/\*

Saxlanilan prosedurlar - SP (Stored procedures - SP)

Saxlanilan prosedur - T-SQL komandalarinin adıqoyilmuş blokudur.Blok giriş və çıxış parametrlərinə malik ola bilər

Saxlanılan prosedur həmişə səhvin kodu olan rəqəmi qaytarir.Əgər prosedur uğurla yerinə yetirilibsə, onda sıfır qayıdır, əgər uğursuz yerinə yetirilibsə onda səhvin kodu qayıdır.

Saxlanılan prosedurlar hansısa əməliyyatlarin yerine yetirilmesi üçün nəzərdə tutulub : məlumatlarda dəyişikliklərin edilməsi, BD arxitekturası ilə birgə əməkdaşlıq

SP ilə nə etmək olar :

+ onları EXECUTE KOMANDASİNİN KOMEYİLE T-SQL KODUNDAN ÇAĞIRMAQ OLAR

;onlara məlumatları giriş parametrlərinin vasitəsilə ötürmək və onlardan məlumatları çıxış parametrləri vasitəsilə geri almaq olar

+ onlar nəticələndirən sorğular qrupunu kliyent proqramına qaytara bilirlər

+ onlar cədvəldəki məlumatlarda dəyişikliklər edə bilirlər

+ оnlar cədvəl və indeks yarada,dəyişdirə və ləğv edə bilirlər

SP-den istifade etmeyin ustunlukleri

+ T-SQL kodunun inkapsulyasiyasi

+ melumatlar bazasinin daha da tehlukesiz olmasinin temin edilmesi

+ Istifadecilere ve proqramlara melumatlardan istifade ucun daha universal girish seviyyesinin tegdim edilmesi

+yerine yetirilme planlarinin yaradilmasi vasitesile mehsuldarligin yukseldilmesi.bu planlar tekraren istifade oluna biler

(\* подробнее стр. 449)

:Demek olar ki, BUTUN T-SQL INSTRUKSIYALARI SAXLANILAN PROSEDURA DAXIL EDILE BILER

+ USE <database name>;komandasini istifade etmek olmaz

+ Create instruksiyasini bu tip komandalarla istifade etmek olmaz: AGGREGATE, RULE, DEFAULT,

CREATE, FUNCTION, TRIGGER, PROCEDURE ya VIEW;

+ CREATE, ALTER ve DROP instruksiyalarinin komeyi ile cedvel ve indeksi yaratmaq, deyishmek ve leğv etmek olar

SP-nin elan edilməsinin sintaksisi

CREATE POCEDURE prosedurun adı

[ parametr 1 AS məlumatlar tipi [OUTPUT], ... ]

AS

BEGIN

prosedurun cismi

END;

SP-nin çağırılması Sintaksisi

EXEC prosedurun adı

[ parametr 1 AS məlumatlar tipi [OUTPUT], ... ];

OUTPUT AÇAR SÖZÜ XÜSUSİ PARAMETRİ TƏYİN EDİR

BU PARAMETR ÇAĞIRAN TƏRƏFƏ ANLAMLARI QAYTARIR

\*/

USE [Library];

GO

-- хранимая процедура возвращает взятые книги, которые еще не вернули saxlanilan prosedur goturulmush, hele qaytarilmamış kitabları qaytarir

CREATE PROCEDURE TakenBooks

AS

BEGIN

SELECT S.FirstName, S.LastName, B.[Name]

FROM Students AS S INNER JOIN S\_Cards AS SC

ON S.Id = SC.Id\_Student

INNER JOIN Books AS B

ON B.Id = SC.Id\_Book

WHERE SC.DateIn IS NULL;

END;

GO

-- saxlananin çağırılması

EXEC TakenBooks;

GO

-- изменяем ХП так, чтобы можно было указать группу в качестве входного параметра SP-ni elə dəyişdiririk ki, qrupu giriş parametri kimi göstərmək mümkün olsun

ALTER PROCEDURE TakenBooks

@group AS nvarchar(50)

AS

BEGIN

SELECT S.FirstName, S.LastName, G.[Name] AS GroupName, B.[Name] AS BookName

FROM Students AS S INNER JOIN S\_Cards AS SC

ON S.Id = SC.Id\_Student

INNER JOIN Books AS B

ON B.Id = SC.Id\_Book

INNER JOIN Groups AS G

ON G.Id = S.Id\_Group

WHERE SC.DateIn IS NULL AND G.[Name] = @group;

END;

GO

-- saxlanilan prosedurun çağırılması və giriş parametrlərinin ötürülməsi

DECLARE @grname AS nvarchar(50);

SET @grname = N'18П2';

EXEC TakenBooks @grname;

GO

-- SP-ni elə dəyişdiririk ki, kitabları qaytarmayan tələbələrin sayini bilmək mümkün olsun

ALTER PROCEDURE TakenBooks

@group\_name AS nvarchar(50),

@count\_studs AS int OUTPUT

AS

BEGIN

SELECT S.FirstName, S.LastName, G.[Name] AS GroupName, B.[Name] AS BookName

FROM Students AS S INNER JOIN S\_Cards AS SC

ON S.Id = SC.Id\_Student

INNER JOIN Books AS B

ON B.Id = SC.Id\_Book

INNER JOIN Groups AS G

ON G.Id = S.Id\_Group

WHERE SC.DateIn IS NULL AND G.[Name] = @group\_name;

SET @count\_studs = @@ROWCOUNT;

END;

GO

-- saxlanilan prosedurun çağırılması və giriş parametrlərinin ötürülməsi

DECLARE @grname AS nvarchar(50);

SET @grname = N'18П2';

DECLARE @stcount AS int;

EXEC TakenBooks @grname, @stcount OUTPUT;

SELECT @stcount AS StudsCount;

GO

-- Prosedur kitabların qaytarilmasi üçün tələbəyə zəng etməyin lazim olub-olmamağını təyin edir

CREATE PROCEDURE NeedCall

@stud\_id AS int

AS

BEGIN

DECLARE @books AS int = 0;

SELECT @books = COUNT(\*)

FROM S\_Cards

WHERE S\_Cards.Id\_Student = @stud\_id;

IF @books > 0

BEGIN

PRINT 'Call!';

PRINT 'This student ('

+ CAST(@stud\_id AS varchar(5))

+ ') should return '

+ CAST(@books AS nvarchar(5))

+ ' books.';

END

END

GO

-- saxlanilan prosedurun çağırılması

EXEC NeedCall 2;

EXEC NeedCall 3;

EXEC NeedCall 4;

EXEC NeedCall 5;

EXEC NeedCall 6;

EXEC NeedCall 7;