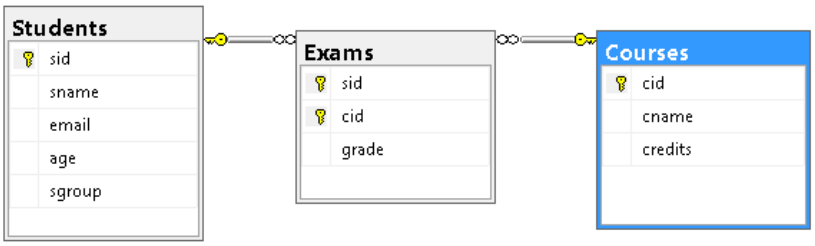
# ***Seminar 3 – Documentations – Examples***

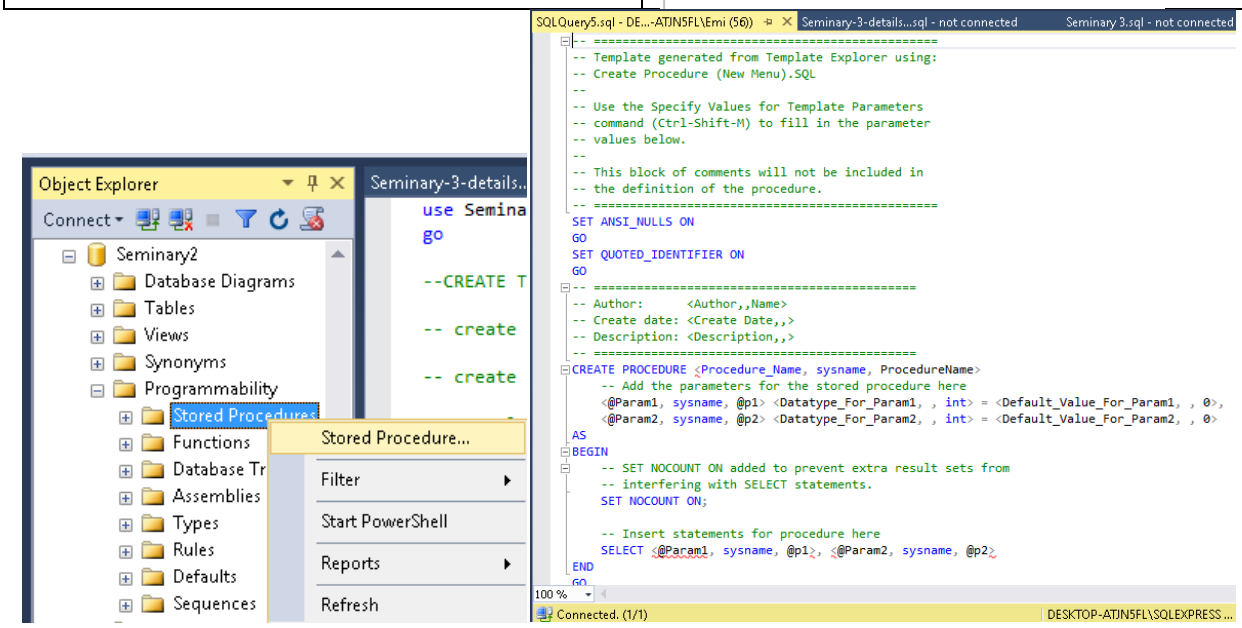
Stored procedures:




```
select * from Students
```

Results | Messages

| | sid | sname | email | age | sgroup |
|---|---|---|---|---|---|
| 1 | 1 | Maria | m@yahoo.com | 21 | 926 |
| 2 | 1234 | Ada | a@cs.ro | 20 | 921 |
| 3 | 1235 | Razvan | r@cs.ro | 21 | 921 |
| 4 | 1236 | Monica | m@cs.ro | 20 | 922 |
| 5 | 1237 | Paula | p@cs.ro | 20 | 924 |
| 6 | 1834 | Alina | al@cs.ro | 21 | 921 |

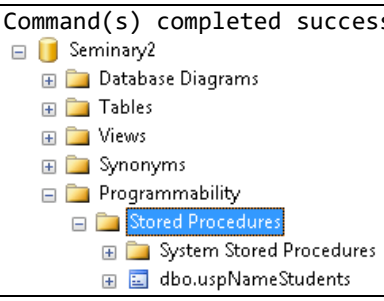



FIRST – the stored procedure is CREATED

```
CREATE PROCEDURE uspNameStudents
AS
        SELECT sname FROM Students
GO

-- DROP PROCEDURE uspNameStudents
```

```
Command(s) completed successfully.
```



SECOND – the stored procedure is EXECUTED

```
EXEC uspNameStudents
EXECUTE uspNameStudents
uspNameStudents
```

Results | Messages

| | sname |
|---|---|
| 1 | Maria |
| 2 | Ada |
| 3 | Razvan |
| 4 | Monica |
| 5 | Paula |
| 6 | Alina |

Raiserror - generate an error message and initiates the error processing to the session

```
RAISERROR (N'This is message %s %d.', -- Message text.
           10, -- Severity,
           1, -- State,
           N'number', -- First argument.
           5); -- Second argument.
-- The message text returned is: This is message number 5.
GO
```

```
This is message number
5.
```

```
RAISERROR (N'<\<%*.*s>>', -- Message text.
           10, -- Severity,
           1, -- State,
           7, -- First argument used for width.
           3, -- Second argument used for precision.
           N'abcde'); -- Third argument supplies the string.
-- The message text returned is: <<    abc>>.
GO
```

```
<\<    abc>>
```

```
RAISERROR (N'<\<%7.3s>>', -- Message text.
           10, -- Severity,
           1, -- State,
           N'abcde'); -- First argument supplies the string.
-- The message text returned is: <<    abc>>.
GO
```

**OR**

```
DECLARE @StringVariable NVARCHAR(50);
SET @StringVariable = N'<\<%7.3s>>';

RAISERROR (@StringVariable, -- Message text.
           10, -- Severity,
           1, -- State,
           N'abcde'); -- First argument supplies the string.
-- The message text returned is: <<    abc>>.
GO
```

```
<\<    abc>>
```

Global Variables

```
-- - @@ERROR – the error number for the last
executed T-SQL statement; 0 - no error occurred
IF @@ERROR <> 0
    PRINT  'Your error message';
```

```
Command(s) completed successfully.
```

```sql
-- @@IDENTITY - the last inserted IDENTITY value
SELECT @@IDENTITY AS 'Identity';
```

Results | Messages

| | Identity |
|---|---|
| 1 | NULL |

```sql
create table test(
tid int primary key identity,
tname varchar(50),
description varchar(50),
code int)

select * from test

INSERT INTO test(tid, tname, description, code)
Values (1, 'test1', 'decription 1', 8)
go
select @@IDENTITY As 'Identity'
```

```
(0 row(s) affected)
Msg 544, Level 16, State 1, Line 112
Cannot insert explicit value for identity column
in table 'test' when IDENTITY_INSERT is set to
OFF.

(1 row(s) affected)
```

```sql
INSERT INTO test(tname, description, code) Values
('test2', 'decription 2', 88)
INSERT INTO test(tname, description, code) Values
('test3', 'decription 3', 888)
go
INSERT INTO test(tname, description, code) Values
('test4', 'decription 4', 8888)

select @@IDENTITY As 'Identity'
```

Results | Messages

| | Identity |
|---|---|
| 1 | 3 |

```sql
-- @@ROWCOUNT - the number of rows affected by the
last statement
IF @@ROWCOUNT = 0
 PRINT 'Warning: No rows were updated';

 select @@ROWCOUNT
```

Results | Messages

| | (No column name) |
|---|---|
| 1 | 1 |

(1 row(s) affected)

```sql
-- @@SERVERNAME - the name of the local server on
which SQL Server is running
 SELECT @@SERVERNAME AS 'Server Name'
```

Results | Messages

| | Server Name |
|---|---|
| 1 | DESKTOP-ATJN5FL\SQLEXPRESS |

```sql
-- @@SPID - the session id for the current user
process
 SELECT @@SPID AS 'ID', SYSTEM_USER AS 'Login
Name', USER AS 'User Name'
```

Results | Messages

| | ID | Login Name | User Name |
|---|---|---|---|
| 1 | 54 | DESKTOP-ATJN5FL\Emi | dbo |

```sql
-- @@VERSION - system and build information
 SELECT @@VERSION AS 'SQL Server Version'
```

Results | Messages

| | SQL Server Version |
|---|---|
| 1 | Microsoft SQL Server 2016 (RTM) - 13.0.1601.5 (X64) Apr 29 2016 23:23:58 Copyright (c) Microsoft Corporation Express Edition (64-bit) o... |

Select INTO – the result will be displayed in another table

```sql
-- SELECT INTO
select * from Students

IF OBJECT_ID ('dbo.StudentsCopy', 'U') IS NOT NULL
DROP TABLE dbo.StudentsCopy;

select sname, age
INTO dbo.StudentsCopy
from Students
where email like'%@cs.ro'
```

Results | Messages

| | sid | sname | email | age | sgroup |
|---|---|---|---|---|---|
| 1 | 1 | Maria | m@yahoo.com | 21 | 926 |
| 2 | 1234 | Ada | a@cs.ro | 20 | 921 |
| 3 | 1235 | Razvan | r@cs.ro | 21 | 921 |
| 4 | 1236 | Monica | m@cs.ro | 20 | 922 |
| 5 | 1237 | Paula | p@cs.ro | 20 | 924 |
| 6 | 1834 | Alina | al@cs.ro | 21 | 921 |

```sql
select * from StudentsCopy
```

| | sname | age |
|---|---|---|
| 1 | Ada | 20 |
| 2 | Razvan | 21 |
| 3 | Monica | 20 |
| 4 | Paula | 20 |
| 5 | Alina | 21 |

## The OUTPUT Clause

```sql
create table ModifiedCourses(
cid varchar(10) primary key,
cname varchar(50) not null,
credits int,
dateM date,
s nvarchar(128))

Select * from Courses
select * from ModifiedCourses
```

Results | Messages

| | cid | cname | credits |
|---|---|---|---|
| 1 | Alg1 | Algorithms 1 | 7 |
| 2 | DB1 | Databases 1 | 6 |
| 3 | DB2 | Databases 2 | 6 |
| 4 | MAP1 | MAP 1 | 5 |
| 5 | MAP2 | MAP 2 | 4 |

| | cid | cname | credits | dateM | s |
|---|---|---|---|---|---|

```sql
UPDATE Courses
SET cname = 'Database Management Systems'
OUTPUT inserted.cid, deleted.cname, inserted.credits, GETDATE(), SUSER_SNAME()  -- Is the
optional login security identification number
INTO ModifiedCourses
WHERE cid = 'DB2'
-- (1 row(s) affected)
```

```sql
select * from Courses
select * from ModifiedCourses
```

Results | Messages

| | cid | cname | credits |
|---|---|---|---|
| 1 | Alg1 | Algorithms 1 | 7 |
| 2 | DB1 | Databases 1 | 6 |
| 3 | DB2 | Database Management Systems | 6 |
| 4 | MAP1 | MAP 1 | 5 |
| 5 | MAP2 | MAP 2 | 4 |

| | cid | cname | credits | dateM | s |
|---|---|---|---|---|---|
| 1 | DB2 | Databases 2 | 6 | 2019-11-05 | DESKTOP-ATJN5FL\Emi |

## Cursors

```sql
DECLARE @sid INT, @sname VARCHAR(50)
DECLARE CursorStudents CURSOR FOR
SELECT sid, sname
FROM Students
OPEN CursorStudents
FETCH CursorStudents
INTO @sid, @sname
WHILE @@FETCH_STATUS = 0
BEGIN
	--code to proccess @sid, @sname
	print 'The record with the sid=' + CAST(@sid AS
varchar(50)) + ' has the name ' + CONVERT(varchar(50), @sname)
	FETCH CursorStudents
	INTO @sid, @sname
END
CLOSE CursorStudents
DEALLOCATE CursorStudents
```

Messages

```
The record with the sid=1 has the name Maria
The record with the sid=1234 has the name Ada
The record with the sid=1235 has the name Razvan
The record with the sid=1236 has the name Monica
The record with the sid=1237 has the name Paula
The record with the sid=1834 has the name Alina
```