARE @errno int,
        @errmsg varchar(255)
/* ERwin Bultin Trigger */
/* tb_produto tb_produtoroteiro_producao on child delete no action */
/* ERWIN_RELATION:CHECKSUM="00017777", PARENT_OWNER="", PARENT_TABLE="tb_produto"
CHILD_OWNER="", CHILD_TABLE="tb_produtoroteiro_producao"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_35", FK_COLUMNS="id_produto_acabado" */
IF EXISTS (SELECT * FROM deleted, tb_produto
WHERE
/* %JoinFKPK(deleted, tb_produto, " = ", " AND") */
deleted.id_produto_acabado = tb_produto.id_produto AND
NOT EXISTS (
SELECT * FROM tb_produtoroteiro_producao
WHERE
/* %JoinFKPK(tb_produtoroteiro_producao, tb_produto, " = ", " AND") */
tb_produtoroteiro_producao.id_produto_acabado = tb_produto.id_produto
)
)
BEGIN
SELECT @errno = 30010,
        @errmsg = 'Cannot delete last tb_produtoroteiro_producao because tb_produto exists.'
GOTO ERROR
END

/* ERwin Bultin Trigger */
RETURN
ERROR:
raiserror @errno @errmsg
rollback transaction
END

```

go

```

CREATE TRIGGER tU_tb_produtoroteiro_producao ON tb_produtoroteiro_producao FOR UPDATE AS
/* ERwin Bultin Trigger */
/* UPDATE trigger on tb_produtoroteiro_producao */
BEGIN
DECLARE @NUMROWS int,
        @nullcnt int,
        @validcnt int,
        @insidroteiro integer,
        @errno int,
        @errmsg varchar(255)

SELECT @NUMROWS = @@rowcount
/* ERwin Bultin Trigger */
/* tb_produto tb_produtoroteiro_producao on child update no action */
/* ERWIN_RELATION:CHECKSUM="00018f6d", PARENT_OWNER="", PARENT_TABLE="tb_produto"
CHILD_OWNER="", CHILD_TABLE="tb_produtoroteiro_producao"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_35", FK_COLUMNS="id_produto_acabado" */
IF
/* %ChildFK(" OR", UPDATE) */
UPDATE(id_produto_acabado)
BEGIN
SELECT @nullcnt = 0
SELECT @validcnt = count(*)
FROM inserted, tb_produto
WHERE
/* %JoinFKPK(inserted, tb_produto) */
inserted.id_produto_acabado = tb_produto.id_produto
/* %NotNullFK(inserted, " IS NULL", "select @nullcnt = count(*) from inserted where", " AND") */
select @nullcnt = count(*) from inserted where

```

```

inserted.id_produto_acabado IS NULL
IF @validcnt + @nullcnt != @NUMROWS
BEGIN
    SELECT @errno = 30007,
           @errmsg = 'Cannot update tb_produtoroteiro_producao because tb_produto does not exist.'
    GOTO ERROR
END
END

```

```

/* ERwin Builtin Trigger */
RETURN
ERROR:
    raiserror @errno @errmsg
    rollback transaction
END

```

go

```

CREATE TRIGGER tD_tb_produto_tabela_preco ON tb_produto_tabela_preco FOR DELETE AS
/* ERwin Builtin Trigger */
/* DELETE trigger on tb_produto_tabela_preco */
BEGIN
    DECLARE @errno int,
            @errmsg varchar(255)
    /* ERwin Builtin Trigger */
    /* tb_tabela_preco tb_produto_tabela_preco on child delete no action */
    /* ERWIN_RELATION:CHECKSUM="000327db", PARENT_OWNER="", PARENT_TABLE="tb_tabela_preco"
    CHILD_OWNER="", CHILD_TABLE="tb_produto_tabela_preco"
    P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
    FK_CONSTRAINT="R_40", FK_COLUMNS="id_tabela_preco" id_tipo_cliente" */
    IF EXISTS (SELECT * FROM deleted, tb_tabela_preco
               WHERE
                   /* %JoinFKPK(deleted, tb_tabela_preco, " = ", "AND") */
                   deleted.id_tabela_preco = tb_tabela_preco.id_tabela_preco AND
                   deleted.id_tipo_cliente = tb_tabela_preco.id_tipo_cliente AND
                   NOT EXISTS (
                       SELECT * FROM tb_produto_tabela_preco
                       WHERE
                           /* %JoinFKPK(tb_produto_tabela_preco, tb_tabela_preco, " = ", "AND") */
                           tb_produto_tabela_preco.id_tabela_preco = tb_tabela_preco.id_tabela_preco AND
                           tb_produto_tabela_preco.id_tipo_cliente = tb_tabela_preco.id_tipo_cliente
                       )
                   )
    )
    BEGIN
        SELECT @errno = 30010,
               @errmsg = 'Cannot delete last tb_produto_tabela_preco because tb_tabela_preco exists.'
        GOTO ERROR
    END

    /* ERwin Builtin Trigger */
    /* tb_produto tb_produto_tabela_preco on child delete no action */
    /* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_produto"
    CHILD_OWNER="", CHILD_TABLE="tb_produto_tabela_preco"
    P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
    FK_CONSTRAINT="R_39", FK_COLUMNS="id_produto_acabado" */
    IF EXISTS (SELECT * FROM deleted, tb_produto
               WHERE
                   /* %JoinFKPK(deleted, tb_produto, " = ", "AND") */
                   deleted.id_produto_acabado = tb_produto.id_produto AND
                   NOT EXISTS (
                       SELECT * FROM tb_produto_tabela_preco
                       WHERE
                           /* %JoinFKPK(tb_produto_tabela_preco, tb_produto, " = ", "AND") */
                           tb_produto_tabela_preco.id_produto_acabado = tb_produto.id_produto
                       )
                   )
    )

```

```

BEGIN
  SELECT @errno = 30010,
         @errmsg = 'Cannot delete last tb_produto_tabela_preco because tb_produto exists.'
  GOTO ERROR
END

```

```

/* ERwin Builtin Trigger */
RETURN
ERROR:
  raiserror @errno @errmsg
  rollback transaction
END

```

go

```

CREATE TRIGGER tU_tb_produto_tabela_preco ON tb_produto_tabela_preco FOR UPDATE AS

```

```

/* ERwin Builtin Trigger */
/* UPDATE trigger on tb_produto_tabela_preco */
BEGIN
  DECLARE @NUMROWS int,
          @nullcnt int,
          @validcnt int,
          @insid_produto_acabado integer,
          @insid_tabela_preco integer,
          @insid_tipo_cliente integer,
          @errno int,
          @errmsg varchar(255)

  SELECT @NUMROWS = @@rowcount
  /* ERwin Builtin Trigger */
  /* tb_tabela_preco tb_produto_tabela_preco on child update no action */
  /* ERWIN_RELATION:CHECKSUM="000321c6", PARENT_OWNER="", PARENT_TABLE="tb_tabela_preco"
  CHILD_OWNER="", CHILD_TABLE="tb_produto_tabela_preco"
  P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
  FK_CONSTRAINT="R_40", FK_COLUMNS="id_tabela_preco" id_tipo_cliente" */
  IF
    /* %ChildFK(" OR",UPDATE) */
    UPDATE(id_tabela_preco) OR
    UPDATE(id_tipo_cliente)
  BEGIN
    SELECT @nullcnt = 0
    SELECT @validcnt = count(*)
      FROM inserted,tb_tabela_preco
      WHERE
        /* %JoinFKPK(inserted,tb_tabela_preco) */
        inserted.id_tabela_preco = tb_tabela_preco.id_tabela_preco and
        inserted.id_tipo_cliente = tb_tabela_preco.id_tipo_cliente
    /* %NotNullFK(inserted," IS NULL","select @nullcnt = count(*) from inserted where"," AND") */

```

```

  IF @validcnt + @nullcnt != @NUMROWS
  BEGIN
    SELECT @errno = 30007,
           @errmsg = 'Cannot update tb_produto_tabela_preco because tb_tabela_preco does not exist.'
    GOTO ERROR
  END
END

```

```

/* ERwin Builtin Trigger */
/* tb_produto tb_produto_tabela_preco on child update no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_produto"
CHILD_OWNER="", CHILD_TABLE="tb_produto_tabela_preco"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_39", FK_COLUMNS="id_produto_acabado" */
IF
  /* %ChildFK(" OR",UPDATE) */
  UPDATE(id_produto_acabado)
BEGIN
  SELECT @nullcnt = 0

```

```

SELECT @validcnt = count(*)
FROM inserted,tb_produto
WHERE
/* %JoinFKPK(inserted,tb_produto) */
inserted.id_produto_acabado = tb_produto.id_produto
/* %NotNullFK(inserted," IS NULL","select @nullcnt = count(*) from inserted where"," AND") */

IF @validcnt + @nullcnt != @NUMROWS
BEGIN
SELECT @errno = 30007,
      @errmsg = 'Cannot update tb_produto_tabela_preco because tb_produto does not exist.'
GOTO ERROR
END
END

```

```

/* ERwin Builtin Trigger */
RETURN
ERROR:
      raiserror @errno @errmsg
      rollback transaction
END

```

go

```

CREATE TRIGGER tD_tb_tabela_preco ON tb_tabela_preco FOR DELETE AS
/* ERwin Builtin Trigger */
/* DELETE trigger on tb_tabela_preco */
BEGIN
DECLARE @errno int,
        @errmsg varchar(255)
/* ERwin Builtin Trigger */
/* tb_tabela_preco tb_produto_tabela_preco on parent delete no action */
/* ERWIN_RELATION:CHECKSUM="0004138d", PARENT_OWNER="", PARENT_TABLE="tb_tabela_preco"
CHILD_OWNER="", CHILD_TABLE="tb_produto_tabela_preco"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_40", FK_COLUMNS="id_tabela_preco""id_tipo_cliente" */
IF EXISTS (
SELECT * FROM deleted,tb_produto_tabela_preco
WHERE
/* %JoinFKPK(tb_produto_tabela_preco,deleted," = "," AND") */
tb_produto_tabela_preco.id_tabela_preco = deleted.id_tabela_preco AND
tb_produto_tabela_preco.id_tipo_cliente = deleted.id_tipo_cliente
)
BEGIN
SELECT @errno = 30001,
      @errmsg = 'Cannot delete tb_tabela_preco because tb_produto_tabela_preco exists.'
GOTO ERROR
END

/* ERwin Builtin Trigger */
/* tb_tabela_preco tb_tabela_preco_cliente on parent delete no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_tabela_preco"
CHILD_OWNER="", CHILD_TABLE="tb_tabela_preco_cliente"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_37", FK_COLUMNS="id_tabela_preco""id_tipo_cliente" */
IF EXISTS (
SELECT * FROM deleted,tb_tabela_preco_cliente
WHERE
/* %JoinFKPK(tb_tabela_preco_cliente,deleted," = "," AND") */
tb_tabela_preco_cliente.id_tabela_preco = deleted.id_tabela_preco AND
tb_tabela_preco_cliente.id_tipo_cliente = deleted.id_tipo_cliente
)
BEGIN
SELECT @errno = 30001,
      @errmsg = 'Cannot delete tb_tabela_preco because tb_tabela_preco_cliente exists.'
GOTO ERROR

```

```

END

/* ERwin Builtin Trigger */
/* tb_tipo_cliente tb_tabela_preco on child delete no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_tipo_cliente"
CHILD_OWNER="", CHILD_TABLE="tb_tabela_preco"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_36", FK_COLUMNS="id_tipo_cliente" */
IF EXISTS (SELECT * FROM deleted, tb_tipo_cliente
WHERE
/* %JoinFKPK(deleted, tb_tipo_cliente, " = ", "AND") */
deleted.id_tipo_cliente = tb_tipo_cliente.id_tipo_cliente AND
NOT EXISTS (
SELECT * FROM tb_tabela_preco
WHERE
/* %JoinFKPK(tb_tabela_preco, tb_tipo_cliente, " = ", "AND") */
tb_tabela_preco.id_tipo_cliente = tb_tipo_cliente.id_tipo_cliente
)
)
BEGIN
SELECT @errno = 30010,
@errmsg = 'Cannot delete last tb_tabela_preco because tb_tipo_cliente exists.'
GOTO ERROR
END

```

```

/* ERwin Builtin Trigger */
RETURN
ERROR:
raiserror @errno @errmsg
rollback transaction
END

```

```

go

```

```

CREATE TRIGGER tU_tb_tabela_preco ON tb_tabela_preco FOR UPDATE AS
/* ERwin Builtin Trigger */
/* UPDATE trigger on tb_tabela_preco */
BEGIN
DECLARE @NUMROWS int,
@nullcnt int,
@validcnt int,
@insid_tabela_preco integer,
@insid_tipo_cliente integer,
@errno int,
@errmsg varchar(255)

SELECT @NUMROWS = @@rowcount
/* ERwin Builtin Trigger */
/* tb_tabela_preco tb_produto_tabela_preco on parent update no action */
/* ERWIN_RELATION:CHECKSUM="00046a3d", PARENT_OWNER="", PARENT_TABLE="tb_tabela_preco"
CHILD_OWNER="", CHILD_TABLE="tb_produto_tabela_preco"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_40", FK_COLUMNS="id_tabela_preco""id_tipo_cliente" */
IF
/* %ParentPK(" OR", UPDATE) */
UPDATE(id_tabela_preco) OR
UPDATE(id_tipo_cliente)
BEGIN
IF EXISTS (
SELECT * FROM deleted, tb_produto_tabela_preco
WHERE
/* %JoinFKPK(tb_produto_tabela_preco, deleted, " = ", "AND") */
tb_produto_tabela_preco.id_tabela_preco = deleted.id_tabela_preco AND
tb_produto_tabela_preco.id_tipo_cliente = deleted.id_tipo_cliente
)
BEGIN
SELECT @errno = 30005,
@errmsg = 'Cannot update tb_tabela_preco because tb_produto_tabela_preco exists.'

```

```

GOTO ERROR
END
END

/* ERwin Builtin Trigger */
/* tb_tabela_preco tb_tabela_preco_cliente on parent update no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_tabela_preco"
CHILD_OWNER="", CHILD_TABLE="tb_tabela_preco_cliente"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_37", FK_COLUMNS="id_tabela_preco""id_tipo_cliente" */
IF
/* %ParentPK(" OR",UPDATE) */
UPDATE(id_tabela_preco) OR
UPDATE(id_tipo_cliente)
BEGIN
IF EXISTS (
SELECT * FROM deleted,tb_tabela_preco_cliente
WHERE
/* %JoinFKPK(tb_tabela_preco_cliente,deleted," = "," AND") */
tb_tabela_preco_cliente.id_tabela_preco = deleted.id_tabela_preco AND
tb_tabela_preco_cliente.id_tipo_cliente = deleted.id_tipo_cliente
)
BEGIN
SELECT @errno = 30005,
@errmsg = 'Cannot update tb_tabela_preco because tb_tabela_preco_cliente exists.'
GOTO ERROR
END
END

/* ERwin Builtin Trigger */
/* tb_tipo_cliente tb_tabela_preco on child update no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_tipo_cliente"
CHILD_OWNER="", CHILD_TABLE="tb_tabela_preco"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_36", FK_COLUMNS="id_tipo_cliente" */
IF
/* %ChildFK(" OR",UPDATE) */
UPDATE(id_tipo_cliente)
BEGIN
SELECT @nullcnt = 0
SELECT @validcnt = count(*)
FROM inserted,tb_tipo_cliente
WHERE
/* %JoinFKPK(inserted,tb_tipo_cliente) */
inserted.id_tipo_cliente = tb_tipo_cliente.id_tipo_cliente
/* %NotNullFK(inserted," IS NULL","select @nullcnt = count(*) from inserted where"," AND") */

IF @validcnt + @nullcnt != @NUMROWS
BEGIN
SELECT @errno = 30007,
@errmsg = 'Cannot update tb_tabela_preco because tb_tipo_cliente does not exist.'
GOTO ERROR
END
END

/* ERwin Builtin Trigger */
RETURN
ERROR:
raiserror @errno @errmsg
rollback transaction
END

go

```

```

CREATE TRIGGER tD_tb_tabela_preco_cliente ON tb_tabela_preco_cliente FOR DELETE AS
/* ERwin Builtin Trigger */

```

```

/* DELETE trigger on tb_tabela_preco_cliente */
BEGIN
  DECLARE @errno int,
           @errmsg varchar(255);
/* ERwin Builtin Trigger */
/* tb_tipo_cliente_cliente tb_tabela_preco_cliente on child delete no action */
/* ERWIN_RELATION:CHECKSUM="0003a016", PARENT_OWNER="", PARENT_TABLE="tb_tipo_cliente_cliente"
CHILD_OWNER="", CHILD_TABLE="tb_tabela_preco_cliente"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_38", FK_COLUMNS="id_tipo_cliente""id_cliente" */
IF EXISTS (SELECT * FROM deleted, tb_tipo_cliente_cliente
WHERE
/* %JoinFKPK(deleted, tb_tipo_cliente_cliente, " = ", " AND") */
deleted.id_tipo_cliente = tb_tipo_cliente_cliente.id_tipo_cliente AND
deleted.id_cliente = tb_tipo_cliente_cliente.id_cliente AND
NOT EXISTS (
  SELECT * FROM tb_tabela_preco_cliente
WHERE
/* %JoinFKPK(tb_tabela_preco_cliente, tb_tipo_cliente_cliente, " = ", " AND") */
tb_tabela_preco_cliente.id_tipo_cliente = tb_tipo_cliente_cliente.id_tipo_cliente AND
tb_tabela_preco_cliente.id_cliente = tb_tipo_cliente_cliente.id_cliente
)
)
BEGIN
  SELECT @errno = 30010,
         @errmsg = 'Cannot delete last tb_tabela_preco_cliente because tb_tipo_cliente_cliente exists.'
  GOTO ERROR
END

/* ERwin Builtin Trigger */
/* tb_tabela_preco tb_tabela_preco_cliente on child delete no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_tabela_preco"
CHILD_OWNER="", CHILD_TABLE="tb_tabela_preco_cliente"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_37", FK_COLUMNS="id_tabela_preco""id_tipo_cliente" */
IF EXISTS (SELECT * FROM deleted, tb_tabela_preco
WHERE
/* %JoinFKPK(deleted, tb_tabela_preco, " = ", " AND") */
deleted.id_tabela_preco = tb_tabela_preco.id_tabela_preco AND
deleted.id_tipo_cliente = tb_tabela_preco.id_tipo_cliente AND
NOT EXISTS (
  SELECT * FROM tb_tabela_preco_cliente
WHERE
/* %JoinFKPK(tb_tabela_preco_cliente, tb_tabela_preco, " = ", " AND") */
tb_tabela_preco_cliente.id_tabela_preco = tb_tabela_preco.id_tabela_preco AND
tb_tabela_preco_cliente.id_tipo_cliente = tb_tabela_preco.id_tipo_cliente
)
)
BEGIN
  SELECT @errno = 30010,
         @errmsg = 'Cannot delete last tb_tabela_preco_cliente because tb_tabela_preco exists.'
  GOTO ERROR
END

/* ERwin Builtin Trigger */
RETURN
ERROR:
  raiserror @errno @errmsg
  rollback transaction
END

```

```

go

```

```

CREATE TRIGGER tU_tb_tabela_preco_cliente ON tb_tabela_preco_cliente FOR UPDATE AS
/* ERwin Builtin Trigger */
/* UPDATE trigger on tb_tabela_preco_cliente */
BEGIN
  DECLARE @NUMROWS int,

```

```

    @nullcnt int,
    @validcnt int,
    @insid_tabela_preco integer,
    @insid_tipo_cliente integer,
    @insid_cliente integer,
    @errno int,
    @errmsg varchar(255)

SELECT @NUMROWS = @@rowcount
/* ERwin Builtin Trigger */
/* tb_tipo_cliente_cliente tb_tabela_preco_cliente on child update no action */
/* ERWIN_RELATION:CHECKSUM="00035de7", PARENT_OWNER="", PARENT_TABLE="tb_tipo_cliente_cliente"
  CHILD_OWNER="", CHILD_TABLE="tb_tabela_preco_cliente"
  P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
  FK_CONSTRAINT="R_38", FK_COLUMNS="id_tipo_cliente""id_cliente" */
IF
  /* %ChildFK(" OR",UPDATE) */
  UPDATE(id_tipo_cliente) OR
  UPDATE(id_cliente)
BEGIN
  SELECT @nullcnt = 0
  SELECT @validcnt = count(*)
    FROM inserted,tb_tipo_cliente_cliente
    WHERE
      /* %JoinFKPK(inserted,tb_tipo_cliente_cliente) */
      inserted.id_tipo_cliente = tb_tipo_cliente_cliente.id_tipo_cliente and
      inserted.id_cliente = tb_tipo_cliente_cliente.id_cliente
  /* %NotNullFK(inserted," IS NULL","select @nullcnt = count(*) from inserted where"," AND") */

  IF @validcnt + @nullcnt != @NUMROWS
  BEGIN
    SELECT @errno = 30007,
      @errmsg = 'Cannot update tb_tabela_preco_cliente because tb_tipo_cliente_cliente does not exist.'
    GOTO ERROR
  END
END

/* ERwin Builtin Trigger */
/* tb_tabela_preco tb_tabela_preco_cliente on child update no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_tabela_preco"
  CHILD_OWNER="", CHILD_TABLE="tb_tabela_preco_cliente"
  P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
  FK_CONSTRAINT="R_37", FK_COLUMNS="id_tabela_preco""id_tipo_cliente" */
IF
  /* %ChildFK(" OR",UPDATE) */
  UPDATE(id_tabela_preco) OR
  UPDATE(id_tipo_cliente)
BEGIN
  SELECT @nullcnt = 0
  SELECT @validcnt = count(*)
    FROM inserted,tb_tabela_preco
    WHERE
      /* %JoinFKPK(inserted,tb_tabela_preco) */
      inserted.id_tabela_preco = tb_tabela_preco.id_tabela_preco and
      inserted.id_tipo_cliente = tb_tabela_preco.id_tipo_cliente
  /* %NotNullFK(inserted," IS NULL","select @nullcnt = count(*) from inserted where"," AND") */

  IF @validcnt + @nullcnt != @NUMROWS
  BEGIN
    SELECT @errno = 30007,
      @errmsg = 'Cannot update tb_tabela_preco_cliente because tb_tabela_preco does not exist.'
    GOTO ERROR
  END
END

/* ERwin Builtin Trigger */
RETURN
ERROR:
  raiserror @errno @errmsg

```



```
rollback transaction
END
```

```
go
```

```
CREATE TRIGGER tD_tb_tipo_cliente ON tb_tipo_cliente FOR DELETE AS
/* ERwin Builtin Trigger */
/* DELETE trigger on tb_tipo_cliente */
BEGIN
DECLARE @ermo int,
        @errmsg varchar(255)
/* ERwin Builtin Trigger */
/* tb_tipo_cliente tb_tabela_preco on parent delete no action */
/* ERWIN_RELATION:CHECKSUM="00049ed6", PARENT_OWNER="", PARENT_TABLE="tb_tipo_cliente"
CHILD_OWNER="", CHILD_TABLE="tb_tabela_preco"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_36", FK_COLUMNS="id_tipo_cliente" */
IF EXISTS (
    SELECT * FROM deleted, tb_tabela_preco
    WHERE
        /* %JoinFKPK(tb_tabela_preco,deleted,"=","AND") */
        tb_tabela_preco.id_tipo_cliente = deleted.id_tipo_cliente
)
BEGIN
    SELECT @ermo = 30001,
           @errmsg = 'Cannot delete tb_tipo_cliente because tb_tabela_preco exists.'
    GOTO ERROR
END

/* ERwin Builtin Trigger */
/* tb_tipo_cliente tb_meio_condicao_pagamento on parent delete no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_tipo_cliente"
CHILD_OWNER="", CHILD_TABLE="tb_meio_condicao_pagamento"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_27", FK_COLUMNS="id_tipo_cliente" */
IF EXISTS (
    SELECT * FROM deleted, tb_meio_condicao_pagamento
    WHERE
        /* %JoinFKPK(tb_meio_condicao_pagamento,deleted,"=","AND") */
        tb_meio_condicao_pagamento.id_tipo_cliente = deleted.id_tipo_cliente
)
BEGIN
    SELECT @ermo = 30001,
           @errmsg = 'Cannot delete tb_tipo_cliente because tb_meio_condicao_pagamento exists.'
    GOTO ERROR
END

/* ERwin Builtin Trigger */
/* tb_tipo_cliente tb_tipo_cliente_condicao on parent delete no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_tipo_cliente"
CHILD_OWNER="", CHILD_TABLE="tb_tipo_cliente_condicao"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_22", FK_COLUMNS="id_tipo_cliente" */
IF EXISTS (
    SELECT * FROM deleted, tb_tipo_cliente_condicao
    WHERE
        /* %JoinFKPK(tb_tipo_cliente_condicao,deleted,"=","AND") */
        tb_tipo_cliente_condicao.id_tipo_cliente = deleted.id_tipo_cliente
)
BEGIN
    SELECT @ermo = 30001,
           @errmsg = 'Cannot delete tb_tipo_cliente because tb_tipo_cliente_condicao exists.'
    GOTO ERROR
END

/* ERwin Builtin Trigger */
/* tb_tipo_cliente tb_tipo_cliente_cliente on parent delete no action */
```

```

/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_tipo_cliente"
CHILD_OWNER="", CHILD_TABLE="tb_tipo_cliente_cliente"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_16", FK_COLUMNS="id_tipo_cliente" */
IF EXISTS (
    SELECT * FROM deleted,tb_tipo_cliente_cliente
    WHERE
        /* %JoinFKPK(tb_tipo_cliente_cliente,deleted," = "," AND") */
        tb_tipo_cliente_cliente.id_tipo_cliente = deleted.id_tipo_cliente
)
BEGIN
    SELECT @errno = 30001,
           @errmsg = 'Cannot delete tb_tipo_cliente because tb_tipo_cliente_cliente exists.'
    GOTO ERROR
END

/* ERwin Builtin Trigger */
RETURN
ERROR:
    raiserror @errno @errmsg
    rollback transaction
END

```

go

```

CREATE TRIGGER tU_tb_tipo_cliente ON tb_tipo_cliente FOR UPDATE AS
/* ERwin Builtin Trigger */
/* UPDATE trigger on tb_tipo_cliente */
BEGIN
    DECLARE @NUMROWS int,
            @nullcnt int,
            @validcnt int,
            @insid_tipo_cliente integer,
            @errno int,
            @errmsg varchar(255)

    SELECT @NUMROWS = @@rowcount
    /* ERwin Builtin Trigger */
    /* tb_tipo_cliente tb_tabela_preco on parent update no action */
    /* ERWIN_RELATION:CHECKSUM="000519f7", PARENT_OWNER="", PARENT_TABLE="tb_tipo_cliente"
    CHILD_OWNER="", CHILD_TABLE="tb_tabela_preco"
    P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
    FK_CONSTRAINT="R_36", FK_COLUMNS="id_tipo_cliente" */
    IF
        /* %ParentPK(" OR",UPDATE) */
        UPDATE(id_tipo_cliente)
    BEGIN
        IF EXISTS (
            SELECT * FROM deleted,tb_tabela_preco
            WHERE
                /* %JoinFKPK(tb_tabela_preco,deleted," = "," AND") */
                tb_tabela_preco.id_tipo_cliente = deleted.id_tipo_cliente
        )
        BEGIN
            SELECT @errno = 30005,
                   @errmsg = 'Cannot update tb_tipo_cliente because tb_tabela_preco exists.'
            GOTO ERROR
        END
    END

/* ERwin Builtin Trigger */
/* tb_tipo_cliente tb_meio_condicao_pagamento on parent update no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_tipo_cliente"
CHILD_OWNER="", CHILD_TABLE="tb_meio_condicao_pagamento"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_27", FK_COLUMNS="id_tipo_cliente" */
IF
    /* %ParentPK(" OR",UPDATE) */

```

```

UPDATE(id_tipo_cliente)
BEGIN
IF EXISTS (
  SELECT * FROM deleted,tb_meio_condicao_pagamento
  WHERE
    /* %JoinFKPK(tb_meio_condicao_pagamento,deleted," = "," AND") */
    tb_meio_condicao_pagamento.id_tipo_cliente = deleted.id_tipo_cliente
)
BEGIN
  SELECT @errno = 30005,
         @errmsg = 'Cannot update tb_tipo_cliente because tb_meio_condicao_pagamento exists.'
  GOTO ERROR
END
END

```

```

/* ERwin Builtin Trigger */
/* tb_tipo_cliente tb_tipo_cliente_condicao on parent update no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_tipo_cliente"
  CHILD_OWNER="", CHILD_TABLE="tb_tipo_cliente_condicao"
  P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
  FK_CONSTRAINT="R_22", FK_COLUMNS="id_tipo_cliente" */
IF
  /* %ParentPK(" OR",UPDATE) */
  UPDATE(id_tipo_cliente)
BEGIN
IF EXISTS (
  SELECT * FROM deleted,tb_tipo_cliente_condicao
  WHERE
    /* %JoinFKPK(tb_tipo_cliente_condicao,deleted," = "," AND") */
    tb_tipo_cliente_condicao.id_tipo_cliente = deleted.id_tipo_cliente
)
BEGIN
  SELECT @errno = 30005,
         @errmsg = 'Cannot update tb_tipo_cliente because tb_tipo_cliente_condicao exists.'
  GOTO ERROR
END
END

```

```

/* ERwin Builtin Trigger */
/* tb_tipo_cliente tb_tipo_cliente_cliente on parent update no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_tipo_cliente"
  CHILD_OWNER="", CHILD_TABLE="tb_tipo_cliente_cliente"
  P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
  FK_CONSTRAINT="R_16", FK_COLUMNS="id_tipo_cliente" */
IF
  /* %ParentPK(" OR",UPDATE) */
  UPDATE(id_tipo_cliente)
BEGIN
IF EXISTS (
  SELECT * FROM deleted,tb_tipo_cliente_cliente
  WHERE
    /* %JoinFKPK(tb_tipo_cliente_cliente,deleted," = "," AND") */
    tb_tipo_cliente_cliente.id_tipo_cliente = deleted.id_tipo_cliente
)
BEGIN
  SELECT @errno = 30005,
         @errmsg = 'Cannot update tb_tipo_cliente because tb_tipo_cliente_cliente exists.'
  GOTO ERROR
END
END

```

```

/* ERwin Builtin Trigger */
RETURN
ERROR:
  raiserror @errno @errmsg
  rollback transaction
END

```

go

```

CREATE TRIGGER tD_tb_tipo_cliente_cliente ON tb_tipo_cliente_cliente FOR DELETE AS
/* ERwin Builtin Trigger */
/* DELETE trigger on tb_tipo_cliente_cliente */
BEGIN
  DECLARE @errno int,
          @errmsg varchar(255)
/* ERwin Builtin Trigger */
/* tb_tipo_cliente_cliente tb_tabela_preco_cliente on parent delete no action */
/* ERWIN_RELATION:CHECKSUM="00059b2", PARENT_OWNER="", PARENT_TABLE="tb_tipo_cliente_cliente"
CHILD_OWNER="", CHILD_TABLE="tb_tabela_preco_cliente"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_38", FK_COLUMNS="id_tipo_cliente""id_cliente" */
IF EXISTS (
  SELECT * FROM deleted, tb_tabela_preco_cliente
  WHERE
    /* %JoinFKPK(tb_tabela_preco_cliente,deleted,"=","AND") */
    tb_tabela_preco_cliente.id_tipo_cliente = deleted.id_tipo_cliente AND
    tb_tabela_preco_cliente.id_cliente = deleted.id_cliente
)
BEGIN
  SELECT @errno = 30001,
         @errmsg = 'Cannot delete tb_tipo_cliente_cliente because tb_tabela_preco_cliente exists.'
  GOTO ERROR
END

/* ERwin Builtin Trigger */
/* tb_tipo_cliente_cliente tb_tipo_cliente_meio_condicao on parent delete no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_tipo_cliente_cliente"
CHILD_OWNER="", CHILD_TABLE="tb_tipo_cliente_meio_condicao"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_28", FK_COLUMNS="id_tipo_cliente""id_cliente" */
IF EXISTS (
  SELECT * FROM deleted, tb_tipo_cliente_meio_condicao
  WHERE
    /* %JoinFKPK(tb_tipo_cliente_meio_condicao,deleted,"=","AND") */
    tb_tipo_cliente_meio_condicao.id_tipo_cliente = deleted.id_tipo_cliente AND
    tb_tipo_cliente_meio_condicao.id_cliente = deleted.id_cliente
)
BEGIN
  SELECT @errno = 30001,
         @errmsg = 'Cannot delete tb_tipo_cliente_cliente because tb_tipo_cliente_meio_condicao exists.'
  GOTO ERROR
END

/* ERwin Builtin Trigger */
/* tb_cliente tb_tipo_cliente_cliente on child delete no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_cliente"
CHILD_OWNER="", CHILD_TABLE="tb_tipo_cliente_cliente"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_17", FK_COLUMNS="id_cliente" */
IF EXISTS (SELECT * FROM deleted, tb_cliente
  WHERE
    /* %JoinFKPK(deleted, tb_cliente, "=" AND") */
    deleted.id_cliente = tb_cliente.id_cliente AND
    NOT EXISTS (
      SELECT * FROM tb_tipo_cliente_cliente
      WHERE
        /* %JoinFKPK(tb_tipo_cliente_cliente, tb_cliente, "=" AND") */
        tb_tipo_cliente_cliente.id_cliente = tb_cliente.id_cliente
    )
)
BEGIN
  SELECT @errno = 30010,
         @errmsg = 'Cannot delete last tb_tipo_cliente_cliente because tb_cliente exists.'
  GOTO ERROR
END

```

```

/* ERwin Builtin Trigger */
/* tb_tipo_cliente tb_tipo_cliente_cliente on child delete no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_tipo_cliente"
CHILD_OWNER="", CHILD_TABLE="tb_tipo_cliente_cliente"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_16", FK_COLUMNS="id_tipo_cliente" */
IF EXISTS (SELECT * FROM deleted, tb_tipo_cliente
WHERE
/* %JoinFKPK(deleted, tb_tipo_cliente, " = ", "AND") */
deleted.id_tipo_cliente = tb_tipo_cliente.id_tipo_cliente AND
NOT EXISTS (
SELECT * FROM tb_tipo_cliente_cliente
WHERE
/* %JoinFKPK(tb_tipo_cliente_cliente, tb_tipo_cliente, " = ", "AND") */
tb_tipo_cliente_cliente.id_tipo_cliente = tb_tipo_cliente.id_tipo_cliente
)
)
BEGIN
SELECT @errno = 30010,
       @errmsg = 'Cannot delete last tb_tipo_cliente_cliente because tb_tipo_cliente exists.'
GOTO ERROR
END

/* ERwin Builtin Trigger */
RETURN
ERROR:
raiserror @errno @errmsg
rollback transaction
END

go

CREATE TRIGGER tU_tb_tipo_cliente_cliente ON tb_tipo_cliente_cliente FOR UPDATE AS
/* ERwin Builtin Trigger */
/* UPDATE trigger on tb_tipo_cliente_cliente */
BEGIN
DECLARE @NUMROWS int,
        @nullcnt int,
        @validcnt int,
        @insid_tipo_cliente integer,
        @insid_cliente integer,
        @errno int,
        @errmsg varchar(255)

SELECT @NUMROWS = @@rowcount
/* ERwin Builtin Trigger */
/* tb_tipo_cliente_cliente tb_tabela_preco_cliente on parent update no action */
/* ERWIN_RELATION:CHECKSUM="0005f621", PARENT_OWNER="", PARENT_TABLE="tb_tipo_cliente_cliente"
CHILD_OWNER="", CHILD_TABLE="tb_tabela_preco_cliente"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_38", FK_COLUMNS="id_tipo_cliente""id_cliente" */
IF
/* %ParentPK(" OR", UPDATE) */
UPDATE(id_tipo_cliente) OR
UPDATE(id_cliente)
BEGIN
IF EXISTS (
SELECT * FROM deleted, tb_tabela_preco_cliente
WHERE
/* %JoinFKPK(tb_tabela_preco_cliente, deleted, " = ", "AND") */
tb_tabela_preco_cliente.id_tipo_cliente = deleted.id_tipo_cliente AND
tb_tabela_preco_cliente.id_cliente = deleted.id_cliente
)
BEGIN
SELECT @errno = 30005,
       @errmsg = 'Cannot update tb_tipo_cliente_cliente because tb_tabela_preco_cliente exists.'
GOTO ERROR

```

```

END
END

/* ERwin Builtin Trigger */
/* tb_tipo_cliente_cliente tb_tipo_cliente_meio_condicao on parent update no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_tipo_cliente_cliente"
CHILD_OWNER="", CHILD_TABLE="tb_tipo_cliente_meio_condicao"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_28", FK_COLUMNS="id_tipo_cliente""id_cliente" */
IF
/* %ParentPK(" OR",UPDATE) */
UPDATE(id_tipo_cliente) OR
UPDATE(id_cliente)
BEGIN
IF EXISTS (
SELECT * FROM deleted,tb_tipo_cliente_meio_condicao
WHERE
/* %JoinFKPK(tb_tipo_cliente_meio_condicao,deleted," = "," AND") */
tb_tipo_cliente_meio_condicao.id_tipo_cliente = deleted.id_tipo_cliente AND
tb_tipo_cliente_meio_condicao.id_cliente = deleted.id_cliente
)
BEGIN
SELECT @errno = 30005,
@errmsg = 'Cannot update tb_tipo_cliente_cliente because tb_tipo_cliente_meio_condicao exists.'
GOTO ERROR
END
END

/* ERwin Builtin Trigger */
/* tb_cliente tb_tipo_cliente_cliente on child update no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_cliente"
CHILD_OWNER="", CHILD_TABLE="tb_tipo_cliente_cliente"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_17", FK_COLUMNS="id_cliente" */
IF
/* %ChildFK(" OR",UPDATE) */
UPDATE(id_cliente)
BEGIN
SELECT @nullcnt = 0
SELECT @validcnt = count(*)
FROM inserted,tb_cliente
WHERE
/* %JoinFKPK(inserted,tb_cliente) */
inserted.id_cliente = tb_cliente.id_cliente
/* %NotNullFK(inserted," IS NULL","select @nullcnt = count(*) from inserted where"," AND") */

IF @validcnt + @nullcnt != @NUMROWS
BEGIN
SELECT @errno = 30007,
@errmsg = 'Cannot update tb_tipo_cliente_cliente because tb_cliente does not exist.'
GOTO ERROR
END
END

/* ERwin Builtin Trigger */
/* tb_tipo_cliente tb_tipo_cliente_cliente on child update no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_tipo_cliente"
CHILD_OWNER="", CHILD_TABLE="tb_tipo_cliente_cliente"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_16", FK_COLUMNS="id_tipo_cliente" */
IF
/* %ChildFK(" OR",UPDATE) */
UPDATE(id_tipo_cliente)
BEGIN
SELECT @nullcnt = 0
SELECT @validcnt = count(*)
FROM inserted,tb_tipo_cliente
WHERE
/* %JoinFKPK(inserted,tb_tipo_cliente) */
inserted.id_tipo_cliente = tb_tipo_cliente.id_tipo_cliente

```

```

/* %NotNullFK(inserted," IS NULL","select @nullcnt = count(*) from inserted where"," AND") */

IF @validcnt + @nullcnt != @NUMROWS
BEGIN
    SELECT @errno = 30007,
           @errmsg = 'Cannot update tb_tipo_cliente_cliente because tb_tipo_cliente does not exist.'
    GOTO ERROR
END
END
END

```

```

/* ERwin Builtin Trigger */
RETURN
ERROR:
    raiserror @errno @errmsg
    rollback transaction
END

```

```

go

```

```

CREATE TRIGGER tD_tb_tipo_cliente_condicao ON tb_tipo_cliente_condicao FOR DELETE AS
/* ERwin Builtin Trigger */
/* DELETE trigger on tb_tipo_cliente_condicao */
BEGIN
    DECLARE @errno int,
            @errmsg varchar(255)
    /* ERwin Builtin Trigger */
    /* tb_condicao_pagamento tb_tipo_cliente_condicao on child delete no action */
    /* ERWIN_RELATION:CHECKSUM="0002f584", PARENT_OWNER="", PARENT_TABLE="tb_condicao_pagamento"
    CHILD_OWNER="", CHILD_TABLE="tb_tipo_cliente_condicao"
    P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
    FK_CONSTRAINT="R_23", FK_COLUMNS="id_condicao" */
    IF EXISTS (SELECT * FROM deleted, tb_condicao_pagamento
    WHERE
        /* %JoinFKPK(deleted, tb_condicao_pagamento, " = "," AND") */
        deleted.id_condicao = tb_condicao_pagamento.id_condicao AND
        NOT EXISTS (
            SELECT * FROM tb_tipo_cliente_condicao
            WHERE
                /* %JoinFKPK(tb_tipo_cliente_condicao, tb_condicao_pagamento, " = "," AND") */
                tb_tipo_cliente_condicao.id_condicao = tb_condicao_pagamento.id_condicao
        )
    )
BEGIN
    SELECT @errno = 30010,
           @errmsg = 'Cannot delete last tb_tipo_cliente_condicao because tb_condicao_pagamento exists.'
    GOTO ERROR
END

/* ERwin Builtin Trigger */
/* tb_tipo_cliente tb_tipo_cliente_condicao on child delete no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_tipo_cliente"
CHILD_OWNER="", CHILD_TABLE="tb_tipo_cliente_condicao"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_22", FK_COLUMNS="id_tipo_cliente" */
IF EXISTS (SELECT * FROM deleted, tb_tipo_cliente
WHERE
    /* %JoinFKPK(deleted, tb_tipo_cliente, " = "," AND") */
    deleted.id_tipo_cliente = tb_tipo_cliente.id_tipo_cliente AND
    NOT EXISTS (
        SELECT * FROM tb_tipo_cliente_condicao
        WHERE
            /* %JoinFKPK(tb_tipo_cliente_condicao, tb_tipo_cliente, " = "," AND") */
            tb_tipo_cliente_condicao.id_tipo_cliente = tb_tipo_cliente.id_tipo_cliente
    )
)
BEGIN

```

```

SELECT @errno = 30010,
       @errmsg = 'Cannot delete last tb_tipo_cliente_condicao because tb_tipo_cliente exists.'
GOTO ERROR
END

```

```

/* ERwin Builtin Trigger */
RETURN
ERROR:
raiserror @errno @errmsg
rollback transaction
END

```

```

go

```

```

CREATE TRIGGER tU_tb_tipo_cliente_condicao ON tb_tipo_cliente_condicao FOR UPDATE AS
/* ERwin Builtin Trigger */
/* UPDATE trigger on tb_tipo_cliente_condicao */
BEGIN
  DECLARE @NUMROWS int,
          @nullcnt int,
          @validcnt int,
          @insid_tipo_cliente integer,
          @insid_condicao integer,
          @errno int,
          @errmsg varchar(255)

  SELECT @NUMROWS = @@rowcount
  /* ERwin Builtin Trigger */
  /* tb_condicao_pagamento tb_tipo_cliente_condicao on child update no action */
  /* ERWIN_RELATION:CHECKSUM="0002fccf", PARENT_OWNER="", PARENT_TABLE="tb_condicao_pagamento"
  CHILD_OWNER="", CHILD_TABLE="tb_tipo_cliente_condicao"
  P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
  FK_CONSTRAINT="R_23", FK_COLUMNS="id_condicao" */
  IF
    /* %ChildFK(" OR",UPDATE) */
    UPDATE(id_condicao)
  BEGIN
    SELECT @nullcnt = 0
    SELECT @validcnt = count(*)
      FROM inserted,tb_condicao_pagamento
      WHERE
        /* %JoinFKPK(inserted,tb_condicao_pagamento) */
        inserted.id_condicao = tb_condicao_pagamento.id_condicao
    /* %NotNullFK(inserted," IS NULL","select @nullcnt = count(*) from inserted where"," AND") */

    IF @validcnt + @nullcnt != @NUMROWS
    BEGIN
      SELECT @errno = 30007,
             @errmsg = 'Cannot update tb_tipo_cliente_condicao because tb_condicao_pagamento does not exist.'
      GOTO ERROR
    END
  END

```

```

/* ERwin Builtin Trigger */
/* tb_tipo_cliente tb_tipo_cliente_condicao on child update no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_tipo_cliente"
CHILD_OWNER="", CHILD_TABLE="tb_tipo_cliente_condicao"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_22", FK_COLUMNS="id_tipo_cliente" */
IF
  /* %ChildFK(" OR",UPDATE) */
  UPDATE(id_tipo_cliente)
BEGIN
  SELECT @nullcnt = 0
  SELECT @validcnt = count(*)
    FROM inserted,tb_tipo_cliente
    WHERE
      /* %JoinFKPK(inserted,tb_tipo_cliente) */

```



```

        inserted.id_tipo_cliente = tb_tipo_cliente.id_tipo_cliente
/* %%NotNullFK(inserted," IS NULL","select @nullcnt = count(*) from inserted where"," AND") */

```

```

IF @validcnt + @nullcnt != @NUMROWS
BEGIN
    SELECT @errno = 30007,
           @errmsg = 'Cannot update tb_tipo_cliente_condicao because tb_tipo_cliente does not exist.'
    GOTO ERROR
END
END

```

```

/* ERwin Builtin Trigger */
RETURN
ERROR:
    raiserror @errno @errmsg
    rollback transaction
END

```

```

go

```

```

CREATE TRIGGER tD_tb_tipo_cliente_meio_condicao ON tb_tipo_cliente_meio_condicao FOR DELETE AS
/* ERwin Builtin Trigger */
/* DELETE trigger on tb_tipo_cliente_meio_condicao */
BEGIN
    DECLARE @errno int,
            @errmsg varchar(255)
/* ERwin Builtin Trigger */
/* tb_meio_condicao_pagamento tb_tipo_cliente_meio_condicao on child delete no action */
/* ERWIN_RELATION:CHECKSUM="0004536e",PARENT_OWNER="",PARENT_TABLE="tb_meio_condicao_pagamento"
CHILD_OWNER="",CHILD_TABLE="tb_tipo_cliente_meio_condicao"
P2C_VERB_PHRASE="",C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_29",FK_COLUMNS="id_condicao""id_meio_pagamento""id_tipo_cliente" */
IF EXISTS (SELECT * FROM deleted,tb_meio_condicao_pagamento
           WHERE
               /* %%JoinFKPK(deleted,tb_meio_condicao_pagamento,"="," AND") */
               deleted.id_condicao = tb_meio_condicao_pagamento.id_condicao AND
               deleted.id_meio_pagamento = tb_meio_condicao_pagamento.id_meio_pagamento AND
               deleted.id_tipo_cliente = tb_meio_condicao_pagamento.id_tipo_cliente AND
               NOT EXISTS (
                   SELECT * FROM tb_tipo_cliente_meio_condicao
                   WHERE
                       /* %%JoinFKPK(tb_tipo_cliente_meio_condicao,tb_meio_condicao_pagamento,"="," AND") */
                       tb_tipo_cliente_meio_condicao.id_condicao = tb_meio_condicao_pagamento.id_condicao AND
                       tb_tipo_cliente_meio_condicao.id_meio_pagamento = tb_meio_condicao_pagamento.id_meio_pagamento AND
                       tb_tipo_cliente_meio_condicao.id_tipo_cliente = tb_meio_condicao_pagamento.id_tipo_cliente
               )
           )
BEGIN
    SELECT @errno = 30010,
           @errmsg = 'Cannot delete last tb_tipo_cliente_meio_condicao because tb_meio_condicao_pagamento exists.'
    GOTO ERROR
END

```

```

/* ERwin Builtin Trigger */
/* tb_tipo_cliente_cliente tb_tipo_cliente_meio_condicao on child delete no action */
/* ERWIN_RELATION:CHECKSUM="00000000",PARENT_OWNER="",PARENT_TABLE="tb_tipo_cliente_cliente"
CHILD_OWNER="",CHILD_TABLE="tb_tipo_cliente_meio_condicao"
P2C_VERB_PHRASE="",C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_28",FK_COLUMNS="id_tipo_cliente""id_cliente" */
IF EXISTS (SELECT * FROM deleted,tb_tipo_cliente_cliente
           WHERE
               /* %%JoinFKPK(deleted,tb_tipo_cliente_cliente,"="," AND") */
               deleted.id_tipo_cliente = tb_tipo_cliente_cliente.id_tipo_cliente AND
               deleted.id_cliente = tb_tipo_cliente_cliente.id_cliente AND
               NOT EXISTS (
                   SELECT * FROM tb_tipo_cliente_meio_condicao

```

```

WHERE
/* %JoinFKPK(tb_tipo_cliente_meio_condicao,tb_tipo_cliente_cliente,"=","AND") */
tb_tipo_cliente_meio_condicao.id_tipo_cliente = tb_tipo_cliente_cliente.id_tipo_cliente AND
tb_tipo_cliente_meio_condicao.id_cliente = tb_tipo_cliente_cliente.id_cliente
)
)
BEGIN
SELECT @errno = 30010,
       @errmsg = 'Cannot delete last tb_tipo_cliente_meio_condicao because tb_tipo_cliente_cliente exists.'
GOTO ERROR
END

/* ERwin Builtin Trigger */
RETURN
ERROR:
raiserror @errno @errmsg
rollback transaction
END

go

CREATE TRIGGER tU_tb_tipo_cliente_meio_condicao ON tb_tipo_cliente_meio_condicao FOR UPDATE AS
/* ERwin Builtin Trigger */
/* UPDATE trigger on tb_tipo_cliente_meio_condicao */
BEGIN
DECLARE @NUMROWS int,
        @nullcnt int,
        @validcnt int,
        @insid_tipo_cliente integer,
        @insid_cliente integer,
        @insid_condicao integer,
        @insid_meio_pagamento integer,
        @errno int,
        @errmsg varchar(255)

SELECT @NUMROWS = @@rowcount
/* ERwin Builtin Trigger */
/* tb_meio_condicao_pagamento tb_tipo_cliente_meio_condicao on child update no action */
/* ERWIN_RELATION:CHECKSUM="0003e729", PARENT_OWNER="", PARENT_TABLE="tb_meio_condicao_pagamento"
CHILD_OWNER="", CHILD_TABLE="tb_tipo_cliente_meio_condicao"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_29", FK_COLUMNS="id_condicao""id_meio_pagamento""id_tipo_cliente" */
IF
/* %ChildFK(" OR",UPDATE) */
UPDATE(id_condicao) OR
UPDATE(id_meio_pagamento) OR
UPDATE(id_tipo_cliente)
BEGIN
SELECT @nullcnt = 0
SELECT @validcnt = count(*)
FROM inserted,tb_meio_condicao_pagamento
WHERE
/* %JoinFKPK(inserted,tb_meio_condicao_pagamento) */
inserted.id_condicao = tb_meio_condicao_pagamento.id_condicao and
inserted.id_meio_pagamento = tb_meio_condicao_pagamento.id_meio_pagamento and
inserted.id_tipo_cliente = tb_meio_condicao_pagamento.id_tipo_cliente
/* %NotNullFK(inserted," IS NULL","select @nullcnt = count(*) from inserted where","AND") */

IF @validcnt + @nullcnt != @NUMROWS
BEGIN
SELECT @errno = 30007,
       @errmsg = 'Cannot update tb_tipo_cliente_meio_condicao because tb_meio_condicao_pagamento does not exist.'
GOTO ERROR
END
END

/* ERwin Builtin Trigger */
/* tb_tipo_cliente_cliente tb_tipo_cliente_meio_condicao on child update no action */

```

```

/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_tipo_cliente_cliente"
  CHILD_OWNER="", CHILD_TABLE="tb_tipo_cliente_meio_condicao"
  P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
  FK_CONSTRAINT="R_28", FK_COLUMNS="id_tipo_cliente""id_cliente" */
IF
/* %ChildFK(" OR",UPDATE) */
UPDATE(id_tipo_cliente) OR
UPDATE(id_cliente)
BEGIN
SELECT @nullcnt = 0
SELECT @validcnt = count(*)
  FROM inserted,tb_tipo_cliente_cliente
  WHERE
    /* %JoinFKPK(inserted,tb_tipo_cliente_cliente) */
    inserted.id_tipo_cliente = tb_tipo_cliente_cliente.id_tipo_cliente and
    inserted.id_cliente = tb_tipo_cliente_cliente.id_cliente
/* %NotNullFK(inserted," IS NULL","select @nullcnt = count(*) from inserted where"," AND") */

IF @validcnt + @nullcnt != @NUMROWS
BEGIN
  SELECT @errno = 30007,
    @errmsg = 'Cannot update tb_tipo_cliente_meio_condicao because tb_tipo_cliente_cliente does not exist.'
  GOTO ERROR
END
END

/* ERwin Builtin Trigger */
RETURN
ERROR:
  raiserror @errno @errmsg
  rollback transaction
END

go

```

```

CREATE TRIGGER tD_tb_tipo_movimentacao_estoque ON tb_tipo_movimentacao_estoque FOR DELETE AS
/* ERwin Builtin Trigger */
/* DELETE trigger on tb_tipo_movimentacao_estoque */
BEGIN
  DECLARE @errno int,
    @errmsg varchar(255)
  /* ERwin Builtin Trigger */
  /* tb_tipo_movimentacao_estoque tb_produto_insumo_movimentacao_estoque on parent delete no action */
  /* ERWIN_RELATION:CHECKSUM="0002f87c", PARENT_OWNER="", PARENT_TABLE="tb_tipo_movimentacao_estoque"
  CHILD_OWNER="", CHILD_TABLE="tb_produto_insumo_movimentacao_estoque"
  P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
  FK_CONSTRAINT="R_48", FK_COLUMNS="id_tipo_movimentacao_estoque" */
  IF EXISTS (
    SELECT * FROM deleted,tb_produto_insumo_movimentacao_estoque
    WHERE
      /* %JoinFKPK(tb_produto_insumo_movimentacao_estoque,deleted," = "," AND") */
      tb_produto_insumo_movimentacao_estoque.id_tipo_movimentacao_estoque = deleted.id_tipo_movimentacao_estoque
  )
  BEGIN
    SELECT @errno = 30001,
      @errmsg = 'Cannot delete tb_tipo_movimentacao_estoque because tb_produto_insumo_movimentacao_estoque exists.'
    GOTO ERROR
  END

  /* ERwin Builtin Trigger */
  /* tb_tipo_movimentacao_estoque tb_produto_acabado_movimentacao_estoque on parent delete no action */
  /* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_tipo_movimentacao_estoque"
  CHILD_OWNER="", CHILD_TABLE="tb_produto_acabado_movimentacao_estoque"
  P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
  FK_CONSTRAINT="R_44", FK_COLUMNS="id_tipo_movimentacao_estoque" */
  IF EXISTS (

```

```

SELECT * FROM deleted,tb_produto_acabado_movimentacao_estoque
WHERE
/* %JoinFKPK(tb_produto_acabado_movimentacao_estoque,deleted,"=","AND") */
tb_produto_acabado_movimentacao_estoque.id_tipo_movimentacao_estoque = deleted.id_tipo_movimentacao_estoque
)
BEGIN
SELECT @errno = 30001,
       @errmsg = 'Cannot delete tb_tipo_movimentacao_estoque because tb_produto_acabado_movimentacao_estoque exists.'
GOTO ERROR
END

```

```

/* ERwin Builtin Trigger */
RETURN
ERROR:
raiserror @errno @errmsg
rollback transaction
END

```

go

```

CREATE TRIGGER tU_tb_tipo_movimentacao_estoque ON tb_tipo_movimentacao_estoque FOR UPDATE AS

```

```

/* ERwin Builtin Trigger */
/* UPDATE trigger on tb_tipo_movimentacao_estoque */
BEGIN
DECLARE @NUMROWS int,
        @nullcnt int,
        @validcnt int,
        @insid_tipo_movimentacao_estoque integer,
        @errno int,
        @errmsg varchar(255)

```

```

SELECT @NUMROWS = @@rowcount
/* ERwin Builtin Trigger */
/* tb_tipo_movimentacao_estoque tb_produto_insumo_movimentacao_estoque on parent update no action */
/* ERWIN_RELATION:CHECKSUM="0003346e", PARENT_OWNER="", PARENT_TABLE="tb_tipo_movimentacao_estoque"
CHILD_OWNER="", CHILD_TABLE="tb_produto_insumo_movimentacao_estoque"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_48", FK_COLUMNS="id_tipo_movimentacao_estoque" */

```

```

IF
/* %ParentPK(" OR",UPDATE) */
UPDATE(id_tipo_movimentacao_estoque)

```

```

BEGIN
IF EXISTS (
SELECT * FROM deleted,tb_produto_insumo_movimentacao_estoque
WHERE
/* %JoinFKPK(tb_produto_insumo_movimentacao_estoque,deleted,"=","AND") */
tb_produto_insumo_movimentacao_estoque.id_tipo_movimentacao_estoque = deleted.id_tipo_movimentacao_estoque
)
BEGIN
SELECT @errno = 30005,
       @errmsg = 'Cannot update tb_tipo_movimentacao_estoque because tb_produto_insumo_movimentacao_estoque exists.'
GOTO ERROR
END
END

```

```

/* ERwin Builtin Trigger */
/* tb_tipo_movimentacao_estoque tb_produto_acabado_movimentacao_estoque on parent update no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_tipo_movimentacao_estoque"
CHILD_OWNER="", CHILD_TABLE="tb_produto_acabado_movimentacao_estoque"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_44", FK_COLUMNS="id_tipo_movimentacao_estoque" */

```

```

IF
/* %ParentPK(" OR",UPDATE) */
UPDATE(id_tipo_movimentacao_estoque)
BEGIN
IF EXISTS (
SELECT * FROM deleted,tb_produto_acabado_movimentacao_estoque
WHERE

```

```

/* %JoinFKPK(tb_produto_acabado_movimentacao_estoque,deleted,"=","AND") */
tb_produto_acabado_movimentacao_estoque.id_tipo_movimentacao_estoque = deleted.id_tipo_movimentacao_estoque
)
BEGIN
SELECT @errno = 30005,
       @errmsg = 'Cannot update tb_tipo_movimentacao_estoque because tb_produto_acabado_movimentacao_estoque exists.'
GOTO ERROR
END
END

```

```

/* ERwin Builtin Trigger */
RETURN
ERROR:
  raiserror @errno @errmsg
  rollback transaction
END

```

go

```

CREATE TRIGGER tD_tb_tipo_produto ON tb_tipo_produto FOR DELETE AS
/* ERwin Builtin Trigger */
/* DELETE trigger on tb_tipo_produto */
BEGIN
  DECLARE @errno int,
          @errmsg varchar(255)
  /* ERwin Builtin Trigger */
  /* tb_tipo_produto tb_produto on parent delete no action */
  /* ERWIN_RELATION:CHECKSUM="0000f8d", PARENT_OWNER="", PARENT_TABLE="tb_tipo_produto"
  CHILD_OWNER="", CHILD_TABLE="tb_produto"
  P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
  FK_CONSTRAINT="R_30", FK_COLUMNS="id_tipo_produto" */
  IF EXISTS (
    SELECT * FROM deleted, tb_produto
    WHERE
      /* %JoinFKPK(tb_produto,deleted,"=","AND") */
      tb_produto.id_tipo_produto = deleted.id_tipo_produto
  )
  BEGIN
    SELECT @errno = 30001,
           @errmsg = 'Cannot delete tb_tipo_produto because tb_produto exists.'
    GOTO ERROR
  END

```

```

/* ERwin Builtin Trigger */
RETURN
ERROR:
  raiserror @errno @errmsg
  rollback transaction
END

```

go

```

CREATE TRIGGER tU_tb_tipo_produto ON tb_tipo_produto FOR UPDATE AS
/* ERwin Builtin Trigger */
/* UPDATE trigger on tb_tipo_produto */
BEGIN
  DECLARE @NUMROWS int,
          @nullcnt int,
          @validcnt int,
          @insid_tipo_produto char(18),
          @errno int,
          @errmsg varchar(255)

  SELECT @NUMROWS = @@rowcount

```

```

/* ERwin Builtin Trigger */
/* tb_tipo_produto tb_produto on parent update no action */
/* ERWIN_RELATION:CHECKSUM="00011da6", PARENT_OWNER="", PARENT_TABLE="tb_tipo_produto"
CHILD_OWNER="", CHILD_TABLE="tb_produto"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_30", FK_COLUMNS="id_tipo_produto" */
IF
/* %ParentPK(" OR",UPDATE) */
UPDATE(id_tipo_produto)
BEGIN
IF EXISTS (
SELECT * FROM deleted,tb_produto
WHERE
/* %JoinFKPK(tb_produto,deleted," = "," AND") */
tb_produto.id_tipo_produto = deleted.id_tipo_produto
)
BEGIN
SELECT @errno = 30005,
@errmsg = 'Cannot update tb_tipo_produto because tb_produto exists.'
GOTO ERROR
END
END
END

/* ERwin Builtin Trigger */
RETURN
ERROR:
raiserror @errno @errmsg
rollback transaction
END

go

```

```

CREATE TRIGGER tD_tb_unidade_medida ON tb_unidade_medida FOR DELETE AS
/* ERwin Builtin Trigger */
/* DELETE trigger on tb_unidade_medida */
BEGIN
DECLARE @errno int,
@errmsg varchar(255)
/* ERwin Builtin Trigger */
/* tb_unidade_medida tb_produto_insumo_estoque on parent delete no action */
/* ERWIN_RELATION:CHECKSUM="00049c7e", PARENT_OWNER="", PARENT_TABLE="tb_unidade_medida"
CHILD_OWNER="", CHILD_TABLE="tb_produto_insumo_estoque"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_46", FK_COLUMNS="id_unidade_medida" */
IF EXISTS (
SELECT * FROM deleted,tb_produto_insumo_estoque
WHERE
/* %JoinFKPK(tb_produto_insumo_estoque,deleted," = "," AND") */
tb_produto_insumo_estoque.id_unidade_medida = deleted.id_unidade_medida
)
BEGIN
SELECT @errno = 30001,
@errmsg = 'Cannot delete tb_unidade_medida because tb_produto_insumo_estoque exists.'
GOTO ERROR
END
END

/* ERwin Builtin Trigger */
/* tb_unidade_medida tb_produto_acabado_estoque on parent delete no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_unidade_medida"
CHILD_OWNER="", CHILD_TABLE="tb_produto_acabado_estoque"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_42", FK_COLUMNS="id_unidade_medida" */
IF EXISTS (
SELECT * FROM deleted,tb_produto_acabado_estoque
WHERE
/* %JoinFKPK(tb_produto_acabado_estoque,deleted," = "," AND") */

```

```

        tb_produto_acabado_estoque.id_unidade_medida = deleted.id_unidade_medida
    )
BEGIN
    SELECT @errno = 30001,
           @errmsg = 'Cannot delete tb_unidade_medida because tb_produto_acabado_estoque exists.'
    GOTO ERROR
END

/* ERwin Builtin Trigger */
/* tb_unidade_medida tb_produto_ficha_tecnica on parent delete no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_unidade_medida"
CHILD_OWNER="", CHILD_TABLE="tb_produto_ficha_tecnica"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_34", FK_COLUMNS="id_unidade_medida" */
IF EXISTS (
    SELECT * FROM deleted,tb_produto_ficha_tecnica
    WHERE
        /* %JoinFKPK(tb_produto_ficha_tecnica,deleted,"=","AND") */
        tb_produto_ficha_tecnica.id_unidade_medida = deleted.id_unidade_medida
    )
BEGIN
    SELECT @errno = 30001,
           @errmsg = 'Cannot delete tb_unidade_medida because tb_produto_ficha_tecnica exists.'
    GOTO ERROR
END

/* ERwin Builtin Trigger */
/* tb_unidade_medida tb_produto on parent delete no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_unidade_medida"
CHILD_OWNER="", CHILD_TABLE="tb_produto"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_31", FK_COLUMNS="id_unidade_medida" */
IF EXISTS (
    SELECT * FROM deleted,tb_produto
    WHERE
        /* %JoinFKPK(tb_produto,deleted,"=","AND") */
        tb_produto.id_unidade_medida = deleted.id_unidade_medida
    )
BEGIN
    SELECT @errno = 30001,
           @errmsg = 'Cannot delete tb_unidade_medida because tb_produto exists.'
    GOTO ERROR
END

/* ERwin Builtin Trigger */
RETURN
ERROR:
    raiserror @errno @errmsg
    rollback transaction
END

```

go

```

CREATE TRIGGER tU_tb_unidade_medida ON tb_unidade_medida FOR UPDATE AS
/* ERwin Builtin Trigger */
/* UPDATE trigger on tb_unidade_medida */
BEGIN
    DECLARE @NUMROWS int,
            @nullcnt int,
            @validcnt int,
            @insid_unidade_medida integer,
            @errno int,
            @errmsg varchar(255)

    SELECT @NUMROWS = @@rowcount
    /* ERwin Builtin Trigger */
    /* tb_unidade_medida tb_produto_insumo_estoque on parent update no action */
    /* ERWIN_RELATION:CHECKSUM="00050361", PARENT_OWNER="", PARENT_TABLE="tb_unidade_medida"

```

```

CHILD_OWNER="", CHILD_TABLE="tb_produto_insumo_estoque"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_46", FK_COLUMNS="id_unidade_medida" */
IF
/* %ParentPK(" OR",UPDATE) */
UPDATE(id_unidade_medida)
BEGIN
IF EXISTS (
SELECT * FROM deleted,tb_produto_insumo_estoque
WHERE
/* %JoinFKPK(tb_produto_insumo_estoque,deleted,"=","AND") */
tb_produto_insumo_estoque.id_unidade_medida = deleted.id_unidade_medida
)
BEGIN
SELECT @errno = 30005,
@errmsg = 'Cannot update tb_unidade_medida because tb_produto_insumo_estoque exists.'
GOTO ERROR
END
END

/* ERwin Builtin Trigger */
/* tb_unidade_medida tb_produto_acabado_estoque on parent update no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_unidade_medida"
CHILD_OWNER="", CHILD_TABLE="tb_produto_acabado_estoque"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_42", FK_COLUMNS="id_unidade_medida" */
IF
/* %ParentPK(" OR",UPDATE) */
UPDATE(id_unidade_medida)
BEGIN
IF EXISTS (
SELECT * FROM deleted,tb_produto_acabado_estoque
WHERE
/* %JoinFKPK(tb_produto_acabado_estoque,deleted,"=","AND") */
tb_produto_acabado_estoque.id_unidade_medida = deleted.id_unidade_medida
)
BEGIN
SELECT @errno = 30005,
@errmsg = 'Cannot update tb_unidade_medida because tb_produto_acabado_estoque exists.'
GOTO ERROR
END
END

/* ERwin Builtin Trigger */
/* tb_unidade_medida tb_produto_ficha_tecnica on parent update no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_unidade_medida"
CHILD_OWNER="", CHILD_TABLE="tb_produto_ficha_tecnica"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_34", FK_COLUMNS="id_unidade_medida" */
IF
/* %ParentPK(" OR",UPDATE) */
UPDATE(id_unidade_medida)
BEGIN
IF EXISTS (
SELECT * FROM deleted,tb_produto_ficha_tecnica
WHERE
/* %JoinFKPK(tb_produto_ficha_tecnica,deleted,"=","AND") */
tb_produto_ficha_tecnica.id_unidade_medida = deleted.id_unidade_medida
)
BEGIN
SELECT @errno = 30005,
@errmsg = 'Cannot update tb_unidade_medida because tb_produto_ficha_tecnica exists.'
GOTO ERROR
END
END

/* ERwin Builtin Trigger */
/* tb_unidade_medida tb_produto on parent update no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_unidade_medida"
CHILD_OWNER="", CHILD_TABLE="tb_produto"

```



```

P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_31", FK_COLUMNS="id_unidade_medida" */
IF
/* %ParentPK(" OR",UPDATE) */
UPDATE(id_unidade_medida)
BEGIN
IF EXISTS (
SELECT * FROM deleted,tb_produto
WHERE
/* %JoinFKPK(tb_produto,deleted," = "" AND") */
tb_produto.id_unidade_medida = deleted.id_unidade_medida
)
BEGIN
SELECT @errno = 30005,
@errmsg = 'Cannot update tb_unidade_medida because tb_produto exists.'
GOTO ERROR
END
END
END

```

```

/* ERwin Builtin Trigger */
RETURN
ERROR:
raiserror @errno @errmsg
rollback transaction
END

```

go

```

CREATE TRIGGER tD_tb_usuario ON tb_usuario FOR DELETE AS
/* ERwin Builtin Trigger */
/* DELETE trigger on tb_usuario */
BEGIN
DECLARE @errno int,
@errmsg varchar(255)
/* ERwin Builtin Trigger */
/* tb_usuario tb_usuario_estrutura_dados on parent delete no action */
/* ERWIN_RELATION:CHECKSUM="00034ceb", PARENT_OWNER="", PARENT_TABLE="tb_usuario"
CHILD_OWNER="", CHILD_TABLE="tb_usuario_estrutura_dados"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_15", FK_COLUMNS="id_usuario" */
IF EXISTS (
SELECT * FROM deleted,tb_usuario_estrutura_dados
WHERE
/* %JoinFKPK(tb_usuario_estrutura_dados,deleted," = "" AND") */
tb_usuario_estrutura_dados.id_usuario = deleted.id_usuario
)
BEGIN
SELECT @errno = 30001,
@errmsg = 'Cannot delete tb_usuario because tb_usuario_estrutura_dados exists.'
GOTO ERROR
END

/* ERwin Builtin Trigger */
/* tb_usuario tb_log_auditoria on parent delete no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_usuario"
CHILD_OWNER="", CHILD_TABLE="tb_log_auditoria"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_13", FK_COLUMNS="id_usuario" */
IF EXISTS (
SELECT * FROM deleted,tb_log_auditoria
WHERE
/* %JoinFKPK(tb_log_auditoria,deleted," = "" AND") */
tb_log_auditoria.id_usuario = deleted.id_usuario
)
BEGIN
SELECT @errno = 30001,

```

```

        @errmsg = 'Cannot delete tb_usuario because tb_log_auditoria exists.'
    GOTO ERROR
END

/* ERwin Builtin Trigger */
/* tb_usuario tb_usuario_estrutura_sistema on parent delete no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_usuario"
CHILD_OWNER="", CHILD_TABLE="tb_usuario_estrutura_sistema"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_11", FK_COLUMNS="id_usuario" */
IF EXISTS (
    SELECT * FROM deleted,tb_usuario_estrutura_sistema
    WHERE
        /* %JoinFKPK(tb_usuario_estrutura_sistema,deleted," = ","AND") */
        tb_usuario_estrutura_sistema.id_usuario = deleted.id_usuario
)
BEGIN
    SELECT @errno = 30001,
        @errmsg = 'Cannot delete tb_usuario because tb_usuario_estrutura_sistema exists.'
    GOTO ERROR
END

/* ERwin Builtin Trigger */
RETURN
ERROR:
    raiserror @errno @errmsg
    rollback transaction
END

```

go

```

CREATE TRIGGER tU_tb_usuario ON tb_usuario FOR UPDATE AS
/* ERwin Builtin Trigger */
/* UPDATE trigger on tb_usuario */
BEGIN
    DECLARE @NUMROWS int,
        @nullcnt int,
        @validcnt int,
        @insid_usuario integer,
        @errno int,
        @errmsg varchar(255)

    SELECT @NUMROWS = @@rowcount
    /* ERwin Builtin Trigger */
    /* tb_usuario tb_usuario_estrutura_dados on parent update no action */
    /* ERWIN_RELATION:CHECKSUM="000399ce", PARENT_OWNER="", PARENT_TABLE="tb_usuario"
    CHILD_OWNER="", CHILD_TABLE="tb_usuario_estrutura_dados"
    P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
    FK_CONSTRAINT="R_15", FK_COLUMNS="id_usuario" */
    IF
        /* %ParentPK(" OR",UPDATE) */
        UPDATE(id_usuario)
    BEGIN
        IF EXISTS (
            SELECT * FROM deleted,tb_usuario_estrutura_dados
            WHERE
                /* %JoinFKPK(tb_usuario_estrutura_dados,deleted," = ","AND") */
                tb_usuario_estrutura_dados.id_usuario = deleted.id_usuario
        )
        BEGIN
            SELECT @errno = 30005,
                @errmsg = 'Cannot update tb_usuario because tb_usuario_estrutura_dados exists.'
            GOTO ERROR
        END
    END
END

/* ERwin Builtin Trigger */
/* tb_usuario tb_log_auditoria on parent update no action */

```

```

/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_usuario"
CHILD_OWNER="", CHILD_TABLE="tb_log_auditoria"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_13", FK_COLUMNS="id_usuario" */
IF
/* %ParentPK(" OR",UPDATE) */
UPDATE(id_usuario)
BEGIN
IF EXISTS (
SELECT * FROM deleted,tb_log_auditoria
WHERE
/* %JoinFKPK(tb_log_auditoria,deleted," = "," AND") */
tb_log_auditoria.id_usuario = deleted.id_usuario
)
BEGIN
SELECT @errno = 30005,
@errmsg = 'Cannot update tb_usuario because tb_log_auditoria exists.'
GOTO ERROR
END
END

```

```

/* ERwin Builtin Trigger */
/* tb_usuario tb_usuario_estrutura_sistema on parent update no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_usuario"
CHILD_OWNER="", CHILD_TABLE="tb_usuario_estrutura_sistema"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_11", FK_COLUMNS="id_usuario" */
IF
/* %ParentPK(" OR",UPDATE) */
UPDATE(id_usuario)
BEGIN
IF EXISTS (
SELECT * FROM deleted,tb_usuario_estrutura_sistema
WHERE
/* %JoinFKPK(tb_usuario_estrutura_sistema,deleted," = "," AND") */
tb_usuario_estrutura_sistema.id_usuario = deleted.id_usuario
)
BEGIN
SELECT @errno = 30005,
@errmsg = 'Cannot update tb_usuario because tb_usuario_estrutura_sistema exists.'
GOTO ERROR
END
END

```

```

/* ERwin Builtin Trigger */
RETURN
ERROR:
raiserror @errno @errmsg
rollback transaction
END

```

go

```

CREATE TRIGGER tD_tb_usuario_estrutura_dados ON tb_usuario_estrutura_dados FOR DELETE AS
/* ERwin Builtin Trigger */
/* DELETE trigger on tb_usuario_estrutura_dados */
BEGIN
DECLARE @errno int,
@errmsg varchar(255)
/* ERwin Builtin Trigger */
/* tb_usuario tb_usuario_estrutura_dados on child delete no action */
/* ERWIN_RELATION:CHECKSUM="0003b8a4", PARENT_OWNER="", PARENT_TABLE="tb_usuario"
CHILD_OWNER="", CHILD_TABLE="tb_usuario_estrutura_dados"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_15", FK_COLUMNS="id_usuario" */
IF EXISTS (SELECT * FROM deleted,tb_usuario

```

```

WHERE
/* %JoinFKPK(deleted,tb_usuario,"=","AND") */
deleted.id_usuario = tb_usuario.id_usuario AND
NOT EXISTS (
SELECT * FROM tb_usuario_estrutura_dados
WHERE
/* %JoinFKPK(tb_usuario_estrutura_dados,tb_usuario,"=","AND") */
tb_usuario_estrutura_dados.id_usuario = tb_usuario.id_usuario
)
)
BEGIN
SELECT @errno = 30010,
       @errmsg = 'Cannot delete last tb_usuario_estrutura_dados because tb_usuario exists.'
GOTO ERROR
END

/* ERwin Builtin Trigger */
/* tb_estrutura_dados_sensíveis Campos tb_usuario_estrutura_dados on child delete no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_estrutura_dados_sensíveis Campos"
CHILD_OWNER="", CHILD_TABLE="tb_usuario_estrutura_dados"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_14", FK_COLUMNS="id_estrutura_campos" id_estrutura_dados */
IF EXISTS (SELECT * FROM deleted,tb_estrutura_dados_sensíveis Campos
WHERE
/* %JoinFKPK(deleted,tb_estrutura_dados_sensíveis Campos,"=","AND") */
deleted.id_estrutura_campos = tb_estrutura_dados_sensíveis Campos.id_estrutura_campos AND
deleted.id_estrutura_dados = tb_estrutura_dados_sensíveis Campos.id_estrutura_dados AND
NOT EXISTS (
SELECT * FROM tb_usuario_estrutura_dados
WHERE
/* %JoinFKPK(tb_usuario_estrutura_dados,tb_estrutura_dados_sensíveis Campos,"=","AND") */
tb_usuario_estrutura_dados.id_estrutura_campos = tb_estrutura_dados_sensíveis Campos.id_estrutura_campos AND
tb_usuario_estrutura_dados.id_estrutura_dados = tb_estrutura_dados_sensíveis Campos.id_estrutura_dados
)
)
BEGIN
SELECT @errno = 30010,
       @errmsg = 'Cannot delete last tb_usuario_estrutura_dados because tb_estrutura_dados_sensíveis Campos exists.'
GOTO ERROR
END

/* ERwin Builtin Trigger */
RETURN
ERROR:
raiserror @errno @errmsg
rollback transaction
END

```

go

```

CREATE TRIGGER tU_tb_usuario_estrutura_dados ON tb_usuario_estrutura_dados FOR UPDATE AS
/* ERwin Builtin Trigger */
/* UPDATE trigger on tb_usuario_estrutura_dados */
BEGIN
DECLARE @NUMROWS int,
        @nullcnt int,
        @validcnt int,
        @insid_estrutura_campos integer,
        @insid_estrutura_dados integer,
        @insid_usuario integer,
        @errno int,
        @errmsg varchar(255)

SELECT @NUMROWS = @@rowcount
/* ERwin Builtin Trigger */
/* tb_usuario tb_usuario_estrutura_dados on child update no action */
/* ERWIN_RELATION:CHECKSUM="0003697a", PARENT_OWNER="", PARENT_TABLE="tb_usuario"
CHILD_OWNER="", CHILD_TABLE="tb_usuario_estrutura_dados"

```

```

P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_15", FK_COLUMNS="id_usuario" */
IF
/* %ChildFK(" OR",UPDATE) */
UPDATE(id_usuario)
BEGIN
SELECT @nullcnt = 0
SELECT @validcnt = count(*)
FROM inserted,tb_usuario
WHERE
/* %JoinFKPK(inserted,tb_usuario) */
inserted.id_usuario = tb_usuario.id_usuario
/* %NotNullFK(inserted," IS NULL","select @nullcnt = count(*) from inserted where"," AND") */

IF @validcnt + @nullcnt != @NUMROWS
BEGIN
SELECT @errno = 30007,
@errmsg = 'Cannot update tb_usuario_estrutura_dados because tb_usuario does not exist'
GOTO ERROR
END
END

/* ERwin Builtin Trigger */
/* tb_estrutura_dados_sensiveis_campos tb_usuario_estrutura_dados on child update no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_estrutura_dados_sensiveis_campos"
CHILD_OWNER="", CHILD_TABLE="tb_usuario_estrutura_dados"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_14", FK_COLUMNS="id_estrutura_campos" id_estrutura_dados" */
IF
/* %ChildFK(" OR",UPDATE) */
UPDATE(id_estrutura_campos) OR
UPDATE(id_estrutura_dados)
BEGIN
SELECT @nullcnt = 0
SELECT @validcnt = count(*)
FROM inserted,tb_estrutura_dados_sensiveis_campos
WHERE
/* %JoinFKPK(inserted,tb_estrutura_dados_sensiveis_campos) */
inserted.id_estrutura_campos = tb_estrutura_dados_sensiveis_campos.id_estrutura_campos and
inserted.id_estrutura_dados = tb_estrutura_dados_sensiveis_campos.id_estrutura_dados
/* %NotNullFK(inserted," IS NULL","select @nullcnt = count(*) from inserted where"," AND") */

IF @validcnt + @nullcnt != @NUMROWS
BEGIN
SELECT @errno = 30007,
@errmsg = 'Cannot update tb_usuario_estrutura_dados because tb_estrutura_dados_sensiveis_campos does not exist.'
GOTO ERROR
END
END

/* ERwin Builtin Trigger */
RETURN
ERROR:
raiserror @errno @errmsg
rollback transaction
END

go

CREATE TRIGGER tD_tb_usuario_estrutura_sistema ON tb_usuario_estrutura_sistema FOR DELETE AS
/* ERwin Builtin Trigger */
/* DELETE trigger on tb_usuario_estrutura_sistema */
BEGIN
DECLARE @errno int,
@errmsg varchar(255)
/* ERwin Builtin Trigger */

```

```

/* tb_usuario tb_usuario_estrutura_sistema on child delete no action */
/* ERWIN_RELATION:CHECKSUM="0003668c", PARENT_OWNER="", PARENT_TABLE="tb_usuario"
CHILD_OWNER="", CHILD_TABLE="tb_usuario_estrutura_sistema"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_11", FK_COLUMNS="id_usuario" */
IF EXISTS (SELECT * FROM deleted, tb_usuario
WHERE
/* %JoinFKPK(deleted, tb_usuario, " = " AND") */
deleted.id_usuario = tb_usuario.id_usuario AND
NOT EXISTS (
SELECT * FROM tb_usuario_estrutura_sistema
WHERE
/* %JoinFKPK(tb_usuario_estrutura_sistema, tb_usuario, " = " AND") */
tb_usuario_estrutura_sistema.id_usuario = tb_usuario.id_usuario
)
)
BEGIN
SELECT @errno = 30010,
@errmsg = 'Cannot delete last tb_usuario_estrutura_sistema because tb_usuario exists.'
GOTO ERROR
END

/* ERwin Builtin Trigger */
/* tb_estrutura_sistema tb_usuario_estrutura_sistema on child delete no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_estrutura_sistema"
CHILD_OWNER="", CHILD_TABLE="tb_usuario_estrutura_sistema"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_10", FK_COLUMNS="id_estrutura_sistema""id_modulo" */
IF EXISTS (SELECT * FROM deleted, tb_estrutura_sistema
WHERE
/* %JoinFKPK(deleted, tb_estrutura_sistema, " = " AND") */
deleted.id_estrutura_sistema = tb_estrutura_sistema.id_estrutura_sistema AND
deleted.id_modulo = tb_estrutura_sistema.id_modulo AND
NOT EXISTS (
SELECT * FROM tb_usuario_estrutura_sistema
WHERE
/* %JoinFKPK(tb_usuario_estrutura_sistema, tb_estrutura_sistema, " = " AND") */
tb_usuario_estrutura_sistema.id_estrutura_sistema = tb_estrutura_sistema.id_estrutura_sistema AND
tb_usuario_estrutura_sistema.id_modulo = tb_estrutura_sistema.id_modulo
)
)
BEGIN
SELECT @errno = 30010,
@errmsg = 'Cannot delete last tb_usuario_estrutura_sistema because tb_estrutura_sistema exists.'
GOTO ERROR
END

/* ERwin Builtin Trigger */
RETURN
ERROR:
raiserror @errno @errmsg
rollback transaction
END

go

```

```

CREATE TRIGGER tU_tb_usuario_estrutura_sistema ON tb_usuario_estrutura_sistema FOR UPDATE AS
/* ERwin Builtin Trigger */
/* UPDATE trigger on tb_usuario_estrutura_sistema */
BEGIN
DECLARE @NUMROWS int,
@nullcnt int,
@validcnt int,
@insid_estrutura_sistema integer,
@insid_modulo integer,
@insid_usuario integer,
@errno int,
@errmsg varchar(255)

```

```

SELECT @NUMROWS = @@rowcount
/* ERwin Builtin Trigger */
/* tb_usuario tb_usuario estrutura sistema on child update no action */
/* ERWIN_RELATION:CHECKSUM="000323f9", PARENT_OWNER="", PARENT_TABLE="tb_usuario"
  CHILD_OWNER="", CHILD_TABLE="tb_usuario_estrutura_sistema"
  P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
  FK_CONSTRAINT="R_11", FK_COLUMNS="id_usuario" */
IF
/* %ChildFK(" OR",UPDATE) */
UPDATE(id_usuario)
BEGIN
SELECT @nullcnt = 0
SELECT @validcnt = count(*)
FROM inserted,tb_usuario
WHERE
/* %JoinFKPK(inserted,tb_usuario) */
inserted.id_usuario = tb_usuario.id_usuario
/* %NotNullFK(inserted," IS NULL","select @nullcnt = count(*) from inserted where"," AND") */

IF @validcnt + @nullcnt != @NUMROWS
BEGIN
SELECT @errno = 30007,
       @errmsg = 'Cannot update tb_usuario_estrutura_sistema because tb_usuario does not exist.'
GOTO ERROR
END
END

/* ERwin Builtin Trigger */
/* tb_estrutura_sistema tb_usuario_estrutura_sistema on child update no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="tb_estrutura_sistema"
  CHILD_OWNER="", CHILD_TABLE="tb_usuario_estrutura_sistema"
  P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
  FK_CONSTRAINT="R_10", FK_COLUMNS="id_estrutura_sistema""id_modulo" */
IF
/* %ChildFK(" OR",UPDATE) */
UPDATE(id_estrutura_sistema) OR
UPDATE(id_modulo)
BEGIN
SELECT @nullcnt = 0
SELECT @validcnt = count(*)
FROM inserted,tb_estrutura_sistema
WHERE
/* %JoinFKPK(inserted,tb_estrutura_sistema) */
inserted.id_estrutura_sistema = tb_estrutura_sistema.id_estrutura_sistema and
inserted.id_modulo = tb_estrutura_sistema.id_modulo
/* %NotNullFK(inserted," IS NULL","select @nullcnt = count(*) from inserted where"," AND") */

IF @validcnt + @nullcnt != @NUMROWS
BEGIN
SELECT @errno = 30007,
       @errmsg = 'Cannot update tb_usuario_estrutura_sistema because tb_estrutura_sistema does not exist.'
GOTO ERROR
END
END

/* ERwin Builtin Trigger */
RETURN
ERROR:
raiserror @errno @errmsg
rollback transaction
END

```

go