Министерство образования Республики Беларусь

Учреждение образования

«Брестский Государственный технический университет»

Кафедра ИИТ

**Лабораторная работа №4**

По дисциплине «Проектирование баз знаний»

Тема: «Хранимые процедуры, функции и триггеры в MS SQL Server»

**Выполнил:**

Студент 3 курса

Группы ИИ-24

Крейдич А.А.

**Проверил:**

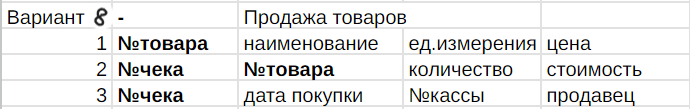
Козик И.Д.

Брест 2025

**Цель работы:** Изучение и решение заданий по реализации хранимых процедур, функций и триггеровразных видов и фильтров в базах данных.

**Ход работы:**

**Вариант 8**



Задание.

1. Изучить материал, приведенный в “К лабораторной работе 4.doc”.

2. На основании логической модели (в соответствии с вариантом Лабораторной работы №1) создать и проверить на работоспособность хранимые процедуры, функции и триггеры для каждой из таблиц БД: реализовать хранимые процедуры, функции и триггеры разных видов для каждой таблицы .

**Таблица Товары**

CREATE TABLE Товары (

"№товара" INTEGER PRIMARY KEY,

наименование NVARCHAR(255),

"единица измерения" NVARCHAR(255),

цена INTEGER

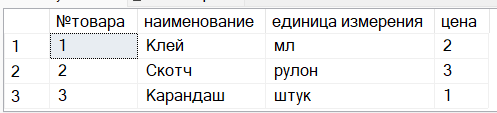
);

INSERT INTO Товары ("№товара", наименование, "единица измерения", цена) VALUES

(1, 'Клей', 'мл', 2),

(2, 'Скотч', 'рулон', 3),

(3, 'Карандаш', 'штук', 1);



**Процедура:**

-- Создание хранимой процедуры для добавления товара

CREATE PROCEDURE добавить\_товар

@номер\_товара INTEGER,

@название NVARCHAR(255),

@единица\_измерения NVARCHAR(255),

@цена INTEGER

AS

BEGIN

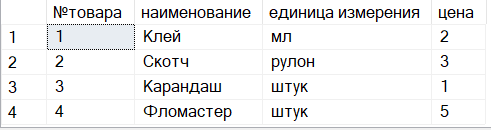
INSERT INTO Товары ("№товара", наименование, "единица измерения", цена)

VALUES (@номер\_товара, @название, @единица\_измерения, @цена);

END;

-- Проверка работы процедур

EXEC добавить\_товар @номер\_товара = 4, @название = 'Фломастер', @единица\_измерения = 'штук', @цена = 5;



**Функция:**

-- Создание функции для подсчета количества товаров с выбранной единицей измерения

CREATE FUNCTION количество\_товаров (@в\_единице NVARCHAR(255))

RETURNS INTEGER

AS

BEGIN

DECLARE @количество INTEGER;

SELECT @количество = COUNT(\*) FROM Товары WHERE "единица измерения" = @в\_единице;

RETURN @количество;

END;

-- Проверка работы функций

SELECT dbo.количество\_товаров('штук') AS Количество\_товаров;



**Триггер:**

-- Создание триггера для проверки цены товара (она не должна быть меньше 1)

CREATE TRIGGER проверить\_цену\_товара

ON Товары

AFTER INSERT, UPDATE

AS

BEGIN

IF EXISTS (

SELECT 1 FROM Товары WHERE цена < 1

)

BEGIN

RAISERROR('Цена товара должна быть больше 0!', 16, 1);

ROLLBACK TRANSACTION;

END;

END;

-Проверка  
INSERT INTO Товары ("№товара", наименование, "единица измерения", цена) VALUES

(5, 'Стол', 'штук', -1);



**Таблица Чек**

CREATE TABLE Чек (

"№чека" INTEGER PRIMARY KEY,

"№товара" INTEGER,

количество INTEGER,

стоимость INTEGER,

FOREIGN KEY ("№товара") REFERENCES Товары("№товара")

);

INSERT INTO Чек ("№чека", "№товара", количество, стоимость) VALUES

(1, 1, 3, 6),

(2, 2, 4, 12),

(3, 3, 2, 2);



**Процедура:**

-- Создание хранимой процедуры для добавления чека

CREATE PROCEDURE добавить\_чек

@номер\_чека INTEGER,

@номер\_товара INTEGER,

@количество INTEGER,

@стоимость INTEGER

AS

BEGIN

INSERT INTO Чек ("№чека", "№товара", количество, стоимость)

VALUES (@номер\_чека, @номер\_товара, @количество, @стоимость);

END;

-- Проверка работы процедур

EXEC добавить\_чек @номер\_чека = 4, @номер\_товара = 4, @количество = 5, @стоимость = 15;



**Функция:**

-- Создание функции для расчета общей стоимости выбранного чека

CREATE FUNCTION общая\_стоимость\_чека (@номер\_чека INTEGER)

RETURNS INTEGER

AS

BEGIN

DECLARE @общая\_стоимость INTEGER;

SELECT @общая\_стоимость = SUM(стоимость) FROM Чек WHERE "№чека" = @номер\_чека;

RETURN @общая\_стоимость;

END;

-- Проверка работы функций

SELECT dbo.общая\_стоимость\_чека(1) AS Общая\_стоимость\_чека;



**Триггер:**

-- Создание триггера для проверки стоимости чека (она должна совпадать с расчетом количества \* цена)

CREATE TRIGGER проверить\_стоимость\_чека

ON Чек

AFTER INSERT, UPDATE

AS

BEGIN

IF EXISTS (

SELECT 1

FROM Чек C

JOIN Товары T ON C."№товара" = T."№товара"

WHERE C.стоимость <> C.количество \* T.цена

)

BEGIN

RAISERROR('Неверная стоимость чека!', 16, 1);

ROLLBACK TRANSACTION;

END;

END;

-Провека

INSERT INTO Чек ("№чека", "№товара", количество, стоимость) VALUES

(5, 1, 3, 1);  


**Таблица Продавец:**

CREATE TABLE Продавец (

"№чека" INTEGER PRIMARY KEY,

"дата покупки" DATE,

"№кассы" INTEGER,

продавец NVARCHAR(255),

FOREIGN KEY ("№чека") REFERENCES Чек("№чека")

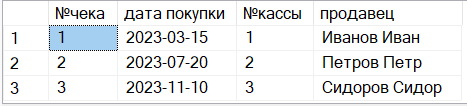
);

INSERT INTO Продавец ("№чека", "дата покупки", "№кассы", продавец) VALUES

(1, '2023-03-15', 1, 'Иванов Иван'),

(2, '2023-07-20', 2, 'Петров Петр'),

(3, '2023-11-10', 3, 'Сидоров Сидор');



**Процедура:**

-- Создание хранимой процедуры для добавления продавца

CREATE PROCEDURE добавить\_продавца

@номер\_чека INTEGER,

@дата\_покупки DATE,

@номер\_кассы INTEGER,

@продавец NVARCHAR(255)

AS

BEGIN

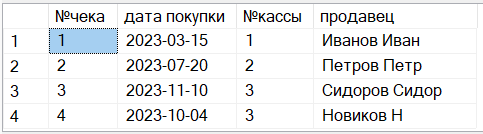
INSERT INTO Продавец ("№чека", "дата покупки", "№кассы", продавец)

VALUES (@номер\_чека, @дата\_покупки, @номер\_кассы, @продавец);

END;

-- Проверка работы процедур

EXEC добавить\_продавца @номер\_чека = 4, @дата\_покупки = '2023-10-04', @номер\_кассы = 3, @продавец = 'Новиков Н';



**Функция:**

-- Создание функции для получения информации о продавце

CREATE FUNCTION получить\_информацию\_о\_продавце (@номер\_чека INTEGER)

RETURNS TABLE

AS

RETURN

(

SELECT "№чека", "дата покупки", "№кассы", продавец

FROM Продавец

WHERE "№чека" = @номер\_чека

);

-- Проверка работы функций

SELECT \* FROM dbo.получить\_информацию\_о\_продавце(1);



**Триггер:**

-- Создание триггера для проверки номера кассы (чтобы касса была >= 1)

CREATE TRIGGER проверить\_номер\_кассы

ON Продавец

AFTER INSERT, UPDATE

AS

BEGIN

IF EXISTS (

SELECT 1 FROM Продавец WHERE "№кассы" < 1

)

BEGIN

RAISERROR('Номер кассы должен быть больше или равен 1!', 16, 1);

ROLLBACK TRANSACTION;

END;

END;

-Проверка  
INSERT INTO Продавец ("№чека", "дата покупки", "№кассы", продавец) VALUES

(4, '2023-03-13', -1, 'Иванов Иван');



**Вывод:** Изучил и решил задания по реализации хранимых процедур, функций и триггеровразных видов и фильтров в базах данных.