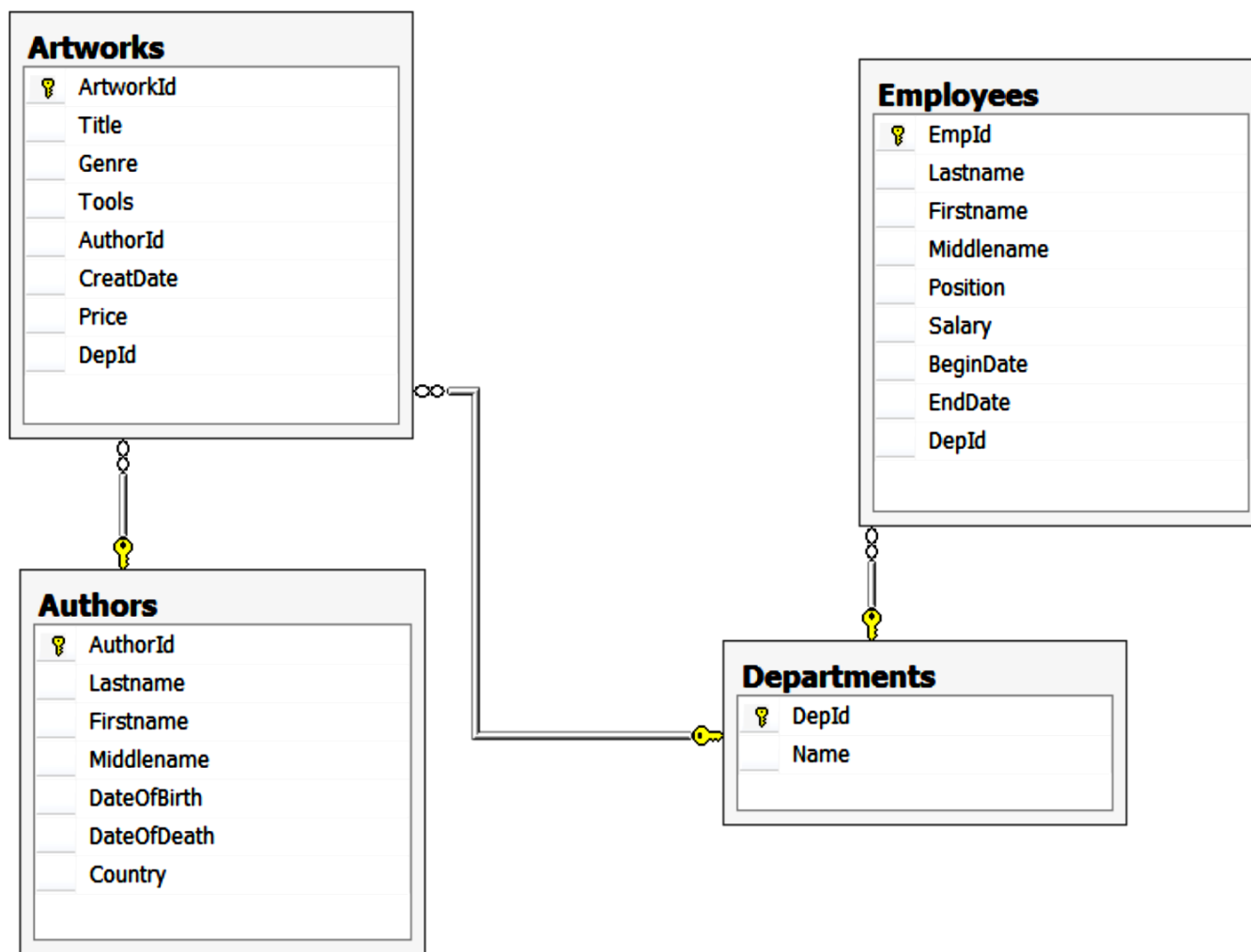


Запросы и фильтры

Пример



Пример (запрос)

- Создать запрос «Отдел кадров» (связывает таблицы «Employees» и «Departments» по полю «DepId»)

```
SELECT E.Lastname, E.Firstname, E.Middlename,  
       E.Position, D.Name, E.BeginDate  
FROM dbo.Employees AS E  
JOIN dbo.Departments AS D  
ON E.DepId = D.DepId;
```

Пример (запрос)

- Создать запрос «Отдел кадров» (связывает таблицы «Employees» и «Departments» по полю «DepId»)

Результаты		Сообщения				
	Lastname	Firstname	Middlename	Position	Name	BeginDate
1	Иванов	Иван	Иванович	Реставратор	Отдел реставрации	2013-01-09
2	Петров	Петр	Петрович	Смотритель	Отдел искусства фотографии	2013-01-09
3	Степанов	Степан	Степанович	Начальник	Отдел гравюры и рисунка	2013-01-09
4	Федорова	Анна	Федоровна	Уборщица	Отдел искусства Античного мира	2013-01-09
5	Потапов	Виктор	Сергеевич	Реставратор	Отдел искусства стран Европы XIX-XX вв.	2013-01-09
6	Сергеев	Николай	Петрович	Экскурсовод	Отдел русского искусства XIX-XX вв.	2013-01-09
7	Коваленко	Мария	Владимировна	Экскурсовод	Отдел искусства фотографии	2013-01-09
8	Верецагин	Вадим	Петрович	Экскурсовод	Отдел гравюры и рисунка	2013-01-09
9	Сидорова	Татьяна	Викторовна	Экскурсовод	Отдел искусства Античного мира	2013-01-09
10	Семенова	Инна	Николаевна	Экскурсовод	Отдел искусства Древнего Востока	2013-01-09

Пример (фильтр)

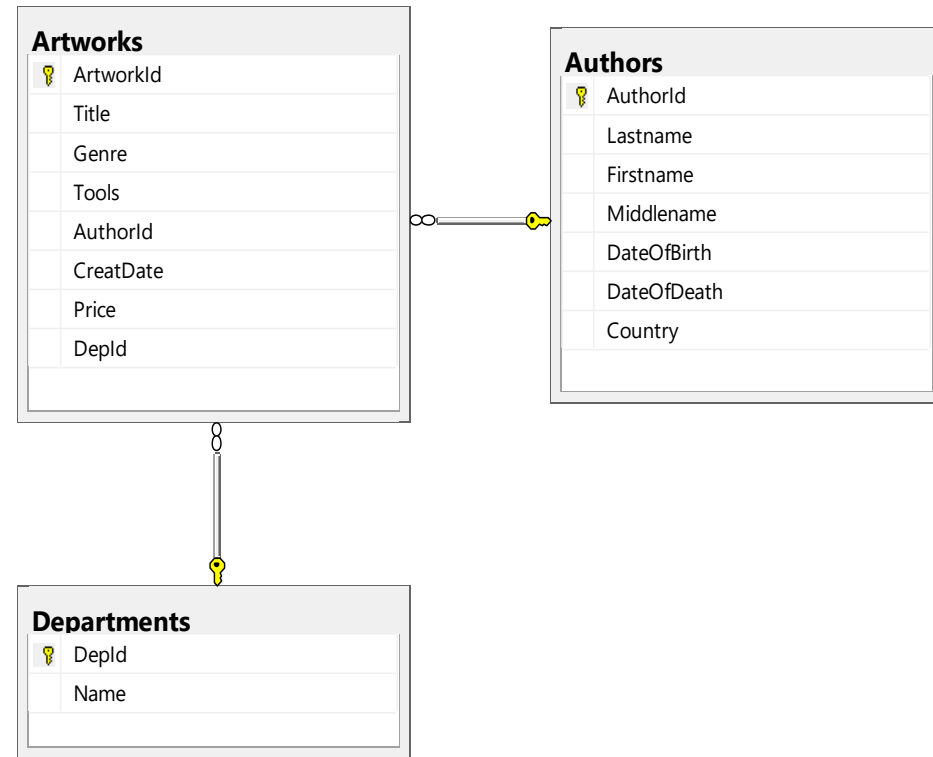
- Создать фильтр для отображения сотрудников отдельных отделов (На основе запроса «Отдел кадров»)

```
SELECT E.Lastname, E.Firstname, E.Middlename,  
       E.Position, D.Name, E.BeginDate  
FROM dbo.Employees AS E  
JOIN dbo.Departments AS D  
ON E.DepId = D.DepId AND  
   D.Name = 'Отдел искусства фотографии';
```

Результаты		Сообщения				
	Lastname	Firstname	Middlename	Position	Name	BeginDate
1	Петров	Петр	Петрович	Смотритель	Отдел искусства фотографии	2013-01-09
2	Коваленко	Мария	Владимировна	Экскурсовод	Отдел искусства фотографии	2013-01-09

Пример (многотабличный запрос)

```
SELECT A.Lastname, A.Firstname, A.Middlename,  
       Ar.Title, Ar.Genre, Ar.Tools, D.Name  
FROM dbo.Authors AS A  
JOIN dbo.Artworks AS Ar  
ON A.AuthorId = Ar.AuthorId  
JOIN dbo.Departments AS D  
ON Ar.DepId = D.DepId;
```



Пример (многотабличный запрос)

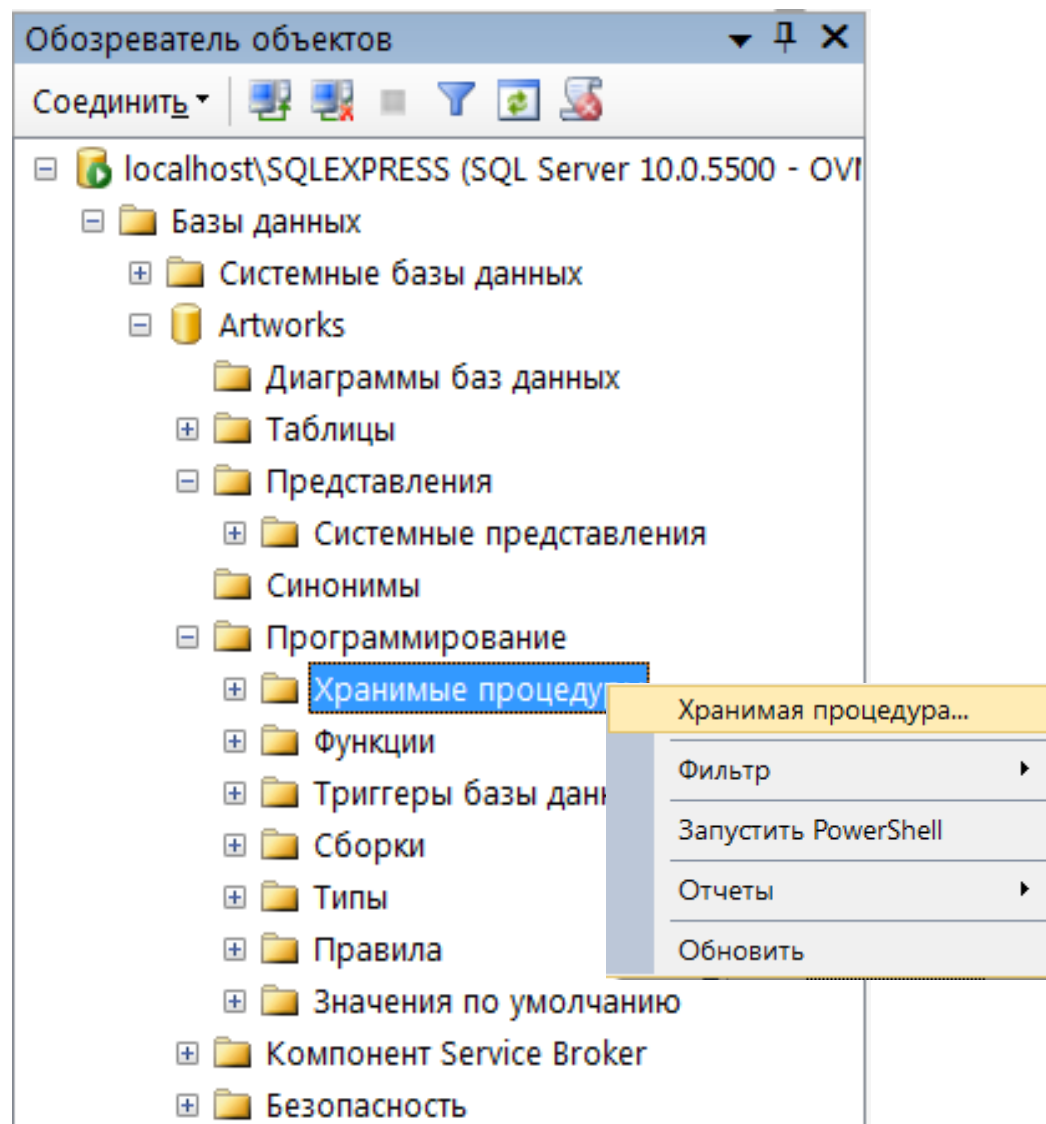
Результаты		Сообщения					
	Lastname	Firstna...	Middlename	Title	Genre	Tools	Name
1	Суриков	Василий	Иванович	Взятие снежного городка	Исторический	Холст, масло	Отдел русского искусства XIX-XX вв.
2	Айвазовский	Иван	Константинович	Девятый вал	Пейзаж	Холст, масло	Отдел русского искусства XIX-XX вв.
3	Васнецов	Виктор	Михайлович	Богатыри	Исторический	Холст, масло	Отдел русского искусства XIX-XX вв.
4	Шишкин	Иван	Иванович	Утро в сосновом лесу	Пейзаж	Холст, масло	Отдел русского искусства XIX-XX вв.
5	Поленов	Василий	Дмитриевич	Московский дворик	Пейзаж	Холст, масло	Отдел русского искусства XIX-XX вв.

Хранимые процедуры

Хранимая процедура

- 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.

Создание хранимой процедуры



Создание хранимой процедуры

```
-- =====
-- Template generated from Template Explorer using:
-- Create Procedure (New Menu).SQL
--
-- Use the Specify Values for Template Parameters
-- command (Ctrl-Shift-M) to fill in the parameter
-- values below.
--
-- This block of comments will not be included in
-- the definition of the procedure.
-- =====

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

-- =====
-- Author:      <Author,,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>
-- =====

CREATE PROCEDURE <Procedure_Name, sysname, ProcedureName>
    -- Add the parameters for the stored procedure here
    <@Param1, sysname, @p1> <Datatype_For_Param1, , int> = <Default_Value_For_Param1, , 0>,
    <@Param2, sysname, @p2> <Datatype_For_Param2, , int> = <Default_Value_For_Param2, , 0>
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for procedure here
    SELECT <@Param1, sysname, @p1>, <@Param2, sysname, @p2>
END
GO
```

1. SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

2. CREATE PROCEDURE <Procedure_Name, sysname, ProcedureName>

3. BEGIN

4. SET NOCOUNT ON;

5. SELECT <@Param1, sysname, @p1>, <@Param2, sysname, @p2>

Создание хранимой процедуры

```
+-- =====  
SET ANSI_NULLS ON  
GO  
SET QUOTED_IDENTIFIER ON  
GO  
-- =====  
-- Author:      <Author,,Name>  
-- Create date: <Create Date,,>  
-- Description: <Description,,>  
-- =====  
CREATE PROCEDURE GetPictures  
    -- Add the parameters for the stored procedure here  
AS  
BEGIN  
    -- SET NOCOUNT ON added to prevent extra result sets from  
    -- interfering with SELECT statements.  
    SET NOCOUNT ON;  
  
    -- Insert statements for procedure here  
    SELECT A.Lastname, A.Firstname, A.Middlename, Ar.Title, Ar.Genre, Ar.Tools, D.Name  
    FROM dbo.Authors AS A  
    JOIN dbo.Artworks AS Ar  
    ON A.AuthorId = Ar.AuthorId  
    JOIN dbo.Departments AS D  
    ON Ar.DepId = D.DepId;  
END  
GO
```

Выполнение хранимой процедуры

```
exec dbo.GetPictures;
```

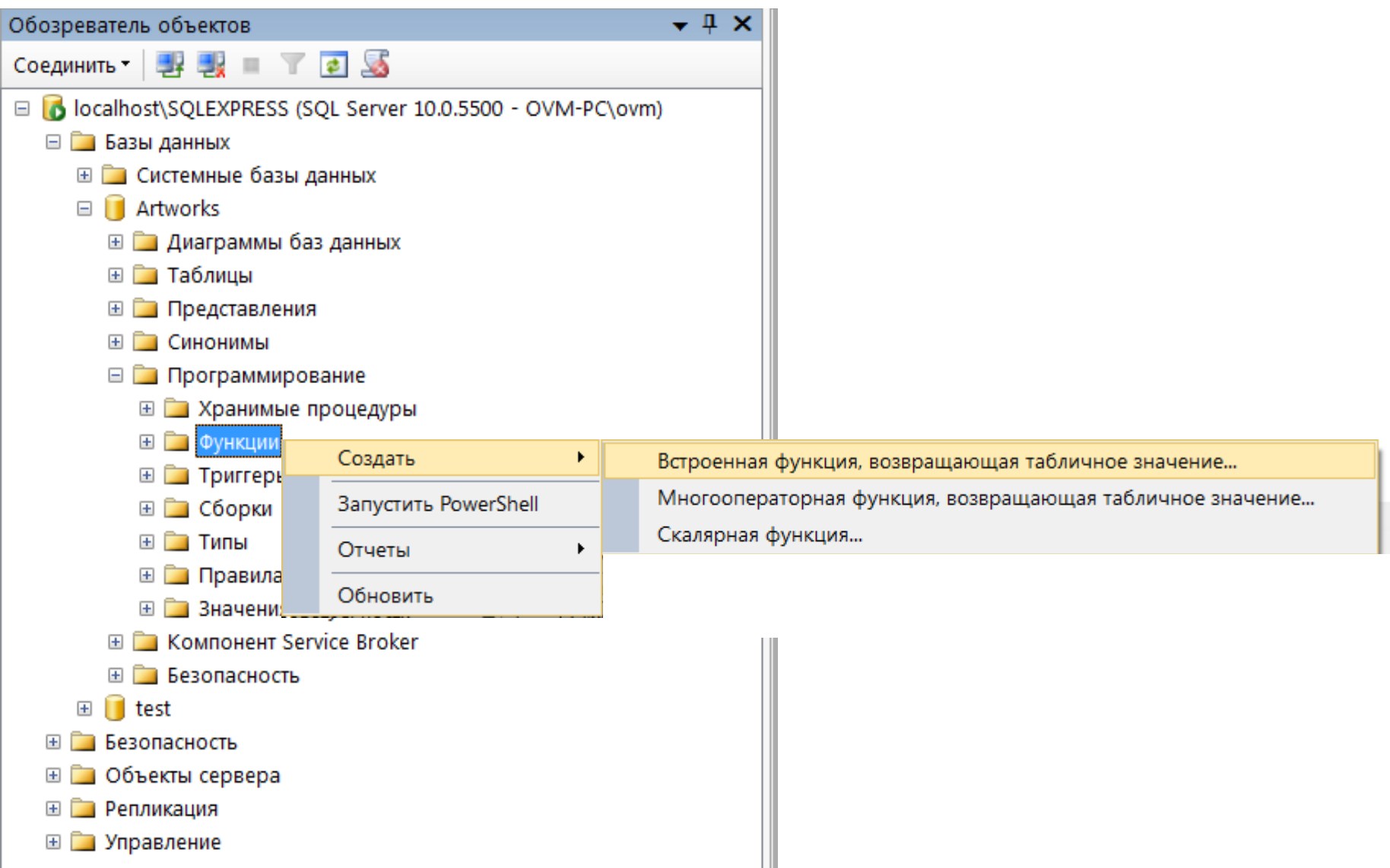
100 % <

Результаты Сообщения

	Lastname	Firstna...	Middlename	Title	Genre	Tools	Name
1	Суриков	Василий	Иванович	Взятие снежного городка	Исторический	Холст, масло	Отдел русского искусства XIX-XX вв.
2	Айвазовский	Иван	Константинович	Девятый вал	Пейзаж	Холст, масло	Отдел русского искусства XIX-XX вв.
3	Васнецов	Виктор	Михайлович	Богатыри	Исторический	Холст, масло	Отдел русского искусства XIX-XX вв.
4	Шишкин	Иван	Иванович	Утро в сосновом лесу	Пейзаж	Холст, масло	Отдел русского искусства XIX-XX вв.
5	Поленов	Василий	Дмитриевич	Московский дворик	Пейзаж	Холст, масло	Отдел русского искусства XIX-XX вв.

Функции

Создание функции



Создание функции

```
-- =====
-- Template generated from Template Explorer using:
-- Create Inline Function (New Menu).SQL
--
-- Use the Specify Values for Template Parameters
-- command (Ctrl-Shift-M) to fill in the parameter
-- values below.
--
-- This block of comments will not be included in
-- the definition of the function.
-- =====

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

-- =====
-- Author:      <Author,,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>
-- =====

CREATE FUNCTION <Inline_Function_Name, sysname, FunctionName>
(
    -- Add the parameters for the function here
    <@param1, sysname, @p1> <Data_Type_For_Param1, , int>,
    <@param2, sysname, @p2> <Data_Type_For_Param2, , char>
)
RETURNS TABLE
AS
RETURN
(
    -- Add the SELECT statement with parameter references here
    SELECT 0
)
GO
```

1

2

3

4

Создание функции

```
+ -- =====  
SET ANSI_NULLS ON  
GO  
SET QUOTED_IDENTIFIER ON  
GO  
- -- =====  
-- Author:      <Author,,Name>  
-- Create date: <Create Date,,>  
-- Description: <Description,,>  
-- =====  
- CREATE FUNCTION GetEmps  
(  
    -- Add the parameters for the function here  
    @DepName varchar(50)  
)  
RETURNS TABLE  
AS  
RETURN  
(  
    -- Add the SELECT statement with parameter references here  
    SELECT E.Lastname, E.Firstname, E.Middlename,  
           E.Position, D.Name, E.BeginDate  
    FROM dbo.Employees AS E  
    JOIN dbo.Departments AS D  
    ON E.DepId = D.DepId AND  
       D.Name = @DepName  
)  
GO
```

Выполнение функции

```
select * from GetEmps ('Отдел искусства фотографии');
```

100 % <



Результаты



Сообщения

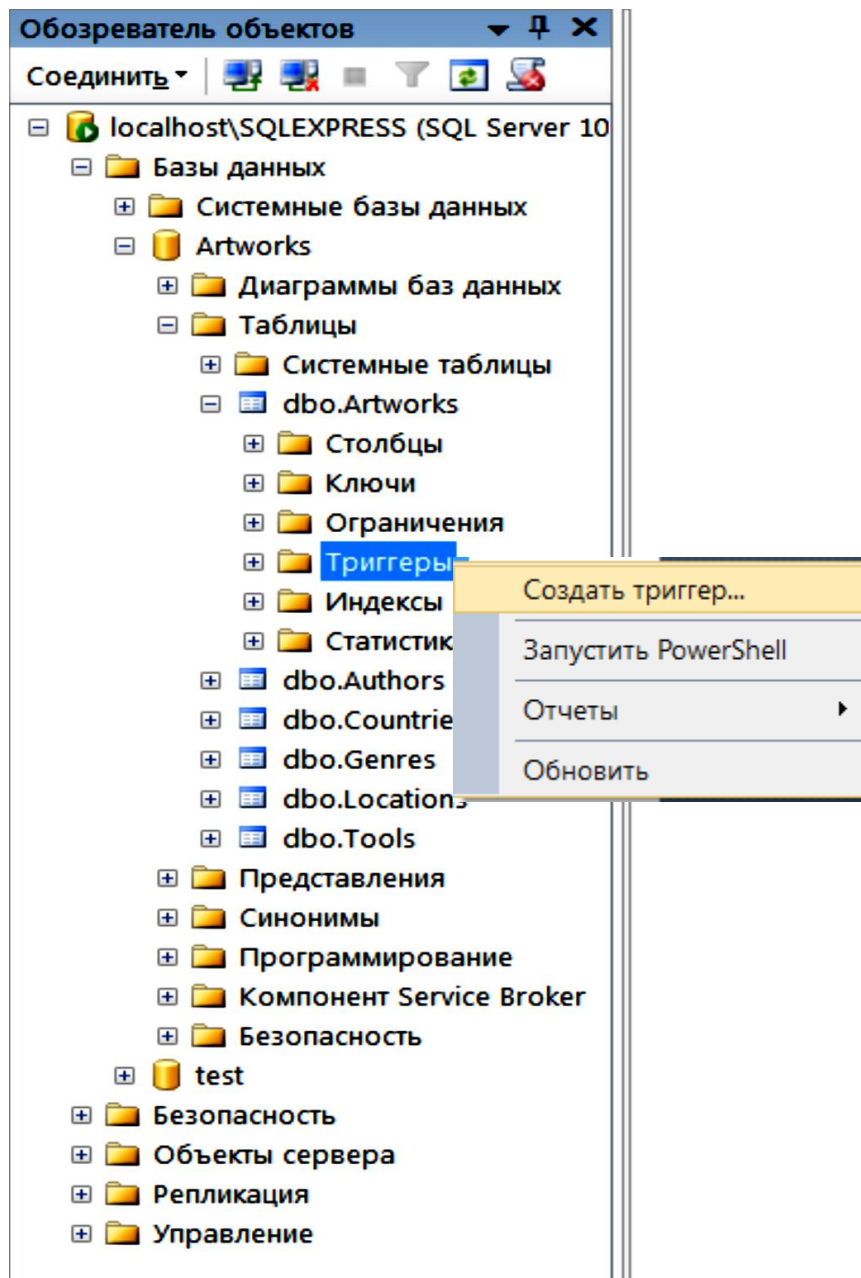
	Lastname	Firstname	Middlename	Position	Name	BeginDate
1	Петров	Петр	Петрович	Смотритель	Отдел искусства фотографии	2013-09-01 00:00:00.000
2	Кузнецова	Наталья	Ивановна	Директор	Отдел искусства фотографии	2013-09-01 00:00:00.000
3	Коваленко	Мария	Владимировна	Экскурсовод	Отдел искусства фотографии	2013-09-01 00:00:00.000

Триггеры

Триггер

- Триггер – это сочетание хранимой в базе данных процедуры и события, которое заставляет ее выполняться
- Такими событиями могут быть: ввод новой строки таблицы, изменение значений одного или нескольких ее столбцов и (или) удаление строки таблицы
- Подобно процедурам и функциям, триггеры хранятся как автономные объекты в базе данных
- Триггеры создаются отдельно для каждой таблицы и располагаются в обозревателе объектов в папке «Триггеры»

Создание триггера



Создание триггера

```
+|-----|
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

-- =====
-- Author:      <Author,,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>
-- =====

CREATE TRIGGER <Schema_Name, sysname, Schema_Name>.<Trigger_Name, sysname, Trigger_Name>
ON <Schema_Name, sysname, Schema_Name>.<Table_Name, sysname, Table_Name>
AFTER <Data_Modification_Statements, , INSERT,DELETE,UPDATE>
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for trigger here

END
GO
```

1

2

3

4

Создание триггера на добавление данных в таблицу

```
+ -- =====  
SET ANSI_NULLS ON  
GO  
SET QUOTED_IDENTIFIER ON  
GO  
  
-- =====  
-- Author:      <Author,,Name>  
-- Create date: <Create Date,,>  
-- Description: <Description,,>  
-- =====  
  
CREATE TRIGGER dbo.DepsInsert  
ON  dbo.Departments  
AFTER INSERT  
AS  
BEGIN  
    -- SET NOCOUNT ON added to prevent extra result sets from  
    -- interfering with SELECT statements.  
    SET NOCOUNT ON;  
  
    -- Insert statements for trigger here  
    PRINT ('Запись добавлена!');  
END  
GO
```

Выполнение триггера

```
insert into dbo.Departments  
values ('Отдел скульптуры');
```

100 %



Сообщения

Запись добавлена!

Создание триггера на удаление данных из таблицы

```
+ -- =====  
SET ANSI_NULLS ON  
GO  
SET QUOTED_IDENTIFIER ON  
GO  
- -- =====  
-- Author:      <Author,,Name>  
-- Create date: <Create Date,,>  
-- Description: <Description,,>  
-- =====  
- CREATE TRIGGER dbo.DepsDelete  
  ON  dbo.Departments  
  AFTER DELETE  
  AS  
- BEGIN  
-   -- SET NOCOUNT ON added to prevent extra result sets from  
  -- interfering with SELECT statements.  
  SET NOCOUNT ON;  
  
  -- Insert statements for trigger here  
  PRINT('Запись удалена!');  
END  
GO
```

Выполнение триггера

```
delete from dbo.Departments  
where dbo.Departments.Name = 'Отдел скульптуры';
```

100 %



Сообщения

Запись удалена!