**Stored Procedures Assignment**

1. Create a procedure that uses a cursor to generate the below report, which contains the first and last name of the customer that bought a Hrd. cloth, 1/4-in., 2x50. Use the tables: sales\_co.CUSTOMER, sales\_co.product, sales\_co.line, and sales\_co.invoice i

Query:

use db2; go declare

@first nvarchar(30),

@last nvarchar(30), @product1 nvarchar(30);

set @product1 = 'Hrd. cloth, 1/4-in., 2x50';

exec proc1 @first1 output, @last1 output, @product1;

print '=============================================';

print @first1 + ' ' + @last1 +Bought'+ @product1;

print'==================================';

Format you output as show on example document.

Code:

CREATE PROCEDURE proc1

(

@first nvarchar(30) output,

@last nvarchar(30) output,

@product nvarchar(30)

)

AS

DECLARE db\_cursor CURSOR FOR

SELECT CUS\_FNAME, CUS\_LNAME, P\_DESCRIPT FROM [sales\_co].[LINE] a

JOIN [sales\_co].[INVOICE] c ON c.INV\_decimal = a.INV\_decimal

JOIN [sales\_co].[PRODUCT] b ON a.P\_CODE = b.P\_CODE

JOIN [sales\_co].[CUSTOMER] d ON c.CUS\_CODE = d.CUS\_CODE

WHERE P\_DESCRIPT = @product;

OPEN db\_cursor

FETCH NEXT FROM db\_cursor INTO @first, @last, @product

WHILE @@FETCH\_STATUS = 0

BEGIN

SET @first = @first;

SET @last = @last;

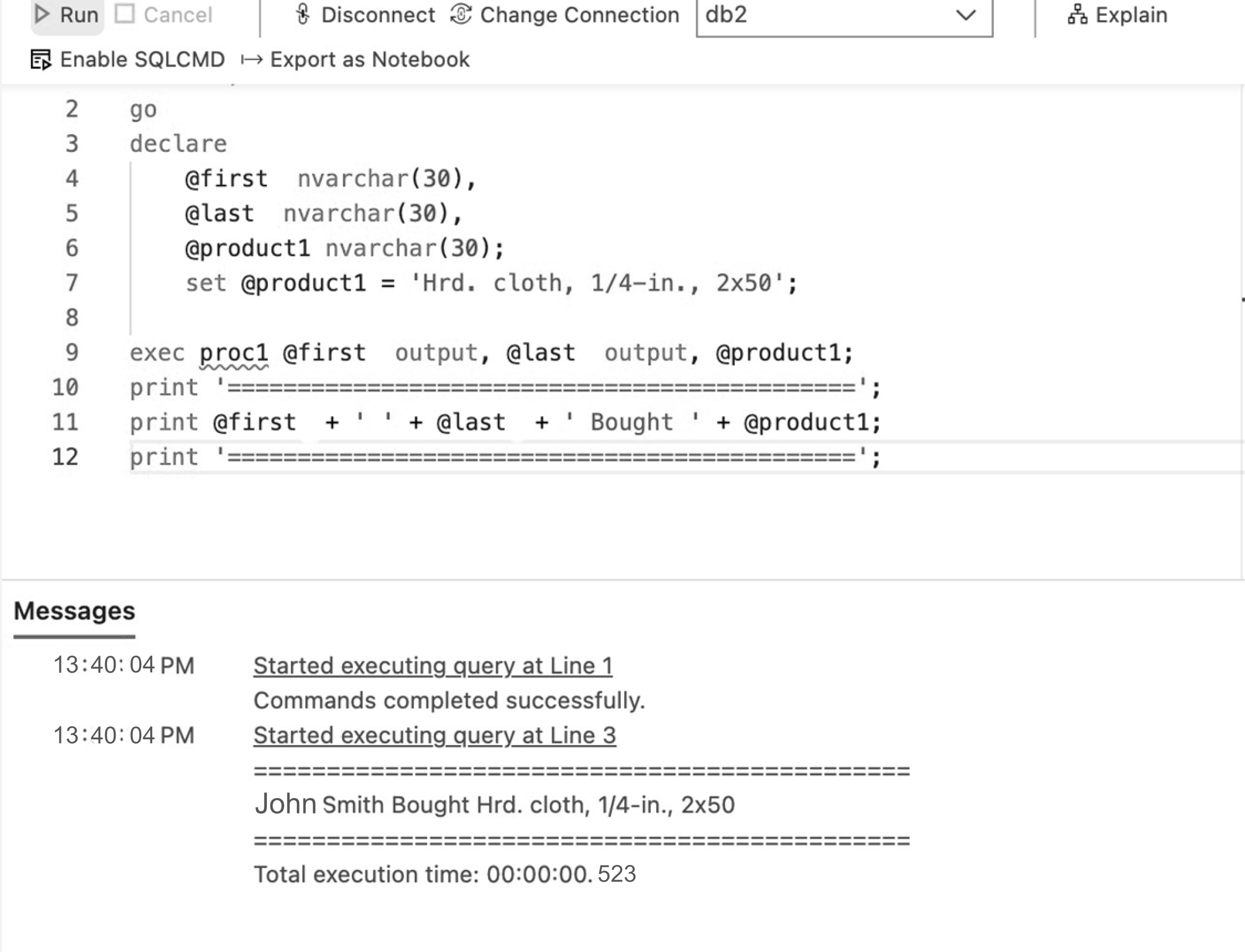
FETCH NEXT FROM db\_cursor INTO @first, @last, @product

END

CLOSE db\_cursor

DEALLOCATE db\_cursor

Results:



2. Create a procedure that removes your name from the table db2.college.students. If your name is not in the database use 'Amber' 'Howard'. If the your name or 'Amber' 'Howard' is not in the database, print out the name

+ ' is not in the database anymore'

Query:

AS

BEGIN

IF EXISTS (SELECT \* FROM db2.College.Students WHERE LastName = @last1)

BEGIN

SET @last1 = 'Rotger'

IF EXISTS (SELECT \* FROM db2.College.Students WHERE LastName = @last1)

BEGIN

DELETE FROM db2.College.Students WHERE LastName = @last1

PRINT @last1 + ' has been deleted';

END

ELSE

BEGIN

PRINT @last1 + ' is not in the database anymore.'

END

END

ELSE

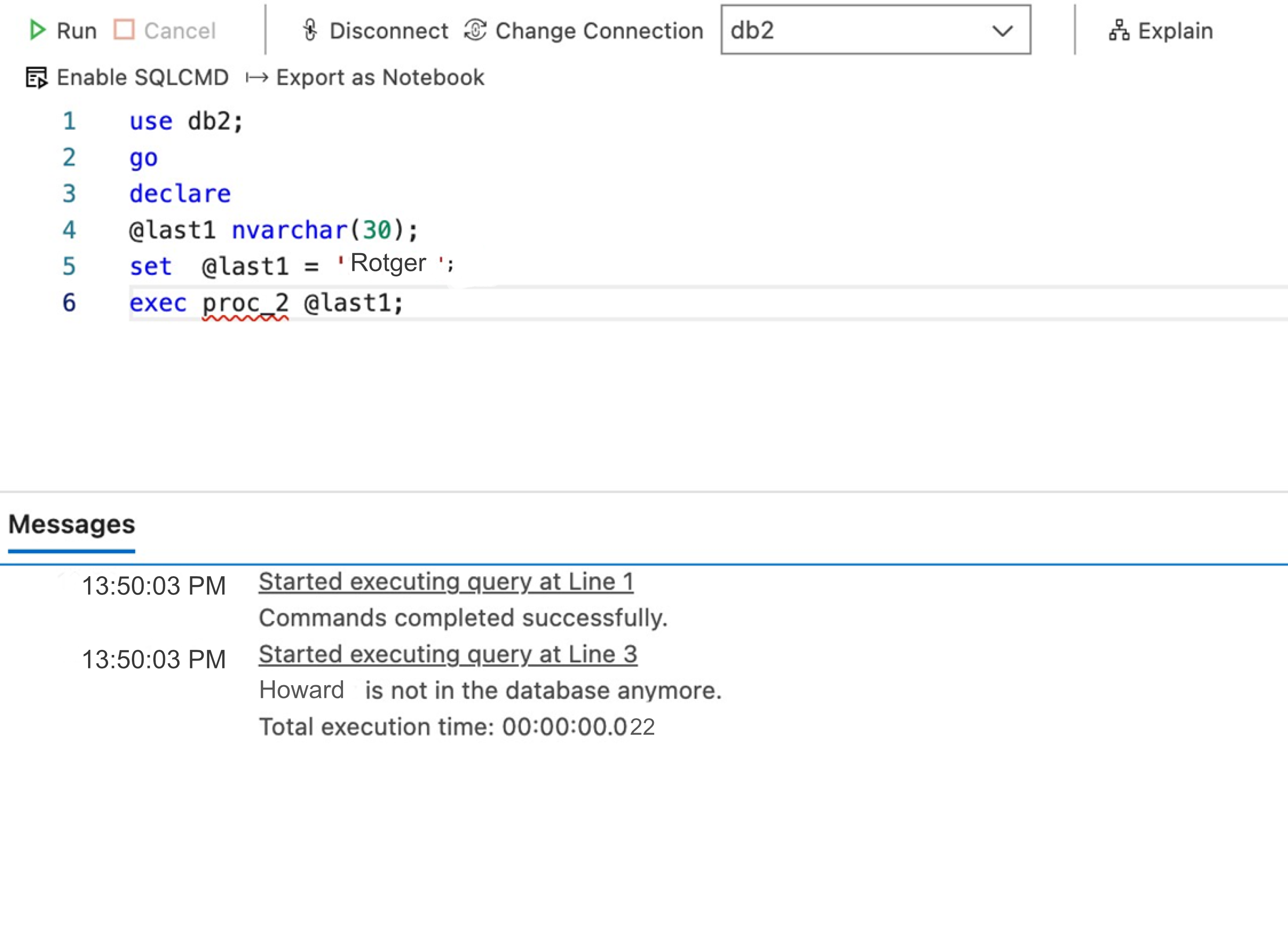
BEGIN

PRINT @last1 + ' is not in the database anymore.'

END

END

Results:



3. Use the db2 database and the MM schema. Create 2 procedures named MOVIE\_RENTAL & MOVIE\_RETURN. One procedure movie\_rental has 4 parameters, 2 input (passed to procedure) and two output (passed out of the procedure). This procedure will take in the title of the movie and the last name of the person renting the movie. It will then populate the output variables with the movie\_id and the member\_id. The second procedure movie\_return has 3 parameters, 2 input one output. movie\_return will be executed by the script in the assignment receiving the title of the movie and the last name of the person renting the movie, calling the procedure movie\_rental at the beginning of the procedure. Using the output of the procedure move\_rental it will determine if the member has rented the movie, if they have rented the movie print out that the member has returned the movie and insert today's date for the check\_in date

Query::

GO

CREATE PROC MOVIE\_RENTAL

(

@LAST\_NAME VARCHAR(50),

@MOVIE\_TITLE VARCHAR(50),

@MOVIE\_ID VARCHAR(50) OUTPUT,

@MEMBER\_ID VARCHAR(50) OUTPUT

)

AS

SELECT @MEMBER\_ID = member\_id

FROM db2.MM.member

WHERE last = @LAST\_NAME

SELECT @MOVIE\_ID = movie\_id

FROM db2.MM.movie

WHERE movie\_title = @MOVIE\_TITLE

GO

CREATE PROC MOVIE\_RETURN

(

@LAST\_NAME VARCHAR(50),

@MOVIE\_TITLE VARCHAR(50),

@MESSAGE VARCHAR(200) OUTPUT

)

AS

BEGIN

DECLARE @MOVIE\_ID VARCHAR(50), @MEMBER\_ID VARCHAR(50)

EXEC MOVIE\_RENTAL @LAST\_NAME, @MOVIE\_TITLE, @MOVIE\_ID OUTPUT, @MEMBER\_ID OUTPUT

SET @MESSAGE = @LAST\_NAME + ' did not return the movie ' + @MOVIE\_TITLE

END

Results:

