Цель работы: Практическое освоение приемов и методов обеспечения целостности данных

## Ход работы:

Рисунок 1 – Создание стандартного значения

UNUM	OCENKA	UDATE	SNUM	PNUM
2	4	1980-03-12	1	13
3	5	1981-05-13	2	1
6	3	1978-08-22	3	12
7	4	1984-09-23	4	2
8	5	1990-01-01	5	3
10	2	1987-03-03	6	5
11	3	1999-03-03	7	6
12	4	1982-11-14	8	7
13	5	1976-10-11	9	8
17	5	2019-04-03	2	3
18	4	2019-04-03	2	3
NULL	NULL	NULL	NULL	NULL

Рисунок 2 – После автоматического задания UDATE

```
-- Create Bound Rule template
   -- -----
   -- This feature is marked for deprecation
   USE laba_1
   G0
   -- Define the rule for use in the database
  □CREATE RULE MARK_VALUES
   AS
       @MARK BETWEEN 1 AND 5
   G0
   -- Bind the rule to a specific column in a table
  □EXEC sp_bindrule
       'dbo.mark_Values',
       'dbo.usp.ocenka'
   G0
75 %
      ▼ 4
В Сообщения
  Правило привязано к столбцу таблицы.
```

Рисунок 3 — Создание правила для ограничения оценки между значениями 1 и 5

Рисунок 4 - Поддержка стратегий целостности данных

Закрыть

Добавить

Удалить

```
USE laba_1
GO

IF OBJECT_ID ('dbo.INSCHECK','TR') IS NOT NULL
DROP TRIGGER INSCHECK
GO

CCREATE TRIGGER INSCHECK
ON [dbo].[USP]
FOR INSERT
AS
DECLARE @X INT SELECT @X=I.SNUM FROM USP A, INSERTED I WHERE A.SNUM=I.SNUM
IF NOT(EXISTS (SELECT * FROM STUDENTS WHERE SNUM=@X))
BEGIN
ROLLBACK TRAN
RAISERROR('Own6ka целостности!', 16,10)
END
GO

'5 %

COOGUGHHUR
```

Рисунок 5 – Создание триггера

```
USE laba_1
GO

□IF OBJECT_ID ('dbo.IN_USE','TR') IS NOT NULL
DROP TRIGGER IN_USE
GO

□CREATE TRIGGER IN_USE
ON [dbo].[USP]
FOR INSERT
AS
DECLARE @X1 INT, @X2 INT,@X3 smalldatetime, @X4 INT, @X5 INT
SELECT @X1=I.UNUM, @X2=I.OCENKA, @X3=I.UDATE, @X4=I.SNUM, @X5=I.PNUM FROM INSERTED I
INSERT INTO TAB(UNUM1, OCENKA1, UDATE1, SNUM1, PNUM1) VALUES (@X1,@X2,@X3,@X4,@X5)
GO

□COOGЩЕНИЯ
Выполнение команд успешно вавершено.
```

Рисунок 6 – Создание триггера in\_use

AN-PC\SQLEXPRESS.laba_1 - dbo.TAB   □   ×   AN-PC\SQLEXPRESS.laba_1 - db								
	UNUM1	OCENKA1	UDATE1	PNUM1	SNUM1			
•	25	5	2019-04-03	12	5			
*	NULL	NULL	NULL	NULL	NULL			

Рисунок 7 — Таблица ТАВ после автоматического добавления новой записи из USP

```
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE PROCEDURE INT_STIP_KF
(@KF_INDEX REAL, @FOR_SUM SMALLMONEY)
AS

SET NOCOUNT ON;
UPDATE STUDENTS SET STIP=STIP*@KF_INDEX WHERE STIP=@FOR_SUM
GO
'5 %

©COOGЩения

Выполнение команд услешно завершено.
```

Рисунок 8 – Создание хранимой процедуры

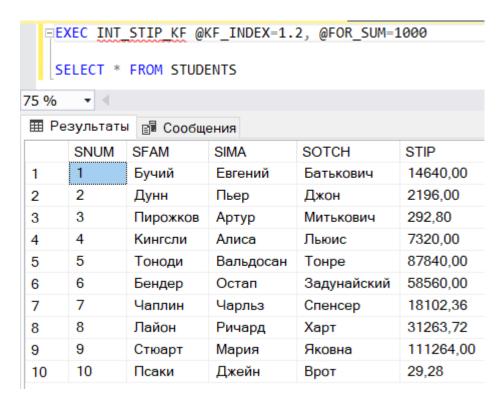


Рисунок 9 – Пример работы хранимой процедуры

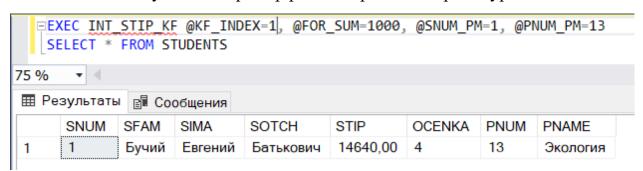


Рисунок 10 – Хранимая процедура, обрабатывающая только успешно закрытый предмет студента

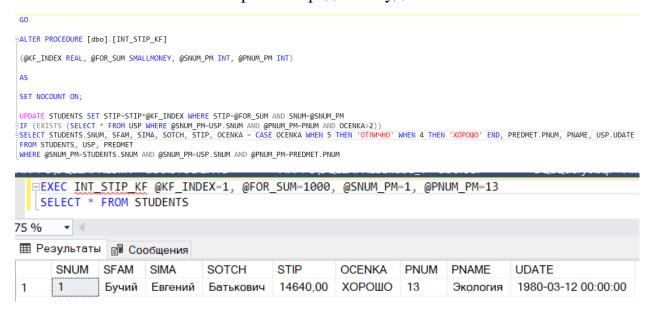


Рисунок 11 – Хранимая процедура, заменяющая оценку, и ее код