**Министерство науки и высшего образования**

**Российской Федерации**

**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Р****язанский государственный радиотехнический университет  
имени В.Ф. Уткина»**

Кафедра ЭВМ

Отчёт о лабораторной работе № 6

«Разработка триггеров»

по дисциплине

**«Проектирование моделей данных»**

Выполнили:

студенты группы 140

Сафаров Д. А.

Тимохин Е. М.

Проверили:

ас. каф. ЭВМ. Панина И.С.

ас. каф. ЭВМ Баранова С.Н.

**Практическая часть**

1. Изучение оператора CREATE TRIGGER для создания триггеров к базовым таблицам БД.

1.1. Разработать AFTER-триггер, который позволяет удалять информацию обо всех предоставляемых услугах из таблицы Услуги, кроме услуги с кодом 1. Считать, что пользователю предоставляется возможность удаления множества записей в таблице.

CREATE TRIGGER After\_DeleteService

ON Услуги

AFTER Delete

AS

IF EXISTS (SELECT 1 FROM deleted WHERE Код = 1)

BEGIN

INSERT INTO Услуги

SELECT \* FROM deleted

WHERE Код = 1

END

GO

===\

1.2. Разработать INSTEAD OF-триггер для рассматриваемого в п.1.1 задания.

CREATE TRIGGER DeleteServiceInsteadOf

ON Услуги

INSTEAD OF DELETE

AS

-- Удаляем разрешенные записи из основной таблицы

DELETE FROM Услуги WHERE Код IN

(SELECT Код FROM deleted WHERE Код !=1)

GO

===\

1.3. Разработать триггер, который позволяет изменять значение стоимости приёма только на большее значение. Считать, что пользователю предоставляется возможность модификации множества записей в таблице.

CREATE TRIGGER UpdateServiceCost

ON ОказанныеУслуги

INSTEAD OF UPDATE

AS

BEGIN

IF UPDATE(Стоимость)

BEGIN

DECLARE @Пациент smallint,

@Время time,

@НоваяСтоимость smallmoney

DECLARE cur CURSOR FOR

SELECT Пациент, Время, Стоимость

FROM inserted

OPEN cur

WHILE @@FETCH\_STATUS = 0

BEGIN

FETCH NEXT FROM cur INTO @Пациент, @Время, @НоваяСтоимость

UPDATE ОказанныеУслуги

SET Стоимость = CASE WHEN @НоваяСтоимость > Стоимость

THEN @НоваяСтоимость

ELSE Стоимость END

WHERE Пациент = @Пациент AND Время = @Время

END

CLOSE cur

DEALLOCATE cur

END

END

GO

===\

2. Создание триггеров для представлений.

2.1. Создать представление с информацией об оказанных услугах. Представление должно включать столбцы: Номер пациента, Фамилия пациента, Код услуги, Название услуги, Время, Стоимость (в руб.).

CREATE VIEW Информация\_об\_оказанных\_услугах

AS

SELECT

П.Номер AS 'Номер пациента',

П.Фамилия AS 'Фамилия пациента',

ОУ.Услуга AS 'Код услуги',

У.Название AS 'Название услуги',

ОУ.Время,

ОУ.Стоимость AS 'Стоимость, руб.'

FROM

ОказанныеУслуги ОУ

INNER JOIN

Пациенты П ON ОУ.Пациент = П.Номер

INNER JOIN

Услуги У ON ОУ.Услуга = У.Код

GO

===\

2.2. Разработать триггер, который позволяет вносить информацию о приёмах через представление. Считать, что пользователю предоставляется возможность вставки только одной записи через представление. Предусмотреть вывод сообщения об ошибке в случае ввода информации о несуществующих услуге или пациенте.

CREATE TRIGGER InsertAppointmentInfo ON Информация\_об\_оказанных\_услугах

INSTEAD OF INSERT

AS

BEGIN

-- Проверка на существование указанной услуги и пациента

IF (SELECT [Номер пациента] FROM inserted) IN (SELECT Номер FROM Пациенты)

AND(SELECT [Код услуги] FROM inserted) IN (SELECT Код FROM Услуги)

BEGIN

-- Вставка информации об оказанных услугах

INSERT INTO ОказанныеУслуги (Пациент, Услуга, Время, Стоимость)

SELECT

i.[Номер пациента],

i.[Код услуги],

i.Время,

i.[Стоимость, руб.]

FROM

inserted i;

END

ELSE

BEGIN

-- Если указаны несуществующие пациент или услуга, выводим сообщение об ошибке

PRINT 'Ошибка: Введена информация о несуществующем пациенте или услуге.'

ROLLBACK TRAN

END

END;

===\

**Вывод**

В ходе лабораторной работы была рассмотрена и закреплена на практике теория по триггерам в SQL.