Veri Tabanı

İsim: Eren

Soy isim: Kara

Numara: B211210031

Şube: 1-B

Dersi Veren Öğretim Görevlisi: İsmail ÖZTEL

E-posta: errennkaaraa@hotmail.com

Uygulamanın Tanıtımı

Oto sanayi dükkanının giriş çıkışlarını kontrol eden bir uygulama tasarladım. Uygulamada tamire giren bir aracın kaç gün kaldığından, tamir ücretine; personel maaşlarından stokta kaç ürün kaldığına kadar birçok işlemin uygulama sayesinde kolayca hesaplanmasını, eklenip, çıkartılmasını yani kontrol edilmesini sağladım. Bunu yaparken ise postgresql veri tabanını kullandım.

İş Kuralları

- İş yeri birden fazla dükkândan oluşur.
- Dükkânın adı, adresi, telefonu ve patronun kim olduğu bilgileri bulunur.
- Dükkân faturalar oluşturur. Faturaların tutarı, tarihi, tipi ve hangi mağazaya ait olduğu tutulur.
- Dükkânın gelir ve giderleri tutulur. Gelir ve giderlerin tutar bilgisi ve tarih bilgisi vardır.
- Faturaların ne faturası olduğu bilgisi ayrı tutulur.
- İnsanların T.C. No, isim, soy isim, telefon, adres bilgileri bulunur.
- Kişiler ya personel ya müşteri ya da tedarikçidir. Bunlar haricinde bir kişi bulunmamaktadır.
- Müşterinin vergi adresi tutulur.
- Personellere dükkân tarafından ücret ödenir.
- Ücret; maaş, SSK, yol, yemek ve toplam ücretten oluşur.
- Dükkân ürün sipariş eder.
- Bir siparişte en az bir ürün olmak zorundadır ancak birden fazla üründe olabilir. Bir ürün birden fazla siparişte bulunabilir ya da hiçbir siparişte bulunmayabilir.
- Siparişlerdeki ürünlerin kaç adet olduğu ve birim fiyatları da saklanmalıdır.
- Ürünlerin ürün kodu, ürün ismi, ürün modeli ve üründen kaç tane kaldığı bilgileri tutulur.
- Müşterilere ürün satılabilir. Satışların dükkândan satıldığı bilgisi tutulur.
- Bir satışta birden fazla ürün bulunabilir. Bu satıştaki ürünleri adedi ve birim fiyatı tutulur.
- Müşterilerin araçları bulunur. Bu araçların plaka, model, km, bilgileri tutulur.
- Araçlar personeller tarafından tamir edilirler.
- Tamir hakkında aracın tamirine başlanma tarihi, aracın tamirinin sona erdiği tarih, tamirin ücreti ve tamirde ne yapıldığı tutulur.
- Bir araç birçok defa tamir edilebilir veya hiç tamir edilmez. Bir tamir yalnızca bir araca yapılır.
- Bir personel birden fazla aracı tamir edebilir veya hiçbir aracı tamir etmeyebilir. Bir tamir en az bir personel tarafından yapılmak zorundadır fakat birden fazla personel tarafından da yapılabilir.
- Bir müşteri birden fazla araca sahip olabilir veya hiçbir araca sahip olmayabilir. Bir araç yalnızca bir müşteriye aittir.
- Bir müşteriye birden fazla satış yapılabilir veya hiçbir satış yapılmaz. Bir satış yalnızca bir müşteriye yapılır.
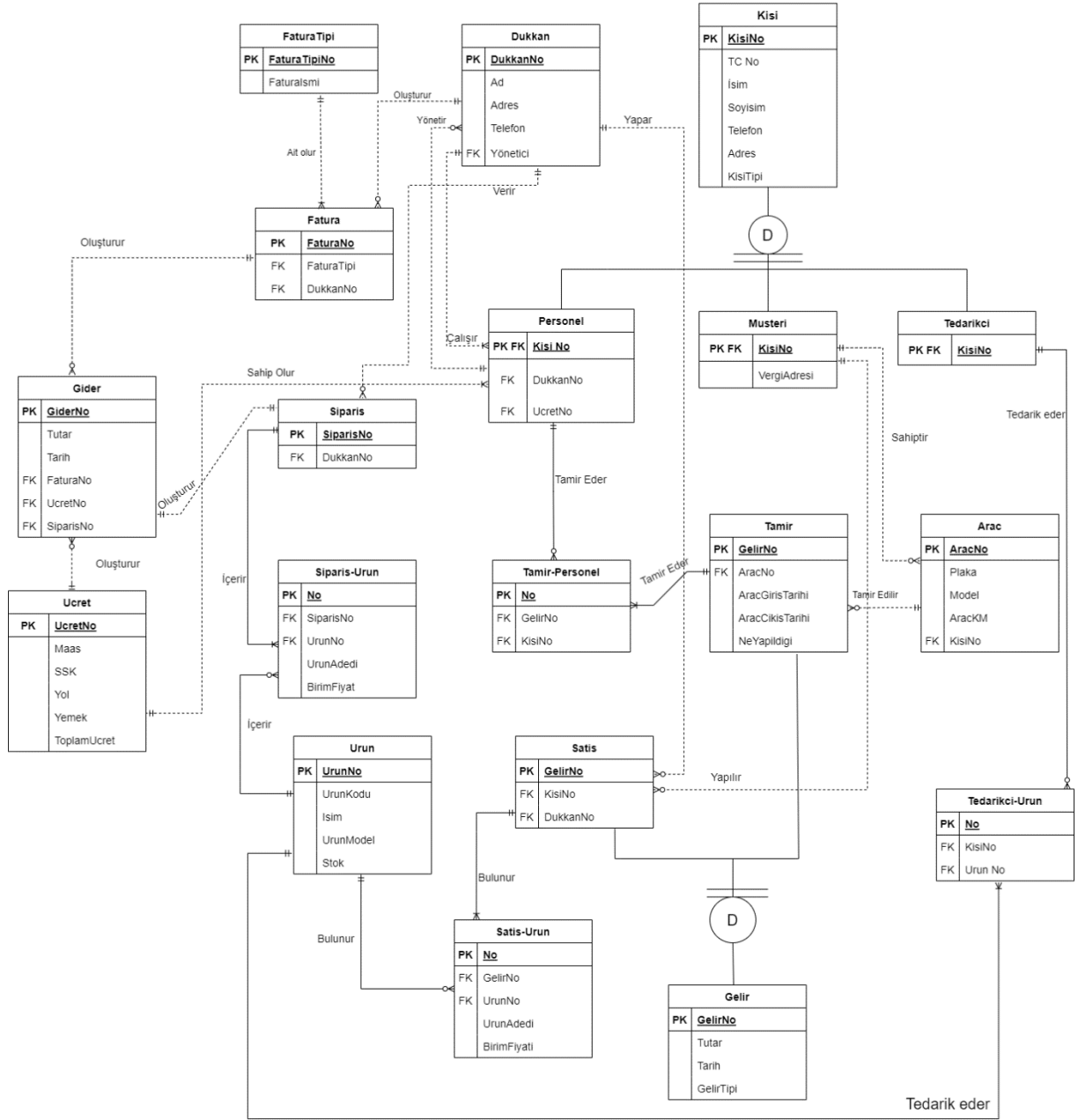
- Bir dükkân birden fazla satış yapabilir veya hiçbir satış yapamaz. Bir satış yalnızca bir dükkândan yapılır.
- Bir satışta en az bir ürün bulunmak zorundadır ancak birden fazla üründe bulunabilir. Bir ürün birden fazla satışta bulunabilir veya hiçbir satışta bulunmayabilir.
- Bir ürün birden fazla tedarikçi tarafından tedarik edilebilir veya sadece bir tedarikçiden tedarik edilir. Bir tedarikçi birden fazla ürün tedarik edebilir veya hiçbir ürün tedarik etmeyebilir.
- Bir dükkân birden fazla sipariş verebilir veya hiçbir sipariş vermez. Bir sipariş yalnızca bir dükkândan verilebilir.
- Bir dükkânın en az bir personeli vardır. Bir personel yalnızca bir dükkânda çalışır.
- Bir dükkân yalnızca bir personel tarafından yönetilir. Bir personel birden fazla dükkânı yönetebilir veya hiçbir dükkânı yönetmez.
- Bir ücret en az bir personele aittir. Bir personelin yalnızca bir ücreti vardır.
- Bir dükkân birden fazla fatura oluşturabilir veya hiçbir fatura oluşturmaz. Bir fatura yalnızca bir dükkân tarafından oluşturulur.
- Bir fatura yalnızca bir fatura tipine ait olabilir. Bir fatura tipi birden fazla faturaya ait olabilir.
- Bir fatura birden fazla gider oluşturabilir veya hiç oluşturmaz. Bir gider yalnızca bir fatura tarafından oluşturulur.
- Bir sipariş yalnızca bir gider oluşturur. Bir gider yalnızca bir sipariş tarafından oluşturulur.
- Bir ücret birden fazla gider oluşturabilir veya hiç oluşturmaz. Bir gider yalnızca bir ücret tarafından oluşturulabilir.
- Gelirler ya satış ya da tamirdir. Bunlar haricinde bir gelir bulunmamaktadır.

İlişkisel Şema

- Arac (<u>AracNo: int</u>, Kilometre: int, <u>KisiNo: int</u>, Model: varchar(50), Plaka: char(8))
- Dukkan (<u>DukkanNo: int</u>, Ad: varchar(60),Adres: text, Telefon: char(11), <u>Yonetici: int</u>)
- Fatura (<u>FaturaNo: int</u>, <u>DukkanNo: int</u>, <u>FaturaTipi: int</u>)
- FaturaTipi (<u>FaturaTipiNo: int</u>, FaturaIsmi: varchar(30))
- Gelir (<u>GelirNo: int</u>, Tutar: double precision, Tarih: date, GelirTİpi: char(1))
- Gider (<u>GiderNo: int</u>, Tutar: double precision, Tarih: date, <u>FaturaNo: int</u>, <u>SiparisNo: int</u>, <u>UcretNo: int</u>)
- Kisi (<u>KisiNo: int</u>, Isim: varchar(50), Soyisim: varchar(50), TCNo: char(11), Telefon: char(11), Adres: text, KisiTipi: char(1))
- Musteri (<u>KisiNo: int</u>, VergiAdresi: text)
- Personel (<u>KisiNo: int</u>, <u>DukkanNo: int</u>, <u>UcretNo: int</u>)
- Satis (<u>GelirNo: int</u>, <u>DukkanNo: int</u>, <u>KisiNo: int</u>)
- Satis-Urun (<u>No: int</u>, <u>GelirNo: int</u>, <u>UrunNo: int</u>, UrunAdedi: smallint, BirimFiyat: double precision)
- Siparis (<u>SiparisNo: int</u>, <u>DukkanNo: int</u>)
- Siparis-Urun (<u>No: int</u>, <u>SiparisNo: int</u>, <u>UrunNo: int</u>, UrunAdedi: smallint, BirimFiyat: double precision)
- Tamir (<u>GelirNo: int</u>, <u>AracNo: int</u>, AracGirisTarihi: date, AracCikisTarihi: date, NeYapildigi: text)
- Tamir-Personel (<u>No: int</u>, <u>KisiNo: int</u>, <u>GelirNo: int</u>)
- Tedarikci (<u>KisiNo: int</u>)
- Tedarikci-Urun (<u>No: int</u>, <u>KisiNo: int</u>, <u>UrunNo: int</u>)

- Ucret (<u>UcretNo: int</u>, Maas: double precision, SSK: double precision, Yol: double precision, Yemek: double precision, ToplamUcret: double precision)
- Urun (<u>UrunNo: int</u>, Isim: varchar(40), UrunModel varchar(40), Stok: int)

Crow's Foot Gösterim

FaturaTipi

PK	FaturaTipiNo
	FaturaIsmi

Dukkan

PK	DukkanNo
	Ad
	Adres
	Telefon
FK	Yönetici

Kisi

PK	KisiNo
	TC No
	İsim
	Soyisim
	Telefon
	Adres
	KisiTipi

Fatura

PK	FaturaNo
FK	FaturaTipi
FK	DukkanNo

Gider

PK	GiderNo
	Tutar
	Tarih
FK	FaturaNo
FK	UcretNo
FK	SiparisNo

Siparis

PK	SiparisNo
FK	DukkanNo

Personel

PK FK	Kisi No
FK	DukkanNo
FK	UcretNo

Musteri

PK FK	KisiNo
	VergiAdresi

Tedarikci

PK FK	KisiNo

Ucret

PK	UcretNo
	Maas
	SSK
	Yol
	Yemek
	ToplamUcret

Siparis-Urun

PK	No
FK	SiparisNo
FK	UrunNo
	UrunAdedi
	BirimFiyat

Tamir-Personel

PK	No
FK	GelirNo
FK	KisiNo

Tamir

PK	GelirNo
FK	AracNo
	AracGirisTarihi
	AracCikisTarihi
	NeYapildigi

Arac

PK	AracNo
	Plaka
	Model
	AracKM
FK	KisiNo

Urun

PK	UrunNo
	UrunKodu
	Isim
	UrunModel
	Stok

Satis

PK	GelirNo
FK	KisiNo
FK	DukkanNo

Tedarikci-Urun

PK	No
FK	KisiNo
FK	Urun No

Satis-Urun

PK	No
FK	GelirNo
FK	UrunNo
	UrunAdedi
	BirimFiyati

Gelir

PK	GelirNo
	Tutar
	Tarih
	GelirTipi

Relationship labels: Oluşturur, Ait olur, Yönetir, Yapar, Verir, Oluşturur, Sahip Olur, Çalışır, Oluşturur, İçerir, Tamir Eder, Tamir Eder, Sahiptir, Tedarik eder, Tamir Edilir, İçerir, Bulunur, Bulunur, Yapılır, Tedarik eder

SQL İfadeleri

```sql
--
-- PostgreSQL database dump
--

-- Dumped from database version 14.5
-- Dumped by pg_dump version 15rc2

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;


--
-- Name: public; Type: SCHEMA; Schema: -; Owner: postgres
--

-- *not* creating schema, since initdb creates it


ALTER SCHEMA public OWNER TO postgres;


--
-- Name: my_type; Type: TYPE; Schema: public; Owner: postgres
--

CREATE TYPE public.my_type AS (
        kisi_id integer,
        isim character varying(50)
);


ALTER TYPE public.my_type OWNER TO postgres;


--
-- Name: my_type2; Type: TYPE; Schema: public; Owner: postgres
--

CREATE TYPE public.my_type2 AS (
        gelir bigint,
        gider bigint,
        karzarar bigint
);


ALTER TYPE public.my_type2 OWNER TO postgres;


--
-- Name: my_type3; Type: TYPE; Schema: public; Owner: postgres
--

CREATE TYPE public.my_type3 AS (
        "GelirTutar" bigint,
        "GiderTutar" bigint,
        "Kar-Zarar" bigint,
        "Tarih" date
);
```

```sql
ALTER TYPE public.my_type3 OWNER TO postgres;


--
-- Name: my_type4; Type: TYPE; Schema: public; Owner: postgres
--

CREATE TYPE public.my_type4 AS (
        "GelirTutar" double precision,
        "GiderTutar" double precision,
        "Kar-Zarar" double precision,
        "Tarih" date
);


ALTER TYPE public.my_type4 OWNER TO postgres;


--
-- Name: my_type5; Type: TYPE; Schema: public; Owner: postgres
--

CREATE TYPE public.my_type5 AS (
        "GelirTutar" double precision,
        "GiderTutar" double precision,
        "Kar-Zarar" double precision,
        "Tarih1" date,
        "Tarih2" date
);


ALTER TYPE public.my_type5 OWNER TO postgres;


--
-- Name: my_type6; Type: TYPE; Schema: public; Owner: postgres
--

CREATE TYPE public.my_type6 AS (
        "GelirTutar" double precision,
        "GiderTutar" double precision,
        "Kar-Zarar" double precision,
        "Baslangic" date,
        "Bitis" date
);


ALTER TYPE public.my_type6 OWNER TO postgres;


--
-- Name: KisiVarmiKontrolEt(integer); Type: FUNCTION; Schema: public; Owner: postgres
--

CREATE FUNCTION public."KisiVarmiKontrolEt"(kisino integer) RETURNS boolean
    LANGUAGE plpgsql
    AS $$
declare
varmi boolean;
begin
varmi:=false;
if (select "KisiNo" from "Personel" where "KisiNo"=kisino) is not null then
varmi:=true;
elseif(select "KisiNo" from "Musteri" where "KisiNo"=kisino) is not null then
varmi:=true;
elseif(select "KisiNo" from "Tedarikci" where "KisiNo"=kisino)is not null then
varmi:=true;
end if;
return varmi;
end;
$$;


ALTER FUNCTION public."KisiVarmiKontrolEt"(kisino integer) OWNER TO postgres;


--
-- Name: TrigFuncSatis-Urun_GelirGuncelle(); Type: FUNCTION; Schema: public; Owner: postgres
--

CREATE FUNCTION public."TrigFuncSatis-Urun_GelirGuncelle"() RETURNS trigger
```

```
    LANGUAGE plpgsql
    AS $$
Declare
fiyat double precision;
tutar int;
Begin
tutar:=(select "Tutar" from "Gelir"
where "Gelir"."GelirNo"=new."GelirNo");
if tutar is null then
fiyat:= ((new."UrunAdedi") * (new."BirimFiyat"));
else
fiyat:= tutar+((new."UrunAdedi") * (new."BirimFiyat"));
end if;
Update "Gelir" set "Tutar"=fiyat where "GelirNo"=new."GelirNo" ;
return new;
End;
$$;


ALTER FUNCTION public."TrigFuncSatis-Urun_GelirGuncelle"() OWNER TO postgres;


--
-- Name: TrigFuncSatisStokDüsme(); Type: FUNCTION; Schema: public; Owner: postgres
--

CREATE FUNCTION public."TrigFuncSatisStokDüsme"() RETURNS trigger
    LANGUAGE plpgsql
    AS $$
declare
stok int;
begin
stok:=(select "Stok" from "Urun" where "UrunNo"=new."UrunNo");
update "Urun" set "Stok"=stok-new."UrunAdedi" where "UrunNo"=new."UrunNo";
return new;
end;
$$;


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


--
-- Name: TrigFuncSiparis-Urun_GiderGuncelle(); Type: FUNCTION; Schema: public; Owner: postgres
--

CREATE FUNCTION public."TrigFuncSiparis-Urun_GiderGuncelle"() RETURNS trigger
    LANGUAGE plpgsql
    AS $$
Declare
fiyat double precision;
tutar int;
Begin
tutar:=(select "Tutar" from "Gider"
where "Gider"."SiparisNo"=new."SiparisNo");
if tutar is null then
fiyat:= ((new."UrunAdedi") * (new."BirimFiyat"));
else
fiyat:= tutar+((new."UrunAdedi") * (new."BirimFiyat"));
end if;
Update "Gider" set "Tutar"=fiyat where "SiparisNo"=new."SiparisNo" ;
return new;
End;
$$;


ALTER FUNCTION public."TrigFuncSiparis-Urun_GiderGuncelle"() OWNER TO postgres;


--
-- Name: TrigFuncSiparisStokEkleme(); Type: FUNCTION; Schema: public; Owner: postgres
--

CREATE FUNCTION public."TrigFuncSiparisStokEkleme"() RETURNS trigger
    LANGUAGE plpgsql
    AS $$
Declare
stok int;
Begin
```

```sql
        stok:=(select "Stok" from "Urun" where "UrunNo"=new."UrunNo");
        update "Urun" set "Stok"=stok+new."UrunAdedi" where "UrunNo"=new."UrunNo" ;


return New;
End;
$$;


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


--
-- Name: TrigFuncUcretToplamHesapla(); Type: FUNCTION; Schema: public; Owner: postgres
--

CREATE FUNCTION public."TrigFuncUcretToplamHesapla"() RETURNS trigger
    LANGUAGE plpgsql
    AS $$
declare
toplam int;
Begin

if new."Maas"<>old."Maas" or new."Yol"<>old."Yol" or new."Yemek"<>old."Yemek" or new."SSK"<>old."SSK" or
new."ToplamUcret" is NULL then
toplam:=(new."Maas"+new."Yol"+new."SSK"+new."Yemek");
update "Ucret" set "ToplamUcret"=toplam where "UcretNo"=new."UcretNo";
end if;
return new;
End;
$$;


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


--
-- Name: get_data_kisi(integer); Type: FUNCTION; Schema: public; Owner: postgres
--

CREATE FUNCTION public.get_data_kisi(kisino integer) RETURNS character varying
    LANGUAGE plpgsql
    AS $$
declare
isim varchar(50);
soyisim varchar(50);
ad varchar(110);
begin
isim:=(select "Isim" from "Kisi" where "KisiNo"=kisino);
soyisim:=(select "Soyisim" from "Kisi" where "KisiNo"=kisino);
ad:=isim||' '||soyisim;

return ad;
end;
$$;


ALTER FUNCTION public.get_data_kisi(kisino integer) OWNER TO postgres;


--
-- Name: karzararhesapla(integer); Type: FUNCTION; Schema: public; Owner: postgres
--

CREATE FUNCTION public.karzararhesapla(sayi integer) RETURNS TABLE("GelirTutar" double precision, "GiderTutar"
double precision, "Kar-Zarar" double precision, "Tarih1" date, "Tarih2" date)
    LANGUAGE plpgsql
    AS $$
declare
begin
if (select sum("Tutar")as toplamgelir from "Gelir" where age("Gelir"."Tarih")BETWEEN INTERVAL '1 month'*(sayi-1) and
interval '1month'*sayi) is NULL then
return query
select "toplamgelir", "toplamgider",0-"toplamgider" as "karzarar" ,date(current_date-interval '1month'*(sayi)) as
"Baslangic",date(current_date-interval '1month'*(sayi-1)) as "Bitis" from (select sum("Tutar")as toplamgelir from "Gelir" where
age("Gelir"."Tarih")BETWEEN INTERVAL '1 month'*(sayi-1) -interval'1day' and interval '1month'*sayi) as gelir ,(select
sum("Tutar") as toplamgider from "Gider" where age("Gider"."Tarih")BETWEEN INTERVAL '1 month'*(sayi-1)-interval'1day'
and interval '1month'*sayi) as gider;
```

```sql
elseif (select sum("Tutar") as toplamgider from "Gider" where age("Gider"."Tarih")BETWEEN INTERVAL '1 month'*(sayi-1)
and interval '1 month'*sayi) is null then
return query
select "toplamgelir", "toplamgider",toplamgelir-0 as "karzarar",date(current_date-interval '1month'*(sayi)) as
"Baslangic",date(current_date-interval '1month'*(sayi-1)) as "Bitis" from (select sum("Tutar")as toplamgelir from "Gelir" where
age("Gelir"."Tarih")BETWEEN INTERVAL '1 month'*(sayi-1)-interval'1day' and interval '1month'*sayi) as gelir ,(select
sum("Tutar") as toplamgider from "Gider" where age("Gider"."Tarih")BETWEEN INTERVAL '1 month'*(sayi-1)-interval'1day'
and interval '1month'*sayi) as gider;


else
return query
select "toplamgelir", "toplamgider","toplamgelir"-"toplamgider" as "karzarar",date(current_date-interval '1month'*(sayi)) as
"Baslangic",date(current_date-interval '1month'*(sayi-1)) as "Bitis" from (select sum("Tutar")as toplamgelir from "Gelir" where
age("Gelir"."Tarih")BETWEEN INTERVAL '1 month'*(sayi-1)-interval'1day' and interval '1month'*sayi) as gelir ,(select
sum("Tutar") as toplamgider from "Gider" where age("Gider"."Tarih")BETWEEN INTERVAL '1 month'*(sayi-1)-interval'1day'
and interval '1month'*sayi) as gider;
end if;
end;
$$;


ALTER FUNCTION public.karzararhesapla(sayi integer) OWNER TO postgres;

--
-- Name: karzarartablo(integer); Type: FUNCTION; Schema: public; Owner: postgres
--

CREATE FUNCTION public.karzarartablo(sayac integer) RETURNS SETOF public.my_type6
    LANGUAGE plpgsql
    AS $$
declare
begin
for i in 1 .. sayac loop
    return next karzararhesapla(i);
end loop;
end $$;


ALTER FUNCTION public.karzarartablo(sayac integer) OWNER TO postgres;

--
-- Name: stoktavarmi(integer, integer); Type: FUNCTION; Schema: public; Owner: postgres
--

CREATE FUNCTION public.stoktavarmi(urunno integer, urunadet integer) RETURNS boolean
    LANGUAGE plpgsql
    AS $$
declare
varmi boolean;
stok integer;
begin
stok:=(select "Stok" from "Urun" where "UrunNo"=urunno);
if stok>=urunadet then
return true;
else
return false;
end if;
end;
$$;


ALTER FUNCTION public.stoktavarmi(urunno integer, urunadet integer) OWNER TO postgres;

--
-- Name: tamirsilinince(); Type: FUNCTION; Schema: public; Owner: postgres
--

CREATE FUNCTION public.tamirsilinince() RETURNS trigger
    LANGUAGE plpgsql
    AS $$
declare
begin

delete from "Tamir-Personel" where "Tamir-Personel"."GelirNo"=new."GelirNo";
return new;
```

```sql
end;
$$;


ALTER FUNCTION public.tamirsilinince() OWNER TO postgres;

SET default_tablespace = '';

SET default_table_access_method = heap;


--
-- Name: Arac; Type: TABLE; Schema: public; Owner: postgres
--

CREATE TABLE public."Arac" (
    "AracNo" integer NOT NULL,
    "Plaka" character(8) NOT NULL,
    "Model" character varying(50),
    "Kilometre" integer NOT NULL,
    "KisiNo" integer NOT NULL,
    CONSTRAINT "AracKMCheck" CHECK (("Kilometre" >= 0))
);


ALTER TABLE public."Arac" OWNER TO postgres;


--
-- Name: Arac_AracNo_seq; Type: SEQUENCE; Schema: public; Owner: postgres
--

CREATE SEQUENCE public."Arac_AracNo_seq"
    AS integer
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;


ALTER TABLE public."Arac_AracNo_seq" OWNER TO postgres;


--
-- Name: Arac_AracNo_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: postgres
--

ALTER SEQUENCE public."Arac_AracNo_seq" OWNED BY public."Arac"."AracNo";



--
-- Name: Dukkan; Type: TABLE; Schema: public; Owner: postgres
--

CREATE TABLE public."Dukkan" (
    "DukkanNo" integer NOT NULL,
    "Ad" character varying(60),
    "Adres" text,
    "Telefon" character(11),
    "Yonetici" integer
);


ALTER TABLE public."Dukkan" OWNER TO postgres;


--
-- Name: faturaSayac; Type: SEQUENCE; Schema: public; Owner: postgres
--

CREATE SEQUENCE public."faturaSayac"
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;


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

```
--
-- Name: Fatura; Type: TABLE; Schema: public; Owner: postgres
--

CREATE TABLE public."Fatura" (
    "FaturaNo" integer DEFAULT nextval('public."faturaSayac"'::regclass) NOT NULL,
    "FaturaTipi" integer NOT NULL,
    "DukkanNo" integer NOT NULL
);


ALTER TABLE public."Fatura" OWNER TO postgres;


--
-- Name: FaturaTipi; Type: TABLE; Schema: public; Owner: postgres
--

CREATE TABLE public."FaturaTipi" (
    "FaturaTipiNo" integer NOT NULL,
    "FaturaIsmi" character varying(30) NOT NULL
);


ALTER TABLE public."FaturaTipi" OWNER TO postgres;


--
-- Name: FaturaTipi_FaturaTipiNo_seq; Type: SEQUENCE; Schema: public; Owner: postgres
--

CREATE SEQUENCE public."FaturaTipi_FaturaTipiNo_seq"
    AS integer
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;


ALTER TABLE public."FaturaTipi_FaturaTipiNo_seq" OWNER TO postgres;


--
-- Name: FaturaTipi_FaturaTipiNo_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: postgres
--

ALTER SEQUENCE public."FaturaTipi_FaturaTipiNo_seq" OWNED BY public."FaturaTipi"."FaturaTipiNo";


--
-- Name: Tamir; Type: TABLE; Schema: public; Owner: postgres
--

CREATE TABLE public."Tamir" (
    "GelirNo" integer NOT NULL,
    "AracGirisTarihi" date NOT NULL,
    "AracCikisTarihi" date NOT NULL,
    "NeYapildigi" text,
    "AracNo" integer NOT NULL
);


ALTER TABLE public."Tamir" OWNER TO postgres;


--
-- Name: gelirSayac; Type: SEQUENCE; Schema: public; Owner: postgres
--

CREATE SEQUENCE public."gelirSayac"
    AS integer
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;
```

```sql
ALTER TABLE public."gelirSayac" OWNER TO postgres;

--
-- Name: gelirSayac; Type: SEQUENCE OWNED BY; Schema: public; Owner: postgres
--

ALTER SEQUENCE public."gelirSayac" OWNED BY public."Tamir"."GelirNo";


--
-- Name: Gelir; Type: TABLE; Schema: public; Owner: postgres
--

CREATE TABLE public."Gelir" (
    "GelirNo" integer DEFAULT nextval('public."gelirSayac"'::regclass) NOT NULL,
    "Tutar" double precision,
    "Tarih" date NOT NULL,
    "GelirTipi" character(1) NOT NULL,
    CONSTRAINT "GelirCheck" CHECK (("Tutar" >= (0)::double precision)),
    CONSTRAINT "GelirCheckChildFk" CHECK ((("GelirTipi" = 'S'::bpchar) OR ("GelirTipi" = 'T'::bpchar)))
);


ALTER TABLE public."Gelir" OWNER TO postgres;

--
-- Name: giderSayac; Type: SEQUENCE; Schema: public; Owner: postgres
--

CREATE SEQUENCE public."giderSayac"
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;


ALTER TABLE public."giderSayac" OWNER TO postgres;

--
-- Name: Gider; Type: TABLE; Schema: public; Owner: postgres
--

CREATE TABLE public."Gider" (
    "GiderNo" integer DEFAULT nextval('public."giderSayac"'::regclass) NOT NULL,
    "Tutar" double precision,
    "Tarih" date NOT NULL,
    "FaturaNo" integer,
    "UcretNo" integer,
    "SiparisNo" integer,
    CONSTRAINT "GiderTutarCheck" CHECK (("Tutar" >= (0)::double precision))
);


ALTER TABLE public."Gider" OWNER TO postgres;

--
-- Name: Kisi; Type: TABLE; Schema: public; Owner: postgres
--

CREATE TABLE public."Kisi" (
    "KisiNo" integer NOT NULL,
    "TCNo" character(11),
    "Isim" character varying(50),
    "Soyisim" character varying(50),
    "Telefon" character(11),
    "Adres" text,
    "KisiTipi" character(1) NOT NULL,
    CONSTRAINT "KisiCheck" CHECK ((("KisiTipi" = 'M'::bpchar) OR ("KisiTipi" = 'T'::bpchar) OR ("KisiTipi" =
'P'::bpchar)))
);


ALTER TABLE public."Kisi" OWNER TO postgres;

--
```

```sql
-- Name: kisiSayac; Type: SEQUENCE; Schema: public; Owner: postgres
--

CREATE SEQUENCE public."kisiSayac"
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;


ALTER TABLE public."kisiSayac" OWNER TO postgres;


--
-- Name: kisiSayac; Type: SEQUENCE OWNED BY; Schema: public; Owner: postgres
--

ALTER SEQUENCE public."kisiSayac" OWNED BY public."Kisi"."KisiNo";



--
-- Name: Musteri; Type: TABLE; Schema: public; Owner: postgres
--

CREATE TABLE public."Musteri" (
    "KisiNo" integer DEFAULT currval('public."kisiSayac"'::regclass) NOT NULL,
    "VergiAdresi" text
);


ALTER TABLE public."Musteri" OWNER TO postgres;


--
-- Name: Personel; Type: TABLE; Schema: public; Owner: postgres
--

CREATE TABLE public."Personel" (
    "KisiNo" integer DEFAULT currval('public."kisiSayac"'::regclass) NOT NULL,
    "DukkanNo" integer NOT NULL,
    "UcretNo" integer NOT NULL
);


ALTER TABLE public."Personel" OWNER TO postgres;


--
-- Name: Satis; Type: TABLE; Schema: public; Owner: postgres
--

CREATE TABLE public."Satis" (
    "GelirNo" integer DEFAULT currval('public."gelirSayac"'::regclass) NOT NULL,
    "KisiNo" integer NOT NULL,
    "DukkanNo" integer NOT NULL
);


ALTER TABLE public."Satis" OWNER TO postgres;


--
-- Name: Satis-Urun; Type: TABLE; Schema: public; Owner: postgres
--

CREATE TABLE public."Satis-Urun" (
    "No" integer NOT NULL,
    "GelirNo" integer DEFAULT currval('public."gelirSayac"'::regclass) NOT NULL,
    "UrunNo" integer NOT NULL,
    "UrunAdedi" smallint NOT NULL,
    "BirimFiyat" double precision NOT NULL,
    CONSTRAINT "Satis-Urun_Check" CHECK ((("UrunAdedi" >= 0) AND ("BirimFiyat" >= (0)::double precision)))
);


ALTER TABLE public."Satis-Urun" OWNER TO postgres;


--
-- Name: Satis-Urun_No_seq; Type: SEQUENCE; Schema: public; Owner: postgres
```

```
--

CREATE SEQUENCE public."Satis-Urun_No_seq"
    AS integer
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;


ALTER TABLE public."Satis-Urun_No_seq" OWNER TO postgres;


--
-- Name: Satis-Urun_No_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: postgres
--

ALTER SEQUENCE public."Satis-Urun_No_seq" OWNED BY public."Satis-Urun"."No";



--
-- Name: siparisSayac; Type: SEQUENCE; Schema: public; Owner: postgres
--

CREATE SEQUENCE public."siparisSayac"
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;


ALTER TABLE public."siparisSayac" OWNER TO postgres;


--
-- Name: Siparis; Type: TABLE; Schema: public; Owner: postgres
--

CREATE TABLE public."Siparis" (
    "SiparisNo" integer DEFAULT nextval('public."siparisSayac"'::regclass) NOT NULL,
    "DukkanNo" integer NOT NULL
);


ALTER TABLE public."Siparis" OWNER TO postgres;


--
-- Name: Siparis-Urun; Type: TABLE; Schema: public; Owner: postgres
--

CREATE TABLE public."Siparis-Urun" (
    "No" integer NOT NULL,
    "SiparisNo" integer NOT NULL,
    "UrunNo" integer NOT NULL,
    "UrunAdedi" smallint NOT NULL,
    "BirimFiyat" double precision,
    CONSTRAINT "Siparis-Urun_Check" CHECK (("BirimFiyat" >= (0)::double precision))
);


ALTER TABLE public."Siparis-Urun" OWNER TO postgres;


--
-- Name: Siparis-Urun_No_seq; Type: SEQUENCE; Schema: public; Owner: postgres
--

CREATE SEQUENCE public."Siparis-Urun_No_seq"
    AS integer
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;


ALTER TABLE public."Siparis-Urun_No_seq" OWNER TO postgres;
```

```sql
--
-- Name: Siparis-Urun_No_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: postgres
--

ALTER SEQUENCE public."Siparis-Urun_No_seq" OWNED BY public."Siparis-Urun"."No";


--
-- Name: Tamir-Personel; Type: TABLE; Schema: public; Owner: postgres
--

CREATE TABLE public."Tamir-Personel" (
    "No" integer NOT NULL,
    "GelirNo" integer DEFAULT currval('public."gelirSayac"'::regclass) NOT NULL,
    "KisiNo" integer NOT NULL
);


ALTER TABLE public."Tamir-Personel" OWNER TO postgres;

--
-- Name: Tamir-Personel_No_seq; Type: SEQUENCE; Schema: public; Owner: postgres
--

CREATE SEQUENCE public."Tamir-Personel_No_seq"
    AS integer
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;


ALTER TABLE public."Tamir-Personel_No_seq" OWNER TO postgres;

--
-- Name: Tamir-Personel_No_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: postgres
--

ALTER SEQUENCE public."Tamir-Personel_No_seq" OWNED BY public."Tamir-Personel"."No";


--
-- Name: Tedarikci; Type: TABLE; Schema: public; Owner: postgres
--

CREATE TABLE public."Tedarikci" (
    "KisiNo" integer DEFAULT currval('public."kisiSayac"'::regclass) NOT NULL
);


ALTER TABLE public."Tedarikci" OWNER TO postgres;

--
-- Name: Tedarikci-Urun; Type: TABLE; Schema: public; Owner: postgres
--

CREATE TABLE public."Tedarikci-Urun" (
    "No" integer NOT NULL,
    "KisiNo" integer NOT NULL,
    "UrunNo" integer NOT NULL
);


ALTER TABLE public."Tedarikci-Urun" OWNER TO postgres;

--
-- Name: Tedarikci-Urun_No_seq; Type: SEQUENCE; Schema: public; Owner: postgres
--

CREATE SEQUENCE public."Tedarikci-Urun_No_seq"
    AS integer
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
```

```sql
    NO MAXVALUE
    CACHE 1;


ALTER TABLE public."Tedarikci-Urun_No_seq" OWNER TO postgres;


--
-- Name: Tedarikci-Urun_No_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: postgres
--

ALTER SEQUENCE public."Tedarikci-Urun_No_seq" OWNED BY public."Tedarikci-Urun"."No";



--
-- Name: ucretSayac; Type: SEQUENCE; Schema: public; Owner: postgres
--

CREATE SEQUENCE public."ucretSayac"
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;


ALTER TABLE public."ucretSayac" OWNER TO postgres;


--
-- Name: Ucret; Type: TABLE; Schema: public; Owner: postgres
--

CREATE TABLE public."Ucret" (
    "UcretNo" integer DEFAULT nextval('public."ucretSayac"'::regclass) NOT NULL,
    "Maas" double precision NOT NULL,
    "SSK" double precision NOT NULL,
    "Yol" double precision NOT NULL,
    "Yemek" double precision NOT NULL,
    "ToplamUcret" double precision,
    CONSTRAINT "UcretCheck" CHECK ((("Maas" >= (0)::double precision) AND ("SSK" >= (0)::double precision) AND ("Yol"
>= (0)::double precision) AND ("Yemek" >= (0)::double precision)))
);


ALTER TABLE public."Ucret" OWNER TO postgres;


--
-- Name: Urun; Type: TABLE; Schema: public; Owner: postgres
--

CREATE TABLE public."Urun" (
    "UrunNo" integer NOT NULL,
    "Isim" character varying(40) NOT NULL,
    "UrunModel" character varying(40) NOT NULL,
    "Stok" integer NOT NULL,
    "UrunKodu" character(5),
    CONSTRAINT "UrunCheck" CHECK (("Stok" >= 0))
);


ALTER TABLE public."Urun" OWNER TO postgres;


--
-- Name: Urun_UrunNo_seq; Type: SEQUENCE; Schema: public; Owner: postgres
--

CREATE SEQUENCE public."Urun_UrunNo_seq"
    AS integer
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;


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

```
--
-- Name: Urun_UrunNo_seq; Type: SEQUENCE OWNED BY; Schema: public; Owner: postgres
--

ALTER SEQUENCE public."Urun_UrunNo_seq" OWNED BY public."Urun"."UrunNo";


--
-- Name: dukkanSayac; Type: SEQUENCE; Schema: public; Owner: postgres
--

CREATE SEQUENCE public."dukkanSayac"
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;


ALTER TABLE public."dukkanSayac" OWNER TO postgres;


--
-- Name: dukkanSayac; Type: SEQUENCE OWNED BY; Schema: public; Owner: postgres
--

ALTER SEQUENCE public."dukkanSayac" OWNED BY public."Dukkan"."DukkanNo";


--
-- Name: Arac AracNo; Type: DEFAULT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Arac" ALTER COLUMN "AracNo" SET DEFAULT
nextval('public."Arac_AracNo_seq"'::regclass);


--
-- Name: Dukkan DukkanNo; Type: DEFAULT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Dukkan" ALTER COLUMN "DukkanNo" SET DEFAULT
nextval('public."dukkanSayac"'::regclass);


--
-- Name: FaturaTipi FaturaTipiNo; Type: DEFAULT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."FaturaTipi" ALTER COLUMN "FaturaTipiNo" SET DEFAULT
nextval('public."FaturaTipi_FaturaTipiNo_seq"'::regclass);


--
-- Name: Kisi KisiNo; Type: DEFAULT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Kisi" ALTER COLUMN "KisiNo" SET DEFAULT nextval('public."kisiSayac"'::regclass);


--
-- Name: Satis-Urun No; Type: DEFAULT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Satis-Urun" ALTER COLUMN "No" SET DEFAULT nextval('public."Satis-
Urun_No_seq"'::regclass);


--
-- Name: Siparis-Urun No; Type: DEFAULT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Siparis-Urun" ALTER COLUMN "No" SET DEFAULT nextval('public."Siparis-
Urun_No_seq"'::regclass);


--
```

```
--
-- Name: Tamir GelirNo; Type: DEFAULT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Tamir" ALTER COLUMN "GelirNo" SET DEFAULT currval('public."gelirSayac"'::regclass);


--
-- Name: Tamir-Personel No; Type: DEFAULT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Tamir-Personel" ALTER COLUMN "No" SET DEFAULT nextval('public."Tamir-Personel_No_seq"'::regclass);


--
-- Name: Tedarikci-Urun No; Type: DEFAULT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Tedarikci-Urun" ALTER COLUMN "No" SET DEFAULT nextval('public."Tedarikci-Urun_No_seq"'::regclass);


--
-- Name: Urun UrunNo; Type: DEFAULT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Urun" ALTER COLUMN "UrunNo" SET DEFAULT nextval('public."Urun_UrunNo_seq"'::regclass);


--
-- Data for Name: Arac; Type: TABLE DATA; Schema: public; Owner: postgres
--

INSERT INTO public."Arac" VALUES
        (1, '34SA5370', 'Sportage', 600, 61),
        (2, '32AS5757', 'Fiat', 12123, 62),
        (11, '34TU9898', 'Araba', 1500, 62);


--
-- Data for Name: Dukkan; Type: TABLE DATA; Schema: public; Owner: postgres
--

INSERT INTO public."Dukkan" VALUES
        (1, 'Grup Oto', 'Beylikdüzü', '05000000000', NULL);


--
-- Data for Name: Fatura; Type: TABLE DATA; Schema: public; Owner: postgres
--



--
-- Data for Name: FaturaTipi; Type: TABLE DATA; Schema: public; Owner: postgres
--



--
-- Data for Name: Gelir; Type: TABLE DATA; Schema: public; Owner: postgres
--

INSERT INTO public."Gelir" VALUES
        (57, 50, '2022-12-27', 'S'),
        (61, 3600, '2022-12-28', 'S'),
        (62, 16000, '2022-12-28', 'S'),
        (31, 100, '2022-12-27', 'T');


--
-- Data for Name: Gider; Type: TABLE DATA; Schema: public; Owner: postgres
--

INSERT INTO public."Gider" VALUES
```

```sql
            (6, 13500, '2022-12-28', NULL, 46, NULL),
            (7, 9000, '2022-08-27', NULL, 47, NULL);


--
-- Data for Name: Kisi; Type: TABLE DATA; Schema: public; Owner: postgres
--

INSERT INTO public."Kisi" VALUES
            (62, '12345678902', 'Muhammet', 'Karaçete', '00000    ', 'Hatay', 'M'),
            (25, '81248214891', 'Ahmet', 'Kara', '81248214891', 'asd', 'P'),
            (61, '12345678901', 'Eren', 'Kara', '11111    ', 'Sinop', 'M'),
            (68, '12345679999', 'Sedat', 'Sagiltici', '22222    ', 'Hatay', 'P');


--
-- Data for Name: Musteri; Type: TABLE DATA; Schema: public; Owner: postgres
--

INSERT INTO public."Musteri" VALUES
            (61, 'Sinop'),
            (62, 'Hatay');


--
-- Data for Name: Personel; Type: TABLE DATA; Schema: public; Owner: postgres
--

INSERT INTO public."Personel" VALUES
            (25, 1, 46),
            (68, 1, 46);


--
-- Data for Name: Satis; Type: TABLE DATA; Schema: public; Owner: postgres
--

INSERT INTO public."Satis" VALUES
            (57, 61, 1),
            (61, 61, 1),
            (62, 61, 1);


--
-- Data for Name: Satis-Urun; Type: TABLE DATA; Schema: public; Owner: postgres
--

INSERT INTO public."Satis-Urun" VALUES
            (23, 57, 2, 1, 25),
            (24, 57, 4, 1, 25),
            (28, 61, 4, 1, 3600),
            (29, 62, 2, 4, 2000),
            (30, 62, 4, 4, 2000);


--
-- Data for Name: Siparis; Type: TABLE DATA; Schema: public; Owner: postgres
--



--
-- Data for Name: Siparis-Urun; Type: TABLE DATA; Schema: public; Owner: postgres
--



--
-- Data for Name: Tamir; Type: TABLE DATA; Schema: public; Owner: postgres
--

INSERT INTO public."Tamir" VALUES
            (31, '2022-12-27', '2022-12-27', 'Tamir2', 2);


--
```

```sql
--
-- Data for Name: Tamir-Personel; Type: TABLE DATA; Schema: public; Owner: postgres
--

INSERT INTO public."Tamir-Personel" VALUES
        (15, 31, 68);


--
-- Data for Name: Tedarikci; Type: TABLE DATA; Schema: public; Owner: postgres
--



--
-- Data for Name: Tedarikci-Urun; Type: TABLE DATA; Schema: public; Owner: postgres
--



--
-- Data for Name: Ucret; Type: TABLE DATA; Schema: public; Owner: postgres
--

INSERT INTO public."Ucret" VALUES
        (47, 6000, 1000, 500, 1500, 9000),
        (46, 8500, 2000, 1000, 2000, 13500);


--
-- Data for Name: Urun; Type: TABLE DATA; Schema: public; Owner: postgres
--

INSERT INTO public."Urun" VALUES
        (2, 'Siyah', 'Lastik', 17, '#2   '),
        (4, 'Kırmızı', 'Kaporta', 6, '#1   ');


--
-- Name: Arac_AracNo_seq; Type: SEQUENCE SET; Schema: public; Owner: postgres
--

SELECT pg_catalog.setval('public."Arac_AracNo_seq"', 11, true);


--
-- Name: FaturaTipi_FaturaTipiNo_seq; Type: SEQUENCE SET; Schema: public; Owner: postgres
--

SELECT pg_catalog.setval('public."FaturaTipi_FaturaTipiNo_seq"', 1, false);


--
-- Name: Satis-Urun_No_seq; Type: SEQUENCE SET; Schema: public; Owner: postgres
--

SELECT pg_catalog.setval('public."Satis-Urun_No_seq"', 30, true);


--
-- Name: Siparis-Urun_No_seq; Type: SEQUENCE SET; Schema: public; Owner: postgres
--

SELECT pg_catalog.setval('public."Siparis-Urun_No_seq"', 46, true);


--
-- Name: Tamir-Personel_No_seq; Type: SEQUENCE SET; Schema: public; Owner: postgres
--

SELECT pg_catalog.setval('public."Tamir-Personel_No_seq"', 15, true);


--
-- Name: Tedarikci-Urun_No_seq; Type: SEQUENCE SET; Schema: public; Owner: postgres
--
```

```sql
SELECT pg_catalog.setval('public."Tedarikci-Urun_No_seq"', 1, false);


--
-- Name: Urun_UrunNo_seq; Type: SEQUENCE SET; Schema: public; Owner: postgres
--

SELECT pg_catalog.setval('public."Urun_UrunNo_seq"', 4, true);


--
-- Name: dukkanSayac; Type: SEQUENCE SET; Schema: public; Owner: postgres
--

SELECT pg_catalog.setval('public."dukkanSayac"', 1, true);


--
-- Name: faturaSayac; Type: SEQUENCE SET; Schema: public; Owner: postgres
--

SELECT pg_catalog.setval('public."faturaSayac"', 1, false);


--
-- Name: gelirSayac; Type: SEQUENCE SET; Schema: public; Owner: postgres
--

SELECT pg_catalog.setval('public."gelirSayac"', 62, true);


--
-- Name: giderSayac; Type: SEQUENCE SET; Schema: public; Owner: postgres
--

SELECT pg_catalog.setval('public."giderSayac"', 7, true);


--
-- Name: kisiSayac; Type: SEQUENCE SET; Schema: public; Owner: postgres
--

SELECT pg_catalog.setval('public."kisiSayac"', 76, true);


--
-- Name: siparisSayac; Type: SEQUENCE SET; Schema: public; Owner: postgres
--

SELECT pg_catalog.setval('public."siparisSayac"', 1, true);


--
-- Name: ucretSayac; Type: SEQUENCE SET; Schema: public; Owner: postgres
--

SELECT pg_catalog.setval('public."ucretSayac"', 49, true);


--
-- Name: Arac AracPK; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Arac"
    ADD CONSTRAINT "AracPK" PRIMARY KEY ("AracNo");


--
-- Name: Dukkan DukkanPK; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Dukkan"
    ADD CONSTRAINT "DukkanPK" PRIMARY KEY ("DukkanNo");


--
```

```sql
--
-- Name: Fatura FaturaPK; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Fatura"
    ADD CONSTRAINT "FaturaPK" PRIMARY KEY ("FaturaNo");


--
-- Name: FaturaTipi FaturaTipiPK; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."FaturaTipi"
    ADD CONSTRAINT "FaturaTipiPK" PRIMARY KEY ("FaturaTipiNo");


--
-- Name: Gelir GelirPK; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Gelir"
    ADD CONSTRAINT "GelirPK" PRIMARY KEY ("GelirNo");


--
-- Name: Gider GiderPK; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Gider"
    ADD CONSTRAINT "GiderPK" PRIMARY KEY ("GiderNo");


--
-- Name: Kisi KisiPK; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Kisi"
    ADD CONSTRAINT "KisiPK" PRIMARY KEY ("KisiNo");


--
-- Name: Kisi KisiUnique; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Kisi"
    ADD CONSTRAINT "KisiUnique" UNIQUE ("TCNo");


--
-- Name: Musteri MusteriPK; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Musteri"
    ADD CONSTRAINT "MusteriPK" PRIMARY KEY ("KisiNo");


--
-- Name: Personel PersonelPK; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Personel"
    ADD CONSTRAINT "PersonelPK" PRIMARY KEY ("KisiNo");


--
-- Name: Satis-Urun Satis-Urun_PK; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Satis-Urun"
    ADD CONSTRAINT "Satis-Urun_PK" PRIMARY KEY ("No");


--
-- Name: Satis SatisPK; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Satis"
```

```sql
    ADD CONSTRAINT "SatisPK" PRIMARY KEY ("GelirNo");


--
-- Name: Siparis-Urun Siparis-Urun_PK; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Siparis-Urun"
    ADD CONSTRAINT "Siparis-Urun_PK" PRIMARY KEY ("No");


--
-- Name: Siparis SiparisPK; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Siparis"
    ADD CONSTRAINT "SiparisPK" PRIMARY KEY ("SiparisNo");


--
-- Name: Tamir-Personel Tamir-PersonelUnique; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Tamir-Personel"
    ADD CONSTRAINT "Tamir-PersonelUnique" UNIQUE ("GelirNo", "KisiNo");


--
-- Name: Tamir-Personel Tamir-Personel_PK; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Tamir-Personel"
    ADD CONSTRAINT "Tamir-Personel_PK" PRIMARY KEY ("No");


--
-- Name: Tamir TamirPK; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Tamir"
    ADD CONSTRAINT "TamirPK" PRIMARY KEY ("GelirNo");


--
-- Name: Tedarikci-Urun Tedarikci-UrunUnique; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Tedarikci-Urun"
    ADD CONSTRAINT "Tedarikci-UrunUnique" UNIQUE ("KisiNo", "UrunNo");


--
-- Name: Tedarikci-Urun Tedarikci-Urun_PK; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Tedarikci-Urun"
    ADD CONSTRAINT "Tedarikci-Urun_PK" PRIMARY KEY ("No");


--
-- Name: Tedarikci TedarikciPK; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Tedarikci"
    ADD CONSTRAINT "TedarikciPK" PRIMARY KEY ("KisiNo");


--
-- Name: Ucret UcretPK; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Ucret"
    ADD CONSTRAINT "UcretPK" PRIMARY KEY ("UcretNo");


--
```

```sql
-- Name: Urun UrunPK; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Urun"
    ADD CONSTRAINT "UrunPK" PRIMARY KEY ("UrunNo");


--
-- Name: Urun Urunkodunique; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Urun"
    ADD CONSTRAINT "Urunkodunique" UNIQUE ("UrunKodu");


--
-- Name: Gider gider_siparis_unique; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Gider"
    ADD CONSTRAINT gider_siparis_unique UNIQUE ("SiparisNo");


--
-- Name: Satis-Urun satisUnique; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Satis-Urun"
    ADD CONSTRAINT "satisUnique" UNIQUE ("GelirNo", "UrunNo");


--
-- Name: Siparis-Urun siparisUnique; Type: CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Siparis-Urun"
    ADD CONSTRAINT "siparisUnique" UNIQUE ("SiparisNo", "UrunNo");


--
-- Name: Satis-Urun TrigGelirGuncelle; Type: TRIGGER; Schema: public; Owner: postgres
--

CREATE TRIGGER "TrigGelirGuncelle" AFTER INSERT ON public."Satis-Urun" FOR EACH ROW EXECUTE FUNCTION
public."TrigFuncSatis-Urun_GelirGuncelle"();


--
-- Name: Siparis-Urun TrigGiderGuncelle; Type: TRIGGER; Schema: public; Owner: postgres
--

CREATE TRIGGER "TrigGiderGuncelle" AFTER INSERT ON public."Siparis-Urun" FOR EACH ROW EXECUTE
FUNCTION public."TrigFuncSiparis-Urun_GiderGuncelle"();


--
-- Name: Siparis-Urun TrigSiparis-Urun_StokGuncelleme; Type: TRIGGER; Schema: public; Owner: postgres
--

CREATE TRIGGER "TrigSiparis-Urun_StokGuncelleme" AFTER INSERT ON public."Siparis-Urun" FOR EACH ROW
EXECUTE FUNCTION public."TrigFuncSiparisStokEkleme"();


--
-- Name: Satis-Urun TrigStokCikarma; Type: TRIGGER; Schema: public; Owner: postgres
--

CREATE TRIGGER "TrigStokCikarma" AFTER INSERT ON public."Satis-Urun" FOR EACH ROW EXECUTE FUNCTION
public."TrigFuncSatisStokDüsme"();


--
-- Name: Ucret TrigUcretToplamHesapla; Type: TRIGGER; Schema: public; Owner: postgres
--
```

CREATE TRIGGER "TrigUcretToplamHesapla" AFTER INSERT OR UPDATE ON public."Ucret" FOR EACH ROW EXECUTE FUNCTION public."TrigFuncUcretToplamHesapla"();

--
-- Name: Tamir tamirsilinince; Type: TRIGGER; Schema: public; Owner: postgres
--

CREATE TRIGGER tamirsilinince AFTER DELETE ON public."Tamir" FOR EACH ROW EXECUTE FUNCTION public.tamirsilinince();

--
-- Name: Arac Arac_KisiNoFK; Type: FK CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Arac"
    ADD CONSTRAINT "Arac_KisiNoFK" FOREIGN KEY ("KisiNo") REFERENCES public."Musteri"("KisiNo") ON UPDATE CASCADE ON DELETE CASCADE;

--
-- Name: Dukkan Dukkan_KisiNoFK; Type: FK CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Dukkan"
    ADD CONSTRAINT "Dukkan_KisiNoFK" FOREIGN KEY ("Yonetici") REFERENCES public."Personel"("KisiNo") ON UPDATE CASCADE ON DELETE CASCADE;

--
-- Name: Fatura Fatura_DukkanNoFK; Type: FK CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Fatura"
    ADD CONSTRAINT "Fatura_DukkanNoFK" FOREIGN KEY ("DukkanNo") REFERENCES public."Dukkan"("DukkanNo") ON UPDATE CASCADE ON DELETE CASCADE;

--
-- Name: Fatura Fatura_FaturaTipiFK; Type: FK CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Fatura"
    ADD CONSTRAINT "Fatura_FaturaTipiFK" FOREIGN KEY ("FaturaTipi") REFERENCES public."FaturaTipi"("FaturaTipiNo") ON UPDATE CASCADE ON DELETE CASCADE;

--
-- Name: Gider Gider_FaturaNoFK; Type: FK CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Gider"
    ADD CONSTRAINT "Gider_FaturaNoFK" FOREIGN KEY ("FaturaNo") REFERENCES public."Fatura"("FaturaNo");

--
-- Name: Gider Gider_SiparisNoFK; Type: FK CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Gider"
    ADD CONSTRAINT "Gider_SiparisNoFK" FOREIGN KEY ("SiparisNo") REFERENCES public."Siparis"("SiparisNo") ON UPDATE CASCADE ON DELETE CASCADE;

--
-- Name: Gider Gider_UcretNoFK; Type: FK CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Gider"
    ADD CONSTRAINT "Gider_UcretNoFK" FOREIGN KEY ("UcretNo") REFERENCES public."Ucret"("UcretNo");

--
-- Name: Musteri MusteriKisi; Type: FK CONSTRAINT; Schema: public; Owner: postgres
--

```
ALTER TABLE ONLY public."Musteri"
    ADD CONSTRAINT "MusteriKisi" FOREIGN KEY ("KisiNo") REFERENCES public."Kisi"("KisiNo") ON UPDATE
CASCADE ON DELETE CASCADE;


--
-- Name: Personel Personel-Kisi_KisiNoFK; Type: FK CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Personel"
    ADD CONSTRAINT "Personel-Kisi_KisiNoFK" FOREIGN KEY ("KisiNo") REFERENCES public."Kisi"("KisiNo") ON
UPDATE CASCADE ON DELETE CASCADE;


--
-- Name: Personel Personel_DukkanNoFK; Type: FK CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Personel"
    ADD CONSTRAINT "Personel_DukkanNoFK" FOREIGN KEY ("DukkanNo") REFERENCES
public."Dukkan"("DukkanNo") ON UPDATE CASCADE ON DELETE CASCADE;


--
-- Name: Personel Personel_UcretNoFK; Type: FK CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Personel"
    ADD CONSTRAINT "Personel_UcretNoFK" FOREIGN KEY ("UcretNo") REFERENCES public."Ucret"("UcretNo") ON
UPDATE CASCADE ON DELETE CASCADE;


--
-- Name: Satis-Urun Satis-Urun_GelirNoFK; Type: FK CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Satis-Urun"
    ADD CONSTRAINT "Satis-Urun_GelirNoFK" FOREIGN KEY ("GelirNo") REFERENCES public."Satis"("GelirNo") ON
UPDATE CASCADE ON DELETE CASCADE;


--
-- Name: Satis-Urun Satis-Urun_UrunNoFK; Type: FK CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Satis-Urun"
    ADD CONSTRAINT "Satis-Urun_UrunNoFK" FOREIGN KEY ("UrunNo") REFERENCES public."Urun"("UrunNo") ON
UPDATE CASCADE ON DELETE CASCADE;


--
-- Name: Satis SatisFK; Type: FK CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Satis"
    ADD CONSTRAINT "SatisFK" FOREIGN KEY ("GelirNo") REFERENCES public."Gelir"("GelirNo") ON UPDATE
CASCADE ON DELETE CASCADE;


--
-- Name: Satis Satis_DukkanNoFK; Type: FK CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Satis"
    ADD CONSTRAINT "Satis_DukkanNoFK" FOREIGN KEY ("DukkanNo") REFERENCES public."Dukkan"("DukkanNo")
ON UPDATE CASCADE ON DELETE CASCADE;


--
-- Name: Satis Satis_KisiNoFK; Type: FK CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Satis"
    ADD CONSTRAINT "Satis_KisiNoFK" FOREIGN KEY ("KisiNo") REFERENCES public."Musteri"("KisiNo") ON UPDATE
CASCADE ON DELETE CASCADE;
```

```
--
-- Name: Siparis Siparis_DukkanNoFK; Type: FK CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Siparis"
    ADD CONSTRAINT "Siparis_DukkanNoFK" FOREIGN KEY ("DukkanNo") REFERENCES public."Dukkan"("DukkanNo")
ON UPDATE CASCADE ON DELETE CASCADE;


--
-- Name: Tamir-Personel Tamir-Personel_GelirNoFK; Type: FK CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Tamir-Personel"
    ADD CONSTRAINT "Tamir-Personel_GelirNoFK" FOREIGN KEY ("GelirNo") REFERENCES public."Tamir"("GelirNo")
ON UPDATE CASCADE ON DELETE CASCADE;


--
-- Name: Tamir-Personel Tamir-Personel_KisiNoFK; Type: FK CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Tamir-Personel"
    ADD CONSTRAINT "Tamir-Personel_KisiNoFK" FOREIGN KEY ("KisiNo") REFERENCES public."Personel"("KisiNo") ON
UPDATE CASCADE ON DELETE CASCADE;


--
-- Name: Tamir TamirFK; Type: FK CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Tamir"
    ADD CONSTRAINT "TamirFK" FOREIGN KEY ("GelirNo") REFERENCES public."Gelir"("GelirNo") ON UPDATE
CASCADE ON DELETE CASCADE;


--
-- Name: Tamir Tamir_AracNoFK; Type: FK CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Tamir"
    ADD CONSTRAINT "Tamir_AracNoFK" FOREIGN KEY ("AracNo") REFERENCES public."Arac"("AracNo") ON UPDATE
CASCADE ON DELETE CASCADE;


--
-- Name: Tedarikci-Urun Tedarikci-Urun_KisiNoFK; Type: FK CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Tedarikci-Urun"
    ADD CONSTRAINT "Tedarikci-Urun_KisiNoFK" FOREIGN KEY ("KisiNo") REFERENCES public."Tedarikci"("KisiNo")
ON UPDATE CASCADE ON DELETE CASCADE;


--
-- Name: Tedarikci-Urun Tedarikci-Urun_UrunNoFK; Type: FK CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Tedarikci-Urun"
    ADD CONSTRAINT "Tedarikci-Urun_UrunNoFK" FOREIGN KEY ("UrunNo") REFERENCES public."Urun"("UrunNo")
ON UPDATE CASCADE ON DELETE CASCADE;


--
-- Name: Tedarikci TedarikciKisi; Type: FK CONSTRAINT; Schema: public; Owner: postgres
--

ALTER TABLE ONLY public."Tedarikci"
    ADD CONSTRAINT "TedarikciKisi" FOREIGN KEY ("KisiNo") REFERENCES public."Kisi"("KisiNo") ON UPDATE
CASCADE ON DELETE CASCADE;


--
-- Name: SCHEMA public; Type: ACL; Schema: -; Owner: postgres
```

Fonksiyonlar ne yapıyor

- get_data_kisi(kişino integer) fonksiyonu girilen kisino parametresine göre kişinin ad ve soyadını birleştirip bir text halinde kullanıcıya geri döndürüyor.

```
CREATE OR REPLACE FUNCTION public.get_data_kisi(kisino integer)
 RETURNS character varying
 LANGUAGE plpgsql
AS $function$

declare
        isim varchar(50);
        soyisim varchar(50);
        ad varchar(110);

begin
        isim:=(select "Isim" from "Kisi" where "KisiNo"=kisino);
        soyisim:=(select "Soyisim" from "Kisi" where "KisiNo"=kisino);
        ad:=isim||' '||soyisim;
        return ad;
end;
$function$
```

- karzararhesapla(sayi integer) fonksiyonu geriye bir tablo döndürüyor öyleki bu tablo girilen sayi değerine göre günümüzden girilen değer kadar öncesinin 1 aylık gelir ve giderlerini hesaplıyor. Sonra gelirleri giderlerden çıkararak kar-zarar hesaplıyor. Tarihleri ile birlikte bunu ekrana yazıyor.

```
CREATE OR REPLACE FUNCTION public.karzararhesapla(sayi integer)
 RETURNS TABLE("GelirTutar" double precision, "GiderTutar" double precision, "Kar-Zarar" double precision, "Tarih1" date, "Tarih2" date)
 LANGUAGE plpgsql
AS $function$

declare

begin

        if (select sum("Tutar")as toplamgelir from "Gelir" where age("Gelir"."Tarih")BETWEEN INTERVAL '1 month'*(sayi-1) and interval '1month'*sayi) is NULL then

        return query

        select "toplamgelir", "toplamgider",0-"toplamgider" as "karzarar" ,date(current_date-interval '1month'*(sayi)) as "Baslangic",date(current_date-interval '1month'*(sayi-1)) as "Bitis" from (select sum("Tutar")as toplamgelir from "Gelir" where age("Gelir"."Tarih")BETWEEN INTERVAL '1 month'*(sayi-1) -interval'1day' and interval '1month'*sayi) as gelir ,(select sum("Tutar") as toplamgider from "Gider" where age("Gider"."Tarih")BETWEEN INTERVAL '1 month'*(sayi-1)-interval'1day' and interval '1month'*sayi) as gider;

        elseif (select sum("Tutar") as toplamgider from "Gider" where age("Gider"."Tarih")BETWEEN INTERVAL '1 month'*(sayi-1) and interval '1 month'*sayi) is null then

        return query
```

```
select "toplamgelir", "toplamgider",toplamgelir-0 as "karzarar",date(current_date-interval '1month'*(sayi)) as
"Baslangic",date(current_date-interval '1month'*(sayi-1)) as "Bitis" from (select sum("Tutar")as toplamgelir from
"Gelir" where age("Gelir"."Tarih")BETWEEN INTERVAL '1 month'*(sayi-1)-interval'1day' and interval
'1month'*sayi) as gelir ,(select sum("Tutar") as toplamgider from "Gider" where age("Gider"."Tarih")BETWEEN
INTERVAL '1 month'*(sayi-1)-interval'1day' and interval '1month'*sayi) as gider;
else
return query

select "toplamgelir", "toplamgider","toplamgelir"-"toplamgider" as "karzarar",date(current_date-interval
'1month'*(sayi)) as "Baslangic",date(current_date-interval '1month'*(sayi-1)) as "Bitis" from (select sum("Tutar")as
toplamgelir from "Gelir" where age("Gelir"."Tarih")BETWEEN INTERVAL '1 month'*(sayi-1)-interval'1day' and
interval '1month'*sayi) as gelir ,(select sum("Tutar") as toplamgider from "Gider" where
age("Gider"."Tarih")BETWEEN INTERVAL '1 month'*(sayi-1)-interval'1day' and interval '1month'*sayi) as gider;

end if;
end;

$function$
```

- karzarartablo(sayac integer) fonksiyonu günümüzden girilen sayaç kadar öncesinin kar-zarar tablosunu 1 aylık aralıklarla gösteriyor. Bunu yaparken karzararhesapla(sayi integer) fonksiyonunu kullanıyor. İçinde bir döngü ile sayaç kadar karzararhesapla fonksiyonundan dönen değerleri bir tabloya döküyor.

```
Create Type my_type6 as(("GelirTutar" double precision, "GiderTutar" double precision, "Kar-Zarar" double precision,
"Baslangic" date, "Bitis" date)

CREATE OR REPLACE FUNCTION public.karzarartablo(sayac integer)
 RETURNS SETOF my_type6
 LANGUAGE plpgsql
AS $function$
declare
        begin
        for i in 1 .. sayac loop
            return next karzararhesapla(i);
        end loop;
end $function$
```

- stoktavarmi(urunno int, urunadet int) fonksiyonu girilen urunno'dan istenilen urunadet kadar var mı diye kontrol ediyor. Ürünün stok sayısı istenilen değerden büyük ise geriye true bir boolean değer döndürüyor. Değil ise geriye false döndürüyor.

```
CREATE OR REPLACE FUNCTION public.stoktavarmi(urunno integer, urunadet integer)
 RETURNS boolean
 LANGUAGE plpgsql
AS $function$
declare
        varmi boolean;
        stok integer;
begin
        stok:=(select "Stok" from "Urun" where "UrunNo"=urunno);
        if stok>urunadet then
        return true;
        else
        return false;
        end if;
end;
$function$
```

- "TrigFuncSatis-Urun_GelirGuncelle"() fonksiyonu bir trigger fonksiyondur. Satis-Urun tablosuna bir kayıt eklendikten sonra Gelir tablosundaki Tutarı güncelliyor. İlk önce ürünün birim fiyatı ile ürün adedini çarparak tutar hesaplıyor. Ondan sonra bu tutarı Gelirdeki ilişkili Tutara ekliyor.

```
CREATE OR REPLACE FUNCTION public."TrigFuncSatis-Urun_GelirGuncelle"()
 RETURNS trigger
 LANGUAGE plpgsql
AS $function$
Declare
        fiyat double precision;
        tutar int;
Begin
        tutar:=(select "Tutar" from "Gelir"
        where "GelirNo"=new."GelirNo");
        if tutar is null then
        fiyat:= ((new."UrunAdedi") * (new."BirimFiyat"));
        else
        fiyat:= tutar+((new."UrunAdedi") * (new."BirimFiyat"));
        end if;
        Update "Gelir" set "Tutar"=fiyat where "GelirNo"=new."GelirNo" ;
        return new;
End;
$function$
```

- "TrigFuncSatisStokDüsme"() fonksiyonu bir trigger fonksiyondur. Satis-Urun tablosuna bir kayıt eklendikten sonra satılan ürün adedince ürünün stoğunu düşürüyor.

```
CREATE OR REPLACE FUNCTION public."TrigFuncSatisStokDüsme"()
 RETURNS trigger
 LANGUAGE plpgsql
AS $function$
declare
        stok int;
begin
        stok:=(select "Stok" from "Urun" where "UrunNo"=new."UrunNo");
        update "Urun" set "Stok"=stok-new."UrunAdedi" where "UrunNo"=new."UrunNo";
        return new;
end;
$function$
```

- "TrigFuncSiparis-Urun_GiderGuncelle"() fonksiyonu bir trigger fonksiyondur. Siparis-Urun tablosuna bir kayıt eklendikten sonra Gider tablosunu güncelliyor. Bu şu şekilde yapılıyor ürünün birim fiyatı ile ürün adedini çarpıyor çıkan tutarı Gider tablosundaki ilişkili tutara ekliyor.

```
CREATE OR REPLACE FUNCTION public."TrigFuncSiparis-Urun_GiderGuncelle"()
 RETURNS trigger
 LANGUAGE plpgsql
AS $function$
Declare
        fiyat double precision;
        tutar int;
Begin
        tutar:=(select "Tutar" from "Gider"
        where "Gider"."SiparisNo"=new."SiparisNo");
        if tutar is null then
        fiyat:= ((new."UrunAdedi") * (new."BirimFiyat"));
        else
        fiyat:= tutar+((new."UrunAdedi") * (new."BirimFiyat"));
        end if;Update "Gider" set "Tutar"=fiyat where "SiparisNo"=new."SiparisNo" ;
        return new;
End;
$function$
```

- "TrigFuncSiparisStokEkleme"() fonksiyonu bir trigger fonksiyondur. Siparis-Urun tablosuna kayıt girildikten sonra girilen kayıttaki ürün adedince ürün tablosundaki ilişkili ürünün stoğunu arttırıyor.

```
CREATE OR REPLACE FUNCTION public."TrigFuncSiparisStokEkleme"()
```

```
 RETURNS trigger
 LANGUAGE plpgsql
AS $function$
Declare
        stok int;
Begin
        stok:=(select "Stok" from "Urun" where "UrunNo"=new."UrunNo");
        update "Urun" set "Stok"=stok+new."UrunAdedi" where "UrunNo"=new."UrunNo" ;
return New;
End;
$function$
```

- "TrigFuncUcretToplamHesapla"() fonksiyonu bir trigger fonksiyondur. Ucret
  tablosunda tutulan personellerin dükkana olan giderlerinin toplamını hesaplamaya
  yarıyor. Ucret tablosuna bir kayıt girildikten sonra girilen maas, yol, yemek, SSK
  giderlerini topluyor ve ilişkili ToplamUcreti güncelliyor.

```
CREATE OR REPLACE FUNCTION public."TrigFuncUcretToplamHesapla"()
 RETURNS trigger
 LANGUAGE plpgsql
AS $function$
declare
toplam int;
Begin

if new."Maas"<>old."Maas" or new."Yol"<>old."Yol" or new."Yemek"<>old."Yemek" or new."SSK"<>old."SSK" or
new."ToplamUcret" is NULL then
toplam:=(new."Maas"+new."Yol"+new."SSK"+new."Yemek");
update "Ucret" set "ToplamUcret"=toplam where "UcretNo"=new."UcretNo";
end if;
return new;
End;
$function$
```

- tamirsilinince() bir trigger fonksiyondur. Tamir silinince tamir-personel tablosundaki
  ilişkili verileride siliyor.

```
CREATE OR REPLACE FUNCTION public.tamirsilinince()
 RETURNS trigger
 LANGUAGE plpgsql
AS $function$
declare
begin

delete from "Tamir-Personel" where "Tamir-Personel"."GelirNo"=new."GelirNo";
return new;

end;
$function$
```

Arama Ekleme Silme Güncelleme SS'leri

FPersonel (Window 1)

İsim
Soyisim
TC No 12345679999
Telefon
Adres

Hangi Dükkanda Çalışıyor Grup Oto
Bir Maaş Seçiniz 6000

Ekle   Sil
Ara   Güncelle
Uygula   Arama Sonuçlarını

KisiNo	TCNo	Isim	Soyisim	Telefon	Adres
75	88888888888	Kadir	Kara	05300000000	Trabzon
68	12345679999	Sedat	Sagiltici	22222	Hatay
25	81248214891	Ahmet	Kara	81248214891	asd

FPersonel (Window 2)

İsim
Soyisim
TC No 12345679999
Telefon
Adres

Hangi Dükkanda Çalışıyor Grup Oto
Bir Maaş Seçiniz 6000

Ekle   Sil
Ara   Güncelle
Uygula   Arama Sonuçlarını

KisiNo	TCNo	Isim	Soyisim	Telefon	Adres
68	12345679999	Sedat	Sagiltici	22222	Hatay

FPersonel (Window 3)

İsim
Soyisim
TC No
Telefon
Adres

Hangi Dükkanda Çalışıyor Grup Oto
Bir Maaş Seçiniz 6000

Ekle   Sil
Ara   Güncelle
Uygula   Arama Sonuçlarını

KisiNo	TCNo	Isim	Soyisim	Telefon	Adres
75	88888888888	Kadir	Kara	05300000000	Trabzon
68	12345679999	Sedat	Sagiltici	22222	Hatay
25	81248214891	Ahmet	Kara	81248214891	asd

Silme islemi tamamlandi

OK

FPersonel

İsim [   ] Hangi Dükkanda Çalışıyor [Grup Oto ▾]

Soyisim [   ] Bir Maaş Seçiniz [6000 ▾]

TC No [   ]

Telefon [   ] [Ekle] [Sil]

Adres [   ] [Ara] [Güncelle]

[Uygula] [Arama Sonuçlarını]

KisiNo	TCNo	Isim	Soyisim	Telefon	Adres
68	12345679999	Sedat	Sagiltici	22222	Hatay
25	81248214891	Ahmet	Kara	81248214891	asd

FPersonel

İsim [Kadir] Hangi Dükkanda Çalışıyor [Grup Oto ▾]

Soyisim [Kara] Bir Maaş Seçiniz [6000 ▾]

TC No [88888888888]

Telefon [05322222222] [Ekle] [Sil]

Adres [Trabzon] [Ara] [Güncelle]

[Uygula] [Arama Sonuçlarını]

KisiNo	TCNo	Isim	Soyisim	Telefon	Adres
76	88888888888	Kadır	Kara	05300000000	Trabzon
68	12345679999	Sedat	Sagiltici	22222	Hatay
25	81248214891	Ahmet	Kara	81248214891	asd

Güncelleme işlemi gerçekleşti

[OK]

Uygulamanın Kaynak Kodları

https://github.com/EreenKara/VeriTabaniProjesi