

ՀՏԴ 681/3

Ինֆորմատիկա

Սվետլանա ԵՐԻՑՅԱՆ

*ԱրՊՀ կիրառական մաթեմատիկայի և ինֆորմատիկայի ամբիոնի դոցենտի
պաշտոնակատար*

E-mail. s.a_ericyan@mail.ru

Անուշ ՀԱՐՈՒԹՅՈՒՆՅԱՆ

ԱրՊՀ կիրառական մաթեմատիկայի և ինֆորմատիկայի ամբիոնի դասախոս

E-mail. asrian.anush@mail.ru

Վազգեն ԱՌՍՏԱՄՅԱՆ

ԱրՊՀ կիրառական մաթեմատիկայի և ինֆորմատիկայի ամբիոնի դոցենտ, մ. գ. թ.

E-mail. varustamyan@rambler.ru

DBMS SQL SERVER–Ի ՈՐՈՇ ԾՐԱԳՐԱՅԻՆ ԿՈՄՊՈՆԵՆՏՆԵՐԻ ՍՏԵՂԾՈՒՄ ԵՎ ԿԻՐԱՌՈՒՄ ՕՐԻՆԱԿԻ ՀԻՄԱՆ ՎՐԱ

Սույն հոդվածում ներկայացված է DBMS SQL Server–ի առավել խոր ուսումնասիրություն պահանջող և համեմատաբար բարդ միջոցների կիրառումը, ինչպիսիք են՝ հիշվող պրոցեդուրա(stored procedure), օգտվողի ֆունկցիա(user defined function), տրիգգեր(trigger) և այլն, որոնք դիտարկված են օրինակի միջոցով:

Բանալի բառեր՝ տրիգգեր, հիշվող պրոցեդուրա, օգտվողի ֆունկցիա, տվյալների հենքի օրինակ, ժամանակավոր աղյուսակ:

***С.ЕРИЦЯН, А.АРУТЮНЯН, В.АРСТАМЯН
СОЗДАНИЕ НЕКОТОРЫХ ПРОГРАММНЫХ
КОМПОНЕНТОВ В СУБД SQL SERVER-Е И ИХ
ПРИМЕНЕНИЕ НА ПРИМЕРЕ***

В этой статье представлены некоторые из более сложных инструментов, которые являются основными средствами СУБД SQL Server и требуют более глубокого изучения, например, хранимые процедуры, пользовательские функции, триггеры и т.д.

Ключевые слова: триггер, хранимая процедура, пользовательская функция, пример базы данных, временная таблица.

***S.YERITSYAN, A.HARUTYUNYAN, V. ARSTAMYAN
CREATION OF SOME SOFTWARE COMPONENTS IN
THE SQL SERVER DBMS AND THEIR APPLICATION
BY EXAMPLE***

This article introduces some of the more sophisticated tools that DBMS SQL Server requires that require more in-depth study, such as memorization procedures, user functionality, and triggers that are illustrated.

Key words: trigger, stored procedure, user defined function, database example, temporary table.

Հոդվածում շեշտը դրված է SQL լեզվի տրիգգեր գաղափարին: Այն հանդիսանում է ծրագրային միջոց առանց որի շատ դժվար է որոշ հարցեր իրականացնել, ինչպես նաև դրանց կիրառումը հիմնավորել: Մեր նպատակն է դիտարկել միայն DML տիպի տրիգգերները:

Տրիգգերը հանդիսանում է հատուկ տիպի պահպանվող պրոցեդուրա, որը կցվում է կոնկրետ աղյուսակին կամ ներկայացմանը և կատարվում է ավտոմատ ձևով, երբ կանչվում և կատարվում են տրիգգերում կցված գործողությունները:

Տրիգգերները լինում են երկու տիպի՝ AFTER կամ FOR (ակտիվանում են աղյուսակի տվյալների մոդիֆիկացիայից հետո) և INSTEAD OF (տվյալների մոդիֆիկացիայից առաջ): INSTEAD OF տրիգգերը ունի մեծ հնարավորություններ և պատկանում է շատ բարդ ծրագրային միջոցներին:

Ի տարբերություն պրոցեդուրաների, տրիգգերները չեն ստանում պարամետրեր և չեն վերադարձնում ավարտման կոդը: Տրիգգերների կոդում կարելի է օգտագործել այնպիսի մեխանիզմ, որը հնարավորություն կտա պարզել, որ տողերի վրա պետք է տարածվի տրիգգերի ազդեցությունը: Մյուս առանձնահատկությունը կայանում է նրանում, որ թույլատրվում է օգտագործել return բանալի բառը, որը հնարավորություն է տալիս կառավարել ավարտման մի քանի դեպքեր:

Պարզագույն օրինակի վրա՝ խոհանոցում տնտեսուհու աշխատանքի հեշտացում ստեղծենք տվյալների հենք, որը հնարավորություն կտա սպասարկել տնտեսուհու աշխատանքում անհրաժեշտ գործողությունները, որտեղ կիրառված են INSTEAD OF և AFTER կամ FOR տիպի տրիգգերներ:

Դիցուք տնտեսուհու առջև դրված են հետևյալ խնդիրները՝

1. եղած մթերքների հաշվառում,
2. ունենալ ճաշատեսակները իրենց բաղադրատոմսերով,
3. տեղեկություններ ստանալ տվյալ ճաշատեսակի համապատասխան չափաբաժնի անհրաժեշտ մթերքների քանակի վերաբերյալ հետևյալ 3 դեպքերում՝

- ճաշատեսակը պատրաստելու համար, եթե մթերքները հերիքում են, կատարվում է համապատասխան մթերքների քանակի նվազեցում,
- ճաշատեսակը պատրաստելու համար, եթե մթերքները հերիքում են, վերադարձնում է նաև անհրաժեշտ մթերքների ցուցակը քանակներով,
- եթե մթերքները չեն հերիքում, վերադարձնում է անհրաժեշտ պակաս մթերքների ցուցակը քանակներով,

4. ստանալ ճաշատեսակի բաղադրատոմսը:

Ստեղծենք xohanoc_db տվյալների հենք, որը պարունակում է հետևյալ անհրաժեշտ աղյուսակները՝ վերը նշված խնդիրները լուծելու համար՝

```
CREATE database xohanoc_db
```

```
GO
```

```
USE xohanoc_db
```

```
GO
```

```
CREATE TABLE tesakner --ճաշատեսակների անուններ;
```

```
(
```

```
    ID INT IDENTITY PRIMARY KEY, --ճաշատեսակի իդենտիֆիկատոր
```

```
    name NVARCHAR(20) --ճաշատեսակի անվանում
```

```
)
```

```
CREATE TABLE mterqner -- ունեցած մթերքների աղյուսակ
```

```
(
```

```
    ID INT IDENTITY PRIMARY KEY, -- մթերքի իդենտիֆիկատոր
```

```
    name NVARCHAR(20), -- մթերքի անվանումը
```

```
    cnt INT --քանակը
```

```
)
```

```
CREATE TABLE receiptner -- ունեցած բաղադրատոմսերի աղյուսակ
```

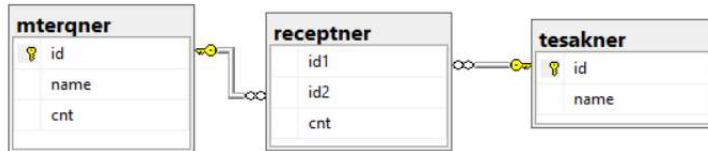
```
(
```

```
    id1 INT FOREIGN KEY REFERENCES tesakner(id) DEFAULT NULL, --ճաշատեսակի  
    իդենտիֆիկատոր
```

id2 **INT FOREIGN KEY REFERENCES** *mterqner(id)*, -- մթերքի իդենտիֆիկատոր
cnt **INT** -- բաղադրատոմսում նշված մթերքի քանակը

)

Ըստ վերը նշվածի տվյալների հենքի՝ աղյուսակները ունեն հետևյալ դիագրամը.



1. Եղած մթերքների հաշվառում: Մուտքագրենք եղած մթերքները *mterqner* աղյուսակում:
2. Մթերքների ավելացման արդյունքում կստանանք անորոշություն. միննույն մթերքը կգրանցվի տարբեր իդենտիֆիկատորների տակ: Այստեղ նպատակահարմար է **INSTEAD OF** տիպի տրիգգեր: Շնորհիվ տրիգգերի՝ *Insert_Trigger*, ցուցակում եղած մթերքներին ավելացվում է նույն անունով մթերքը, այսինքն՝ պետք է կատարվի **UPDATE** օպերատորը, իսկ չեղած մթերքները կցվում են ցուցակի վերջում. կատարվի **INSERT** օպերատորը: Այստեղ օգտագործված է նաև *#name_mterqner* ժամանակավոր աղյուսակ՝ տրիգգերի աշխատանքը արագացնելու համար:

Տրիգգերի կոդը հետևյալն է՝

```

CREATE TRIGGER Insert_Trigger
ON mterqner
INSTEAD OF INSERT
AS
IF @@ROWCOUNT=0
RETURN
SELECT inserted.name
INTO #name_mterqner
FROM INSERTED
JOIN mterqner
ON inserted.name=mterqner.name
IF EXISTS
(SELECT 'TRUE' FROM mterqner
JOIN inserted
ON inserted.name=mterqner.name)
BEGIN
UPDATE mterqner
SET mterqner.cnt=mterqner.cnt+inserted.cnt
FROM mterqner
JOIN inserted
ON mterqner.name=inserted.name
  
```

```

WHERE mterqner.name in
(SELECT name FROM #name_mterqner)
END
INSERT mterqner
SELECT inserted.name, inserted.cnt FROM inserted
WHERE inserted.name NOT IN
(SELECT name FROM #name_mterqner)

```

Կատարենք մթերքների գրանցում:

```

INSERT mterqner
VALUES
(N'ձու',20),
(N'ալյուր',5),
(N'շաքարավազ',4),
(N'յուղ',4)

```

Ստուգենք կատարած գործողությունը:

```

SELECT N'Մթերքներ'
SELECT * FROM mterqner

```

Մթերքներ		
id	name	cnt
1	ձու	20
2	ալյուր	5
3	շաքարավազ	4
4	յուղ	4

Այս դեպքում աշխատեց միայն տրիգգերի **INSERT** դեպքը, քանի որ աղյուսակը դատարկ էր: Ստուգենք **UPDATE** դեպքը, երբ ավելացնում ենք միայն աղյուսակում գոյություն ունեցող մթերքներ:

```

INSERT mterqner
VALUES
(N'ձու',1),
(N'ալյուր',1),
(N'շաքարավազ',1)

```

Ստուգենք կատարած գործողությունը:

```

SELECT N'Մթերքներ'
SELECT * FROM mterqner

```

Ունենիք

Ստացանք

Մթերքներ		
id	name	cnt
1	ծու	20
2	ալյուր	5
3	շաքարավազ	4
4	յուղ	4

Մթերքներ		
id	name	cnt
1	ծու	21
2	ալյուր	6
3	շաքարավազ	5
4	յուղ	4

Ստուգենք 2 դեպքերը միասին՝ եղած մթերքի և նոր մթերքի ավելացում:

INSERT mterqner

VALUES

(N'ծու',1),

(N'ալյուր',1),

(N'շաքարավազ',1),

(N'սոդա',1)

Ստուգենք կատարած գործողությունը.

SELECT N'Մթերքներ'

SELECT * FROM mterqner

Կատանանք՝

Մթերքներ		
id	name	cnt
1	ծու	22
2	ալյուր	7
3	շաքարավազ	6
4	յուղ	4
5	սոդա	1

Նորից ստուգենք միայն **INSERT** դեպքը, երբ մթերքը չկա:

INSERT mterqner

VALUES (N'մեղր',1)

Մթերքներ		
id	name	cnt
1	ծու	19
2	ալյուր	5
3	շաքարավազ	4
4	յուղ	4
5	սոդա	1
6	ընկույզ	0
7	աղ	1
8	վանիլ	2
9	մեղր	1

3. Ճաշատեսակները իրենց բաղադրատոմսերով

Ավելացնենք բաղադրատոմսի բաղադրությունը: Դրա համար օգտագործել ենք #receipt ժամանակավոր աղյուսակ, որի տվյալները մշակելուց հետո պարունակությունը ջնջվում է: Ժամանակավոր աղյուսակի մեջ մտցնենք

անհրաժեշտ բաղադրատոմսի բաղադրիչները: Պրոցեդուրայի ավարտին հեռացվում է պարունակությունը, և կատարվում է նախապատրաստում՝ հաջորդ բաղադրատոմսը ավելացնելու համար:

```
CREATE TABLE #recept      -- ժամանակավոր աղյուսակ նոր բաղադրատոմս ավելացնելիս,
                             -- mterqner աղյուսակում բացակա մթերքի իդենտիֆիկատորի
                             որոշման համար
(
    name NVARCHAR (20),    -- մթերքի անվանումը
    cnt INT,                -- քանակը
    cnt1 INT DEFAULT 0     -- բացակա մթերքների ավելացումը 0 քանակով
)
```

Նոր տեսակի բաղադրատոմս մտցնելիս կրկնությունը բացառելու համար ստեղծենք **AFTER** տիպի տրիգգեր, որը կցենք **INSERT** գործողությանը.

```
CREATE TRIGGER tr_notrepeat_tesak
ON tesakner
AFTER INSERT
IF EXISTS
(SELECT 'true' FROM tesakner
 WHERE inserted.name = tesakner.name)
BEGIN
    RAISERROR (N'Սրդեն կա',16,1)
    ROLLBACK TRAN
END
```

1-ին քայլ. Մտցնենք նոր տեսակ, օրինակ՝ գաթա.

```
INSERT tesakner
VALUES (N'Գաթա')
```

Շնորհիվ տրիգգերի չլինելու դեպքում այն կավելացվի: Շարունակելով պետք է անցնել երկրորդ քայլին, այսինքն ավելացնենք բաղադրատոմսը: Հակառակ դեպքում ստանում ենք <<Արդեն կա>> հաղորդագրությունը, և 2-րդ խնդիրը ավարտվում է:

2-րդ քայլ. Լրացնենք նոր ճաշատեսակի բաղադրատոմսը #recept աղյուսակում.

```
INSERT #recept (name,cnt)
VALUES
(N'ձու',2),
(N'ալյուր',1),
(N'շաքարավազ',1),
(N'ընկույզ',1)
```

Ստուգենք պարունակությունը.

```
SELECT N'Բաղադրատոմս'
SELECT * FROM #recept
```

Բաղադրատոմս		
name	cnt	cnt1
ծու	2	0
այլուր	1	0
շաքարավազ	1	0
ընկույզ	1	0

3-րդ քայլ. Բաղադրատոմս ավելացնելու համար receiptner աղյուսակում ստեղծված է պահվող պրոցոդուրա, որտեղ նախատեսված է այն դեպքը, երբ մթերքների ցուցակում բաղադրատոմսում նշված մթերքը բացակա է, իդենտիֆիկատոր չունի: Դրա համար այն գրանցում ենք մթերքների աղյուսակում 0 քանակով: Այս դեպքի համար, որպեսզի տրիգգերը չաշխատի, այն անջատել ենք:

```

CREATE PROC spAddRecept1
    @name NVARCHAR(20)
AS
    DECLARE @n INT
    SELECT id=@n
    FROM tesaknner
    WHERE name=@name
    IF @@ROWCOUNT=0
BEGIN
        PRINT @name + ' Ճաշատեսակը չի մտցվել, նորից մտցրեք Ճաշատեսակը'
    RETURN
    END

    IF EXISTS
        (SELECT 'true'
         FROM mterqner
         RIGHT OUTER JOIN #recept
             ON #recept.name=mterqner.name
         WHERE mterqner.name IS NULL)
BEGIN
    ALTER TABLE mterqner
    DISABLE TRIGGER ALL
    INSERT mterqner
    SELECT #recept.name, #recept.cnt1 FROM #recept
    WHERE #recept.name NOT IN
        (SELECT mterqner.name FROM mterqner
         JOIN #recept
             ON mterqner.name=#recept.name)
END
INSERT INTO #receptner (receiptner.id2, receiptner.cnt)
    SELECT mterqner.id, #recept.cnt
    FROM mterqner
    JOIN #recept

```

```

ON mterqner.name=#receipt.name
DELETE #receipt
UPDATE receiptner
SET receiptner.id1=@n
WHERE receiptner.id1 is null
ALTER TABLE mterqner
ENABLE TRIGGER ALL
SELECT N'Ավելացրած է ' + CAST (@name AS VARCHAR)

```

Դիտարկված խնդիրը իրականացնելու համար կատարենք պրոցեդուրայի կանչ *Գաթա* ճաշատեսակի համար, առաջին և երկրորդ քայլերի կատարումից հետո,

```
EXEC spAddReceipt N'Գաթա'
```

Ստուգենք՝

```

SELECT N'Տեսակներ'
SELECT * FROM tesakner
SELECT N'Բաղադրատոմսեր'
SELECT * FROM receiptner
SELECT N'Մթերքներ'
SELECT * FROM mterqner

```

Մթերքների աղյուսակում կատարվել է հետևյալ փոփոխությունը՝ բաղադրատոմսի վանիլը մթերքներում անելացված է 0 քանակով.

Ունեինք

Մթերքներ		
id	name	cnt
1	ծուլ	22
2	ալյուր	7
3	շաքարավազ	6
4	յուղ	4
5	սոդա	1

Ստացանք

Մթերքներ		
id	name	cnt
1	ծուլ	22
2	ալյուր	7
3	շաքարա...	6
4	յուղ	4
5	սոդա	1
6	ընկույզ	0

Տեսակներ

id	name
1	գաթա

Բաղադրատոմսեր

id1	id2	cnt
1	1	2
1	2	1
1	3	1
1	6	1

```

INSERT tesakner
VALUES (N'Գաթա')

```

Ստանում ենք <<Արդեն կա >> հաղորդագրությունը:

Ստուգենք.

```

SELECT N'Տեսակներ'
SELECT * FROM tesakner

```

Տեսակներ	
id	name
1	Գաթա

Տեսակը չի ավելացվել :

Նոր ճաշատեսակ ավելացնելու համար՝ օրինակ հաց, կատարում ենք հետևյալ քայլերը.

Քայլ 1 *INSERT* *tesakner*
VALUES (*N'hաց'*)

INSERT *#receipt*
VALUES
(*N'շաքարավազ',1,0*),
(*N'ալյուր',1,0*),
(*N'աղ',1,0*)

Քայլ 2 *EXEC* *spAddReceipt* *N'hաց'*

id	name
1	գաթա
2	հաց

Բաղադրատոմսեր		
id1	id2	cnt
1	1	2
1	2	1
1	3	1
1	6	1
2	2	1
2	3	1
2	7	1

Մթերքներ		
id	name	cnt
1	ծու	22
2	ալյուր	7
3	շաքարավազ	6
4	յուղ	4
5	սոդա	1
6	ընկույզ	0
7	աղ	0

1. Ունեցած մթերքների բավարարվածության ստուգումը տված չափաբաժինով ճաշատեսակ պատրաստելու համար

Ստուգենք հերիքող մթերքների քանակը նշված ճաշատեսակի ու չափաբաժինի համար:

Ստեղծված է օգտագործողի ֆունկցիա՝ `dbo.Resource()`, որը վերադարձնում է աղյուսակ տիպի տվյալներ, չհերիքող քանակով մթերքների ցուցակը, որպեսզի տանտիրուհին գնումներ կատարի՝ ճաշատեսակ պատրաստելու համար.

```
CREATE FUNCTION dbo.Resource(
    @name VARCHAR(20),
    @n INT=1)
RETURNS TABLE
AS
RETURN (SELECT t.name AS tesak, m.name, (@n*r.cnt-m.cnt) AS pakas      --n-
    չափաբաժին
    FROM tesakner AS t
    JOIN receiptner AS r
        ON t.id=r.id1
    JOIN mterqner AS m
        ON m.id=r.id2
    WHERE @n*r.cnt>m.cnt and t.name=@name)
```

Խնդիրը կարելի է իրականացնել 2 եղանակով՝

1-ին եղանակ՝ Իրականացնենք հարցումը՝ կանչելով ֆունկցիան, որի համաձայն՝ պահանջում

ենք Գաթայի 2 բաժնի համար չհերիքող մթերքները.

```
SELECT N'Գաթա 2 բաժին'
```

```
SELECT * FROM dbo.Resource(N'Գաթա',2)
```

Գաթա 2 բաժին

tesak	name	qanak	pakas
Գաթա	վանիլ	0	2

Ունեինք

Մթերքներ

id	name	cnt
1	ծու	20
2	ալյուր	5
3	շաք...	4
4	յուղ	4
5	վանիլ	0

Ստացանք

Բաղադրատոմս

name	cnt
ծու	2
ալյուր	1
շաքարավազ	1
յուղ	1
վանիլ	1

2-րդ եղանակ՝

Ճաշատեսակ պատրաստելու համար ստեղծված է պրոցեդուրա, որը ստուգում է, եթե մթերքները հերիքում են նշված չափաբաժինի համար վերադարձնում է անհրաժեշտ մթերքների ցուցակը անհրաժեշտ քանակով, հակառակ դեպքում վերադարձնում է պակաս մթերքները պահանջված քանակով: *spPrepare* պրոցեդուրան իրականացնում է 3-րդ խնդրի բոլոր 3 կետերը:

N չափաբաժին ճաշատեսակ պատրաստելու համար տնտեսուհին պետք է ստուգի եղած մթերքները: Եթե հերիքում են, ապա համապատասխան քանակով պակասեցվում է, և արտածվում են անհրաժեշտ մթերքները քանակով իր հերթին պակասեցնելով եղած մթերքներից: Հակառակ դեպքում արտածում է չհերիքող ապրանքները.

```
CREATE PROC spPrepare
```

```
  @name NVARCHAR (20),
```

```
  @n INT
```

AS

```
IF NOT EXISTS (SELECT 'true' FROM tesakner
```

```
WHERE name=@name)
```

```
RETURN
```

```
ELSE
```

BEGIN

```
  DECLARE @id INT
```

```
  IF NOT EXISTS
```

```
    (SELECT 'true'
```

```
    FROM dbo.Resource1(@name,@n))
```

BEGIN

```
  SELECT @id=ID FROM tesakner
```

```

WHERE name=@name
UPDATE mterqner
SET mterqner.cnt=mterqner.cnt-@n*receptner.cnt
FROM mterqner
JOIN receptner
ON receptner.id2=mterqner.id
WHERE receptner.id1=@id
SELECT 'patrast e'+@name+cast(@n as NVARCHAR)
SELECT mterqner.name, receptner.cnt*@n as N'Քանակ'
FROM receptner
JOIN mterqner
ON receptner.id2=mterqner.id
WHERE receptner.id1=@id
END
ELSE
BEGIN
SELECT 'pakas mterqner'
SELECT * FROM dbo.Resource1(@name,@n)
END
END

```

Ստուգենք՝

```
EXEC spPrepare N'Գաթա', 2
```

Վերադարձնում է պակաս մթերքները հետևյալ ձևով.

Պակաս մթերքներ				
tesak	name	cnt	qanak	pakas
Գաթա	վանիլ	0	1	2

Ավելացնենք պակաս մթերքները:

```

INSERT mterqner
VALUES
(N'վանիլ',2),
(N'աղ',1)

```

Ստուգենք մթերքներ աղյուսակը ավելացնելուց հետո:

```
SELECT N'Մթերքներ'
```

```
SELECT * FROM mterqner
```

Մթերքներ		
id	name	cnt
1	ծու	23
2	ալյուր	7
3	շաքարավազ	6
4	յուղ	4
5	սոդա	1
6	ընկույզ	2
7	աղ	1
8	վանիլ	2

Շնորհիվ տրիգգերի՝ *Insert_Trigger*, եղած մթերքներին ավելացվել է, իսկ չեղած մթերքները կցվել են:

Արդյունքում ստանում ենք հետևյալ մթերքների աղյուսակը.

Ունեինք

Մթերքներ		
id	name	cnt
1	ծու	20
2	այլուր	5
3	շաքարավազ	4
4	յուղ	4
5	վանիլ	0

Ստացանք

Մթերքներ		
id	name	cnt
1	ծու	20
2	այլուր	6
3	շաքարավազ	5
4	յուղ	5
5	վանիլ	4
13	ընկույզ	2

Անցնենք

ճաշատեսակի

պատրաստման

գործընթացին: Ավելացնելուց հետո կատարենք *sp_Prepare* պրոցեդուրան, որպեսզի այս դեպքում ստանանք անհրաժեշտ մթերքների ցուցակը և պակասեցնենք համապատասխան աղյուսակի մթերքները.

EXEC *spPrepare* N'Գաթա',2

Անհրաժեշտ մթերքներգաթա2	
name	(No column name)
ծու	4
այլուր	2
շաքարավազ	2
ընկույզ	2

Ստացանք

Մթերքներ		
id	name	cnt
1	ծու	19
2	այլուր	5
3	շաքարավազ	4
4	յուղ	4
5	սոդա	1
6	ընկույզ	0
7	աղ	1
8	վանիլ	2

Ունեինք

Մթերքներ		
id	name	cnt
1	ծու	23
2	այլուր	7
3	շաքարավազ	6
4	յուղ	4
5	սոդա	1
6	ընկույզ	2
7	աղ	1
8	վանիլ	2

Օրինակ՝ ունեցել ենք 23 ձու, ստացանք՝ 19 ձու և այլն:

4. Ստանալ ճաշատեսակի բաղադրատոմսը:

Ձևակերպենք պրոցեդուրա՝ ճաշատեսակի բաղադրատոմսը արտաձելու համար.

CREATE PROC *sp_recept*
 @name **NVARCHAR** (20)

AS

IF NOT EXISTS

(**SELECT** 'true'

FROM *tesakner*

WHERE @name=name)

BEGIN

SELECT @name + N'-Այդպիսի ճաշատեսակ չկա '

RETURN

```

END
ELSE
BEGIN
    SELECT @name
    SELECT m.name, r.cnt
    FROM tesakner AS t
    JOIN receiptner AS r
    ON t.id=r.id1
    JOIN mterqner AS m
    ON r.id2=m.id
    WHERE t.name=@name
END

```

Արտաձենք գաթայի բաղադրատոմսը.

```
EXEC sp_recept3 N'գաթա'
```

Results		Messages	
		(No column name)	
1	գաթա		
	name	cnt	
1	ծու	2	
2	ալյուր	1	
3	շաքարավազ	1	
4	ծեթ	1	
5	կաթ	1	

Եվ այսպես տնտեսուհին վերը նշված խնդիրները իրականացնելու համար կկատարի հետևյալը.

1. Խնդիր՝

Եթե ունենք մթերքներ, ապա մտցնենք, չլինելու դեպքում, եթե ունենք նոր տեսակ կարող ենք անցնել 2-րդ խնդրին.

```

INSERT mterqner
VALUES
    (N'ծու',20),
    (N'ալյուր',5),
    (N'շաքարավազ',4),
    (N'յուղ',4)

```

2. Խնդիր՝

1-ին քայլ՝

```

INSERT tesakner
VALUES (N'Գաթա')

```

Եթե ստանում ենք հաղորդագրություն «Արդեն կա» կատարում ենք անհրաժեշտ հաջորդ խնդիրը, հակառակ դեպքում լրացնում ենք բաղադրատոմսը.

2-րդ քայլ՝

```
INSERT #receipt
(name,cnt)
VALUES
(N'ձու', 2),
(N'ալյուր', 1),
(N'շաքարավազ', 1),
(N'ընկույզ', 1)
```

3-րդ քայլ՝

```
EXEC spAddReceipt N'Գաթա'
```

3. Խնդիր՝

```
EXEC spPrepare N'Գաթա', 2
```

Վերադարձնում է պակաս մթերքները քանակով: Ավելացնելու համար պետք է անցնել 1-ին խնդրի կատարմանը, հակառակ դեպքում վերադարձվում է անհրաժեշտ մթերքների ցուցակը քանակով, և համապատասխան մթերքների քանակը նվազեցվում է մթերքների աղյուկում.

4. Խնդիր՝

```
EXEC sp_receipt3 N'գաթա'
```

Արտածում է պահանջած տեսակի բաղադրատոմսը:

Գրականություն

1. ВиейраР., Программирование баз данных Microsoft SQL Server 2005. Вильямс, 2007.- 832 с.
2. Ульман Д., Уидом Д., Введение в системы баз данных,. Учебное пособие-М.: Вильямс, 2000. -384 с.
3. Гарсиа-Молина Г., Ульман Д., Уидом Д., Системы баз данных. Полный курс, 2003.- 379 с.
4. Дейт К., Введение в системы баз данных, 2001. Издательский дом "Вильямс", 2001.- 1072 с.

Հոդվածը տպագրության է երաշխավորել խմբագրական խորհրդի անդամ, ֆ.մ.գ.թ. Գ. Հ.Սահակյանը: