



GAZİ ÜNİVERSİTESİ
Veritabanı Yönetim
Sistemleri BMT-311 Proje
Raporu

MAĞAZA
OTOMASYONU

AD & SOYAD:
Mehmet Demir 21181617001

Ali Osman Seyis 21181617016

İçindekiler

| | |
|-------------------------------------------------------------|----|
| 1 Proje Hakkında Bilgi | 3 |
| 2 Fonksiyon-Trigger- Procedure Kodları | 3 |
| 2.1 Trigger Fonksiyonu | 3 |
| 2.2 Procedure | 5 |
| 2.3 Fonksiyon | 5 |
| 3 Önemli Ekran Görüntüleri | 7 |
| 3.1 Giriş ve Kayıt..... | 7 |
| | 7 |
| 4 Mantıksal Fiziksel Kavramsal Proje Tasarım Adımları | 11 |
| 4.1 Kavramsal Tasarım | 11 |
| 4.2 Kavramsal Veritabanı Tasarımı | 11 |
| 4.3 Birinci Adım: Varlık Tiplerini Belirlemek | 12 |
| 4.4 Fiziksel Tasarım | 13 |
| 5 NORMALİZASYON | 13 |
| 6 Kaynakça | 13 |

1 Proje Hakkında Bilgi

Ürün, bilgilerini detaylı bir şekilde içeren bir sistem geliştirilmiştir. Bu sistemde, Postgresql veritabanını kullanarak çeşitli işlemler gerçekleştiren bir giyim mağazası tasarlanmıştır. Ekleme, silme, güncelleme ve arama gibi çeşitli metodlar, C# programlama dilinde kullanılmıştır. Bu sayede, kullanıcılar ürünleri yönetebilir, personel bilgilerini güncelleyebilir ve siparişleri kolayca takip edebilirler.

2 Fonksiyon-Trigger- Procedure Kodları

2.1 Trigger Fonksiyonu

Bu PostgreSQL fonksiyonu ve tetikleyici (trigger) kombinasyonu, bir sipariş eklendiğinde belirli bir ürünün stok miktarını güncellemek için kullanılır. İşlev ve tetikleyici şu şekilde çalışır:

Fonksiyon (siparisekle_stoksil);

Bir sipariş eklenirse (after insert on siparis tetikleyicisiyle yakalanır),

Fonksiyon, urunler tablosundaki ilgili ürünün stok miktarını azaltır (update urunler set urunstok=urunstok-1).

Tetikleyici(stoksil);

Stoksil tetikleyici, siparis tablosuna bir satır eklendiğinde çalışır (after insert on siparis).

Tetikleyici, siparisekle_stoksil adlı fonksiyonu çağırır (execute procedure siparisekle_stoksil()).

```
create or replace function siparisekle_stoksil()
returns trigger
as
$$
begin
update urunler set urunstok=urunstok-1 from siparis where
urunler.urunid=siparis.siparisurunid;
return new;
end;
$$
language 'plpgsql';

create trigger stoksil
after insert on siparis
for each row
execute procedure siparisekle_stoksil();
```

Bu işlev ve tetikleyici kombinasyonu, bir sipariş silindiğinde ilgili ürünün stok miktarını güncellemek için kullanılır. Detaylar şu şekildedir:

Fonksiyon (siparissil_stokekle):

Bir sipariş silinirse (after delete on siparis tetikleyicisiyle yakalanır),

Fonksiyon, urunler tablosundaki ilgili ürünün stok miktarını artırır (update urunler set urunstok=urunstok+1).

Tetikleyici (stokekle):

stokekle tetikleyici, siparis tablosundan bir satır silindiğinde çalışır (after delete on siparis).

Tetikleyici, siparissil_stokekle adlı fonksiyonu çağırır (execute procedure siparissil_stokekle()).

```
create or replace function siparissil_stokekle()
returns trigger
as
$$
Begin
```

```
update urunler set urunstok=urunstok+1 from siparis where urunler.urunid=siparis.siparisurunid;
return new;
end;
$$
language 'plpgsql';
```

```
create trigger stokekle
after delete on siparis
for each row
execute procedure siparissil_stokekle();
delete from siparis where siparisid = 1
select * from urunler where urunid=2
select * from urunler order by urunid
select * from siparis
```

2.2 Procedure

Bu kodun içindeki prosedür, bir tetikleyici (trigger) ile ilişkilidir ve aşağıdaki görevi yerine getirir:

Bu prosedür, stokekle adlı bir tetikleyici tarafından çağrılır. Tetikleyici, siparis tablosundan bir satır silindiğinde çalışır (after delete on siparis). İlgili prosedür, urunler tablosundaki ilgili ürünün stok miktarını bir arttırarak çalışır. Bu şekilde, bir sipariş silindiğinde ilgili ürünün stok miktarı güncellenir.

Not: Verilen kod bloğu içindeki delete from siparis where siparisid = 1 ve ardından gelen select ifadeleri ayrı sorgulardır ve bu kod içinde doğru bir şekilde yer almamaktadır. Bu ifadelerin amacını ve kullanımını doğrulamak önemlidir.

```
create or replace function siparissil_stokekle()
returns trigger
as
$$
Begin

    update urunler set urunstok=urunstok+1 from siparis where
urunler.urunid=siparis.siparisurunid;
    return new;
end;
$$

language 'plpgsql';

create trigger stokekle
after delete on siparis
for each row
execute procedure siparissil_stokekle();
```

2.3 Fonksiyon

Kampanya kurulduğu zaman ürünün yeni fiyatını hesaplayan fonksiyondur.

```
create function kampanyalı(s float ,b double precision)
returns float
language plpgsql
```

```
as
$$
declare
kampanyalıfiyat float;
begin
kampanyalıfiyat :=(b*(100-s)) / 100; $$ LANGUAGE plpgsql;

return kampanyalıfiyat;
end;
$$;
```

```
CREATE OR REPLACE PROCEDURE delete_transaction(
```

```
    IN t_num INTEGER

)

AS $$

BEGIN

DELETE FROM transactions WHERE tnum=t_num;

END;

$$ LANGUAGE plpgsql;
```

Yukarıdaki prosedürde parametre olarak “t_num” kullanılmaktadır. tnum parametresi “transactions” tablosundaki “tnum” sütunundaki verilerle kıyaslanıp eşleşme olduğu takdirde ilgili satırdaki verileri silecektir.

3 Önemli Ekran Görüntüleri

3.1 Giriş ve Kayıt

The image displays two screenshots of a web application interface, likely for a login and registration system. The top window, titled "giris", features a light blue background and contains two input fields: "E-Mail" and "Şifre". Below these fields are two buttons: "Kaydol" and "Giriş Yap". The bottom window, titled "Kayıt", also has a light blue background and contains three input fields: "Kullanıcı Adı", "E Mail", and "Şifre". Below these fields is a single button labeled "Kayıt Ol".

giris

E-Mail

Şifre

Kaydol

Giriş Yap

Kayıt

Kullanıcı Adı

E Mail

Şifre

Kayıt Ol

Kategoriler

KATEGORİ ID :
KATEGORİ AD :

Listele

Ekle

Sil

Ara

Güncelle

MENO İÇİN

Kampanya

| | urunid | urunad | urunstok | urunsatis | kategoriad | bedenad |
|---|--------|----------------|----------|-----------|------------|------------|
| ▶ | 2 | Beyaz Gömlek | 50 | 229 | Erkek | L Beden |
| | 4 | Kırmızı Gömlek | 50 | 249 | Kadın | M Beden |
| | 6 | Siyah Tişört | 80 | 159 | Erkek | L Beden |
| | 7 | Mavi Tişört | 75 | 149 | Kadın | S Beden |
| | 8 | Mavi Tişört | 75 | 149 | Kadın | M Beden |
| | 9 | Mont | 40 | 429 | Erkek | L Beden |
| | 10 | Mont | 40 | 469 | Kadın | M Beden |
| | 13 | Bere | 60 | 89 | Unisex | Standart B |
| | 11 | a | 5 | 25 | Kadın | S Beden |
| | 12 | Memoli | 1 | 1 | Kadın | S Beden |

Ürünleri
Listele

Kampanya ID :
Kampanyalı Ürün :
İndirim Oranı (%) :

Listele

Ekle

Sil

Ara

Güncelle

MENO İÇİN

Müşteri

| | musteriid | musteriadsoyad |
|---|-----------|-----------------|
| ▶ | 1 | Patrick BATE... |
| | 2 | Mahmut Yıldırım |
| | 3 | Leyla Kanık |
| | 4 | Ali Osman SE... |
| | 5 | Mehmet DEMİR |
| * | | |

Müşteri ID :

Müşteri AdVeSoyad :

Listele

Ekle

Sil

Ara

Güncelle

MENO İÇİN

Siparis ID :

Siparis Urun :

Siparis Musteri :

Urun
Listele

Listele

Ekle

Sil

MENO İÇİN

Ara

Güncelle

Tedarikçi

TEDARIKÇI ID :

TEDARIKÇI AD :

Listele

Ekle

Sil

Ara

Güncelle

MENO İÇİN

Form1

URUN ID :

URUN AD :

URUN STOK :

URUN ALIS :

URUN SATIS :

KATEGORİ :

TEDARİKÇİ :

BEDEN :

URUN SİGORTA :

MENO İÇİN

| | urunid | urunad | urunstok | urunalis | urunsatis |
|---|--------|----------------|----------|----------|-----------|
| ▶ | 1 | Beyaz Gömlek | 95 | 120 | 250 |
| | 2 | Beyaz Gömlek | 50 | 165 | 229 |
| | 3 | Kırmızı Gömlek | 48 | 180 | 249 |
| | 4 | Kırmızı Gömlek | 50 | 180 | 249 |
| | 5 | Siyah Tişört | 76 | 100 | 159 |
| | 6 | Siyah Tişört | 80 | 100 | 159 |
| | 7 | Mavi Tişört | 75 | 85 | 149 |
| | 8 | Mavi Tişört | 75 | 85 | 149 |
| | 9 | Mont | 40 | 265 | 429 |
| | 10 | Mont | 40 | 285 | 469 |
| * | | | | | |

Liste

Ekle

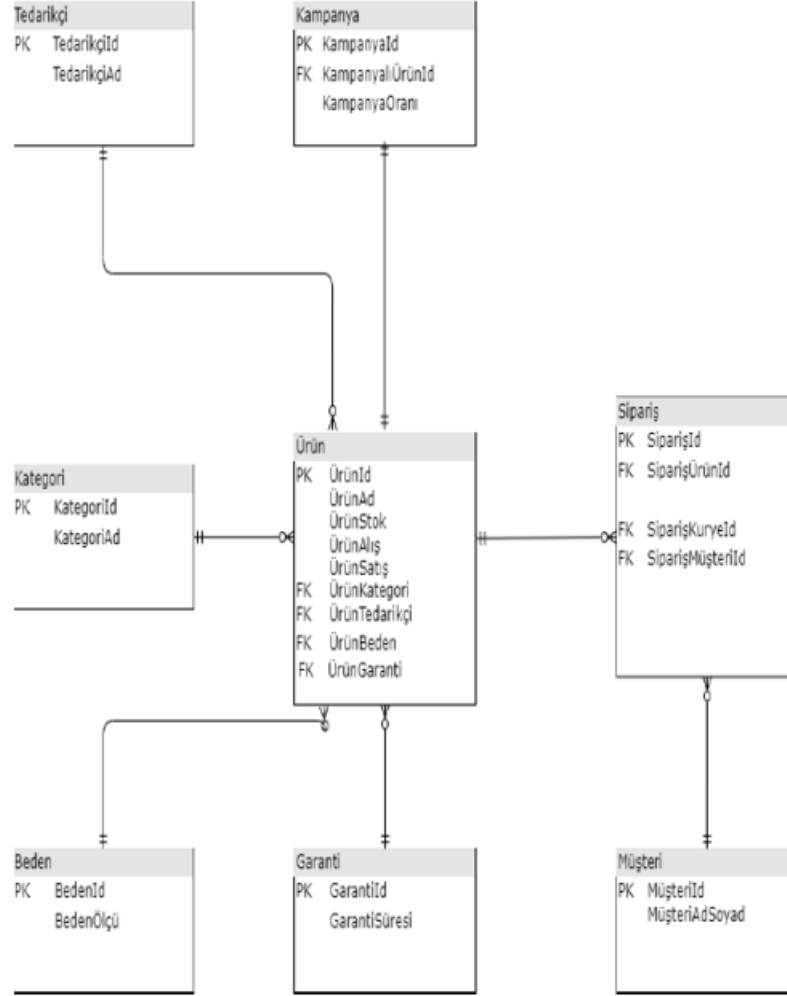
Sil

Ara

Güncelle

4 Mantıksal Fiziksel Kavramsal Proje Tasarım Adımları

4.1 Kavramsal Tasarım



4.2 Kavramsal Veritabanı Tasarımı

- Nitelik alanlarını belirlemek
- İlişki tiplerini belirlemek
- Varlık tiplerini belirlemek
- Varlık veya ilişki tipleriyle nitelikleri belirlemek ve birleştirmek
- Aday, birincil ve alternatif anahtar niteliklerini belirlemek
- Gelişmiş modelleme kavramlarının kullanımını düşünmek (isteğe bağlı adım)
- Fazlalık için modeli kontrol etmek
- Kullanıcı hareketlerine karşı kavramsal modeli doğrulamak
- Kullanıcı ile kavramsal veri modelini gözden geçirmek

Kavramsal veritabanının tasarımında, kavramsal veri modelini oluşturmak için gereken adımlar sırası ile gerçekleştirilmiştir.

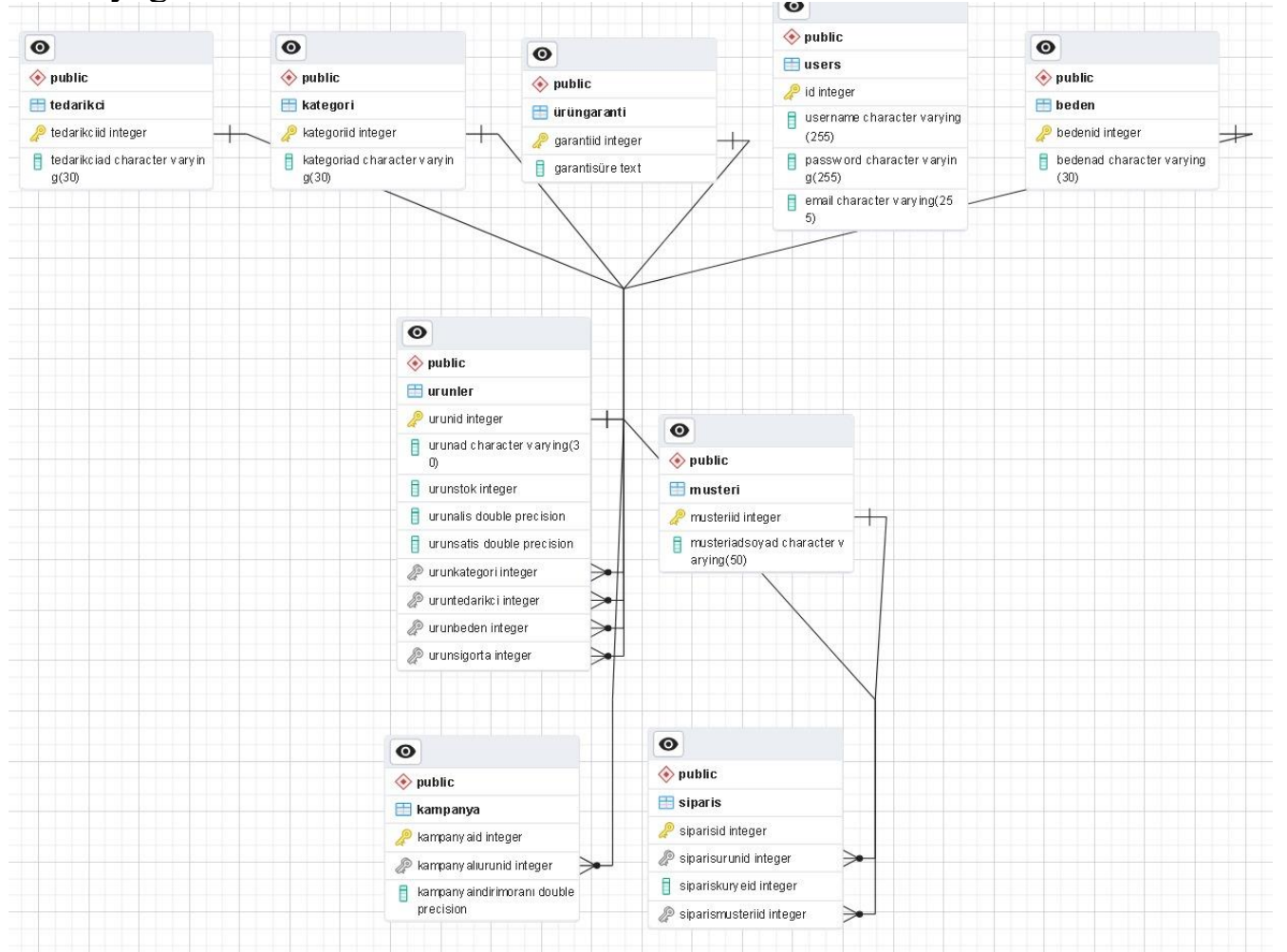
4.3 Birinci Adım: Varlık Tiplerini Belirlemek

Öncelikle varlık tipleri ve açıklamaları ile ilgili tablo oluşturulmuştur

| EntityName | Description | Aliases | Occurrence |
|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|------------------------------------------------------------------------------|
| Supply | Ürünlerin tadarığı ile ilgilenen terim | tedarikçi | Ürünlerin stok kontrolünü yapar. |
| Category | Aralarında herhangi bir bakımdan ilgi veya benzerlik bulunan şeylerin tamamı | kategori | Müşterilerin seçim yapabileceği kısım. |
| Warranty | İmalatçı-üretici veya ithalatçıların, sattıkları, ücretsiz olarak yenisi ile değiştirilmesinin, tamirinin, bedel iadesinin veya bedel indiriminin taahhüt edildiğini söyleyen terim | ürüngaranti | Garanti kısmına bakarak ürünleri tercih yapabilirler |
| Customer | Sunulan bir ürün ya da hizmetin tüketicisidir. | musteri | Müşterilerin listelendiği kısımdır. |
| Body | Bir şeyin ölçüleri uzunluğu, boyu, genişliği içeren terim | beden | Müşterilerin isteğine göre seçebileceği farklı boydaki ürünleri temsil eder. |
| Products | Bir müşterinin arzusunu veya ihtiyacını karşılamak için bir pazara sunulabilecek herhangi bir şeydir. | urunler | Ürünlerin listelendiği kısımdır . |
| Order | Bir ürünün veya hizmetin alınması terimi | siparis | Siparişlerin takip edileceği kısımdır. |
| Offer | Geniş bir kitleye ulaşmayı hedefleyen tanıtım, reklam veya satış faaliyetlerini içeren terim | kampanya | Kampanyaların görüneceği kısımdır. |

4.4 Fiziksel Tasarım

ER Diyagramı



5 NORMALİZASYON

1. Normal Form (1NF):

Her tablo, atomik değerlere sahip olacak şekilde düzenlenmiştir. Her hücrede tek bir değer bulunmaktadır. Projemizde her tablo bu şartları sağlamaktadır.

2. Normal Form (2NF):

Her tablonun tüm alanları, tablonun anahtarlarına tam olarak bağımlıdır. İki anahtar alan içeren tablolar ayrılmıştır. Projemizde her tablo anahtarlar a tam olarak bağımlıdır.

3. Normal Form (3NF):

Her tablonun alanları, anahtarları dışında hiçbir başka anahtar alanına bağlı değildir. Bağımlılıklar ortadan kaldırılmıştır.

6 Kaynakça

1. <https://chat.openai.com/>
2. <https://michaelscodingspot.com/postgres-in-csharp/>