

CUIEEEPrintFile2

CREATE TABLE APPLIANCE

```
(
    Appliance_Model_Number varchar(150) NULL ,
    Appliance_Brand        varchar(20)  NULL ,
    Appliance_Type         varchar(150)  NULL ,
    Appliance_Price        float  NULL ,
    Appliance_Attributes   varchar(200)  NULL ,
    Appliance_ID           integer NOT NULL ,
    Vendor_ID              integer NOT NULL
)
go
```

ALTER TABLE APPLIANCE

```
    ADD CONSTRAINT XPKAPPLIANCE PRIMARY KEY CLUSTERED (Appliance_ID ASC)
go
```

CREATE TABLE ASSOCIATE

```
(
    Associate_ID           integer NOT NULL ,
    Associate_First_Name   varchar(50)  NULL ,
    Associate_Middle_Name  varchar(50)  NULL ,
    Associate_Last_Name    varchar(50)  NULL ,
    Associate_SSN          integer  NULL ,

```

```

                                CUIEEPrintFile2
Showroom_ID1          integer NOT NULL ,
Showroom_ID           char(18) NOT NULL

```

```
)
```

```
go
```

```
ALTER TABLE ASSOCIATE
```

```

    ADD CONSTRAINT XPKASSOCIATE PRIMARY KEY CLUSTERED (Associate_ID
ASC,Showroom_ID1 ASC,Showroom_ID ASC)

```

```
go
```

```
CREATE TABLE COMMISIION
```

```
(
```

```

    Salse_Commission_Amount float NULL ,
    Order_ID                integer NOT NULL ,
    Commission_ID           integer NOT NULL ,
    Customer_ID1            integer NOT NULL ,
    Customer_ID             char(18) NOT NULL

```

```
)
```

```
go
```

```
ALTER TABLE COMMISIION
```

```

    ADD CONSTRAINT XPKCOMMISIION PRIMARY KEY CLUSTERED (Commission_ID
ASC,Order_ID ASC,Customer_ID1 ASC,Customer_ID ASC)

```

```
go
```

CREATE TABLE CUSTOMER

```
(
    Customer_ID          integer  NOT NULL ,
    Customer_First_Name  varchar(100)  NULL ,
    Customer_Middle_Name varchar(100)  NULL ,
    Customer_Last_Name   varchar(100)  NULL ,
    Customer_Street_Address varchar(100)  NULL ,
    Customer_City         varchar(100)  NULL ,
    Customer_State        varchar(20)   NULL ,
    Customer_ZipCode      varchar(20)   NULL ,
    Customer_Contact_No   integer  NULL
)
go
```

ALTER TABLE CUSTOMER

```
ADD CONSTRAINT XPKCUSTOMER PRIMARY KEY CLUSTERED (Customer_ID ASC)
go
```

CREATE TABLE DELIVERY

```
(
    Delivery_ID          integer  NOT NULL ,
    Delivery_Date_Time    datetime  NULL ,
    Installation_Required varchar(20)  NULL ,
    Order_ID1            integer  NOT NULL ,
```

```

                                CUIEEPrintFile2
Customer_ID1          integer NOT NULL ,
Order_ID              char(18) NOT NULL ,
Customer_ID11         char(18) NOT NULL
)
go

```

ALTER TABLE DELIVERY

```

ADD CONSTRAINT XPKDELIVERY PRIMARY KEY CLUSTERED (Delivery_ID ASC,Order_ID1
ASC,Customer_ID1 ASC,Order_ID ASC,Customer_ID11 ASC)
go

```

CREATE TABLE INSTALLATION

```

(
    Delivery_ID1          integer NOT NULL ,
    Warehouse_ID          integer NOT NULL ,
    Technician_ID1        integer NOT NULL ,
    Installation_Date      datetime NULL ,
    Delivery_and_Installation_Charges integer NULL ,
    Order_ID1             integer NOT NULL ,
    Customer_ID1          integer NOT NULL ,
    Delivery_ID           char(18) NOT NULL ,
    Technician_ID         char(18) NOT NULL ,
    Order_ID              char(18) NOT NULL ,
    Customer_ID11         char(18) NOT NULL
)

```

CUIEEPrintFile2

go

ALTER TABLE INSTALLATION

ADD CONSTRAINT XPKINSTALLATION PRIMARY KEY CLUSTERED (Delivery_ID1
ASC, Technician_ID1 ASC, Delivery_ID ASC, Technician_ID ASC, Customer_ID1 ASC, Order_ID1
ASC, Order_ID ASC, Customer_ID11 ASC)

go

CREATE TABLE INVENTORY

(
Showroom_ID integer NOT NULL ,
Warehouse_ID1 integer NOT NULL ,
Appliance_ID1 integer NOT NULL ,
Showroom_Availibility varchar(20) NULL ,
Warehouse_ID char(18) NOT NULL ,
Appliance_ID char(18) NOT NULL
)

go

ALTER TABLE INVENTORY

ADD CONSTRAINT XPKINVENTORY PRIMARY KEY CLUSTERED (Warehouse_ID1
ASC, Appliance_ID1 ASC, Warehouse_ID ASC, Appliance_ID ASC)

go

CREATE TABLE ORDER

CUIEEEPrintFile2

```
(
    Order_ID          integer NOT NULL ,
    Salse_Price       float  NULL ,
    Total_Order_Quantity integer NULL ,
    Order_Date        datetime NULL ,
    Order_Time        datetime NULL ,
    Customer_ID1      integer NOT NULL ,
    Associate_ID       integer NULL ,
    Showroom_ID1      integer NULL ,
    Customer_ID        char(18) NOT NULL ,
    Showroom_ID        char(18) NULL
)
go
```

ALTER TABLE ORDER

ADD CONSTRAINT XPKORDER PRIMARY KEY CLUSTERED (Order_ID ASC,Customer_ID1
ASC,Customer_ID ASC)

go

CREATE TABLE ORDER_LINE

```
(
    Order_ID1          integer NOT NULL ,
    Appliance_ID1      integer NOT NULL ,
    Quantity           integer NULL ,
    Showroom_Pickup    varchar(20) NULL ,
```

CUIEEPrintFile2

```
Customer_ID1      integer  NOT NULL ,
Order_ID          char(18)  NOT NULL ,
Appliance_ID      char(18)  NOT NULL ,
Customer_ID       char(18)  NOT NULL
)
go
```

ALTER TABLE ORDER_LINE

```
ADD CONSTRAINT XPKORDER_LINE PRIMARY KEY CLUSTERED (Order_ID1
ASC,Appliance_ID1 ASC,Order_ID ASC,Appliance_ID ASC,Customer_ID1 ASC,Customer_ID
ASC)
go
```

CREATE TABLE SHIPMENT

```
(
    Shiment_ID      integer  NOT NULL ,
    Vendor_ID1      integer  NOT NULL ,
    Warehouse_ID1    integer  NOT NULL ,
    Shipper_Details  varchar(20) NULL ,
    Shipment_Origin  varchar(20) NULL ,
    Shipment_Destination varchar(20) NULL ,
    Vendor_ID        char(18)  NOT NULL ,
    Warehouse_ID     char(18)  NOT NULL
)
go
```

CUIEEPrintFile2

ALTER TABLE SHIPMENT

ADD CONSTRAINT XPKSHIPMENT PRIMARY KEY CLUSTERED (Shiment_ID ASC, Vendor_ID1
ASC, Warehouse_ID1 ASC, Vendor_ID ASC, Warehouse_ID ASC)

go

CREATE TABLE SHOWROOM

(

 Showroom_ID integer NOT NULL ,
 Showroom_Street varchar(20) NULL ,
 Showroom_City varchar(20) NULL ,
 Showroom_State varchar(20) NULL ,
 Showroom_Zip_Code varchar(20) NULL

)

go

ALTER TABLE SHOWROOM

ADD CONSTRAINT XPKSHOWROOM PRIMARY KEY CLUSTERED (Showroom_ID ASC)

go

CREATE TABLE TECHNICIAN

(

 Technician_ID integer NOT NULL ,
 Supervisor_ID integer NULL ,


```

                                CUIEEPrintFile2
Technician_First_Name varchar(150)  NULL ,
Technician_Middle_Name varchar(150)  NULL ,
Technician_Last_Name varchar(150)  NULL ,
Technican_SSN          integer  NOT NULL ,
Warehouse_ID           integer  NOT NULL ,
Technician_Status      varchar(20)  NULL
)
go

ALTER TABLE TECHNICIAN

        ADD CONSTRAINT XPKTECHNICIAN PRIMARY KEY  CLUSTERED (Technician_ID
ASC,Warehouse_ID ASC)
go

CREATE TABLE VENDOR
(
        Vendor_ID          integer  NOT NULL ,
        Vendor_Name        varchar(150)  NULL ,
        Vendor_Attributes  varchar(200)  NULL
)
go

ALTER TABLE VENDOR

        ADD CONSTRAINT XPKVENDOR PRIMARY KEY  CLUSTERED (Vendor_ID ASC)
go

```

CREATE TABLE WAREHOUSE

```
(  
    Warehouse_ID      integer NOT NULL ,  
    Warehouse_Street  varchar(150) NULL ,  
    Warehouse_City     varchar(20)  NULL ,  
    Warehouse_State    varchar(20)  NULL ,  
    Warehouse_Zip_Code varchar(20)  NULL  
)
```

)

go

ALTER TABLE WAREHOUSE

ADD CONSTRAINT XPKWAREHOUSE PRIMARY KEY CLUSTERED (Warehouse_ID ASC)

go

ALTER TABLE APPLIANCE

ADD CONSTRAINT R_29 FOREIGN KEY (Vendor_ID) REFERENCES VENDOR(Vendor_ID)

ON DELETE NO ACTION

ON UPDATE NO ACTION

go

ALTER TABLE ASSOCIATE

CUIEEPrintFile2

```
ADD CONSTRAINT R_2 FOREIGN KEY (Showroom_ID1) REFERENCES  
SHOWROOM(Showroom_ID)
```

```
ON DELETE NO ACTION
```

```
ON UPDATE NO ACTION
```

```
go
```

```
ALTER TABLE COMMISIION
```

```
ADD CONSTRAINT R_7 FOREIGN KEY (Order_ID, Customer_ID1, Customer_ID)  
REFERENCES ORDER(Order_ID, Customer_ID1, Customer_ID)
```

```
ON DELETE NO ACTION
```

```
ON UPDATE NO ACTION
```

```
go
```

```
ALTER TABLE DELIVERY
```

```
ADD CONSTRAINT R_11 FOREIGN KEY (Order_ID1, Customer_ID1, Customer_ID11)  
REFERENCES ORDER(Order_ID, Customer_ID1, Customer_ID)
```

```
ON DELETE NO ACTION
```

```
ON UPDATE NO ACTION
```

```
go
```

```
ALTER TABLE INSTALLATION
```

```
ADD CONSTRAINT R_46 FOREIGN KEY  
(Delivery_ID1, Order_ID1, Customer_ID1, Order_ID, Customer_ID11) REFERENCES
```

```
CUIEEPrintFile2
DELIVERY(Delivery_ID,Order_ID1,Customer_ID1,Order_ID,Customer_ID11)
```

```
ON DELETE NO ACTION
```

```
ON UPDATE NO ACTION
```

```
go
```

```
ALTER TABLE INSTALLATION
```

```
ADD CONSTRAINT R_47 FOREIGN KEY (Technician_ID1,Warehouse_ID) REFERENCES
TECHNICIAN(Technician_ID,Warehouse_ID)
```

```
ON DELETE NO ACTION
```

```
ON UPDATE NO ACTION
```

```
go
```

```
ALTER TABLE INVENTORY
```

```
ADD CONSTRAINT R_55 FOREIGN KEY (Warehouse_ID1) REFERENCES
WAREHOUSE(Warehouse_ID)
```

```
ON DELETE NO ACTION
```

```
ON UPDATE NO ACTION
```

```
go
```

```
ALTER TABLE INVENTORY
```

```
ADD CONSTRAINT R_56 FOREIGN KEY (Appliance_ID1) REFERENCES
APPLIANCE(Appliance_ID)
```

```
ON DELETE NO ACTION
```

```
ON UPDATE NO ACTION
```

```
go
```

ALTER TABLE INVENTORY

ADD CONSTRAINT R_51 FOREIGN KEY (Showroom_ID) REFERENCES
SHOWROOM(Showroom_ID)

ON DELETE NO ACTION

ON UPDATE NO ACTION

go

ALTER TABLE INVENTORY

ADD CONSTRAINT R_57 FOREIGN KEY (Showroom_ID) REFERENCES
SHOWROOM(Showroom_ID)

ON DELETE NO ACTION

ON UPDATE NO ACTION

go

ALTER TABLE ORDER

ADD CONSTRAINT R_4 FOREIGN KEY (Customer_ID1) REFERENCES
CUSTOMER(Customer_ID)

ON DELETE NO ACTION

ON UPDATE NO ACTION

go

ALTER TABLE ORDER

ADD CONSTRAINT R_5 FOREIGN KEY (Associate_ID,Showroom_ID1,Showroom_ID)
REFERENCES ASSOCIATE(Associate_ID,Showroom_ID1,Showroom_ID)

CUIEEPrintFile2

ON DELETE NO ACTION

ON UPDATE NO ACTION

go

ALTER TABLE ORDER_LINE

ADD CONSTRAINT R_42 FOREIGN KEY (Order_ID1, Customer_ID1, Customer_ID)
REFERENCES ORDER(Order_ID, Customer_ID1, Customer_ID)

ON DELETE NO ACTION

ON UPDATE NO ACTION

go

ALTER TABLE ORDER_LINE

ADD CONSTRAINT R_43 FOREIGN KEY (Appliance_ID1) REFERENCES
APPLIANCE(Appliance_ID)

ON DELETE NO ACTION

ON UPDATE NO ACTION

go

ALTER TABLE SHIPMENT

ADD CONSTRAINT R_23 FOREIGN KEY (Vendor_ID1) REFERENCES VENDOR(Vendor_ID)

ON DELETE NO ACTION

ON UPDATE NO ACTION

go

ALTER TABLE SHIPMENT

ADD CONSTRAINT R_24 FOREIGN KEY (Warehouse_ID1) REFERENCES
WAREHOUSE(Warehouse_ID)

ON DELETE NO ACTION

ON UPDATE NO ACTION

go

ALTER TABLE TECHNICIAN

ADD CONSTRAINT R_1 FOREIGN KEY (Supervisor_ID,Warehouse_ID) REFERENCES
TECHNICIAN(Technician_ID,Warehouse_ID)

ON DELETE NO ACTION

ON UPDATE NO ACTION

go

ALTER TABLE TECHNICIAN

ADD CONSTRAINT R_12 FOREIGN KEY (Warehouse_ID) REFERENCES
WAREHOUSE(Warehouse_ID)

ON DELETE NO ACTION

ON UPDATE NO ACTION

go

CREATE TRIGGER tD_APPLIANCE ON APPLIANCE FOR DELETE AS

```

/* ERwin Builtin Trigger */

/* DELETE trigger on APPLIANCE */

BEGIN

    DECLARE @errno    int,

            @severity int,

            @state     int,

            @errmsg    varchar(255)

/* ERwin Builtin Trigger */

/* APPLIANCE INVENTORY on parent delete no action */

/* ERWIN_RELATION:CHECKSUM="00031bc0", PARENT_OWNER="", PARENT_TABLE="APPLIANCE"
CHILD_OWNER="", CHILD_TABLE="INVENTORY"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_56", FK_COLUMNS="Appliance_ID1" */

IF EXISTS (

    SELECT * FROM deleted,INVENTORY

    WHERE

        /* %JoinFKPK(INVENTORY,deleted," = "," AND") */

        INVENTORY.Appliance_ID1 = deleted.Appliance_ID

)

BEGIN

    SELECT @errno = 30001,

           @errmsg = 'Cannot delete APPLIANCE because INVENTORY exists.'

    GOTO error

END

```


CUIEEEPrintFile2

```
/* ERwin Builtin Trigger */

/* APPLIANCE ORDER_LINE on parent delete no action */

/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="APPLIANCE"
CHILD_OWNER="", CHILD_TABLE="ORDER_LINE"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_43", FK_COLUMNS="Appliance_ID1" */
IF EXISTS (
    SELECT * FROM deleted,ORDER_LINE
    WHERE
        /* %JoinFKPK(ORDER_LINE,deleted," = "," AND") */
        ORDER_LINE.Appliance_ID1 = deleted.Appliance_ID
)
BEGIN
    SELECT @errno = 30001,
           @errmsg = 'Cannot delete APPLIANCE because ORDER_LINE exists.'
    GOTO error
END

/* ERwin Builtin Trigger */

/* VENDOR APPLIANCE on child delete no action */

/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="VENDOR"
CHILD_OWNER="", CHILD_TABLE="APPLIANCE"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
```

```

CUIEEEPrintFile2
FK_CONSTRAINT="R_29", FK_COLUMNS="Vendor_ID" */
IF EXISTS (SELECT * FROM deleted,VENDOR
WHERE
/* %JoinFKPK(deleted,VENDOR," = "," AND") */
deleted.Vendor_ID = VENDOR.Vendor_ID AND
NOT EXISTS (
SELECT * FROM APPLIANCE
WHERE
/* %JoinFKPK(APPLIANCE,VENDOR," = "," AND") */
APPLIANCE.Vendor_ID = VENDOR.Vendor_ID
)
)
BEGIN
SELECT @errno = 30010,
@errmsg = 'Cannot delete last APPLIANCE because VENDOR exists.'
GOTO error
END

```

```

/* ERwin Builtin Trigger */

```

```

RETURN

```

error:

```

RAISERROR (@errmsg, -- Message text.
@severity, -- Severity (0~25).

```

```

                                CUIEEPrintFile2
                                @state) -- State (0~255).

rollback transaction

END

go


CREATE TRIGGER tU_APPLIANCE ON APPLIANCE FOR UPDATE AS

/* ERwin Builtin Trigger */

/* UPDATE trigger on APPLIANCE */

BEGIN

    DECLARE @numrows int,

            @nullcnt int,

            @validcnt int,

            @insAppliance_ID integer,

            @errno int,

            @severity int,

            @state int,

            @errmsg varchar(255)


    SELECT @numrows = @@rowcount

    /* ERwin Builtin Trigger */

    /* APPLIANCE INVENTORY on parent update no action */

    /* ERWIN_RELATION:CHECKSUM="00037da9", PARENT_OWNER="", PARENT_TABLE="APPLIANCE"

```

```

                                CUIEEPrintFile2
CHILD_OWNER="", CHILD_TABLE="INVENTORY"

P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",

FK_CONSTRAINT="R_56", FK_COLUMNS="Appliance_ID1" */

IF

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

UPDATE(Appliance_ID)

BEGIN

IF EXISTS (

    SELECT * FROM deleted,INVENTORY

    WHERE

        /* %JoinFKPK(INVENTORY,deleted," = "," AND") */

        INVENTORY.Appliance_ID1 = deleted.Appliance_ID

)

BEGIN

    SELECT @errno  = 30005,

           @errmsg = 'Cannot update APPLIANCE because INVENTORY exists.'

    GOTO error

END

END

/* ERwin Builtin Trigger */

/* APPLIANCE ORDER_LINE on parent update no action */

/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="APPLIANCE"

CHILD_OWNER="", CHILD_TABLE="ORDER_LINE"

```

```

                                CUIEEPrintFile2
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",

FK_CONSTRAINT="R_43", FK_COLUMNS="Appliance_ID1" */

IF

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

UPDATE(Appliance_ID)

BEGIN

IF EXISTS (

SELECT * FROM deleted,ORDER_LINE

WHERE

/* %JoinFKPK(ORDER_LINE,deleted," = "," AND") */

ORDER_LINE.Appliance_ID1 = deleted.Appliance_ID

)

BEGIN

SELECT @errno = 30005,

       @errmsg = 'Cannot update APPLIANCE because ORDER_LINE exists.'

GOTO error

END

END

/* ERwin Builtin Trigger */

/* VENDOR APPLIANCE on child update no action */

/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="VENDOR"

CHILD_OWNER="", CHILD_TABLE="APPLIANCE"

P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",

```

```

                                CUIEEPrintFile2
FK_CONSTRAINT="R_29", FK_COLUMNS="Vendor_ID" */

IF

    /* %ChildFK(" OR",UPDATE) */

    UPDATE(Vendor_ID)

BEGIN

    SELECT @nullcnt = 0

    SELECT @validcnt = count(*)

        FROM inserted,VENDOR

        WHERE

            /* %JoinFKPK(inserted,VENDOR) */

            inserted.Vendor_ID = VENDOR.Vendor_ID

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

    IF @validcnt + @nullcnt != @numrows

    BEGIN

        SELECT @errno  = 30007,

            @errmsg = 'Cannot update APPLIANCE because VENDOR does not exist.'

        GOTO error

    END

END

END

/* ERwin Builtin Trigger */

RETURN

```

CUIEEEPrintFile2

error:

```
RAISERROR (@errmsg, -- Message text.  
           @severity, -- Severity (0~25).  
           @state) -- State (0~255).
```

rollback transaction

END

go

CREATE TRIGGER td_ASSOCIATE ON ASSOCIATE FOR DELETE AS

/* ERwin Builtin Trigger */

/* DELETE trigger on ASSOCIATE */

BEGIN

```
DECLARE @errno int,  
        @severity int,  
        @state int,  
        @errmsg varchar(255)
```

/* ERwin Builtin Trigger */

/* ASSOCIATE ORDER on parent delete no action */

/* ERWIN_RELATION:CHECKSUM="00028ce1", PARENT_OWNER="", PARENT_TABLE="ASSOCIATE"

CHILD_OWNER="", CHILD_TABLE="ORDER"

CUIEEPrintFile2

```
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_5", FK_COLUMNS="Associate_ID""Showroom_ID1""Showroom_ID" */
IF EXISTS (
    SELECT * FROM deleted,ORDER
    WHERE
        /* %JoinFKPK(ORDER,deleted," = "," AND") */
        ORDER.Associate_ID = deleted.Associate_ID AND
        ORDER.Showroom_ID1 = deleted.Showroom_ID1 AND
        ORDER.Showroom_ID = deleted.Showroom_ID
)
BEGIN
    SELECT @errno = 30001,
           @errmsg = 'Cannot delete ASSOCIATE because ORDER exists.'
    GOTO error
END

/* ERwin Builtin Trigger */
/* SHOWROOM ASSOCIATE on child delete no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="SHOWROOM"
CHILD_OWNER="", CHILD_TABLE="ASSOCIATE"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_2", FK_COLUMNS="Showroom_ID1" */
IF EXISTS (SELECT * FROM deleted,SHOWROOM
    WHERE
```


CUIEEPrintFile2

```
/* %JoinFKPK(deleted,SHOWROOM," = "," AND") */
deleted.Showroom_ID1 = SHOWROOM.Showroom_ID AND
NOT EXISTS (
    SELECT * FROM ASSOCIATE
    WHERE
        /* %JoinFKPK(ASSOCIATE,SHOWROOM," = "," AND") */
        ASSOCIATE.Showroom_ID1 = SHOWROOM.Showroom_ID
)
)
BEGIN
    SELECT @errno = 30010,
           @errmsg = 'Cannot delete last ASSOCIATE because SHOWROOM exists.'
    GOTO error
END

/* ERwin Builtin Trigger */
RETURN
error:
RAISERROR (@errmsg, -- Message text.
           @severity, -- Severity (0~25).
           @state) -- State (0~255).
rollback transaction
END
```

CUIEEPrintFile2

go

```
CREATE TRIGGER tu_ASSOCIATE ON ASSOCIATE FOR UPDATE AS
```

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

```
/* UPDATE trigger on ASSOCIATE */
```

```
BEGIN
```

```
    DECLARE @numrows int,  
            @nullcnt int,  
            @validcnt int,  
            @insAssociate_ID integer,  
            @insShowroom_ID1 integer,  
            @insShowroom_ID char(18),  
            @errno int,  
            @severity int,  
            @state int,  
            @errmsg varchar(255)
```

```
SELECT @numrows = @@rowcount
```

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

```
/* ASSOCIATE ORDER on parent update no action */
```

```
/* ERWIN_RELATION:CHECKSUM="0002c5a1", PARENT_OWNER="", PARENT_TABLE="ASSOCIATE"  
   CHILD_OWNER="", CHILD_TABLE="ORDER"
```

CUIEEPrintFile2

```
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_5", FK_COLUMNS="Associate_ID""Showroom_ID1""Showroom_ID" */
IF
/* %ParentPK(" OR",UPDATE) */
UPDATE(Associate_ID) OR
UPDATE>Showroom_ID1) OR
UPDATE>Showroom_ID)
BEGIN
IF EXISTS (
SELECT * FROM deleted,ORDER
WHERE
/* %JoinFKPK(ORDER,deleted," = "," AND") */
ORDER.Associate_ID = deleted.Associate_ID AND
ORDER.Showroom_ID1 = deleted.Showroom_ID1 AND
ORDER.Showroom_ID = deleted.Showroom_ID
)
BEGIN
SELECT @errno = 30005,
        @errmsg = 'Cannot update ASSOCIATE because ORDER exists.'
GOTO error
END
END

/* ERwin Builtin Trigger */
```

CUIEEPrintFile2

```
/* SHOWROOM ASSOCIATE on child update no action */

/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="SHOWROOM"
   CHILD_OWNER="", CHILD_TABLE="ASSOCIATE"
   P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
   FK_CONSTRAINT="R_2", FK_COLUMNS="Showroom_ID1" */

IF

/* %ChildFK(" OR",UPDATE) */

UPDATE(Showroom_ID1)

BEGIN

SELECT @nullcnt = 0

SELECT @validcnt = count(*)

FROM inserted,SHOWROOM

WHERE

/* %JoinFKPK(inserted,SHOWROOM) */

inserted.Showroom_ID1 = SHOWROOM.Showroom_ID

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

IF @validcnt + @nullcnt != @numrows

BEGIN

SELECT @errno = 30007,

       @errmsg = 'Cannot update ASSOCIATE because SHOWROOM does not exist.'

GOTO error

END
```

CUIEEPrintFile2

END

/* ERwin Builtin Trigger */

RETURN

error:

RAISERROR (@errmsg, -- Message text.

 @severity, -- Severity (0~25).

 @state) -- State (0~255).

rollback transaction

END

go

CREATE TRIGGER tD_COMMISIION ON COMMISIION FOR DELETE AS

/* ERwin Builtin Trigger */

/* DELETE trigger on COMMISIION */

BEGIN

DECLARE @errno int,

 @severity int,

 @state int,

```

CUIEEEPrintFile2
    @errmsg varchar(255)

/* ERwin Builtin Trigger */

/* ORDER  COMMISIION on child delete no action */

/* ERWIN_RELATION:CHECKSUM="0001b02f", PARENT_OWNER="", PARENT_TABLE="ORDER"
CHILD_OWNER="", CHILD_TABLE="COMMISIION"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_7", FK_COLUMNS="Customer_ID""Order_ID""Customer_ID1" */
IF EXISTS (SELECT * FROM deleted,ORDER
WHERE
    /* %JoinFKPK(deleted,ORDER," = "," AND") */
    deleted.Order_ID = ORDER.Order_ID AND
    deleted.Customer_ID1 = ORDER.Customer_ID1 AND
    deleted.Customer_ID = ORDER.Customer_ID AND
    NOT EXISTS (
        SELECT * FROM COMMISIION
        WHERE
            /* %JoinFKPK(COMMISIION,ORDER," = "," AND") */
            COMMISIION.Order_ID = ORDER.Order_ID AND
            COMMISIION.Customer_ID1 = ORDER.Customer_ID1 AND
            COMMISIION.Customer_ID = ORDER.Customer_ID
    )
)
)
BEGIN
    SELECT @errno = 30010,

```

```

                                CUIEEPrintFile2
                                @errmsg = 'Cannot delete last COMMISIION because ORDER exists.'

                                GOTO error

                                END

                                /* ERwin Builtin Trigger */

                                RETURN

error:

                                RAISERROR (@errmsg, -- Message text.
                                            @severity, -- Severity (0~25).
                                            @state) -- State (0~255).

                                rollback transaction

                                END

                                go

CREATE TRIGGER tU_COMMISIION ON COMMISIION FOR UPDATE AS

/* ERwin Builtin Trigger */

/* UPDATE trigger on COMMISIION */

BEGIN

    DECLARE @numrows int,
            @nullcnt int,
            @validcnt int,

```

```

                                CUIEEPrintFile2
    @insOrder_ID integer,

    @insCommission_ID integer,

    @insCustomer_ID1 integer,

    @insCustomer_ID char(18),

    @errno    int,

    @severity int,

    @state    int,

    @errmsg   varchar(255)


SELECT @numrows = @@rowcount

/* ERwin Builtin Trigger */

/* ORDER  COMMISIION on child update no action */

/* ERWIN_RELATION:CHECKSUM="0001ac6b", PARENT_OWNER="", PARENT_TABLE="ORDER"
   CHILD_OWNER="", CHILD_TABLE="COMMISIION"
   P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
   FK_CONSTRAINT="R_7", FK_COLUMNS="Customer_ID""Order_ID""Customer_ID1" */
IF

    /* %ChildFK(" OR",UPDATE) */

    UPDATE(Order_ID) OR

    UPDATE(Customer_ID1) OR

    UPDATE(Customer_ID)

BEGIN

    SELECT @nullcnt = 0

    SELECT @validcnt = count(*)

```


CUIEEPrintFile2

FROM inserted,ORDER

WHERE

/* %JoinFKPK(inserted,ORDER) */

inserted.Order_ID = ORDER.Order_ID and

inserted.Customer_ID1 = ORDER.Customer_ID1 and

inserted.Customer_ID = ORDER.Customer_ID

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

IF @validcnt + @nullcnt != @numrows

BEGIN

SELECT @errno = 30007,

@errmsg = 'Cannot update COMMISIIION because ORDER does not exist.'

GOTO error

END

END

/* ERwin Builtin Trigger */

RETURN

error:

RAISERROR (@errmsg, -- Message text.

@severity, -- Severity (0~25).

@state) -- State (0~255).

rollback transaction

CUIEEPrintFile2

END

go

```
CREATE TRIGGER td_CUSTOMER ON CUSTOMER FOR DELETE AS

/* ERwin Builtin Trigger */

/* DELETE trigger on CUSTOMER */

BEGIN

    DECLARE @errno    int,

            @severity int,

            @state     int,

            @errmsg    varchar(255)

    /* ERwin Builtin Trigger */

    /* CUSTOMER  ORDER on parent delete no action */

    /* ERWIN_RELATION:CHECKSUM="00010d67", PARENT_OWNER="", PARENT_TABLE="CUSTOMER"
    CHILD_OWNER="", CHILD_TABLE="ORDER"
    P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
    FK_CONSTRAINT="R_4", FK_COLUMNS="Customer_ID1" */

    IF EXISTS (

        SELECT * FROM deleted,ORDER

        WHERE
```

CUIEEPrintFile2

```
/* %JoinFKPK(ORDER,deleted," = "," AND") */  
ORDER.Customer_ID1 = deleted.Customer_ID  
)  
BEGIN  
    SELECT @errno = 30001,  
           @errmsg = 'Cannot delete CUSTOMER because ORDER exists.'  
    GOTO error  
END
```

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

```
RETURN
```

error:

```
RAISERROR (@errmsg, -- Message text.  
           @severity, -- Severity (0~25).  
           @state) -- State (0~255).  
rollback transaction
```

```
END
```

```
go
```

```
CREATE TRIGGER tU_CUSTOMER ON CUSTOMER FOR UPDATE AS
```

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

CUIEEEPrintFile2

```
/* UPDATE trigger on CUSTOMER */

BEGIN

    DECLARE @numrows int,

            @nullcnt int,

            @validcnt int,

            @insCustomer_ID integer,

            @errno int,

            @severity int,

            @state int,

            @errmsg varchar(255)


    SELECT @numrows = @@rowcount

    /* ERwin Builtin Trigger */

    /* CUSTOMER ORDER on parent update no action */

    /* ERWIN_RELATION:CHECKSUM="00012e5c", PARENT_OWNER="", PARENT_TABLE="CUSTOMER"

    CHILD_OWNER="", CHILD_TABLE="ORDER"

    P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",

    FK_CONSTRAINT="R_4", FK_COLUMNS="Customer_ID1" */

    IF

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

        UPDATE(Customer_ID)

    BEGIN

        IF EXISTS (

            SELECT * FROM deleted,ORDER
```

CUIEEPrintFile2

WHERE

/* %JoinFKPK(ORDER,deleted," = "," AND") */

ORDER.Customer_ID1 = deleted.Customer_ID

)

BEGIN

SELECT @errno = 30005,

@errmsg = 'Cannot update CUSTOMER because ORDER exists.'

GOTO error

END

END

/* ERwin Builtin Trigger */

RETURN

error:

RAISERROR (@errmsg, -- Message text.

@severity, -- Severity (0~25).

@state) -- State (0~255).

rollback transaction

END

go

CUIEEEPrintFile2

```
CREATE TRIGGER tD_DELIVERY ON DELIVERY FOR DELETE AS

/* ERwin Builtin Trigger */

/* DELETE trigger on DELIVERY */

BEGIN

    DECLARE @errno    int,

            @severity int,

            @state     int,

            @errmsg    varchar(255)

    /* ERwin Builtin Trigger */

    /* DELIVERY  INSTALLATION on parent delete no action */

    /* ERWIN_RELATION:CHECKSUM="000305ca", PARENT_OWNER="", PARENT_TABLE="DELIVERY"

    CHILD_OWNER="", CHILD_TABLE="INSTALLATION"

    P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",

    FK_CONSTRAINT="R_46",

    FK_COLUMNS="Customer_ID11" "Delivery_ID1" "Order_ID1" "Customer_ID1" "Order_ID" */

    IF EXISTS (

        SELECT * FROM deleted,INSTALLATION

        WHERE

            /* %JoinFKPK(INSTALLATION,deleted," = "," AND") */

            INSTALLATION.Delivery_ID1 = deleted.Delivery_ID AND

            INSTALLATION.Order_ID1 = deleted.Order_ID1 AND

            INSTALLATION.Customer_ID1 = deleted.Customer_ID1 AND
```

```

                                CUIEEPrintFile2
INSTALLATION.Order_ID = deleted.Order_ID AND

INSTALLATION.Customer_ID11 = deleted.Customer_ID11

)

BEGIN

    SELECT @errno  = 30001,

           @errmsg = 'Cannot delete DELIVERY because INSTALLATION exists.'

    GOTO error

END

/* ERwin Builtin Trigger */

/* ORDER  DELIVERY on child delete no action */

/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="ORDER"
CHILD_OWNER="", CHILD_TABLE="DELIVERY"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_11", FK_COLUMNS="Customer_ID11""Order_ID1""Customer_ID1" */
IF EXISTS (SELECT * FROM deleted,ORDER

WHERE

    /* %JoinFKPK(deleted,ORDER," = "," AND") */

    deleted.Order_ID1 = ORDER.Order_ID AND

    deleted.Customer_ID1 = ORDER.Customer_ID1 AND

    deleted.Customer_ID11 = ORDER.Customer_ID AND

    NOT EXISTS (

        SELECT * FROM DELIVERY

        WHERE

```

```

                                CUIEEPrintFile2
/* %JoinFKPK(DELIVERY,ORDER," = "," AND") */

DELIVERY.Order_ID1 = ORDER.Order_ID AND

DELIVERY.Customer_ID1 = ORDER.Customer_ID1 AND

DELIVERY.Customer_ID11 = ORDER.Customer_ID

    )
)
BEGIN

    SELECT @errno  = 30010,

           @errmsg = 'Cannot delete last DELIVERY because ORDER exists.'

    GOTO error

END

/* ERwin Builtin Trigger */

RETURN

error:

RAISERROR (@errmsg, -- Message text.

           @severity, -- Severity (0~25).

           @state) -- State (0~255).

rollback transaction

END

go

```


CUIEEPrintFile2

```
CREATE TRIGGER tU_DELIVERY ON DELIVERY FOR UPDATE AS

/* ERwin Builtin Trigger */

/* UPDATE trigger on DELIVERY */

BEGIN

    DECLARE @numrows int,

            @nullcnt int,

            @validcnt int,

            @insDelivery_ID integer,

            @insOrder_ID1 integer,

            @insCustomer_ID1 integer,

            @insOrder_ID char(18),

            @insCustomer_ID11 char(18),

            @errno int,

            @severity int,

            @state int,

            @errmsg varchar(255)


    SELECT @numrows = @@rowcount

/* ERwin Builtin Trigger */

/* DELIVERY INSTALLATION on parent update no action */

/* ERWIN_RELATION:CHECKSUM="00036c1a", PARENT_OWNER="", PARENT_TABLE="DELIVERY"

   CHILD_OWNER="", CHILD_TABLE="INSTALLATION"

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

CUIEEPrintFile2

```
FK_CONSTRAINT="R_46",  
FK_COLUMNS="Customer_ID11""Delivery_ID1""Order_ID1""Customer_ID1""Order_ID" */
```

```
IF
```

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

```
UPDATE(Delivery_ID) OR
```

```
UPDATE(Order_ID1) OR
```

```
UPDATE(Customer_ID1) OR
```

```
UPDATE(Order_ID) OR
```

```
UPDATE(Customer_ID11)
```

```
BEGIN
```

```
IF EXISTS (
```

```
SELECT * FROM deleted,INSTALLATION
```

```
WHERE
```

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

```
INSTALLATION.Delivery_ID1 = deleted.Delivery_ID AND
```

```
INSTALLATION.Order_ID1 = deleted.Order_ID1 AND
```

```
INSTALLATION.Customer_ID1 = deleted.Customer_ID1 AND
```

```
INSTALLATION.Order_ID = deleted.Order_ID AND
```

```
INSTALLATION.Customer_ID11 = deleted.Customer_ID11
```

```
)
```

```
BEGIN
```

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

```
@errmsg = 'Cannot update DELIVERY because INSTALLATION exists.'
```

```
GOTO error
```

```
END
```

CUIEEPrintFile2

END

/* ERwin Builtin Trigger */

/* ORDER DELIVERY on child update no action */

/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="ORDER"

CHILD_OWNER="", CHILD_TABLE="DELIVERY"

P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",

FK_CONSTRAINT="R_11", FK_COLUMNS="Customer_ID11""Order_ID1""Customer_ID1" */

IF

/* %ChildFK(" OR",UPDATE) */

UPDATE(Order_ID1) OR

UPDATE(Customer_ID1) OR

UPDATE(Customer_ID11)

BEGIN

SELECT @nullcnt = 0

SELECT @validcnt = count(*)

FROM inserted,ORDER

WHERE

/* %JoinFKPK(inserted,ORDER) */

inserted.Order_ID1 = ORDER.Order_ID and

inserted.Customer_ID1 = ORDER.Customer_ID1 and

inserted.Customer_ID11 = ORDER.Customer_ID

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

CUIEEPrintFile2

```
IF @validcnt + @nullcnt != @numrows
```

```
BEGIN
```

```
    SELECT @errno  = 30007,
```

```
           @errmsg = 'Cannot update DELIVERY because ORDER does not exist.'
```

```
    GOTO error
```

```
END
```

```
END
```

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

```
RETURN
```

```
error:
```

```
    RAISERROR (@errmsg, -- Message text.
```

```
              @severity, -- Severity (0~25).
```

```
              @state) -- State (0~255).
```

```
    rollback transaction
```

```
END
```

```
go
```

```

                                CUIEEPrintFile2
CREATE TRIGGER tD_INSTALLATION ON INSTALLATION FOR DELETE AS

/* ERwin Builtin Trigger */

/* DELETE trigger on INSTALLATION */

BEGIN

    DECLARE @errno    int,

            @severity int,

            @state     int,

            @errmsg    varchar(255)

    /* ERwin Builtin Trigger */

    /* TECHNICIAN  INSTALLATION on child delete no action */

    /* ERWIN_RELATION:CHECKSUM="0003985a", PARENT_OWNER="",
PARENT_TABLE="TECHNICIAN"

    CHILD_OWNER="", CHILD_TABLE="INSTALLATION"

    P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",

    FK_CONSTRAINT="R_47", FK_COLUMNS="Technician_ID1""Warehouse_ID" */

    IF EXISTS (SELECT * FROM deleted,TECHNICIAN

        WHERE

            /* %JoinFKPK(deleted,TECHNICIAN," = "," AND") */

            deleted.Technician_ID1 =TECHNICIAN.Technician_ID AND

            deleted.Warehouse_ID = TECHNICIAN.Warehouse_ID AND

            NOT EXISTS (

                SELECT * FROM INSTALLATION

                WHERE

                    /* %JoinFKPK(INSTALLATION,TECHNICIAN," = "," AND") */

                    INSTALLATION.Technician_ID1 = TECHNICIAN.Technician_ID AND

```

CUIEEPrintFile2

```
        INSTALLATION.Warehouse_ID = TECHNICIAN.Warehouse_ID

    )

)

BEGIN

    SELECT @errno  = 30010,

           @errmsg = 'Cannot delete last INSTALLATION because TECHNICIAN exists.'

    GOTO error

END


/* ERwin Builtin Trigger */

/* DELIVERY  INSTALLATION on child delete no action */

/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="DELIVERY"
CHILD_OWNER="", CHILD_TABLE="INSTALLATION"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_46",
FK_COLUMNS="Customer_ID11""Delivery_ID1""Order_ID1""Customer_ID1""Order_ID" */

IF EXISTS (SELECT * FROM deleted,DELIVERY

WHERE

    /* %JoinFKPK(deleted,DELIVERY," = "," AND") */

    deleted.Delivery_ID1 = DELIVERY.Delivery_ID AND

    deleted.Order_ID1 = DELIVERY.Order_ID1 AND

    deleted.Customer_ID1 = DELIVERY.Customer_ID1 AND

    deleted.Order_ID = DELIVERY.Order_ID AND

    deleted.Customer_ID11 = DELIVERY.Customer_ID11 AND
```

CUIEEPrintFile2

NOT EXISTS (

SELECT * FROM INSTALLATION

WHERE

/* %JoinFKPK(INSTALLATION,DELIVERY," = "," AND") */

INSTALLATION.Delivery_ID1 = DELIVERY.Delivery_ID AND

INSTALLATION.Order_ID1 = DELIVERY.Order_ID1 AND

INSTALLATION.Customer_ID1 = DELIVERY.Customer_ID1 AND

INSTALLATION.Order_ID = DELIVERY.Order_ID AND

INSTALLATION.Customer_ID11 = DELIVERY.Customer_ID11

)

)

BEGIN

SELECT @errno = 30010,

@errmsg = 'Cannot delete last INSTALLATION because DELIVERY exists.'

GOTO error

END

/* ERwin Builtin Trigger */

RETURN

error:

RAISERROR (@errmsg, -- Message text.

@severity, -- Severity (0~25).

@state) -- State (0~255).

```
rollback transaction
```

```
END
```

```
go
```

```
CREATE TRIGGER tU_INSTALLATION ON INSTALLATION FOR UPDATE AS
```

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

```
/* UPDATE trigger on INSTALLATION */
```

```
BEGIN
```

```
    DECLARE  @numrows int,  
             @nullcnt int,  
             @validcnt int,  
             @insDelivery_ID1 integer,  
             @insTechnician_ID1 integer,  
             @insOrder_ID1 integer,  
             @insCustomer_ID1 integer,  
             @insDelivery_ID char(18),  
             @insTechnician_ID char(18),  
             @insOrder_ID char(18),  
             @insCustomer_ID11 char(18),  
             @errno  int,  
             @severity int,  
             @state  int,
```


CUIEEPrintFile2

@errmsg varchar(255)

SELECT @numrows = @@rowcount

/* ERwin Builtin Trigger */

/* TECHNICIAN INSTALLATION on child update no action */

/* ERWIN_RELATION:CHECKSUM="00039314", PARENT_OWNER="", PARENT_TABLE="TECHNICIAN"

CHILD_OWNER="", CHILD_TABLE="INSTALLATION"

P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",

FK_CONSTRAINT="R_47", FK_COLUMNS="Technician_ID1""Warehouse_ID" */

IF

/* %ChildFK(" OR",UPDATE) */

UPDATE(Technician_ID1) OR

UPDATE(Warehouse_ID)

BEGIN

SELECT @nullcnt = 0

SELECT @validcnt = count(*)

FROM inserted,TECHNICIAN

WHERE

/* %JoinFKPK(inserted,TECHNICIAN) */

inserted.Technician_ID1 = TECHNICIAN.Technician_ID and

inserted.Warehouse_ID = TECHNICIAN.Warehouse_ID

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

IF @validcnt + @nullcnt != @numrows

CUIEEPrintFile2

```
BEGIN

    SELECT @errno  = 30007,

           @errmsg = 'Cannot update INSTALLATION because TECHNICIAN does not
exist.'
```

CUIEEPrintFile2

FROM inserted,DELIVERY

WHERE

/* %JoinFKPK(inserted,DELIVERY) */

inserted.Delivery_ID1 = DELIVERY.Delivery_ID and

inserted.Order_ID1 = DELIVERY.Order_ID1 and

inserted.Customer_ID1 = DELIVERY.Customer_ID1 and

inserted.Order_ID = DELIVERY.Order_ID and

inserted.Customer_ID11 = DELIVERY.Customer_ID11

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

IF @validcnt + @nullcnt != @numrows

BEGIN

SELECT @errno = 30007,

@errmsg = 'Cannot update INSTALLATION because DELIVERY does not exist.'

GOTO error

END

END

/* ERwin Builtin Trigger */

RETURN

error:

RAISERROR (@errmsg, -- Message text.

```

                                CUIEEPrintFile2
                                @severity, -- Severity (0~25).

                                @state) -- State (0~255).

rollback transaction

END

go


CREATE TRIGGER td_INVENTORY ON INVENTORY FOR DELETE AS

/* ERwin Builtin Trigger */

/* DELETE trigger on INVENTORY */

BEGIN

    DECLARE @errno    int,

            @severity int,

            @state     int,

            @errmsg    varchar(255)

    /* ERwin Builtin Trigger */

    /* SHOWROOM INVENTORY on child delete no action */

    /* ERWIN_RELATION:CHECKSUM="0004dbb1", PARENT_OWNER="", PARENT_TABLE="SHOWROOM"
    CHILD_OWNER="", CHILD_TABLE="INVENTORY"
    P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
    FK_CONSTRAINT="R_57", FK_COLUMNS="Showroom_ID" */

```

```

                                CUIEEPrintFile2
IF EXISTS (SELECT * FROM deleted,SHOWROOM

WHERE

    /* %JoinFKPK(deleted,SHOWROOM," = "," AND") */

    deleted.Showroom_ID = SHOWROOM.Showroom_ID AND

    NOT EXISTS (

        SELECT * FROM INVENTORY

        WHERE

            /* %JoinFKPK(INVENTORY,SHOWROOM," = "," AND") */

            INVENTORY.Showroom_ID = SHOWROOM.Showroom_ID

    )

)

BEGIN

    SELECT @errno  = 30010,

           @errmsg = 'Cannot delete last INVENTORY because SHOWROOM exists.'

    GOTO error

END

/* ERwin Builtin Trigger */

/* APPLIANCE  INVENTORY on child delete no action */

/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="APPLIANCE"
CHILD_OWNER="", CHILD_TABLE="INVENTORY"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_56", FK_COLUMNS="Appliance_ID1" */
IF EXISTS (SELECT * FROM deleted,APPLIANCE

```

CUIEEPrintFile2

```

WHERE

    /* %JoinFKPK(deleted,APPLIANCE," = "," AND") */

    deleted.Appliance_ID1 = APPLIANCE.Appliance_ID AND

    NOT EXISTS (

        SELECT * FROM INVENTORY

        WHERE

            /* %JoinFKPK(INVENTORY,APPLIANCE," = "," AND") */

            INVENTORY.Appliance_ID1 = APPLIANCE.Appliance_ID

    )

)

BEGIN

    SELECT @errno  = 30010,

           @errmsg = 'Cannot delete last INVENTORY because APPLIANCE exists.'

    GOTO error

END

/* ERwin Builtin Trigger */

/* WAREHOUSE INVENTORY on child delete no action */

/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="WAREHOUSE"
CHILD_OWNER="", CHILD_TABLE="INVENTORY"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_55", FK_COLUMNS="Warehouse_ID1" */

IF EXISTS (SELECT * FROM deleted,WAREHOUSE

    WHERE

```

```

                                CUIEEPrintFile2
/* %JoinFKPK(deleted,WAREHOUSE," = "," AND") */

deleted.Warehouse_ID1 = WAREHOUSE.Warehouse_ID AND

NOT EXISTS (

    SELECT * FROM INVENTORY

    WHERE

        /* %JoinFKPK(INVENTORY,WAREHOUSE," = "," AND") */

        INVENTORY.Warehouse_ID1 = WAREHOUSE.Warehouse_ID

    )

)

BEGIN

    SELECT @errno  = 30010,

           @errmsg = 'Cannot delete last INVENTORY because WAREHOUSE exists.'

    GOTO error

END

/* ERwin Builtin Trigger */

/* SHOWROOM INVENTORY on child delete no action */

/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="SHOWROOM"

CHILD_OWNER="", CHILD_TABLE="INVENTORY"

P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",

FK_CONSTRAINT="R_51", FK_COLUMNS="Showroom_ID" */

IF EXISTS (SELECT * FROM deleted,SHOWROOM

WHERE

    /* %JoinFKPK(deleted,SHOWROOM," = "," AND") */

```

```

                                CUIEEPrintFile2
deleted.Showroom_ID = SHOWROOM.Showroom_ID AND

NOT EXISTS (

    SELECT * FROM INVENTORY

    WHERE

        /* %JoinFKPK(INVENTORY,SHOWROOM," = "," AND") */

        INVENTORY.Showroom_ID = SHOWROOM.Showroom_ID

    )

)

BEGIN

    SELECT @errno  = 30010,

           @errmsg = 'Cannot delete last INVENTORY because SHOWROOM exists.'

    GOTO error

END

/* ERwin Builtin Trigger */

RETURN

error:

RAISERROR (@errmsg, -- Message text.

          @severity, -- Severity (0~25).

          @state) -- State (0~255).

rollback transaction

END

```


go

```
CREATE TRIGGER tU_INVENTORY ON INVENTORY FOR UPDATE AS
```

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

```
/* UPDATE trigger on INVENTORY */
```

```
BEGIN
```

```
DECLARE @numrows int,  
        @nullcnt int,  
        @validcnt int,  
        @insWarehouse_ID1 integer,  
        @insAppliance_ID1 integer,  
        @insWarehouse_ID char(18),  
        @insAppliance_ID char(18),  
        @errno int,  
        @severity int,  
        @state int,  
        @errmsg varchar(255)
```

```
SELECT @numrows = @@rowcount
```

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

```
/* SHOWROOM INVENTORY on child update no action */
```

```
/* ERWIN_RELATION:CHECKSUM="00058b93", PARENT_OWNER="", PARENT_TABLE="SHOWROOM"  
   CHILD_OWNER="", CHILD_TABLE="INVENTORY"
```

```

CUIEEPrintFile2
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_57", FK_COLUMNS="Showroom_ID" */
IF
/* %ChildFK(" OR",UPDATE) */
UPDATE(Showroom_ID)
BEGIN
SELECT @nullcnt = 0
SELECT @validcnt = count(*)
FROM inserted,SHOWROOM
WHERE
/* %JoinFKPK(inserted,SHOWROOM) */
inserted.Showroom_ID = SHOWROOM.Showroom_ID
/* %NotNullFK(inserted," IS NULL","select @nullcnt = count(*) from inserted
where"," AND") */
select @nullcnt = count(*) from inserted where
inserted.Showroom_ID IS NULL
IF @validcnt + @nullcnt != @numrows
BEGIN
SELECT @errno = 30007,
@errmsg = 'Cannot update INVENTORY because SHOWROOM does not exist.'
GOTO error
END
END
END

/* ERwin Builtin Trigger */

```

CUIEEPrintFile2

```
/* APPLIANCE INVENTORY on child update no action */

/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="APPLIANCE"
   CHILD_OWNER="", CHILD_TABLE="INVENTORY"
   P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
   FK_CONSTRAINT="R_56", FK_COLUMNS="Appliance_ID1" */

IF

/* %ChildFK(" OR",UPDATE) */

UPDATE(Appliance_ID1)

BEGIN

SELECT @nullcnt = 0

SELECT @validcnt = count(*)

FROM inserted,APPLIANCE

WHERE

/* %JoinFKPK(inserted,APPLIANCE) */

inserted.Appliance_ID1 = APPLIANCE.Appliance_ID

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

IF @validcnt + @nullcnt != @numrows

BEGIN

SELECT @errno = 30007,

       @errmsg = 'Cannot update INVENTORY because APPLIANCE does not exist.'

GOTO error

END
```

CUIEEPrintFile2

END

/* ERwin Builtin Trigger */

/* WAREHOUSE INVENTORY on child update no action */

/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="WAREHOUSE"

CHILD_OWNER="", CHILD_TABLE="INVENTORY"

P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",

FK_CONSTRAINT="R_55", FK_COLUMNS="Warehouse_ID1" */

IF

/* %ChildFK(" OR",UPDATE) */

UPDATE(Warehouse_ID1)

BEGIN

SELECT @nullcnt = 0

SELECT @validcnt = count(*)

FROM inserted,WAREHOUSE

WHERE

/* %JoinFKPK(inserted,WAREHOUSE) */

inserted.Warehouse_ID1 = WAREHOUSE.Warehouse_ID

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

IF @validcnt + @nullcnt != @numrows

BEGIN

SELECT @errno = 30007,

@errmsg = 'Cannot update INVENTORY because WAREHOUSE does not exist.'

CUIEEEPrintFile2

GOTO error

END

END

/* ERwin Builtin Trigger */

/* SHOWROOM INVENTORY on child update no action */

/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="SHOWROOM"

CHILD_OWNER="", CHILD_TABLE="INVENTORY"

P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",

FK_CONSTRAINT="R_51", FK_COLUMNS="Showroom_ID" */

IF

/* %ChildFK(" OR",UPDATE) */

UPDATE(Showroom_ID)

BEGIN

SELECT @nullcnt = 0

SELECT @validcnt = count(*)

FROM inserted,SHOWROOM

WHERE

/* %JoinFKPK(inserted,SHOWROOM) */

inserted.Showroom_ID = SHOWROOM.Showroom_ID

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

IF @validcnt + @nullcnt != @numrows

CUIEEPrintFile2

BEGIN

SELECT @errno = 30007,

@errmsg = 'Cannot update INVENTORY because SHOWROOM does not exist.'

GOTO error

END

END

/* ERwin Builtin Trigger */

RETURN

error:

RAISERROR (@errmsg, -- Message text.

@severity, -- Severity (0~25).

@state) -- State (0~255).

rollback transaction

END

go

CREATE TRIGGER tD_ORDER ON ORDER FOR DELETE AS

/* ERwin Builtin Trigger */

```

/* DELETE trigger on ORDER */

BEGIN

    DECLARE @errno    int,

            @severity int,

            @state     int,

            @errmsg    varchar(255)

    /* ERwin Builtin Trigger */

    /* ORDER  ORDER_LINE on parent delete no action */

    /* ERWIN_RELATION:CHECKSUM="00064e30", PARENT_OWNER="", PARENT_TABLE="ORDER"
    CHILD_OWNER="", CHILD_TABLE="ORDER_LINE"
    P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
    FK_CONSTRAINT="R_42", FK_COLUMNS="Customer_ID""Order_ID1""Customer_ID1" */

    IF EXISTS (

        SELECT * FROM deleted,ORDER_LINE

        WHERE

            /* %JoinFKPK(ORDER_LINE,deleted," = "," AND") */

            ORDER_LINE.Order_ID1 = deleted.Order_ID AND

            ORDER_LINE.Customer_ID1 = deleted.Customer_ID1 AND

            ORDER_LINE.Customer_ID = deleted.Customer_ID

    )

    BEGIN

        SELECT @errno  = 30001,

               @errmsg = 'Cannot delete ORDER because ORDER_LINE exists.'

        GOTO error
    
```

CUIEEPrintFile2

END

```
/* ERwin Builtin Trigger */

/* ORDER DELIVERY on parent delete no action */

/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="ORDER"
CHILD_OWNER="", CHILD_TABLE="DELIVERY"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_11", FK_COLUMNS="Customer_ID11""Order_ID1""Customer_ID1" */
IF EXISTS (
    SELECT * FROM deleted,DELIVERY
    WHERE
        /* %JoinFKPK(DELIVERY,deleted," = "," AND") */
        DELIVERY.Order_ID1 = deleted.Order_ID AND
        DELIVERY.Customer_ID1 = deleted.Customer_ID1 AND
        DELIVERY.Customer_ID11 = deleted.Customer_ID
)
BEGIN
    SELECT @errno = 30001,
           @errmsg = 'Cannot delete ORDER because DELIVERY exists.'
    GOTO error
END
```

```
/* ERwin Builtin Trigger */

/* ORDER COMMISIION on parent delete no action */
```



```

CUIEEPrintFile2
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="ORDER"
CHILD_OWNER="", CHILD_TABLE="COMMISIION"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_7", FK_COLUMNS="Customer_ID""Order_ID""Customer_ID1" */
IF EXISTS (
    SELECT * FROM deleted,COMMISIION
    WHERE
        /* %JoinFKPK(COMMISIION,deleted," = "," AND") */
        COMMISIION.Order_ID = deleted.Order_ID AND
        COMMISIION.Customer_ID1 = deleted.Customer_ID1 AND
        COMMISIION.Customer_ID = deleted.Customer_ID
)
BEGIN
    SELECT @errno = 30001,
           @errmsg = 'Cannot delete ORDER because COMMISIION exists.'
    GOTO error
END

/* ERwin Builtin Trigger */
/* ASSOCIATE ORDER on child delete no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="ASSOCIATE"
CHILD_OWNER="", CHILD_TABLE="ORDER"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_5", FK_COLUMNS="Associate_ID""Showroom_ID1""Showroom_ID" */

```

```

                                CUIEEPrintFile2
IF EXISTS (SELECT * FROM deleted,ASSOCIATE

WHERE

    /* %JoinFKPK(deleted,ASSOCIATE," = "," AND") */

    deleted.Associate_ID = ASSOCIATE.Associate_ID AND

    deleted.Showroom_ID1 = ASSOCIATE.Showroom_ID1 AND

    deleted.Showroom_ID = ASSOCIATE.Showroom_ID AND

NOT EXISTS (

    SELECT * FROM ORDER

    WHERE

        /* %JoinFKPK(ORDER,ASSOCIATE," = "," AND") */

        ORDER.Associate_ID = ASSOCIATE.Associate_ID AND

        ORDER.Showroom_ID1 = ASSOCIATE.Showroom_ID1 AND

        ORDER.Showroom_ID = ASSOCIATE.Showroom_ID

    )

)

BEGIN

    SELECT @errno  = 30010,

           @errmsg = 'Cannot delete last ORDER because ASSOCIATE exists.'

    GOTO error

END

/* ERwin Builtin Trigger */

/* CUSTOMER  ORDER on child delete no action */

/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="CUSTOMER"

```

```

                                CUIEEPrintFile2
CHILD_OWNER="", CHILD_TABLE="ORDER"

P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",

FK_CONSTRAINT="R_4", FK_COLUMNS="Customer_ID1" */

IF EXISTS (SELECT * FROM deleted,CUSTOMER

    WHERE

        /* %JoinFKPK(deleted,CUSTOMER," = "," AND") */

        deleted.Customer_ID1 = CUSTOMER.Customer_ID AND

        NOT EXISTS (

            SELECT * FROM ORDER

            WHERE

                /* %JoinFKPK(ORDER,CUSTOMER," = "," AND") */

                ORDER.Customer_ID1 = CUSTOMER.Customer_ID

        )

    )

BEGIN

    SELECT @errno  = 30010,

           @errmsg = 'Cannot delete last ORDER because CUSTOMER exists.'

    GOTO error

END

```

```

/* ERwin Builtin Trigger */

RETURN

```

error:

```

                                CUIEEPrintFile2
RAISERROR (@errmsg, -- Message text.

           @severity, -- Severity (0~25).

           @state) -- State (0~255).

rollback transaction

END

go

```

```

CREATE TRIGGER tU_ORDER ON ORDER FOR UPDATE AS

/* ERwin Builtin Trigger */

/* UPDATE trigger on ORDER */

BEGIN

  DECLARE @numrows int,

          @nullcnt int,

          @validcnt int,

          @insOrder_ID integer,

          @insCustomer_ID1 integer,

          @insCustomer_ID char(18),

          @errno int,

          @severity int,

          @state int,

          @errmsg varchar(255)

```

CUIEEPrintFile2

```

SELECT @numrows = @@rowcount

/* ERwin Builtin Trigger */

/* ORDER  ORDER_LINE on parent update no action */

/* ERWIN_RELATION:CHECKSUM="00075673", PARENT_OWNER="", PARENT_TABLE="ORDER"
  CHILD_OWNER="", CHILD_TABLE="ORDER_LINE"
  P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
  FK_CONSTRAINT="R_42", FK_COLUMNS="Customer_ID""Order_ID1""Customer_ID1" */
IF
  /* %ParentPK(" OR",UPDATE) */
  UPDATE(Order_ID) OR
  UPDATE(Customer_ID1) OR
  UPDATE(Customer_ID)
BEGIN
  IF EXISTS (
    SELECT * FROM deleted,ORDER_LINE
    WHERE
      /* %JoinFKPK(ORDER_LINE,deleted," = "," AND") */
      ORDER_LINE.Order_ID1 = deleted.Order_ID AND
      ORDER_LINE.Customer_ID1 = deleted.Customer_ID1 AND
      ORDER_LINE.Customer_ID = deleted.Customer_ID
  )
  BEGIN
    SELECT @errno = 30005,
           @errmsg = 'Cannot update ORDER because ORDER_LINE exists.'
```

CUIEEPrintFile2

```
GOTO error

END

END

/* ERwin Builtin Trigger */

/* ORDER DELIVERY on parent update no action */

/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="ORDER"
   CHILD_OWNER="", CHILD_TABLE="DELIVERY"
   P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
   FK_CONSTRAINT="R_11", FK_COLUMNS="Customer_ID11""Order_ID1""Customer_ID1" */
IF
  /* %ParentPK(" OR",UPDATE) */
  UPDATE(Order_ID) OR
  UPDATE(Customer_ID1) OR
  UPDATE(Customer_ID)
BEGIN
  IF EXISTS (
    SELECT * FROM deleted,DELIVERY
    WHERE
      /* %JoinFKPK(DELIVERY,deleted," = "," AND") */
      DELIVERY.Order_ID1 = deleted.Order_ID AND
      DELIVERY.Customer_ID1 = deleted.Customer_ID1 AND
      DELIVERY.Customer_ID11 = deleted.Customer_ID
  )
```

CUIEEPrintFile2

```
BEGIN

    SELECT @errno  = 30005,

           @errmsg = 'Cannot update ORDER because DELIVERY exists.'

    GOTO error

END

END

/* ERwin Builtin Trigger */

/* ORDER  COMMISIION on parent update no action */

/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="ORDER"

   CHILD_OWNER="", CHILD_TABLE="COMMISIION"

   P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",

   FK_CONSTRAINT="R_7", FK_COLUMNS="Customer_ID""Order_ID""Customer_ID1" */

IF

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

    UPDATE(Order_ID) OR

    UPDATE(Customer_ID1) OR

    UPDATE(Customer_ID)

BEGIN

    IF EXISTS (

        SELECT * FROM deleted,COMMISIION

        WHERE

            /* %JoinFKPK(COMMISIION,deleted," = "," AND") */

            COMMISIION.Order_ID = deleted.Order_ID AND
```

```

                                CUIEEPrintFile2
        COMMISIION.Customer_ID1 = deleted.Customer_ID1 AND

        COMMISIION.Customer_ID = deleted.Customer_ID

    )

BEGIN

    SELECT @errno  = 30005,

           @errmsg = 'Cannot update ORDER because COMMISIION exists.'

    GOTO error

END

END

/* ERwin Builtin Trigger */

/* ASSOCIATE  ORDER on child update no action */

/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="ASSOCIATE"
   CHILD_OWNER="", CHILD_TABLE="ORDER"
   P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
   FK_CONSTRAINT="R_5", FK_COLUMNS="Associate_ID""Showroom_ID1""Showroom_ID" */
IF

    /* %ChildFK(" OR",UPDATE) */

    UPDATE(Associate_ID) OR

    UPDATE(Showroom_ID1) OR

    UPDATE(Showroom_ID)

BEGIN

    SELECT @nullcnt = 0

    SELECT @validcnt = count(*)

```


FROM inserted,ASSOCIATE

WHERE

/* %JoinFKPK(inserted,ASSOCIATE) */

inserted.Associate_ID = ASSOCIATE.Associate_ID and

inserted.Showroom_ID1 = ASSOCIATE.Showroom_ID1 and

inserted.Showroom_ID = ASSOCIATE.Showroom_ID

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

select @nullcnt = count(*) from inserted where

inserted.Associate_ID IS NULL AND

inserted.Showroom_ID1 IS NULL AND

inserted.Showroom_ID IS NULL

IF @validcnt + @nullcnt != @numrows

BEGIN

SELECT @errno = 30007,

@errmsg = 'Cannot update ORDER because ASSOCIATE does not exist.'

GOTO error

END

END

/* ERwin Builtin Trigger */

/* CUSTOMER ORDER on child update no action */

/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="CUSTOMER"

CHILD_OWNER="", CHILD_TABLE="ORDER"

P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",

CUIEEPrintFile2

```
FK_CONSTRAINT="R_4", FK_COLUMNS="Customer_ID1" */
IF
    /* %ChildFK(" OR",UPDATE) */
    UPDATE(Customer_ID1)
BEGIN
    SELECT @nullcnt = 0
    SELECT @validcnt = count(*)
        FROM inserted,CUSTOMER
        WHERE
            /* %JoinFKPK(inserted,CUSTOMER) */
            inserted.Customer_ID1 = CUSTOMER.Customer_ID
    /* %NotnullFK(inserted," IS NULL","select @nullcnt = count(*) from inserted
where"," AND") */

    IF @validcnt + @nullcnt != @numrows
    BEGIN
        SELECT @errno = 30007,
            @errmsg = 'Cannot update ORDER because CUSTOMER does not exist.'
        GOTO error
    END
END

/* ERwin Builtin Trigger */
```

CUIEEEPrintFile2

RETURN

error:

RAISERROR (@errmsg, -- Message text.

 @severity, -- Severity (0~25).

 @state) -- State (0~255).

rollback transaction

END

go

CREATE TRIGGER tD_ORDER_LINE ON ORDER_LINE FOR DELETE AS

/* ERwin Builtin Trigger */

/* DELETE trigger on ORDER_LINE */

BEGIN

DECLARE @errno int,

 @severity int,

 @state int,

 @errmsg varchar(255)

/* ERwin Builtin Trigger */

/* APPLIANCE ORDER_LINE on child delete no action */

/* ERWIN_RELATION:CHECKSUM="0002d67d", PARENT_OWNER="", PARENT_TABLE="APPLIANCE"

```

                                CUIEEPrintFile2
CHILD_OWNER="", CHILD_TABLE="ORDER_LINE"

P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",

FK_CONSTRAINT="R_43", FK_COLUMNS="Appliance_ID1" */

IF EXISTS (SELECT * FROM deleted,APPLIANCE

WHERE

/* %JoinFKPK(deleted,APPLIANCE," = "," AND") */

deleted.Appliance_ID1 = APPLIANCE.Appliance_ID AND

NOT EXISTS (

SELECT * FROM ORDER_LINE

WHERE

/* %JoinFKPK(ORDER_LINE,APPLIANCE," = "," AND") */

ORDER_LINE.Appliance_ID1 = APPLIANCE.Appliance_ID

)

)

BEGIN

SELECT @errno = 30010,

       @errmsg = 'Cannot delete last ORDER_LINE because APPLIANCE exists.'

GOTO error

END

/* ERwin Builtin Trigger */

/* ORDER ORDER_LINE on child delete no action */

/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="ORDER"

CHILD_OWNER="", CHILD_TABLE="ORDER_LINE"

```

```

CUIEEPrintFile2
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_42", FK_COLUMNS="Customer_ID""Order_ID1""Customer_ID1" */
IF EXISTS (SELECT * FROM deleted,ORDER
WHERE
/* %JoinFKPK(deleted,ORDER," = "," AND") */
deleted.Order_ID1 = ORDER.Order_ID AND
deleted.Customer_ID1 = ORDER.Customer_ID1 AND
deleted.Customer_ID = ORDER.Customer_ID AND
NOT EXISTS (
SELECT * FROM ORDER_LINE
WHERE
/* %JoinFKPK(ORDER_LINE,ORDER," = "," AND") */
ORDER_LINE.Order_ID1 = ORDER.Order_ID AND
ORDER_LINE.Customer_ID1 = ORDER.Customer_ID1 AND
ORDER_LINE.Customer_ID = ORDER.Customer_ID
)
)
BEGIN
SELECT @errno = 30010,
@errmsg = 'Cannot delete last ORDER_LINE because ORDER exists.'
GOTO error
END

```

CUIEEEPrintFile2

/* ERwin Builtin Trigger */

RETURN

error:

RAISERROR (@errmsg, -- Message text.

 @severity, -- Severity (0~25).

 @state) -- State (0~255).

rollback transaction

END

go

CREATE TRIGGER tu_ORDER_LINE ON ORDER_LINE FOR UPDATE AS

/* ERwin Builtin Trigger */

/* UPDATE trigger on ORDER_LINE */

BEGIN

DECLARE @numrows int,

 @nullcnt int,

 @validcnt int,

 @insOrder_ID1 integer,

 @insAppliance_ID1 integer,

 @insCustomer_ID1 integer,

 @insOrder_ID char(18),

 @insAppliance_ID char(18),

```

                                CUIEEPrintFile2
    @insCustomer_ID char(18),

    @errno    int,

    @severity int,

    @state    int,

    @errmsg   varchar(255)


SELECT @numrows = @@rowcount

/* ERwin Builtin Trigger */

/* APPLIANCE ORDER_LINE on child update no action */

/* ERWIN_RELATION:CHECKSUM="0002e81d", PARENT_OWNER="", PARENT_TABLE="APPLIANCE"
   CHILD_OWNER="", CHILD_TABLE="ORDER_LINE"
   P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
   FK_CONSTRAINT="R_43", FK_COLUMNS="Appliance_ID1" */

IF

    /* %ChildFK(" OR",UPDATE) */

    UPDATE(Appliance_ID1)

BEGIN

    SELECT @nullcnt = 0

    SELECT @validcnt = count(*)

        FROM inserted,APPLIANCE

        WHERE

            /* %JoinFKPK(inserted,APPLIANCE) */

            inserted.Appliance_ID1 = APPLIANCE.Appliance_ID

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

```

CUIEEPrintFile2

```
IF @validcnt + @nullcnt != @numrows

BEGIN

    SELECT @errno = 30007,

           @errmsg = 'Cannot update ORDER_LINE because APPLIANCE does not exist.'

    GOTO error

END

END

/* ERwin Builtin Trigger */

/* ORDER ORDER_LINE on child update no action */

/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="ORDER"
   CHILD_OWNER="", CHILD_TABLE="ORDER_LINE"
   P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
   FK_CONSTRAINT="R_42", FK_COLUMNS="Customer_ID""Order_ID1""Customer_ID1" */

IF

    /* %ChildFK(" OR",UPDATE) */

    UPDATE(Order_ID1) OR

    UPDATE(Customer_ID1) OR

    UPDATE(Customer_ID)

BEGIN

    SELECT @nullcnt = 0

    SELECT @validcnt = count(*)

    FROM inserted,ORDER
```


CUIEEPrintFile2

WHERE

/* %JoinFKPK(inserted,ORDER) */

inserted.Order_ID1 = ORDER.Order_ID and

inserted.Customer_ID1 = ORDER.Customer_ID1 and

inserted.Customer_ID = ORDER.Customer_ID

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

IF @validcnt + @nullcnt != @numrows

BEGIN

SELECT @errno = 30007,

@errmsg = 'Cannot update ORDER_LINE because ORDER does not exist.'

GOTO error

END

END

/* ERwin Builtin Trigger */

RETURN

error:

RAISERROR (@errmsg, -- Message text.

@severity, -- Severity (0~25).

@state) -- State (0~255).

rollback transaction

CUIEEEPrintFile2

END

go

```
CREATE TRIGGER td_SHIPMENT ON SHIPMENT FOR DELETE AS

/* ERwin Builtin Trigger */

/* DELETE trigger on SHIPMENT */

BEGIN

    DECLARE @errno    int,

            @severity int,

            @state     int,

            @errmsg    varchar(255)

    /* ERwin Builtin Trigger */

    /* WAREHOUSE SHIPMENT on child delete no action */

    /* ERWIN_RELATION:CHECKSUM="0002660f", PARENT_OWNER="", PARENT_TABLE="WAREHOUSE"
    CHILD_OWNER="", CHILD_TABLE="SHIPMENT"
    P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
    FK_CONSTRAINT="R_24", FK_COLUMNS="Warehouse_ID1" */

    IF EXISTS (SELECT * FROM deleted,WAREHOUSE

        WHERE

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

```

                                CUIEEPrintFile2
deleted.Warehouse_ID1 = WAREHOUSE.Warehouse_ID AND

NOT EXISTS (

    SELECT * FROM SHIPMENT

    WHERE

        /* %JoinFKPK(SHIPMENT,WAREHOUSE," = "," AND") */

        SHIPMENT.Warehouse_ID1 = WAREHOUSE.Warehouse_ID

    )

)

BEGIN

    SELECT @errno  = 30010,

           @errmsg = 'Cannot delete last SHIPMENT because WAREHOUSE exists.'

    GOTO error

END

/* ERwin Builtin Trigger */

/* VENDOR  SHIPMENT on child delete no action */

/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="VENDOR"
CHILD_OWNER="", CHILD_TABLE="SHIPMENT"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_23", FK_COLUMNS="Vendor_ID1" */

IF EXISTS (SELECT * FROM deleted,VENDOR

    WHERE

        /* %JoinFKPK(deleted,VENDOR," = "," AND") */

        deleted.Vendor_ID1 = VENDOR.Vendor_ID AND

```

CUIEEPrintFile2

NOT EXISTS (

SELECT * FROM SHIPMENT

WHERE

/* %JoinFKPK(SHIPMENT,VENDOR," = "," AND") */

SHIPMENT.Vendor_ID1 = VENDOR.Vendor_ID

)

)

BEGIN

SELECT @errno = 30010,

@errmsg = 'Cannot delete last SHIPMENT because VENDOR exists.'

GOTO error

END

/* ERwin Builtin Trigger */

RETURN

error:

RAISERROR (@errmsg, -- Message text.

@severity, -- Severity (0~25).

@state) -- State (0~255).

rollback transaction

END

go

CUIEEPrintFile2

```
CREATE TRIGGER tU_SHIPMENT ON SHIPMENT FOR UPDATE AS

/* ERwin Builtin Trigger */

/* UPDATE trigger on SHIPMENT */

BEGIN

    DECLARE @numrows int,

            @nullcnt int,

            @validcnt int,

            @insShiment_ID integer,

            @insVendor_ID1 integer,

            @insWarehouse_ID1 integer,

            @insVendor_ID char(18),

            @insWarehouse_ID char(18),

            @errno int,

            @severity int,

            @state int,

            @errmsg varchar(255)


    SELECT @numrows = @@rowcount

/* ERwin Builtin Trigger */

/* WAREHOUSE SHIPMENT on child update no action */

/* ERWIN_RELATION:CHECKSUM="0002b19d", PARENT_OWNER="", PARENT_TABLE="WAREHOUSE"

   CHILD_OWNER="", CHILD_TABLE="SHIPMENT"
```

```

                                CUIEEPrintFile2
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",

FK_CONSTRAINT="R_24", FK_COLUMNS="Warehouse_ID1" */

IF

/* %ChildFK(" OR",UPDATE) */

UPDATE(Warehouse_ID1)

BEGIN

SELECT @nullcnt = 0

SELECT @validcnt = count(*)

FROM inserted,WAREHOUSE

WHERE

/* %JoinFKPK(inserted,WAREHOUSE) */

inserted.Warehouse_ID1 = WAREHOUSE.Warehouse_ID

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

IF @validcnt + @nullcnt != @numrows

BEGIN

SELECT @errno = 30007,

@errmsg = 'Cannot update SHIPMENT because WAREHOUSE does not exist.'

GOTO error

END

END

/* ERwin Builtin Trigger */

/* VENDOR SHIPMENT on child update no action */

```

CUIEEPrintFile2

```

/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="VENDOR"

CHILD_OWNER="", CHILD_TABLE="SHIPMENT"

P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",

FK_CONSTRAINT="R_23", FK_COLUMNS="Vendor_ID1" */

IF

/* %ChildFK(" OR",UPDATE) */

UPDATE(Vendor_ID1)

BEGIN

SELECT @nullcnt = 0

SELECT @validcnt = count(*)

FROM inserted,VENDOR

WHERE

/* %JoinFKPK(inserted,VENDOR) */

inserted.Vendor_ID1 = VENDOR.Vendor_ID

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

IF @validcnt + @nullcnt != @numrows

BEGIN

SELECT @errno = 30007,

@errmsg = 'Cannot update SHIPMENT because VENDOR does not exist.'

GOTO error

END

END

```

CUIEEPrintFile2

```
/* ERwin Builtin Trigger */

RETURN

error:

    RAISERROR (@errmsg, -- Message text.
               @severity, -- Severity (0~25).
               @state) -- State (0~255).

    rollback transaction

END

go
```

```
CREATE TRIGGER tD_SHOWROOM ON SHOWROOM FOR DELETE AS

/* ERwin Builtin Trigger */

/* DELETE trigger on SHOWROOM */

BEGIN

    DECLARE @errno    int,
            @severity int,
            @state     int,
            @errmsg    varchar(255)
```



```

CUIEEPrintFile2

/* ERwin Builtin Trigger */

/* SHOWROOM INVENTORY on parent delete no action */

/* ERWIN_RELATION:CHECKSUM="0002fcce", PARENT_OWNER="", PARENT_TABLE="SHOWROOM"
CHILD_OWNER="", CHILD_TABLE="INVENTORY"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_57", FK_COLUMNS="Showroom_ID" */

IF EXISTS (

    SELECT * FROM deleted,INVENTORY

    WHERE

        /* %JoinFKPK(INVENTORY,deleted," = "," AND") */

        INVENTORY.Showroom_ID = deleted.Showroom_ID

)

BEGIN

    SELECT @errno = 30001,

           @errmsg = 'Cannot delete SHOWROOM because INVENTORY exists.'

    GOTO error

END

/* ERwin Builtin Trigger */

/* SHOWROOM INVENTORY on parent delete no action */

/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="SHOWROOM"
CHILD_OWNER="", CHILD_TABLE="INVENTORY"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_51", FK_COLUMNS="Showroom_ID" */

```

CUIEEPrintFile2

```
IF EXISTS (
    SELECT * FROM deleted,INVENTORY
    WHERE
        /* %JoinFKPK(INVENTORY,deleted," = "," AND") */
        INVENTORY.Showroom_ID = deleted.Showroom_ID
)
BEGIN
    SELECT @errno = 30001,
           @errmsg = 'Cannot delete SHOWROOM because INVENTORY exists.'
    GOTO error
END

/* ERwin Builtin Trigger */
/* SHOWROOM ASSOCIATE on parent delete no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="SHOWROOM"
CHILD_OWNER="", CHILD_TABLE="ASSOCIATE"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_2", FK_COLUMNS="Showroom_ID1" */
IF EXISTS (
    SELECT * FROM deleted,ASSOCIATE
    WHERE
        /* %JoinFKPK(ASSOCIATE,deleted," = "," AND") */
        ASSOCIATE.Showroom_ID1 = deleted.Showroom_ID
)
```

CUIEEPrintFile2

BEGIN

SELECT @errno = 30001,

@errmsg = 'Cannot delete SHOWROOM because ASSOCIATE exists.'

GOTO error

END

/* ERwin Builtin Trigger */

RETURN

error:

RAISERROR (@errmsg, -- Message text.

@severity, -- Severity (0~25).

@state) -- State (0~255).

rollback transaction

END

go

CREATE TRIGGER tU_SHOWROOM ON SHOWROOM FOR UPDATE AS

/* ERwin Builtin Trigger */

/* UPDATE trigger on SHOWROOM */

BEGIN

DECLARE @numrows int,

CUIEEPrintFile2

```
@nullcnt int,
@validcnt int,
@insShowroom_ID integer,
@errno int,
@severity int,
@state int,
@errmsg varchar(255)

SELECT @numrows = @@rowcount
/* ERwin Builtin Trigger */
/* SHOWROOM INVENTORY on parent update no action */
/* ERWIN_RELATION:CHECKSUM="00034952", PARENT_OWNER="", PARENT_TABLE="SHOWROOM"
CHILD_OWNER="", CHILD_TABLE="INVENTORY"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_57", FK_COLUMNS="Showroom_ID" */
IF
/* %ParentPK(" OR",UPDATE) */
UPDATE(Showroom_ID)
BEGIN
IF EXISTS (
SELECT * FROM deleted,INVENTORY
WHERE
/* %JoinFKPK(INVENTORY,deleted," = "," AND") */
INVENTORY.Showroom_ID = deleted.Showroom_ID
```

```

                                CUIEEPrintFile2
)
BEGIN
    SELECT @errno  = 30005,
           @errmsg = 'Cannot update SHOWROOM because INVENTORY exists.'

    GOTO error

END

END

/* ERwin Builtin Trigger */
/* SHOWROOM INVENTORY on parent update no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="SHOWROOM"
   CHILD_OWNER="", CHILD_TABLE="INVENTORY"
   P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
   FK_CONSTRAINT="R_51", FK_COLUMNS="Showroom_ID" */
IF
    /* %ParentPK(" OR",UPDATE) */
    UPDATE(Showroom_ID)
BEGIN
    IF EXISTS (
        SELECT * FROM deleted,INVENTORY
        WHERE
            /* %JoinFKPK(INVENTORY,deleted," = "," AND") */
            INVENTORY.Showroom_ID = deleted.Showroom_ID
    )

```

```

                                CUIEEPrintFile2

BEGIN

    SELECT @errno  = 30005,

           @errmsg = 'Cannot update SHOWROOM because INVENTORY exists.'

    GOTO error

END

END

/* ERwin Builtin Trigger */

/* SHOWROOM ASSOCIATE on parent update no action */

/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="SHOWROOM"
   CHILD_OWNER="", CHILD_TABLE="ASSOCIATE"
   P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
   FK_CONSTRAINT="R_2", FK_COLUMNS="Showroom_ID1" */

IF

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

    UPDATE(Showroom_ID)

BEGIN

    IF EXISTS (

        SELECT * FROM deleted,ASSOCIATE

        WHERE

            /* %JoinFKPK(ASSOCIATE,deleted," = "," AND") */

            ASSOCIATE.Showroom_ID1 = deleted.Showroom_ID

    )

BEGIN

```

```

                                CUIEEPrintFile2
SELECT @errno  = 30005,
       @errmsg = 'Cannot update SHOWROOM because ASSOCIATE exists.'
GOTO error
END
END

```

```

/* ERwin Builtin Trigger */

RETURN

error:

    RAISERROR (@errmsg, -- Message text.
               @severity, -- Severity (0~25).
               @state) -- State (0~255).

    rollback transaction

END

go

```

```

CREATE TRIGGER td_TECHNICIAN ON TECHNICIAN FOR DELETE AS

/* ERwin Builtin Trigger */

/* DELETE trigger on TECHNICIAN */

```

CUIEEPrintFile2

BEGIN

```
DECLARE @errno    int,
        @severity int,
        @state    int,
        @errmsg   varchar(255)
```

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

```
/* TECHNICIAN  INSTALLATION on parent delete no action */
```

```
/* ERWIN_RELATION:CHECKSUM="0004fb8d", PARENT_OWNER="",
PARENT_TABLE="TECHNICIAN"
```

```
CHILD_OWNER="", CHILD_TABLE="INSTALLATION"
```

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

```
FK_CONSTRAINT="R_47", FK_COLUMNS="Technician_ID1""Warehouse_ID" */
```

```
IF EXISTS (
```

```
    SELECT * FROM deleted,INSTALLATION
```

```
WHERE
```

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

```
    INSTALLATION.Technician_ID1 = deleted.Technician_ID AND
```

```
    INSTALLATION.Warehouse_ID = deleted.Warehouse_ID
```

```
)
```

BEGIN

```
    SELECT @errno = 30001,
```

```
           @errmsg = 'Cannot delete TECHNICIAN because INSTALLATION exists.'
```

```
    GOTO error
```

END

CUIEEPrintFile2

```
/* ERwin Builtin Trigger */

/* TECHNICIAN  TECHNICIAN on parent delete no action */

/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="",
PARENT_TABLE="TECHNICIAN"

CHILD_OWNER="", CHILD_TABLE="TECHNICIAN"

P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",

FK_CONSTRAINT="R_1", FK_COLUMNS="Supervisor_ID""Warehouse_ID" */

IF EXISTS (

    SELECT * FROM deleted,TECHNICIAN

    WHERE

        /* %JoinFKPK(TECHNICIAN,deleted," = "," AND") */

        TECHNICIAN.Supervisor_ID = deleted.Technician_ID AND

        TECHNICIAN.Warehouse_ID = deleted.Warehouse_ID

)

BEGIN

    SELECT @errno  = 30001,

           @errmsg = 'Cannot delete TECHNICIAN because TECHNICIAN exists.'

    GOTO error

END

/* ERwin Builtin Trigger */

/* WAREHOUSE  TECHNICIAN on child delete no action */

/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="WAREHOUSE"

CHILD_OWNER="", CHILD_TABLE="TECHNICIAN"
```

```

CUIEEPrintFile2
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_12", FK_COLUMNS="Warehouse_ID" */
IF EXISTS (SELECT * FROM deleted,WAREHOUSE
WHERE
/* %JoinFKPK(deleted,WAREHOUSE," = "," AND") */
deleted.Warehouse_ID = WAREHOUSE.Warehouse_ID AND
NOT EXISTS (
SELECT * FROM TECHNICIAN
WHERE
/* %JoinFKPK(TECHNICIAN,WAREHOUSE," = "," AND") */
TECHNICIAN.Warehouse_ID = WAREHOUSE.Warehouse_ID
)
)
BEGIN
SELECT @errno = 30010,
@errmsg = 'Cannot delete last TECHNICIAN because WAREHOUSE exists.'
GOTO error
END

/* ERwin Builtin Trigger */
/* TECHNICIAN TECHNICIAN on child delete no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="",
PARENT_TABLE="TECHNICIAN"
CHILD_OWNER="", CHILD_TABLE="TECHNICIAN"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",

```

CUIEEPrintFile2

```
FK_CONSTRAINT="R_1", FK_COLUMNS="Supervisor_ID""Warehouse_ID" */
IF EXISTS (SELECT * FROM deleted,TECHNICIAN
WHERE
/* %JoinFKPK(deleted,TECHNICIAN," = "," AND") */
deleted.Supervisor_ID = TECHNICIAN.Technician_ID AND
deleted.Warehouse_ID = TECHNICIAN.Warehouse_ID AND
NOT EXISTS (
SELECT * FROM TECHNICIAN
WHERE
/* %JoinFKPK(TECHNICIAN,TECHNICIAN," = "," AND") */
TECHNICIAN.Supervisor_ID = TECHNICIAN.Technician_ID AND
TECHNICIAN.Warehouse_ID = TECHNICIAN.Warehouse_ID
)
)
BEGIN
SELECT @errno = 30010,
@errmsg = 'Cannot delete last TECHNICIAN because TECHNICIAN exists.'
GOTO error
END

/* ERwin Builtin Trigger */
RETURN
```

error:

CUIEEEPrintFile2

```
RAISERROR (@errmsg, -- Message text.  
           @severity, -- Severity (0~25).  
           @state) -- State (0~255).
```

```
rollback transaction
```

```
END
```

```
go
```

```
CREATE TRIGGER tU_TECHNICIAN ON TECHNICIAN FOR UPDATE AS
```

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

```
/* UPDATE trigger on TECHNICIAN */
```

```
BEGIN
```

```
DECLARE @numrows int,  
        @nullcnt int,  
        @validcnt int,  
        @insTechnician_ID integer,  
        @insWarehouse_ID integer,  
        @errno int,  
        @severity int,  
        @state int,  
        @errmsg varchar(255)
```

```
SELECT @numrows = @@rowcount
```

CUIEEPrintFile2

```
/* ERwin Builtin Trigger */

/* TECHNICIAN  INSTALLATION on parent update no action */

/* ERWIN_RELATION:CHECKSUM="0005b8c3", PARENT_OWNER="", PARENT_TABLE="TECHNICIAN"
   CHILD_OWNER="", CHILD_TABLE="INSTALLATION"
   P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
   FK_CONSTRAINT="R_47", FK_COLUMNS="Technician_ID1""Warehouse_ID" */

IF

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

UPDATE(Technician_ID) OR

UPDATE(Warehouse_ID)

BEGIN

IF EXISTS (

    SELECT * FROM deleted,INSTALLATION

    WHERE

        /* %JoinFKPK(INSTALLATION,deleted," = "," AND") */

        INSTALLATION.Technician_ID1 = deleted.Technician_ID AND

        INSTALLATION.Warehouse_ID = deleted.Warehouse_ID

)

BEGIN

    SELECT @errno  = 30005,

           @errmsg = 'Cannot update TECHNICIAN because INSTALLATION exists.'

    GOTO error

END

END
```

CUIEEPrintFile2

```
/* ERwin Builtin Trigger */

/* TECHNICIAN  TECHNICIAN on parent update no action */

/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="TECHNICIAN"
   CHILD_OWNER="", CHILD_TABLE="TECHNICIAN"
   P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
   FK_CONSTRAINT="R_1", FK_COLUMNS="Supervisor_ID""Warehouse_ID" */

IF

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

UPDATE(Technician_ID) OR

UPDATE(Warehouse_ID)

BEGIN

IF EXISTS (

    SELECT * FROM deleted,TECHNICIAN

    WHERE

        /* %JoinFKPK(TECHNICIAN,deleted," = "," AND") */

        TECHNICIAN.Supervisor_ID = deleted.Technician_ID AND

        TECHNICIAN.Warehouse_ID = deleted.Warehouse_ID

)

BEGIN

    SELECT @errno  = 30005,

           @errmsg = 'Cannot update TECHNICIAN because TECHNICIAN exists.'

    GOTO error

END
```

CUIEEPrintFile2

END

/* ERwin Builtin Trigger */

/* WAREHOUSE TECHNICIAN on child update no action */

/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="WAREHOUSE"

CHILD_OWNER="", CHILD_TABLE="TECHNICIAN"

P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",

FK_CONSTRAINT="R_12", FK_COLUMNS="Warehouse_ID" */

IF

/* %ChildFK(" OR",UPDATE) */

UPDATE(Warehouse_ID)

BEGIN

SELECT @nullcnt = 0

SELECT @validcnt = count(*)

FROM inserted,WAREHOUSE

WHERE

/* %JoinFKPK(inserted,WAREHOUSE) */

inserted.Warehouse_ID = WAREHOUSE.Warehouse_ID

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

IF @validcnt + @nullcnt != @numrows

BEGIN

SELECT @errno = 30007,

```

                                CUIEEPrintFile2
                                @errmsg = 'Cannot update TECHNICIAN because WAREHOUSE does not exist.'

                                GOTO error

                                END

                                END

/* ERwin Builtin Trigger */

/* TECHNICIAN  TECHNICIAN on child update no action */

/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="TECHNICIAN"
   CHILD_OWNER="", CHILD_TABLE="TECHNICIAN"
   P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
   FK_CONSTRAINT="R_1", FK_COLUMNS="Supervisor_ID""Warehouse_ID" */

IF

/* %ChildFK(" OR",UPDATE) */

UPDATE(Supervisor_ID) OR

UPDATE(Warehouse_ID)

BEGIN

SELECT @nullcnt = 0

SELECT @validcnt = count(*)

FROM inserted,TECHNICIAN

WHERE

/* %JoinFKPK(inserted,TECHNICIAN) */

inserted.Supervisor_ID = TECHNICIAN.Technician_ID and

inserted.Warehouse_ID = TECHNICIAN.Warehouse_ID

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

```


CUIEEPrintFile2

```
select @nullcnt = count(*) from inserted where
```

```
    inserted.Supervisor_ID IS NULL AND
```

```
    inserted.Warehouse_ID IS NULL
```

```
IF @validcnt + @nullcnt != @numrows
```

```
BEGIN
```

```
    SELECT @errno  = 30007,
```

```
           @errmsg = 'Cannot update TECHNICIAN because TECHNICIAN does not exist.'
```

```
    GOTO error
```

```
END
```

```
END
```

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

```
RETURN
```

```
error:
```

```
    RAISERROR (@errmsg, -- Message text.
```

```
               @severity, -- Severity (0~25).
```

```
               @state) -- State (0~255).
```

```
    rollback transaction
```

```
END
```

```
go
```

CUIEEPrintFile2

```
CREATE TRIGGER tD_VENDOR ON VENDOR FOR DELETE AS

/* ERwin Builtin Trigger */

/* DELETE trigger on VENDOR */

BEGIN

    DECLARE @errno    int,

            @severity int,

            @state     int,

            @errmsg    varchar(255)

    /* ERwin Builtin Trigger */

    /* VENDOR  APPLIANCE on parent delete no action */

    /* ERWIN_RELATION:CHECKSUM="0001f54c", PARENT_OWNER="", PARENT_TABLE="VENDOR"
    CHILD_OWNER="", CHILD_TABLE="APPLIANCE"
    P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
    FK_CONSTRAINT="R_29", FK_COLUMNS="Vendor_ID" */

    IF EXISTS (

        SELECT * FROM deleted,APPLIANCE

        WHERE

            /* %JoinFKPK(APPLIANCE,deleted," = "," AND") */

            APPLIANCE.Vendor_ID = deleted.Vendor_ID

    )

    BEGIN

        SELECT @errno = 30001,
```

CUIEEPrintFile2

```
@errmsg = 'Cannot delete VENDOR because APPLIANCE exists.'

GOTO error

END

/* ERwin Builtin Trigger */

/* VENDOR SHIPMENT on parent delete no action */

/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="VENDOR"
CHILD_OWNER="", CHILD_TABLE="SHIPMENT"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_23", FK_COLUMNS="Vendor_ID1" */
IF EXISTS (
    SELECT * FROM deleted,SHIPMENT
    WHERE
        /* %JoinFKPK(SHIPMENT,deleted," = "," AND") */
        SHIPMENT.Vendor_ID1 = deleted.Vendor_ID
)
BEGIN
    SELECT @errno = 30001,
           @errmsg = 'Cannot delete VENDOR because SHIPMENT exists.'
    GOTO error
END

/* ERwin Builtin Trigger */
```

CUIEEEPrintFile2

RETURN

error:

RAISERROR (@errmsg, -- Message text.

 @severity, -- Severity (0~25).

 @state) -- State (0~255).

rollback transaction

END

go

CREATE TRIGGER tU_VENDOR ON VENDOR FOR UPDATE AS

/* ERwin Builtin Trigger */

/* UPDATE trigger on VENDOR */

BEGIN

DECLARE @numrows int,

 @nullcnt int,

 @validcnt int,

 @insVendor_ID integer,

 @errno int,

 @severity int,

 @state int,

 @errmsg varchar(255)

CUIEEEPrintFile2

```
SELECT @numrows = @@rowcount

/* ERwin Builtin Trigger */

/* VENDOR  APPLIANCE on parent update no action */

/* ERWIN_RELATION:CHECKSUM="0002223d", PARENT_OWNER="", PARENT_TABLE="VENDOR"
   CHILD_OWNER="", CHILD_TABLE="APPLIANCE"
   P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
   FK_CONSTRAINT="R_29", FK_COLUMNS="Vendor_ID" */

IF

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

UPDATE(Vendor_ID)

BEGIN

IF EXISTS (

SELECT * FROM deleted,APPLIANCE

WHERE

/* %JoinFKPK(APPLIANCE,deleted," = "," AND") */

APPLIANCE.Vendor_ID = deleted.Vendor_ID

)

BEGIN

SELECT @errno = 30005,

       @errmsg = 'Cannot update VENDOR because APPLIANCE exists.'

GOTO error

END

END
```

CUIEEPrintFile2

```
/* ERwin Builtin Trigger */

/* VENDOR SHIPMENT on parent update no action */

/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="VENDOR"
   CHILD_OWNER="", CHILD_TABLE="SHIPMENT"
   P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
   FK_CONSTRAINT="R_23", FK_COLUMNS="Vendor_ID1" */

IF

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

UPDATE(Vendor_ID)

BEGIN

IF EXISTS (

SELECT * FROM deleted,SHIPMENT

WHERE

/* %JoinFKPK(SHIPMENT,deleted," = "," AND") */

SHIPMENT.Vendor_ID1 = deleted.Vendor_ID

)

BEGIN

SELECT @errno = 30005,

       @errmsg = 'Cannot update VENDOR because SHIPMENT exists.'

GOTO error

END

END
```

CUIEEPrintFile2

```
/* ERwin Builtin Trigger */

RETURN

error:

    RAISERROR (@errmsg, -- Message text.
               @severity, -- Severity (0~25).
               @state) -- State (0~255).

    rollback transaction

END

go


CREATE TRIGGER tD_WAREHOUSE ON WAREHOUSE FOR DELETE AS

/* ERwin Builtin Trigger */

/* DELETE trigger on WAREHOUSE */

BEGIN

    DECLARE @errno    int,
            @severity int,
            @state     int,
            @errmsg    varchar(255)

    /* ERwin Builtin Trigger */

    /* WAREHOUSE INVENTORY on parent delete no action */
```

CUIEEPrintFile2

```
/* ERWIN_RELATION:CHECKSUM="0002ecce", PARENT_OWNER="", PARENT_TABLE="WAREHOUSE"
CHILD_OWNER="", CHILD_TABLE="INVENTORY"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_55", FK_COLUMNS="Warehouse_ID1" */
IF EXISTS (
    SELECT * FROM deleted,INVENTORY
    WHERE
        /* %JoinFKPK(INVENTORY,deleted," = "," AND") */
        INVENTORY.Warehouse_ID1 = deleted.Warehouse_ID
)
BEGIN
    SELECT @errno = 30001,
           @errmsg = 'Cannot delete WAREHOUSE because INVENTORY exists.'
    GOTO error
END

/* ERwin Builtin Trigger */
/* WAREHOUSE SHIPMENT on parent delete no action */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="WAREHOUSE"
CHILD_OWNER="", CHILD_TABLE="SHIPMENT"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_24", FK_COLUMNS="Warehouse_ID1" */
IF EXISTS (
    SELECT * FROM deleted,SHIPMENT
```


CUIEEPrintFile2

```
WHERE

    /* %JoinFKPK(SHIPMENT,deleted," = "," AND") */

    SHIPMENT.Warehouse_ID1 = deleted.Warehouse_ID

)

BEGIN

    SELECT @errno = 30001,

           @errmsg = 'Cannot delete WAREHOUSE because SHIPMENT exists.'

    GOTO error

END

/* ERwin Builtin Trigger */

/* WAREHOUSE  TECHNICIAN on parent delete no action */

/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="WAREHOUSE"
CHILD_OWNER="", CHILD_TABLE="TECHNICIAN"
P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_12", FK_COLUMNS="Warehouse_ID" */

IF EXISTS (

    SELECT * FROM deleted,TECHNICIAN

    WHERE

        /* %JoinFKPK(TECHNICIAN,deleted," = "," AND") */

        TECHNICIAN.Warehouse_ID = deleted.Warehouse_ID

)

BEGIN

    SELECT @errno = 30001,
```

CUIEEPrintFile2

```
@errmsg = 'Cannot delete WAREHOUSE because TECHNICIAN exists.'
```

```
GOTO error
```

```
END
```

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

```
RETURN
```

```
error:
```

```
RAISERROR (@errmsg, -- Message text.
```

```
    @severity, -- Severity (0~25).
```

```
    @state) -- State (0~255).
```

```
rollback transaction
```

```
END
```

```
go
```

```
CREATE TRIGGER tU_WAREHOUSE ON WAREHOUSE FOR UPDATE AS
```

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

```
/* UPDATE trigger on WAREHOUSE */
```

```
BEGIN
```

```
    DECLARE @numrows int,
```

```
            @nullcnt int,
```

```
            @validcnt int,
```

CUIEEPrintFile2

```
@insWarehouse_ID integer,  
  
@errno    int,  
  
@severity int,  
  
@state    int,  
  
@errmsg   varchar(255)  
  
SELECT @numrows = @@rowcount  
  
/* ERwin Builtin Trigger */  
  
/* WAREHOUSE INVENTORY on parent update no action */  
  
/* ERWIN_RELATION:CHECKSUM="00033f47", PARENT_OWNER="", PARENT_TABLE="WAREHOUSE"  
   CHILD_OWNER="", CHILD_TABLE="INVENTORY"  
   P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",  
   FK_CONSTRAINT="R_55", FK_COLUMNS="Warehouse_ID1" */  
  
IF  
  
    /* %ParentPK(" OR",UPDATE) */  
  
    UPDATE(Warehouse_ID)  
  
BEGIN  
  
    IF EXISTS (  
  
        SELECT * FROM deleted,INVENTORY  
  
        WHERE  
  
            /* %JoinFKPK(INVENTORY,deleted," = "," AND") */  
  
            INVENTORY.Warehouse_ID1 = deleted.Warehouse_ID  
  
    )  
  
BEGIN
```

CUIEEEPrintFile2

```
SELECT @errno = 30005,

        @errmsg = 'Cannot update WAREHOUSE because INVENTORY exists.'

GOTO error

END

END

/* ERwin Builtin Trigger */

/* WAREHOUSE SHIPMENT on parent update no action */

/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="WAREHOUSE"

   CHILD_OWNER="", CHILD_TABLE="SHIPMENT"

   P2C_VERB_PHRASE="", C2P_VERB_PHRASE="",

   FK_CONSTRAINT="R_24", FK_COLUMNS="Warehouse_ID1" */

IF

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

UPDATE(Warehouse_ID)

BEGIN

IF EXISTS (

SELECT * FROM deleted,SHIPMENT

WHERE

/* %JoinFKPK(SHIPMENT,deleted," = "," AND") */

SHIPMENT.Warehouse_ID1 = deleted.Warehouse_ID

)

BEGIN

SELECT @errno = 30005,
```

CUIEEPrintFile2

```
@errmsg = 'Cannot update WAREHOUSE because SHIPMENT exists.'
```

GOTO error

END

END


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

```
/* WAREHOUSE  TECHNICIAN on parent update no action */
```

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

```
  CHILD_OWNER="", CHILD_TABLE="TECHNICIAN"
```

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

```
  FK_CONSTRAINT="R_12", FK_COLUMNS="Warehouse_ID" */
```

IF

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

```
  UPDATE(Warehouse_ID)
```

BEGIN

```
  IF EXISTS (
```

```
    SELECT * FROM deleted,TECHNICIAN
```

```
    WHERE
```

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

```
      TECHNICIAN.Warehouse_ID = deleted.Warehouse_ID
```

```
  )
```

BEGIN

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

```
         @errmsg = 'Cannot update WAREHOUSE because TECHNICIAN exists.'
```

CUIEEEPrintFile2

GOTO error

END

END

/* ERwin Builtin Trigger */

RETURN

error:

RAISERROR (@errmsg, -- Message text.

 @severity, -- Severity (0~25).

 @state) -- State (0~255).

rollback transaction

END

go