

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования «Рязанский государственный радиотехнический
университет имени В.Ф. Уткина»
Рязанский станкостроительный колледж

Отчёт о практической работе №17
Разработка хранимых процедур.
По дисциплине
«Основы проектирования баз данных»

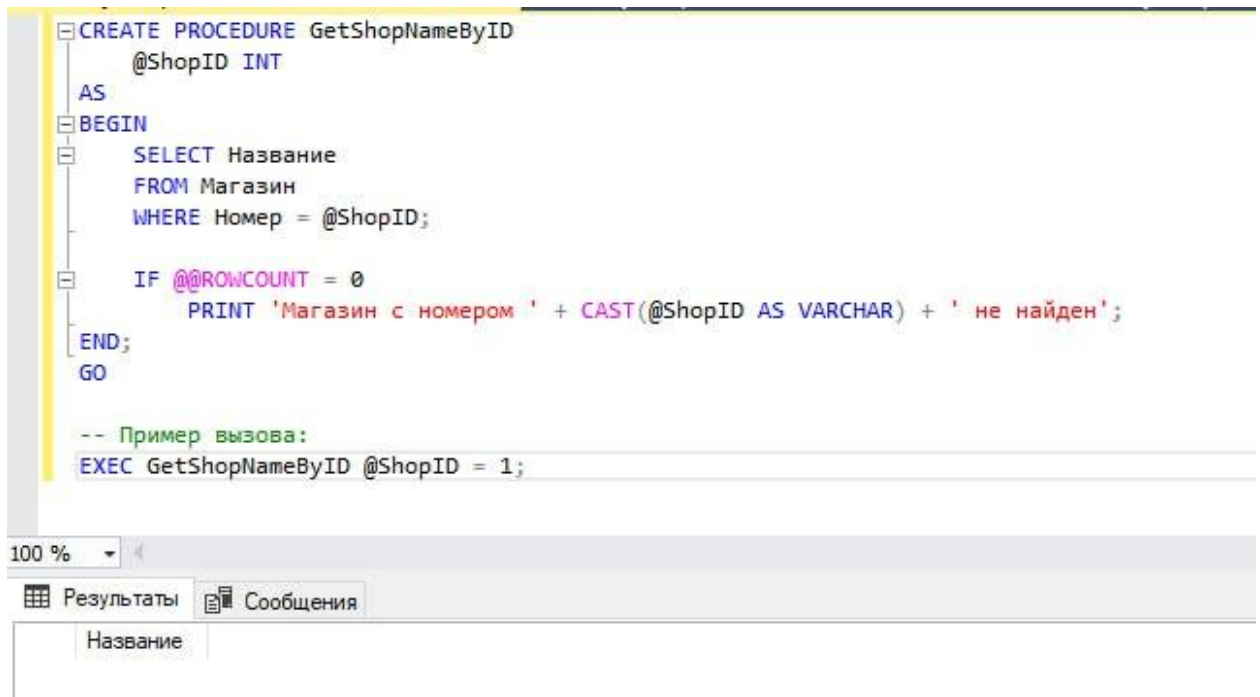
Выполнил:
студент группы ИСП-22
Маркина Н.А
Проверил:
Родин Е.Н

Рязань 2025

Цели работы:

- изучить набор встроенных функций языка T-SQL для работы с разными типами данных;
- научиться использовать встроенные функции в запросах.

1. Изучение оператора CREATE PROCEDURE для создания хранимых процедур.



```
CREATE PROCEDURE GetShopNameByID
    @ShopID INT
AS
BEGIN
    SELECT Название
    FROM Магазин
    WHERE Номер = @ShopID;

    IF @@ROWCOUNT = 0
        PRINT 'Магазин с номером ' + CAST(@ShopID AS VARCHAR) + ' не найден';
END;
GO

-- Пример вызова:
EXEC GetShopNameByID @ShopID = 1;
```

The screenshot shows a SQL script editor with a tree view on the left. The script defines a stored procedure named GetShopNameByID that takes an integer parameter @ShopID. It uses a SELECT statement to retrieve the name of a shop from a table named Магазин where the number matches the parameter. If no results are found, it prints a message. The script concludes with an example call to the procedure using @ShopID = 1. The interface includes a zoom level of 100% and tabs for 'Результаты' (Results) and 'Сообщения' (Messages).

Рис.1

Создала хранимую процедуру GetShopNameByID для поиска названия магазина по его ID с обработкой случая отсутствия записи.

```
CREATE PROCEDURE InsertComponent
    @Model NVARCHAR(50),
    @Name NVARCHAR(100),
    @Manufacturer NVARCHAR(50),
    @Price MONEY
AS
BEGIN
    IF EXISTS (
        SELECT 1
        FROM Комплектующее
        WHERE Наименование = @Name AND Цена = @Price
    )
    BEGIN
        RAISERROR('Ошибка: Комплектующее с таким названием и ценой уже существует', 16, 1);
        RETURN;
    END

    INSERT INTO Комплектующее (Модель, Наименование, [Фирма-производитель], Цена)
    VALUES (@Model, @Name, @Manufacturer, @Price);

    PRINT 'Комплектующее успешно добавлено';
END;
GO

-- Пример вызова:
EXEC InsertComponent
    @Model = 'KB-2023',
    @Name = 'Клавиатура',
    @Manufacturer = 'Logitech',
    @Price = 2500;
```

100 %

Сообщения

(затронута одна строка)
Комплектующее успешно добавлено

Время выполнения: 2025-05-18T18:05:39.4167467+03:00

Рис.2

Создала хранимую процедуру **InsertComponent** для добавления комплектующего с проверкой на дубликаты по названию и цене. В случае выводится подтверждение, при обнаружении дубликата – «ОШИБКА»

```
CREATE PROCEDURE GetComponentInfo
    @ComponentName NVARCHAR(100),
    @Currency CHAR(3) = 'RUB' -- RUB, USD, EUR
AS
BEGIN
    DECLARE @Rate DECIMAL(10,4);

    -- Установка курсов валют (условные значения)
    SET @Rate = CASE @Currency
        WHEN 'USD' THEN 0.013
        WHEN 'EUR' THEN 0.012
        ELSE 1
    END;

    IF NOT EXISTS (
        SELECT 1
        FROM Комплектующее
        WHERE Наименование LIKE '%' + @ComponentName + '%'
    )
    BEGIN
        RAISERROR('Ошибка: Комплектующее с таким названием не найдено', 16, 1);
        RETURN;
    END

    SELECT
        к.Модель,
        к.Наименование AS 'Комплектующая',
        к.Цена * @Rate AS 'Цена',
        SUM(н.Количество) AS 'Общее количество',
        SUM(н.Количество * к.Цена) * @Rate AS 'Общая стоимость',
        @Currency AS 'Валюта'
    FROM
        Комплектующее к
    JOIN
        Наличие н ON к.Модель = н.Модель
    WHERE
        к.Наименование LIKE '%' + @ComponentName + '%'
        AND н.Год = 2019
    GROUP BY
        к.Модель, к.Наименование, к.Цена;
END;
GO
```

100 %

Результаты Сообщения

Модель	Комплектующая	Цена	Общее количество	Общая стоимость	Валюта
--------	---------------	------	------------------	-----------------	--------

Рис.3

Создала хранимую процедуру GetComponentInfo для поиска информации о комплектующих с возможностью конвертации цены в USD, EUR или RUB. Процедура проверяет наличие записи, выводит модель, название, цену в выбранной валюте, общее количество и стоимость. При отсутствии данных возвращает ошибку.

```

CREATE PROCEDURE ApplyDiscountToComponent
    @ComponentModel NVARCHAR(50)
AS
BEGIN
    DECLARE @TotalQuantity INT;
    DECLARE @CurrentPrice MONEY;
    DECLARE @NewPrice MONEY;
    DECLARE @Discount DECIMAL(5,2);

    -- Проверка существования комплектующего
    IF NOT EXISTS (SELECT 1 FROM Комплектующее WHERE Модель = @ComponentModel)
    BEGIN
        RAISERROR('Ошибка: Комплектующее с такой моделью не найдено', 16, 1);
        RETURN;
    END

    -- Получение текущей цены
    SELECT @CurrentPrice = Цена
    FROM Комплектующее
    WHERE Модель = @ComponentModel;

    -- Расчет суммарного количества за 3 года
    SELECT @TotalQuantity = SUM(Количество)
    FROM Наличие
    WHERE Модель = @ComponentModel
    AND Год BETWEEN YEAR(GETDATE())-3 AND YEAR(GETDATE());

```

Рис.4

Создала хранимую процедуру ApplyDiscountToComponent для применения скидки к комплектующему на основе его модели. Процедура проверяет наличие модели, получает текущую цену и рассчитывает общее количество за последние 3 года.

```

-- Если нет данных о наличии, считаем количество = 0
IF @TotalQuantity IS NULL
    SET @TotalQuantity = 0;

-- Определение размера скидки
SET @Discount = CASE
    WHEN @TotalQuantity = 0 THEN 0.5
    WHEN @TotalQuantity BETWEEN 1 AND 100 THEN 0.2
    ELSE 0.1
END;

-- Расчет новой цены
SET @NewPrice = @CurrentPrice * (1 - @Discount);

-- Обновление цены
UPDATE Комплектующее
SET Цена = @NewPrice
WHERE Модель = @ComponentModel;

-- Вывод информации о примененной скидке
SELECT
    @ComponentModel AS 'Модель',
    @CurrentPrice AS 'Старая цена',
    @NewPrice AS 'Новая цена',
    @Discount*100 AS 'Скидка (%)',
    @TotalQuantity AS 'Суммарное количество за 3 года';

END;
GO

-- Пример вызова:
EXEC ApplyDiscountToComponent @ComponentModel = 'KB-2023';

```

100 %

Результаты Сообщения

	Модель	Старая цена	Новая цена	Скидка (%)	Суммарное количество за 3 года
1	KB-2023	2500,00	1250,00	50,00	0

Рис.5

Дописала хранимую процедуру **ApplyDiscountToComponent**, которая применяет скидку к комплектующему в зависимости от его наличия за 3 года. Если данных нет, скидка 50%, при количестве 1-100-20%, в остальных случаях – 10%. Процедура обновляет цену и выводит детали скидки.

Заключение:

В ходе выполненной работы я изучила и разработала хранимые процедуры.