Öğrenci numarası: S141210374

Öğrenci adı: ÖZGÜR BARIŞ

Öğrenci soyadı: AYHAN

Öğrenci e posta: barisayhan01@gmail.com

Uygulamanın kısa tanıtımı

Bu uygulamada ürünler kısmı ana sayfayı oluşturmaktadır. Stokta bulunan ürünlerin özellikleri güncellenebilir,yeni ürünler eklenebilir veya silinebilir. Her ürün bir kategoriye dahil edilir. Kategoriler bölümünde yeni kategoriler eklenebilir,silinebilir ve güncellenebilir. Bu işlemler müşteriler kısmında da yapılabilir. Ürün ve müşterinin birleştiği satış bölümü bulunmaktadır.

İş kuralları

Bir ürün sadece bir kategoride bulunabilir.

Bir kategoride birden fazla ürün bulunabilir.

Personelin adı ve soyadı boş bırakılamaz.

Satış yapılan ürün, ürünler tablosunda bulunmak zorundadır.

Satış yapılan ürün;rezervde yeteri kadar bulunmalıdır.

Her satıştan sorumlu bir personel bulunmalıdır.

Müşterinin bakiyesi ürünün fiyatından yüksek olmalıdır.

Her müşterinin yaşadığı bir şehir vardır.

İlişkisel Şema(Metinsel Gösterim)

Categories(categoryid:integer,categoryname:text)

City(id: integer,name: varchar)

Customer(id: bigint, name: text, city_id: integer, bakiye: integer)

Personel(personelid:integer ,personelad: varchar,personelsoyad: varchar)

Product(productid:integer, productname:text, reserve:integer, buyingprice:integer , saleprice:integer , category:integer)

Sale(saleid:integer, productid:integer , quantity: integer, customerid: integer, saledate:timestamp, personel: integer)

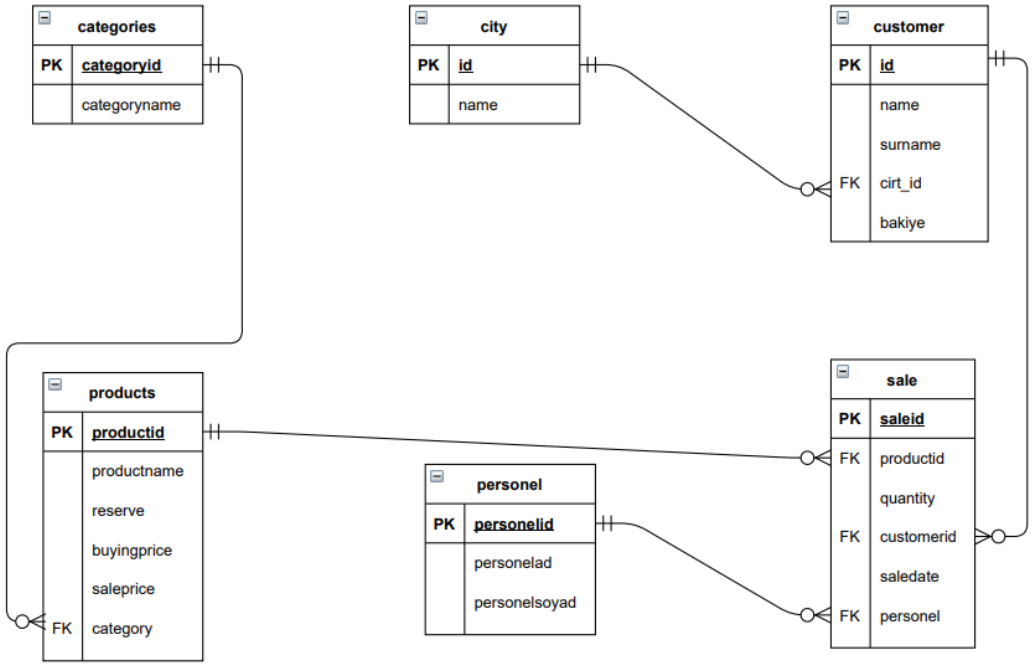
Toplamkategori (sayi: integer

Toplammusteri (sayi: integer)

Toplamsehir(sayi: integer)

Totalutun(int: integer)

Varlık bağıntı diyagramı



Veritabanı kodları

TABLOLAR

```
create table categories
(
   categoryid   integer not null
      constraint categories_pkey
         primary key,
   categoryname text
);

alter table categories
   owner to postgres;

create trigger testtrigger
   after insert
   on categories
   for each row
execute procedure test();

create table products
(
   productid   integer not null
      constraint products_pkey
         primary key,
   productname text,
   reserve     integer,
   buyingprice integer,
   saleprice   integer,
   category    integer
      constraint products_categories_categoryid_fk
         references categories
);
```

```
alter table products
    owner to postgres;

create index fki_products_foreign
    on products (category);

create trigger testtrig2
    after insert
    on products
    for each row
execute procedure triggertoplamurun();

create trigger productname_changes
    before update
    on products
    for each row
execute procedure log_productname_changes();

create table personel
(
    personelid    integer not null
        constraint personel_pkey
            primary key,
    personelad    varchar not null,
    personelsoyad varchar not null
);

alter table personel
    owner to postgres;

create table city
(
    id   serial
        constraint city_pk
```

```
        primary key,
    name varchar(50)
);

alter table city
    owner to postgres;

create table customer
(
    id      bigserial
        constraint customer_pk
            primary key,
    name    text,
    city_id integer
        constraint customer_city_id_fk
            references city,
    surname text
);

alter table customer
    owner to postgres;

create trigger sehirtrigger
    after insert
    on city
    for each row
execute procedure sehirtrigger();

create trigger totalutuntrigger
    after insert
    on city
    for each row
execute procedure totalutuntrigger();

create table sale
```

```
(
    sale_id       integer not null
        constraint sale_pk
            primary key,
    product_id    integer
        constraint sale_product_fk
            references products,
    quantity      integer,
    customer_id   integer
        constraint sale_customer_fk
            references customer,
    sale_date     timestamp,
    sale_personel integer
        constraint sale_personel_fk
            references personel
);

alter table sale
    owner to postgres;

create table toplamkategori
(
    sayi integer
);

alter table toplamkategori
    owner to postgres;

create table toplamsehir
(
    sayi integer
);

alter table toplamsehir
    owner to postgres;
```

```
create table totalutun
(
    int integer
);

alter table totalutun
    owner to postgres;

create table toplammusteri
(
    sayi integer
);

alter table toplammusteri
    owner to postgres;
```

TRIGGER VE PROCEDUR'LER

```
create function triggertoplamurun() returns trigger
    language plpgsql
as
$$
begin
    update toplamurun set ürünsayisi=ürünsayisi + 1;
    return new;
end;
$$;

alter function triggertoplamurun() owner to postgres;

create procedure programsahibi()
    language plpgsql
as
$$
begin
    Raise Notice 'özgür barış ayhan';
end;
$$;

alter procedure programsahibi() owner to postgres;

create function totalbp() returns integer
    language plpgsql
as
$$
declare
    totalbp integer;
begin
    select count(*) into totalbp from buyingprice where products;
    return totalbp;
end;
```

```
$$;

alter function totalbp() owner to postgres;

create function totalreserve() returns integer
    language plpgsql
as
$$
declare
    totalreserve integer;
begin
    select count(*) into totalreserve from products where reserve;
    return reserve;
end;
$$;

alter function totalreserve() owner to postgres;

create function discount(saleprice integer) returns integer
    language plpgsql
as
$$
begin
    saleprice := saleprice - saleprice * 0.10;
    return saleprice;
end;
$$;

alter function discount(integer) owner to postgres;

create function productsec(prdc character varying)
    returns TABLE
        (
            idproduct   integer,
            nameproduct text
```

```
        )
    language plpgsql
as
$$
Begin
    Return Query
        Select productid,
            productname
        From products
        where productname like prdc;
End;
$$;

alter function productsec(varchar) owner to postgres;

create function log_productname_changes() returns trigger
    language plpgsql
as
$$
BEGIN
    IF NEW.productname <> OLD.productname THEN
        INSERT INTO products(productid, productname)
        VALUES (OLD.id, OLD.productname);
    END IF;

    RETURN NEW;
END;
$$;

alter function log_productname_changes() owner to postgres;

create function getadanalisfunc(city_id_prm integer)
    returns TABLE
        (
            id   bigint,
```

```
            name character varying
        )
    language plpgsql
as
$$
begin
    RETURN QUERY select customer.id, customer.name
            from customer
            where customer.city_id = city_id_prm; --adanalıları getir;
end
$$;

alter function getadanalisfunc(integer) owner to postgres;

create function totalsp() returns integer
    language plpgsql
as
$$
declare
    totalsp integer;
begin
    select count(*) into totalsp from products where saleprice;
    return totalsp;
end;
$$;

alter function totalsp() owner to postgres;

create procedure sehirekle(p1 integer, p2 text)
    language sql
as
$$
insert into city(id, name)
values (p1, p2);
$$;
```

```
alter procedure sehirekle(integer, text) owner to postgres;


create procedure personelekle(p1 integer, p2 text, p3 text)
    language sql
as
$$
insert into personel (personelid, personelad, personelsoyad)
values (p1, p2, p3);
$$;


alter procedure personelekle(integer, text, text) owner to postgres;


create procedure kategoriekle(p1 integer, p2 text)
    language sql
as
$$
insert into categories (categoryid, categoryname)
values (p1, p2);
$$;


alter procedure kategoriekle(integer, text) owner to postgres;


create function test() returns trigger
    language plpgsql
as
$$
begin
    update toplamkategori set sayi=sayi + 1;
    return new;
end;
$$;


alter function test() owner to postgres;
```

```
create function sehirtrigger() returns trigger
    language plpgsql
as
$$
begin
    update toplamsehir set sayi=sayi + 1;
    return new;
end;
$$;

alter function sehirtrigger() owner to postgres;

create function totalutuntrigger() returns trigger
    language plpgsql
as
$$
begin
    update totalutun set sayi=sayi + 1;
    return new;
end;
$$;
alter function totalutuntrigger() owner to postgres;
```

Not:Tüm veritabanı işlemleri github sayfasında da bulunmktadır.

UYGULAMA KODLARI

https://github.com/ayhanozgurbaris/PostgreProjectSakaryaVTYS

VİDEO LİNKİ

https://github.com/ayhanozgurbaris/PostgreProjectSakaryaVTYS/blob/master/ayhanob_vtys.mp4