All these concepts are asked in many interviews. Please watch the Parts 30, 31 and 32.  
[**Scalar User Defined Functions - Part 30**](http://csharp-video-tutorials.blogspot.com/2012/09/scalar-user-defined-functions-in-sql.html)  
[**Inline table valued functions - Part 31**](http://csharp-video-tutorials.blogspot.com/2012/09/inline-table-valued-functions-part-31.html)  
[**Multi-Statement Table Valued Functions - Part 32**](http://csharp-video-tutorials.blogspot.com/2012/09/multi-statement-table-valued-functions.html)  
  
**Deterministic and Nondeterministic Functions:**  
Deterministic functions always return the **same result** any time they are called with a specific set of input values and given the same state of the database.   
**Examples**: Sum(), AVG(), Square(), Power() and Count()  
  
**Note**: All aggregate functions are deterministic functions.  
  
**Nondeterministic functions** may return **different results** each time they are called with a specific set of input values even if the database state that they access remains the same.  
**Examples**: GetDate() and CURRENT\_TIMESTAMP  
  
Rand() function is a **Non-deterministic function**, but if you provide the **seed value**, the function becomes **deterministic**, as the same value gets returned for the same seed value.   
  
   
  
   
  
   
  
  
**We will be using tblEmployees table, for the rest of our examples**. Please, create the table using this script.  
CREATE TABLE [dbo].[tblEmployees]  
(  
 [Id] [int] Primary Key,  
 [Name] [nvarchar](50) NULL,  
 [DateOfBirth] [datetime] NULL,  
 [Gender] [nvarchar](10) NULL,  
 [DepartmentId] [int] NULL  
)  
  
**Insert rows into the table using the insert script below.**  
Insert into tblEmployees values(1,'Sam','1980-12-30 00:00:00.000','Male',1)  
Insert into tblEmployees values(2,'Pam','1982-09-01 12:02:36.260','Female',2)  
Insert into tblEmployees values(3,'John','1985-08-22 12:03:30.370','Male',1)  
Insert into tblEmployees values(4,'Sara','1979-11-29 12:59:30.670','Female',3)  
Insert into tblEmployees values(5,'Todd','1978-11-29 12:59:30.670','Male',1)  
  
**Encrypting a function definiton using WITH ENCRYPTION OPTION:**  
We have learnt how to encrypt Stored procedure text using WITH ENCRYPTION OPTION in [**Part 18 of this video series**](http://csharp-video-tutorials.blogspot.com/2012/08/stored-procedures-part-18.html). Along the same lines, you can also encrypt a function text. Once, encrypted, you cannot view the text of the function, using **sp\_helptext**system stored procedure. If you try to, you will get a message stating 'The text for object is encrypted.' There are ways to decrypt, which is beyond the scope of this video.  
  
**Scalar Function without encryption option:**  
Create Function fn\_GetEmployeeNameById(@Id int)  
Returns nvarchar(20)  
as  
Begin  
 Return (Select Name from tblEmployees Where Id = @Id)  
End  
  
**To view text of the function:**  
sp\_helptex fn\_GetEmployeeNameById  
  
**Now, let's alter the function to use WITH ENCRYPTION OPTION**  
Alter Function fn\_GetEmployeeNameById(@Id int)  
Returns nvarchar(20)  
With Encryption  
as  
Begin  
 Return (Select Name from tblEmployees Where Id = @Id)  
End  
  
**Now try to retrieve, the text of the function, using sp\_helptex fn\_GetEmployeeNameById**. You will get a message stating 'The text for object 'fn\_GetEmployeeNameById' is encrypted.'  
  
**Creating a function WITH SCHEMABINDING option:**  
1. **The function fn\_GetEmployeeNameById**(), is dependent on tblEmployees table.   
2. Delete the table **tblEmployees** from the database.   
Drop Table tblEmployees  
3. Now, execute the function fn\_GetEmployeeNameById(), you will get an error stating 'Invalid object name tblEmployees'. So, we are able to delete the table, while the function is still refrencing it.  
4. Now, **recreate the table** and insert data, using the scripts provided.  
5. Next, **Alter the function fn\_GetEmployeeNameById()**, to use WITH SCHEMABINDING option.  
Alter Function fn\_GetEmployeeNameById(@Id int)  
Returns nvarchar(20)  
With SchemaBinding  
as  
Begin  
 Return (Select Name from dbo.tblEmployees Where Id = @Id)  
End  
  
**Note**: You have to use the **2 part object name** i.e, dbo.tblEmployees, to use WITH SCHEMABINDING option. dbo is the schema name or owner name, tblEmployees is the table name.  
6. Now, **try to drop the table using** - Drop Table tblEmployees. You will get a message stating, 'Cannot DROP TABLE tblEmployees because it is being referenced by object fn\_GetEmployeeNameById.'  
  
So, Schemabinding, specifies that the function is bound to the database objects that it references. When SCHEMABINDING is specified, the base objects cannot be modified in any way that would affect the function definition. The function definition itself must first be modified or dropped to remove dependencies on the object that is to be modified.