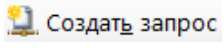
Лабораторная работа 5.
Хранимые процедуры

Цель работы: научиться создавать хранимые процедуры в среде SQL Server Management Studio.

Теоретические сведения

Хранимая процедура – SQL-запрос, который имеет параметры, то есть он выполняется как обычная процедура. В зависимости от значения параметров хранимой процедуры мы получаем тот или иной результат запроса. В SQL сервере хранимые процедуры реализуют динамические запросы, выполняемые на стороне сервера.

Рассмотрим создание хранимых процедур при помощи команд языка SQL.

Чтобы отобразить хранимые процедуры рабочей БД панели «Обозреватель объектов» нужно выбрать пункт «Программирование», а в нем – «Хранимые процедуры». Чтобы создать новую процедуру при помощи команд языка SQL нужно щелкнуть левой кнопкой мыши по кнопке 

на панели инструментов. В рабочей области окна сервера появится вкладка SQLQuery1.sql, где нужно набрать код новой процедуры, который имеет следующий синтаксис:

```
CREATE { PROC | PROCEDURE } procedure_name [ ; number ]
    [ { @parameter data_type }
        [ = default ] [ OUT | OUTPUT ] [READONLY]
    ] [ ,...n ]
[ WITH [ RECOMPILE ] [ ,...n ] ]
[ FOR REPLICATION ]
AS { <sql_statement> [;][ ...n ] }
[;]

<sql_statement> ::=
{ [ BEGIN ] statements [ END ] }
```

Здесь:

- `procedure_name` – имя новой хранимой процедуры. Имена процедур должны соответствовать правилам, предъявляемым к идентификаторам, и должны быть уникальными.
- `number` – необязательное целое число, используемое для группирования процедур с одним именем. Все сгруппированные процедуры можно удалить, выполнив одну инструкцию `DROP PROCEDURE`.
- `@parameter` – параметр процедуры. В инструкции `CREATE PROCEDURE` можно объявить один или более параметров. При выполнении процедуры значение каждого из объявленных параметров должно быть указано пользователем, если для параметра не определено значение по умолчанию или значение не задано равным другому параметру. Хранимая процедура может иметь не более 2 100 параметров. Определяет имя параметра, используя знак `@` как первый символ. Имя параметра должно соответствовать правилам для идентификаторов. Параметры являются локальными в пределах процедуры; в разных процедурах

могут быть использованы одинаковые имена параметров.

- data_type – тип данных параметра. Все типы данных, которые могут использоваться в качестве параметра хранимой процедуры Transact-SQL. Можно использовать определяемый пользователем табличный тип, чтобы объявить возвращающий табличное значение параметр в качестве параметра хранимой процедуры Transact-SQL.
- default – значение параметра по умолчанию. Если значение default определено, процедуру можно выполнить без указания значения соответствующего параметра. Значение по умолчанию должно быть константой или может равняться NULL.
- OUTPUT показывает, что параметр процедуры является выходным. Значение этого параметра можно получить при помощи инструкции EXECUTE. Используйте параметры OUTPUT для возврата значений коду, вызвавшему процедуру.
- READONLY указывает, что параметр не может быть обновлен или изменен в тексте процедуры. Если тип параметра является определяемым пользователем табличным типом, должно быть указано ключевое слово READONLY.
- RECOMPILE показывает, что компонент Database Engine не кэширует план выполнения процедуры и что процедура компилируется во время выполнения.
- EXECUTE AS определяет контекст безопасности, в котором должна быть выполнена хранимая процедура.
- <sql_statement> – одна или несколько инструкций языка SQL, которые будут включены в состав процедуры.

Если параметры сравниваются с какими-то полями или выражениями, то они должны иметь точно такой же тип данных, как эти поля или выражения.

После создания процедура помещается в раздел «Хранимые процедуры» текущей БД на панели «Обозреватель объектов».

Чтобы посмотреть информацию о хранимой процедуре необходимо выполнить команду

```
EXEC SP_HELPTEXT <Имя процедуры>
```

Хранимые процедуры могут быть запущены следующей командой

```
EXEC <Имя процедуры> [<Параметр1>, <Параметр2>, …]
```

Здесь <Имя процедуры> – имя выполняемой процедуры; <Параметр1>, <Параметр2>, … – значения параметров.

Пример: Создание хранимой процедуры, которая выводит имена студентов со средним баллом, большим заданной величины:

```
CREATE PROCEDURE СрБАЛЛ
@X Real
AS
SELECT *
FROM Студенты
WHERE
```

```
(Оценка1 + Оценка2 + Оценка3)/3>@X
```

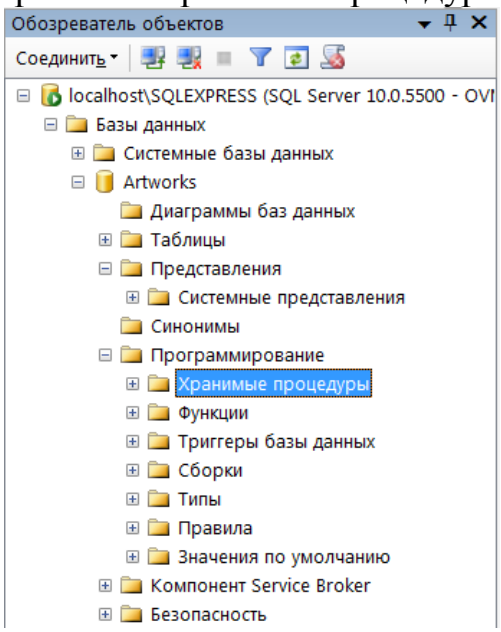
Команда вызова этой процедуры выглядит следующим образом:

```
EXEC СрБАЛЛ 4
```

Команда выводит всех студентов, у которых средний балл больше 4.

Пример

Для работы с хранимыми процедурами в обозревателе объектов необходимо выделить папку «Программирование/Хранимые процедуры» базы данных.



Создадим процедуру, вычисляющую среднее трёх чисел. Для создания новой хранимой процедуры нужно щелкнуть правой кнопкой мыши по папке «Хранимые процедуры» и в появившемся меню выбрать пункт «Создать хранимую процедуру». Появится окно кода новой хранимой процедуры (рис. 1).

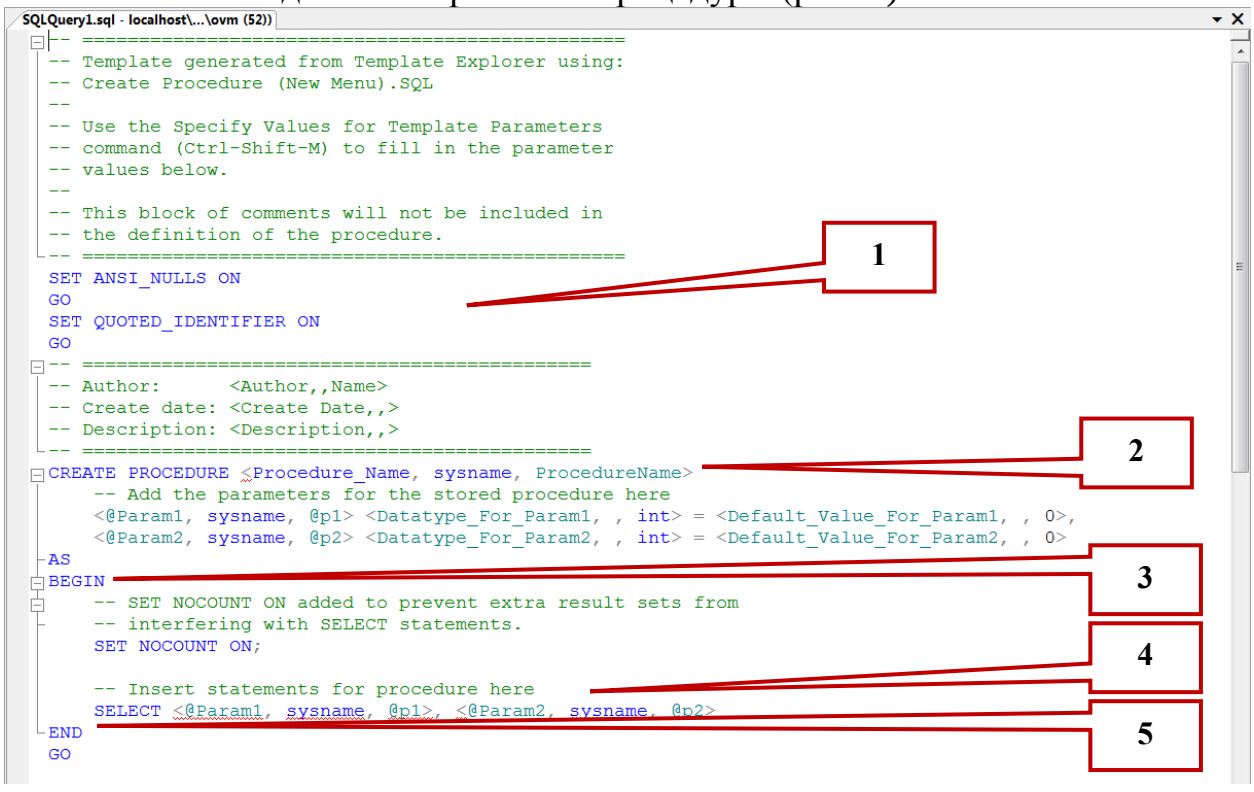


Рис. 1.

Хранимая процедура имеет следующую структуру:

1. Область настройки параметров синтаксиса процедуры. Позволяет настраивать некоторые синтаксические правила, используемые при наборе кода процедуры. В нашем случае это:

`SET ANSI_NULLS ON` включает использование значений `NULL` в кодировке `ANSI`,

`SET QUOTED_IDENTIFIER ON` включает возможность использования двойных кавычек для определения идентификаторов;

2. Область определения имени процедуры и параметров, передаваемых в процедуру. Определение параметров имеет следующий синтаксис:

```
@<Имя параметра> <Тип данных> = <Значение по умолчанию>
```

Параметры разделяются между собой запятыми;

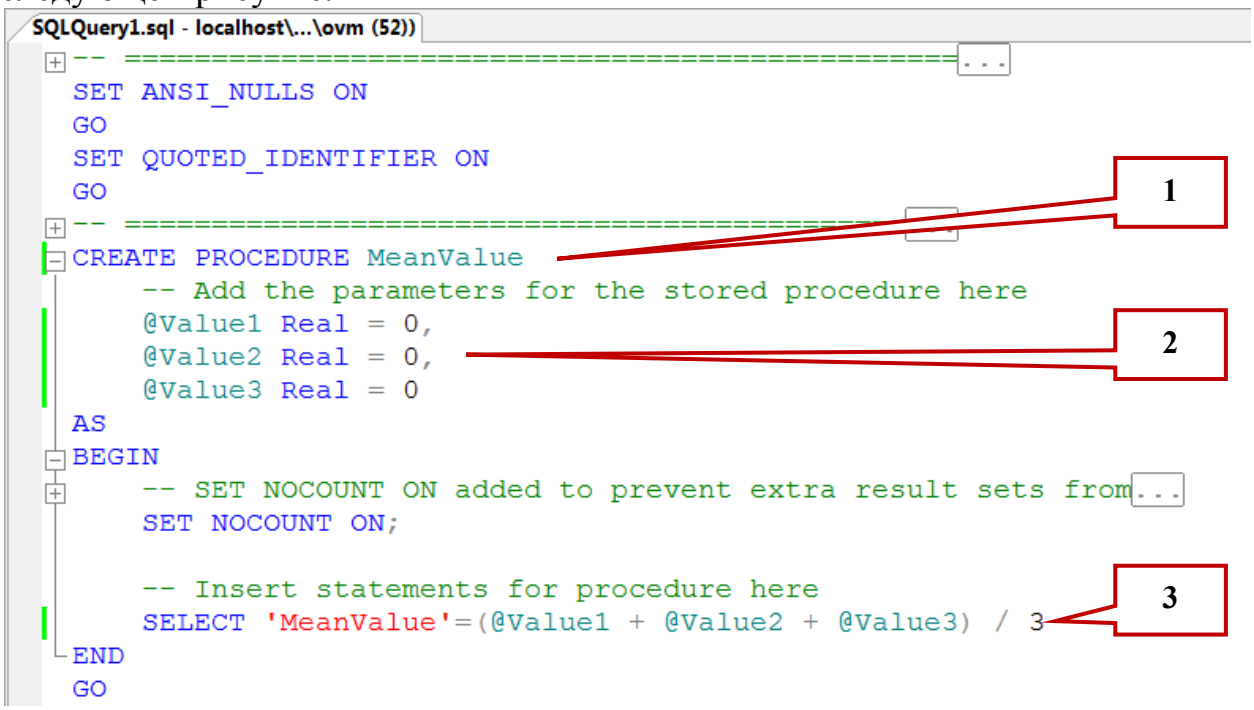
3. Начало тела процедуры, обозначается служебным словом `BEGIN`;

4. Тело процедуры, содержит команды языка SQL;

5. Конец тела процедуры, обозначается служебным словом `END`.

В коде зелёным цветом выделяются комментарии. Они не обрабатываются сервером и выполняют функцию пояснений к коду. Строки комментариев начинаются с подстроки «--». Далее в коде, мы не будем отображать комментарии, они будут свёрнуты. Слева от раздела с комментариями будет стоять знак «+», щёлкнув по которому можно развернуть комментарий.

Наберём код процедуры вычисляющей среднее трёх чисел, как это показано на следующем рисунке.



Рассмотрим код данной процедуры более подробно:

1. `CREATE PROCEDURE MeanValue` определяет имя создаваемой процедуры как «MeanValue»;

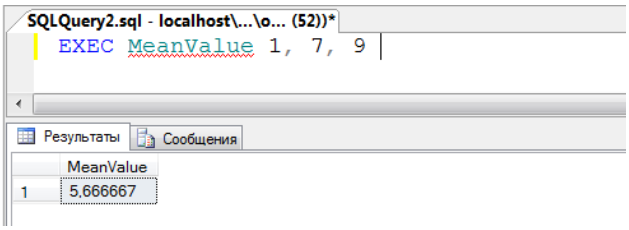
2. `@Value1 Real = 0, @Value2 Real = 0, @Value3 Real = 0` – определение трех параметра процедуры Value1, Value2 и Value3. Тип параметров – Real, значения по умолчанию равны 0;
3. `SELECT 'Mean Value'= (@Value1+@Value2+@Value3)/3` – вычисление среднего и вывод результата с подписью «Среднее значение».

Для создания процедуры выполним ее код, нажав кнопку Выполнить на панели инструментов. В нижней части окна с кодом появится сообщение «Выполнение команд успешно завершено». Закройте окно с кодом, щёлкнув мышью по кнопке закрытия, расположенной в верхнем правом углу окна с кодом процедуры.

Проверим работоспособность созданной хранимой процедуры. Для запуска хранимой процедуры необходимо создать новый пустой запрос, нажав на кнопку Создать запрос на панели инструментов. В появившемся окне с пустым запросом наберите команду

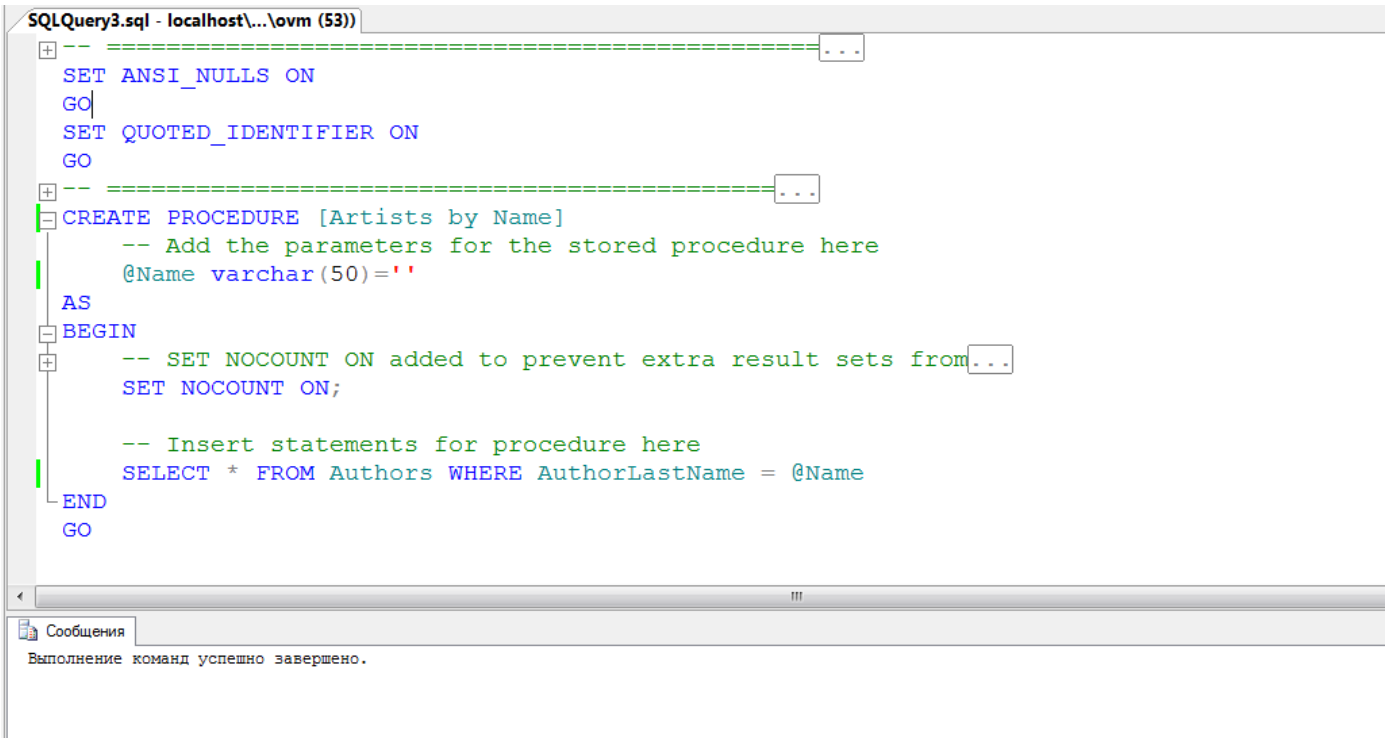
```
EXEC MeanValue 1, 7, 9
```

и нажмите Выполнить кнопку на панели инструментов.

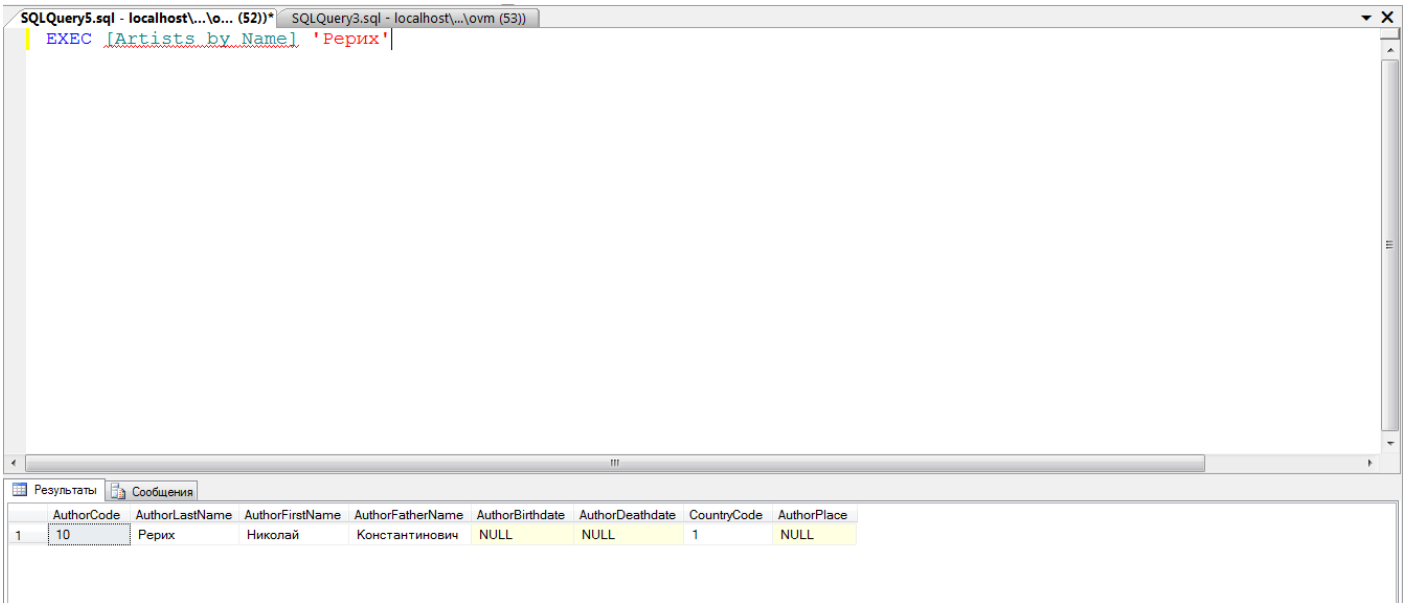


В нижней части окна с кодом появиться результат выполнения новой хранимой процедуры: Среднее значение равно 5,66667.

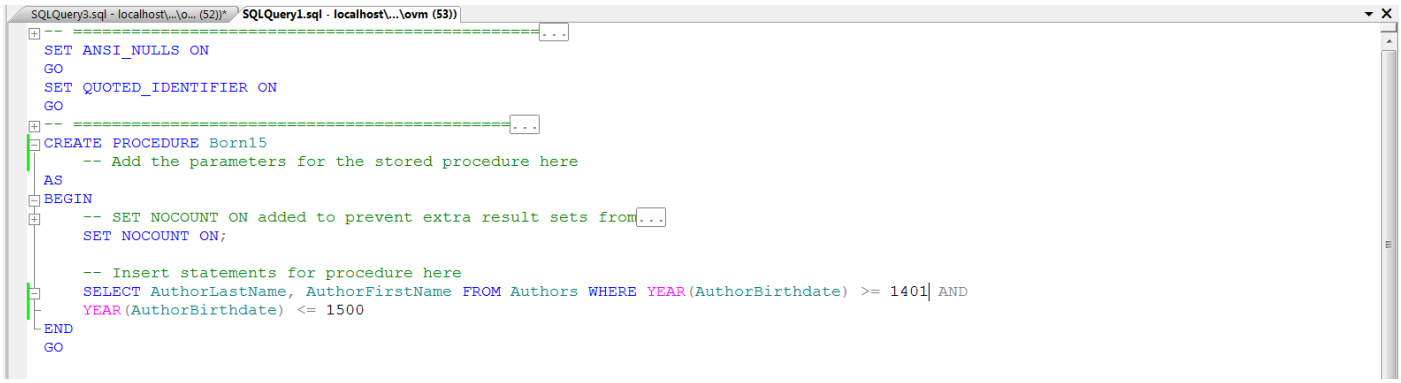
Теперь создадим хранимую процедуру для отбора художников из таблицы «Authors» по их фамилиям. Для этого создадим новую хранимую процедуру по описанию, приведенному выше, и наберем следующий код новой процедуры.



Проверим работоспособность созданной хранимой процедуры. Создадим новый пустой запрос. В появившемся окне с пустым запросом наберем команду `EXEC [Artists by Name] 'Рерих'` и выполним запрос.



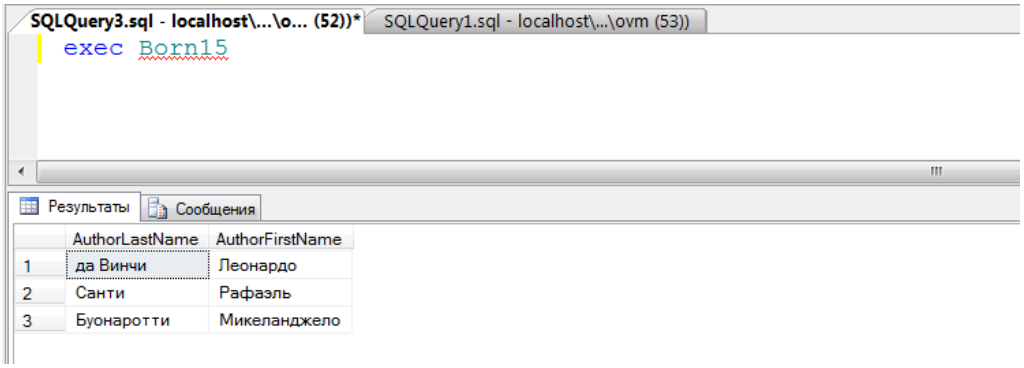
В заключение решим более сложную задачу: отображение имен художников, родившихся в XV веке. Создадим новую хранимую процедуру и наберем следующий код новой процедуры.



Функция `YEAR` имеет следующий синтаксис `YEAR(date)` и возвращает целое число, представляющее год указанной даты `date`.

Полученный с помощью функции `YEAR` год рождения проверяется на вход в диапазон годов, определяющих XV век.

Результат выполнения этой хранимой процедуры приведен на следующем рисунке.



Задание

Создать хранимые процедуры в соответствии с заданием для своего варианта.