

```

CREATE TABLE EMP_MANAGEMENT
(
    EMP_ID          CHAR(18) NOT NULL ,
    EMP_NAME        CHAR(18) NULL ,
    EMP_SEX         CHAR(18) NULL ,
    EMP_AGE         CHAR(18) NULL ,
    EMP_POST_NUM    CHAR(18) NULL ,
    EMP_POST        CHAR(18) NULL ,
    EMP_TEL_NUM     CHAR(18) NULL ,
    EMP_MOB_NUM     CHAR(18) NULL ,
    EMP_RANK        CHAR(18) NULL ,
    CONSTRAINT XPK직원관리 PRIMARY KEY (EMP_ID)
);

```

```

CREATE TABLE BRAND_MANAGEMENT
(
    BRAND_NAME      CHAR(18) NOT NULL ,
    EMP_ID          CHAR(18) NOT NULL ,
    BRAND_TIME      CHAR(18) NULL ,
    BRAND_TEL_NUM   CHAR(18) NULL ,
    CONSTRAINT XPK업체관리 PRIMARY KEY (EMP_ID,BRAND_NAME),
    CONSTRAINT R_41 FOREIGN KEY (EMP_ID) REFERENCES EMP_MANAGEMENT(EMP_ID)
);

```

```

CREATE INDEX XIF1업체관리 ON BRAND_MANAGEMENT
(EMP_ID ASC);

```

```

CREATE TABLE CUS_MANAGEMENT
(
    CUS_ID          CHAR(18) NOT NULL ,
    CUS_NAME        CHAR(18) NULL ,
    CUS_SEX         CHAR(18) NULL ,
    CUS_AGE         CHAR(18) NULL ,
    CUS_POST_NUM    CHAR(18) NULL ,
    CUS_POST        CHAR(18) NULL ,
    CUS_TEL_NUM     CHAR(18) NULL ,
    CUS_MOB_NUM     CHAR(18) NULL ,
    CONSTRAINT XPK고객관리 PRIMARY KEY (CUS_ID)
);

```

```

CREATE TABLE PRODUCT_MANAGEMENT
(
    GOODS_CODE      CHAR(18) NOT NULL ,
    GOODS_SEP       CHAR(18) NULL ,
    GOODS_NAME      CHAR(18) NULL ,
    GOODS_TRUE_PRICE CHAR(18) NULL ,
    GOODS_PRICE     CHAR(18) NULL ,
    GOODS_DATE      CHAR(18) NULL ,
    GOODS_SELL_SEP  CHAR(18) NULL ,
    EMP_ID          CHAR(18) NOT NULL ,
    CONSTRAINT XPK재고관리 PRIMARY KEY (EMP_ID,GOODS_CODE),
    CONSTRAINT R_43 FOREIGN KEY (EMP_ID) REFERENCES EMP_MANAGEMENT(EMP_ID)
);

```

```

CREATE INDEX XIF1재고관리 ON PRODUCT_MANAGEMENT
(EMP_ID ASC);

```

```

CREATE TABLE ORDER_MANAGEMENT
(
    CUS_ID          CHAR(18) NOT NULL ,

```

```

        GOODS_CODE          CHAR(18) NOT NULL ,
        ORDER_NUM           CHAR(18) NOT NULL ,
        ORDER_AMOUNT        CHAR(18) NULL ,
        ORDER_SIZE          CHAR(18) NULL ,
        ORDER_PRICE         CHAR(18) NULL ,
        ORDER_SUM_PRICE     CHAR(18) NULL ,
        ORDER_DATE          CHAR(18) NULL ,
        ORDER_PAYMENT       CHAR(18) NULL ,
        EMP_ID              CHAR(18) NOT NULL ,
CONSTRAINT XPK주문관리 PRIMARY KEY (EMP_ID,CUS_ID,ORDER_NUM,GOODS_CODE),
CONSTRAINT R_30 FOREIGN KEY (CUS_ID) REFERENCES CUS_MANAGEMENT(CUS_ID),
CONSTRAINT R_32 FOREIGN KEY (EMP_ID,GOODS_CODE) REFERENCES PRODUCT_MANAGEMENT(EMP_ID,GOODS_CODE)
);

```

```

CREATE INDEX XIF1주문관리 ON ORDER_MANAGEMENT
(CUS_ID ASC);

```

```

CREATE INDEX XIF2주문관리 ON ORDER_MANAGEMENT
(GOODS_CODE ASC,EMP_ID ASC);

```

```

CREATE TABLE CUS_ORDER_RECODE
(
        CUS_PAST_NUM          CHAR(18) NOT NULL ,
        CUS_PAST_CODE        CHAR(18) NULL ,
        CUS_PAST_PRICE       CHAR(18) NULL ,
        CUS_PAST_DATE        CHAR(18) NULL ,
        CUS_ID              CHAR(18) NOT NULL ,
CONSTRAINT XPK회원주문기록 PRIMARY KEY (CUS_ID,CUS_PAST_NUM),
CONSTRAINT R_34 FOREIGN KEY (CUS_ID) REFERENCES CUS_MANAGEMENT(CUS_ID)
);

```

```

CREATE INDEX XIF1회원주문기록 ON CUS_ORDER_RECODE
(CUS_ID ASC);

```

```

CREATE TABLE PRODUCT_SIZE_AMOUNT
(
        GOODS_SIZE          CHAR(18) NOT NULL ,
        GOODS_AMOUNT        CHAR(18) NULL ,
        GOODS_CODE          CHAR(18) NOT NULL ,
        EMP_ID              CHAR(18) NOT NULL ,
CONSTRAINT XPK제품별_사이즈_별_수량 PRIMARY KEY (EMP_ID,GOODS_CODE,GOODS_SIZE),
CONSTRAINT R_38 FOREIGN KEY (EMP_ID,GOODS_CODE) REFERENCES PRODUCT_MANAGEMENT(EMP_ID,GOODS_CODE)
);

```

```

CREATE INDEX XIF1제품별_사이즈_별_수량 ON PRODUCT_SIZE_AMOUNT
(GOODS_CODE ASC,EMP_ID ASC);

```

```

CREATE TABLE CUS_GRADE_BENEFIT
(
        CUS_TOTAL_ORDER     CHAR(18) NULL ,
        CUS_RANK            CHAR(18) NOT NULL ,
        CUS_ID              CHAR(18) NOT NULL ,
        CUS_COUPON          CHAR(18) NULL ,
        CUS_SAVE_MONEY      CHAR(18) NULL ,
CONSTRAINT XPK고객_개인_등급 PRIMARY KEY (CUS_ID,CUS_RANK),
CONSTRAINT R_35 FOREIGN KEY (CUS_ID) REFERENCES CUS_MANAGEMENT(CUS_ID)
);

```

```
CREATE INDEX XIF1고객_개인_등급 ON CUS_GRADE_BENEFIT
(CUS_ID ASC);
```

```
CREATE TABLE EMP_ATTITUDE
(
    EMP_ATT          CHAR(18) NOT NULL ,
    EMP_ID           CHAR(18) NOT NULL ,
    EMP_LATE         CHAR(18) NULL ,
    EMP_LEAVE        CHAR(18) NULL ,
    EMP_ABSENT       CHAR(18) NULL ,
    CONSTRAINT XPK직원근무태도 PRIMARY KEY (EMP_ID,EMP_ATT),
    CONSTRAINT R_16 FOREIGN KEY (EMP_ID) REFERENCES EMP_MANAGEMENT(EMP_ID)
);
```

```
CREATE INDEX XIF1직원근무태도 ON EMP_ATTITUDE
(EMP_ID ASC);
```

```
CREATE TABLE EMP_SCHEDULE
(
    EMP_TIME         CHAR(18) NOT NULL ,
    EMP_REST         CHAR(18) NULL ,
    EMP_MONEY        CHAR(18) NULL ,
    EMP_ID           CHAR(18) NOT NULL ,
    CONSTRAINT XPK직원스케줄 PRIMARY KEY (EMP_ID,EMP_TIME),
    CONSTRAINT R_12 FOREIGN KEY (EMP_ID) REFERENCES EMP_MANAGEMENT(EMP_ID)
);
```

```
CREATE INDEX XIF1직원스케줄 ON EMP_SCHEDULE
(EMP_ID ASC);
```

```
CREATE TRIGGER td_EMP_MANAGEMENT AFTER DELETE ON EMP_MANAGEMENT for each row
-- ERwin Builtin 2017년 11월 16일 목요일 오후 10:12:30
-- DELETE trigger on EMP_MANAGEMENT
DECLARE NUMROWS INTEGER;
BEGIN
    /* ERwin Builtin 2017년 11월 16일 목요일 오후 10:12:30 */
    /* EMP_MANAGEMENT EMP_SCHEDULE on parent delete restrict */
    /* ERWIN_RELATION:CHECKSUM="0004351e", PARENT_OWNER="", PARENT_TABLE="EMP_MANAGEMENT"
    CHILD_OWNER="", CHILD_TABLE="EMP_SCHEDULE"
    P2C_VERB_PHRASE="R/12", C2P_VERB_PHRASE="",
    FK_CONSTRAINT="R_12", FK_COLUMNS="EMP_ID" */
    SELECT count(*) INTO NUMROWS
    FROM EMP_SCHEDULE
    WHERE
        /* %JoinFKPK(EMP_SCHEDULE,:%old," = "," AND") */
        EMP_SCHEDULE.EMP_ID = :old.EMP_ID;
    IF (NUMROWS > 0)
    THEN
        raise_application_error(
            -20001,
            'Cannot delete EMP_MANAGEMENT because EMP_SCHEDULE exists.'
        );
    END IF;

    /* ERwin Builtin 2017년 11월 16일 목요일 오후 10:12:30 */
    /* EMP_MANAGEMENT EMP_ATTITUDE on parent delete restrict */
    /* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="EMP_MANAGEMENT"
    CHILD_OWNER="", CHILD_TABLE="EMP_ATTITUDE"
    P2C_VERB_PHRASE="R/16", C2P_VERB_PHRASE="",
    FK_CONSTRAINT="R_16", FK_COLUMNS="EMP_ID" */
    SELECT count(*) INTO NUMROWS
    FROM EMP_ATTITUDE
```

```

WHERE
  /* %JoinFKPK(EMP_ATTITUDE,:%old," = "," AND") */
  EMP_ATTITUDE.EMP_ID = :old.EMP_ID;
IF (NUMROWS > 0)
THEN
  raise_application_error(
    -20001,
    'Cannot delete EMP_MANAGEMENT because EMP_ATTITUDE exists.'
  );
END IF;

/* Erwin Builtin 2017년 11월 16일 목요일 오후 10:12:30 */
/* EMP_MANAGEMENT BRAND_MANAGEMENT on parent delete restrict */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="EMP_MANAGEMENT"
CHILD_OWNER="", CHILD_TABLE="BRAND_MANAGEMENT"
P2C_VERB_PHRASE="R/41", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_41", FK_COLUMNS="EMP_ID" */
SELECT count(*) INTO NUMROWS
FROM BRAND_MANAGEMENT
WHERE
  /* %JoinFKPK(BRAND_MANAGEMENT,:%old," = "," AND") */
  BRAND_MANAGEMENT.EMP_ID = :old.EMP_ID;
IF (NUMROWS > 0)
THEN
  raise_application_error(
    -20001,
    'Cannot delete EMP_MANAGEMENT because BRAND_MANAGEMENT exists.'
  );
END IF;

/* Erwin Builtin 2017년 11월 16일 목요일 오후 10:12:30 */
/* EMP_MANAGEMENT PRODUCT_MANAGEMENT on parent delete restrict */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="EMP_MANAGEMENT"
CHILD_OWNER="", CHILD_TABLE="PRODUCT_MANAGEMENT"
P2C_VERB_PHRASE="R/43", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_43", FK_COLUMNS="EMP_ID" */
SELECT count(*) INTO NUMROWS
FROM PRODUCT_MANAGEMENT
WHERE
  /* %JoinFKPK(PRODUCT_MANAGEMENT,:%old," = "," AND") */
  PRODUCT_MANAGEMENT.EMP_ID = :old.EMP_ID;
IF (NUMROWS > 0)
THEN
  raise_application_error(
    -20001,
    'Cannot delete EMP_MANAGEMENT because PRODUCT_MANAGEMENT exists.'
  );
END IF;

```

```
-- Erwin Builtin 2017년 11월 16일 목요일 오후 10:12:30
```

```
END;
```

```
/
```

```
CREATE TRIGGER tu_EMP_MANAGEMENT AFTER UPDATE ON EMP_MANAGEMENT for each row
```

```
-- Erwin Builtin 2017년 11월 16일 목요일 오후 10:12:30
```

```
-- UPDATE trigger on EMP_MANAGEMENT
```

```
DECLARE NUMROWS INTEGER;
```

```
BEGIN
```

```
  /* Erwin Builtin 2017년 11월 16일 목요일 오후 10:12:30 */
```

```
  /* EMP_MANAGEMENT EMP_SCHEDULE on parent update restrict */
```

```
  /* ERWIN_RELATION:CHECKSUM="0004d80d", PARENT_OWNER="", PARENT_TABLE="EMP_MANAGEMENT"
```

```
  CHILD_OWNER="", CHILD_TABLE="EMP_SCHEDULE"
```

```
  P2C_VERB_PHRASE="R/12", C2P_VERB_PHRASE="",
```

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

```
  IF
```

```
    /* %JoinPKPK(:%old,:%New," <> "," OR ") */
```

```
    :old.EMP_ID <> :new.EMP_ID
```

```
  THEN
```

```
    SELECT count(*) INTO NUMROWS
```

```
    FROM EMP_SCHEDULE
```

```
    WHERE
```

```
      /* %JoinFKPK(EMP_SCHEDULE,:%old," = "," AND") */
```

```

EMP_SCHEDULE.EMP_ID = :old.EMP_ID;
IF (NUMROWS > 0)
THEN
    raise_application_error(
        -20005,
        'Cannot update EMP_MANAGEMENT because EMP_SCHEDULE exists.'
    );
END IF;
END IF;

/* ERwin Bultin 2017년 11월 16일 목요일 오후 10:12:30 */
/* EMP_MANAGEMENT EMP_ATTITUDE on parent update restrict */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="EMP_MANAGEMENT"
CHILD_OWNER="", CHILD_TABLE="EMP_ATTITUDE"
P2C_VERB_PHRASE="R/16", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_16", FK_COLUMNS="EMP_ID" */
IF
/* %JoinPKPK(:%Old,:%New," <> "," OR ") */
:old.EMP_ID <> :new.EMP_ID
THEN
    SELECT count(*) INTO NUMROWS
    FROM EMP_ATTITUDE
    WHERE
        /* %JoinFKPK(EMP_ATTITUDE,:%Old," = "," AND") */
        EMP_ATTITUDE.EMP_ID = :old.EMP_ID;
    IF (NUMROWS > 0)
    THEN
        raise_application_error(
            -20005,
            'Cannot update EMP_MANAGEMENT because EMP_ATTITUDE exists.'
        );
    END IF;
END IF;

/* ERwin Bultin 2017년 11월 16일 목요일 오후 10:12:30 */
/* EMP_MANAGEMENT BRAND_MANAGEMENT on parent update restrict */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="EMP_MANAGEMENT"
CHILD_OWNER="", CHILD_TABLE="BRAND_MANAGEMENT"
P2C_VERB_PHRASE="R/41", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_41", FK_COLUMNS="EMP_ID" */
IF
/* %JoinPKPK(:%Old,:%New," <> "," OR ") */
:old.EMP_ID <> :new.EMP_ID
THEN
    SELECT count(*) INTO NUMROWS
    FROM BRAND_MANAGEMENT
    WHERE
        /* %JoinFKPK(BRAND_MANAGEMENT,:%Old," = "," AND") */
        BRAND_MANAGEMENT.EMP_ID = :old.EMP_ID;
    IF (NUMROWS > 0)
    THEN
        raise_application_error(
            -20005,
            'Cannot update EMP_MANAGEMENT because BRAND_MANAGEMENT exists.'
        );
    END IF;
END IF;

/* ERwin Bultin 2017년 11월 16일 목요일 오후 10:12:30 */
/* EMP_MANAGEMENT PRODUCT_MANAGEMENT on parent update restrict */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="EMP_MANAGEMENT"
CHILD_OWNER="", CHILD_TABLE="PRODUCT_MANAGEMENT"
P2C_VERB_PHRASE="R/43", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_43", FK_COLUMNS="EMP_ID" */
IF
/* %JoinPKPK(:%Old,:%New," <> "," OR ") */
:old.EMP_ID <> :new.EMP_ID
THEN
    SELECT count(*) INTO NUMROWS
    FROM PRODUCT_MANAGEMENT
    WHERE
        /* %JoinFKPK(PRODUCT_MANAGEMENT,:%Old," = "," AND") */
        PRODUCT_MANAGEMENT.EMP_ID = :old.EMP_ID;

```

```

IF (NUMROWS > 0)
THEN
    raise_application_error(
        -20005,
        'Cannot update EMP_MANAGEMENT because PRODUCT_MANAGEMENT exists.'
    );
END IF;
END IF;

```

```

-- ERwin Builtin 2017년 11월 16일 목요일 오후 10:12:30
END;
/

```

```

CREATE TRIGGER tI_BRAND_MANAGEMENT BEFORE INSERT ON BRAND_MANAGEMENT for each row
-- ERwin Builtin 2017년 11월 16일 목요일 오후 10:12:30
-- INSERT trigger on BRAND_MANAGEMENT
DECLARE NUMROWS INTEGER;
BEGIN
    /* ERwin Builtin 2017년 11월 16일 목요일 오후 10:12:30 */
    /* EMP_MANAGEMENT BRAND_MANAGEMENT on child insert restrict */
    /* ERWIN_RELATION:CHECKSUM="0001ladc", PARENT_OWNER="", PARENT_TABLE="EMP_MANAGEMENT"
    CHILD_OWNER="", CHILD_TABLE="BRAND_MANAGEMENT"
    P2C_VERB_PHRASE="R/41", C2P_VERB_PHRASE="",
    FK_CONSTRAINT="R_41", FK_COLUMNS="EMP_ID" */
    SELECT count(*) INTO NUMROWS
    FROM EMP_MANAGEMENT
    WHERE
        /* %JoinFKPK(:%New,EMP_MANAGEMENT," = "," AND") */
        :new.EMP_ID = EMP_MANAGEMENT.EMP_ID;
    IF (
        /* %NotNullFK(:%New," IS NOT NULL AND") */

        NUMROWS = 0
    )
    THEN
        raise_application_error(
            -20002,
            'Cannot insert BRAND_MANAGEMENT because EMP_MANAGEMENT does not exist.'
        );
    END IF;

```

```

-- ERwin Builtin 2017년 11월 16일 목요일 오후 10:12:30
END;
/

```

```

CREATE TRIGGER tU_BRAND_MANAGEMENT AFTER UPDATE ON BRAND_MANAGEMENT for each row
-- ERwin Builtin 2017년 11월 16일 목요일 오후 10:12:30
-- UPDATE trigger on BRAND_MANAGEMENT
DECLARE NUMROWS INTEGER;
BEGIN
    /* ERwin Builtin 2017년 11월 16일 목요일 오후 10:12:30 */
    /* EMP_MANAGEMENT BRAND_MANAGEMENT on child update restrict */
    /* ERWIN_RELATION:CHECKSUM="000118af", PARENT_OWNER="", PARENT_TABLE="EMP_MANAGEMENT"
    CHILD_OWNER="", CHILD_TABLE="BRAND_MANAGEMENT"
    P2C_VERB_PHRASE="R/41", C2P_VERB_PHRASE="",
    FK_CONSTRAINT="R_41", FK_COLUMNS="EMP_ID" */
    SELECT count(*) INTO NUMROWS
    FROM EMP_MANAGEMENT
    WHERE
        /* %JoinFKPK(:%New,EMP_MANAGEMENT," = "," AND") */
        :new.EMP_ID = EMP_MANAGEMENT.EMP_ID;
    IF (
        /* %NotNullFK(:%New," IS NOT NULL AND") */

        NUMROWS = 0
    )
    THEN
        raise_application_error(
            -20007,
            'Cannot update BRAND_MANAGEMENT because EMP_MANAGEMENT does not exist.'
        );
    END IF;

```

```

);
END IF;

-- ERwin Bultin 2017년 11월 16일 목요일 오후 10:12:30
END;
/

CREATE TRIGGER td_CUS_MANAGEMENT AFTER DELETE ON CUS_MANAGEMENT for each row
-- ERwin Bultin 2017년 11월 16일 목요일 오후 10:12:30
-- DELETE trigger on CUS_MANAGEMENT
DECLARE NUMROWS INTEGER;
BEGIN
/* ERwin Bultin 2017년 11월 16일 목요일 오후 10:12:30 */
/* CUS_MANAGEMENT ORDER_MANAGEMENT on parent delete restrict */
/* ERWIN_RELATION:CHECKSUM="00032068", PARENT_OWNER="", PARENT_TABLE="CUS_MANAGEMENT"
CHILD_OWNER="", CHILD_TABLE="ORDER_MANAGEMENT"
P2C_VERB_PHRASE="R/30", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_30", FK_COLUMNS="CUS_ID" */
SELECT count(*) INTO NUMROWS
FROM ORDER_MANAGEMENT
WHERE
/* %JoinFKPK(ORDER_MANAGEMENT,:%old," = "," AND") */
ORDER_MANAGEMENT.CUS_ID = :old.CUS_ID;
IF (NUMROWS > 0)
THEN
raise_application_error(
-20001,
'Cannot delete CUS_MANAGEMENT because ORDER_MANAGEMENT exists.'
);
END IF;

/* ERwin Bultin 2017년 11월 16일 목요일 오후 10:12:30 */
/* CUS_MANAGEMENT CUS_ORDER_RECODE on parent delete restrict */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="CUS_MANAGEMENT"
CHILD_OWNER="", CHILD_TABLE="CUS_ORDER_RECODE"
P2C_VERB_PHRASE="R/34", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_34", FK_COLUMNS="CUS_ID" */
SELECT count(*) INTO NUMROWS
FROM CUS_ORDER_RECODE
WHERE
/* %JoinFKPK(CUS_ORDER_RECODE,:%old," = "," AND") */
CUS_ORDER_RECODE.CUS_ID = :old.CUS_ID;
IF (NUMROWS > 0)
THEN
raise_application_error(
-20001,
'Cannot delete CUS_MANAGEMENT because CUS_ORDER_RECODE exists.'
);
END IF;

/* ERwin Bultin 2017년 11월 16일 목요일 오후 10:12:30 */
/* CUS_MANAGEMENT CUS_GRADE_BENEFIT on parent delete restrict */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="CUS_MANAGEMENT"
CHILD_OWNER="", CHILD_TABLE="CUS_GRADE_BENEFIT"
P2C_VERB_PHRASE="R/35", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_35", FK_COLUMNS="CUS_ID" */
SELECT count(*) INTO NUMROWS
FROM CUS_GRADE_BENEFIT
WHERE
/* %JoinFKPK(CUS_GRADE_BENEFIT,:%old," = "," AND") */
CUS_GRADE_BENEFIT.CUS_ID = :old.CUS_ID;
IF (NUMROWS > 0)
THEN
raise_application_error(
-20001,
'Cannot delete CUS_MANAGEMENT because CUS_GRADE_BENEFIT exists.'
);
END IF;

-- ERwin Bultin 2017년 11월 16일 목요일 오후 10:12:30

```

```

END;
/

CREATE TRIGGER tu_CUS_MANAGEMENT AFTER UPDATE ON CUS_MANAGEMENT for each row
-- ERwin Builtin 2017년 11월 16일 목요일 오후 10:12:30
-- UPDATE trigger on CUS_MANAGEMENT
DECLARE NUMROWS INTEGER;
BEGIN
  /* ERwin Builtin 2017년 11월 16일 목요일 오후 10:12:30 */
  /* CUS_MANAGEMENT ORDER_MANAGEMENT on parent update restrict */
  /* ERWIN_RELATION:CHECKSUM="0003a026", PARENT_OWNER="", PARENT_TABLE="CUS_MANAGEMENT"
  CHILD_OWNER="", CHILD_TABLE="ORDER_MANAGEMENT"
  P2C_VERB_PHRASE="R/30", C2P_VERB_PHRASE="",
  FK_CONSTRAINT="R_30", FK_COLUMNS="CUS_ID" */
  IF
    /* %JoinPKPK(:%Old,:%New," <> "," OR ") */
    :old.CUS_ID <> :new.CUS_ID
  THEN
    SELECT count(*) INTO NUMROWS
    FROM ORDER_MANAGEMENT
    WHERE
      /* %JoinFKPK(ORDER_MANAGEMENT,:%Old," = "," AND") */
      ORDER_MANAGEMENT.CUS_ID = :old.CUS_ID;
    IF (NUMROWS > 0)
    THEN
      raise_application_error(
        -20005,
        'Cannot update CUS_MANAGEMENT because ORDER_MANAGEMENT exists.'
      );
    END IF;
  END IF;

  /* ERwin Builtin 2017년 11월 16일 목요일 오후 10:12:30 */
  /* CUS_MANAGEMENT CUS_ORDER_RECODE on parent update restrict */
  /* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="CUS_MANAGEMENT"
  CHILD_OWNER="", CHILD_TABLE="CUS_ORDER_RECODE"
  P2C_VERB_PHRASE="R/34", C2P_VERB_PHRASE="",
  FK_CONSTRAINT="R_34", FK_COLUMNS="CUS_ID" */
  IF
    /* %JoinPKPK(:%Old,:%New," <> "," OR ") */
    :old.CUS_ID <> :new.CUS_ID
  THEN
    SELECT count(*) INTO NUMROWS
    FROM CUS_ORDER_RECODE
    WHERE
      /* %JoinFKPK(CUS_ORDER_RECODE,:%Old," = "," AND") */
      CUS_ORDER_RECODE.CUS_ID = :old.CUS_ID;
    IF (NUMROWS > 0)
    THEN
      raise_application_error(
        -20005,
        'Cannot update CUS_MANAGEMENT because CUS_ORDER_RECODE exists.'
      );
    END IF;
  END IF;

  /* ERwin Builtin 2017년 11월 16일 목요일 오후 10:12:30 */
  /* CUS_MANAGEMENT CUS_GRADE_BENEFIT on parent update restrict */
  /* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="CUS_MANAGEMENT"
  CHILD_OWNER="", CHILD_TABLE="CUS_GRADE_BENEFIT"
  P2C_VERB_PHRASE="R/35", C2P_VERB_PHRASE="",
  FK_CONSTRAINT="R_35", FK_COLUMNS="CUS_ID" */
  IF
    /* %JoinPKPK(:%Old,:%New," <> "," OR ") */
    :old.CUS_ID <> :new.CUS_ID
  THEN
    SELECT count(*) INTO NUMROWS
    FROM CUS_GRADE_BENEFIT
    WHERE
      /* %JoinFKPK(CUS_GRADE_BENEFIT,:%Old," = "," AND") */
      CUS_GRADE_BENEFIT.CUS_ID = :old.CUS_ID;
    IF (NUMROWS > 0)
    THEN

```



```

        raise_application_error(
            -20005,
            'Cannot update CUS_MANAGEMENT because CUS_GRADE_BENEFIT exists.'
        );
    END IF;
END IF;

-- ERwin Builtin 2017년 11월 16일 목요일 오후 10:12:30
END;
/

CREATE TRIGGER tI_PRODUCT_MANAGEMENT BEFORE INSERT ON PRODUCT_MANAGEMENT for each row
-- ERwin Builtin 2017년 11월 16일 목요일 오후 10:12:30
-- INSERT trigger on PRODUCT_MANAGEMENT
DECLARE NUMROWS INTEGER;
BEGIN
    /* ERwin Builtin 2017년 11월 16일 목요일 오후 10:12:30 */
    /* EMP_MANAGEMENT PRODUCT_MANAGEMENT on child insert restrict */
    /* ERWIN_RELATION:CHECKSUM="000114a5", PARENT_OWNER="", PARENT_TABLE="EMP_MANAGEMENT"
    CHILD_OWNER="", CHILD_TABLE="PRODUCT_MANAGEMENT"
    P2C_VERB_PHRASE="R/43", C2P_VERB_PHRASE="",
    FK_CONSTRAINT="R_43", FK_COLUMNS="EMP_ID" */
    SELECT count(*) INTO NUMROWS
    FROM EMP_MANAGEMENT
    WHERE
        /* %JoinFKPK(:%New,EMP_MANAGEMENT," = "," AND") */
        :new.EMP_ID = EMP_MANAGEMENT.EMP_ID;
    IF (
        /* %NotNullFK(:%New," IS NOT NULL AND") */

        NUMROWS = 0
    )
    THEN
        raise_application_error(
            -20002,
            'Cannot insert PRODUCT_MANAGEMENT because EMP_MANAGEMENT does not exist.'
        );
    END IF;

-- ERwin Builtin 2017년 11월 16일 목요일 오후 10:12:30
END;
/

CREATE TRIGGER tD_PRODUCT_MANAGEMENT AFTER DELETE ON PRODUCT_MANAGEMENT for each row
-- ERwin Builtin 2017년 11월 16일 목요일 오후 10:12:30
-- DELETE trigger on PRODUCT_MANAGEMENT
DECLARE NUMROWS INTEGER;
BEGIN
    /* ERwin Builtin 2017년 11월 16일 목요일 오후 10:12:30 */
    /* PRODUCT_MANAGEMENT ORDER_MANAGEMENT on parent delete restrict */
    /* ERWIN_RELATION:CHECKSUM="00026b3b", PARENT_OWNER="", PARENT_TABLE="PRODUCT_MANAGEMENT"
    CHILD_OWNER="", CHILD_TABLE="ORDER_MANAGEMENT"
    P2C_VERB_PHRASE="R/32", C2P_VERB_PHRASE="",
    FK_CONSTRAINT="R_32", FK_COLUMNS="EMP_ID""GOODS_CODE" */
    SELECT count(*) INTO NUMROWS
    FROM ORDER_MANAGEMENT
    WHERE
        /* %JoinFKPK(ORDER_MANAGEMENT,:%Old," = "," AND") */
        ORDER_MANAGEMENT.GOODS_CODE = :old.GOODS_CODE AND
        ORDER_MANAGEMENT.EMP_ID = :old.EMP_ID;
    IF (NUMROWS > 0)
    THEN
        raise_application_error(
            -20001,
            'Cannot delete PRODUCT_MANAGEMENT because ORDER_MANAGEMENT exists.'
        );
    END IF;

    /* ERwin Builtin 2017년 11월 16일 목요일 오후 10:12:30 */
    /* PRODUCT_MANAGEMENT PRODUCT_SIZE_AMOUNT on parent delete restrict */

```

```

/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="PRODUCT_MANAGEMENT"
CHILD_OWNER="", CHILD_TABLE="PRODUCT_SIZE_AMOUNT"
P2C_VERB_PHRASE="R/38", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_38", FK_COLUMNS="EMP_ID""GOODS_CODE" */
SELECT count(*) INTO NUMROWS
FROM PRODUCT_SIZE_AMOUNT
WHERE
/* %JoinFKPK(PRODUCT_SIZE_AMOUNT,:%old," = "," AND") */
PRODUCT_SIZE_AMOUNT.GOODS_CODE = :old.GOODS_CODE AND
PRODUCT_SIZE_AMOUNT.EMP_ID = :old.EMP_ID;
IF (NUMROWS > 0)
THEN
raise_application_error(
-20001,
'Cannot delete PRODUCT_MANAGEMENT because PRODUCT_SIZE_AMOUNT exists.'
);
END IF;

```

-- ERwin Bultin 2017년 11월 16일 목요일 오후 10:12:30

END;

/

CREATE TRIGGER tu_PRODUCT_MANAGEMENT AFTER UPDATE ON PRODUCT_MANAGEMENT for each row

-- ERwin Bultin 2017년 11월 16일 목요일 오후 10:12:30

-- UPDATE trigger on PRODUCT_MANAGEMENT

DECLARE NUMROWS INTEGER;

BEGIN

/* ERwin Bultin 2017년 11월 16일 목요일 오후 10:12:30 */

/* PRODUCT_MANAGEMENT ORDER_MANAGEMENT on parent update restrict */

/* ERWIN_RELATION:CHECKSUM="000405fd", PARENT_OWNER="", PARENT_TABLE="PRODUCT_MANAGEMENT"

CHILD_OWNER="", CHILD_TABLE="ORDER_MANAGEMENT"

P2C_VERB_PHRASE="R/32", C2P_VERB_PHRASE="",

FK_CONSTRAINT="R_32", FK_COLUMNS="EMP_ID""GOODS_CODE" */

IF

/* %JoinPKPK(:%old,:%new," <> "," OR ") */

:old.GOODS_CODE <> :new.GOODS_CODE OR

:old.EMP_ID <> :new.EMP_ID

THEN

SELECT count(*) INTO NUMROWS

FROM ORDER_MANAGEMENT

WHERE

/* %JoinFKPK(ORDER_MANAGEMENT,:%old," = "," AND") */

ORDER_MANAGEMENT.GOODS_CODE = :old.GOODS_CODE AND

ORDER_MANAGEMENT.EMP_ID = :old.EMP_ID;

IF (NUMROWS > 0)

THEN

raise_application_error(

-20005,

'Cannot update PRODUCT_MANAGEMENT because ORDER_MANAGEMENT exists.'

);

END IF;

END IF;

/* ERwin Bultin 2017년 11월 16일 목요일 오후 10:12:30 */

/* PRODUCT_MANAGEMENT PRODUCT_SIZE_AMOUNT on parent update restrict */

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

CHILD_OWNER="", CHILD_TABLE="PRODUCT_SIZE_AMOUNT"

P2C_VERB_PHRASE="R/38", C2P_VERB_PHRASE="",

FK_CONSTRAINT="R_38", FK_COLUMNS="EMP_ID""GOODS_CODE" */

IF

/* %JoinPKPK(:%old,:%new," <> "," OR ") */

:old.GOODS_CODE <> :new.GOODS_CODE OR

:old.EMP_ID <> :new.EMP_ID

THEN

SELECT count(*) INTO NUMROWS

FROM PRODUCT_SIZE_AMOUNT

WHERE

/* %JoinFKPK(PRODUCT_SIZE_AMOUNT,:%old," = "," AND") */

PRODUCT_SIZE_AMOUNT.GOODS_CODE = :old.GOODS_CODE AND

PRODUCT_SIZE_AMOUNT.EMP_ID = :old.EMP_ID;

IF (NUMROWS > 0)

THEN

```

        raise_application_error(
            -20005,
            'Cannot update PRODUCT_MANAGEMENT because PRODUCT_SIZE_AMOUNT exists.'
        );
    END IF;
END IF;

/* Erwin Builtin 2017년 11월 16일 목요일 오후 10:12:30 */
/* EMP_MANAGEMENT PRODUCT_MANAGEMENT on child update restrict */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="EMP_MANAGEMENT"
   CHILD_OWNER="", CHILD_TABLE="PRODUCT_MANAGEMENT"
   P2C_VERB_PHRASE="R/43", C2P_VERB_PHRASE="",
   FK_CONSTRAINT="R_43", FK_COLUMNS="EMP_ID" */
SELECT count(*) INTO NUMROWS
FROM EMP_MANAGEMENT
WHERE
    /* %JoinFKPK(:%New,EMP_MANAGEMENT," = "," AND") */
    :new.EMP_ID = EMP_MANAGEMENT.EMP_ID;
IF (
    /* %NotnullFK(:%New," IS NOT NULL AND") */

    NUMROWS = 0
)
THEN
    raise_application_error(
        -20007,
        'Cannot update PRODUCT_MANAGEMENT because EMP_MANAGEMENT does not exist.'
    );
END IF;

```

```

-- Erwin Builtin 2017년 11월 16일 목요일 오후 10:12:30
END;
/

```

```

CREATE TRIGGER tI_ORDER_MANAGEMENT BEFORE INSERT ON ORDER_MANAGEMENT for each row
-- Erwin Builtin 2017년 11월 16일 목요일 오후 10:12:30
-- INSERT trigger on ORDER_MANAGEMENT
DECLARE NUMROWS INTEGER;
BEGIN
    /* Erwin Builtin 2017년 11월 16일 목요일 오후 10:12:30 */
    /* CUS_MANAGEMENT ORDER_MANAGEMENT on child insert restrict */
    /* ERWIN_RELATION:CHECKSUM="00026216", PARENT_OWNER="", PARENT_TABLE="CUS_MANAGEMENT"
       CHILD_OWNER="", CHILD_TABLE="ORDER_MANAGEMENT"
       P2C_VERB_PHRASE="R/30", C2P_VERB_PHRASE="",
       FK_CONSTRAINT="R_30", FK_COLUMNS="CUS_ID" */
    SELECT count(*) INTO NUMROWS
    FROM CUS_MANAGEMENT
    WHERE
        /* %JoinFKPK(:%New,CUS_MANAGEMENT," = "," AND") */
        :new.CUS_ID = CUS_MANAGEMENT.CUS_ID;
    IF (
        /* %NotnullFK(:%New," IS NOT NULL AND") */

        NUMROWS = 0
    )
    THEN
        raise_application_error(
            -20002,
            'Cannot insert ORDER_MANAGEMENT because CUS_MANAGEMENT does not exist.'
        );
    END IF;

    /* Erwin Builtin 2017년 11월 16일 목요일 오후 10:12:30 */
    /* PRODUCT_MANAGEMENT ORDER_MANAGEMENT on child insert restrict */
    /* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="PRODUCT_MANAGEMENT"
       CHILD_OWNER="", CHILD_TABLE="ORDER_MANAGEMENT"
       P2C_VERB_PHRASE="R/32", C2P_VERB_PHRASE="",
       FK_CONSTRAINT="R_32", FK_COLUMNS="EMP_ID""GOODS_CODE" */
    SELECT count(*) INTO NUMROWS
    FROM PRODUCT_MANAGEMENT
    WHERE

```

```

        /* %JoinFKPK(:%New,PRODUCT_MANAGEMENT," = "," AND") */
        :new.GOODS_CODE = PRODUCT_MANAGEMENT.GOODS_CODE AND
        :new.EMP_ID = PRODUCT_MANAGEMENT.EMP_ID;
IF (
    /* %NotNullFK(:%New," IS NOT NULL AND") */

    NUMROWS = 0
)
THEN
    raise_application_error(
        -20002,
        'Cannot insert ORDER_MANAGEMENT because PRODUCT_MANAGEMENT does not exist.'
    );
END IF;

-- ERwin Builtin 2017년 11월 16일 목요일 오후 10:12:30
END;
/

CREATE TRIGGER tu_ORDER_MANAGEMENT AFTER UPDATE ON ORDER_MANAGEMENT for each row
-- ERwin Builtin 2017년 11월 16일 목요일 오후 10:12:30
-- UPDATE trigger on ORDER_MANAGEMENT
DECLARE NUMROWS INTEGER;
BEGIN
    /* ERwin Builtin 2017년 11월 16일 목요일 오후 10:12:30 */
    /* CUS_MANAGEMENT ORDER_MANAGEMENT on child update restrict */
    /* ERWIN_RELATION:CHECKSUM="00025c44", PARENT_OWNER="", PARENT_TABLE="CUS_MANAGEMENT"
    CHILD_OWNER="", CHILD_TABLE="ORDER_MANAGEMENT"
    P2C_VERB_PHRASE="R/30", C2P_VERB_PHRASE="",
    FK_CONSTRAINT="R_30", FK_COLUMNS="CUS_ID" */
    SELECT count(*) INTO NUMROWS
    FROM CUS_MANAGEMENT
    WHERE
        /* %JoinFKPK(:%New,CUS_MANAGEMENT," = "," AND") */
        :new.CUS_ID = CUS_MANAGEMENT.CUS_ID;
IF (
    /* %NotNullFK(:%New," IS NOT NULL AND") */

    NUMROWS = 0
)
THEN
    raise_application_error(
        -20007,
        'Cannot update ORDER_MANAGEMENT because CUS_MANAGEMENT does not exist.'
    );
END IF;

/* ERwin Builtin 2017년 11월 16일 목요일 오후 10:12:30 */
/* PRODUCT_MANAGEMENT ORDER_MANAGEMENT on child update restrict */
/* ERWIN_RELATION:CHECKSUM="00000000", PARENT_OWNER="", PARENT_TABLE="PRODUCT_MANAGEMENT"
CHILD_OWNER="", CHILD_TABLE="ORDER_MANAGEMENT"
P2C_VERB_PHRASE="R/32", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_32", FK_COLUMNS="EMP_ID"GOODS_CODE" */
SELECT count(*) INTO NUMROWS
FROM PRODUCT_MANAGEMENT
WHERE
    /* %JoinFKPK(:%New,PRODUCT_MANAGEMENT," = "," AND") */
    :new.GOODS_CODE = PRODUCT_MANAGEMENT.GOODS_CODE AND
    :new.EMP_ID = PRODUCT_MANAGEMENT.EMP_ID;
IF (
    /* %NotNullFK(:%New," IS NOT NULL AND") */

    NUMROWS = 0
)
THEN
    raise_application_error(
        -20007,
        'Cannot update ORDER_MANAGEMENT because PRODUCT_MANAGEMENT does not exist.'
    );
END IF;

```

```
-- ERwin Builtin 2017년 11월 16일 목요일 오후 10:12:30
END;
/
```

```
CREATE TRIGGER tI_CUS_ORDER_RECODE BEFORE INSERT ON CUS_ORDER_RECODE for each row
```

```
-- ERwin Builtin 2017년 11월 16일 목요일 오후 10:12:30
```

```
-- INSERT trigger on CUS_ORDER_RECODE
```

```
DECLARE NUMROWS INTEGER;
```

```
BEGIN
```

```
/* ERwin Builtin 2017년 11월 16일 목요일 오후 10:12:30 */
```

```
/* CUS_MANAGEMENT CUS_ORDER_RECODE on child insert restrict */
```

```
/* ERWIN_RELATION:CHECKSUM="00010996", PARENT_OWNER="", PARENT_TABLE="CUS_MANAGEMENT"
```

```
CHILD_OWNER="", CHILD_TABLE="CUS_ORDER_RECODE"
```

```
P2C_VERB_PHRASE="R/34", C2P_VERB_PHRASE="",
```

```
FK_CONSTRAINT="R_34", FK_COLUMNS="CUS_ID" */
```

```
SELECT count(*) INTO NUMROWS
```

```
FROM CUS_MANAGEMENT
```

```
WHERE
```

```
/* %JoinFKPK(:%New,CUS_MANAGEMENT," = "," AND") */
```

```
:new.CUS_ID = CUS_MANAGEMENT.CUS_ID;
```

```
IF (
```

```
/* %NotnullFK(:%New," IS NOT NULL AND") */
```

```
NUMROWS = 0
```

```
)
```

```
THEN
```

```
raise_application_error(
```

```
-20002,
```

```
'Cannot insert CUS_ORDER_RECODE because CUS_MANAGEMENT does not exist.'
```

```
);
```

```
END IF;
```

```
-- ERwin Builtin 2017년 11월 16일 목요일 오후 10:12:30
```

```
END;
```

```
/
```

```
CREATE TRIGGER tU_CUS_ORDER_RECODE AFTER UPDATE ON CUS_ORDER_RECODE for each row
```

```
-- ERwin Builtin 2017년 11월 16일 목요일 오후 10:12:30
```

```
-- UPDATE trigger on CUS_ORDER_RECODE
```

```
DECLARE NUMROWS INTEGER;
```

```
BEGIN
```

```
/* ERwin Builtin 2017년 11월 16일 목요일 오후 10:12:30 */
```

```
/* CUS_MANAGEMENT CUS_ORDER_RECODE on child update restrict */
```

```
/* ERWIN_RELATION:CHECKSUM="000104ea", PARENT_OWNER="", PARENT_TABLE="CUS_MANAGEMENT"
```

```
CHILD_OWNER="", CHILD_TABLE="CUS_ORDER_RECODE"
```

```
P2C_VERB_PHRASE="R/34", C2P_VERB_PHRASE="",
```

```
FK_CONSTRAINT="R_34", FK_COLUMNS="CUS_ID" */
```

```
SELECT count(*) INTO NUMROWS
```

```
FROM CUS_MANAGEMENT
```

```
WHERE
```

```
/* %JoinFKPK(:%New,CUS_MANAGEMENT," = "," AND") */
```

```
:new.CUS_ID = CUS_MANAGEMENT.CUS_ID;
```

```
IF (
```

```
/* %NotnullFK(:%New," IS NOT NULL AND") */
```

```
NUMROWS = 0
```

```
)
```

```
THEN
```

```
raise_application_error(
```

```
-20007,
```

```
'Cannot update CUS_ORDER_RECODE because CUS_MANAGEMENT does not exist.'
```

```
);
```

```
END IF;
```

```
-- ERwin Builtin 2017년 11월 16일 목요일 오후 10:12:30
```

```
END;
```

```
/
```

```
CREATE TRIGGER tI_PRODUCT_SIZE_AMOUNT BEFORE INSERT ON PRODUCT_SIZE_AMOUNT for each row
```

```
-- ERwin Builtin 2017년 11월 16일 목요일 오후 10:12:30
-- INSERT trigger on PRODUCT_SIZE_AMOUNT
DECLARE NUMROWS INTEGER;
BEGIN
  /* ERwin Builtin 2017년 11월 16일 목요일 오후 10:12:30 */
  /* PRODUCT_MANAGEMENT PRODUCT_SIZE_AMOUNT on child insert restrict */
  /* ERWIN_RELATION:CHECKSUM="00012f7b", PARENT_OWNER="", PARENT_TABLE="PRODUCT_MANAGEMENT"
  CHILD_OWNER="", CHILD_TABLE="PRODUCT_SIZE_AMOUNT"
  P2C_VERB_PHRASE="R/38", C2P_VERB_PHRASE="",
  FK_CONSTRAINT="R_38", FK_COLUMNS="EMP_ID" "GOODS_CODE" */
  SELECT count(*) INTO NUMROWS
    FROM PRODUCT_MANAGEMENT
    WHERE
      /* %JoinFKPK(:%New,PRODUCT_MANAGEMENT," = "," AND") */
      :new.GOODS_CODE = PRODUCT_MANAGEMENT.GOODS_CODE AND
      :new.EMP_ID = PRODUCT_MANAGEMENT.EMP_ID;
  IF (
    /* %NotNullFK(:%New," IS NOT NULL AND") */

    NUMROWS = 0
  )
  THEN
    raise_application_error(
      -20002,
      'Cannot insert PRODUCT_SIZE_AMOUNT because PRODUCT_MANAGEMENT does not exist.'
    );
  END IF;
```

```
-- ERwin Builtin 2017년 11월 16일 목요일 오후 10:12:30
END;
/
```

```
CREATE TRIGGER tu_PRODUCT_SIZE_AMOUNT AFTER UPDATE ON PRODUCT_SIZE_AMOUNT for each row
-- ERwin Builtin 2017년 11월 16일 목요일 오후 10:12:30
-- UPDATE trigger on PRODUCT_SIZE_AMOUNT
DECLARE NUMROWS INTEGER;
BEGIN
  /* ERwin Builtin 2017년 11월 16일 목요일 오후 10:12:30 */
  /* PRODUCT_MANAGEMENT PRODUCT_SIZE_AMOUNT on child update restrict */
  /* ERWIN_RELATION:CHECKSUM="00012dbf", PARENT_OWNER="", PARENT_TABLE="PRODUCT_MANAGEMENT"
  CHILD_OWNER="", CHILD_TABLE="PRODUCT_SIZE_AMOUNT"
  P2C_VERB_PHRASE="R/38", C2P_VERB_PHRASE="",
  FK_CONSTRAINT="R_38", FK_COLUMNS="EMP_ID" "GOODS_CODE" */
  SELECT count(*) INTO NUMROWS
    FROM PRODUCT_MANAGEMENT
    WHERE
      /* %JoinFKPK(:%New,PRODUCT_MANAGEMENT," = "," AND") */
      :new.GOODS_CODE = PRODUCT_MANAGEMENT.GOODS_CODE AND
      :new.EMP_ID = PRODUCT_MANAGEMENT.EMP_ID;
  IF (
    /* %NotNullFK(:%New," IS NOT NULL AND") */

    NUMROWS = 0
  )
  THEN
    raise_application_error(
      -20007,
      'Cannot update PRODUCT_SIZE_AMOUNT because PRODUCT_MANAGEMENT does not exist.'
    );
  END IF;
```

```
-- ERwin Builtin 2017년 11월 16일 목요일 오후 10:12:30
END;
/
```

```
CREATE TRIGGER ti_CUS_GRADE_BENEFIT BEFORE INSERT ON CUS_GRADE_BENEFIT for each row
-- ERwin Builtin 2017년 11월 16일 목요일 오후 10:12:30
-- INSERT trigger on CUS_GRADE_BENEFIT
DECLARE NUMROWS INTEGER;
BEGIN
```

```

/* Erwin Bultin 2017년 11월 16일 목요일 오후 10:12:30 */
/* CUS_MANAGEMENT CUS_GRADE_BENEFIT on child insert restrict */
/* ERWIN_RELATION:CHECKSUM="000100fc", PARENT_OWNER="", PARENT_TABLE="CUS_MANAGEMENT"
CHILD_OWNER="", CHILD_TABLE="CUS_GRADE_BENEFIT"
P2C_VERB_PHRASE="R/35", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_35", FK_COLUMNS="CUS_ID" */
SELECT count(*) INTO NUMROWS
FROM CUS_MANAGEMENT
WHERE
/* %JoinFKPK(:%New,CUS_MANAGEMENT," = "," AND") */
:new.CUS_ID = CUS_MANAGEMENT.CUS_ID;
IF (
/* %NotNullFK(:%New," IS NOT NULL AND") */

NUMROWS = 0
)
THEN
raise_application_error(
-20002,
'Cannot insert CUS_GRADE_BENEFIT because CUS_MANAGEMENT does not exist.'
);
END IF;

```

```

-- Erwin Bultin 2017년 11월 16일 목요일 오후 10:12:30
END;
/

```

```

CREATE TRIGGER tu_CUS_GRADE_BENEFIT AFTER UPDATE ON CUS_GRADE_BENEFIT for each row
-- Erwin Bultin 2017년 11월 16일 목요일 오후 10:12:30
-- UPDATE trigger on CUS_GRADE_BENEFIT
DECLARE NUMROWS INTEGER;
BEGIN
/* Erwin Bultin 2017년 11월 16일 목요일 오후 10:12:30 */
/* CUS_MANAGEMENT CUS_GRADE_BENEFIT on child update restrict */
/* ERWIN_RELATION:CHECKSUM="00010197", PARENT_OWNER="", PARENT_TABLE="CUS_MANAGEMENT"
CHILD_OWNER="", CHILD_TABLE="CUS_GRADE_BENEFIT"
P2C_VERB_PHRASE="R/35", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_35", FK_COLUMNS="CUS_ID" */
SELECT count(*) INTO NUMROWS
FROM CUS_MANAGEMENT
WHERE
/* %JoinFKPK(:%New,CUS_MANAGEMENT," = "," AND") */
:new.CUS_ID = CUS_MANAGEMENT.CUS_ID;
IF (
/* %NotNullFK(:%New," IS NOT NULL AND") */

NUMROWS = 0
)
THEN
raise_application_error(
-20007,
'Cannot update CUS_GRADE_BENEFIT because CUS_MANAGEMENT does not exist.'
);
END IF;

```

```

-- Erwin Bultin 2017년 11월 16일 목요일 오후 10:12:30
END;
/

```

```

CREATE TRIGGER ti_EMP_ATTITUDE BEFORE INSERT ON EMP_ATTITUDE for each row
-- Erwin Bultin 2017년 11월 16일 목요일 오후 10:12:30
-- INSERT trigger on EMP_ATTITUDE
DECLARE NUMROWS INTEGER;
BEGIN
/* Erwin Bultin 2017년 11월 16일 목요일 오후 10:12:30 */
/* EMP_MANAGEMENT EMP_ATTITUDE on child insert restrict */
/* ERWIN_RELATION:CHECKSUM="0001076e", PARENT_OWNER="", PARENT_TABLE="EMP_MANAGEMENT"
CHILD_OWNER="", CHILD_TABLE="EMP_ATTITUDE"
P2C_VERB_PHRASE="R/16", C2P_VERB_PHRASE="",
FK_CONSTRAINT="R_16", FK_COLUMNS="EMP_ID" */

```

```

SELECT count(*) INTO NUMROWS
FROM EMP_MANAGEMENT
WHERE
    /* %JoinFKPK(:%New,EMP_MANAGEMENT," = "," AND") */
    :new.EMP_ID = EMP_MANAGEMENT.EMP_ID;
IF (
    /* %NotNullFK(:%New," IS NOT NULL AND") */

    NUMROWS = 0
)
THEN
    raise_application_error(
        -20002,
        'Cannot insert EMP_ATTITUDE because EMP_MANAGEMENT does not exist.'
    );
END IF;

```

```

-- ERwin Builtin 2017년 11월 16일 목요일 오후 10:12:30
END;
/

```

```

CREATE TRIGGER tu_EMP_ATTITUDE AFTER UPDATE ON EMP_ATTITUDE for each row
-- ERwin Builtin 2017년 11월 16일 목요일 오후 10:12:30
-- UPDATE trigger on EMP_ATTITUDE
DECLARE NUMROWS INTEGER;
BEGIN
    /* ERwin Builtin 2017년 11월 16일 목요일 오후 10:12:30 */
    /* EMP_MANAGEMENT EMP_ATTITUDE on child update restrict */
    /* ERWIN_RELATION:CHECKSUM="000107b1", PARENT_OWNER="", PARENT_TABLE="EMP_MANAGEMENT"
    CHILD_OWNER="", CHILD_TABLE="EMP_ATTITUDE"
    P2C_VERB_PHRASE="R/16", C2P_VERB_PHRASE="",
    FK_CONSTRAINT="R_16", FK_COLUMNS="EMP_ID" */
    SELECT count(*) INTO NUMROWS
    FROM EMP_MANAGEMENT
    WHERE
        /* %JoinFKPK(:%New,EMP_MANAGEMENT," = "," AND") */
        :new.EMP_ID = EMP_MANAGEMENT.EMP_ID;
    IF (
        /* %NotNullFK(:%New," IS NOT NULL AND") */

        NUMROWS = 0
    )
    THEN
        raise_application_error(
            -20007,
            'Cannot update EMP_ATTITUDE because EMP_MANAGEMENT does not exist.'
        );
    END IF;

```

```

-- ERwin Builtin 2017년 11월 16일 목요일 오후 10:12:30
END;
/

```

```

CREATE TRIGGER ti_EMP_SCHEDULE BEFORE INSERT ON EMP_SCHEDULE for each row
-- ERwin Builtin 2017년 11월 16일 목요일 오후 10:12:30
-- INSERT trigger on EMP_SCHEDULE
DECLARE NUMROWS INTEGER;
BEGIN
    /* ERwin Builtin 2017년 11월 16일 목요일 오후 10:12:30 */
    /* EMP_MANAGEMENT EMP_SCHEDULE on child insert restrict */
    /* ERWIN_RELATION:CHECKSUM="00010d09", PARENT_OWNER="", PARENT_TABLE="EMP_MANAGEMENT"
    CHILD_OWNER="", CHILD_TABLE="EMP_SCHEDULE"
    P2C_VERB_PHRASE="R/12", C2P_VERB_PHRASE="",
    FK_CONSTRAINT="R_12", FK_COLUMNS="EMP_ID" */
    SELECT count(*) INTO NUMROWS
    FROM EMP_MANAGEMENT
    WHERE
        /* %JoinFKPK(:%New,EMP_MANAGEMENT," = "," AND") */
        :new.EMP_ID = EMP_MANAGEMENT.EMP_ID;
    IF (

```



```

        /* %NotNullFK(:%New," IS NOT NULL AND") */

        NUMROWS = 0
    )
    THEN
        raise_application_error(
            -20002,
            'Cannot insert EMP_SCHEDULE because EMP_MANAGEMENT does not exist.'
        );
    END IF;

-- ERwin Builtin 2017년 11월 16일 목요일 오후 10:12:30
END;
/

CREATE TRIGGER tu_EMP_SCHEDULE AFTER UPDATE ON EMP_SCHEDULE for each row
-- ERwin Builtin 2017년 11월 16일 목요일 오후 10:12:30
-- UPDATE trigger on EMP_SCHEDULE
DECLARE NUMROWS INTEGER;
BEGIN
    /* ERwin Builtin 2017년 11월 16일 목요일 오후 10:12:30 */
    /* EMP_MANAGEMENT EMP_SCHEDULE on child update restrict */
    /* ERWIN_RELATION:CHECKSUM="0001085d", PARENT_OWNER="", PARENT_TABLE="EMP_MANAGEMENT"
    CHILD_OWNER="", CHILD_TABLE="EMP_SCHEDULE"
    P2C_VERB_PHRASE="R/12", C2P_VERB_PHRASE="",
    FK_CONSTRAINT="R_12", FK_COLUMNS="EMP_ID" */
    SELECT count(*) INTO NUMROWS
    FROM EMP_MANAGEMENT
    WHERE
        /* %JoinFKPK(:%New,EMP_MANAGEMENT," = "," AND") */
        :new.EMP_ID = EMP_MANAGEMENT.EMP_ID;
    IF (
        /* %NotNullFK(:%New," IS NOT NULL AND") */

        NUMROWS = 0
    )
    THEN
        raise_application_error(
            -20007,
            'Cannot update EMP_SCHEDULE because EMP_MANAGEMENT does not exist.'
        );
    END IF;

-- ERwin Builtin 2017년 11월 16일 목요일 오후 10:12:30
END;
/

```