



OYUN PLATFORMU

Web Programlama



Hasan Nail Kalkan

28 AĞUSTOS 2022

SAKARYA ÜNİVERSİTESİ
Sakarya

Uygulama:

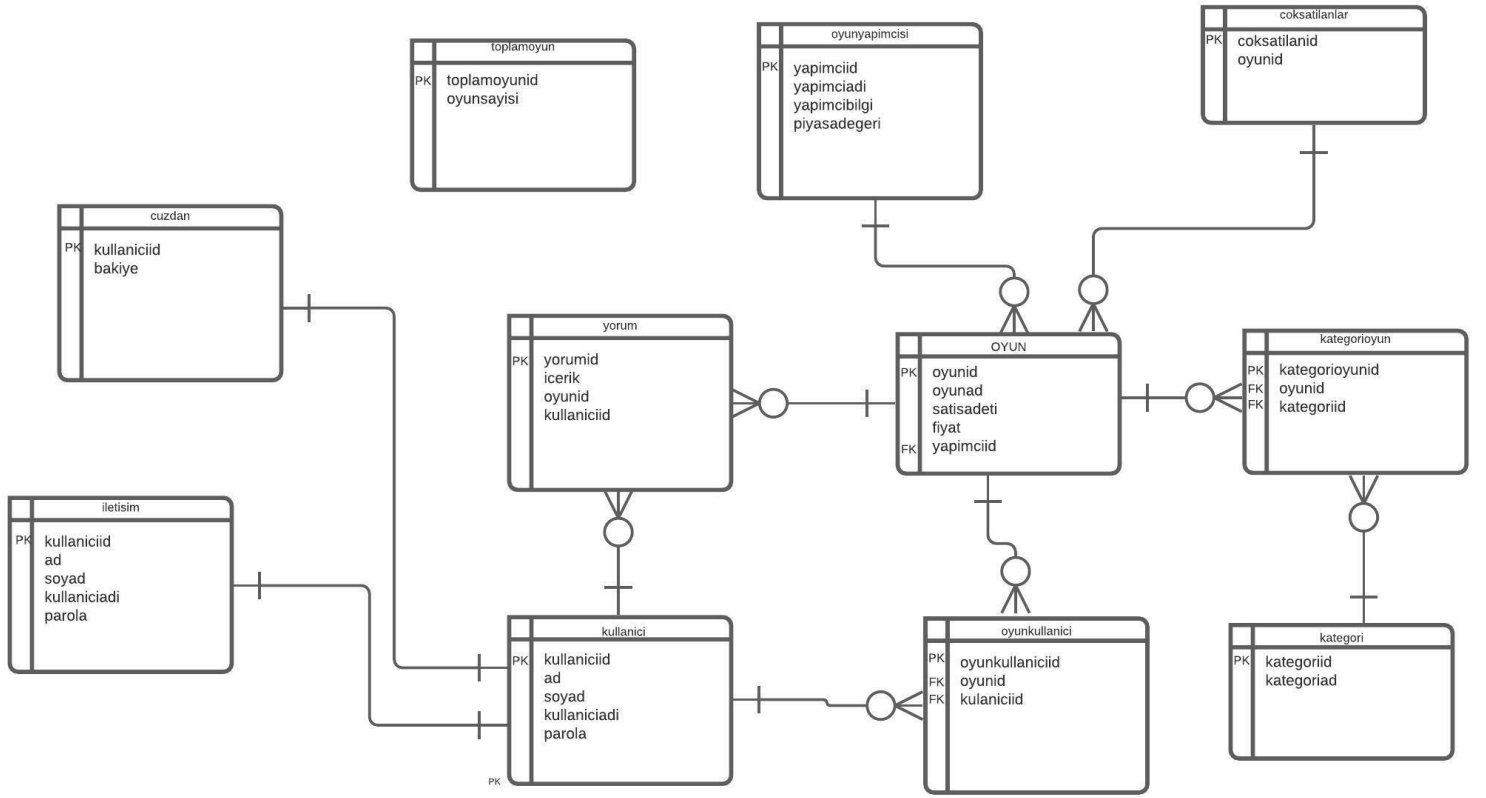
Geliřtirdiđimiz proje bir oyun satın alma programıdır. Uygulamada kullanıcılar kaydolduktan sonra farklı oyunlar satın alabilmektedir. Bakiyelerine yükleme yapıktan sonra oyun satın alma işlemi yapabilirsiniz. Ayrıca kullanıcılar oyunlara yorum ekleyebilmekteler. Admin panelinden mağaza oyun ekleyebiliyor, silebiliyor, güncelleme yapabiliyoruz. Aynı şekilde kullanıcı bilgileri de güncellenebiliyor. Çok satılanlar listesindeki oyunları kullanıcılar inceleyebilir ve oyunları yorumlayabilir. Mağaza bölümünde oyunlar listeleniyor. Seçilen oyunu alabiliyor ya da oyun hakkında detaylı bilgi alabiliyoruz. Örneđin kategori bilgisi, yapılan yorumlar ve ya yapımcı bilgiler.

İlişkisel Şema:

- kullanıcı:(kullaniciid:int,ad:string,soyad:string,kullaniciadi:string,parola:string)
- oyun:(oyunid:int,oyunad:string,satisadeti:int,fiyat:int,yapimciid:int)
- kullanicioyun:(kullanicioyunid:int,oyunid:int,kullaniciid:int)
- kategori:(kategoriid:int,kategoriad:string)
- kategorioyun:(kategorioyunid:int,kategoriid:int,oyunid:int)
- yorum:(yorumid:int,içerik:string,oyunid:int,kullaniciid:int)
- toplamoyun:(toplamoyunid:int,oyunsayisi:int)
- oyunyapimcisi:(yapimciid:int,yapimciadi:string,yapimcibilgi:string,piyasadegeri:double)
- iletişim:(iletisimid:int,email:string,adres:string,kullaniciid:int)
- cüzdan:(kullaniciadi:int,bakiye:int)
- coksatilanlar:(coksatilanlarid:int,oyunid:int)
- beğeni:(begeniid:int,kullaniciid:int,oyunid:int)
- envanter:(envanterid:int,oyunid:int)
- rozet:(rozetid:int,rozetad:string,kullaniciid:int)
- topluluk:(toplulukid,toplulukad:string)

İş Kuralları:

- Bir kullanıcının hiç ya da birden fazla oyunu olabilir, bir oyun hiç ya da birden fazla kullanıcıda bulunabilir
- Bir kullanıcı hiç yada birden fazla yorum yapabilir, bir yorum yalnızca bir kullanıcıya ait olabilir.
- Bir oyun hiç yada birden fazla yorum alabilir, bir yorum yalnızca bir oyunda var olabilir.
- Bir oyun en az bir ya da birden fazla kategoride bulunabilir, bir kategori birden fazla oyun bulundurabilir.
- Bir yapımcı birden fazla oyunda bulunabilir, bir oyunun yalnızca bir yapımcısı olabilir.
- Bir beğeni yalnızca bir kullanıcıya ait olabilir, bir kullanıcı birden fazla beğeni yapabilir.
- Bir rozet hiç yada birden fazla kullanıcıda olabilir, bir kullanıcı yalnızca bir rozete sahip olabilir.
- Bir topluluk hiç yada birden fazla kullanıcıya sahip olabilir, bir kullanıcı hiç ya da birden fazla toplulukta var olabilir
- Bir oyun çok satılanlar da bir kere var olabilir, bu tablo hiç ya da birden fazla tablo bulundurabilir.
- Bir cüzdan yalnızca bir kullanıcıya ait olabilir, bir kullanıcı yalnızca bir cüzdana sahip olabilir.
- Bir iletişim yalnızca bir kullanıcıya ait olabilir, bir kullanıcı yalnızca bir iletişime sahip olabilir.
- Bir oyun yalnızca bir envantere sahip olabilir, bir envanter yalnızca bir oyuna ait olabilir



Postgresql İfadeleri

```
CREATE DATABASE DbOyunMagazasi;

CREATE TABLE kullnaici (
kullaniciid integer PRIMARY KEY,
ad character varying,
soyad character varying,
kullaniciadi character varying,
parola character varying
);
CREATE TABLE oyun (
oyunid integer PRIMARY KEY,
oyunad character varying,
fiyat integer,
satisadeti integer,
yapimciid integer
);

CREATE TABLE oyunyapimcisi (
yapimciid integer PRIMARY KEY,
yapimciadi character varying,
piyasadegeri double precision,
bilgi integer
);

insert into kullanici
(ad,soyad,kullaniciadi,parola) values
('Ahmet','Kaya','ahmetkaya','1234');
insert into oyunyapimcisi
(yapimciadi,piyasadegeri,bilgi) values
('Ubisoft','5.8','Ubisoft hakkında bilgi');
insert into oyun
(oyunad,fiyat,satisadeti,yapimciid) values ('Assasins
Creed',170,0,0);
```

Saklı yordam ve Tetikleyiciler:

```
-- coksatilan trigger fonksiyonu
create or replace function coksatilan_ekle()
returns trigger
as $$
declare
satisadet integer;
oyun_id integer;
begin
oyun_id:=(SELECT oyunid FROM kullanicioyun ORDER BY
kul_oyun_id DESC LIMIT 1);
satisadet:=(select satisadeti from oyun
where oyunid=oyun_id); if satisadet >= 3 then
INSERT INTO coksatilanlar (oyunid) SELECT oyun_id
WHERE NOT EXISTS ( ( SELECT oyunid FROM
coksatilanlar WHERE oyunid = oyun_id);
end if;
return new;
end;
$$
language plpgsql
```

```
coksatilan tirgger
create or replace trigger
coksatilanEkle_trigger after insert
on kullanicioyun
for each row
execute procedure coksatilan_ekle()-----
```

```
-- oyunarttir trigger fonksiyonu
create or replace function oyun_arttir()
returns trigger
as
$$
begin
update toplamoyun set
oyunsayisi=oyunsayisi+1; return new;
end;
$$
language plpgsql
```

```
-- oyunarttir trigger
create trigger oyun_arttir_trigger
after insert
on oyun
for each row
execute procedure oyunarttir()
```

```

-----
        oyunazalt trigger fonksiyonu create or replace
function oyunazalt()
    returns trigger as
    $$
    begin
        update toplamoyun set oynusayisi=oynusayisi-1;
return new;
    end;
    $$
    language plpgsql

--oyun azalt trigger
        create trigger oyun_azalt_trigger after
delete
        on oyun
        for each row
        execute procedure oyunazalt()

-----

--oyun satisadeti arttir tigger fonksiyonu
        create or replace function satisadet_arttir()
returns trigger
        as $$ begin
        update oyun set satisadeti=satisadeti+1 where oyunid=(SELECT oyunid FROM
kullanicioyun ORDER BY kul_oyun_id DESC LIMIT 1);
        return new; end;
        $$
        language plpgsql;

-- oyun satisadeti arttir tigger
        create trigger satisadet_arttir_trigger after
insert
        on kullanicioyun
        for each row
        execute procedure satisadet_arttir()

-----STORED PROCEDURE-----
        Create or replace procedure oyuniade(p1 text)
        language plpgsql
        as $$
        begin
        delete from kullanicioyun where oyunid=(select oyunid from oyun where
oyunad=p1);
        end;
        $$
        Create or replace procedure yorumsil(p1 text)
        as $$
        begin
        delete from yorum where oyunid=(select oyunid from oyun where oyunad=p1); end;

```



```

    $$
    language plpgsql;

    Create or replace procedure kategorisil(p1 text)
    as $$
    begin
    delete from kategorioyun where oyunid=(select oyunid from oyun where
oyunad=p1);
    end;
    $$
    language plpgsql;

    Create or replace procedure oyunsil(p1 text)
    as $$
    begin
    delete from oyun where oyunad=p1;
    end;
    $$
    language plpgsql;

    Create or replace procedure yorumekle(p1 text,p2 text,p3 text) language
plpgsql
    as $$
    begin
    insert into yorum (icerik,oyunid,kullaniciid) values (p1,(select oyunid from
oyun where oyunad=p2),
    (select kullaniciid from kullanıcı where kullaniciadi=p3));
    end;
    $$
    Create or replace procedure coksatilan_sil(p1 text)
    as $$
    begin
    delete from coksatilanlar where oyunid=(select oyunid from oyun where
oyunad=p1);
    end;
    $$
    language plpgsql;

```

VERİ EKLEME

```
private void btn_ekle_Click(object sender, EventArgs e)
{
    try
    {
        baglanti.Open();
        NpgsqlCommand komut = new NpgsqlCommand("call yapimciekle(:p1,:p2,:p3)", baglanti); // yordam(procedure)
        komut.CommandType = CommandType.Text;
        komut.Parameters.AddWithValue("p1", DbType.String).Value = tb_yapimciadi.Text;
        komut.Parameters.AddWithValue("p2", DbType.String).Value = rtb_bilgi.Text;
        komut.Parameters.AddWithValue("p3", DbType.String).Value = Double.Parse(tb_piyasadegeri.Text);
        komut.ExecuteNonQuery();
        baglanti.Close();
        MessageBox.Show("Kayıt eklendi");
    }
    catch (Exception)
    {
        MessageBox.Show("Kayıt eklenemedi");
    }
}
```

```
private void btn_ekle_Click(object sender, EventArgs e)
{
    try
    {
        baglanti.Open();
        NpgsqlCommand komut = new NpgsqlCommand("call yapimciekle(:p1,:p2,:p3)", baglanti); // yordam(procedure)
        komut.CommandType = CommandType.Text;
        komut.Parameters.AddWithValue("p1", DbType.String).Value = tb_yapimciadi.Text;
        komut.Parameters.AddWithValue("p2", DbType.String).Value = rtb_bilgi.Text;
        komut.Parameters.AddWithValue("p3", DbType.String).Value = Double.Parse(tb_piyasadegeri.Text);
        komut.ExecuteNonQuery();
        baglanti.Close();
        MessageBox.Show("Kayıt eklendi");
    }
    catch (Exception)
    {
        MessageBox.Show("Kayıt eklenemedi");
    }
}
```

VERİ GÜNCELLEME

```
private void btn_profil_guncelle_Click(object sender, EventArgs e)
{
    string ad = tb_profil_ad.Text;
    string soyad = tb_profil_soyad.Text;
    string kull_adi = tb_profil_kulAdi.Text;
    string parola = tb_profil_parola.Text;

    try
    {
        string sorgu = "update kullanıcı set ad='"+ad+"',soyad='"+soyad+"',kullaniciadi='"+kull_adi+"',parola='"+parola+"' where " +
            "kullaniciid=(select kullaniciid from kullanıcı where kullaniciadi='"+ frm_giris_kayit.kul_Ad + "')";
        baglanti.Open();
        NpgsqlCommand komut = new NpgsqlCommand(sorgu, baglanti);
        komut.ExecuteNonQuery();
        baglanti.Close();
        ProfilGetir();
        MessageBox.Show("Kayıt güncellendi");
    }
    catch
    {
        MessageBox.Show("Kayıt güncellenemedi");
    }
}
```

```
private void btn_guncelle_Click(object sender, EventArgs e)
{
    if (lbl_secilen.Text != "")
    {
        lbl_guncelle_uyari.Text = "Kutucuğa yeni veriyi girin";

        string sorgu = "";
        if (tb_guncelle.Text != lbl_secilen.Text)
        {
            baglanti.Open();
            if (lbl_secilen.Text != "")
                sorgu = "update oyun set oyunad='" + tb_guncelle.Text + "' " +
                    "where oyunid=(select oyunid from oyun where oyunad='" + lbl_secilen.Text + "')";
            else
                MessageBox.Show("Bir oyun seciniz");
            NpgsqlCommand komut = new NpgsqlCommand(sorgu, baglanti);
            komut.ExecuteNonQuery();
            baglanti.Close();
            OyunListele();
            MessageBox.Show("Kayıt Güncellendi");
        }
        else
            MessageBox.Show("Veride değişiklik yapmadınız");
    }
    else
        MessageBox.Show("Bir oyun seçiniz");
}
```

VERİ SİLME

```
ivate void btn_hesapsil_Click(object sender, EventArgs e)

    baglanti.Open();

    // yorum sil
    NpgsqlCommand komut = new NpgsqlCommand("delete from yorum where kullaniciid=(select kullaniciid from kullanıcı where kullaniciadi='"+frm_giris_kayit.
    komut.ExecuteNonQuery();
    baglanti.Close();

    // alınan oyun sil
    baglanti.Open();
    komut = new NpgsqlCommand("delete from kullanicioyun where kullaniciid=(select kullaniciid from kullanıcı where kullaniciadi='" + frm_giris_kayit.kul_
    komut.ExecuteNonQuery();
    baglanti.Close();

    // cüzdan sil
    baglanti.Open();
    komut = new NpgsqlCommand("delete from cüzdan where kullaniciid=(select kullaniciid from kullanıcı where kullaniciadi='" + frm_giris_kayit.kul_Ad + ""
    komut.ExecuteNonQuery();
    baglanti.Close();

    // iletişim sil
    baglanti.Open();
    komut = new NpgsqlCommand("delete from iletisim where kullaniciid=(select kullaniciid from kullanıcı where kullaniciadi='" + frm_giris_kayit.kul_Ad + "
    komut.ExecuteNonQuery();
    baglanti.Close();
```

```
// oyuna ait kategorileri sil
baglanti.Open();
komut = new NpgsqlCommand("call kategorisil(:p1)", baglanti); // yordam(procedure)
komut.CommandType = CommandType.Text;
komut.Parameters.AddWithValue("p1", DbType.String).Value = lbl_secilen.Text;
komut.ExecuteNonQuery();
baglanti.Close();

// oyunu satın alan kişilerden oyunu sil
baglanti.Open();
komut = new NpgsqlCommand("call oyuniade(:p1)", baglanti); // yordam(procedure)
komut.CommandType = CommandType.Text;
komut.Parameters.AddWithValue("p1", DbType.String).Value = lbl_secilen.Text;
komut.ExecuteNonQuery();
baglanti.Close();

// son olarak oyunu sil
baglanti.Open();
komut = new NpgsqlCommand("call oynunsil(:p1)", baglanti); // yordam(procedure)
komut.CommandType = CommandType.Text;
komut.Parameters.AddWithValue("p1", DbType.String).Value = lbl_secilen.Text;
komut.ExecuteNonQuery();
baglanti.Close();

OyunListele();
MessageBox.Show("Kayıt silindi");
tb_guncelle.Text = "";
baglanti.Close();
```