KULLANICILAR TABLOSU

ALTER TABLE STOK.KULLANICILAR

DROP PRIMARY KEY CASCADE;

DROP TABLE STOK.KULLANICILAR CASCADE CONSTRAINTS;

CREATE TABLE STOK.KULLANICILAR

(

ID NUMBER(38),

TC\_KIMLIK\_NO VARCHAR2(11 BYTE),

ADI VARCHAR2(250 BYTE),

SOYADI VARCHAR2(250 BYTE),

ROL\_ID NUMBER(38),

KULLANICI\_ADI VARCHAR2(250 BYTE),

SIFRE VARCHAR2(250 BYTE),

DOGUM\_TARIHI DATE,

MAIL VARCHAR2(250 BYTE),

KAYIT\_DURUM NUMBER(38),

CEP\_TEL VARCHAR2(25 BYTE)

)

TABLESPACE USERS

PCTUSED 0

PCTFREE 10

INITRANS 1

MAXTRANS 255

STORAGE (

INITIAL 64K

NEXT 1M

MINEXTENTS 1

MAXEXTENTS UNLIMITED

PCTINCREASE 0

BUFFER\_POOL DEFAULT

)

LOGGING

NOCOMPRESS

NOCACHE

NOPARALLEL

MONITORING;

CREATE UNIQUE INDEX STOK.KULLANICILAR\_PK ON STOK.KULLANICILAR

(ID)

LOGGING

TABLESPACE USERS

PCTFREE 10

INITRANS 2

MAXTRANS 255

STORAGE (

INITIAL 64K

NEXT 1M

MINEXTENTS 1

MAXEXTENTS UNLIMITED

PCTINCREASE 0

BUFFER\_POOL DEFAULT

)

NOPARALLEL;

CREATE OR REPLACE TRIGGER STOK.TRG\_KULLANICILAR

BEFORE INSERT

ON STOK.KULLANICILAR

REFERENCING NEW AS New OLD AS Old

FOR EACH ROW

BEGIN

*-- BEGIN*

SELECT STOK.SQ\_KULLANICILAR.NEXTVAL INTO :NEW.ID FROM DUAL;

END;

/

ALTER TABLE STOK.KULLANICILAR ADD (

CONSTRAINT KULLANICILAR\_PK

PRIMARY KEY

(ID)

USING INDEX STOK.KULLANICILAR\_PK);

ALTER TABLE STOK.KULLANICILAR ADD (

FOREIGN KEY (ROL\_ID)

REFERENCES STOK.ROLLER (ID));

ROLLER TABLOSU

ALTER TABLE STOK.ROLLER

DROP PRIMARY KEY CASCADE;

DROP TABLE STOK.ROLLER CASCADE CONSTRAINTS;

CREATE TABLE STOK.ROLLER

(

ID NUMBER(38),

KULLANICI\_TURU VARCHAR2(250 BYTE),

KAYIT\_DURUM NUMBER(38)

)

TABLESPACE USERS

PCTUSED 0

PCTFREE 10

INITRANS 1

MAXTRANS 255

STORAGE (

INITIAL 64K

NEXT 1M

MINEXTENTS 1

MAXEXTENTS UNLIMITED

PCTINCREASE 0

BUFFER\_POOL DEFAULT

)

LOGGING

NOCOMPRESS

NOCACHE

NOPARALLEL

MONITORING;

CREATE UNIQUE INDEX STOK.ROLLER\_PK ON STOK.ROLLER

(ID)

LOGGING

TABLESPACE USERS

PCTFREE 10

INITRANS 2

MAXTRANS 255

STORAGE (

INITIAL 64K

NEXT 1M

MINEXTENTS 1

MAXEXTENTS UNLIMITED

PCTINCREASE 0

BUFFER\_POOL DEFAULT

)

NOPARALLEL;

CREATE OR REPLACE TRIGGER STOK.TRG\_ROLLER

BEFORE INSERT

ON STOK.ROLLER

REFERENCING NEW AS New OLD AS Old

FOR EACH ROW

BEGIN

*-- BEGIN*

SELECT STOK.SQ\_ROLLER.NEXTVAL INTO :NEW.ID FROM DUAL;

END;

/

ALTER TABLE STOK.ROLLER ADD (

CONSTRAINT ROLLER\_PK

PRIMARY KEY

(ID)

USING INDEX STOK.ROLLER\_PK);

SATIŞ TABLOSU

ALTER TABLE STOK.SATIS

DROP PRIMARY KEY CASCADE;

DROP TABLE STOK.SATIS CASCADE CONSTRAINTS;

CREATE TABLE STOK.SATIS

(

ID NUMBER(38),

URUN\_ID NUMBER(38),

KULLANICI\_ID NUMBER(38),

MIKTARI NUMBER(38),

TANIM VARCHAR2(450 BYTE),

URUN\_GRUP\_ID NUMBER(38),

FIYAT NUMBER(10,2),

ACIKLAMA VARCHAR2(450 BYTE),

TARIH DATE,

FATURA\_NO VARCHAR2(100 BYTE)

)

TABLESPACE USERS

PCTUSED 0

PCTFREE 10

INITRANS 1

MAXTRANS 255

STORAGE (

INITIAL 64K

NEXT 1M

MINEXTENTS 1

MAXEXTENTS UNLIMITED

PCTINCREASE 0

BUFFER\_POOL DEFAULT

)

LOGGING

NOCOMPRESS

NOCACHE

NOPARALLEL

MONITORING;

CREATE UNIQUE INDEX STOK.SATIS\_PK ON STOK.SATIS

(ID)

LOGGING

TABLESPACE USERS

PCTFREE 10

INITRANS 2

MAXTRANS 255

STORAGE (

INITIAL 64K

NEXT 1M

MINEXTENTS 1

MAXEXTENTS UNLIMITED

PCTINCREASE 0

BUFFER\_POOL DEFAULT

)

NOPARALLEL;

CREATE OR REPLACE TRIGGER STOK.TRG\_SATIS

BEFORE INSERT

ON STOK.SATIS

REFERENCING NEW AS New OLD AS Old

FOR EACH ROW

BEGIN

*-- BEGIN*

SELECT STOK.SQ\_SATIS.NEXTVAL INTO :NEW.ID FROM DUAL;

END;

/

CREATE OR REPLACE TRIGGER STOK.TRG\_STOK\_DUS

AFTER INSERT

ON STOK.SATIS FOR EACH ROW

DECLARE

urunmiktari INTEGER;

kalanurun INTEGER;

BEGIN

SELECT STOK\_MIKTARI

INTO urunmiktari

FROM URUNLER

WHERE ID = :NEW.URUN\_ID;

kalanurun := (urunmiktari - :NEW.MIKTARI);

UPDATE URUNLER

SET STOK\_MIKTARI = kalanurun

WHERE ID = :NEW.URUN\_ID;

END TRG\_STOK\_DUS;

/

ALTER TABLE STOK.SATIS ADD (

CONSTRAINT SATIS\_PK

PRIMARY KEY

(ID)

USING INDEX STOK.SATIS\_PK);

ALTER TABLE STOK.SATIS ADD (

FOREIGN KEY (URUN\_ID)

REFERENCES STOK.URUNLER (ID),

FOREIGN KEY (KULLANICI\_ID)

REFERENCES STOK.KULLANICILAR (ID),

FOREIGN KEY (URUN\_GRUP\_ID)

REFERENCES STOK.URUN\_GRUPLARI (ID));

SEPET TABLOSU

ALTER TABLE STOK.SEPET

DROP PRIMARY KEY CASCADE;

DROP TABLE STOK.SEPET CASCADE CONSTRAINTS;

CREATE TABLE STOK.SEPET

(

ID NUMBER(38),

URUN\_ID NUMBER(38),

KULLANICI\_ID NUMBER(38),

MIKTARI NUMBER(38)

)

TABLESPACE USERS

PCTUSED 0

PCTFREE 10

INITRANS 1

MAXTRANS 255

STORAGE (

INITIAL 64K

NEXT 1M

MINEXTENTS 1

MAXEXTENTS UNLIMITED

PCTINCREASE 0

BUFFER\_POOL DEFAULT

)

LOGGING

NOCOMPRESS

NOCACHE

NOPARALLEL

MONITORING;

CREATE UNIQUE INDEX STOK.SEPET\_PK ON STOK.SEPET

(ID)

LOGGING

TABLESPACE USERS

PCTFREE 10

INITRANS 2

MAXTRANS 255

STORAGE (

INITIAL 64K

NEXT 1M

MINEXTENTS 1

MAXEXTENTS UNLIMITED

PCTINCREASE 0

BUFFER\_POOL DEFAULT

)

NOPARALLEL;

CREATE OR REPLACE TRIGGER STOK.TRG\_SEPET

BEFORE INSERT

ON STOK.SEPET

REFERENCING NEW AS New OLD AS Old

FOR EACH ROW

BEGIN

*-- BEGIN*

SELECT STOK.SQ\_SEPET.NEXTVAL INTO :NEW.ID FROM DUAL;

END;

/

ALTER TABLE STOK.SEPET ADD (

CONSTRAINT SEPET\_PK

PRIMARY KEY

(ID)

USING INDEX STOK.SEPET\_PK);

ALTER TABLE STOK.SEPET ADD (

FOREIGN KEY (URUN\_ID)

REFERENCES STOK.URUNLER (ID),

FOREIGN KEY (KULLANICI\_ID)

REFERENCES STOK.KULLANICILAR (ID));

URUN GRUPLARI TABLOSU

ALTER TABLE STOK.URUN\_GRUPLARI

DROP PRIMARY KEY CASCADE;

DROP TABLE STOK.URUN\_GRUPLARI CASCADE CONSTRAINTS;

CREATE TABLE STOK.URUN\_GRUPLARI

(

ID NUMBER(38),

TANIM VARCHAR2(250 BYTE),

KAYIT\_DURUM NUMBER(38)

)

TABLESPACE USERS

PCTUSED 0

PCTFREE 10

INITRANS 1

MAXTRANS 255

STORAGE (

INITIAL 64K

NEXT 1M

MINEXTENTS 1

MAXEXTENTS UNLIMITED

PCTINCREASE 0

BUFFER\_POOL DEFAULT

)

LOGGING

NOCOMPRESS

NOCACHE

NOPARALLEL

MONITORING;

CREATE UNIQUE INDEX STOK.URUN\_GRUPLARI\_PK ON STOK.URUN\_GRUPLARI

(ID)

LOGGING

TABLESPACE USERS

PCTFREE 10

INITRANS 2

MAXTRANS 255

STORAGE (

INITIAL 64K

NEXT 1M

MINEXTENTS 1

MAXEXTENTS UNLIMITED

PCTINCREASE 0

BUFFER\_POOL DEFAULT

)

NOPARALLEL;

CREATE OR REPLACE TRIGGER STOK.TRG\_URUN\_GRUPLARI

BEFORE INSERT

ON STOK.URUN\_GRUPLARI

REFERENCING NEW AS New OLD AS Old

FOR EACH ROW

BEGIN

*-- BEGIN*

SELECT STOK.SQ\_URUN\_GRUPLARI.NEXTVAL INTO :NEW.ID FROM DUAL;

END;

/

ALTER TABLE STOK.URUN\_GRUPLARI ADD (

CONSTRAINT URUN\_GRUPLARI\_PK

PRIMARY KEY

(ID)

USING INDEX STOK.URUN\_GRUPLARI\_PK);

URUNLER TABLOSU

ALTER TABLE STOK.URUNLER

DROP PRIMARY KEY CASCADE;

DROP TABLE STOK.URUNLER CASCADE CONSTRAINTS;

CREATE TABLE STOK.URUNLER

(

ID NUMBER(38),

TANIM VARCHAR2(250 BYTE),

URUN\_GRUP\_ID NUMBER(38),

FIYAT NUMBER(10,2),

STOK\_MIKTARI NUMBER(38),

ACIKLAMA VARCHAR2(450 BYTE),

DOSYA\_YOLU VARCHAR2(450 BYTE)

)

TABLESPACE USERS

PCTUSED 0

PCTFREE 10

INITRANS 1

MAXTRANS 255

STORAGE (

INITIAL 64K

NEXT 1M

MINEXTENTS 1

MAXEXTENTS UNLIMITED

PCTINCREASE 0

BUFFER\_POOL DEFAULT

)

LOGGING

NOCOMPRESS

NOCACHE

NOPARALLEL

MONITORING;

CREATE UNIQUE INDEX STOK.URUNLER\_PK ON STOK.URUNLER

(ID)

LOGGING

TABLESPACE USERS

PCTFREE 10

INITRANS 2

MAXTRANS 255

STORAGE (

INITIAL 64K

NEXT 1M

MINEXTENTS 1

MAXEXTENTS UNLIMITED

PCTINCREASE 0

BUFFER\_POOL DEFAULT

)

NOPARALLEL;

CREATE OR REPLACE TRIGGER STOK.TRG\_URUNLER

BEFORE INSERT

ON STOK.URUNLER

REFERENCING NEW AS New OLD AS Old

FOR EACH ROW

BEGIN

*-- BEGIN*

SELECT STOK.SQ\_URUNLER.NEXTVAL INTO :NEW.ID FROM DUAL;

END;

/

ALTER TABLE STOK.URUNLER ADD (

CONSTRAINT URUNLER\_PK

PRIMARY KEY

(ID)

USING INDEX STOK.URUNLER\_PK);

ALTER TABLE STOK.URUNLER ADD (

FOREIGN KEY (URUN\_GRUP\_ID)

REFERENCES STOK.URUN\_GRUPLARI (ID));

STORED PROCEDURE VE TRIGGERLAR

CREATE OR REPLACE PROCEDURE STOK.SEPETTEKI\_TOPLAM\_URUN\_SAYISI (

sepettekiSayi OUT NUMBER)

IS

BEGIN

SELECT SUM (S.MIKTARI)

INTO sepettekiSayi

FROM SEPET S;

END SEPETTEKI\_TOPLAM\_URUN\_SAYISI;

CREATE OR REPLACE PROCEDURE STOK.TOPLAM\_URUN\_SAYISI (

toplamSayi OUT NUMBER)

IS

BEGIN

SELECT SUM (U.STOK\_MIKTARI)

INTO toplamSayi

FROM URUNLER U;

END TOPLAM\_URUN\_SAYISI;

DROP TRIGGER STOK.TRG\_STOK\_DUS;

CREATE OR REPLACE TRIGGER STOK.TRG\_STOK\_DUS

AFTER INSERT

ON STOK.SATIS FOR EACH ROW

DECLARE

urunmiktari INTEGER;

kalanurun INTEGER;

BEGIN

SELECT STOK\_MIKTARI

INTO urunmiktari

FROM URUNLER

WHERE ID = :NEW.URUN\_ID;

kalanurun := (urunmiktari - :NEW.MIKTARI);

UPDATE URUNLER

SET STOK\_MIKTARI = kalanurun

WHERE ID = :NEW.URUN\_ID;

END TRG\_STOK\_DUS;