Национальный Исследовательский Университет

«Московский Энергетический Институт»

Кафедра прикладной математики и искусственного интеллекта

Тема: Модульное тестирование кода базы данных.

Студент: Ростовых Александра

Москва 2021

**Цель работы**

Научиться разрабатывать модульные тесты кода БД. Разработать модульный тест для кода базы данных с применением Microsoft Visual Studio.

**1. Создать с помощью MS SQL Server Management Studio новую тестовую БД.**

Код создание таблиц БД:

CREATE TABLE Employee

(

Id\_Emp INTEGER NOT NULL PRIMARY KEY,

Sal INTEGER NOT NULL CHECK (Sal between 1000 and 80000),

Sur VARCHAR(30) NOT NULL,

Num INTEGER NOT NULL,

);

CREATE TABLE Client

(

Id\_Cl INTEGER NOT NULL PRIMARY KEY,

SurCl VARCHAR(30) NOT NULL ,

Category INT NOT NULL DEFAULT (1),

NumCl INTEGER NOT NULL,

Cty VARCHAR(20) NOT NULL,

);

CREATE TABLE Serv

(

Id\_Serv INTEGER NOT NULL PRIMARY KEY,

Typ VARCHAR(30) NOT NULL,

Cost INTEGER NOT NULL CHECK (Cost between 500 and 100000),

Stat VARCHAR(30) NOT NULL CHECK (Stat IN('Отменен','Выполнен','Выполняется')),

Cond VARCHAR(30) NOT NULL,

);

CREATE TABLE Contract

(

Id\_Cont INTEGER NOT NULL PRIMARY KEY,

Dte DATETIME NOT NULL,

Dte\_end DATETIME,

Id\_Serv INTEGER NOT NULL FOREIGN KEY REFERENCES Serv(Id\_Serv)

ON DELETE CASCADE

ON UPDATE CASCADE,

Id\_Cl INTEGER NOT NULL FOREIGN KEY REFERENCES Client(Id\_Cl)

ON DELETE CASCADE

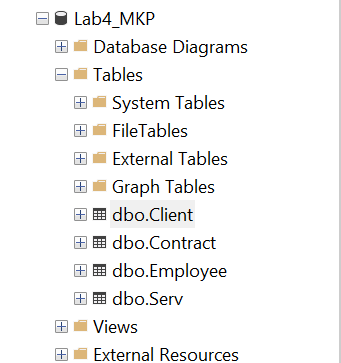
ON UPDATE CASCADE,

Id\_Emp INTEGER NOT NULL FOREIGN KEY REFERENCES Employee(Id\_Emp)

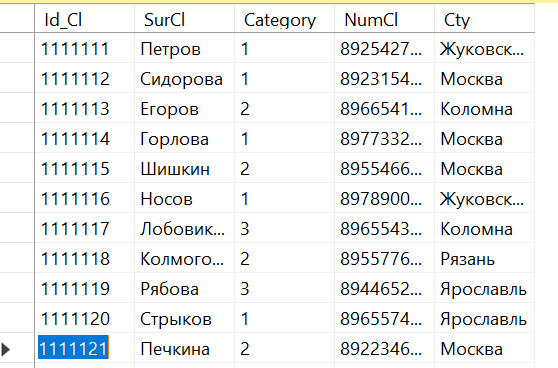
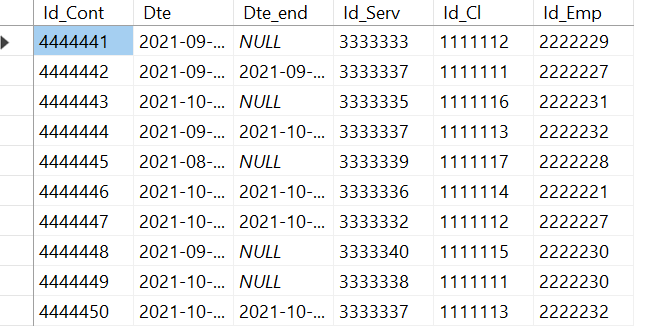
ON DELETE CASCADE

ON UPDATE CASCADE

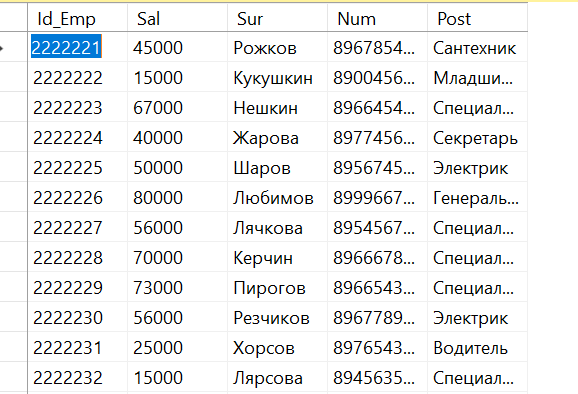
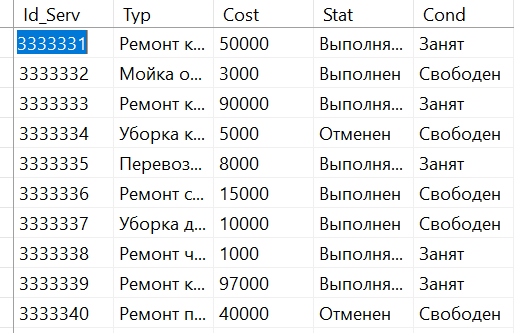
);

****

Затем заполним эти таблицы какими-либо данными.

*Таблица “Client” Таблица “Contract”*

*Таблица “Employee” Таблица “Serv”*

**2. Создать в тестовой БД хранимые процедуры (2-3 штуки).**

Процедура, выводящая список клиентов, заказавших заданную услугу в фирме:

CREATE PROCEDURE ListOfClients (@Tp varchar(30)) AS

BEGIN

SELECT SurCl, Category, NumCl, Cty FROM Client

JOIN Contract ON Contract.Id\_Cl = Client.Id\_Cl

JOIN Serv ON Serv.Id\_Serv = Contract.Id\_Serv

WHERE Typ = @Tp ORDER BY SurCl;

END

Процедура, выводящая список клиентов, заказавших услугу, стоимость которой меньше заданной:

CREATE PROCEDURE less (@Cos Integer) AS

BEGIN

SELECT Client.SurCl, Client.Category, Client.NumCl, Client.Cty, Serv.Cost, Serv.Typ FROM Client

JOIN Contract ON Contract.Id\_Cl = Client.Id\_Cl

JOIN Serv ON Serv.Id\_Serv = Contract.Id\_Serv

WHERE Serv.Cost <= @Cos ;

END

Процедура для подсчета суммы, на которую клиент подписал контракты:

CREATE PROCEDURE Summ (@Sur varchar(30), @SumC int OUTPUT) AS

BEGIN

SELECT @SumC = SUM(Serv.Cost) FROM Client

JOIN Contract ON Contract.Id\_Cl = Client.Id\_Cl

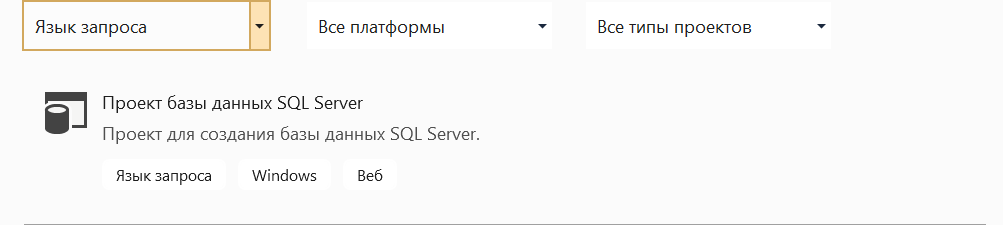
JOIN Serv ON Serv.Id\_Serv = Contract.Id\_Serv

WHERE SurCl = @Sur;

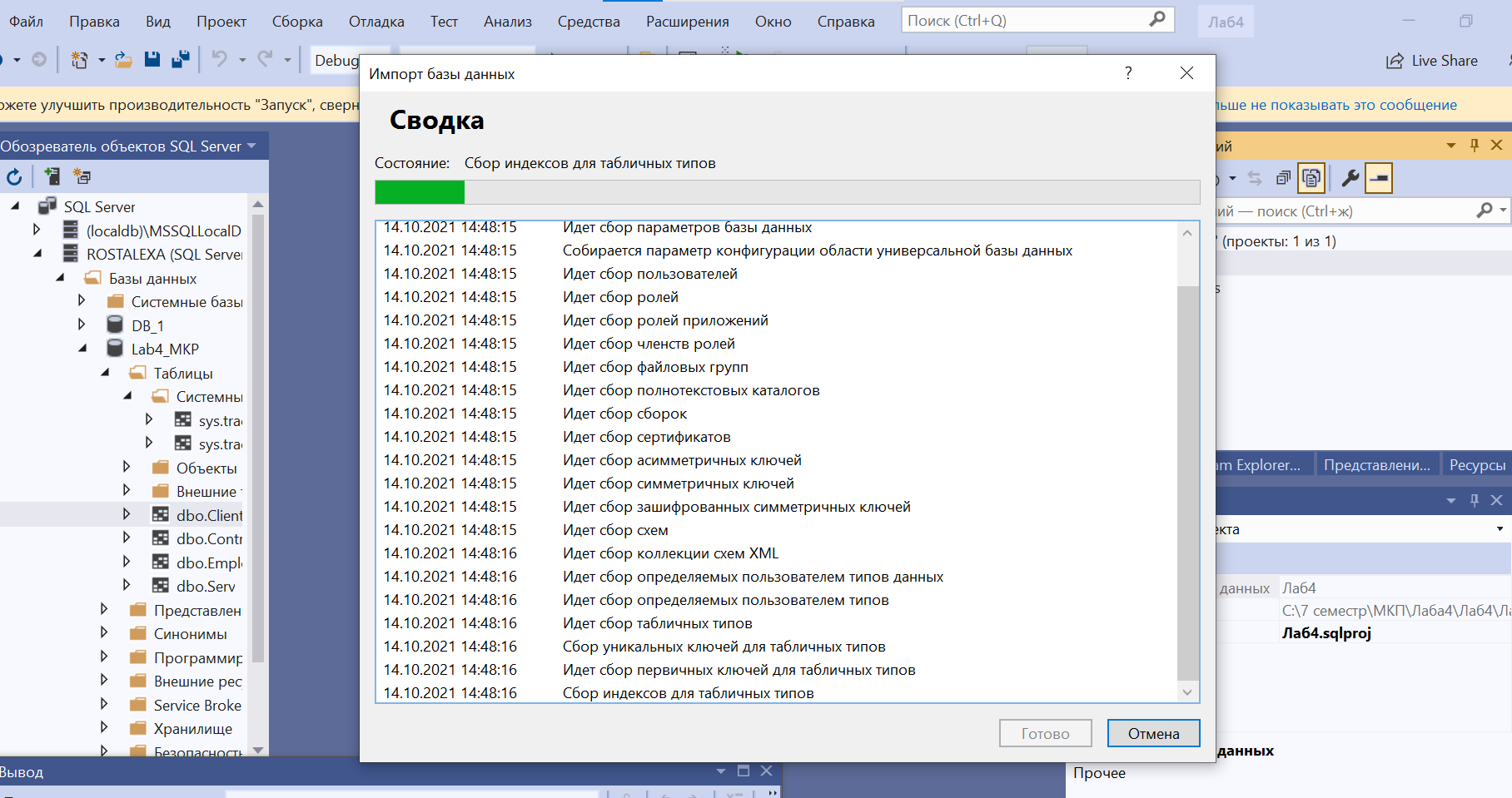
END

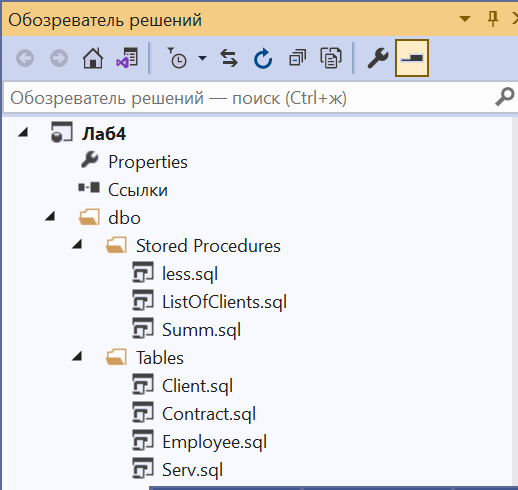
**3. Создать в Visual Studio новый тестовый проект (Test Project).**

Создаем проект

****

В проект импортируем нашу базу данных (Lab4\_MKP), кликая правой кнопкой на «Решение» в обозревателе, затем нажимаем «Импорт» и подключаем нашу БД.



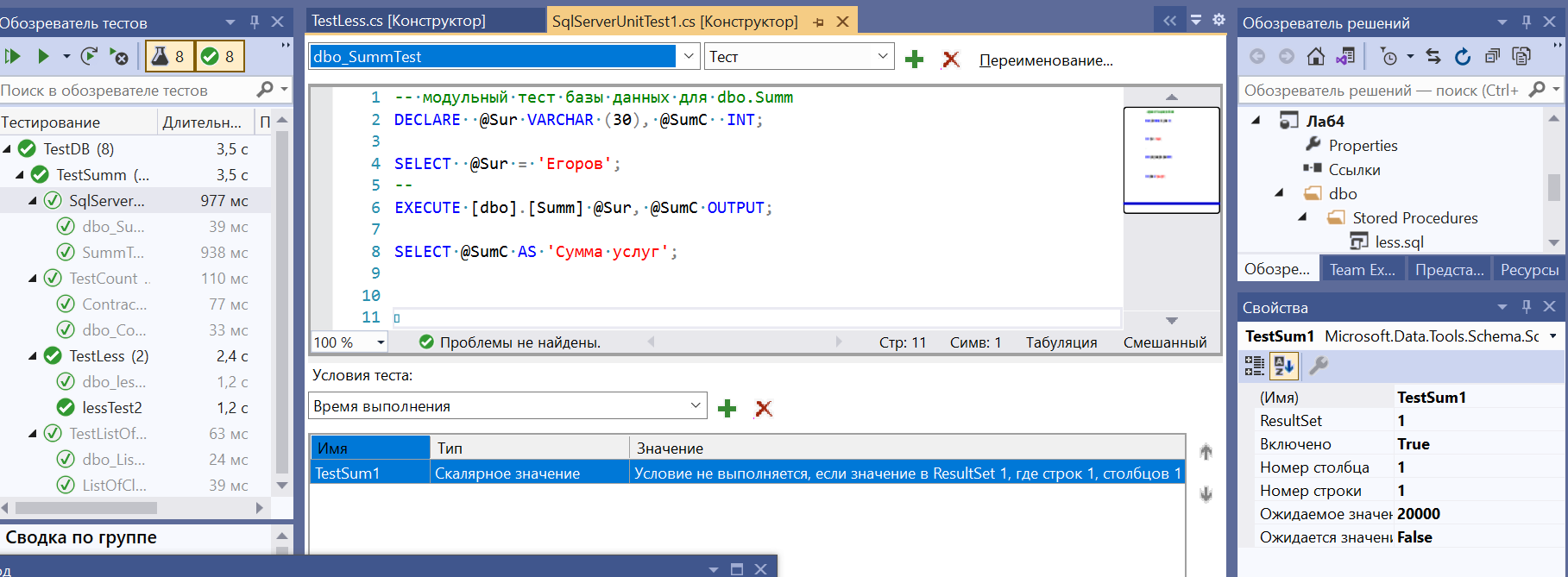


**4. Добавить в тестовый проект модульный тест БД (Database Unit Test).**

Правой кнопкой мыши можем кликнуть на хранимую процедуру, для нее будем писать тест (выбираем пункт «Создать модульные тесты»).

**5. Реализовать не менее пяти тестирующих функций. При разработке этих функций использовать различные выборки и условия корректности выполнения тестов.**

Прописываем модульные тесты для хранимых процедур. Внизу в условиях теста мы можем выбирать тип теста, справа в свойствах, можем определять ожидаемые значения. При нажатии на зеленый плюс вверху, можем добавлять тесты.

****

*Пример теста для хранимой процедуры «Summ»*

*SqlServerUnitTest1.cs*

Реализовано два теста для процедуры, вычисляющей сумму контрактов, которые подписал клиент. Типом является скалярное значение, проверяется соответствие ожидаемой суммы реальной. Ожидаемое значение задается справа в свойствах.

*Dbo\_SummTest*

*DECLARE @Sur VARCHAR (30), @SumC INT;*

*SELECT @Sur = 'Егоров';*

*EXECUTE [dbo].[Summ] @Sur, @SumC OUTPUT;*

*SELECT @SumC AS 'Сумма услуг';*

*SummTest2*

*DECLARE @Sur VARCHAR (30), @SumC INT;*

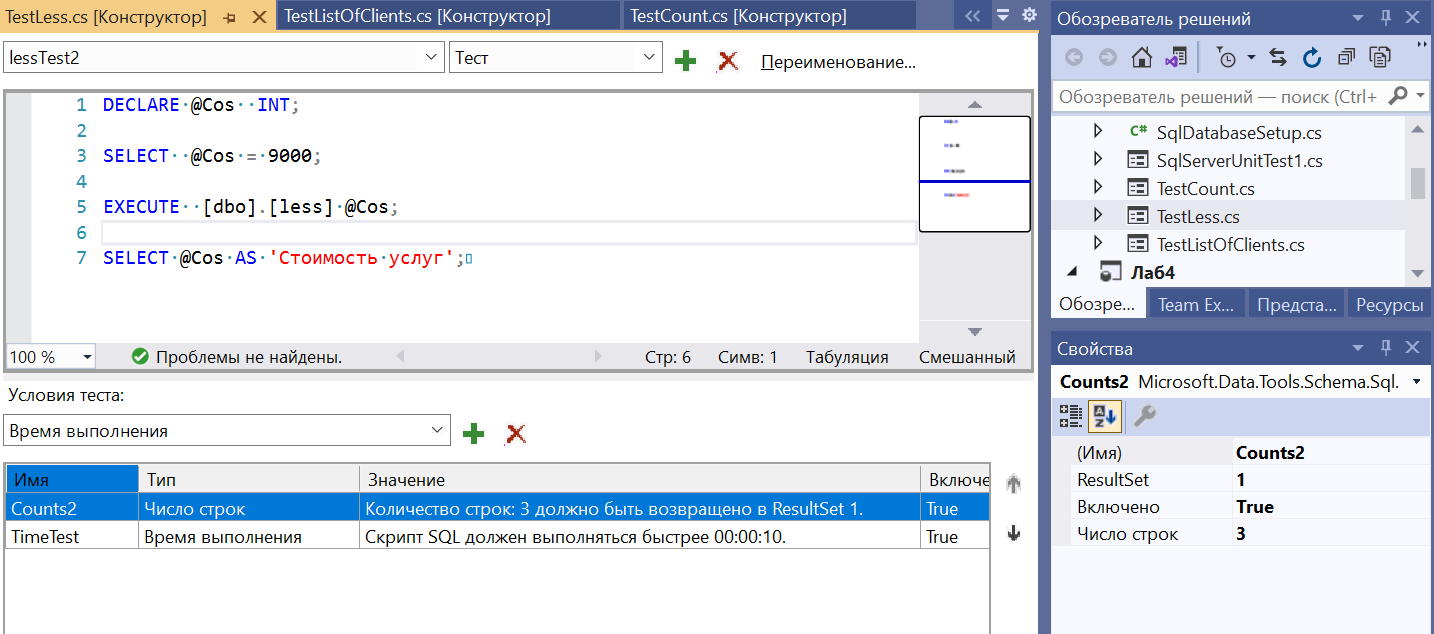
*SELECT @Sur = 'Петров';*

*EXECUTE [dbo].[Summ] @Sur, @SumC OUTPUT;*

*SELECT @SumC AS 'Сумма услуг';*

*TestLess.cs*

Реализовано два теста для процедуры, выводящей список клиентов, стоимость услуг для которых не превышает заданной суммы. Условия теста заключаются в том, что проверяется число строк в итоговом списке клиентов, а так же время выполнения в тесте *lessTest2.* Скрипт должен выполняться быстрее чем 00:00:10. Ожидаемые значения так же задаются в свойствах справа.

**

*Dbo\_lessTest*

DECLARE @Cos INT;

SELECT @Cos = 15000;

EXECUTE [dbo].[less] @Cos;

SELECT @Cos AS 'Стоимость услуг';

*lessTest2*

*DECLARE @Cos INT;*

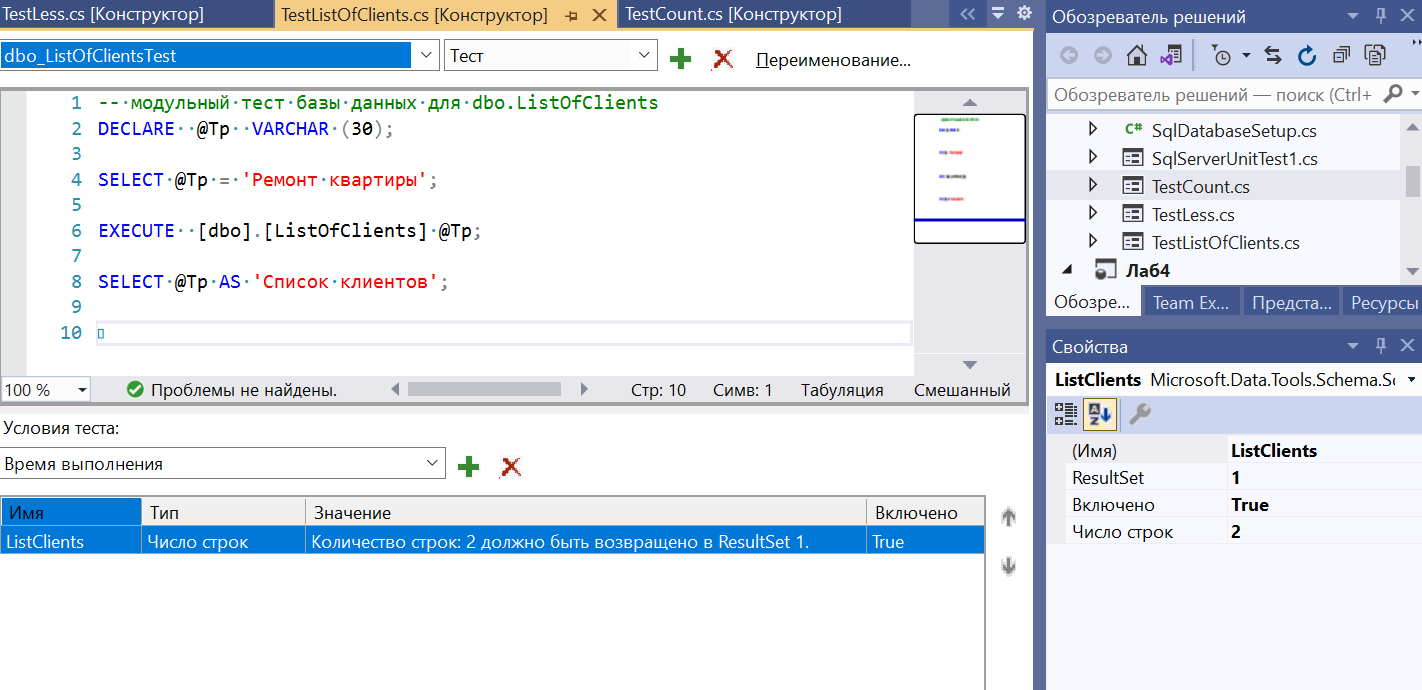
*SELECT @Cos = 9000;*

*EXECUTE [dbo].[less] @Cos;*

*SELECT @Cos AS 'Стоимость услуг';*

*TestListOfClients.cs*

Здесь реализовано два теста для процедуры, которая выводит список клиентов. Заказавших услугу заданного типа. Условия теста заключаются в том, что происходит проверка на количество возвращаемых строк в таблице (Число ожидаемых строк определяется в свойствах справа, тип теста «Число строк»)

**

*Пример теста для хранимой процедуры «ListOfClients»*

*Dbo\_ListOfClientsTest*

*DECLARE @Tp VARCHAR (30);*

*SELECT @Tp = 'Ремонт квартиры';*

*EXECUTE [dbo].[ListOfClients] @Tp;*

*SELECT @Tp AS 'Список клиентов';*

*ListOfClients2*

*DECLARE @Tp VARCHAR (30);*

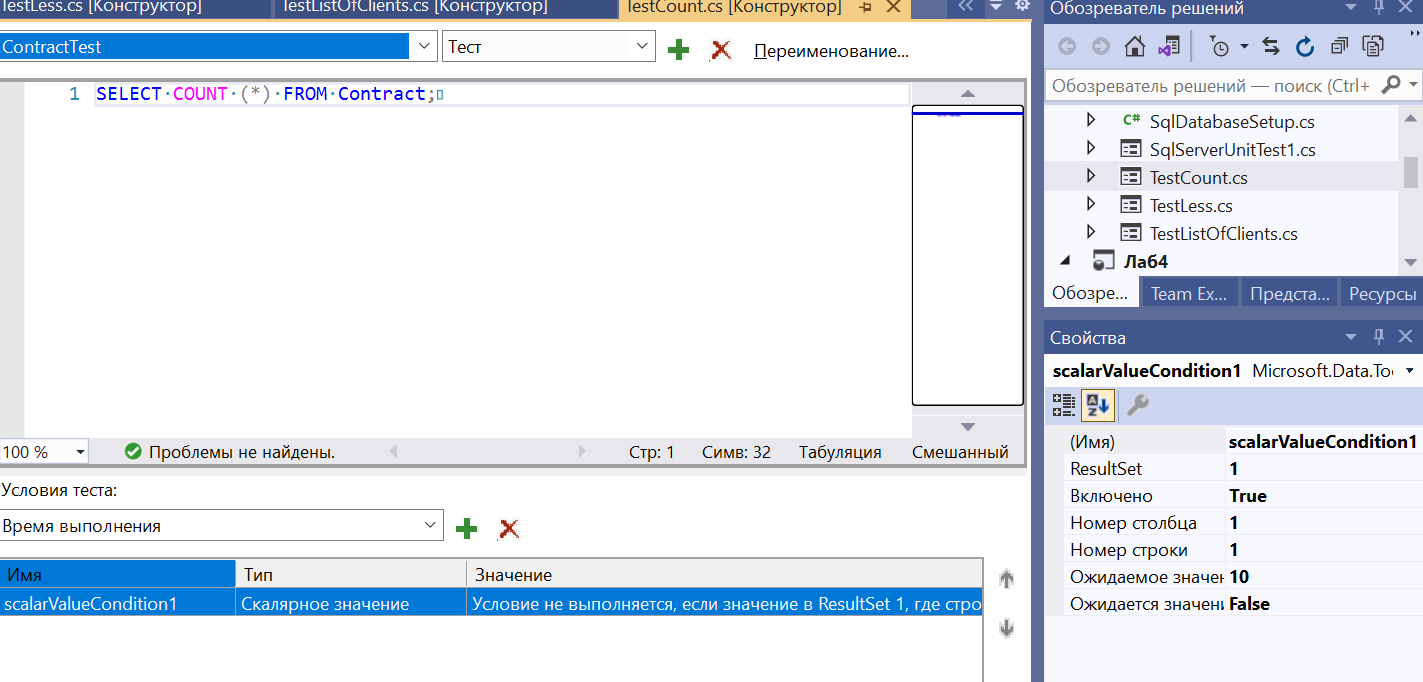
*SELECT @Tp = 'Перевозка мебели';*

*EXECUTE [dbo].[ListOfClients] @Tp;*

*SELECT @Tp AS 'Список клиентов';*

*TestCount*

Тест для проверки количества внесенных контрактов или клиентов. Проверяется количество в заполненной таблице БД.



*Пример теста*

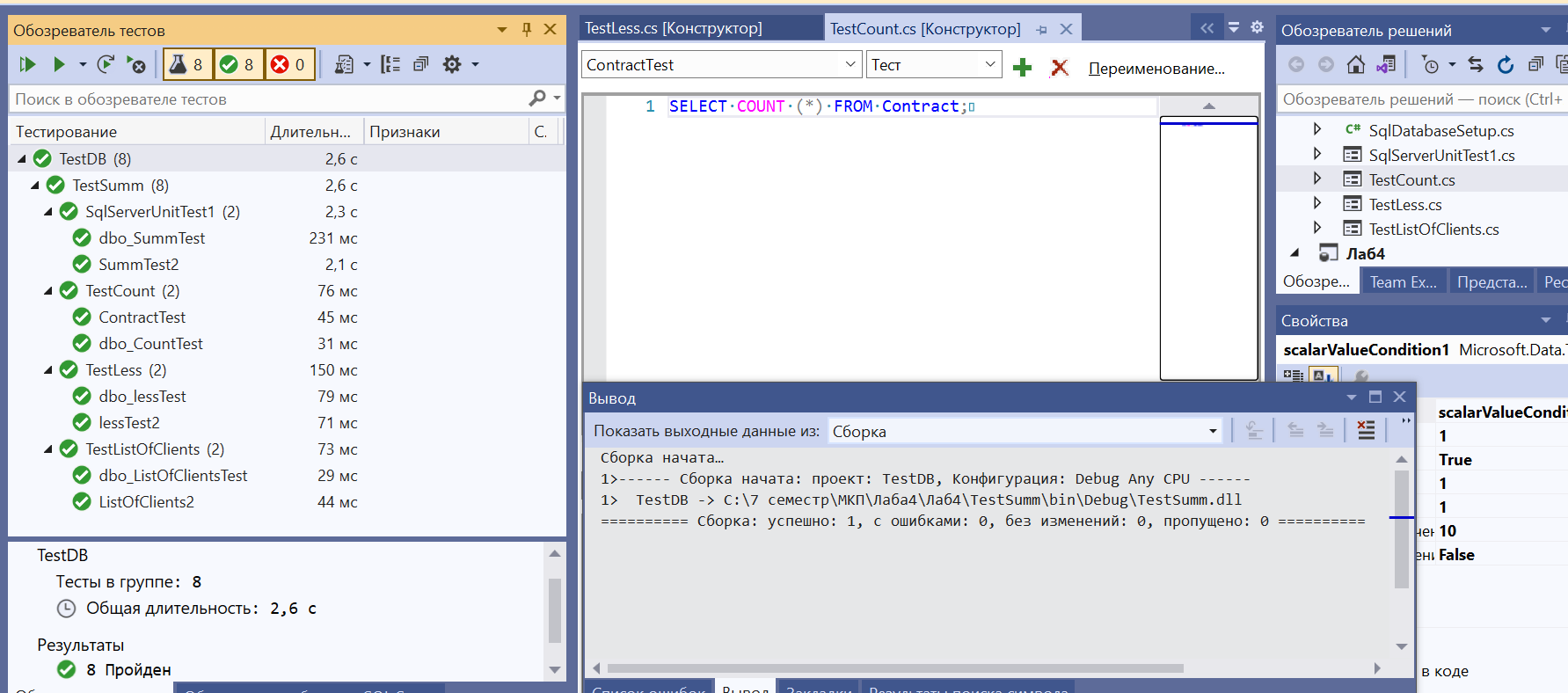
*ContractTest*

*SELECT COUNT (\*) FROM Contract;*

*Dbo\_CountTest*

*SELECT COUNT (\*) FROM Client;*

**6. Запустить модульные тесты. Проанализировать результаты запуска.**



*Результаты запуска тестов*

Все тесты успешно пройдены!

**7. Внести в одну из хранимых процедур изменения, приводящие к ошибкам.**

В процедуре less изменим знак <= на просто =, (можем сделать это с помощью обозревателя объектов SQL), обновим нашу процедуру, запустим заново все тесты, видим, что тесты для процедуры less не пройдены. Ошибки пойманы модульным тестом.

*CREATE PROCEDURE less (@Cos Integer) AS*

*BEGIN*

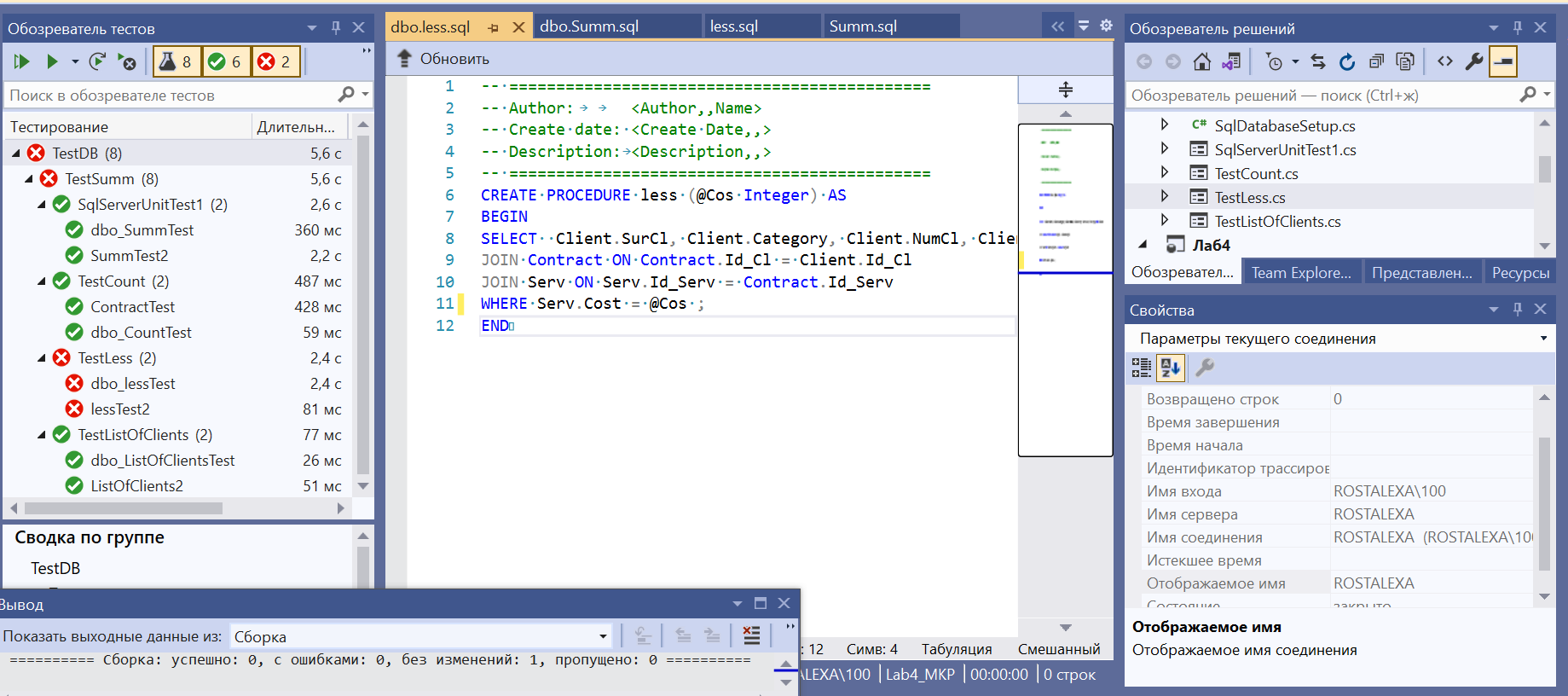
*SELECT Client.SurCl, Client.Category, Client.NumCl, Client.Cty, Serv.Cost, Serv.Typ FROM Client*

*JOIN Contract ON Contract.Id\_Cl = Client.Id\_Cl*

*JOIN Serv ON Serv.Id\_Serv = Contract.Id\_Serv*

*WHERE Serv.Cost = @Cos ;*

*END*

****

*Запуск тестов для измененной процедуры*