






Dersin Kodu:	BIMU3064
Dersin Adı:	Veri Tabanı Yönetim Sistemleri
Öğrenci Numaram:	1306170114
Adım:	Enver
Soyadım:	Usta
Ödev Numarası:	Ödev4

**1.** Fis tablosundaki toplam alanını gerektiğinde güncelleyen trigger(lar)ı yazınız. İpucu: fisUrunleri tablosuna bir kayıt eklendiğinde, silindiğinde veya bir kayıttaki (barkod veya miktar veya birimFiyat alanları) guncellendiğinde çalışacak olan bu trigger fiş toplamı değerini fiş ürünlerindeki “miktar x birimFiyat” değerlerini toplayarak bulur.

```
CREATE OR REPLACE FUNCTION func_fis_toplam_guncelle()
RETURNS TRIGGER AS $fis_toplam_guncelle$
BEGIN
    IF(TG_OP = 'DELETE') THEN
        update fis set toplam=toplam-
(NEW.miktar * NEW.birimfiyat) WHERE OLD.fisno=fis.fisno;
        RETURN OLD;
    ELSIF (TG_OP = 'INSERT') THEN
        update fis set toplam=toplam+(NEW.miktar * NEW.birimfiyat) WHERE NEW.f
isno=fis.fisno;
        RETURN NEW;
    ELSIF(TG_OP = 'UPDATE') THEN
        update fis set toplam=toplam-
(OLD.miktar * OLD.birimfiyat)+(NEW.miktar * NEW.birimfiyat) WHERE NEW.fisno=fi
s.fisno;
        RETURN NEW;
    END IF;
END;
$fis_toplam_guncelle$
LANGUAGE plpgsql;
```

```
CREATE TRIGGER trigger_fis_toplam_guncelle
AFTER INSERT OR UPDATE OR DELETE
ON fisurunleri
FOR EACH ROW
EXECUTE PROCEDURE func_fis_toplam_guncelle();
```

## Fis tablosundaki ilk görünümüm:

Data Output		Explain	Messages	Notifications
	<b>fisno</b> [PK] integer 	<b>tarihsaat</b> timestamp without time zone 	<b>toplam</b> integer 	<b>tckimlik</b> bigint 
1	2	2021-01-01 23:26:39.190508	12	48017060002
2	1	2020-01-01 18:15:00	39	48008062090
3	3	2021-01-02 00:16:57.140897	8	48008062090

## Tablomuzdaki Değişimler:

INSERT INTO fisurunleri VALUES(2, 124, 5, 'kg', 10);

	fisno [PK] integer	tarihsaat timestamp without time zone	toplam integer	tckimlik bigint
1	1	2020-01-01 18:15:00	39	48008062090
2	3	2021-01-02 00:16:57.140897	8	48008062090
3	2	2021-01-01 23:26:39.190508	62	48017060002

--Az önceki eklediğim ürünün 5kg olan miktarını 10kg olarak güncelleyeceğim

--Böylece 2 numaralı fişteki 62tl'nin tekrar değişmesi gerekmekte.

UPDATE fisurunleri SET miktar=10 WHERE fisno=2 and barkod=124;

	fisno [PK] integer	tarihsaat timestamp without time zone	toplam integer	tckimlik bigint
1	1	2020-01-01 18:15:00	39	48008062090
2	3	2021-01-02 00:16:57.140897	8	48008062090
3	2	2021-01-01 23:26:39.190508	112	48017060002

--Şimdi ise az önce ekleyip hem de güncellediğim 124 barkod numaralı ürünü

--sileceğim. En baştaki görünüme geri gelmesi gerekiyor.

DELETE FROM fisurunleri WHERE barkod=124 AND fisno=2;

	fisno [PK] integer	tarihsaat timestamp without time zone	toplam integer	tckimlik bigint
1	1	2020-01-01 18:15:00	39	48008062090
2	3	2021-01-02 00:16:57.140897	8	48008062090
3	2	2021-01-01 23:26:39.190508	12	48017060002

2. Ürün tablosundaki stokMiktari alanını gerektiğinde güncelleyen trigger(lar)ı yazınız.  
İpucu: stokMiktari bir ürünün tüm alımlardaki(fatura) miktarlar toplamından tüm satışlardaki miktarlar toplamının çıkarılmasıyla hesaplanabilir. Bu hesaplamadan birim alanı kullanılmaz.

Fiş Ürünleri için yazdığım trigger:

```
CREATE OR REPLACE FUNCTION func_fis_urun_stok_guncelle()
RETURNS TRIGGER AS $fis_urun_stok_guncelle$
BEGIN
    IF(TG_OP = 'INSERT') THEN
        UPDATE urun SET stokmiktari = (stokmiktari - NEW.miktar)
        WHERE barkod=NEW.barkod;
        RETURN NEW;
    ELSIF(TG_OP = 'DELETE') THEN
        UPDATE urun SET stokmiktari = (stokmiktari + OLD.miktar)
        WHERE barkod=OLD.barkod;
        RETURN OLD;
    ELSIF(TG_OP = 'UPDATE') THEN
        UPDATE urun SET stokmiktari =
            (stokmiktari + OLD.miktar - NEW.miktar)
        WHERE barkod=OLD.barkod;
        RETURN NEW;
    END IF;
END;
$fis_urun_stok_guncelle$ LANGUAGE plpgsql;

CREATE TRIGGER trigger_fis_urun_stok_guncelle
AFTER INSERT OR UPDATE OR DELETE
ON fisurunleri
FOR EACH ROW EXECUTE PROCEDURE func_fis_urun_stok_guncelle();
```

Ürün tablomdaki ilk görünüm:







Data Output		Explain	Messages	Notifications	
	barkod [PK] integer	adi character varying (30)	stokmiktari integer	uruntoplamkar integer	satisfiyattrendi character varying (30)
1	125	biskuvi	69	0	artan
2	123	ekmek	136	0	azalan
3	124	un	80	0	[null]

## Tablomuzdaki Değişimler:







--5kg miktarındaki un'u fisurunlerine ekliyorum.

--stok miktarının 5kg azalması bekleniyor.







INSERT INTO fisurunleri VALUES(2, 124, 5, 'kg', 10);

Data Output		Explain	Messages	Notifications	
	barkod [PK] integer 	adi character varying (30) 	stokmiktari integer 	uruntoplamlar integer 	satisfiyattrendi character varying (30) 
1	125	biskuvi	69	0	artan
2	123	ekmek	136	0	azalan
3	124	un	75	0	[null]

UPDATE fisurunleri SET miktar=10 WHERE fisno=2 and barkod=124;

Data Output		Explain	Messages	Notifications	
	barkod [PK] integer 	adi character varying (30) 	stokmiktari integer 	uruntoplamlar integer 	satisfiyattrendi character varying (30) 
1	125	biskuvi	69	0	artan
2	123	ekmek	136	0	azalan
3	124	un	70	0	[null]

DELETE FROM fisurunleri WHERE barkod=124 AND fisno=2;

Data Output		Explain	Messages	Notifications						
	barkod [PK] integer		adi character varying (30)		stokmiktari integer		uruntoplamlar integer		satisfiyattrendi character varying (30)	
1		125	biskuvi			69		0	artan	
2		123	ekmek			136		0	azalan	
3		124	un			80		0	[null]	

Fatura Ürünleri için yazdığım trigger:

```
CREATE OR REPLACE FUNCTION func_fatura_urun_stok_guncelle()
RETURNS TRIGGER AS $fatura_urun_stok_guncelle$
BEGIN
    IF(TG_OP = 'INSERT') THEN
        UPDATE urun SET stokmiktari = (stokmiktari + NEW.miktar)
        WHERE barkod=NEW.barkod;
        RETURN NEW;
    ELSIF(TG_OP = 'DELETE') THEN
        UPDATE urun SET stokmiktari = (stokmiktari - OLD.miktar)
        WHERE barkod=OLD.barkod;
```

```

        RETURN OLD;
    ELSIF(TG_OP = 'UPDATE') THEN
        UPDATE urun SET stokmiktari = (stokmiktari - OLD.miktar + NEW.mikt
ar)
        WHERE barkod=OLD.barkod;
        RETURN NEW;
    END IF;
END;
$fatura_urun_stok_guncelle$ LANGUAGE plpgsql;

```

```







CREATE TRIGGER trigger_fatura_urun_stok_guncelle
AFTER INSERT OR UPDATE OR DELETE
ON faturaurunleri
FOR EACH ROW EXECUTE PROCEDURE func_fatura_urun_stok_guncelle();

```







-- Faturaurunlerindeki deęişimlerde bizim stok miktarımızı etkileyecektir.  
-- 2 numaralı faturamıza biraz un ekleyelim ve stoktaki deęişimi gözlemleyelim.

**NOT:** Aralarda denemeler yaptığım için tablolarımda sorular arasında geçiş yaparken de deęişimler gözlenebilir, lütfen benim önceki şeklinde gösterdiğim tabloyu dikkate alınız.

### Ürün Tablom:

Data Output		Explain	Messages	Notifications	
	barkod [PK] integer 	adi character varying (30) 	stokmiktari integer 	uruntoplamlar integer 	satisfiyattrendi character varying (30) 
1	124	un	79	0	[null]
2	125	biskuvi	67	0	artan
3	123	ekmek	132	0	azalan

**INSERT INTO faturaurunleri VALUES(2, 124, 100, 'kg', 1);**

Data Output		Explain	Messages	Notifications	
	barkod [PK] integer 	adi character varying (30) 	stokmiktari integer 	uruntoplamlar integer 	satisfiyattrendi character varying (30) 
1	125	biskuvi	67	0	artan
2	123	ekmek	132	0	azalan
3	124	un	179	0	[null]

--100kg un fazla gelmiş gibi duruyor, bunu 50kg olarak güncelleyelim.

UPDATE faturaurunleri SET miktar=50 WHERE faturano=2 AND barkod=124;

Data Output		Explain	Messages	Notifications	
	barkod [PK] integer	adi character varying (30)	stokmiktari integer	uruntoplamlar integer	satisfiyattrendi character varying (30)
1	125	biskuvi	67	0	artan
2	123	ekmek	132	0	azalan
3	124	un	129	0	[null]

--faturaurunlerinden kaldırıyoruz. Birdaha kg'ını 1tl'den un nasıl buluruz hiç bilmiyorum :(

DELETE FROM faturaurunleri WHERE faturano=2 AND barkod=124;

Data Output		Explain	Messages	Notifications	
	<div>barkod</div> <div>[PK] integer</div>	<div>adi</div> <div>character varying (30)</div>	<div>stokmiktari</div> <div>integer</div>	<div>uruntoplamlar</div> <div>integer</div>	<div>satisfiyattrendi</div> <div>character varying (30)</div>
1	125	biskuvi	67	0	artan
2	123	ekmek	132	0	azalan
3	124	un	79	0	[null]

3. Urun tablosundaki satisFiyatTrendi alanını gerektiğinde güncelleyen trigger(lar)ı yazınız. İpucu: satisFiyatTrendi alanı “artan” veya “azalan” olabilir. Bu değer ürünün tarihSaat'e göre son iki satışındaki (fisUrunlari) birimFiyat değerlerinin karşılaştırılmasıyla bulunur: Son satıştaki fiyat bir öncesinden büyükse trend “artan”, değilse “azalan” olacaktır.

### Önemli Not!

Şayet kendisinden önce eklenen bir değer yoksa satisfiyattrendini NULL olarak eklemektedir.

```
CREATE OR REPLACE FUNCTION func_urun_satis_fiyat_trendi_guncelle()
RETURNS TRIGGER AS $urun_satis_fiyat_trendi_guncelle$
DECLARE
    prevurunbirimfiyat integer;
BEGIN
    IF(TG_OP = 'INSERT') THEN
        prevurunbirimfiyat = (SELECT birimfiyat FROM fisurunleri
                             WHERE NEW.barkod=barkod AND fisno=(
                                 SELECT fisno FROM fis
```

```

ORDER BY tarihsaat DESC
LIMIT 1 OFFSET 1
)
);

IF prevurunbirimfiyat is NULL THEN
    UPDATE urun SET satisfiyattrendi=NULL
    WHERE barkod=NEW.barkod;
ELSIF(prevurunbirimfiyat < NEW.birimfiyat) THEN
    UPDATE urun SET satisfiyattrendi='artan'
    WHERE barkod=NEW.barkod;
ELSIF(prevurunbirimfiyat > NEW.birimfiyat) THEN
    UPDATE urun SET satisfiyattrendi='azalan'
    WHERE barkod=NEW.barkod;
END IF;
RETURN NEW;
END IF;
END;
$urun_satis_fiyat_trendi_guncelle$ LANGUAGE plpgsql;







```

```

CREATE TRIGGER trigger_urun_satis_fiyat_trendi_guncelle
AFTER INSERT
ON fisurunleri
FOR EACH ROW EXECUTE PROCEDURE func_urun_satis_fiyat_trendi_guncelle();

```

### Ürün tablomdaki ilk görünüm:

Data Output		Explain	Messages	Notifications
 barkod [PK] integer 	adi character varying (30) 	stokmiktari integer 	uruntoplamlar integer 	satisfiyattrendi character varying (30) 
1	124	un	79	0 [null]
2	125	biskuvi	67	0 artan
3	123	ekmek	134	0 artan

-- Fırsosu 5 olan yeni bir fiş oluşturalım (4 numaralı fişte 10tl gibi fahiş bir fiyattan satılan ekmeği 5 numaralı fişte insancıl değer olan 1tl'den satalım. Fırsatçılara yer yok!!!)

```
INSERT INTO fis VALUES(5, CURRENT_TIMESTAMP, 0, 48008062090);
```

```
INSERT INTO fisurunleri VALUES(5, 123, 2, 'adet', 1);
```

Data Output		Explain	Messages	Notifications	
	barkod [PK] integer	adi character varying (30)	stokmiktari integer	uruntoplamlar integer	satisfiyattrendi character varying (30)
1	124	un	79	0	[null]
2	125	biskuvi	67	0	artan
3	123	ekmek	132	0	azalan

olması gerektiği gibi trend azalan

olması gerektiği gibi trend azalan

4. Fatura tablosundaki toplan alanını gerektiğinde güncelleyen trigger(lar)ı yazınız. İpucu: faturaUrunleri tablosuna bir kayıt eklendiğinde, silindiğinde veya bir kayıttaki (barkod veya miktar veya birimFiyat alanları) guncellendiğinde çalışacak olan bu trigger toplam değerini fatura ürünlerindeki “miktar x birimFiyat” değerlerini toplayarak bulur.

```
CREATE OR REPLACE FUNCTION func_fatura_toplam_guncelle()
RETURNS TRIGGER AS $fatura_toplam_guncelle$
BEGIN
    IF(TG_OP = 'DELETE') THEN
        update fatura set toplam=toplam-
(OLD.miktar * OLD.birimfiyat) WHERE OLD.faturano=fatura.faturano;
        RETURN OLD;
    ELSIF (TG_OP = 'INSERT') THEN
        update fatura set toplam=toplam+(NEW.miktar * NEW.birimfiyat) WHERE NE
W.faturano=fatura.faturano;
        RETURN NEW;
    ELSIF(TG_OP = 'UPDATE') THEN
        update fatura set toplam=toplam-
(OLD.miktar * OLD.birimfiyat)+(NEW.miktar * NEW.birimfiyat) WHERE NEW.faturano
=fatura.faturano;
        RETURN NEW;
    END IF;
END;
$fatura_toplam_guncelle$
LANGUAGE plpgsql;
```

```
CREATE TRIGGER trigger_fatura_toplam_guncelle
AFTER INSERT OR UPDATE OR DELETE
ON faturaurunleri
FOR EACH ROW
EXECUTE PROCEDURE func_fatura_toplam_guncelle();
```






**Fatura tablosundaki ilk görünümüm:**

Data Output		Explain	Messages	Notifications
	faturano [PK] integer	tarih date	toplam integer	verginos integer
1	1	2021-01...	320	111
2	2	2021-01...	372	112








## Tablomuzdaki Değişimler:






INSERT INTO faturaurunleri VALUES(2, 124, 100, 'kg', 1);

Data Output		Explain	Messages	Notifications	
	faturano [PK] integer 	tarih date 	toplam integer 	vergino integer 	
1	1	2021-01...	320	111	
2	2	2021-01...	472	112	

UPDATE faturaurunleri SET miktar=50 WHERE faturano=2 AND barkod=124;

Data Output		Explain	Messages	Notifications
	faturano [PK] integer 	tarih date 	toplam integer 	vergino integer 
1	1	2021-01...	320	111
2	2	2021-01...	422	112

DELETE FROM faturaurunleri WHERE faturano=2 AND barkod=124;

Data Output		Explain	Messages	Notifications
	faturano [PK] integer 	tarih date 	toplam integer 	vergino integer 
1	1	2021-01...	320	111
2	2	2021-01...	372	112

5. Tedarikçi tablosundaki faturaSayisi, faturaToplamı, faturaOrtalaması alanlarını gerektiğinde güncelleyen trigger(lar)ı yazınız. İpucu: fatura tablosuna kayıt eklenmesi, silinmesi veya bu tablodaki toplam değerinin güncellenmesi durumunda çalışacak bu trigger, soru 4'teki trigger'I dolaylı olarak kullanılmalıdır. Çünkü faturaUrunleri tablosuna bir kayıt eklenmesi, silinmesi veya güncelleme durumunda tedarikci tablosundaki FaturaToplamı ve faturaOrtalaması değerlerinin yeniden hesaplanması gerekecektir.

```
CREATE OR REPLACE FUNCTION func_tedarikci_guncelle()
RETURNS TRIGGER AS $tedarikci_guncelle$
BEGIN
    IF (TG_OP = 'INSERT') THEN
        UPDATE tedarikci
        SET faturasayisi= faturasayisi + 1, faturatoplami= faturatoplami+NEW.t
        oplam, faturaortalaması=( (faturatoplami+NEW.toplam)/(faturasayisi+1) )
        WHERE NEW.vergino=tedarikci.vergino;
```

```

        RETURN NEW;
    ELSIF(TG_OP = 'DELETE') THEN
        UPDATE tedarikci
        SET faturasayisi= faturasayisi - 1, faturatoplami= faturatoplami-
        OLD.toplam, faturaortalamasi=( (faturatoplami-OLD.toplam)/(faturasayisi-1) )
        WHERE NEW.vergino=tedarikci.vergino;
        RETURN OLD;
    ELSIF(TG_OP = 'UPDATE') THEN
        UPDATE tedarikci
        SET faturatoplami= faturatoplami-
        OLD.toplam+NEW.toplam, faturaortalamasi=( (faturatoplami-faturatoplami-
        OLD.toplam+NEW.toplam)/(faturasayisi) )
        WHERE NEW.vergino=tedarikci.vergino;
        RETURN NEW;
    END IF;
END;
$tedarikci_guncelle$
LANGUAGE plpgsql;








```

```

CREATE TRIGGER trigger_tedarikci_guncelle
AFTER INSERT OR UPDATE OR DELETE
ON fatura
FOR EACH ROW
EXECUTE PROCEDURE func_tedarikci_guncelle();

```

## Tedarikci tablosundaki ilk görünümüm

Data Output		Explain	Messages	Notifications		
 vergino [PK] integer	 isim character varying (30)	 tel integer	 eposta character varying (30)	 faturasayisi integer	 faturatoplami integer	 faturaortalamasi integer
1	111 mehm	50312	mehmet@gmail.com	0	0	0
2	112 ahmet	50313	ahmet@gmail.com	1	372	372

## Tablomuzdaki Değişimler:





--Yeni bir fatura ekleyelim 112 nolu Ahmet tedarikcisine(toplam değerini elle 100 giriyorum 0 yerine).

**INSERT INTO fatura VALUES(3, CURRENT\_TIMESTAMP, 100, 112);**

Data Output								Explain	Messages	Notifications				
	vergino [PK] integer		isim character varying (30)		tel integer		eposta character varying (30)		faturasayisi integer		faturatoplami integer		faturaortalamasi integer	
1		111	mehmet		50312		mehmet@gmail.com		0		0		0	
2		112	ahmet		50313		ahmet@gmail.com		2		472		236	

--Az önceki 100 olan toplam değerini 50 yapalım.









**UPDATE fatura SET toplam=50 WHERE faturano=3;**

Data Output		Explain	Messages	Notifications		
 vergino [PK] integer	 isim character varying (30)	 tel integer	 eposta character varying (30)	 faturasayisi integer	 faturatoplami integer	 faturaortalamasi integer
1	111	mehmet	50312	mehmet@gmail.com	0	0
2	112	ahmet	50313	ahmet@gmail.com	2	422

--Bu yeni oluşturduğumuz faturaya(faturanosu 3 olan) yeni bir faturaurunu ekleyelim ve bakalım önceki triggerimiz bu triggerimizi tetikliyor mu.

--1000adet ekmek ekliyoruz yeni faturamıza.








**INSERT INTO faturaurunleri VALUES(3, 123, 1000, 'adet', 1);**

Data Output		Explain	Messages	Notifications			
	 vergino [PK] integer	 isim character varying (30)	 tel integer	 eposta character varying (30)	 faturasayisi integer	 faturatoplami integer	 faturaortalamasi integer
1	111	mehmet	50312	mehmet@gmail.com	0	0	0
2	112	ahmet	50313	ahmet@gmail.com	2	1422	711

--Eklediğimiz 1000 ekmeği kaldıralım.

--Unutma! Faturayı tamamen silmiyoruz sadece içindeki faturaurununu kaldırıyoruz.

**DELETE FROM faturaurunleri WHERE faturano=3 AND barkod=123;**

Data Output		Explain	Messages	Notifications		
 vergino [PK] integer 	isim character varying (30) 	tel integer 	eposta character varying (30) 	faturasayisi integer 	faturatoplami integer 	faturaortalamasi integer 
1	111	mehmet	50312	mehmet@gmail.com	0	0
2	112	ahmet	50313	ahmet@gmail.com	2	422

**6.** "... satıslar(barkod, tarihSaat) returns table..." gibi tanımlanan barkodu verilen bir ürünün verilen bir tarihSaatten sonraki tüm satış kayıtlarını (fisUrunlari tablosundaki kayıtları) döndüren stored function'ı yazınız ve bu fonksiyonu çalıştığını gösterebilmek için bir SQL komutunda kullanınız. İpucu: "... satıslar(barkod, tarihSaat) returns table..."

**CREATE OR REPLACE FUNCTION satıslar(barkodumuz integer, tarihsaatimiz timestamp)**  
**RETURNS TABLE (**

    fisno integer,  
    barkod integer,  
    miktar integer,  
    birim varchar(10),  
    birimfiyat integer

**) AS \$\$**

**BEGIN**

**RETURN QUERY SELECT fisurunleri.\***

**FROM fisurunleri, fis**

**WHERE fisurunleri.fisno=fis.fisno AND**

**fisurunleri.barkod=barkodumuz AND**

**fis.tarihsaat > tarihsaatimiz;**

**END;**

**\$\$ LANGUAGE plpgsql;**

## Fisurunleri tablosundaki görünümüm:







Data Output		Explain	Messages	Notifications	
	<div>fisno</div> <div>[PK] integer</div>	<div>barkod</div> <div>[PK] integer</div>	<div>miktar</div> <div>integer</div>	<div>birim</div> <div>character varying (10)</div>	<div>birimfiyat</div> <div>integer</div>
1	1	124	2	kg	10
2	1	123	13	adet	1
3	2	123	3	adet	2
4	2	125	1	adet	6
5	1	125	1	adet	6
6	3	125	1	adet	7
7	3	123	1	adet	1
8	4	124	1	kg	15
9	4	125	2	adet	100
10	4	123	2	adet	10
11	5	123	2	adet	1

## Fis tablosundaki görünümüm:







	Data Output	Explain	Messages	Notifications
	fisno [PK] integer	tarihsaat timestamp without time zone	toplam integer	tkimlik bigint
1	1	2020-01-01 18:15:00	39	48008062090
2	3	2021-01-02 00:16:57.140897	8	48008062090
3	2	2021-01-01 23:26:39.190508	12	48017060002
4	4	2021-01-03 14:40:00.627484	235	48008062090
5	5	2021-01-03 14:53:30.578761	2	48008062090

## Sonuçlar:

--1999'dan sonraki ekmekleri getirelim. SELECT \* FROM satislar(123, '1999-01-08 04:05:06');

Data Output		Explain	Messages	Notifications	
	fisno integer 	barkod integer 	miktar integer 	birim character varying 	birimfiyat integer 
1	1	123	13	adet	1
2	2	123	3	adet	2
3	3	123	1	adet	1
4	4	123	2	adet	10
5	5	123	2	adet	1

-- 2021-01-02 00:17:00'dan sonraki ekmekler. SELECT \* FROM satislar(123, '2021-01-02 00:17:00');

Data Output		Explain	Messages	Notifications	
	fisno integer 	barkod integer 	miktar integer 	birim character varying 	birimfiyat integer 
1	4	123	2	adet	10
2	5	123	2	adet	1