

МИНОБРНАУКИ РОССИИ
Санкт-Петербургский государственный
электротехнический университет
«ЛЭТИ» им. В.И. Ульянова (Ленина)
Кафедра систем автоматизированного проектирования (САПР)

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

	Аршин А. Д
	Баймухамедов Р. Р.
Студенты гр. 3311	Пасечный Л. В.
Преподаватель	<hr/> Новакова Н. Е. <hr/>

Санкт-Петербург

2025

Цель работы

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

Упражнение 1

Создание хранимой процедуры без параметров

Запрос 1

Создайте хранимую хранимую процедуру GetDiscounts в схеме Sales, которая извлекает следующие столбцы из таблицы Sales.SpecialOffer: Description, DiscountPct, Type, Category, StartDate, EndDate, MinQty и MaxQty. Процедура должна возвращать все строки, отсортированные по параметрам StartDate и EndDate.

```
CREATE PROCEDURE Sales.GetDiscounts as
```

```
BEGIN
```

```
    Select Description,DiscountPct,
```

```
    Type,Category,StartDate,EndDate,MinQty,MaxQty
```

```
    From AdventureWorks.Sales.SpecialOffer
```

```
    Order by StartDate,EndDate
```

```
END
```

Результат

Выполнение команд успешно завершено.

Время выполнения: 2025-10-08T19:05:08.6644333+03:00

Запрос 2

Проверка хранимой процедуры

```
EXEC Sales.GetDiscounts
```

Результат

	Description	DiscountPct	Type	Category	StartDate	EndDate	MinQty	MaxQty
1	No Discount	0,00	No Discount	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	Volume Discount	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	Volume Discount	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	Volume Discount	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	Volume Discount	Reseller	2001-07-01 00:00:00.000	2004-06-30 00:00:00.000	41	60
6	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
7	Mountain-100 Clearance Sale	0,35	Discontinued Product	Reseller	2002-05-15 00:00:00.000	2002-06-30 00:00:00.000	0	NULL
8	Sport Helmet Discount-2002	0,10	Seasonal Discount	Reseller	2002-07-01 00:00:00.000	2002-07-31 00:00:00.000	0	NULL
9	Road-650 Overstock	0,30	Excess Inventory	Reseller	2002-07-01 00:00:00.000	2002-08-31 00:00:00.000	0	NULL
10	Mountain Tire Sale	0,50	Excess Inventory	Customer	2003-06-15 00:00:00.000	2003-08-30 00:00:00.000	0	NULL
11	Sport Helmet Discount-2003	0,15	Seasonal Discount	Reseller	2003-07-01 00:00:00.000	2003-07-31 00:00:00.000	0	NULL
12	LL Road Frame Sale	0,35	Excess Inventory	Reseller	2003-07-01 00:00:00.000	2003-08-15 00:00:00.000	0	NULL
13	Touring-3000 Promotion	0,15	New Product	Reseller	2003-07-01 00:00:00.000	2003-09-30 00:00:00.000	0	NULL
14	Touring-1000 Promotion	0,20	New Product	Reseller	2003-07-01 00:00:00.000	2003-09-30 00:00:00.000	0	NULL
15	Half-Price Pedal Sale	0,50	Seasonal Discount	Customer	2003-08-15 00:00:00.000	2003-09-15 00:00:00.000	0	NULL
16	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

(Затронутو строк: 16)

Время выполнения: 2025-10-08T19:07:17.6422472+03:00

Упражнение 2

Создание хранимой процедуры с параметром.

Запрос 1

Создайте хранимую процедуру GetDiscountForCategory в схеме Sales. Эта процедура должна иметь входной параметр @Category, имеющий тип данных nvarchar и принимающий до 50 символов. Она должна извлекать те же столбцы, что и запрос в процедуре GetDiscounts, но фильтровать строки на основе параметра @Category.

CREATE PROC Sales.GetDiscountForCategory @Category **as** nvarchar(50)

as

BEGIN

Select Description,DiscountPct,

Type,Category,StartDate,EndDate,MinQty,MaxQty

From AdventureWorks.Sales.SpecialOffer

Where Category = @Category

Order by StartDate,EndDate

END

Результат

Выполнение команд успешно завершено.

Время выполнения: 2025-10-08T19:15:21.5619877+03:00

Запрос 2

Проверка хранимой процедуры

EXEC Sales.GetDiscountForCategory 'Reseller'

Результат

	Description	DiscountPct	Type	Category	StartDate	EndDate	MinQty	MaxQty
1	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
2	Volume Discount 15 to 24	0.05	Volume Discount	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	Volume Discount	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	Volume Discount	Reseller	2001-07-01 00:00:00.000	2004-06-30 00:00:00.000	41	60
5	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
6	Mountain-100 Clearance Sale	0.35	Discontinued Product	Reseller	2002-05-15 00:00:00.000	2002-06-30 00:00:00.000	0	NULL
7	Sport Helmet Discount-2002	0.10	Seasonal Discount	Reseller	2002-07-01 00:00:00.000	2002-07-31 00:00:00.000	0	NULL
8	Road-650 Overstock	0.30	Excess Inventory	Reseller	2002-07-01 00:00:00.000	2002-08-31 00:00:00.000	0	NULL
9	Sport Helmet Discount-2003	0.15	Seasonal Discount	Reseller	2003-07-01 00:00:00.000	2003-07-31 00:00:00.000	0	NULL
10	LL Road Frame Sale	0.35	Excess Inventory	Reseller	2003-07-01 00:00:00.000	2003-08-15 00:00:00.000	0	NULL
11	Touring-3000 Promotion	0.15	New Product	Reseller	2003-07-01 00:00:00.000	2003-09-30 00:00:00.000	0	NULL
12	Touring-1000 Promotion	0.20	New Product	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	Discontinued Product	Reseller	2004-05-01 00:00:00.000	2004-06-30 00:00:00.000	0	NULL

(Затронуто строк: 13)

Время выполнения: 2025-10-08T19:16:17.3980405+03:00

Упражнение 3

Создание хранимой процедуры с параметрами и значениями по умолчанию

Запрос 1

В схеме Sales создайте хранимую процедуру GetDiscountsForCategoryAndDate. Эта процедура должна иметь входной параметр @Category, как и процедура GetDiscountForCategory, но включает дополнительный входной параметр @DateToCheck datetime. Параметр @DateToCheck должен иметь возможность принимать стандартное значение NULL. Если значение NULL указано на параметр @DateToCheck, задайте для этого параметр значение текущих даты и времени, используя функцию GETDATE ().

CREATE PROC Sales.GetDiscountsForCategoryAndDate

@Category as nvarchar(50),

```

@DateToCheck as datetime = NULL
as
BEGIN
    if(@DateToCheck is null)
        set @DateToCheck = GETDATE()

    Select

    Description,DiscountPct,Type,Category,StartDate,EndDate,MinQty,Max
Qty
    From
        AdventureWorks.Sales.SpecialOffer
    Where
        Category=@Category
        and StartDate <= @DateToCheck
        and EndDate >= @DateToCheck
    Order by
        StartDate,EndDate
END

```

Результат

Выполнение команд успешно завершено.

Время выполнения: 2025-10-08T19:28:44.5471038+03:00

Запрос 2

Проверка хранимой процедуры

EXEC Sales.GetDiscountsForCategoryAndDate 'Reseller'

Результат

Description	DiscountPct	Type	Category	StartDate	EndDate	MinQty	MaxQty
-------------	-------------	------	----------	-----------	---------	--------	--------

(Затронутو строк: 0)

Время выполнения: 2025-10-08T19:29:26.9601044+03:00

Запрос 3

Проверка хранимой процедуры, когда указываются оба параметра

DECLARE @DateToCheck **datetime**

SET @DateToCheck = **DATEADD**(**YEAR**, -1, **GETDATE**())

EXEC Sales.GetDiscountsForCategoryAndDate **'Reseller'**, @DateToCheck

Результат

Description	DiscountPct	Type	Category	StartDate	EndDate	MinQty	MaxQty
-------------	-------------	------	----------	-----------	---------	--------	--------

(Затронуто строк: 0)

Время выполнения: 2025-10-08T19:33:12.7165188+03:00

Упражнение 4

Создание хранимой процедуры с входными и выходными параметрами

Запрос 1

Создайте процедуру AddDiscount в схеме Sales, которая вставляет новые записи в таблицу Sales.SpecialOffer. В следующей таблице указаны требуемые параметры для вставки:

Название параметра	Тип данных
@Description	nvarchar(255)
@DiscountPct	Smallmoney
@Type	nvarchar(50)
@Category	nvarchar(50)
@StartDate	Datetime
@EndDate	Datetime
@MinQty	Int

@MaxQty	Int
@NewProductID	Int output

Команда INSERT должна быть защищена соответствующей обработкой ошибок, и любые ошибки должны быть записаны в таблицу dbo.ErrorLog. Если новая вставка завершится успешно, параметр @NewProductID будет обновлен на значение функции SCOPE_IDENTITY. Возвращаемое значение должно также указывать успех или неудачу вставки. Для обработки ошибок используйте конструкцию Try...Catch.

CREATE PROC Sales.AddDiscount

 @Description **as nvarchar**(255),

 @DiscountPct **as smallmoney**,

 @Type **as nvarchar**(50),

 @Category **as nvarchar**(50) ,

 @StartDate **as Datetime**,

 @EndDate **as Datetime**,

 @MinQty **as int** = 0,

 @MaxQty **as int** = null,

 @NewProductID **as int output**

as

BEGIN

BEGIN TRY

BEGIN TRAN

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()

    COMMIT TRAN

    RETURN 0

END TRY

BEGIN CATCH

    IF @@TRANCOUNT > 0
        ROLLBACK TRAN

    INSERT INTO dbo.ErrorLog (
        ErrorMessage, ErrorLine, ErrorNumber,
        ErrorSeverity, ErrorTime, UserName, ErrorProcedure
    )
    VALUES (
        ERROR_MESSAGE(),                ERROR_LINE(),
        ERROR_NUMBER(),
        ERROR_SEVERITY(),                SYSDATETIME(),
        SYSTEM_USER, ERROR_PROCEDURE()
    )

    RETURN ERROR_NUMBER()

END CATCH

END
```

Результат

Выполнение команд успешно завершено.

Время выполнения: 2025-10-08T19:54:44.2421055+03:00

Запрос 2

Проверка хранимой процедуры, где в результатах запроса возвращается новое значение параметра SpecialOfferID

```
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

(Затронута 1 строка)

17

(Затронута 1 строка)

Время выполнения: 2025-10-08T19:56:49.6050542+03:00

Запрос 3

Повторная проверка хранимой процедуры, проверяем отображается ли запись об ошибке

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

Результат

	ErrorLogID	ErrorTime	UserName	ErrorNumber	ErrorSeverity	ErrorState	ErrorProcedure	ErrorLine	ErrorMessage
1	2	2025-10-08 20:00:08.670	SKRAIPY\sa957	547	16	NULL	Sales.AddDiscount	15	Конфликт инструкции INSERT с ограничением CHECK ...

(Затронуты строки: 0)

(Затронута 1 строка)

(Затронута 1 строка)

Время выполнения: 2025-10-08T20:00:08.7078094+03:00

Вывод

В ходе выполнения лабораторной работы по теме «Создание и использование хранимых процедур» были успешно освоены принципы работы с хранимыми процедурами в системе управления базами данных. Были получены практические навыки создания различных типов хранимых процедур - без параметров, с входными параметрами, с параметрами по умолчанию, а также с входными и выходными параметрами.

В процессе работы научились правильно использовать синтаксис CREATE PROCEDURE, работать с параметрами различных типов данных, реализовывать логику обработки значений по умолчанию. Особое внимание было уделено организации обработки ошибок с помощью конструкции TRY...CATCH, что является важным аспектом обеспечения надежности работы с базой данных. Также был освоен механизм транзакций для обеспечения целостности данных при выполнении операций вставки и обновления.

Важным приобретенным навыком стала работа с выходными параметрами и возвращаемыми значениями, в частности правильное применение функции SCOPE_IDENTITY () для получения идентификаторов вновь созданных записей. В ходе работы была настроена система логирования ошибок в специализированную таблицу, что позволяет отслеживать и анализировать возникающие проблемы в работе процедур.

Несмотря на то, что в некоторых случаях процедуры не возвращали данные из-за временных ограничений в тестовых наборах, это позволило понять важность тщательного тестирования хранимых процедур с различными сценариями использования. В целом, лабораторная работа закрепила теоретические знания о хранимых процедурах и показала их

практическую ценность для организации эффективной и безопасной работы с базами данных.