

T.C.
SAKARYA ÜNİVERSİTESİ
BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

Ders : Veritabanı Yönetim Sistemleri

Dönem : 2020-2021 GÜZ Dönemi

Adı Soyadı : Yunus Emre Eminler

Okul No : G191210046

Grup : 2-C

Eposta : yunus.eminler@ogr.sakarya.edu.tr

Konu : Proje Ödevi

1-Uygulama Tanıtımı-İş Kuralları- İlişkisel Model

A) Uygulama Tanıtımı

Uygulamamızın amacı bir süpermarketin veritabanını tasarlayıp personellerin kasaları etkin şekilde kullanmasını sağlamaktır. Uygulamanın kasayı kullanan personellerin kullanacağı düşünülmüştür.

B) İş Kuralları

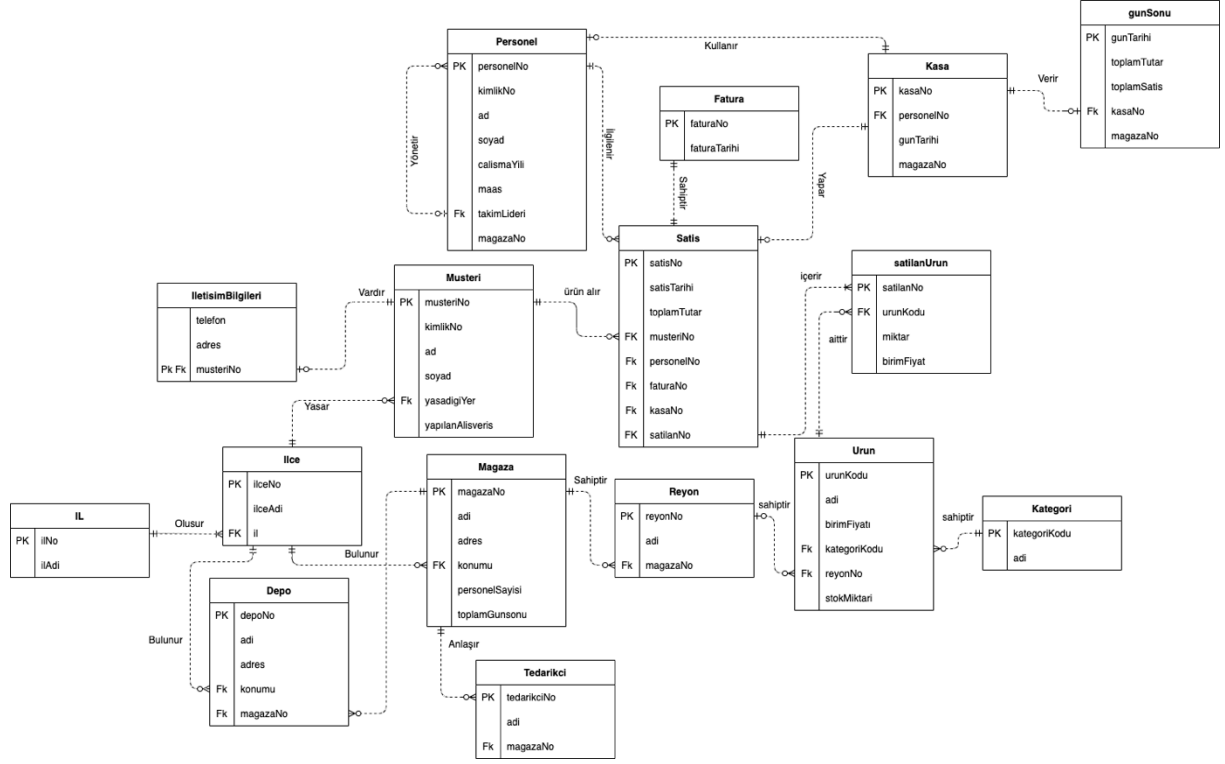
- Her müşterinin müşteriNo'su olması gerekmektedir.
- Müşterinin iletişim bilgilerinin saklanması gerekir.
- İl ilçelerden oluşur.
- Bir il en az bir ilçeden oluşur.
- Bir ilçenin en çok bir ili olabilir.
- Her satışın faturasının olması gerekir.
- Bir satışın en fazla bir faturası olabilir.
- Satışlar kasadan yapılır
- Personel kasayı kullanır.
- Bir kasayı en çok bir personel kullanır.
- Bir personel en çok bir kasa kullanabilir.
- Personellere en az bir Personel liderlik yapar.
- Liderin olması için en az bir personel gerekir.
- Her satışın satış numarası olmalıdır.
- Her satış ürün içermelidir.
- Her ürünün ürünKodu olması gerekir.
- Her ürünün kategorisi vardır.
- Her ürün reyona sahiptir.
- Bir reyon çok sayıda ürüne sahip olabilir.
- Bir mağazada çok sayıda reyon bulunur.
- Bir mağazanın çok sayıda deposu olabilir.
- Bir deponun olması için en az bir mağazanın olması gerekir.
- Her mağaza tedarikçiler ile anlaşabilir.
- Her mağaza en az sıfır en çok çok sayıda tedarikçi ile anlaşabilir.
- Kasa her gün gün sonu tablosu verir.
- Gün sonu tablosu olması için kasanın olması gerekir.

- Bir kasa en çok bir günsonu tablosu verir.
- Bir satışın olması için en az bir kasanın olması gerekir.
- Bir satışın olması için en az bir personelin olması gerekir.

C) İlişkisel Model

- Fatura(faturaNo: String, faturaTarihi: Date)
- Kasa(kasaNo: String, personelNo: String, gunTarihi:Date, magazaNo: String)
- gunSonu(gunTarihi: Date, toplamTutar: Integer, toplamSatis: Integer,kasaNo: String, magazaNo: String)
- Personel(personelNo: String, kimlikNo: Char(11), ad: String, soyad: String, calismaYili: Char(4), maas: Integer, takimLideri: String, magazaNo: String)
- Satis(satisNo: String, satisTarihi: Date, satilanNo: String, toplamTutar: Integer, musterino: String, personelNo: String, faturaNo: String, kasaNo: String)
- satilanUrun(satilanNo: String, urunKodu:String, miktar: Integer, birimFiyat: Integer)
- Musteri(musteriNo: String, kimlikNo: char(11), ad: String, soyad:String, yasadigiYer: String, yapilanAlisveris: Integer)
- iletisimBilgileri(telefon: String, adres:String, musterino: String)
- ilce(ilceNo: char(2), ilceAdi: String, il: String)
- il(ilNo: String, ilAdi: String)
- Depo(depoNo: String, adi:String, adres: String, konumu: String, magazaNo: String)
- Magaza(magazaNo: String, adi: String, adres:String, konumu:String, personelSayisi: Integer, toplamGunSonu: Integer)
- Reyon(reyonNo: String, adi: String, magazaNo:String)
- Tedarikci(tedarikciNo: String, adi: String,magazaNo)
- Urun(urunKodu: String, adi: String, birimFiyat: Integer, kategoriKodu: String, reyonNo: String, stokMiktari: Integer)
- Kategori(kategoriKodu: String, adi: String)

2-Varlık Bağntı Modeli



3- VeriTabanı SQL İfadeleri

```
SET statement_timeout = 0;
SET lock_timeout = 0;
SET idle_in_transaction_session_timeout = 0;
SET client_encoding = 'UTF8';
SET standard_conforming_strings = on;
SELECT pg_catalog.set_config('search_path', '', false);
SET check_function_bodies = false;
SET xmloption = content;
SET client_min_messages = warning;
SET row_security = off;
```

```
CREATE FUNCTION public.alisveris() RETURNS trigger
    LANGUAGE plpgsql
    AS $$
    DECLARE
        musteri varchar;
    BEGIN
        musteri:= (SELECT MAX("satisNo") FROM "Satis");
        musteri:= (select "musteriNo" from "Satis" where "satisNo" = musteri);
        UPDATE "Musteri" SET "yapilanAlisveris" = "yapilanAlisveris" + 1 Where
"musteriNo" = musteri;
        RETURN NEW;
    END;
    $$;
```

```
ALTER FUNCTION public.alisveris() OWNER TO postgres;
```

```
CREATE FUNCTION public."calismaSuresi"(personel character varying) RETURNS integer
    LANGUAGE plpgsql
    AS $$
    DECLARE
        yil Integer;
    BEGIN
        yil:= (Select "calismaYili" from "Personel" where "personelNo" = personel);
        return (2020-yil);
    END;
    $$;
```

```
ALTER FUNCTION public."calismaSuresi"(personel character varying) OWNER TO
postgres;
```

```
CREATE FUNCTION public."gunSonuEkleme"() RETURNS trigger
    LANGUAGE plpgsql
    AS $$
    DECLARE
        gun DATE;
```

```

        magaza VARCHAR;
BEGIN
    gun := (SELECT MAX("gunTarihi") From "gunSonu");
    magaza := (SELECT "magazaNo" from "gunSonu" where "gunTarihi" = gun);
    UPDATE "Magaza" SET "toplamGunSonu" = "toplamGunSonu" + 1 Where "magazaNo"
= magaza;
    return NEW;
END;
$$;

```

```

ALTER FUNCTION public."gunSonuEkleme"() OWNER TO postgres;

```

```

CREATE FUNCTION public.kasasatis(kasa character varying) RETURNS TABLE(musterino
character varying, satis date, tutar integer, satisno character varying, fatura
character varying, personel character varying, satilan character varying)
LANGUAGE plpgsql
AS $$
BEGIN
    RETURN QUERY SELECT "musteriNo", "satisTarihi", "toplamTutar",
"satisNo","faturaNo","personelNo","satilanNo" FROM "Satis" WHERE "kasaNo" = kasa;
END;
$$;

```

```

ALTER FUNCTION public.kasasatis(kasa character varying) OWNER TO postgres;

```

```

CREATE FUNCTION public.musterisatis(musteri character varying) RETURNS
TABLE(musterino character varying, satis date, tutar integer, satisno character
varying)
LANGUAGE plpgsql
AS $$
BEGIN
    RETURN QUERY SELECT "musteriNo", "satisTarihi", "toplamTutar", "satisNo" FROM
"Satis"
        WHERE "musteriNo" = muster;
END;
$$;

```

```

ALTER FUNCTION public.musterisatis(musteri character varying) OWNER TO postgres;

```

```

CREATE FUNCTION public."personelEkleme"() RETURNS trigger
LANGUAGE plpgsql
AS $$
DECLARE
    magaza varchar;
BEGIN
    magaza := (SeLECT MAX("personelNo") FROM "Personel");
    magaza := (SeLECT "magazaNo" FROM "Personel" WHERE "personelNo"= magaza);

```

```

        UPDATE "Magaza" SET "personelSayisi" = "personelSayisi" + 1 Where "magazaNo"
= magaza;
    RETURN NEW;
END;
$$;

```

```

ALTER FUNCTION public."personelEkleme"() OWNER TO postgres;

```

```

CREATE FUNCTION public."satilanEkle"(urun character varying, sayi integer) RETURNS
boolean
    LANGUAGE plpgsql
    AS $$
    DECLARE
        satilan VARCHAR;
        birim INTEGER;
    BEGIN
        birim := (SELECT "birimFiyat" FROM "Urun" where "urunKodu"=urun);
        satilan := CAST((SELECT CAST(MAX("satilanNo") AS int) from "satilanUrun")+1
AS varchar(5));
        INSERT INTO "satilanUrun" VALUES (satilan, urun, sayi,birim);

        return true;

    END;
    $$;

```

```

ALTER FUNCTION public."satilanEkle"(urun character varying, sayi integer) OWNER TO
postgres;

```

```

CREATE FUNCTION public.sonlandir(gun date, kasa character varying, magaza character
varying) RETURNS boolean
    LANGUAGE plpgsql
    AS $$
    DECLARE
        satis INTEGER;
        tutar VARCHAR;

    BEGIN
        satis:= (SELECT COUNT(*) FROM "Satis" WHERE "satisTarihi" = gun AND "kasaNo" =
kasa);
        tutar:= (SELECT SUM("toplamTutar") FROM "Satis" WHERE "satisTarihi" = gun AND
"kasaNo" = kasa);
        tutar:= (SELECT "coalesce" FROM COALESCE(tutar,'0'));
        INSERT INTO "gunSonu" VALUES (gun, tutar, satis,kasa,magaza);
        return TRUE;
    END
    $$;

```

```
ALTER FUNCTION public.sonlandir(gun date, kasa character varying, magaza character varying) OWNER TO postgres;
```

```
CREATE FUNCTION public."urunSatis"() RETURNS trigger
    LANGUAGE plpgsql
    AS $$
    DECLARE
        urun varchar;
        miktari Integer;
        satilan VARCHAR;
    BEGIN
        satilan := (SELECT MAX("satilanNo") FROM "satilanUrun");
        urun := (SELECT "urunKodu" FROM "satilanUrun" WHERE "satilanNo"= satilan);
        miktari:= (SELECT "miktar" FROM "satilanUrun" WHERE "satilanNo"= satilan);
        UPDATE "Urun" SET "stokMiktari" = "stokMiktari" - miktari Where "urunKodu"
= urun;
        RETURN NEW;
    END;
    $$;
```

```
ALTER FUNCTION public."urunSatis"() OWNER TO postgres;
```

```
SET default_tablespace = '';
```

```
SET default_table_access_method = heap;
```

```
CREATE TABLE public."Depo" (
    "depoNo" character varying NOT NULL,
    adi character varying NOT NULL,
    adres text NOT NULL,
    konumu character varying NOT NULL,
    "magazaNo" character varying NOT NULL
);
```

```
ALTER TABLE public."Depo" OWNER TO postgres;
```

```
CREATE TABLE public."Fatura" (
    "faturaNo" character varying NOT NULL,
    "faturaTarihi" date NOT NULL
);
```

```
ALTER TABLE public."Fatura" OWNER TO postgres;
```

```
CREATE TABLE public."Kasa" (
    "kasaNo" character varying NOT NULL,
    "gunTarihi" date NOT NULL,
    "personelNo" character varying NOT NULL,
    "magazaNo" character varying NOT NULL
);
```



```
ALTER TABLE public."Kasa" OWNER TO postgres;
```

```
CREATE TABLE public."Kategori" (  
    "kategoriKodu" character varying NOT NULL,  
    adi character varying NOT NULL  
);
```

```
ALTER TABLE public."Kategori" OWNER TO postgres;
```

```
CREATE TABLE public."Magaza" (  
    "magazaNo" character varying NOT NULL,  
    adi character varying NOT NULL,  
    adres text NOT NULL,  
    konumu character varying NOT NULL,  
    "toplamGunSonu" integer DEFAULT 0 NOT NULL,  
    "personelSayisi" integer  
);
```

```
ALTER TABLE public."Magaza" OWNER TO postgres;
```

```
CREATE TABLE public."Musteri" (  
    "musteriNo" character varying NOT NULL,  
    "kimlikNo" character varying(11) NOT NULL,  
    ad character varying NOT NULL,  
    soyad character varying NOT NULL,  
    "yasadigiYer" character varying NOT NULL,  
    "yapilanAlisveris" integer DEFAULT 0 NOT NULL  
);
```

```
ALTER TABLE public."Musteri" OWNER TO postgres;
```

```
CREATE TABLE public."Personel" (  
    "personelNo" character varying NOT NULL,  
    "kimlikNo" character varying(11) NOT NULL,  
    ad character varying NOT NULL,  
    soyad character varying NOT NULL,  
    "calismaYili" integer NOT NULL,  
    maas integer NOT NULL,  
    "takimLideri" character varying NOT NULL,  
    "magazaNo" character varying NOT NULL  
);
```

```
ALTER TABLE public."Personel" OWNER TO postgres;
```

```
CREATE TABLE public."Reyon" (  
    "reyonNo" character varying NOT NULL,  
    adi character varying NOT NULL,  
    "magazaNo" character varying NOT NULL  
);
```

```
ALTER TABLE public."Reyon" OWNER TO postgres;
```

```
CREATE TABLE public."Satis" (  
    "satisNo" character varying NOT NULL,  
    "satisTarihi" date NOT NULL,  
    "toplamTutar" integer NOT NULL,  
    "musteriNo" character varying NOT NULL,  
    "personelNo" character varying NOT NULL,  
    "faturaNo" character varying NOT NULL,  
    "kasaNo" character varying NOT NULL,  
    "satilanNo" character varying NOT NULL  
);
```

```
ALTER TABLE public."Satis" OWNER TO postgres;
```

```
CREATE TABLE public."Tedarikci" (  
    "tedarikciNo" character varying NOT NULL,  
    adi character varying NOT NULL,  
    "magazaNo" character varying NOT NULL  
);
```

```
ALTER TABLE public."Tedarikci" OWNER TO postgres;
```

```
CREATE TABLE public."Urun" (  
    "urunKodu" character varying NOT NULL,  
    adi character varying NOT NULL,  
    "birimFiyat" integer NOT NULL,  
    "kategoriKodu" character varying NOT NULL,  
    "reyonNo" character varying NOT NULL,  
    "stokMiktari" integer NOT NULL  
);
```

```
ALTER TABLE public."Urun" OWNER TO postgres;
```

```
CREATE TABLE public."gunSonu" (  
    "gunTarihi" date NOT NULL,  
    "toplamTutar" character varying NOT NULL,  
    "toplamSatis" integer NOT NULL,  
    "kasaNo" character varying NOT NULL,  
    "magazaNo" character varying NOT NULL  
);
```

```
ALTER TABLE public."gunSonu" OWNER TO postgres;
```

```
CREATE TABLE public.il (  
    "ilNo" character varying NOT NULL,  
    adi character varying NOT NULL  
);
```

```
ALTER TABLE public.il OWNER TO postgres;
```

```

CREATE TABLE public.ilce (
    "ilceNo" character varying NOT NULL,
    "ilceAdi" character varying NOT NULL,
    il character varying NOT NULL
);

ALTER TABLE public.ilce OWNER TO postgres;

CREATE TABLE public."iletisimBilgileri" (
    telefon character varying NOT NULL,
    adres text NOT NULL,
    "musteriNo" character varying NOT NULL
);

ALTER TABLE public."iletisimBilgileri" OWNER TO postgres;

CREATE TABLE public."satilanUrun" (
    "satilanNo" character varying NOT NULL,
    "urunKodu" character varying NOT NULL,
    miktar integer NOT NULL,
    "birimFiyat" integer NOT NULL
);

ALTER TABLE public."satilanUrun" OWNER TO postgres;

INSERT INTO public."Depo" VALUES
    ('001', 'HendekDepo', 'hendek mh.', '6', '001'),
    ('002', 'EsenlerDepo', 'Esenler mh.', '5', '001');

INSERT INTO public."Fatura" VALUES
    ('001', '2020-05-27'),
    ('002', '2020-05-27');

INSERT INTO public."Kasa" VALUES
    ('001', '2020-05-27', '001', '001');

INSERT INTO public."Kategori" VALUES
    ('001', 'UnluMamuller'),
    ('002', 'icecekler'),
    ('003', 'pazarUrunu');

INSERT INTO public."Magaza" VALUES
    ('003', 'Migros', 'esenler mh', '5', 0, 0),
    ('002', 'Migros', 'hendek mh. ', '6', 0, 2),
    ('001', 'Migros', 'yenidoğan mh caner sk.', '1', 1, 3);

INSERT INTO public."Musteri" VALUES
    ('003', '12345678933', 'Eminler', 'Eminler', '6', 0),
    ('001', '12345678911', 'Yunus', 'Yunus', '1', 2),
    ('002', '12345678922', 'Emre', 'Emre', '1', 0);

```

```
INSERT INTO public."Personel" VALUES
  ('005', '12345678922', 'as', 'as', 2001, 5000, '001', '002'),
  ('001', '12345678911', 'Elif', 'can', 2010, 5000, '001', '001'),
  ('002', '12345678911', 'Ali', 'can', 2015, 4000, '001', '002'),
  ('003', '12345678911', 'Ayşe', 'can', 2017, 2500, '001', '001'),
  ('004', '12345678911', 'Mustafa', 'can', 2019, 2750, '001', '001');
```

```
INSERT INTO public."Reyon" VALUES
  ('001', 'İcecekler', '001'),
  ('002', 'Sebzeler', '001'),
  ('003', 'Meyveler', '001'),
  ('004', 'Yiyecekler', '001');
```

```
INSERT INTO public."Satis" VALUES
  ('001', '2020-05-27', 50, '001', '001', '001', '001', '1'),
  ('002', '2020-05-27', 35, '001', '001', '002', '001', '2');
```

```
INSERT INTO public."Tedarikci" VALUES
  ('001', 'Kardeşler', '001'),
  ('002', 'AliBaba', '002'),
  ('003', 'Sütcü', '003');
```

```
INSERT INTO public."Urun" VALUES
  ('003', 'Kola', 8, '002', '001', 37),
  ('001', 'Ekmek', 1, '001', '004', 36),
  ('002', 'Elma', 5, '003', '003', 97);
```

```
INSERT INTO public."gunSonu" VALUES
  ('2020-05-27', '50', 1, '001', '001');
```

```
INSERT INTO public.il VALUES
  ('54', 'Sakarya'),
  ('34', 'İstanbul'),
  ('06', 'Ankara');
```

```
INSERT INTO public.ilce VALUES
  ('1', 'Adapazarı', '54'),
  ('5', 'Esenler', '34'),
  ('6', 'Hendek', '54');
```

```
INSERT INTO public."iletisimBilgileri" VALUES
  ('5415613970', 'yenidoğan', '001'),
  ('56468541345', 'asd', '003'),
  ('54125645212', 'caner', '002');
```

```
INSERT INTO public."satilanUrun" VALUES
  ('1', '001', 5, 1),
  ('2', '001', 4, 1),
  ('3', '001', 5, 1),
  ('4', '002', 5, 5);
```

```
ALTER TABLE ONLY public."Fatura"  
    ADD CONSTRAINT "Fatura_pkey" PRIMARY KEY ("faturaNo");  
  
ALTER TABLE ONLY public."Satis"  
    ADD CONSTRAINT "Satis_pkey" PRIMARY KEY ("satisNo");  
  
ALTER TABLE ONLY public."Urun"  
    ADD CONSTRAINT "Urun_pkey" PRIMARY KEY ("urunKodu");  
  
ALTER TABLE ONLY public."gunSonu"  
    ADD CONSTRAINT "gunSonu_pkey" PRIMARY KEY ("gunTarihi");  
  
ALTER TABLE ONLY public."satilanUrun"  
    ADD CONSTRAINT "satilanUrun_pkey" PRIMARY KEY ("satilanNo");  
  
ALTER TABLE ONLY public."Depo"  
    ADD CONSTRAINT "unique_Depo_depoNo" PRIMARY KEY ("depoNo");  
  
ALTER TABLE ONLY public."Fatura"  
    ADD CONSTRAINT "unique_Fatura_faturaNo" UNIQUE ("faturaNo");  
  
ALTER TABLE ONLY public."Kategori"  
    ADD CONSTRAINT "unique_Kategori_kategoriKodu" PRIMARY KEY ("kategoriKodu");  
  
ALTER TABLE ONLY public."Magaza"  
    ADD CONSTRAINT "unique_Magaza_magazaNo" PRIMARY KEY ("magazaNo");  
  
ALTER TABLE ONLY public."Musteri"  
    ADD CONSTRAINT "unique_Musteri_musteriNo" PRIMARY KEY ("musteriNo");  
  
ALTER TABLE ONLY public."Personel"  
    ADD CONSTRAINT "unique_Personel_personelNo" PRIMARY KEY ("personelNo");  
  
ALTER TABLE ONLY public."Reyon"  
    ADD CONSTRAINT "unique_Reyon_reyonNo" PRIMARY KEY ("reyonNo");  
  
ALTER TABLE ONLY public."Satis"  
    ADD CONSTRAINT "unique_Satis_satisNo" UNIQUE ("satisNo");  
  
ALTER TABLE ONLY public."Tedarikci"  
    ADD CONSTRAINT "unique_Tedarikci_tedarikciNo" PRIMARY KEY ("tedarikciNo");  
  
ALTER TABLE ONLY public."Urun"  
    ADD CONSTRAINT "unique_Urun_urunKodu" UNIQUE ("urunKodu");  
  
ALTER TABLE ONLY public."gunSonu"  
    ADD CONSTRAINT "unique_gunSonu_gunTarihi" UNIQUE ("gunTarihi");  
  
ALTER TABLE ONLY public.il  
    ADD CONSTRAINT "unique_il_ilNo" PRIMARY KEY ("ilNo");
```

```

ALTER TABLE ONLY public.ilce
    ADD CONSTRAINT "unique_ilce_ilceNo" PRIMARY KEY ("ilceNo");

ALTER TABLE ONLY public."iletisimBilgileri"
    ADD CONSTRAINT "unique_iletisimBilgileri_musteriNo" PRIMARY KEY ("musteriNo");

ALTER TABLE ONLY public."satilanUrun"
    ADD CONSTRAINT "unique_satilanUrun_satilanNo" UNIQUE ("satilanNo");

CREATE INDEX "index_Magaza_id" ON public."Depo" USING btree ("magazaNo");

CREATE INDEX "index_Magaza_id1" ON public."Personel" USING btree ("magazaNo");

CREATE INDEX index_il ON public.ilce USING btree (il);

CREATE INDEX "index_ilceNo" ON public.ilce USING btree ("ilceNo");

CREATE TRIGGER alisveris AFTER INSERT ON public."Satis" FOR EACH ROW EXECUTE
FUNCTION public.alisveris();

CREATE TRIGGER ekleme AFTER INSERT ON public."gunSonu" FOR EACH ROW EXECUTE
FUNCTION public."gunSonuEkleme"();

CREATE TRIGGER eklenme AFTER INSERT ON public."Personel" FOR EACH ROW EXECUTE
FUNCTION public."personelEkleme"();

CREATE TRIGGER "personelEkleme" AFTER INSERT ON public."Personel" FOR EACH ROW
EXECUTE FUNCTION public."personelEkleme"();

CREATE TRIGGER satis AFTER INSERT ON public."satilanUrun" FOR EACH ROW EXECUTE
FUNCTION public."urunSatis"();

CREATE TRIGGER "urunSatis" AFTER INSERT ON public."satilanUrun" FOR EACH ROW
EXECUTE FUNCTION public."urunSatis"();

ALTER TABLE ONLY public."Satis"
    ADD CONSTRAINT "lnk_Fatura_Satis" FOREIGN KEY ("faturaNo") REFERENCES
public."Fatura"("faturaNo") MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY public."Urun"
    ADD CONSTRAINT "lnk_Kategori_Urun" FOREIGN KEY ("kategoriKodu") REFERENCES
public."Kategori"("kategoriKodu") MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY public."Depo"
    ADD CONSTRAINT "lnk_Magaza_Depo" FOREIGN KEY ("magazaNo") REFERENCES
public."Magaza"("magazaNo") MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY public."Personel"
    ADD CONSTRAINT "lnk_Magaza_Personel" FOREIGN KEY ("magazaNo") REFERENCES
public."Magaza"("magazaNo") MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

```

```

ALTER TABLE ONLY public."Reyon"
    ADD CONSTRAINT "lnk_Magaza_Reyon_2" FOREIGN KEY ("magazaNo") REFERENCES
public."Magaza"("magazaNo") MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY public."Tedarikci"
    ADD CONSTRAINT "lnk_Magaza_Tedarikci" FOREIGN KEY ("magazaNo") REFERENCES
public."Magaza"("magazaNo") MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY public."Satis"
    ADD CONSTRAINT "lnk_Musteri_Satis" FOREIGN KEY ("musteriNo") REFERENCES
public."Musteri"("musteriNo") MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY public."iletisimBilgileri"
    ADD CONSTRAINT "lnk_Musteri_iletisimBilgileri" FOREIGN KEY ("musteriNo")
REFERENCES public."Musteri"("musteriNo") MATCH FULL ON UPDATE CASCADE ON DELETE
CASCADE;

ALTER TABLE ONLY public."Personel"
    ADD CONSTRAINT "lnk_Personel_Personel" FOREIGN KEY ("takimLideri") REFERENCES
public."Personel"("personelNo") MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY public."Satis"
    ADD CONSTRAINT "lnk_Personel_Satis" FOREIGN KEY ("personelNo") REFERENCES
public."Personel"("personelNo") MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY public."Urun"
    ADD CONSTRAINT "lnk_Reyon_Urun" FOREIGN KEY ("reyonNo") REFERENCES
public."Reyon"("reyonNo") MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY public."Satis"
    ADD CONSTRAINT "lnk_Satis_satilanUrun" FOREIGN KEY ("satilanNo") REFERENCES
public."satilanUrun"("satilanNo") MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY public."satilanUrun"
    ADD CONSTRAINT "lnk_Urun_satilanUrun" FOREIGN KEY ("urunKodu") REFERENCES
public."Urun"("urunKodu") MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY public.ilce
    ADD CONSTRAINT lnk_il_ilce FOREIGN KEY (il) REFERENCES public.il("ilNo") MATCH
FULL ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY public."Depo"
    ADD CONSTRAINT "lnk_ilce_Depo" FOREIGN KEY (konumu) REFERENCES
public.ilce("ilceNo") MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY public."Magaza"
    ADD CONSTRAINT "lnk_ilce_Magaza" FOREIGN KEY (konumu) REFERENCES
public.ilce("ilceNo") MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY public."Musteri"

```

```
ADD CONSTRAINT "lnk_ilce_Musteri" FOREIGN KEY ("yasadigiYer") REFERENCES  
public.ilce("ilceNo") MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE;
```

4- Fonksiyonlar(Saklı Yordam)

1.Fonksiyon

```
CREATE OR REPLACE FUNCTION public."calismaSuresi"(personel character varying)  
RETURNS integer  
LANGUAGE plpgsql  
AS $function$  
    DECLARE  
        yil Integer;  
    BEGIN  
        yil:= (Select "calismaYili" from "Personel" where "personelNo" = personel);  
        return (2020-yil);  
    END;  
$function$
```

2.Fonksiyon

```
CREATE OR REPLACE FUNCTION public.kasasatis(kasa character varying)  
    RETURNS TABLE(musterino character varying, satis date, tutar integer, satisno  
character varying, fatura character varying, personel character varying, satilan  
character varying)  
    LANGUAGE plpgsql  
AS $function$  
BEGIN  
    RETURN QUERY SELECT "musteriNo", "satisTarihi", "toplamTutar",  
"satisNo","faturaNo","personelNo","satilanNo" FROM "Satis" WHERE "kasaNo" = kasa;  
END;  
$function$
```

3.Fonksiyon

```
CREATE OR REPLACE FUNCTION public.musterisatis(musteri character varying)  
    RETURNS TABLE(musterino character varying, satis date, tutar integer, satisno  
character varying)  
    LANGUAGE plpgsql  
AS $function$  
BEGIN  
    RETURN QUERY SELECT "musteriNo", "satisTarihi", "toplamTutar", "satisNo" FROM  
"Satis" WHERE "musteriNo" = muster;
```



```
END;  
$function$
```

4.Fonksiyon

```
CREATE OR REPLACE FUNCTION public."satilanEkle"(urun character varying, sayi  
integer)  
RETURNS boolean  
LANGUAGE plpgsql  
AS $function$  
    DECLARE  
        satilan VARCHAR;  
        birim INTEGER;  
    BEGIN  
        birim := (SELECT "birimFiyat" FROM "Urun" where "urunKodu"=urun);  
        satilan := CAST((SELECT CAST(MAX("satilanNo") AS int) from "satilanUrun")+1  
AS varchar(5));  
        INSERT INTO "satilanUrun" VALUES (satilan, urun, sayi,birim);  
  
        return true;  
  
    END;  
$function$
```

5.Fonksiyon

```
CREATE OR REPLACE FUNCTION public.sonlandir(gun date, kasa character varying,  
magaza character varying)  
RETURNS boolean  
LANGUAGE plpgsql  
AS $function$  
    DECLARE  
        satis INTEGER;  
        tutar VARCHAR;  
  
    BEGIN  
        satis:= (SELECT COUNT(*) FROM "Satis" WHERE "satisTarihi" = gun AND "kasaNo" =  
kasa);  
        tutar:= (SELECT SUM("toplamTutar") FROM "Satis" WHERE "satisTarihi" = gun AND  
"kasaNo" = kasa);  
        tutar:= (SELECT "coalesce" FROM COALESCE(tutar,'0'));  
        INSERT INTO "gunSonu" VALUES (gun, tutar, satis,kasa,magaza);  
        return TRUE;
```

```
END
$function$
```

4.1- Trigger(Tetikleyici)

1.Tetikleyici

```
CREATE TRIGGER alisveris AFTER INSERT ON public."Satis" FOR EACH ROW EXECUTE
FUNCTION public.alisveris()
```

```
CREATE OR REPLACE FUNCTION public.alisveris()
RETURNS trigger
LANGUAGE plpgsql
AS $function$
DECLARE
    musteri varchar;
BEGIN
    musteri:= (SELECT MAX("satisNo") FROM "Satis");
    musteri:= (select "musteriNo" from "Satis" where "satisNo" = musteri);
    UPDATE "Musteri" SET "yapilanAlisveris" = "yapilanAlisveris" + 1 Where
    "musteriNo" = musteri;
    RETURN NEW;
END;
$function$
```

2.Tetikleyici

```
CREATE TRIGGER eklme AFTER INSERT ON public."gunSonu" FOR EACH ROW EXECUTE
FUNCTION public."gunSonuEklme"()
```

```
CREATE OR REPLACE FUNCTION public."gunSonuEklme"()
RETURNS trigger
LANGUAGE plpgsql
AS $function$
DECLARE
    gun DATE;
    magaza VARCHAR;
BEGIN
    gun := (SELECT MAX("gunTarihi") From "gunSonu");
    magaza := (SELECT "magazaNo" from "gunSonu" where "gunTarihi" = gun);
    UPDATE "Magaza" SET "toplamGunSonu" = "toplamGunSonu" + 1 Where "magazaNo"
= magaza;
    return NEW;
```

```
END;  
$function$
```

3.Tetikleyici

```
CREATE TRIGGER "personelEkleme" AFTER INSERT ON public."Personel" FOR EACH ROW  
EXECUTE FUNCTION public."personelEkleme"()
```

```
CREATE OR REPLACE FUNCTION public."personelEkleme"()  
RETURNS trigger  
LANGUAGE plpgsql  
AS $function$  
    DECLARE  
        magaza varchar;  
    BEGIN  
        magaza := (SELECT MAX("personelNo") FROM "Personel");  
        magaza := (SELECT "magazaNo" FROM "Personel" WHERE "personelNo"= magaza);  
        UPDATE "Magaza" SET "personelSayisi" = "personelSayisi" + 1 Where "magazaNo"  
= magaza;  
        RETURN NEW;  
    END;  
$function$
```

4.Tetikleyici

```
CREATE TRIGGER "urunSatis" AFTER INSERT ON public."satilanUrun" FOR EACH ROW  
EXECUTE FUNCTION public."urunSatis"()
```

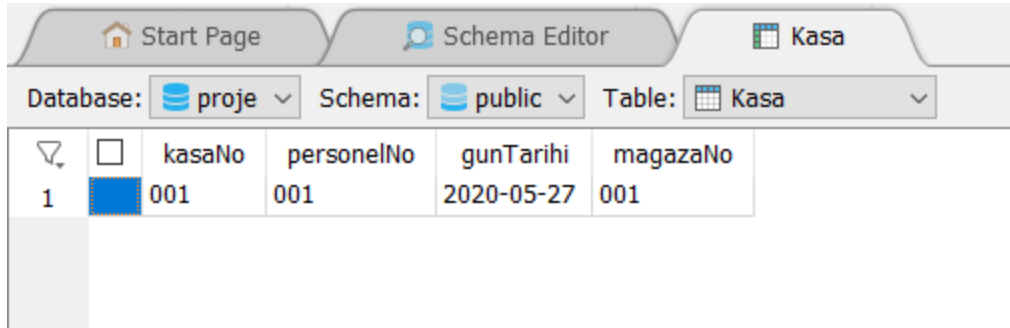
```
CREATE OR REPLACE FUNCTION public."urunSatis"()  
RETURNS trigger  
LANGUAGE plpgsql  
AS $function$  
    DECLARE  
        urun varchar;  
        miktari Integer;  
        satilan VARCHAR;  
    BEGIN  
        satilan := (SELECT MAX("satilanNo") FROM "satilanUrun");  
        urun := (SELECT "urunKodu" FROM "satilanUrun" WHERE "satilanNo"= satilan);  
        miktari:= (SELECT "miktar" FROM "satilanUrun" WHERE "satilanNo"= satilan);
```

```
UPDATE "Urun" SET "stokMiktari" = "stokMiktari" - miktari Where "urunKodu"
= urun;
RETURN NEW;
END;
$function$
```

5- Ekran Görüntüleri

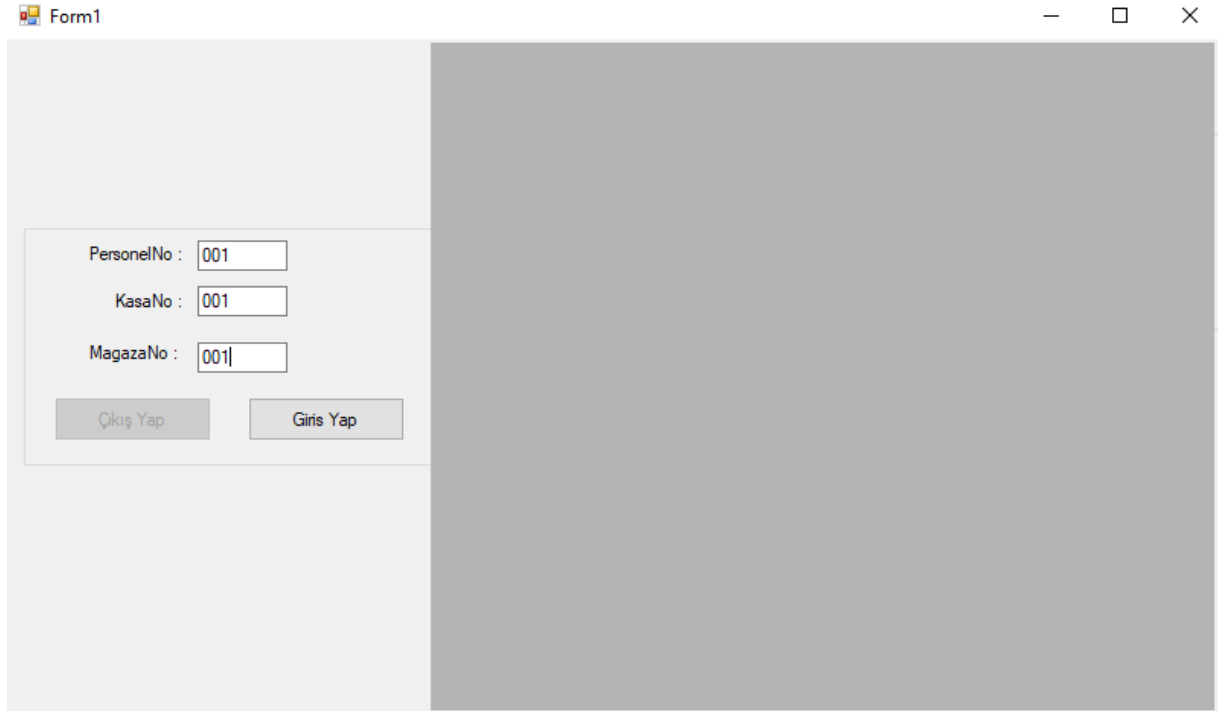
1.Ekleme

1.1 Ekleme yapmadan önce Veritabanı görüntüsü



	kasaNo	personelNo	gunTarihi	magazaNo
1	001	001	2020-05-27	001

personelNo, kasaNo, magazaNo verileri girilip “Giriş Yap” butonuna tıklanıyor.



PersonelNo : 001

KasaNo : 001

MagazaNo : 001

Çıkış Yap Giris Yap

Butona tıklandıktan sonra Kasa tablosuna verilerle birlikte yeni kayıt ekleniyor.Ekleme yaptıktan sonra Veritabanı görüntüsü.

Start Page Schema Editor Kasa					
Database: proje Schema: public Table: Kasa					
		kasaNo	personelNo	gunTarihi	magazaNo
1	<input type="checkbox"/>	001	001	2020-05-27	001
2	<input type="checkbox"/>	001	001	2020-12-29	001

1.2 Ekleme Modülü kullanılmadan önce Veritabanı görüntüsü

Start Page Schema Editor satila					
Database: proje Schema: public Table: satilanUru					
		satilanNo	urunKodu	miktar	birimFiyat
1	<input type="checkbox"/>	1	001	5	1
2	<input type="checkbox"/>	2	001	4	1
3	<input type="checkbox"/>	3	001	5	1
4	<input type="checkbox"/>	4	002	5	5

Modüle urunkodu ve Miktarı giriliyor ve Satis yap butonuna basılıyor.

Ürün Satış

ÜrünKodu:

Miktar :

Satis Yap

Ürün Satış

ÜrünKodu:

Miktar :

Satis Yap

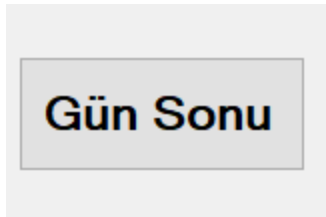
1.2 Ekleme Modülü kullanıldıktan sonra veritabanı görüntüsü

Start Page		Schema Editor		satilanUr	
Database: proje		Schema: public		Table: satilanUrun	
		satilanNo	urunKodu	miktar	birimFiyat
1	<input type="checkbox"/>	1	001	5	1
2	<input type="checkbox"/>	2	001	4	1
3	<input type="checkbox"/>	3	001	5	1
4	<input type="checkbox"/>	4	002	5	5
5	<input type="checkbox"/>	5	002	5	5

1.3 Ekleme Modülü kullanılmadan önceki veritabanı görüntüsü

Start Page		Schema Editor		gunSonu		
Database: proje		Schema: public		Table: gunSonu		
		gunTarihi	toplamTutar	toplamSatis	kasaNo	magazaNo
1	<input type="checkbox"/>	2020-05-27	85	2	001	001
2	<input type="checkbox"/>	2020-12-27	0	0	001	001

GünSonu butonuna basarak kasanın gün boyunca yaptığı satış sayısını tabloya girer



Ekleme Modülü kullanıldıktan sonra veritabanı görüntüsü

Start Page		Schema Editor		gunSonu		
Database: proje		Schema: public		Table: gunSonu		
		gunTarihi	toplamTutar	toplamSatis	kasaNo	magazaNo
1	<input type="checkbox"/>	2020-05-27	85	2	001	001
2	<input type="checkbox"/>	2020-12-27	0	0	001	001
3	<input type="checkbox"/>	2020-12-29	0	0	001	001

2.Arama

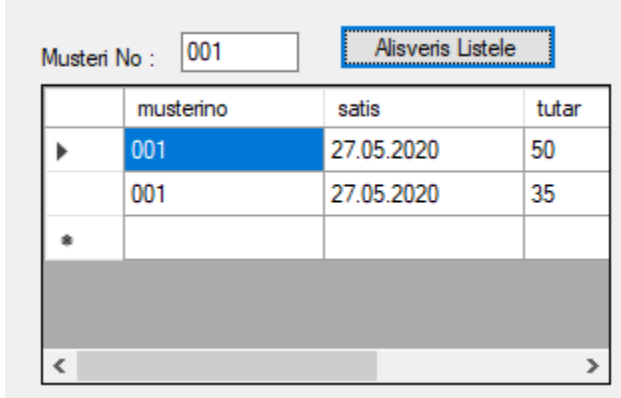
Arama Modülünün aramadan önceki hali



Musteri No :

Alisveris Listele

Musteri No giriliyor ve alisveris Listele butonuna tıklanıyor daha sonra Musteri No'su girilen Musterinin yapmış olduğu satışlar listeleniyor.



Musteri No : 001

Alisveris Listele

	musterino	satis	tutar
▶	001	27.05.2020	50
	001	27.05.2020	35
*			

3.Silme

Silme Modülü kullanılmadan önce Veritabanı görüntüsü

Start Page		Schema Editor		Kasa	
Database: proje		Schema: public		Table: Kasa	
		kasaNo	personelNo	gunTarihi	magazaNo
1	<input type="checkbox"/>	001	001	2020-05-27	001
2	<input type="checkbox"/>	001	001	2020-12-29	001

Çıkış yap butonuna basılarak o gün ev o kasaNosuna sahip kayıt siliniyor.

PersonelNo :	<input type="text" value="001"/>
KasaNo :	<input type="text" value="001"/>
MagazaNo :	<input type="text" value="001"/>
<input type="button" value="Çıkış Yap"/>	<input type="button" value="Giris Yap"/>

Silme modülü kullanıldıktan sonra Veritabanı görüntüsü

Start Page		Schema Editor		Kasa	
Database: proje		Schema: public		Table: Kasa	
		kasaNo	personelNo	gunTarihi	magazaNo
1	<input type="checkbox"/>	001	001	2020-05-27	001

4.Güncelleme

Güncelleme modülü kullanılmadan önce Veritabanı görüntüsü

Start Page		Schema Editor		Musteri-Satis			
Database: proje		Schema: public		Table: Musteri			
		musteriNo	kimlikNo	ad	soyad	yasadigiYer	yapilanAlisveris
1	<input type="checkbox"/>	001	12345678911	Yunus	Yunus	1	2
2	<input type="checkbox"/>	002	12345678922	Emre	Emre	1	0
3	<input type="checkbox"/>	003	12345678933	Eminler	Eminler	6	0

Güncelleme Modülüne güncellenmek istenen musterino ve yeni musterino yazılıyor “Degistir” butonuna tıklanıyor.

Müşteri No yenileme

Eski Musteri No :

Yeni Musteri No :

Müşteri No yenileme

Eski Musteri No :

Yeni Musteri No :

Güncelleme modülü kullanıldıktan sonra Veritabanı görüntüsü

Start Page		Schema Editor		Musteri-Satis			
Database: proje		Schema: public		Table: Musteri			
		musteriNo	kimlikNo	ad	soyad	yasadigiYer	yapilanAlisveris
1	<input type="checkbox"/>	002	12345678922	Emre	Emre	1	0
2	<input type="checkbox"/>	003	12345678933	Eminler	Eminler	6	0
3	<input type="checkbox"/>	006	12345678911	Yunus	Yunus	1	2

6- Uygulamanın Kaynak Kodları

github.com/yunuseminler/SakaryaUniversity/tree/master/C%20%20Uygulamalar%C4%B1/Proje1

7- Videonun Adresi

<https://www.youtube.com/watch?v=BYU1jNRmYWs>