

VERİ TABANI YÖNETİM SİSTEMLERİ

Eyüp DENİZ- 201816044

PETROL İSTASYONU OTOMASYONU

25.12.2022

GİRİŞ

Projemiz adından da anlaşılacağı üzere bir petrol istasyonunun ihtiyaçlarını karşılamak için geliştirilmiştir. Bu proje istasyonda pompadan yapılan satışların ve pompadan satış yapan saha personelinin tüm satışlarının takibini ve raporlamasını yapabilir. İstasyondaki akaryakıt ve LPG pompalarına gerçekleşen tüm satışlar direkt olarak sisteme girer. Tüm işlemler elektronik olarak kayda geçer. Personelin işe alma ve işten çıkarma işlemleri gerçekleştirilebilir. Ayrıca işletmenin günlük aylık ve yıllık kar hesabını hem grafik hem de sayısal olarak gösterir. Bu sistemle tüm pompaların birim fiyatları ve özel müşterilere uygulanmak istenen fiyat indirim programları, otomasyon üzerinden yönetilir. Müşteriler işletmenin belirleyeceği standartlara göre sınıflandırılır ve indirim uygulanır. Tüm değişiklikler geçmişe dönük raporlanabilir. Veri tabanındaki bilgileri silebilir, değiştirebilir ve ekleme yapabilir.

Normalizasyon

Projemizi 1NF, 2NF ve 3NF normalizasyon kuralları çerçevesinde değerlendireceğiz.

1NF

Tüm tablolarda benzersiz olarak tanımlanmış bir sütun bulunmaktadır. Hiçbir tablo da aynı tür iki bilginin bulunduğu bir sütun yoktur. Her sütun kendi bilgisini tutar. Örneğin “akyaakitturu” olarak tanımlanmış sütunda sadece akaryakıt türleri tutulmuştur.

2NF

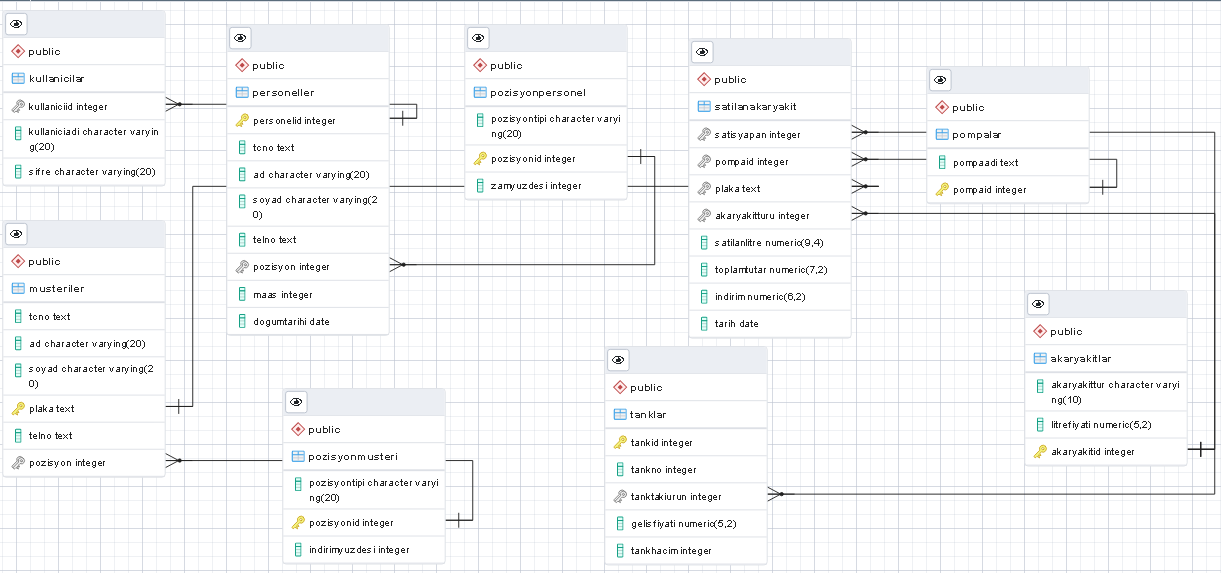
1NF normalizasyon kurallarını sağlamakla beraber. Tablolarda primary key bulunmaktadır. Tablolar arasında ki ilişkiler yabancıl anahtarlar kullanılarak yapılmıştır. Anahtar olmayan sütunlarla bağlantı olmadığı için gereksiz veri tekrarına düşülmemiştir. Kavramsal olarak yakın olan değişkenler aynı tabloda gösterilmiştir. Örneğin: müşteriler, tanklar

3NF

2NF’e uygun olduğundan emin olduktan sonra bakmamız gerek kısımdır. Anahtar olmayan hiçbir sütun birbirine bağlanmamıştır.

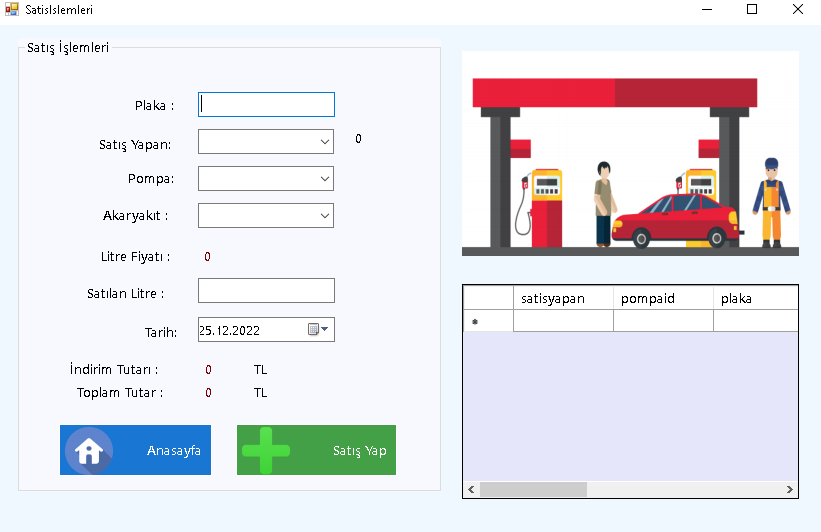
Tablolar ayrıştırılarak küçültülmüştür ancak bu durumda çok fazla tablo ve ilişki içereceğinden tercih edilmemiştir. Normalizasyon kurallarını az da olsa aştığımız tek kısımdır.

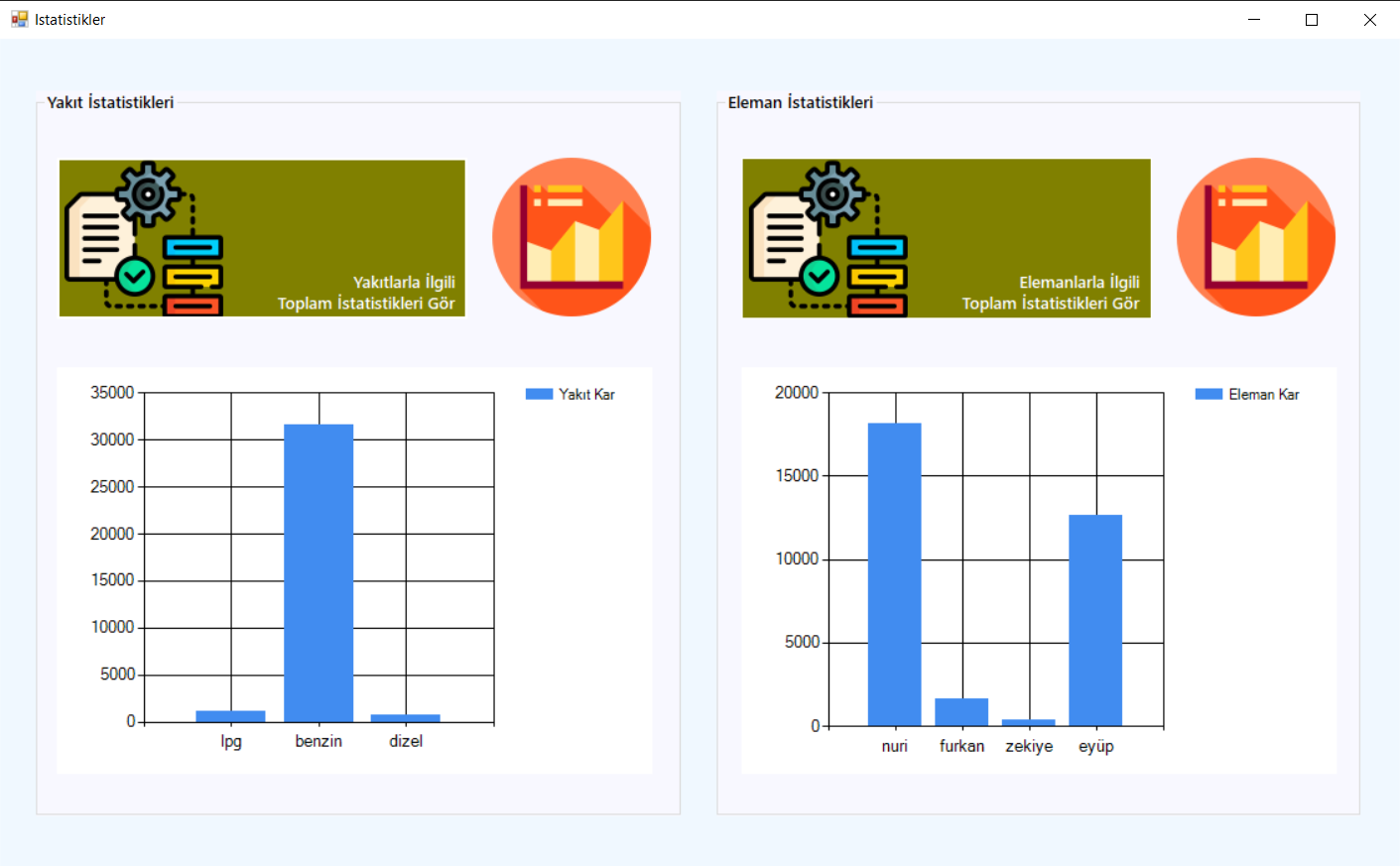
ER DİYAGRAMI



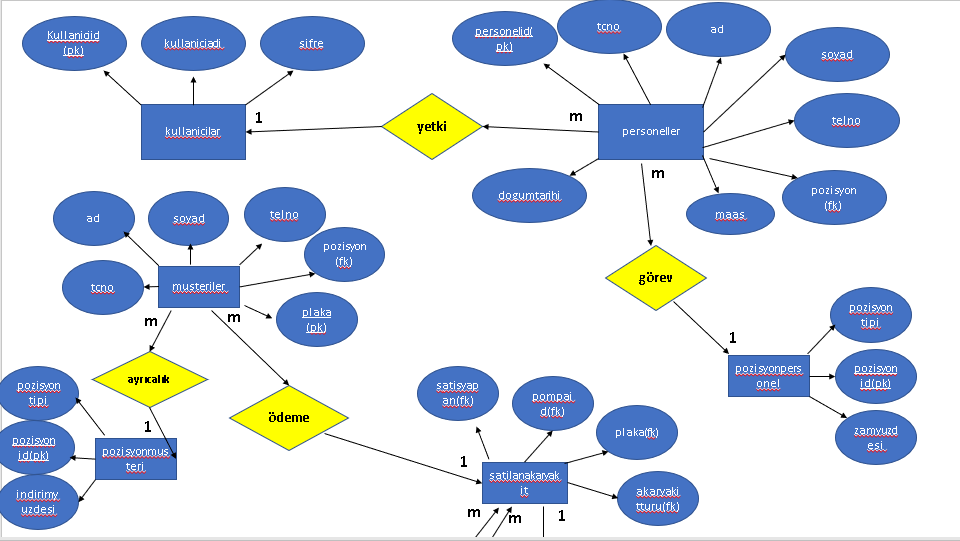
FORMLAR

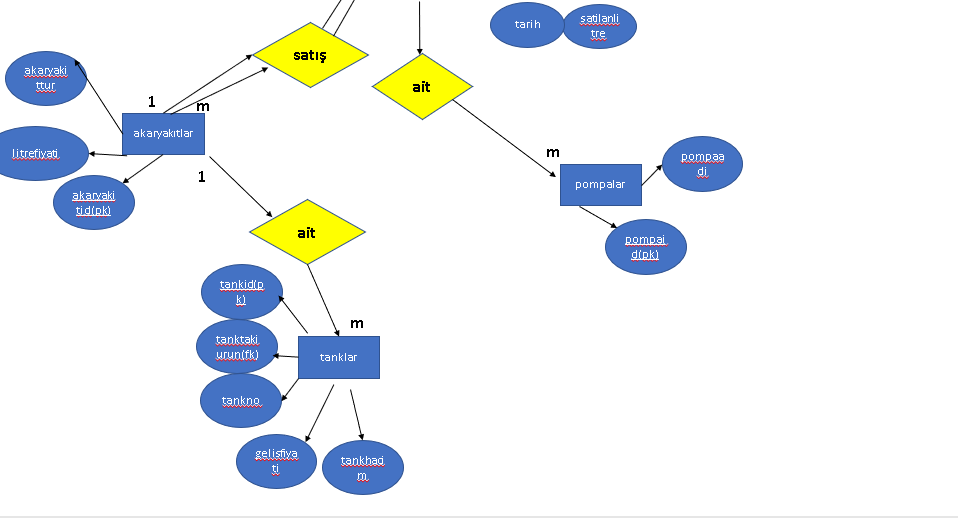






KAVRAMSAL TASARIM





Mantıksal Tasarım

Mantıksal tasarım kısmında veri eklemesi silinmesi ve güncellenmesi durumunda oluşabilecek mantıksal hataların önüne geçilmeye çalışıldı.

Fiziksel Veri Tabanı

Veri tabanında her bir sütun için gerekli olan değişken tipi belirlenip gereksinimler doğrultusunda en verimli durum için son ayar yapılmıştır.

**FONKSİYONLAR**

**create procedure satisislemlerisatisyap(p1 int,p2 decimal (5,2),p4 text,p5 int,p6 decimal(9,4),p7 int,p8 date,p9 decimal(7,2))**

**language plpgsql**

**as**

**$$**

**BEGIN**

**insert into satilanakaryakit(akaryakitturu,indirim,plaka,pompano,satilanlitre,satisyapan,tarih,toplamtutar)**

**values(p1,p2,p4,p5,p6,p7,p8,p9);**

**end;**

**$$;**

**-----------------------------------------------------------------------------------------------------------------**

**create procedure musteriekleme(p1 text,p2 Varchar(20),p3 Varchar(20),p4 text,p5 int,p6 text)**

**language plpgsql**

**as**

**$$**

**BEGIN**

**insert into musteriler(tcno,ad,soyad,telno,pozisyon,plaka)**

**values(p1,p2,p3,p4,p5,p6);**

**end;**

**$$;**

**-----------------------------------------------------------------------------------------------------------------**

**create procedure personelekleme(p1 text,p2 Varchar(20),p3 Varchar(20),p4 text,p5 int,p6 int,p7 int,p8 date)**

**language plpgsql**

**as**

**$$**

**BEGIN**

**insert into personeller(tcno,ad,soyad,telno,pozisyon,maas,personelid,dogumtarihi)**

**values(p1,p2,p3,p4,p5,p6,p7,p8);**

**end;**

**$$;**

**-----------------------------------------------------------------------------------------------------------------**

**create procedure akaryakitekleme(p1 int,p2 Varchar(10),p3 decimal(5,2))**

**language plpgsql**

**as**

**$$**

**BEGIN**

**insert into akaryakitlar(akaryakitid,akaryakittur,litrefiyati)**

**values(p1,p2,p3);**

**end;**

**$$;**

**-----------------------------------------------------------------------------------------------------------------------**

**create procedure yakitseviyesigetir()**

**language plpgsql**

**as**

**$$**

**BEGIN**

**select akaryakittur,tankhacim from akaryakitlar as a inner join tanklar as t**

**on a.akaryakitid= t.tanktakiurun;**

**end;**

**$$;**

**-----------------------------------------------------------------------------------------------------------------**

**Günlük Eleman Kar için Fonksiyon**

**create or replace function kargetir()**

**returns table**

**(adsutun varchar(20),**

**karsutun double precision)**

**language plpgsql**

**as $$**

**BEGIN**

**return query**

**select p.ad,sum(satilanlitre\*(a.litrefiyati-t.gelisfiyati)) as kar**

**from satilanakaryakit as s**

**inner join personeller as p**

**on p.personelid=s.satisyapan**

**inner join akaryakitlar as a**

**on s.akaryakitturu=a.akaryakitid**

**inner join tanklar as t**

**on t.tanktakiurun=a.akaryakitid**

**group by p.personelid,s.tarih**

**having s.tarih=current\_date;**

**end;**

**$$;**

**select \* from kargetir()**

**---Günlük Toplam Satılan Akaryakıt Miktarını Gösteren Fonksiyon---**

**create or replace function akaryakitgetir()**

**returns table**

**(akaryakittipi varchar(10),**

**tlitre numeric)**

**language plpgsql**

**as $$**

**BEGIN**

**return query**

**select akaryakittur, sum(satilanlitre) as litre**

**from akaryakitlar as a inner join satilanakaryakit as s**

**on a.akaryakitid = s.akaryakitturu**

**group by a.akaryakitid,s.tarih**

**having s.tarih = current\_date;**

**end;**

**$$;**

**select \* from akaryakitgetir()**

**---Kar ve Ciro Getiren Fonksiyon---**

**create or replace function gunlukcirogetir()**

**returns table**

**(ciro numeric)**

**language plpgsql**

**as $$**

**BEGIN**

**return query**

**select sum(toplamtutar) as ciro**

**from satilanakaryakit where tarih=current\_date**

**group by tarih;**

**end;**

**$$;**

**select \* from gunlukcirogetir()**

**create or replace function gunlukkargetir()**

**returns table**

**(kar double PRECISION)**

**language plpgsql**

**as $$**

**BEGIN**

**return query**

**select sum(satilanlitre\*(a.litrefiyati-t.gelisfiyati)) as kar**

**from satilanakaryakit as s**

**inner join akaryakitlar as a**

**on s.akaryakitturu=a.akaryakitid**

**inner join tanklar as t**

**on t.tanktakiurun=a.akaryakitid**

**group by s.tarih**

**having s.tarih=current\_date;**

**end;**

**$$;**

**select \* from gunlukkargetir()**

**----------------------------------------------------------------------------------------**

**---Yakıt Kar için Fonksiyon---**

**create or replace function yakitkaristatistik()**

**returns table**

**( akaryakit varchar(10),**

**kar double PRECISION)**

**language plpgsql**

**as $$**

**BEGIN**

**return query**

**select akaryakittur,sum(satilanlitre\*(a.litrefiyati-t.gelisfiyati))**

**from satilanakaryakit as s**

**inner join akaryakitlar as a**

**on s.akaryakitturu=a.akaryakitid**

**inner join tanklar as t**

**on t.tanktakiurun=a.akaryakitid**

**group by a.akaryakitid;**

**end;**

**$$;**

**---Eleman Kar İçin Fonksiyon---**

**create or replace function elemankaristatistik()**

**returns table**

**(adi varchar(20),**

**kar double PRECISION)**

**language plpgsql**

**as $$**

**BEGIN**

**return query**

**select ad,sum(satilanlitre\*(a.litrefiyati-t.gelisfiyati))**

**from satilanakaryakit as s**

**inner join akaryakitlar as a**

**on s.akaryakitturu=a.akaryakitid**

**inner join tanklar as t**

**on t.tanktakiurun=a.akaryakitid**

**inner join personeller as p**

**on p.personelid=s.satisyapan**

**group by p.ad;**

**end;**

**$$;**

**select \* from elemankaristatistik()**

**----------------------------------------------------------------------------------------------------------Pozisyon Güncelleme Triggeri**

**create view pozisyonguncelleview**

**as**

**select pozisyon,personelid from personeller where personelid in (SELECT satisyapan**

**FROM satilanakaryakit**

**GROUP BY satisyapan**

**HAVING SUM(satilanlitre) > 10000) and pozisyon=4;**

**CREATE OR REPLACE FUNCTION pozisyonguncelle()**

**RETURNS TRIGGER AS $$**

**declare**

**personel int;**

**pozisyonn int;**

**indexsayisi int;**

**BEGIN**

**indexsayisi:=(select count(pozisyon) from personeller where personelid in (SELECT satisyapan**

**FROM satilanakaryakit**

**GROUP BY satisyapan**

**HAVING SUM(satilanlitre) > 10000) and pozisyon=4);**

**loop**

**exit when indexsayisi=0;**

**personel:=(select personelid from pozisyonguncelleview order by personelid limit 1);**

**pozisyonn:=( select pozisyon from pozisyonguncelleview order by personelid limit 1);**

**if pozisyonn=4**

**THEN**

**UPDATE personeller**

**SET pozisyon = pozisyon - 1 where personelid= (personel);**

**else**

**end if;**

**indexsayisi=indexsayisi - 1;**

**end loop;**

**RETURN NEW;**

**END;**

**$$ LANGUAGE plpgsql;**

**CREATE TRIGGER pozisyonguncelletrigger**

**AFTER INSERT ON satilanakaryakit**

**FOR EACH ROW**

**EXECUTE PROCEDURE pozisyonguncelle();**

**----------------------------------------------------------------------------------------------------------**

**Toplam Veriler İçin View**

**create view toplamdosya**

**as**

**select ad,sum(satilanlitre) as sattigilitre,sum(satilanlitre\*(a.litrefiyati-t.gelisfiyati)) as ettigikar,sum(s.indirim) as indirimlisatis,maas as gider,**

**sum((s.satilanlitre\*(a.litrefiyati-t.gelisfiyati)-s.indirim))-(maas) as netkar**

**from satilanakaryakit as s**

**inner join akaryakitlar as a**

**on s.akaryakitturu=a.akaryakitid**

**inner join tanklar as t**

**on t.tanktakiurun=a.akaryakitid**

**inner join personeller as p**

**on p.personelid=s.satisyapan**

**group by p.personelid**

**---Gunluk Veriler için View---**

**create view gunlukdosya**

**as**

**select ad,sum(satilanlitre) as sattigilitre,sum(s.satilanlitre\*(a.litrefiyati-t.gelisfiyati)-s.indirim) as ettigikar,maas/30 as gider,**

**sum((s.satilanlitre\*(a.litrefiyati-t.gelisfiyati)-s.indirim))-(maas/30) as netkar**

**from satilanakaryakit as s**

**inner join akaryakitlar as a**

**on s.akaryakitturu=a.akaryakitid**

**inner join tanklar as t**

**on t.tanktakiurun=a.akaryakitid**

**inner join personeller as p**

**on p.personelid=s.satisyapan**

**group by p.personelid,s.tarih**

**having s.tarih=current\_date;**

**--------- GİRİŞ--------**

**create or replace function yoneticigiris()**

**returns table**

**( kullaniciad varchar(20),sifree varchar(20),yoneticiid int )**

**language plpgsql**

**as $$**

**BEGIN**

**return query**

**select kullaniciadi,sifre,personelid from personeller as p inner join pozisyonpersonel as po**

**on p.pozisyon=po.pozisyonid**

**inner join kullanicilar as k**

**on k.kullaniciid=p.personelid**

**group by p.personelid,po.pozisyontipi,k.kullaniciadi,k.sifre**

**having pozisyontipi='müdür';**

**end;**

**$$;**

**create or replace function kullanicigiris()**

**returns table**

**(kullaniciad varchar(20),sifree varchar(20), kullaniciid int )**

**language plpgsql**

**as $$**

**BEGIN**

**return query**

**select kullaniciadi,sifre,personelid from personeller as p inner join pozisyonpersonel as po**

**on p.pozisyon=po.pozisyonid**

**inner join kullanicilar as k**

**on k.kullaniciid=p.personelid**

**group by p.personelid,po.pozisyontipi,k.kullaniciadi,k.sifre**

**having pozisyontipi='kasiyer';**

**end;**

**$$;**