

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра САПР

ОТЧЕТ
по лабораторной работе № 8
по дисциплине "Базы данных"
Тема: "Создание и использование хранимых процедур"

Студент гр. 2311

Завьялов Н.С.

Князев И.М.

Преподаватель

Новакова Н.Е.

Санкт-Петербург

2024

Цель работы: научиться писать и применять хранимые процедуры [лит.].

Используемая база данных (БД): AdventureWorks.

Упражнение 1 – создание хранимой процедуры без параметров

Запрос 1: создание хранимой процедуры GetDiscounts в схеме Sales, которая извлекает поля Description, DiscountPct, Type, Category, StartDate, EndDate, MinQty и MaxQty из таблицы Sales.SpecialOffer, а также сортирует записи по полям StartDate и EndDate по возрастанию.

```
USE AdventureWorks;
```

```
GO
```

```
CREATE PROCEDURE Sales.GetDiscounts AS
```

```
SELECT
```

```
    Description,
```

```
    DiscountPct,
```

```
    Type,
```

```
    Category,
```

```
    StartDate,
```

```
    EndDate,
```

```
    MinQty,
```

```
    MaxQty
```

```
FROM
```

```
    Sales.SpecialOffer
```

```
ORDER BY
```

```
    StartDate,
```

```
    EndDate;
```

```
GO
```

Результат выполнения:

Commands completed successfully.

Completion time: 2024-10-23T21:01:11.4587909+03:00

Запрос 2: выполнение процедуры Sales.GetDiscounts.

```
EXEC Sales.GetDiscounts;
```

Результат выполнения:

Description	Type
DiscountPct	
Category	
StartDate	EndDate
MaxQty	MinQty

No	Discount
0,00	No Discount
No	Discount
2001-06-01	00:00:00.000
2004-12-31	00:00:00.000
NULL	0

```

...
Mountain-500      Silver      Clearance      Sale
0,40              Discontinued  Product
Reseller
2004-05-01      00:00:00.000    2004-06-30      00:00:00.000    0
NULL

```

(16 rows affected)

Completion time: 2024-10-23T21:04:37.8303217+03:00

Упражнение 2 – создание хранимой процедуры с параметром

Запрос 1: создание хранимой процедуры GetDiscountsForCategory в схеме Sales с параметром @Category типа nvarchar(50), которая извлекает поля Description, DiscountPct, Type, Category, StartDate, EndDate, MinQty и MaxQty из таблицы Sales.SpecialOffer при условии, что поле Category равно @Category, а также сортирует записи по полям StartDate и EndDate по возрастанию.

```
USE AdventureWorks;
```

```
GO
```

```
CREATE PROCEDURE Sales.GetDiscountsForCategory @Category
nvarchar(50) AS
```

```
SELECT Description,
        DiscountPct,
        TYPE,
        Category,
        StartDate,
```

```

        EndDate,
        MinQty,
        MaxQty
FROM Sales.SpecialOffer
WHERE Category = @Category
ORDER BY StartDate,
        EndDate;
GO

```

Результат выполнения:

Commands completed successfully.

Completion time: 2024-10-23T21:08:19.9266021+03:00

Запрос 2: выполнение процедуры Sales.GetDiscountsForCategory с параметром 'Reseller'.

```
EXEC Sales.GetDiscountsForCategory 'Reseller';
```

Результат выполнения:

Description	DiscountPct	Category	StartDate	EndDate	MinQty	MaxQty	TYPE

Упражнение 3 – создание хранимой процедуры с параметрами и значениями по умолчанию

Запрос 1: создание хранимой процедуры GetDiscountsForCategoryAndDate в схеме Sales с параметрами @Category типа nvarchar(50) и @DateToCheck типа datetime со значением по умолчанию NULL. Процедура устанавливает значение параметра @DateToCheck в GETDATE(), если оно равно NULL и извлекает поля Description, DiscountPct, Type, Category, StartDate, EndDate, MinQty и MaxQty из таблицы Sales.SpecialOffer при условии, что поле Category равно @Category и @DateToCheck находится в пределах между StartDate и EndDate, а также сортирует записи по полям StartDate и EndDate по возрастанию.

```
USE AdventureWorks;
```

```
GO
```

```
CREATE PROCEDURE Sales.GetDiscountsForCategoryAndDate  
@Category nvarchar(50),
```

```
@DateToCheck datetime = NULL AS BEGIN IF @DateToCheck IS NULL  
SET @DateToCheck = GETDATE();
```

```
SELECT Description,
```

```
DiscountPct,
```

```
TYPE,
```

```
Category,
```

```
StartDate,
```

```
EndDate,
```

```
MinQty,
```

```
MaxQty
```

```
FROM Sales.SpecialOffer
```

```

WHERE Category = @Category
      AND @DateToCheck BETWEEN StartDate AND EndDate
ORDER BY StartDate,
          EndDate;

END;

GO

```

Результат выполнения:

Commands completed successfully.

Completion time: 2024-10-23T21:14:50.1269560+03:00

Запрос 2: выполнение процедуры Sales.GetDiscountsForCategoryAndDate с параметром 'Reseller'.

```
EXEC Sales.GetDiscountsForCategoryAndDate 'Reseller';
```

Результат выполнения:

Description			
DiscountPct			TYPE
Category			
StartDate	EndDate		MinQty
MaxQty			

(0 rows affected)

Completion time: 2024-10-23T21:17:05.7604936+03:00

Запрос 3: выполнение процедуры Sales.GetDiscountsForCategoryAnd с параметром 'Reseller' и @DateToCheck, который установлен в значение текущей даты минус 21 год.

```
DECLARE @DateToCheck datetime
SET
    @DateToCheck = DATEADD(
        YEAR,
        -21,
        GETDATE()
    ) EXEC Sales.GetDiscountsForCategoryAndDate 'Reseller',
    @DateToCheck;
```

Результат выполнения:

Description				
DiscountPct				TYPE
Category				
StartDate		EndDate		MinQty
MaxQty				

Volume	Discount	11	to	14
0,02			Volume	Discount
Reseller				
2001-07-01	00:00:00.000	2004-06-30	00:00:00.000	11
14				

```

...
Volume                Discount                over                60
0,20                                Volume Discount
Reseller
2001-07-01    00:00:00.000    2004-06-30    00:00:00.000    61
NULL

```

(5 rows affected)

Completion time: 2024-10-23T21:19:30.4451169+03:00

Упражнение 4 – создание хранимой процедуры с входными и выходными параметрами

Запрос 1: создание хранимой процедуры AddDiscount в схеме Sales с параметрами @Description типа nvarchar(50), @DiscountPct типа smallmoney, @Type типа nvarchar(50), @Category типа nvarchar(50), @StartDate типа Datetime, @EndDate типа Datetime, @MinQty типа int, @MaxQty типа int и @NewProductID типа int как выходное. Процедура вставляет все указанные, кроме @NewProductID, параметры в таблицу Sales.SpecialOffer, устанавливает значение параметра @NewProductID в значение SCOPE_IDENTITY() и возвращает значение 0 в случае, если при вставке не произошло ошибок. При возникновении ошибки, происходит вставка значений GETDATE(), USER_NAME(), ERROR_NUMBER(), ERROR_SEVERITY(), ERROR_STATE(), ERROR_PROCEDURE(), ERROR_LINE(), ERROR_MESSAGE() в поля ErrorTime, UserName, ErrorNumber, ErrorSeverity, ErrorState, ErrorProcedure, ErrorLine, ErrorMessage и возвращает значение -1.

USE AdventureWorks;

GO

```

CREATE PROCEDURE Sales.AddDiscount @Description
nvarchar(255),
@DiscountPct smallmoney,
@Type nvarchar(50),
@Category nvarchar(50),
@StartDate Datetime,
@EndDate Datetime,
@MinQty int,
@MaxQty int,
@NewProductID int OUTPUT AS BEGIN BEGIN TRY INSERT INTO
Sales.SpecialOffer (
    Description, DiscountPct, Type, Category,
    StartDate, EndDate, MinQty, MaxQty
)
VALUES
(
    @Description, @DiscountPct, @Type,
    @Category, @StartDate, @EndDate,
    @MinQty, @MaxQty
);
SET
    @NewProductID = SCOPE_IDENTITY();
RETURN 0;
END TRY BEGIN CATCH INSERT INTO dbo.ErrorLog (
    ErrorTime, UserName, ErrorNumber,
    ErrorSeverity, ErrorState, ErrorProcedure,
    ErrorLine, ErrorMessage
)
VALUES
(
    GETDATE(),
    USER_NAME(),

```

```

        ERROR_NUMBER(),
        ERROR_SEVERITY(),
        ERROR_STATE(),
        ERROR_PROCEDURE(),
        ERROR_LINE(),
        ERROR_MESSAGE()
    );
RETURN -1;
END CATCH;
END;
GO

```

Результат выполнения:

Commands completed successfully.

Completion time: 2024-10-23T21:22:52.6383926+03:00

Запрос 2: определение переменных @StartDate типа datetime, @EndDate типа datetime, присваивание первой значение возвращаемое GETDATE(), второе значение возвращаемое DATEADD(MONTH, 1, @StartDate), далее определение переменной @NewID типа int и выполнение процедуры Sales.AddDiscount с параметрами 'Half price off everything', 0.5, 'Seasonal Discount', 'Customer', @StartDate, @EndDate, 0, 20, @NewID как возвращаемое и вывод значения выходного параметра @NewID.

```

DECLARE @StartDate datetime, @EndDate datetime
SET @StartDate = GETDATE()
SET @EndDate = DATEADD(MONTH, 1, @StartDate)
DECLARE @NewID int
EXEC Sales.AddDiscount 'Half price off everything', 0.5,
'Seasonal Discount', 'Customer', @StartDate, @EndDate, 0, 20,
@NewID OUTPUT

```

```
SELECT @NewID;
```

Результат выполнения:

```
(1 row affected)
```

```
-----
```

```
17
```

```
(1 row affected)
```

```
Completion time: 2024-10-23T21:24:06.0490869+03:00
```

Запрос 3: определение переменных @StartDate типа datetime, @EndDate типа datetime, присваивание первой значение возвращаемое GETDATE(), второе значение возвращаемое DATEADD(MONTH, 1, @StartDate), далее определение переменной @NewID типа int, @ReturnValue типа int и выполнение процедуры Sales.AddDiscount с параметрами 'Half price off everything', -0.5, 'Seasonal Discount', 'Customer', @StartDate, @EndDate, 0, 20, @NewID как возвращаемое и вывод значения выходного параметра @NewID в случае если возвращаемое значение процедуры равно 0, в противном случае вывод первой записи из таблицы dbo.ErrorLog, отсортированной по полю ErrorTime по убыванию.

```
DECLARE @StartDate datetime, @EndDate datetime
SET @StartDate = GETDATE()
SET @EndDate = DATEADD(MONTH, 1, @StartDate)
DECLARE @NewID int, @ReturnValue int
EXEC @ReturnValue = Sales.AddDiscount 'Half price off
everything', -0.5, 'Seasonal Discount', 'Customer',
@StartDate, @EndDate, 0, 20, @NewID OUTPUT
IF(@ReturnValue = 0) SELECT @NewID;
ELSE
SELECT TOP 1 * FROM dbo.ErrorLog ORDER BY ErrorTime DESC;
```

Результат выполнения:

(0 rows affected)

(1 row affected)

ErrorLogID	ErrorTime	UserName	
ErrorNumber	ErrorSeverity	ErrorState	ErrorProcedure
ErrorLine	ErrorMessage		
1	2024-10-23 21:31:17.357	dbo	
547	16	0	Sales.AddDiscount
9	Конфликт инструкции INSERT с ограничением CHECK "CK_SpecialOffer_DiscountPct". Конфликт произошел в базе данных "AdventureWorks", таблица "Sales.SpecialOffer", column 'DiscountPct'.		

(1 row affected)

Вывод

В данной лабораторной работе были получены навыки написания и применения хранимых процедур.

Хранимые процедуры упрощают доступ к данным, обеспечивают их безопасность и повышают производительность. Процедуры могут быть написаны с использованием одного или нескольких параметров. Параметры могут иметь значения по умолчанию или быть выходными.

Хранимая процедура – это группа из одного или нескольких операторов Transact-SQL, которые выполняются как единое целое. Хранимые процедуры обеспечивают следующее:

- обрабатывают входные параметры и возвращают вызывающей программе значения в виде выходных параметров;
- содержат программные инструкции, которые выполняют операции в базе данных, включая вызов других процедур;
- возвращают значение состояния вызывающей программе, таким образом передавая сведения об успешном или неуспешном завершении (и причины последнего).

Хранимая процедура создается с помощью команды CREATE PROC (CREATE PROCEDURE). Хранимая процедура выполняется с помощью команды EXEC.

Параметры хранимой процедуры используются для обмена данными между хранимой процедурой и вызвавшей ее программой. Для добавления в хранимую процедуру входных параметров при создании процедуры указывается имя параметра, его тип данных, при наличии указывается стандартное значение по умолчанию.