

VERİ TABANI YÖNETİM SİSTEMLERİ

PROJE ODEVİ

PROF.DR. CELAL ÇEKEN

Öğrenci Adı: QUSAI ALKALEH

Öğrenci No: B211210566

Öğrenci Eposta adresi: gusai.alkaleh@ogr.sakarya.edu.tr

Öğrenci Adı: SUHAYB ALHAS OTHMAN

Öğrenci No: B201210560

Öğrenci Eposta adresi: suhayb.othman@ogr.sakarya.edu.tr

PROBLEM TANITIMI

Herhangi açılan yeni bir restoran içerdiği kaynakları yönetmekte zorlanmaktadır.

Bizim uygulmanın amacı bir restoran için ideal bir yönetim sistemi geliştirmektir.

SENARYO

Bizim uygulama bir restoran yönetim sistemidir.

Bu yönetim sisteminde:

- Ana menüyü
- Ürünleri
- siparişleri
- Personelleri
- Fatura işlemlerini yönetmek amaçlıyoruz.

Ana menü 3 bölümden oluşur:

1. Ana tabak
2. İçecek
3. Mezeler

Ürünler sayfasında yemeklerin eklendiği, çıkarıldığı güncellendiği bir sayfadır.

Siparişler sayfasında alınan siparişler kontrol edilmektedir.

Personeller sayfasında da çalışan personellerin yönetimi yapılmaktadır.

Fatura kısmı da bütün siparişler için faturalar kesilecektir.

İŞ KURALLARI

- Restoranda personeller çalışmaktadır.
- Personellerin kodu, adı, soyadı bilgileri bulunur.
- Personeller, garson, şef, yönetici, kaşıyer olamk üzere 4 bölüme ayrılır.
- Bu dördü haricinde bir personel bulunmamaktadır.
- Personellerin iletişim bilgileri bulunmaktadır.
- İletişim bilgileri kodu, telefonNo, email, bilgilerini tutar.

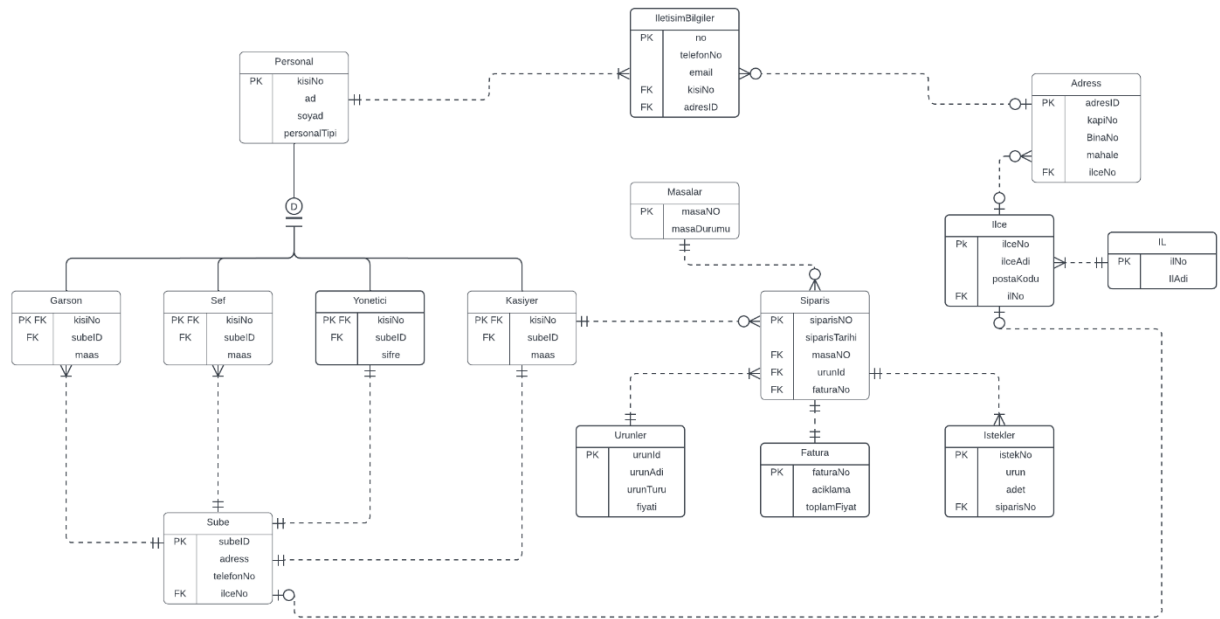
- Personellerin adres bilgileri tutulur.
- Adresin kodu, kapı no, bina no, mahalle, ilçe, il bilgileri bulunmaktadır.
- İlçenin ilçe no, ilçe adı, posta kodu bilgileri saklanır.
- İllerin, il kodu, il adı bilgileri saklanır.
- Restorantta masalar bulunur, masaların no, ve masa durumu bilgileri saklanır.
- Masalara siparişler yapılmaktadır.
- Siparişlerin siparişno ve tarih bilgileri tutulur.
- Restoran şubelerden oluşur.
- Her şubenin, adresi, telefon bilgileri bulunmaktadır.
- Siparişlerde kullanılan ürünler bulunur, ürünlerin kodu, adı, türü ve fiyat bilgileri tutulur.
- Siparişlere kesilen faturalar bulunmaktadır, faturaların numarası, açıklaması, ve toplam fiyat bilgilerini içerir.
- Siparişlerde yapılan istekler bulunur, isteklerin numarası, yemek çeşidi ve adet bilgileri bulunmaktadır.
- Bir personelin yalnızca bir iletişim bilgisi bulunabilir.
- Bir iletişim bilgisi yalnızca bir personele ait olur.
- Bir personele ait yalnızca bir adres bulunmaktadır.
- Bir adres hiç bir personele ait olmayabileceği gibi yalnızca bir personele ait olabilir.
- Bir adres yalnızca bir ilçede bulunur ve ilçe birden fazla adres bulundurabilir.
- Bir ilçe yalnızca bir ilde bulunabilir ve bir il birçok ilçeyi bulundurabilir.
- Bir şubede birden fazla garson çalışabilir yalnız bir garson yalnızca bir şubede çalışır.
- Bir şef yalnızca bir şubede çalışabilir ancak bir şubede birçok şef görev alabilir.
- Bir şubeyi yalnızca bir yönetici yönetebilir.
- Bir yönetici yalnızca bir şubede yer alabilir.
- Bir şubede yalnızca bir kaşiyer yer alabilir ve bir kaşiyer yalnızca bir şubede yer alabilir.
- Siparişler masalara verilir.

- Bir sipariş yalnızca bir masaya ait olabilir ama bir masadan birden fazla sipariş verilebilir.
- Bir sipariş ait yalnızca bir fatura bulunabilir.
- Bir fatura yalnızca bir sipariş için kesilebilir.

İlişkisel şema:

- İletisimBilgiler (no:integer, telefonNo:varchar, email:varchar, kisiNo:varchar, adresID:varchar)
- Adress (adresID:serial , kapiNo:varchar, BinaNo:varchar, mahale:varchar, ilceNo:varchar)
- İlce (ilceNo: serial , ilceAdi:varchar, postaKodu:varchar, ilNo:integer)
- İl (ilNo: serial, İlAdi: varchar)
- İstekler (istekNo: serial, urun:varchar, adet:varchar, siparisNo:integer)
- Fatura (faturaNo: serial , aciklama:varchar, toplamFiyat:varchar)
- Urunler (urunId : serial, urunAdi:varchar, urunTuru:varchar, fiyati:varchar)
- Siparis (siparisNO : serial, siparisTarihi: varchar , masaNO:integer, urunId:integer, faturaNo:integer)
- Masalar (masaNO : serial, masaDurumu: varchar)
- Personal (kisiNo : serial, ad:varchar, soyad:varchar, personalTipi:varchar)
- Kasiyer (kisiNo: integer, subeID: integer , maas: varchar)
- Yonetici (kisiNo:integer , subeID :integer, sifre: varchar)
- Sef(kisiNo:integer , subeID: integer , maas:varchar)
- Garson(kisiNo:integer , subeID: integer, maas:varchar)
- Sube (subeID: serial , address: varchar , telefonNo: varchar , ilceNo: integer)

Varlık Bağlantı modeli:



// Database created RestoranProjesi

CREATE DATABASE RestoranProjesi;

CREATE SCHEMA Parsonal;

FUNCTIONS

--insert function

**create or replace function st_insert(_urunadi varchar(30),
_urunturu varchar(30), _fiyati varchar(6))**

returns int as

\$\$

begin

insert into Urunler(urunAdi, urunTuru, fiyati)

values(_urunadi, _urunturu, _fiyati);

if found then

return 1;

else

return 0;

end if;

end\$\$

language plpgsql

--update function

**create or replace function st_update(_urunId INT, _urunadi
varchar(30), _urunturu varchar(30), _fiyati varchar(6))**

returns int as

\$\$

begin

update Urunler

set

urunAdi = _urunadi,

urunTuru = _urunturu,

fiyati = _fiyati

where urunId = _urunId;

if found then

return 1;

else

return 0;

```
        end if;  
end  
$$  
language plpgsql
```

--select function

create or replace function st_select()

returns table

```
(  
    _urunId INT,  
    _urunadi varchar(30),  
    _urunturu varchar(30),  
    _fiyati varchar(6)
```

```
)as
```

```
$$
```

```
begin
```

```
    return query
```

```
        select urunId, urunAdi, urunTuru, fiyati from Urunler  
        order by urunId;
```


end

\$\$

language plpgsql

--delete function

create or replace function st_delete(_urunId INT)

returns int as

\$\$

begin

delete from Urunler

where urunId = _urunId;

if found then

return 1;

else

return 0;

end if;

end

\$\$

language plpgsql

**create or replace function st_insertSiparis(_urunadi
varchar(30), _urunturu varchar(30), _fiyati varchar(6))**

returns int as

\$\$

begin

insert into Siparis(urunAdi, urunTuru, adet)

values(_urunadi, _urunturu, _fiyati);

if found then

return 1;

else

return 0;

end if;

end\$\$

language plpgsql

create or replace function st_selectAnatabak()

returns table

(

_urunadi varchar(30),

```
        _fiyati varchar(6)
    )as
    $$
begin
    return query
        select urunAdi, fiyati from Urunler where urunTuru =
        'Ana tabak' order by urunId;
end
    $$
language plpgsql
```

```
create or replace function st_selectSalata()
returns table
(
    _urunadi varchar(30),
    _fiyati varchar(6)
)as
    $$
begin
```

```
        return query

        select urunAdi, fiyati from Urunler where urunTuru =
'Salata' order by urunId;
end
$$
language plpgsql
```

```
create or replace function st_selectIcecek()
```

```
returns table
```

```
(
    _urunadi varchar(30),
    _fiyati varchar(6)
```

```
)as
```

```
$$
```

```
begin
```

```
    return query
```

```
        select urunAdi, fiyati from Urunler where urunTuru =
'Içecek' order by urunId;
```

```
end
```

```
$$
```

language plpgsql

create or replace function st_selectGarson()

returns table

(

kisiID INT,

adi varchar(30),

soyadi varchar(30),

personelTipi varchar(6),

maas INT

)as

\$\$

begin

return query

SELECT

Personel.Personal.kisiNo,

Personel.Personal.adi,

Personel.Personal.soyadi,

Personel.Personal.personelTipi,

```
Personel.Garson.maas
from Personel.Personal
INNER JOIN Personel.Garson ON Personel.Personal.kisiNo
= Personel.Garson.kisiNo;
end
$$
language plpgsql
```

```
create or replace function st_insertPersonal(_adi
varchar(30), _soyadi varchar(30), _personalTipi varchar(30),
_maas INT)
returns int as
$$
begin
insert into Personel.Personal(adi, soyadi, personelTipi)
values(_adi, _soyadi, _personalTipi);

insert into Personel.Garson(kisiNo, maas)
values(currval('Personel.Personal_kisiNo_seq'), _maas);
if found then
return 1;
```

```
    else
        return 0;
    end if;
end$$
language plpgsql
```

Kalıtım başlama

```
CREATE TABLE Personel.Personal (
    kisiNo SERIAL,
    adi varchar(40) NOT NULL,
    soyadi varchar(40) NOT NULL,
    personelTipi varchar(20) NOT NULL,
    PRIMARY KEY (kisiNo)
```

);

CREATE TABLE Personel.Garson (

kisiNo SERIAL,

maas INT NOT NULL,

PRIMARY KEY (kisiNo)

);

ALTER TABLE Personel.Garson

ADD FOREIGN KEY (kisiNo)

REFERENCES Personel.Personal(kisiNo)

ON DELETE CASCADE

ON UPDATE CASCADE;

CREATE TABLE Personel.Sef (

kisiNo SERIAL,

maas INT NOT NULL,

PRIMARY KEY (kisiNo)

);


```
ALTER TABLE Personel.Sef  
ADD FOREIGN KEY (kisiNo)  
REFERENCES Personel.Personal(kisiNo)  
ON DELETE CASCADE  
ON UPDATE CASCADE;
```

```
CREATE TABLE Personel.Kasiyer (  
kisiNo SERIAL,  
maas INT NOT NULL,  
PRIMARY KEY (kisiNo)  
);
```

```
ALTER TABLE Personel.Kasiyer  
ADD FOREIGN KEY (kisiNo)  
REFERENCES Personel.Personal(kisiNo)  
ON DELETE CASCADE  
ON UPDATE CASCADE;
```

```
CREATE TABLE Personel.Yonteci (
```

```
kisiNo SERIAL,  
maas INT NOT NULL,  
PRIMARY KEY (kisiNo)  
);
```

```
ALTER TABLE Personel.Yonteci  
ADD FOREIGN KEY (kisiNo)  
REFERENCES Personel.Personal(kisiNo)  
ON DELETE CASCADE  
ON UPDATE CASCADE;
```

TABLES

```
CREATE TABLE Urunler (  
urunId SERIAL NOT NULL,  
urunAdi varchar(50),  
urunTuru varchar(50),  
fiyati varchar(6),  
PRIMARY KEY(urunId)  
);
```

```
CREATE TABLE Fatura (  
    faturaNo SERIAL,  
    urun varchar(30) NOT NULL,  
    tur varchar(10) NOT NULL,  
    adet varchar(2) NOT NULL,  
    toplamFiyat INT,  
    PRIMARY KEY (faturaNo)  
);
```

```
CREATE TABLE Masa(  
  
    masaNo SERIAL,  
    masaDurumu varchar(30) NOT NULL,  
    PRIMARY KEY(masaNo)  
);
```

```
CREATE TABLE Istekler(
```

```
    istekNo SERIAL,  
    urun varchar(30) NOT NULL,  
    tur varchar(10) NOT NULL,  
    adet varchar(2) NOT NULL default '1',  
    PRIMARY KEY(istikNo)  
);
```

```
ALTER TABLE Istekler  
ADD FOREIGN KEY(siparisNo)  
REFERENCES Siparis(siparisNo)  
ON DELETE CASCADE  
ON UPDATE CASCADE;
```

```
CREATE TABLE Siparis (  
    siparisNo SERIAL,  
    siparisTarihi DATE NOT NULL,  
    masaNo INT NOT NULL,  
    urunId INT NOT NULL,  
    faturaNo INT NOT NULL,  
    PRIMARY KEY (siparisNo)
```

);

**ALTER TABLE Siparis
ADD FOREIGN KEY(masaNo)
REFERENCES Masa(masaNo)
ON DELETE CASCADE
ON UPDATE CASCADE;**

**ALTER TABLE Siparis
ADD FOREIGN KEY(urunId)
REFERENCES Urunler(urunId)
ON DELETE CASCADE
ON UPDATE CASCADE;**

**ALTER TABLE Siparis
ADD FOREIGN KEY(faturaNo)
REFERENCES Fatura(faturaNo)
ON DELETE CASCADE
ON UPDATE CASCADE;**

CREATE TABLE Adres(

adresID SERIAL,

kapiNo INT NOT NULL,

binaNO INT NOT NULL,

mahale varchar(50) NOT NULL,

ilceNo INT,

PRIMARY KEY(adresID),

FOREIGN KEY(ilceNo) REFERENCES Ilce(ilceNo)

);

CREATE TABLE Ilce(

ilceNo SERIAL,

ilceAdi varchar(30) NOT NULL,

postaKodu varchar(20) NOT NULL,

ilNo INT,

PRIMARY KEY(ilceNo)

);

```
ALTER TABLE Ilce ADD FOREIGN KEY(ilNo) REFERENCES  
IL(ilNo);
```

```
CREATE TABLE IL(  
  
    ilNo SERIAL,  
    ilAdi varchar(20) NOT NULL  
);
```

TRIGGERS

```
create trigger toplamTrigger  
after insert  
on Urunler  
for each row  
execute procedure toplamUrunler();
```

```
create or replace function toplamUrunler()  
returns trigger
```

as

\$\$

begin

update sayacDegeri set sayac = sayac + 1;

return new;

end;

\$\$

language plpgsql;

create table sayacDegeri(

sayac INt

);

create or replace function addto()

returns trigger as

\$addto_trigger\$

begin

insert into Istekler(urun, tur, adet)


```
        values(new.urunAdi, new.urunTuru, new.adet);  
        return new;  
end;  
$addto_trigger$  
language plpgsql;
```

```
create trigger addto_trigger  
after insert on Siparis  
for each row execute procedure addto();
```



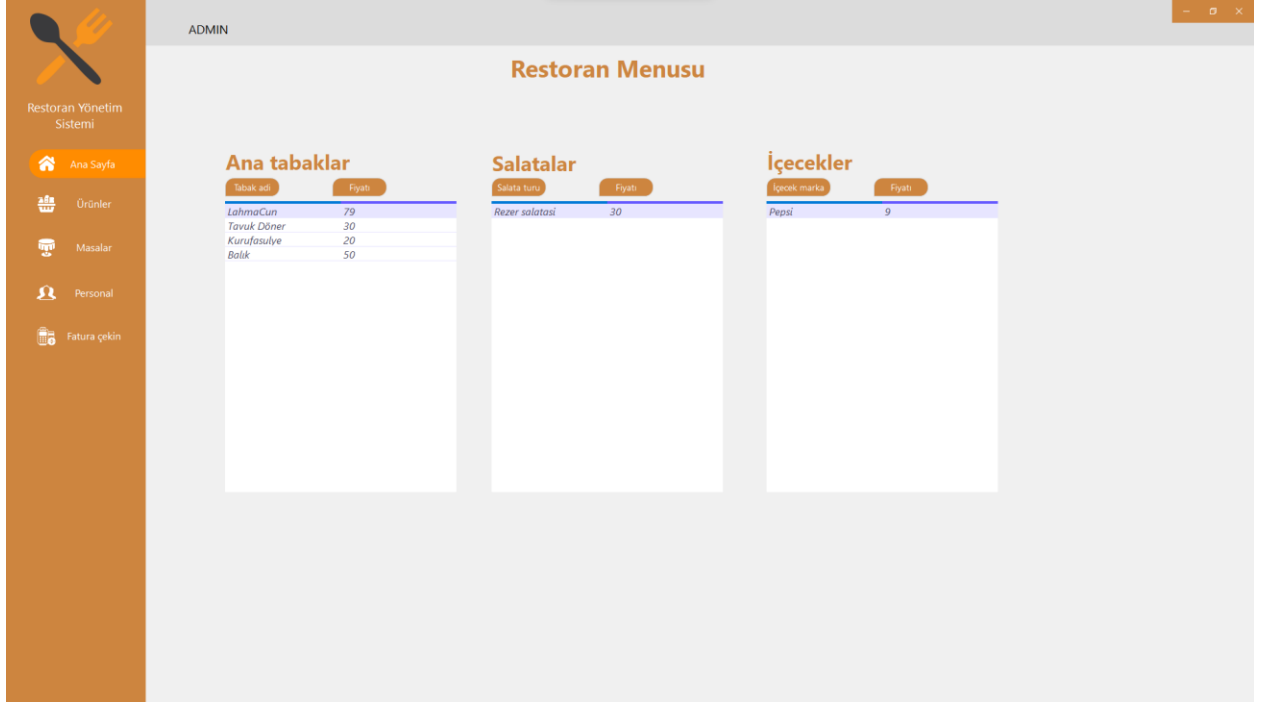
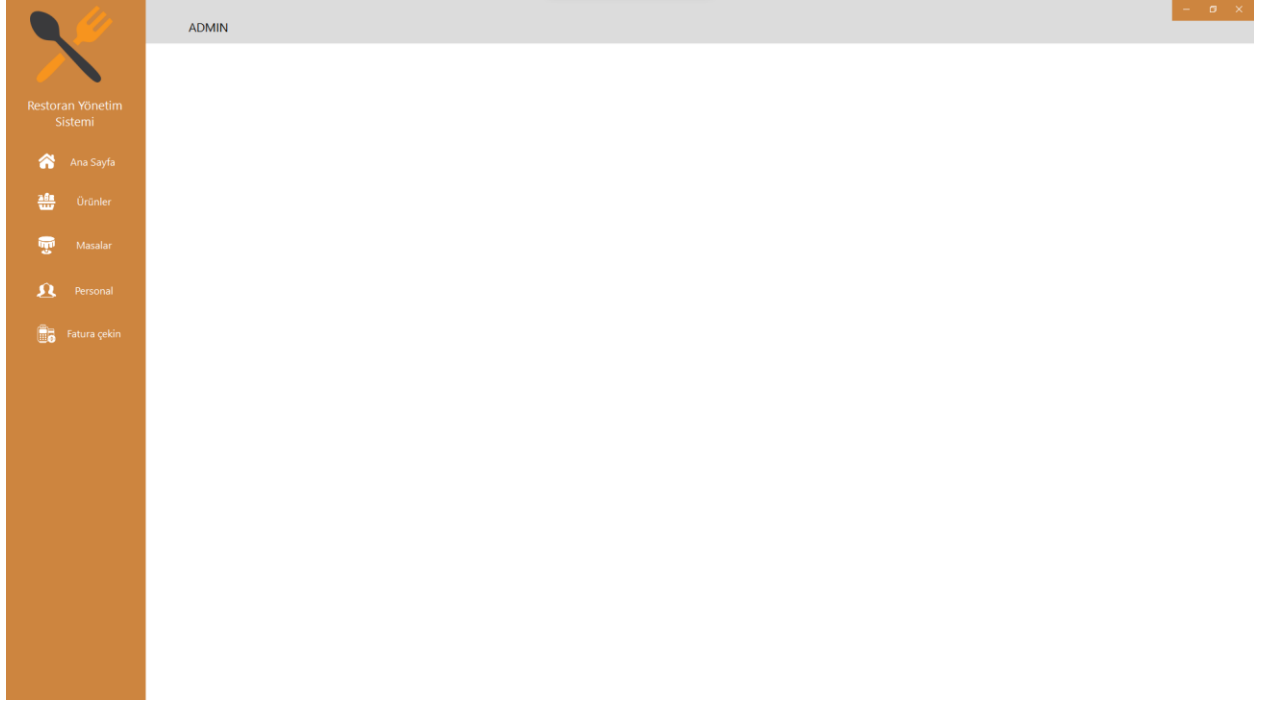
Lütfen aşağıdaki boşlukları doldurun


Yönetici

Parola

Giriş

Çıkış





Restoran Yönetim Sistemi

Ana Sayfa

Ürünler

Masalar

Personal

Fatura Çekin

ADMIN

Ürün adı

Ürün fiyatı

Ürün türü

Ekle

Güncelle


Sil

Kayıt et

Aranan Ürünün ID

Ara

#Sr	Ürün adı	Ürün türü	Fiyatlar
12	LahmaCun	Ana tabak	79
13	Tavuk Döner	Ana tabak	30
16	Kurufasulye	Ana tabak	20
17	Rezer salatası	Salata	30
18	Pepsi	İçecek	9
19	Balık	Ana tabak	50



Restoran Yönetim Sistemi

Ana Sayfa

Ürünler

Masalar

Personal

Fatura Çekin

ADMIN


Ürün adı

Ürün türü


Adeti

Sipariş et


#Sr	Ürün adı	Ürün türü	Adeti	Fiyatlar
12	LahmaCun	Ana tabak		79
13	Tavuk Döner	Ana tabak		30
16	Kurufasulye	Ana tabak		20
17	Rezer salatası	Salata		30
18	Pepsi	İçecek		9
19	Balık	Ana tabak		50




Restoran Yönetim Sistemi




Ana Sayfa




Ürünler



Masalar



Personal



Fatura Çekin

ADMIN

Ad

Soyad

Maas

Telefon No


Personal Tipi

Adress


Ekle

Kaydet


#Sr	Adi	Soyadi	Personal Tipi	Maas
6	qusav	alkaleh	Şef	8000
7	asa	sdas	Garson	sdas




Restoran Yönetim Sistemi




Ana Sayfa




Ürünler



Masalar



Personal



Fatura Çekin

ADMIN

Ürün	Tür	Adet
Tavuk doner	Ana tabak	3

Toplam Fiyat