--DECLARE @num AS INT

--SET @num=250

--PRINT @num

--DECLARE @totalPages AS INT

--SELECT @totalPages=SUM(Books.Pages)

--FROM Books

--PRINT 'Total pages : '+CAST(@totalPages AS NVARCHAR(30))

--PRINT 'Hello World'

--DECLARE @sum1 AS INT = 30,@sum2 AS INT = 30

--IF @sum1=@sum2

-- BEGIN

-- PRINT 'Equality'

-- END

--ELSE IF @sum1>@sum2

-- BEGIN

-- PRINT 'sum1 is greater than sum2'

-- END

--ELSE

-- BEGIN

-- PRINT 'sum1 is less than sum2'

-- END

--DECLARE @count1 AS INT , @firstAuthorId AS INT = 2

--SELECT @count1=COUNT(\*)

--FROM Books AS B

--WHERE B.Id\_Author=@firstAuthorId

--DECLARE @count2 AS INT, @secondAuthorId AS INT = 10

--SELECT @count2=COUNT(\*)

--FROM Books AS B

--WHERE B.Id\_Author=@secondAuthorId

--DECLARE @name AS NVARCHAR(30)

--IF @count1>@count2

-- BEGIN

-- SELECT @name=A.FirstNAme

-- FROM Authors AS A

-- WHERE A.Id=@firstAuthorId

-- PRINT @name + ' has more books'

-- END

--ELSE IF @count2>@count1

-- BEGIN

-- SELECT @name=A.FirstNAme

-- FROM Authors AS A

-- WHERE A.Id=@secondAuthorId

-- PRINT @name + ' has more books'

-- END

--DECLARE @bestAuthor AS NVARCHAR(30)

--SELECT TOP(1) @bestAuthor=A.FirstName

--FROM Books AS B

--INNER JOIN Authors AS A

--ON B.Id\_Author=A.Id

--ORDER BY B.Pages DESC

--PRINT 'Best Author is '+@bestAuthor

--DECLARE @i AS INT = 0

--WHILE @i<10

-- BEGIN

-- PRINT @i

-- SET @i+=1

-- END

--CREATE FUNCTION FunctionName(@var INT)

--RETURNS INT

--AS

-- BEGIN

-- END

CREATE FUNCTION FunctionName(@var INT)

RETURNS INT

AS

BEGIN

DECLARE @bestAuthor AS NVARCHAR(30)

SELECT TOP(1) @bestAuthor=A.FirstName

FROM Books AS B

INNER JOIN Authors AS A

ON B.Id\_Author=A.Id

ORDER BY B.Pages DESC

RETURN @bestAuthor

END

--CREATE FUNCTION GetBestAuthorName()

--RETURNS NVARCHAR(50)

--AS

--BEGIN

--DECLARE @bestAuthor AS NVARCHAR(30)

--SELECT TOP(1) @bestAuthor=A.FirstName

--FROM Books AS B

--INNER JOIN Authors AS A

--ON B.Id\_Author=A.Id

--ORDER BY B.Pages DESC

--RETURN @bestAuthor

--END

--declare @name AS NVARCHAR(50)

--EXECUTE @name= GetBestAuthorName

--print @name

--CREATE FUNCTION GetMaximumPageByAuthorId

--(@author\_id INT)

--RETURNS INT

--AS

--BEGIN

-- DECLARE @maximum INT = 0

--SELECT @maximum=MAX(B.Pages)

--FROM Books AS B

--WHERE B.Id\_Author=@author\_id

--RETURN @maximum

--END

--declare @max int = 0

--execute @max=GetMaximumPageByAuthorId 2

--print @max

--CREATE FUNCTION ShowBooksByAuthorId

--(@author\_id AS INT)

--RETURNS TABLE

--AS

--RETURN(

--SELECT A.FirstName,B.Name,B.Pages FROM Books AS B

--INNER JOIN Authors AS A

--ON B.Id\_Author=A.Id

--WHERE A.Id=@author\_id

--)

--SELECT \* FROM ShowBooksByAuthorId(2)

-- EN SON TARIXDE GOTURULEN KITABIN ADINI QAYTAR - SCALAR

-- GONDERILEN CATEGORIYAYA UYGUN OLAN KITABLAR VE MUELLIMLERI - TABLE RETURN

-- EN CHOX GOTURULEN KITABIN MUELLIFLERINI GOSTERIN ILK 10 NEFER - TABLE