

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

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

Студенты гр. 2308

Рымарь М.И.

Мелихов М.А.

Придчин В.Е.

Преподаватель

Горяинов С.В.

Санкт-Петербург

2024

Цель работы

Научиться создавать и применять хранимые процедуры. В лабораторной работе используется БД AdventureWorks.

Выполнение работы

Упражнение 1 – создание хранимой процедуры без параметров.

Скрипт для создания хранимой процедуры:

```
CREATE PROCEDURE [Sales].[GetDiscounts] AS  
BEGIN  
    SELECT Description, DiscountPct, Category, StartDate,  
        EndDate, MinQty, MaxQty  
    FROM Sales.SpecialOffer  
    ORDER BY StartDate, EndDate ASC  
END
```

Результат выполнения показан на рисунке 1.

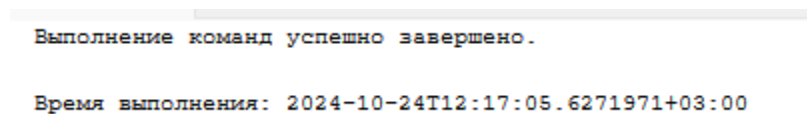


Рисунок 1 – Результат выполнения команд

Проверка хранимой процедуры с помощью команды `EXEC Sales.GetDiscounts`.

Результат показан на рисунке 2.

	Description	DiscountPct	Category	StartDate	EndDate	MinQty	MaxQty
1	No Discount	0,00	No Discount	2001-06-01 00:00:00.000	2004-12-31 00:00:00.000	0	NULL
2	Volume Discount 11 to 14	0,02	Reseller	2001-07-01 00:00:00.000	2004-06-30 00:00:00.000	11	14
3	Volume Discount 15 to 24	0,05	Reseller	2001-07-01 00:00:00.000	2004-06-30 00:00:00.000	15	24
4	Volume Discount 25 to 40	0,10	Reseller	2001-07-01 00:00:00.000	2004-06-30 00:00:00.000	25	40
5	Volume Discount 41 to 60	0,15	Reseller	2001-07-01 00:00:00.000	2004-06-30 00:00:00.000	41	60
6	Volume Discount over 60	0,20	Reseller	2001-07-01 00:00:00.000	2004-06-30 00:00:00.000	61	NULL
7	Mountain-100 Clearance Sale	0,35	Reseller	2002-05-15 00:00:00.000	2002-06-30 00:00:00.000	0	NULL
8	Sport Helmet Discount-2002	0,10	Reseller	2002-07-01 00:00:00.000	2002-07-31 00:00:00.000	0	NULL
9	Road-650 Overstock	0,30	Reseller	2002-07-01 00:00:00.000	2002-08-31 00:00:00.000	0	NULL
10	Mountain Tire Sale	0,50	Customer	2003-06-15 00:00:00.000	2003-08-30 00:00:00.000	0	NULL
11	Sport Helmet Discount-2003	0,15	Reseller	2003-07-01 00:00:00.000	2003-07-31 00:00:00.000	0	NULL
12	LL Road Frame Sale	0,35	Reseller	2003-07-01 00:00:00.000	2003-08-15 00:00:00.000	0	NULL
13	Touring-3000 Promotion	0,15	Reseller	2003-07-01 00:00:00.000	2003-09-30 00:00:00.000	0	NULL
14	Touring-1000 Promotion	0,20	Reseller	2003-07-01 00:00:00.000	2003-09-30 00:00:00.000	0	NULL

(затронута строк: 16)

Время выполнения: 2024-10-24T12:21:39.0087721+03:00

Рисунок 2 – Проверка хранимой процедуры

Упражнение 2 – создание хранимой процедуры с параметром.

Скрипт для создания хранимой процедуры с параметром:

```
CREATE PROCEDURE [Sales].[GetDiscountsForCategory]
@Category AS NVARCHAR(50) AS
BEGIN
    SELECT Description, DiscountPct, Category, StartDate,
EndDate, MinQty, MaxQty
    FROM Sales.SpecialOffer
    WHERE Category = @Category
    ORDER BY StartDate, EndDate ASC
END
```

Результат представлен на рисунке 3.

Выполнение команд успешно завершено.

Время выполнения: 2024-10-24T12:41:32.2841531+03:00

Рисунок 3 – Результат выполнения команд

Проверка хранимой процедуры с параметром с помощью команды `EXEC Sales.GetDiscountsForCategory 'Reseller'`.

Результат показан на рисунке 4.

	Description	DiscountPct	Category	StartDate	EndDate	MinQty	MaxQty
1	Volume Discount 11 to 14	0,02	Reseller	2001-07-01 00:00:00.000	2004-06-30 00:00:00.000	11	14
2	Volume Discount 15 to 24	0,05	Reseller	2001-07-01 00:00:00.000	2004-06-30 00:00:00.000	15	24
3	Volume Discount 25 to 40	0,10	Reseller	2001-07-01 00:00:00.000	2004-06-30 00:00:00.000	25	40
4	Volume Discount 41 to 60	0,15	Reseller	2001-07-01 00:00:00.000	2004-06-30 00:00:00.000	41	60
5	Volume Discount over 60	0,20	Reseller	2001-07-01 00:00:00.000	2004-06-30 00:00:00.000	61	NULL
6	Mountain-100 Clearance Sale	0,35	Reseller	2002-05-15 00:00:00.000	2002-06-30 00:00:00.000	0	NULL
7	Sport Helmet Discount-2002	0,10	Reseller	2002-07-01 00:00:00.000	2002-07-31 00:00:00.000	0	NULL
8	Road-650 Overstock	0,30	Reseller	2002-07-01 00:00:00.000	2002-08-31 00:00:00.000	0	NULL
9	Sport Helmet Discount-2003	0,15	Reseller	2003-07-01 00:00:00.000	2003-07-31 00:00:00.000	0	NULL
10	LL Road Frame Sale	0,35	Reseller	2003-07-01 00:00:00.000	2003-08-15 00:00:00.000	0	NULL
11	Touring-3000 Promotion	0,15	Reseller	2003-07-01 00:00:00.000	2003-09-30 00:00:00.000	0	NULL
12	Touring-1000 Promotion	0,20	Reseller	2003-07-01 00:00:00.000	2003-09-30 00:00:00.000	0	NULL
13	Mountain-500 Silver Clearance Sale	0,40	Reseller	2004-05-01 00:00:00.000	2004-06-30 00:00:00.000	0	NULL

(затронуто строк: 13)

Время выполнения: 2024-10-24T12:45:22.0094074+03:00

Рисунок 4 – Результат проверки хранимой процедуры с параметром

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

Скрипт для создания хранимой процедуры с параметрами и значениями по умолчанию:

```
CREATE PROCEDURE
[Sales].[GetDiscountsForCategoryAndDate]
    @Category AS NVARCHAR(50),
    @DateToCheck AS DATETIME = NULL AS
BEGIN
    IF (@DateToCheck IS NULL) SET @DateToCheck =
GETDATE()
    SELECT Description, DiscountPct, Category, StartDate,
EndDate, MinQty, MaxQty
    FROM Sales.SpecialOffer
    WHERE Category = @Category AND StartDate <=
@DateToCheck
    ORDER BY StartDate, EndDate ASC
END
```

Результат представлен на рисунке 5.

Выполнение команд успешно завершено.

Время выполнения: 2024-10-24T12:58:55.8022689+03:00

Рисунок 5 – Результат выполнения команд

Проверка хранимой процедуры с помощью команды `EXEC Sales.GetDiscountsForCategoryAndDate 'Reseller'`.

Результат выполнения команд представлен на рисунке 6.

	Description	DiscountPct	Category	StartDate	EndDate	MinQty	MaxQty
1	Volume Discount 11 to 14	0,02	Reseller	2001-07-01 00:00:00.000	2004-06-30 00:00:00.000	11	14
2	Volume Discount 15 to 24	0,05	Reseller	2001-07-01 00:00:00.000	2004-06-30 00:00:00.000	15	24
3	Volume Discount 25 to 40	0,10	Reseller	2001-07-01 00:00:00.000	2004-06-30 00:00:00.000	25	40
4	Volume Discount 41 to 60	0,15	Reseller	2001-07-01 00:00:00.000	2004-06-30 00:00:00.000	41	60
5	Volume Discount over 60	0,20	Reseller	2001-07-01 00:00:00.000	2004-06-30 00:00:00.000	61	NULL
6	Mountain-100 Clearance Sale	0,35	Reseller	2002-05-15 00:00:00.000	2002-06-30 00:00:00.000	0	NULL
7	Sport Helmet Discount-2002	0,10	Reseller	2002-07-01 00:00:00.000	2002-07-31 00:00:00.000	0	NULL
8	Road-650 Overstock	0,30	Reseller	2002-07-01 00:00:00.000	2002-08-31 00:00:00.000	0	NULL
9	Sport Helmet Discount-2003	0,15	Reseller	2003-07-01 00:00:00.000	2003-07-31 00:00:00.000	0	NULL
10	LL Road Frame Sale	0,35	Reseller	2003-07-01 00:00:00.000	2003-08-15 00:00:00.000	0	NULL
11	Touring-3000 Promotion	0,15	Reseller	2003-07-01 00:00:00.000	2003-09-30 00:00:00.000	0	NULL
12	Touring-1000 Promotion	0,20	Reseller	2003-07-01 00:00:00.000	2003-09-30 00:00:00.000	0	NULL
13	Mountain-500 Silver Clearance Sale	0,40	Reseller	2004-05-01 00:00:00.000	2004-06-30 00:00:00.000	0	NULL

(затронуто строк: 13)

Время выполнения: 2024-10-24T13:03:41.2078784+03:00

Рисунок 6 – Результат проверки хранимой процедуры

Проверка хранимой процедуры, когда указываются оба параметра, с помощью следующего скрипта:

```
DECLARE @DateToCheck datetime
SET @DateToCheck = DATEADD(MONTH, 1, GETDATE())
EXEC Sales.GetDiscountsForCategoryAndDate 'Reseller',
@DateToCheck
```

Результат выполнения представлен на рисунке 7.

	Description	DiscountPct	Category	StartDate	EndDate	MinQty	MaxQty
1	Volume Discount 11 to 14	0.02	Reseller	2001-07-01 00:00:00.000	2004-06-30 00:00:00.000	11	14
2	Volume Discount 15 to 24	0.05	Reseller	2001-07-01 00:00:00.000	2004-06-30 00:00:00.000	15	24
3	Volume Discount 25 to 40	0.10	Reseller	2001-07-01 00:00:00.000	2004-06-30 00:00:00.000	25	40
4	Volume Discount 41 to 60	0.15	Reseller	2001-07-01 00:00:00.000	2004-06-30 00:00:00.000	41	60
5	Volume Discount over 60	0.20	Reseller	2001-07-01 00:00:00.000	2004-06-30 00:00:00.000	61	NULL
6	Mountain-100 Clearance Sale	0.35	Reseller	2002-05-15 00:00:00.000	2002-06-30 00:00:00.000	0	NULL
7	Sport Helmet Discount-2002	0.10	Reseller	2002-07-01 00:00:00.000	2002-07-31 00:00:00.000	0	NULL
8	Road-650 Overstock	0.30	Reseller	2002-07-01 00:00:00.000	2002-08-31 00:00:00.000	0	NULL
9	Sport Helmet Discount-2003	0.15	Reseller	2003-07-01 00:00:00.000	2003-07-31 00:00:00.000	0	NULL
10	LL Road Frame Sale	0.35	Reseller	2003-07-01 00:00:00.000	2003-08-15 00:00:00.000	0	NULL
11	Touring-3000 Promotion	0.15	Reseller	2003-07-01 00:00:00.000	2003-09-30 00:00:00.000	0	NULL
12	Touring-1000 Promotion	0.20	Reseller	2003-07-01 00:00:00.000	2003-09-30 00:00:00.000	0	NULL
13	Mountain-500 Silver Clearance Sale	0.40	Reseller	2004-05-01 00:00:00.000	2004-06-30 00:00:00.000	0	NULL

(затронуто строк: 13)

Время выполнения: 2024-10-24T13:09:02.4791962+03:00

1

Рисунок 7 – Результат проверки хранимой процедуры

Упражнение 4 – создание хранимой процедуры с входными и выходными параметрами.

Скрипт для создания хранимой процедуры с входными и выходными параметрами:

```
CREATE PROC [Sales].[AddDiscount]
    @Description AS NVARCHAR(255),
    @DiscountPt AS SMALLMONEY,
    @Type AS NVARCHAR(50),
    @Category AS NVARCHAR(50),
    @StartDate AS datetime,
    @EndDate AS datetime,
    @MinQty AS INT,
    @MaxQty AS INT,
    @NewProductID AS INT OUTPUT AS
BEGIN

BEGIN TRY

BEGIN TRAN
```

```

        INSERT Sales.SpecialOffer (Description, DiscountPct,
Type, Category, StartDate, EndDate, MinQty, MaxQty)
        VALUES (@Description, @DiscountPt, @Type, @Category,
@StartDate, @EndDate, @MinQty, @MaxQty);
        SET @NewProductID = SCOPE_IDENTITY()
        COMMIT TRAN
        RETURN 0
    END TRY

    BEGIN CATCH
        IF @@TRANCOUNT > 0 ROLLBACK TRAN
        INSERT dbo.ErrorLog (ErrorMessage, ErrorLine,
ErrorNumber, ErrorSeverity, ErrorTime, UserName,
ErrorProcedure)
        VALUES (ERROR_MESSAGE(), ERROR_LINE(),
ERROR_NUMBER(), ERROR_SEVERITY(), SYSDATETIME(),
SYSTEM_USER, ERROR_PROCEDURE())
        RETURN 1
    END CATCH;
END

```

Результат представлен на рисунке 5.

Время выполнения: 2024-10-24T13:48:55.3180718+03:00

Рисунок 8 – Результат выполнения команд

Проверка хранимой процедуры с помощью следующего набора команд:

```

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 команд представлен на рисунке 9.

	(Отсутствует имя столбца)
1	17

(затронута одна строка)

(затронута одна строка)

Время выполнения: 2024-10-24T14:00:17.4965729+03:00

Рисунок 9 – Результат проверки хранимой процедуры

Повторная проверка хранимой процедуры с помощью следующего набора команд:

```
DECLARE
    @StartDate datetime,
    @EndDate datetime
SET @StartDate = GETDATE()
SET @EndDate = DATEADD(month, 1, @StartDate)
DECLARE
    @NewId int,
    @ReturntValue int
EXEC @ReturntValue = Sales.AddDiscount 'Half price
off everything',
-0.5, 'Seasonal Discount', 'Customer',
@StartDate, @EndDate,
0, 20, @NewId OUTPUT;
IF (@ReturntValue = 0) SELECT @NewId
ELSE SELECT TOP 1 *
```



```
FROM dbo.ErrorLog
ORDER BY ErrorTime DESC
```

Результат выполнения представлен на рисунке 10.

ErrorLogID	ErrorTime	UserName	ErrorNumber	ErrorSeverity	ErrorState	ErrorProcedure	ErrorLine	ErrorMessage
1	2024-10-24 14:04:59.630	RYMARMARYuser	547	16	NULL	Sales.AddDiscount	5	Конфликт инструкции INSERT с ограничением CHECK " ...

(затронута строк: 0)

(затронута одна строка)

(затронута одна строка)

Время выполнения: 2024-10-24T14:04:59.6612874+03:00

Рисунок 10 – Результат повторной проверки хранимой процедуры

Выводы

В ходе лабораторной работы было проведено углубленное изучение хранимых процедур – мощного инструмента для работы с базами данных. Было успешно реализовано создание хранимых процедур с различными вариантами параметров, начиная от простых процедур без параметров и заканчивая процедурами с входными и выходными параметрами, а также значениями по умолчанию. Это позволило освоить основные методы создания и использования хранимых процедур, которые являются ключевыми для эффективной работы с базами данных.

Особое внимание было уделено практическому применению транзакций, которые играют важную роль в обеспечении целостности данных. Изучение блоков TRY...CATCH дало возможность эффективно обрабатывать ошибки, возникающие при работе с хранимыми процедурами, что гарантирует стабильность и надежность работы с базой данных.

Полученные знания и навыки помогут эффективно использовать хранимые процедуры для различных задач. Хранимые процедуры повышают производительность работы с базой данных, так как они выполняются на сервере базы данных, а не на клиенте. Они также повышают безопасность, так как ограничивают доступ к данным только через определенные процедуры, а не

напрямую к таблицам. Хранимые процедуры также повышают удобство работы, так как позволяют выполнять сложные операции с помощью простого вызова процедуры.