Министерство образования Республики Беларусь  
Учреждение образования  
«Брестский государственный технический университет»  
Кафедра ИИТ

Лабораторная работа №4  
По дисциплине: «Хранимые процедуры, функции и триггеры в MS SQL Server 2005/2008»

Выполнил:  
студент 3 курса  
группы ИИ-21(I)  
Романко Н.А.

Проверил:  
Савонюк В. А.

Брест 2023

Цель работы:

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 (не менее 7 на каждую таблицу).

**Вариант 12**

Ход работы:

Структура таблиц:

Table1:

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

Table2:

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Table3:

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

**Таблица table1:**

**Функция для вставки данных:**

DELIMITER //

CREATE FUNCTION InsertIntoTable1(p\_№ INT, p\_adress VARCHAR(255), p\_FIO VARCHAR(255))

RETURNS INT

deterministic

BEGIN

INSERT INTO table1 (№, adress, FIO) VALUES (p\_№, p\_adress, p\_FIO);

RETURN 1;

END;

// DELIMITER ;

**Вызов функции:**

select InsertIntoTable1(12, "Пушкина", "Сергеев Сергей Сергеевич");

**Вывод функции:**

**Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание**

**Функция для удаления данных по номеру:**

**DELIMITER //**

**CREATE FUNCTION DeleteFromTable1(p\_№ INT)**

**RETURNS INT**

**DETERMENISTIC**

**BEGIN**

**DELETE FROM table1 WHERE № = p\_№;**

**RETURN 1;**

**END;**

**// DELIMITER ;**

**Вызов функции:**

select DeleteFromTable1(2);

**Вывод функции:**

**Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание**

**Процедура для обновления данных:**

DELIMITER //

CREATE PROCEDURE UpdateTable1(p\_№ INT, p\_adress VARCHAR(255), p\_FIO VARCHAR(255))

BEGIN

UPDATE table1

SET adress = p\_adress, FIO = p\_FIO

WHERE № = p\_№;

END;

//DELIMITER ;

**Вызов процедуры:**

call UpdateTable1(4, "Брест", "Иванов Иван Иванович");

**Вывод процедуры:**

**Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание**

**Процедура для удаления данных по номеру:**

DELIMITER //

CREATE PROCEDURE DeleteFromTable1ByNumber(p\_№ INT)

BEGIN

DELETE FROM table1 WHERE № = p\_№;

END;

//

DELIMITER ;

**Вызов процедуры:**

call DeleteFromTable1ByNumber(12);

**Вывод процедуры:**

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

**Триггер перед вставкой пустой записи:**

DELIMITER //

CREATE TRIGGER BeforeInsertCheckTable1

BEFORE INSERT ON table1

FOR EACH ROW

BEGIN

IF NEW.adress IS NULL OR NEW.FIO IS NULL THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Не может быть вставлена запись с пустыми данными';

END IF;

END;

//DELIMITER ;

**Вывод триггера:**

****

**Триггер при обновлении таблицы:**

DELIMITER //

CREATE TRIGGER AfterUpdateTable1

AFTER UPDATE ON table1

FOR EACH ROW

BEGIN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Таблица обновленна';

END;

//DELIMITER ;

**Вывод триггера:**



**Триггер при вставке данных в таблицу:**

DELIMITER //

CREATE TRIGGER BeforeInsertTable1

BEFORE INSERT ON table1

FOR EACH ROW

BEGIN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Вставленны новые данные';

END;

//DELIMITER ;

**Вывод триггера:**

****

**Таблица table2:**

**Триггер при вставке данных в таблицу:**

DELIMITER //

CREATE TRIGGER AfterInsertTable22

AFTER INSERT ON table2

FOR EACH ROW

BEGIN

signal sqlstate '45000'

set message\_text = 'Новые данные добавленны';

END;

//DELIMITER ;

**Вывод триггера:**



**Триггер при удалении данных:**

**DELIMITER //**

**CREATE TRIGGER BeforeDeleteTable2**

**BEFORE DELETE ON table2**

**FOR EACH ROW**

**BEGIN**

**signal sqlstate '45000'**

**set message\_text = 'Данные удаленны';**

**END;**

**//DELIMITER ;**

**Вывод триггера**

****

**Триггер при обновлении данных:**

DELIMITER //

CREATE TRIGGER AfterUpdateTable2

AFTER UPDATE ON table2

FOR EACH ROW

BEGIN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Данные обновленны';

END;

//DELIMITER ;

**Вывод триггера:**

****

**Функция для вставки данных**

DELIMITER //

CREATE FUNCTION InsertIntoTable2(p\_category CHAR(1), p\_min\_age INT, p\_max\_age INT)

RETURNS INT

BEGIN

INSERT INTO table2 (category, min\_age, max\_age) VALUES (p\_category, p\_min\_age, p\_max\_age);

RETURN 1;

END;

//DELIMITER ;

**Вызов функции**

select InsertIntoTable2("O", 5, 21);

**Вывод функции**

**Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание**

**Функция для обновления данных:**

DELIMITER //

CREATE FUNCTION UpdateTable2(p\_category CHAR(1), p\_min\_age INT, p\_max\_age INT)

RETURNS INT

deterministic

BEGIN

UPDATE table2

SET min\_age = p\_min\_age, max\_age = p\_max\_age

WHERE category = p\_category;

RETURN 1;

END;

//

DELIMITER ;

**Вызов функции:**

select UpdateTable2("А", 13, 16);

**Вывод функции:**

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

**Процедура для обновления минимального возраста:**

DELIMITER //

CREATE PROCEDURE UpdateMinAgeForCategory(p\_category CHAR(1), p\_new\_min\_age INT)

BEGIN

UPDATE table2

SET min\_age = p\_new\_min\_age

WHERE category = p\_category;

END;

//

DELIMITER ;

**Вызов процедуры:**

call UpdateMinAgeForCategory("Д", 6);

**Вывод процедуры:**

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

**Процедура для обновления максимального возраста:**

DELIMITER //

CREATE PROCEDURE UpdateMaxAgeForCategory(p\_category CHAR(1), p\_new\_max\_age INT)

BEGIN

UPDATE table2

SET max\_age = p\_new\_max\_age

WHERE category = p\_category;

END;

//DELIMITER ;

**Вызов фукнции:**

call UpdateMaxAgeForCategory("Д", 13);

**Вывод процедуры:**

**Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание**

**Таблица table3:**

**Триггер после вставки записи:**

DELIMITER //

CREATE TRIGGER AfterInsertTable3

AFTER INSERT ON table3

FOR EACH ROW

BEGIN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Данные добавленны';

END;

//DELIMITER ;

**Вывод триггера:**

****

**Триггер после обновления данных:**

DELIMITER //

CREATE TRIGGER AfterUpdateTable3

AFTER UPDATE ON table3

FOR EACH ROW

BEGIN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Данные обновленны';

END;

//DELIMITER ;

**Вывод триггера:**



**Триггер после удаления данных:**

DELIMITER //

CREATE TRIGGER AfterDeleteTable3

AFTER DELETE ON table3

FOR EACH ROW

BEGIN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Данные убраны';

END;

//DELIMITER ;

**Вывод триггера:**

****

**Процедура для удаления данных:**

DELIMITER //

CREATE PROCEDURE DeleteFromTable3(p\_№ INT)

BEGIN

DELETE FROM table3 WHERE № = p\_№;

END;

//DELIMITER ;

**Вызов процедуры:**

call DeleteFromTable3(1);

**Вывод процедуры:**

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

**Процедура для вставки данных:**

DELIMITER //

CREATE PROCEDURE InsertIntoTable3(p\_№ INT, p\_category CHAR(1), p\_teacher VARCHAR(255))

BEGIN

INSERT INTO table3 (№, category, teacher) VALUES (p\_№, p\_category, p\_teacher);

END;

//

DELIMITER ;

**Вызов процедуры:**

call InsertIntoTable3(1, "Б", "Симонов");

**Вывод процедуры:**

**Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание**

**Функция для вставки данных:**

DELIMITER //

CREATE FUNCTION InsertIntoTable3(p\_№ INT, p\_category CHAR(1), p\_teacher VARCHAR(255))

RETURNS INT

deterministic

BEGIN

INSERT INTO table3 (№, category, teacher) VALUES (p\_№, p\_category, p\_teacher);

RETURN 1;

END;

//DELIMITER ;

**Вызов функции:**

select InsertIntoTable3(5, "Г", "Васильевич");

**Вывод функции:**

**Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание**

**Функция для обновления категории:**

DELIMITER //

CREATE FUNCTION UpdateTeacherCategory(p\_№ INT, p\_new\_category CHAR(1))

RETURNS INT

deterministic

BEGIN

UPDATE table3

SET category = p\_new\_category

WHERE № = p\_№;

RETURN 1;

END;

//DELIMITER ;

**Вызов функции:**

select UpdateTeacherCategory(3, "В");

**Вывод функции:**

**Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание**

**Вывод:** в ходе данной лабораторной работы мной были изучены способы реализации процедур, функций и триггеров для баз данных в MySQL.