(T1)入門 PK、FK、DefaultConstraint、CheckConstraint、IdentityColumn
CourseGUID: e48417fc-9db5-4e99-822c-706c5ccef6cc
==========
(T1)入門 PK、FK、DefaultConstraint、CheckConstraint、IdentityColumn
==========

==========

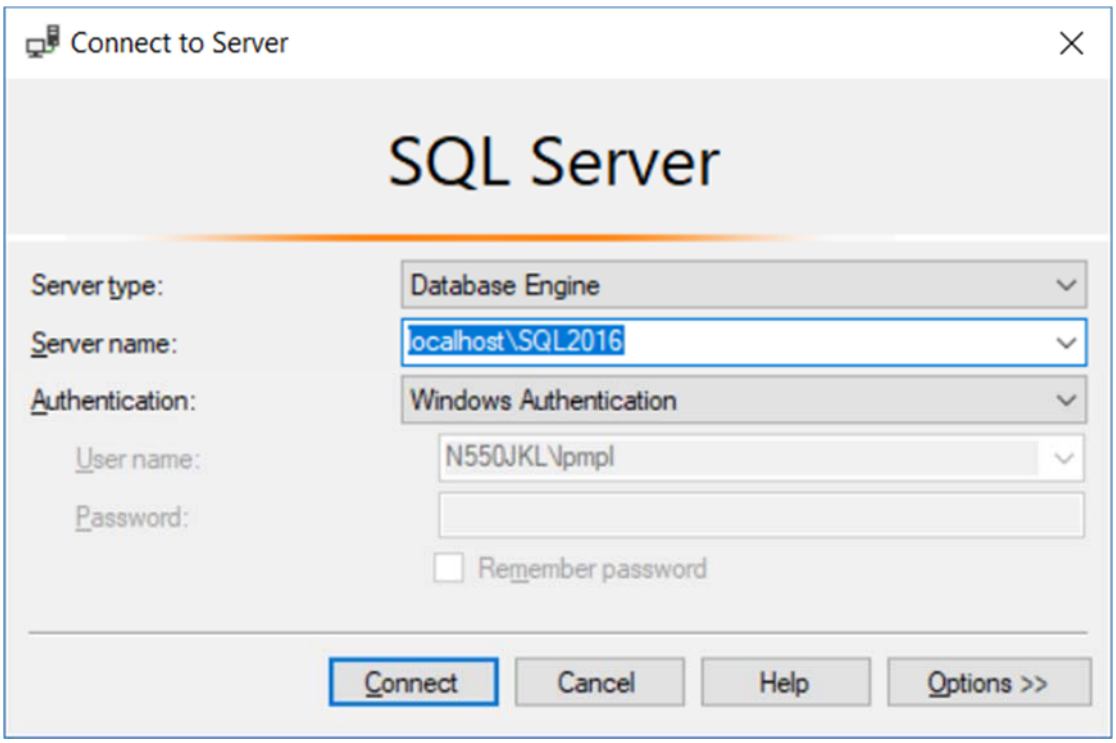
0. What to learn

- Connecting to SQL Server using SQL Server Management Studio(SSMS)
- Windows Authentication and SQL Server Authentication.
- Creating, altering and dropping a database
- Set database to single user mode and delete it.
- Create Table
- Default Constraint
- Check Constraint
- Identity Column
- Primary Key
- Foreign Key
- Insert

1. SSMS

What to learn
- Connecting to SQL Server using SQL Server Management Studio(SSMS)

- Windows Authentication and SQL Server Authentication.



SSMS is just a client side user interface to connect sql server.
I normally installed several version of SQL server such as SQL Server 2014 or SQL Server 2016.
Thus, during installation, I normally created an instance name, **SQL2016** for SQL Server 2016 instance in my case.

Server Name is the place you need to decide which sql server you want to connect.
In my case, I want to connect to SQL Server 2016 instance which instance name is **SQL2016**
Then I can type the following.
Server name = localhost\SQL2016
Server name = 127.0.0.1\SQL2016

During installation, I normally selected **mixed mode authentication**,
Thus, I have both **Windows Authentication** and **SQL Server Authentication** option to connect the SQL Server.

When I connect to SQL server, If I select **SQL Server Authentication.**
Then I have to enter the user name and password to connect the SQL server.

In addition, when I login to Windows, I have already authenticated by local windows account.
When I connect to SQL server, I can select **Windows Authentication.**
That means I don't have to enter user name and password again,
because I was authenticated by local windows account already.
This is what I am going to do.

2. Create and Drop Database

What to learn

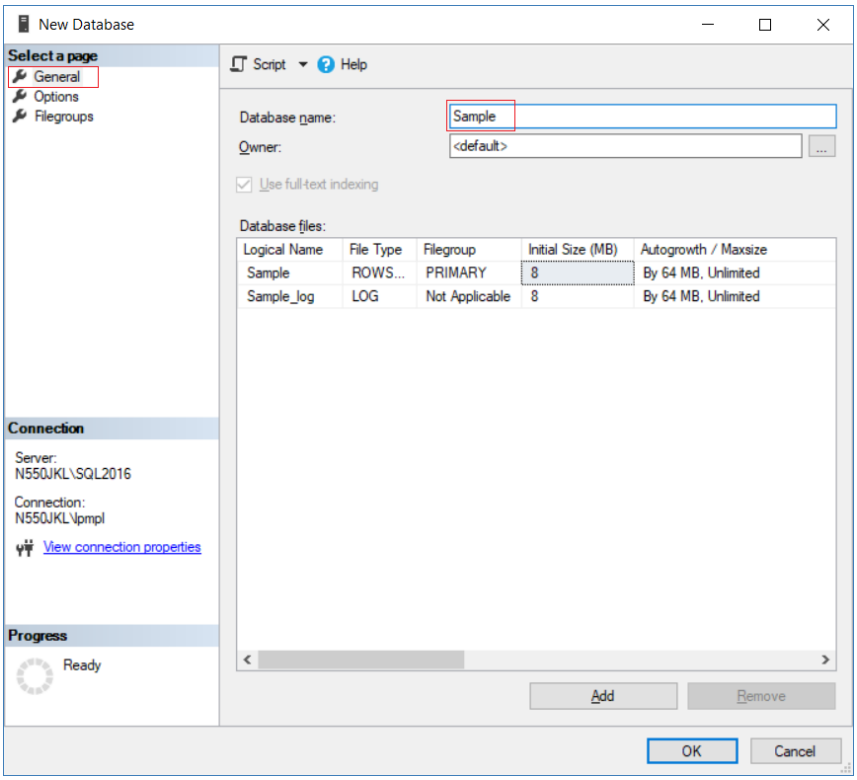
- Creating, altering and dropping a database
- Set database to single user mode and delete it.

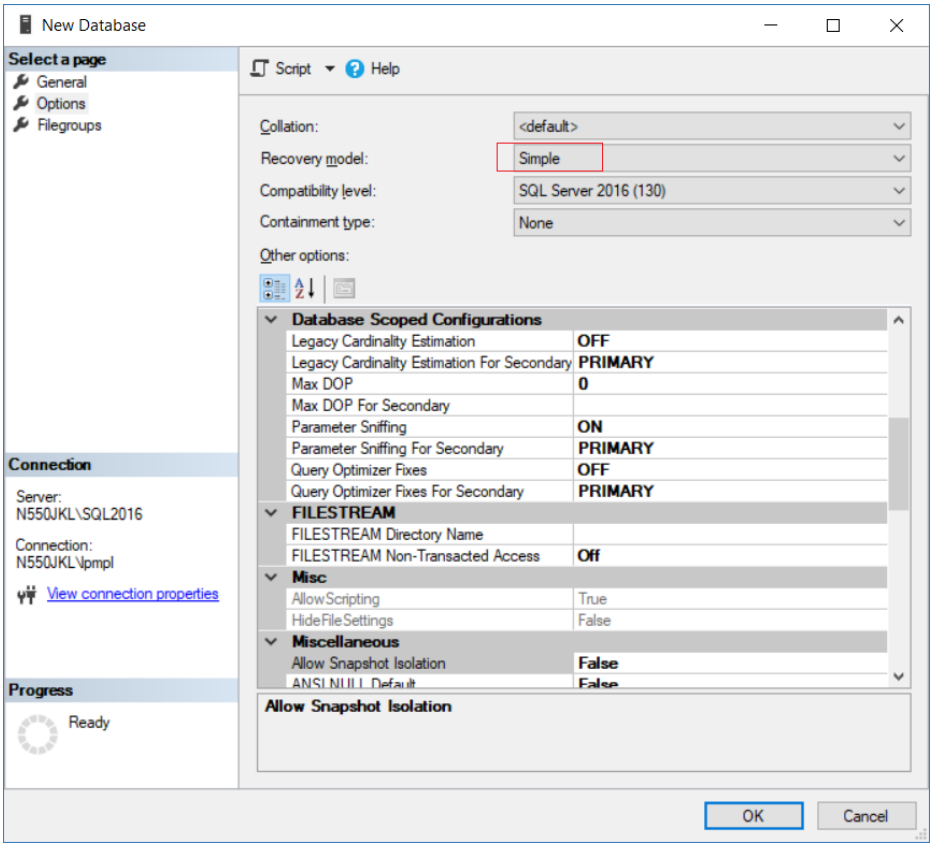
2.1. Using SSMS to Create and Drop Database

2.1.1. Using SSMS to Create Database

To create the database graphically
1. Right Click on Databases folder in the Object explorer
2. Select New Database
3. In the New Database dialog box, enter the Database name and click OK.

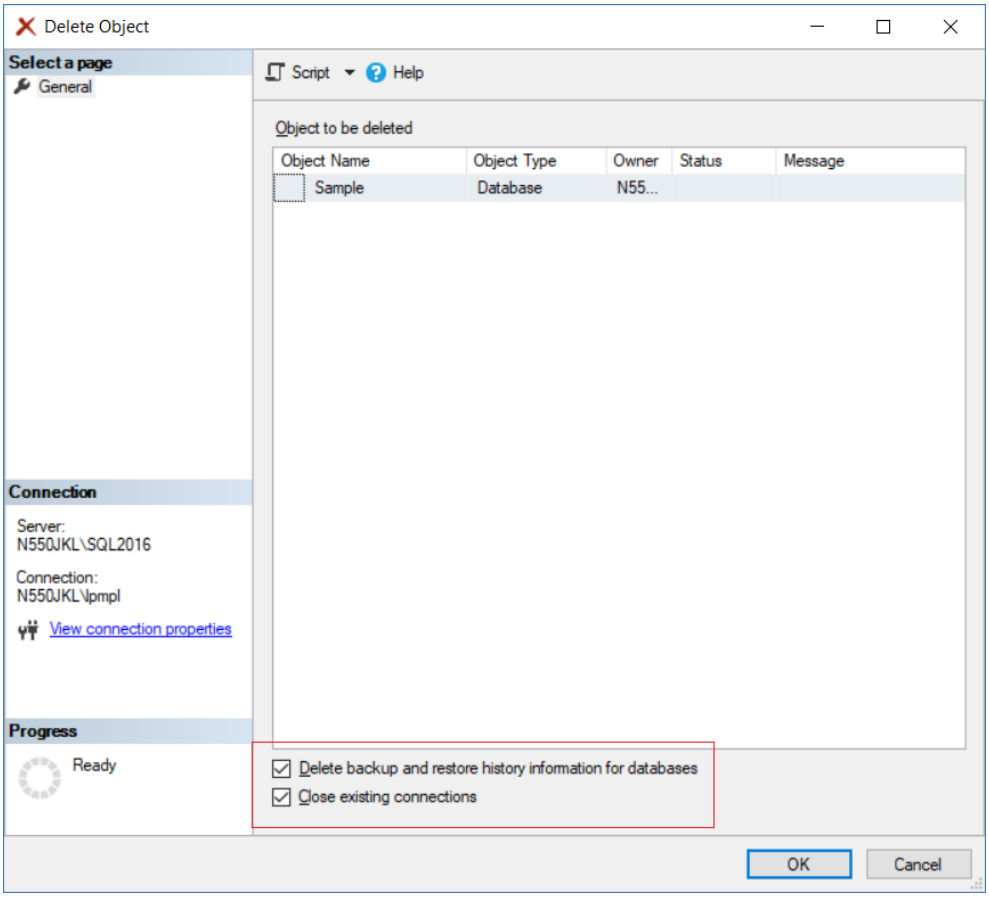
Database Name : **Sample**
Recovery Model : **Simple**





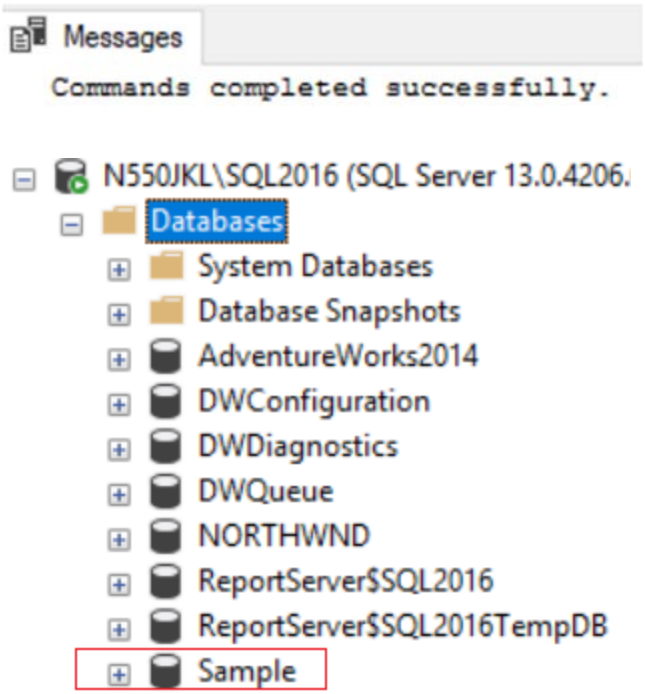
2.1.2. Using SSMS to DropDatabase

If you want to use SSMS to delete table. please see the picture.
Database Name --> Press "Delete" button



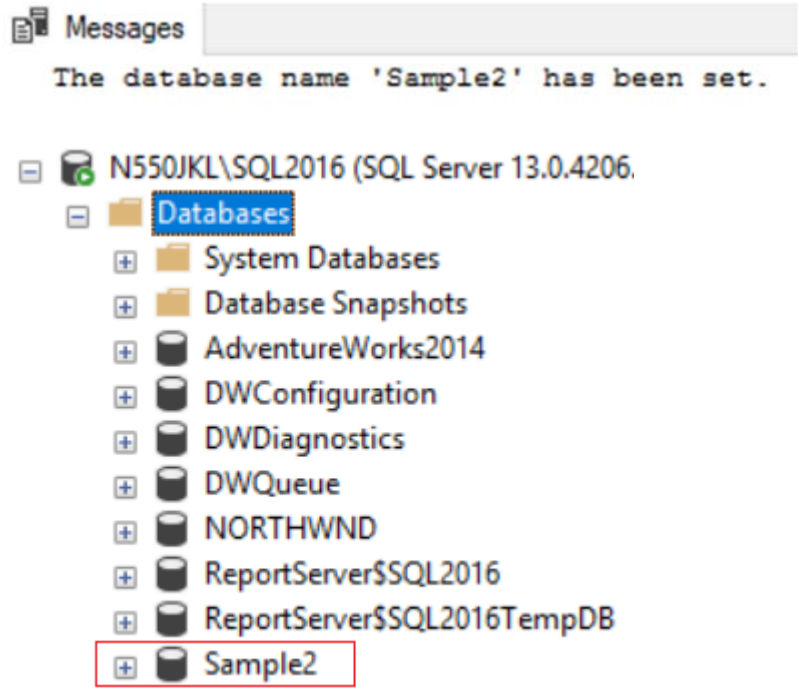
2.2. Using Query to Create and Drop Database

```
/*
What to learn
- Creating, altering and dropping a database
- Set database to single user mode and delete it.
*/


--============================================================================================================
--T001_01
--Database
--------------------------------------------------------------------------------------------------------------
---
--T001_01_01
--Create Database
USE master
GO -- Run the prvious command and begins new batch
CREATE DATABASE [Sample];
GO
/*
-- CREATE DATABASE DatabaseName;
Create Database
*/
```

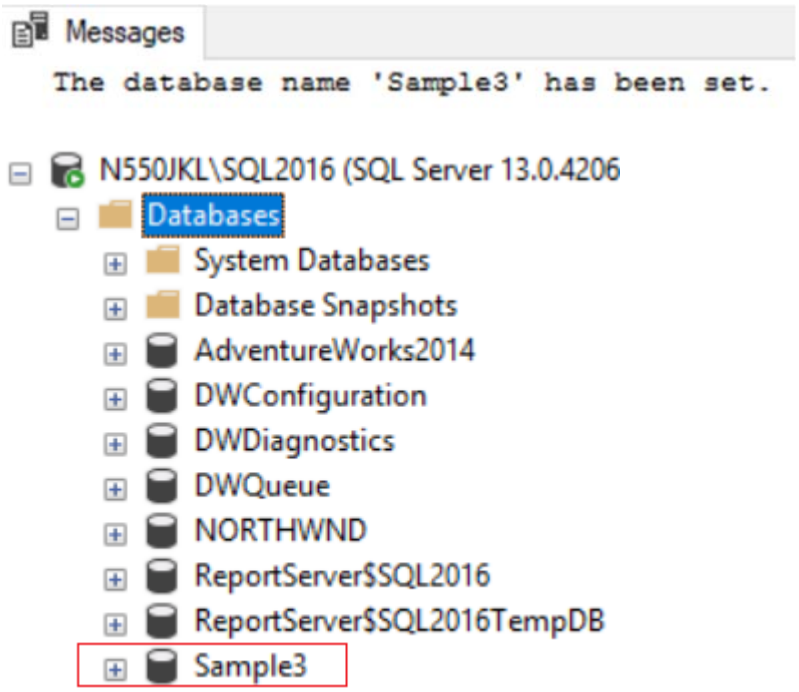



```
--------------------------------------------------------------------------------------------------------------
---
--T001_01_02
--Change Database Name
USE master
GO -- Run the prvious command and begins new batch
ALTER DATABASE [Sample] MODIFY NAME = Sample2;
GO
/*
-- ALTER DATABASE DatabaseName MODIFY NAME = NewDatabaseName;
```

```
Alter Database Name
*/
```




```
-----------------------------------------------------------------------------------------------------------
---
--T001_01_03
--sp_renamedb
USE master
GO -- Run the prvious command and begins new batch
EXEC sp_renamedb N'Sample2', N'Sample3';
GO
/*
-- ALTER EXECUTE sp_renameDB 'OldDatabaseName', 'NewDatabaseName';
Reference:
https://docs.microsoft.com/en-us/sql/relational-databases/system-stored-procedures/sp-renamedb-transact-
sql
Alter Database Name
*/
```



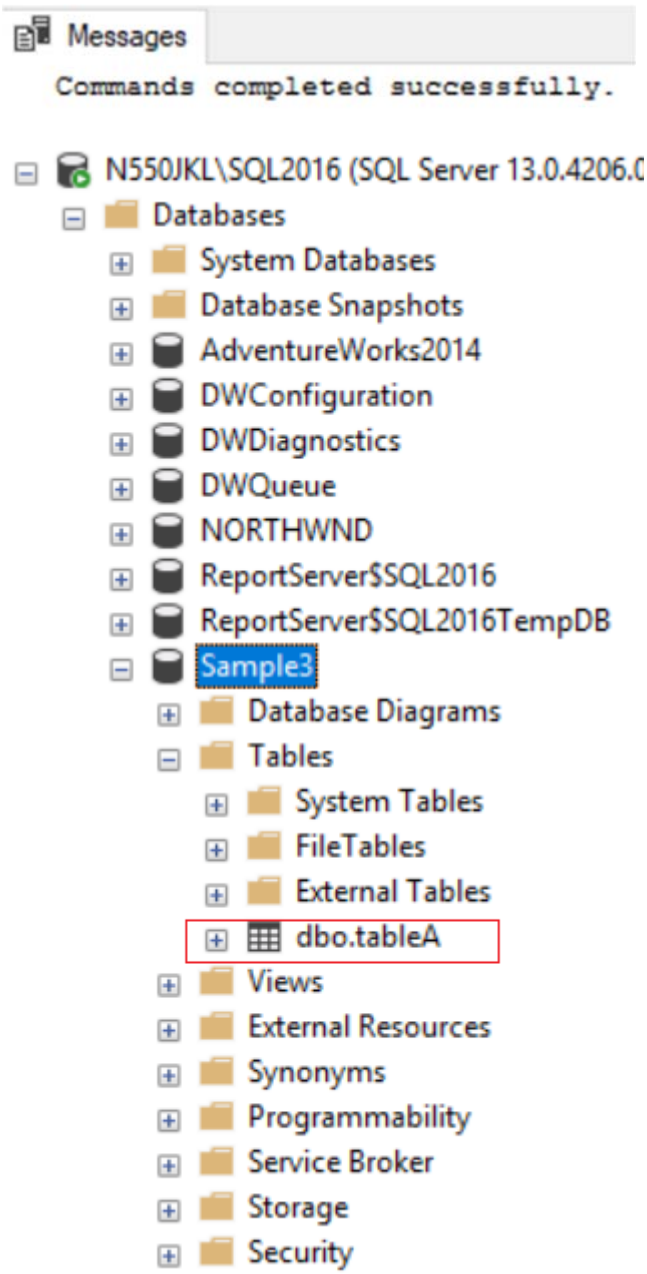
```
-------------------------------------------------------------------------------------------------------------
---
--T001_01_04
--sys.databases
USE master
GO -- Run the prvious command and begins new batch
SELECT [name], database_id, create_date
FROM sys.databases
WHERE name = N'Sample3';
GO
/*
sys.databases is the system database which store all the database list information
*/
```

Results | Messages

	name	database_id	create_date
1	Sample3	10	2017-09-02 01:31:28.110

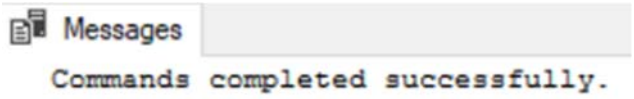
```
-------------------------------------------------------------------------------------------------------------
---
--T001_01_05
--Create Table ON  [PRIMARY] in Sample3
USE Sample3
GO -- Run the prvious command and begins new batch
CREATE TABLE [dbo].[tableA]
    (
          [Id] [INT]  IDENTITY(1, 1) NOT NULL,
         --[Id] [INT] IDENTITY(1, 1) PRIMARY KEY NOT NULL,
       [Name] [NVARCHAR](50) NOT NULL ,
      CONSTRAINT [PK_tableA] PRIMARY KEY CLUSTERED ( [Id] ASC )
        WITH ( PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
               IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
               ALLOW_PAGE_LOCKS = ON ) ON [PRIMARY]
    )
ON  [PRIMARY];
GO
/*
1.
There are 2 ways to set the primary Key
1.1.
--PersonId INT IDENTITY(1, 1) PRIMARY KEY NOT NULL,
1.2.
--[Id] [INT] IDENTITY(1, 1) NOT NULL ,
--CONSTRAINT [PK_Gamer2] PRIMARY KEY CLUSTERED ( [Id] ASC )
--    WITH ( PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
--        IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
--        ALLOW_PAGE_LOCKS = ON ) ON [PRIMARY]
2.
[Id] [int] IDENTITY(1,1) NOT NULL,
It means Id is the Primary Key and the type is int.
Id will start from 1 (the first one is identity seed),
and then increase 1 (the second one is identity increment)
3.
-- ON [PRIMARY]
When you create database, SQL server will generate
one .MDF(primary data file) and one .LDF(log file)
Sometimes a SQL Server database will include one or more .NDF (secondary data files).
-- ON [PRIMARY]
```

```
means create this table on the .MDF(primary data file).
*/
```




```
-------------------------------------------------------------------------------------------------------
---
--T001_01_06
--forced to delete DATABASE Sample3
USE master;
 -- be sure that you're not on the database you want to delete
GO -- Run the prvious command and begins new batch
IF ( EXISTS ( SELECT    [name] ,
                        database_id ,
                        create_date
              FROM      sys.databases
              WHERE     name = N'Sample3' ) )
    BEGIN
        ALTER DATABASE [Sample3] SET SINGLE_USER WITH ROLLBACK IMMEDIATE;
        DROP DATABASE [Sample3];
    END;
```

```
GO -- Run the previous command and begins new batch
/*
1.
--IF ( EXISTS ( SELECT    [name] ,
--                        database_id ,
--                        create_date
--              FROM      sys.databases
--              WHERE     name = N'Sample3' ) )
If the Sample3 exist.
2.
Reference:
https://stackoverflow.com/questions/17095472/cannot-drop-database-because-it-is-currently-in-use-mvc
Error Message:
Cannot drop database "NewDatabaseName" because it is currently in use.
Solutons:
--ALTER DATABASE [Sample3] SET SINGLE_USER WITH ROLLBACK IMMEDIATE
--DROP DATABASE [Sample3];
put the database in single user mode which
will rollback all incomplete transactions and closes the connection to the database.
then drop the database.
*/
```

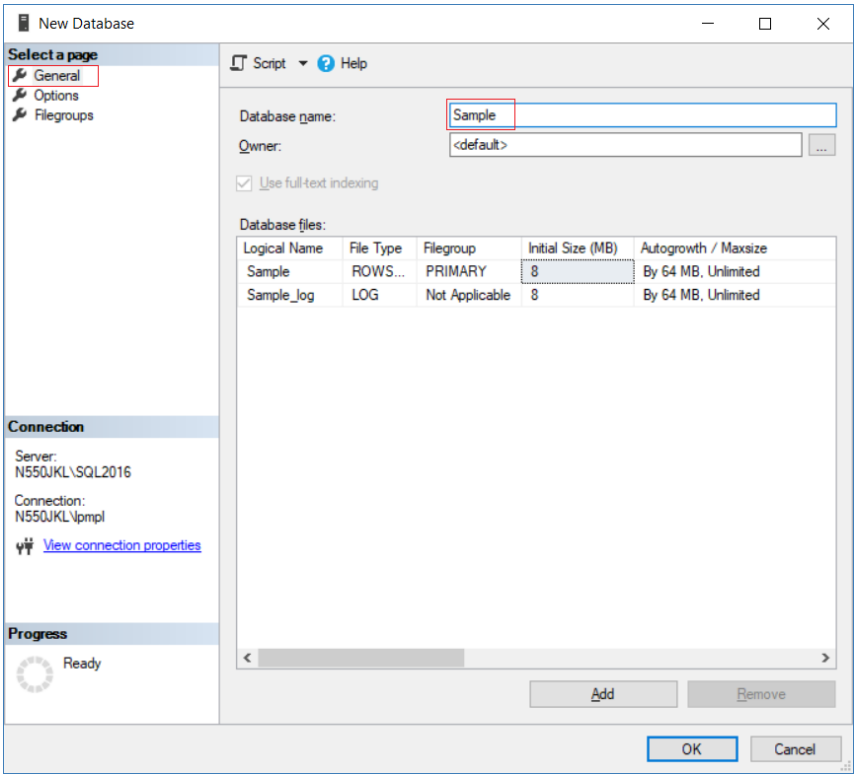


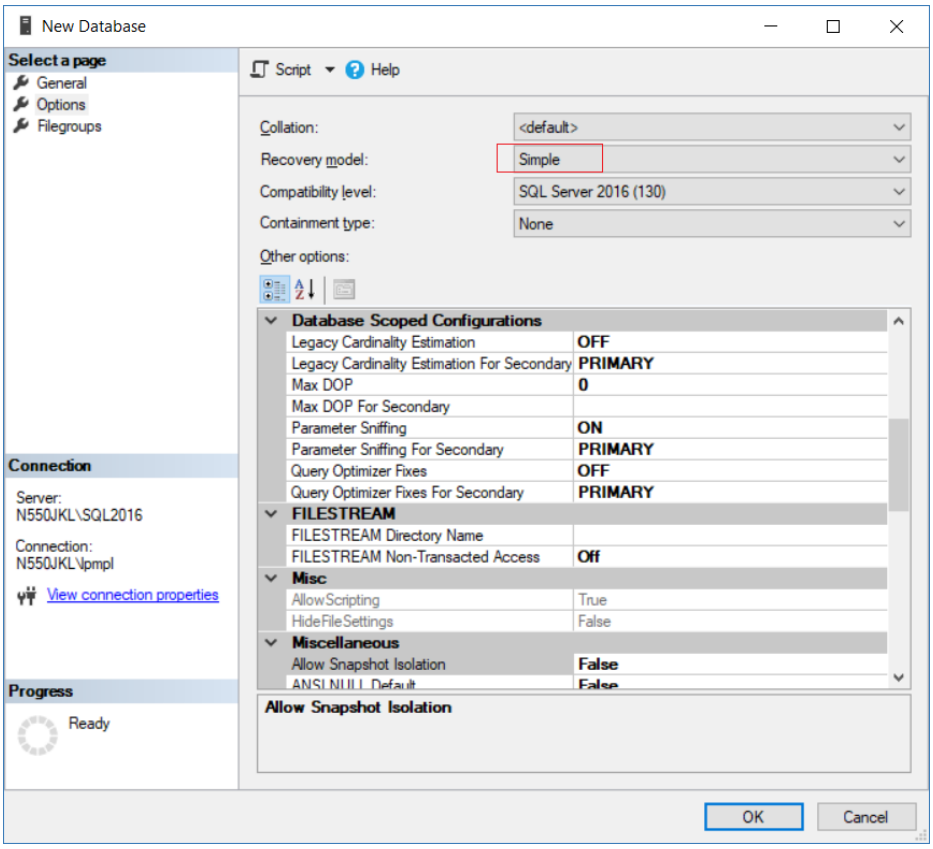
3. Using SSMS to Create Tables -> Set Default Constraint -> Set Check Constraint -> Set Referential Integrity constraint (Foreign Key)

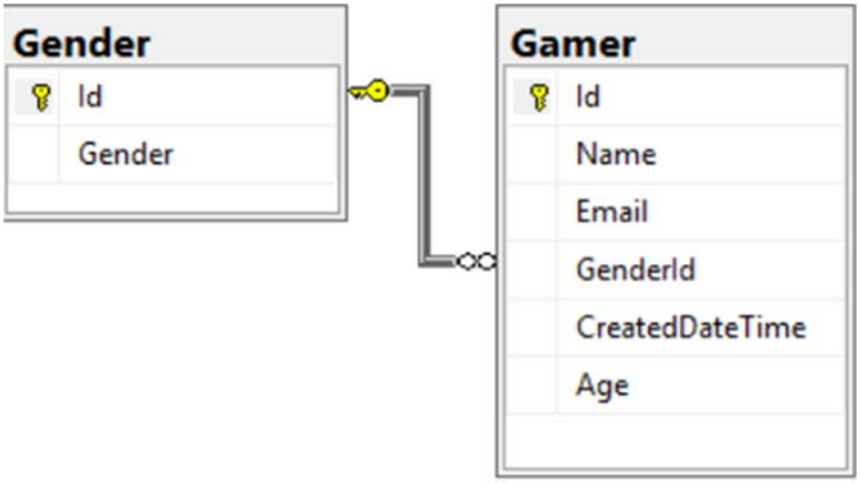
What to learn
- Create Table
- Default Constraint
- Check Constraint
- Identity Column
- Primary Key
- Foreign Key
- Insert

Database --> New Database

Database Name : **Sample**
Recovery Model : **Simple**


New Database
Select a page
General
Options
Filegroups
Script
Help
Database name:
Sample
Owner:
<default>
Use full-text indexing
Database files:
Logical Name
File Type
Filegroup
Initial Size (MB)
Autogrowth / Maxsize
Sample
ROWS...
PRIMARY
8
By 64 MB, Unlimited
Sample_log
LOG
Not Applicable
8
By 64 MB, Unlimited
Connection
Server:
N550JKL\SQL2016
Connection:
N550JKL\lpmpl
View connection properties
Progress
Ready
Add
Remove
OK
Cancel


New Database
Select a page
General
Options
Filegroups
Script
Help
Collation:
<default>
Recovery model:
Simple
Compatibility level:
SQL Server 2016 (130)
Containment type:
None
Other options:
Database Scoped Configurations
Legacy Cardinality Estimation
OFF
Legacy Cardinality Estimation For Secondary
PRIMARY
Max DOP
0
Max DOP For Secondary
Parameter Sniffing
ON
Parameter Sniffing For Secondary
PRIMARY
Query Optimizer Fixes
OFF
Query Optimizer Fixes For Secondary
PRIMARY
FILESTREAM
FILESTREAM Directory Name
FILESTREAM Non-Transacted Access
Off
Misc
AllowScripting
True
HideFileSettings
False
Miscellaneous
Allow Snapshot Isolation
False
Allow Snapshot Isolation
Connection
Server:
N550JKL\SQL2016
Connection:
N550JKL\lpmpl
View connection properties
Progress
Ready
OK
Cancel

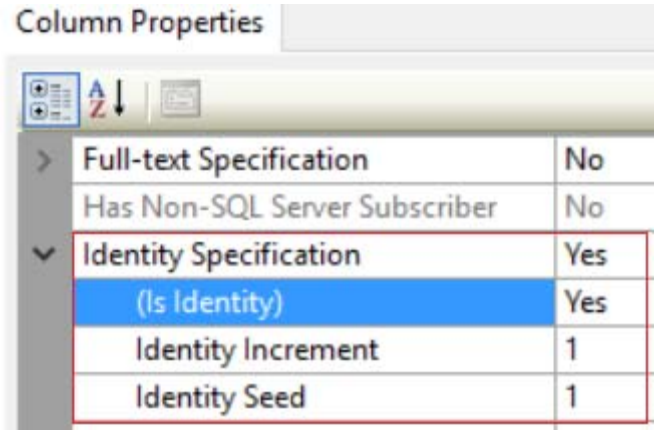


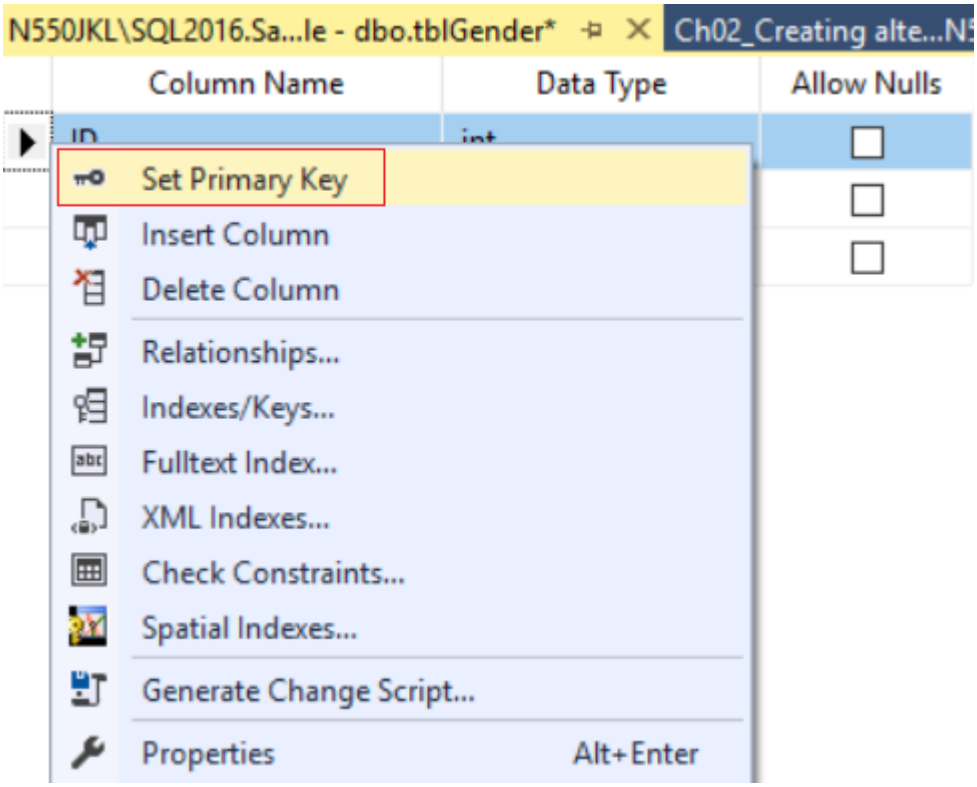
3.1. CreateTables - Gamer, Gender

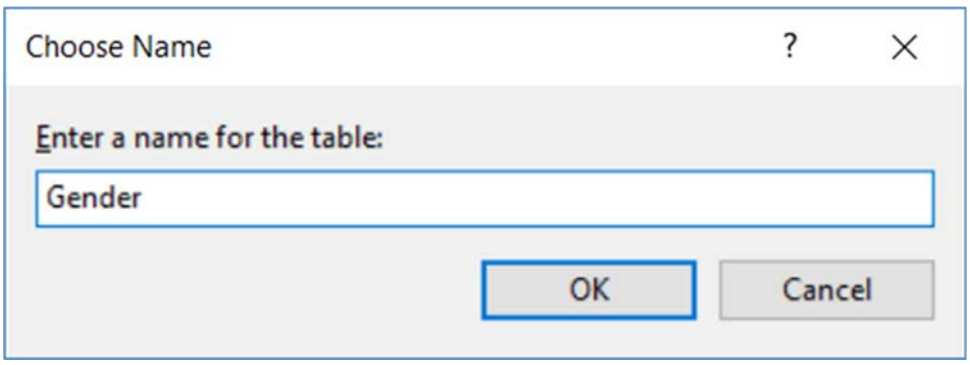
3.1.1. CreateTable - Gender

Database Name --> Tables --> New --> Table...
-->
Id int NULL
Gender nvarchar(50) NULL
-->
Select **Id** column
set the Column Properties
Identity Specification
(Is Identity) Yes
Identity Increment 1
Identity Seed 1
-->
Select Id column --> Right Click
--> Set Primary Key
-->
Ctrl + S -->
Table Name: Gender

Column Name	Data Type	Allow Nulls
Id	int	☑
Gender	nvarchar(50)	☑
		☐



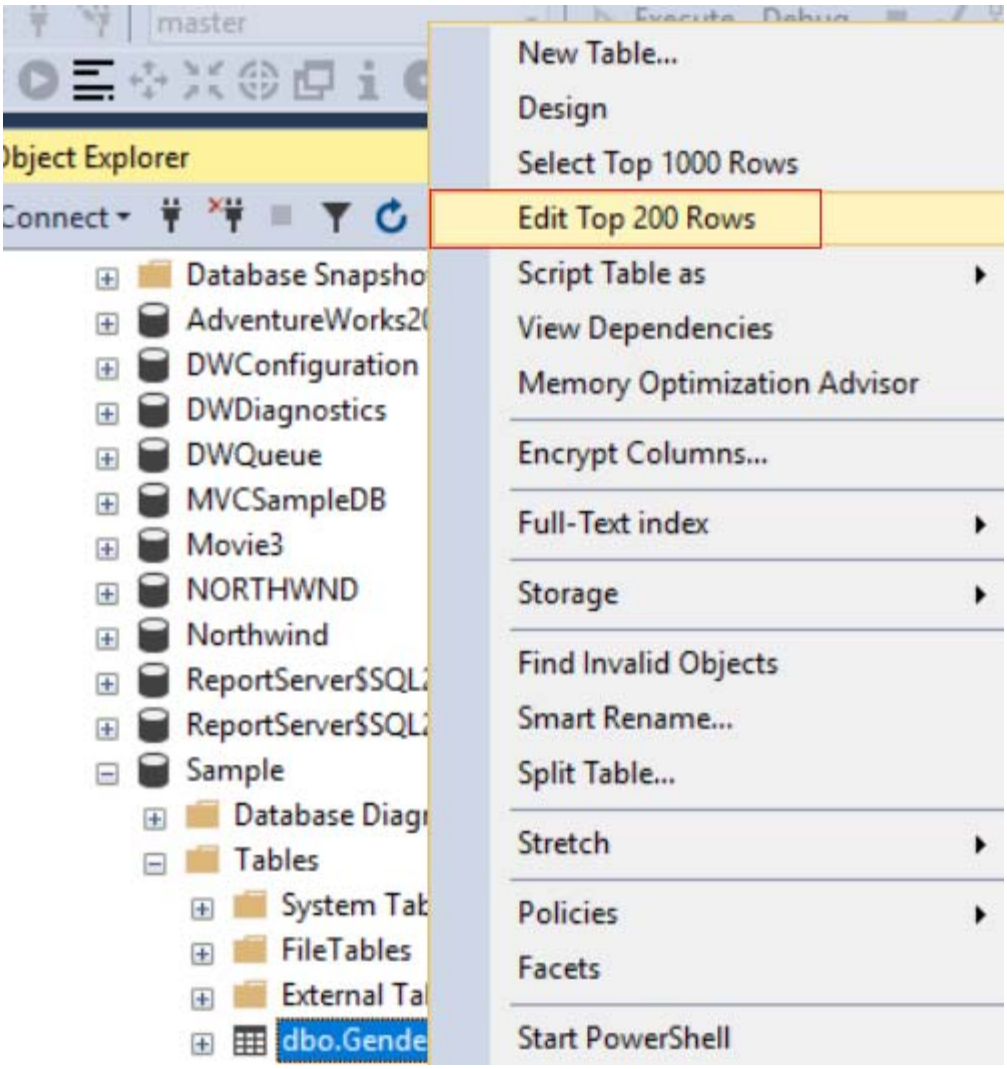




3.1.2. Insert Data to Gender

Database Name --> Tables --> Table Name --> Right Click --> Edit top 200 rows -->
-->
Male
Female
Unknow

-->
Explanation:
[Id] [int] IDENTITY**(1,1) Primary Key NOT NULL,**
When I created table, I set Id is **INT** type,
and I also set Id is **Primary Key**.
Additionally, I set Id is **Identity Column**.
IDENTITY(1,1) means Id will start from 1 (the **first** one is **identity seed**),
and then increase 1 (the **second** one is **identity increment**).
We do not have to set value for **INT Type Identity Column**, because it will be auto generated by SQL server.
Thus, when we insert new data, I only have to provide the rest of data except Id.



	Id	Gender
	1	Male
	2	Female
▶	3	Unknow
*	NULL	NULL

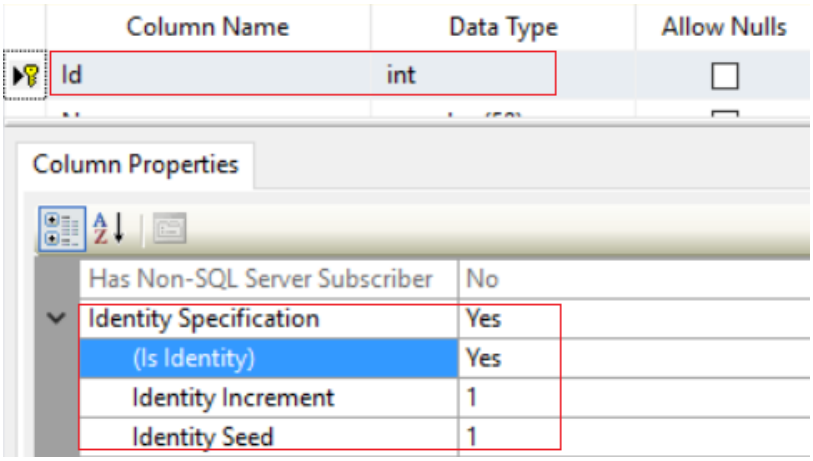
3.1.3. CreateTable - Gamer and Set Default Constraint

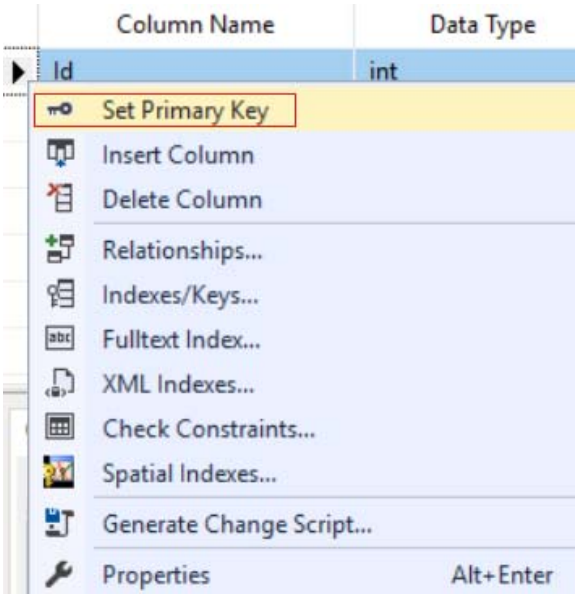
Database Name --> Tables --> New --> Table...

-->
Id int NOT NULL
Name nvarchar(50) NOT NULL
Email nvarchar(50) NOT NULL
GenderId int NULL
CreatedDateTime datetime NOT NULL
Age int NULL
-->
Select Id column
set the Column Properties
Identity Specification
(Is Identity) Yes
Identity Increment 1
Identity Seed 1
-->
Select **CreatedDateTime** column
set the Column Properties
Default Value or Binding
(getutcdate())
-->
Select GenderId column
set the Column Properties
Default Value or Binding
((3))
-->
Select Id column --> Right Click
--> Set Primary Key
-->
Ctrl + S -->
Table Name: Gamer

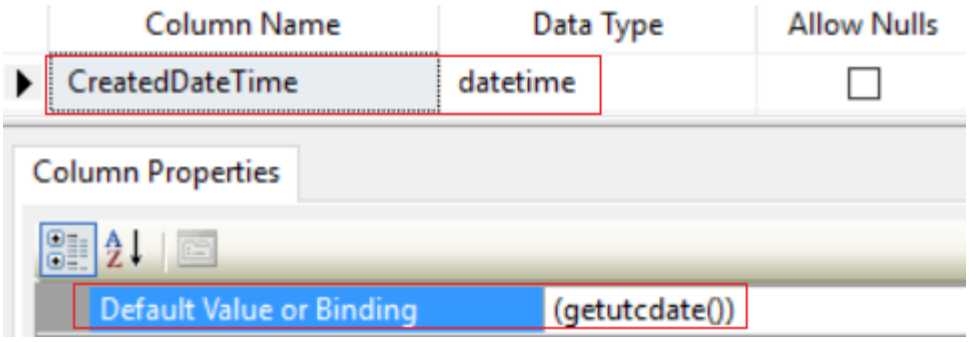
Column Name	Data Type	Allow Nulls
Id	int	☐
Name	nvarchar(50)	☐
Email	nvarchar(50)	☐
GenderId	int	☑
CreatedDateTime	datetime	☐
Age	int	☑

Id column

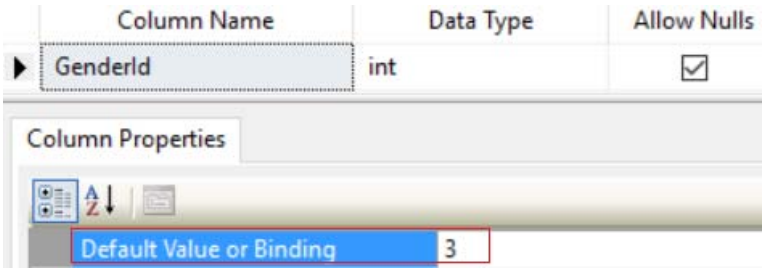


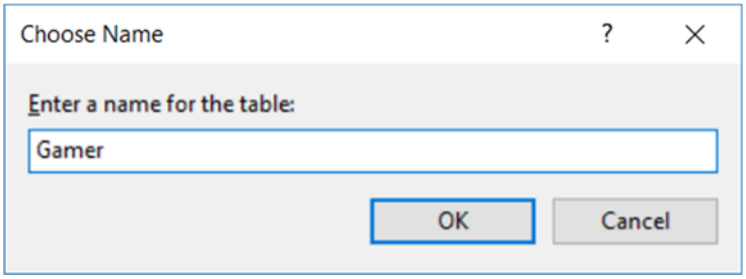


CreatedDateTime Column



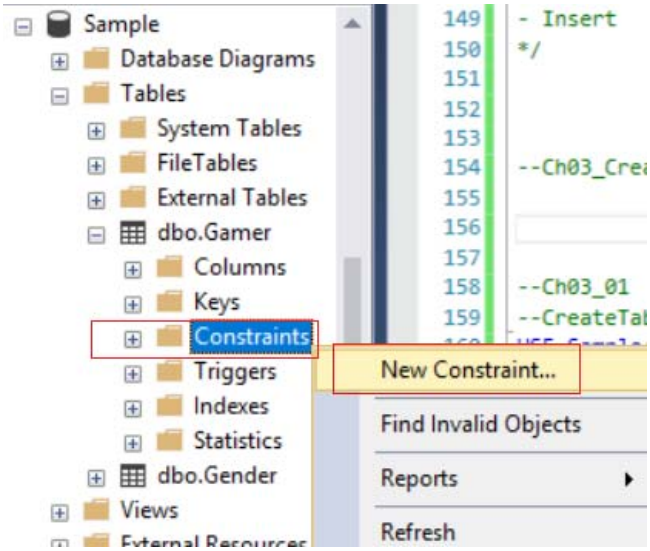
GenderId Column

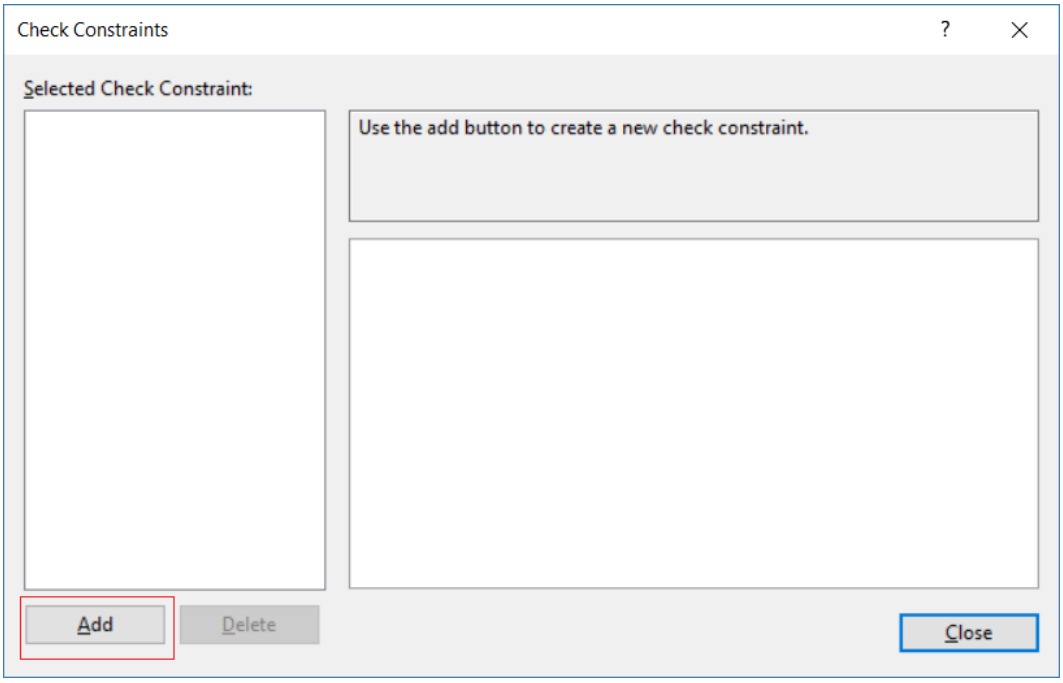


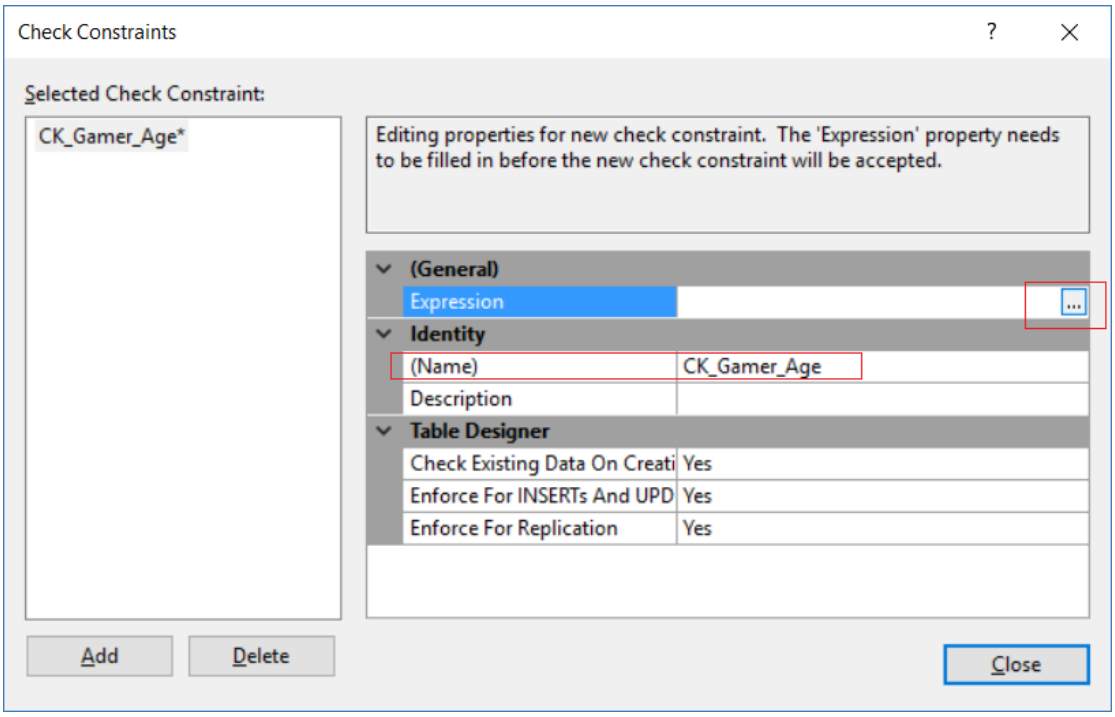


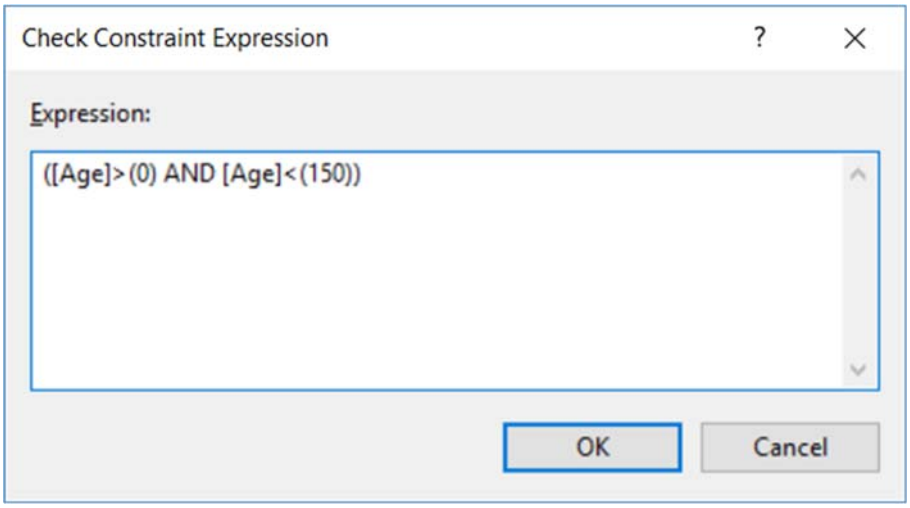
3.1.4. Gamer - Set Check Constraint

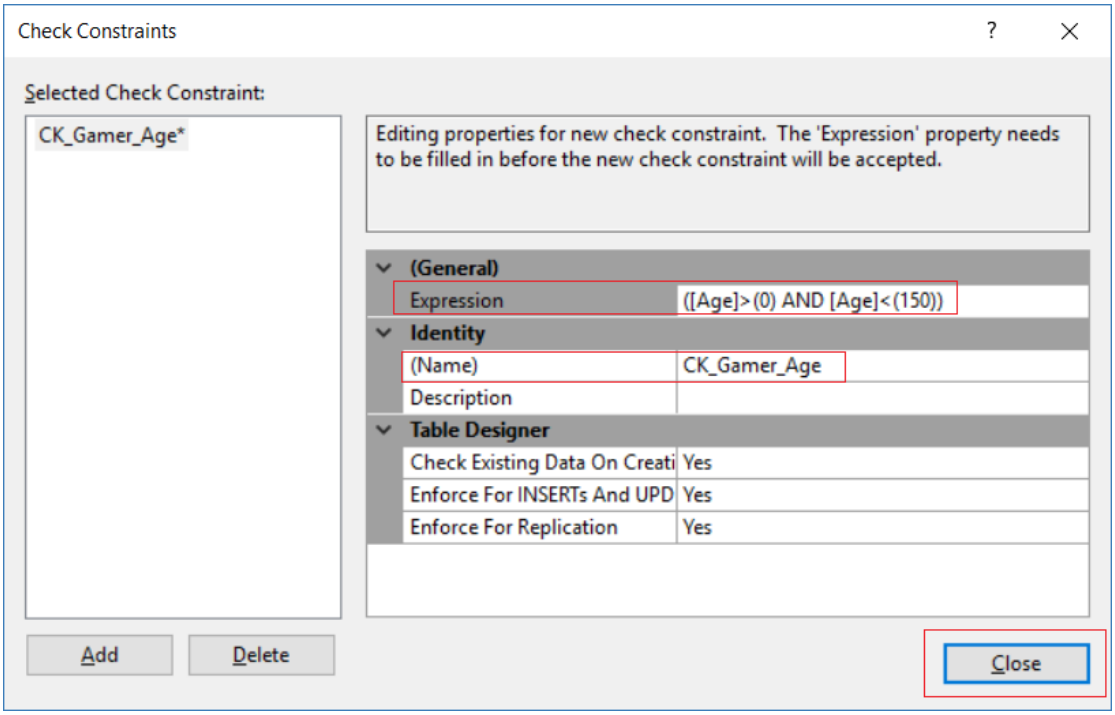
Databsae Name --> Tables --> table name --> Constraints --> New Constraint...
--> Add -->
(Name): **CK_Gamer_Age**
Expression: press **...**
--> **([Age]>(0) AND [Age]<(150))** --> OK
--> Close
--> Save the Table Design

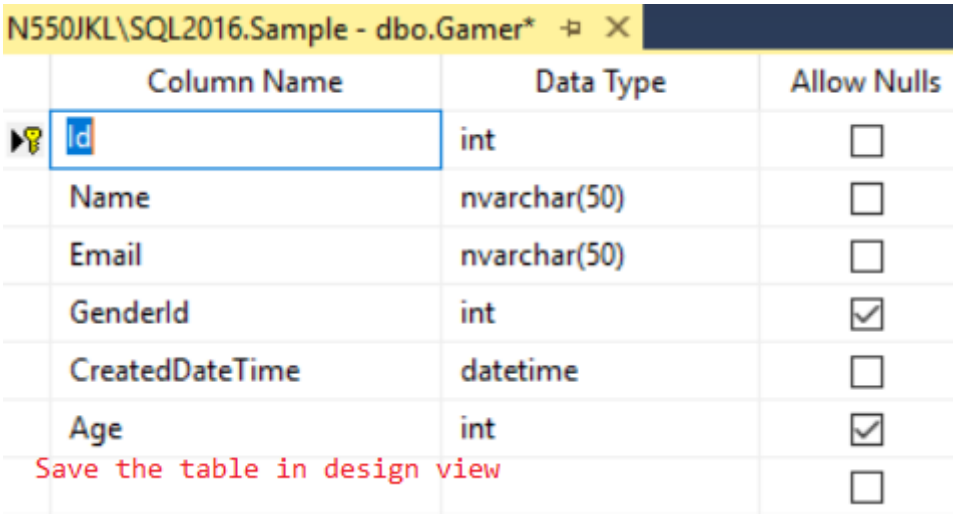


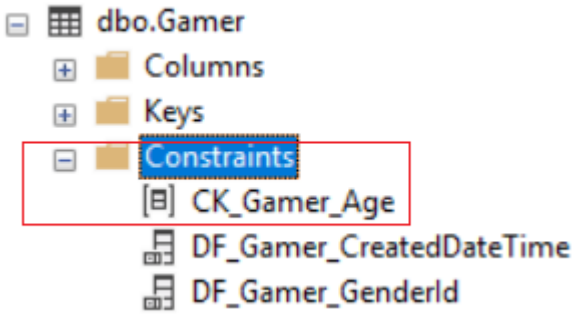

Check Constraints
Selected Check Constraint:
Use the add button to create a new check constraint.
Add
Delete
Close


Check Constraints
Selected Check Constraint:
CK_Gamer_Age*
Editing properties for new check constraint. The 'Expression' property needs to be filled in before the new check constraint will be accepted.
(General)
Expression
Identity
(Name)
CK_Gamer_Age
Description
Table Designer
Check Existing Data On Creati
Yes
Enforce For INSERTs And UPD
Yes
Enforce For Replication
Yes
Add
Delete
Close


Check Constraint Expression
Expression:
([Age]>(0) AND [Age]<(150))
OK
Cancel







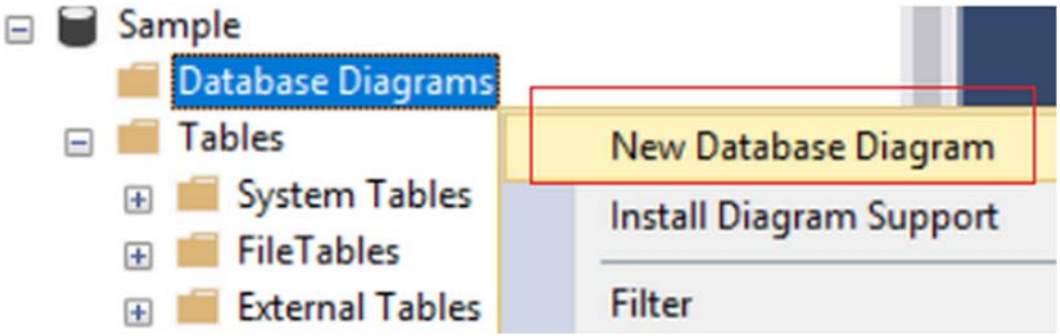
3.1.5. Gamer - Referential Integrity constraint (Foreign Key)

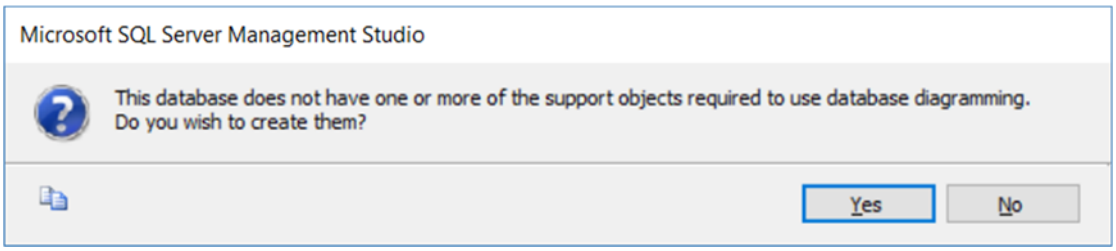
There are 3 ways to create Referential Integrity constraint (Foreign Key) by SSMS
- Create Referential Integrity constraint (Foreign Key) by **Database Diagram**
- Create Referential Integrity constraint (Foreign Key) in **Table**
- Create Referential Integrity constraint (Foreign Key) in **Design View**

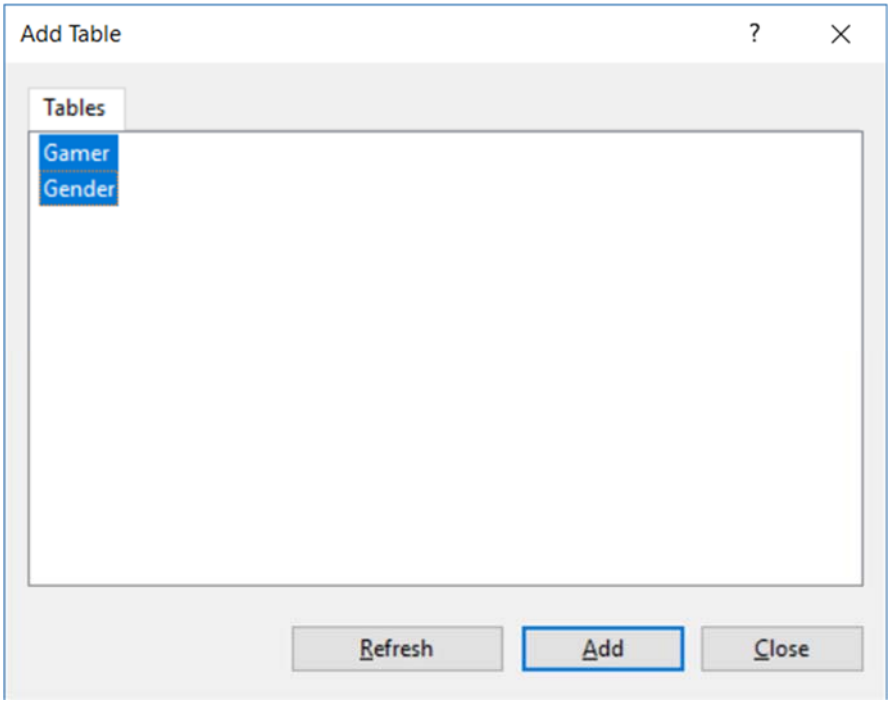
3.1.5.1. Create Referential Integrity constraint (Foreign Key) by Database Diagrams

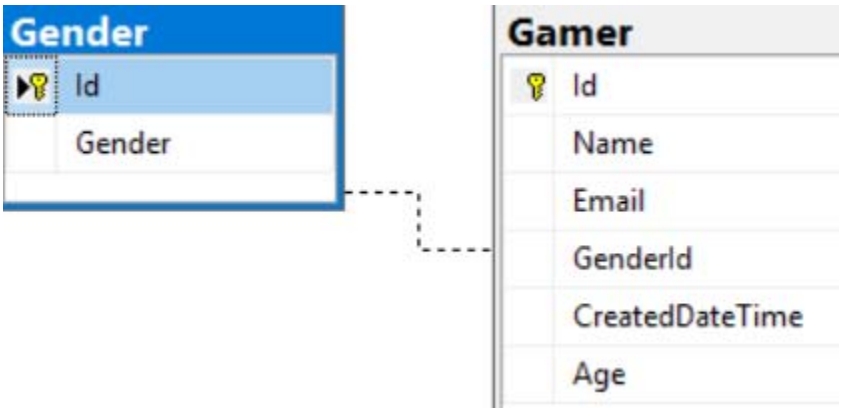
Database Name --> Database Diagrams --> New Database Diagram --> Yes -->
Select All tables from the list --> Add
--> Connect the Relationship
In my case,
Drag from **[**Gender**].[**Id**]** into **[**Gamer**].[**GenderId**]**
-->
It will create **Foreign Key Relationship "FK_**Gamer_Gender"
Make sure I connect **[**Gender**].[**Id**]** into **[**Gamer**].[**GenderId**]**
--> OK --> OK
--> Ctrl + S Save the Database Diagrams
--> Name: **Diagram_0**

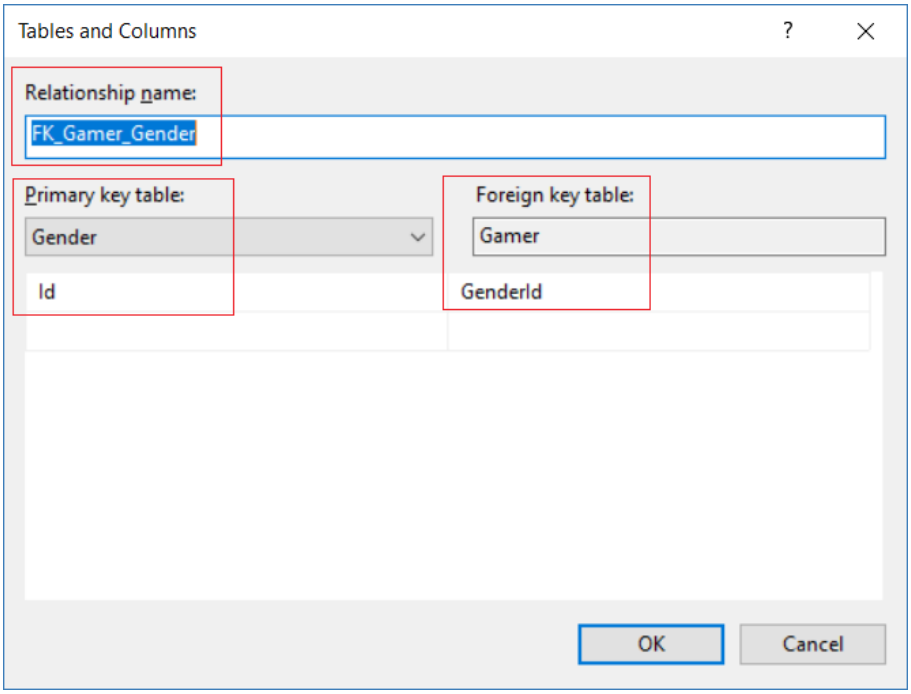
If you want to modify again.
Database Name --> Tables --> Table Name --> Keys --> FK --> Right Click --> Modify -->
Delete Rule / Update Rule --> **No Action / Cascade / Set Null / Set Default**
By default, I set to **No Action**.

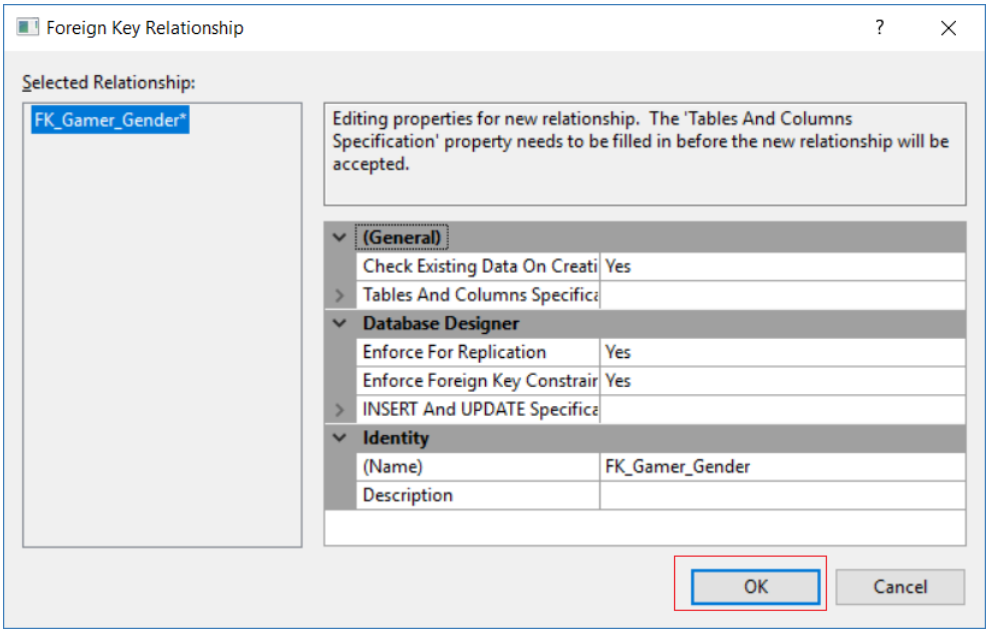


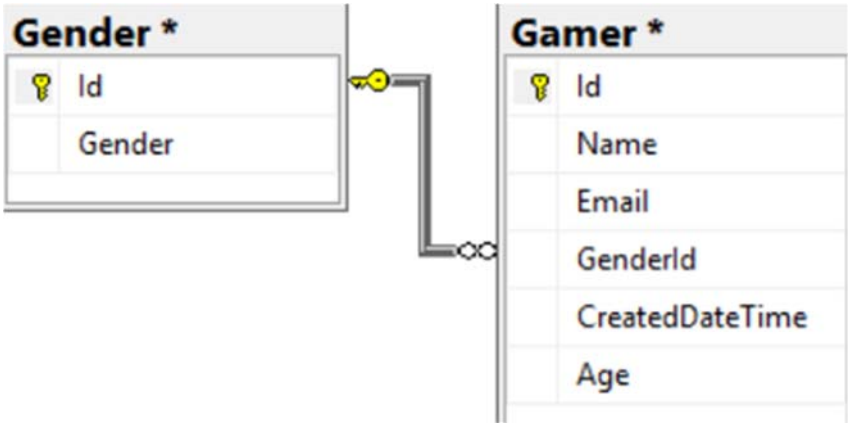


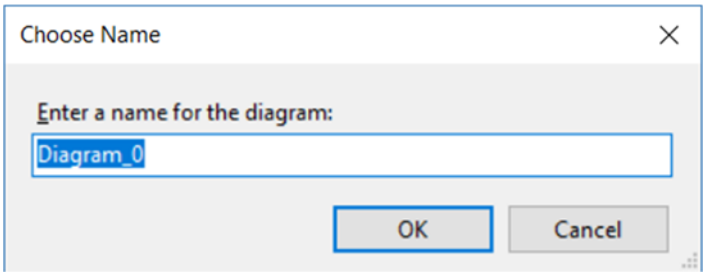

Add Table
Tables
Gamer
Gender
Refresh
Add
Close

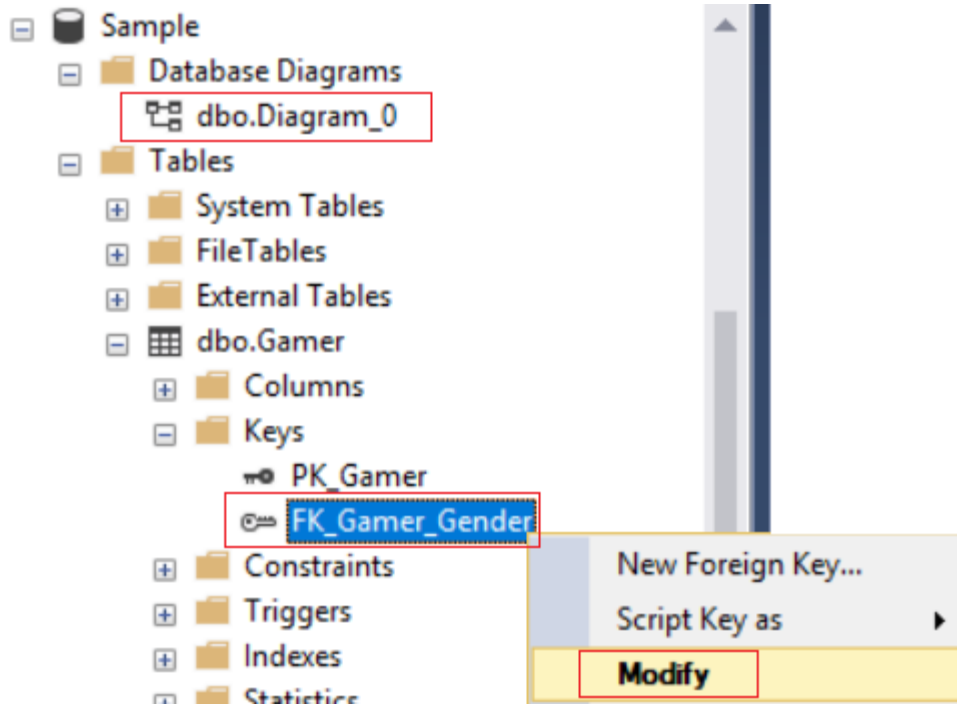

Gender
Id
Gender
Gamer
Id
Name
Email
GenderId
CreatedDateTime
Age

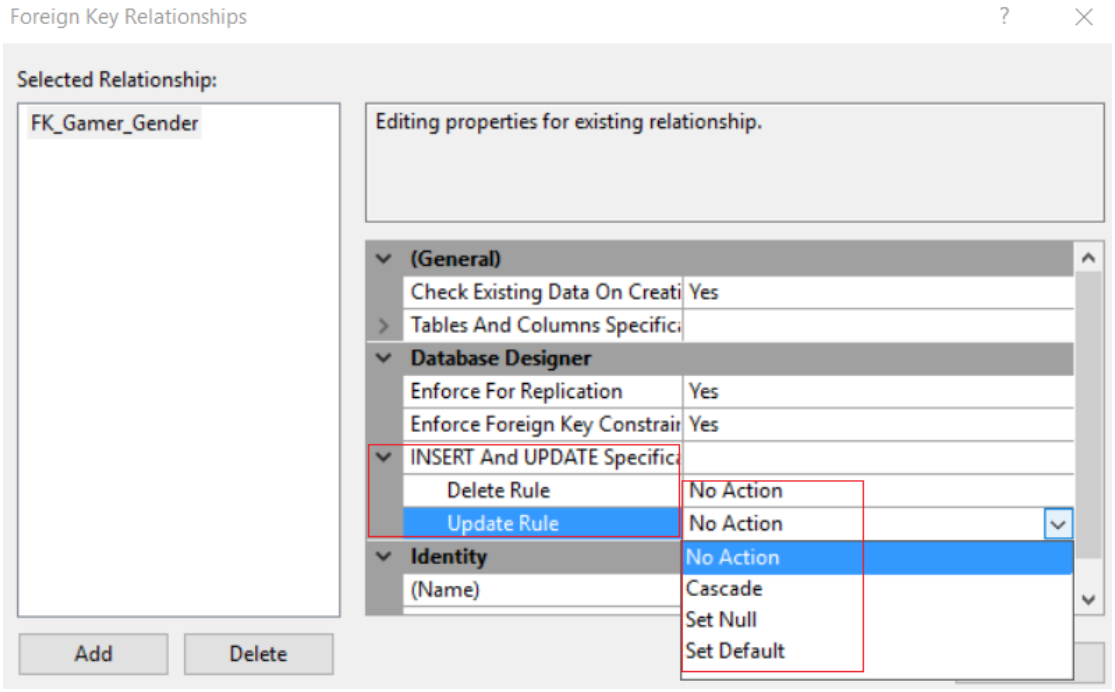

Tables and Columns
Relationship name:
FK_Gamer_Gender
Primary key table:
Gender
Id
Foreign key table:
Gamer
GenderId
OK
Cancel


Foreign Key Relationship
Selected Relationship:
FK_Gamer_Gender*
Editing properties for new relationship. The 'Tables And Columns Specification' property needs to be filled in before the new relationship will be accepted.
(General)
Check Existing Data On Creati
Yes
Tables And Columns Specifica
Database Designer
Enforce For Replication
Yes
Enforce Foreign Key Constrair
Yes
INSERT And UPDATE Specifica
Identity
(Name)
FK_Gamer_Gender
Description
OK
Cancel


Gender *
Id
Gender
Gamer *
Id
Name
Email
GenderId
CreatedDateTime
Age


Choose Name
Enter a name for the diagram:
Diagram_0
OK
Cancel

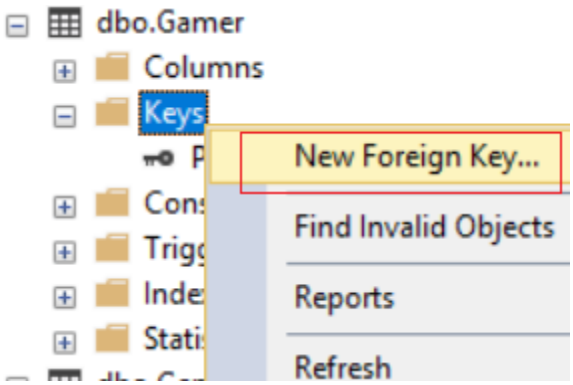


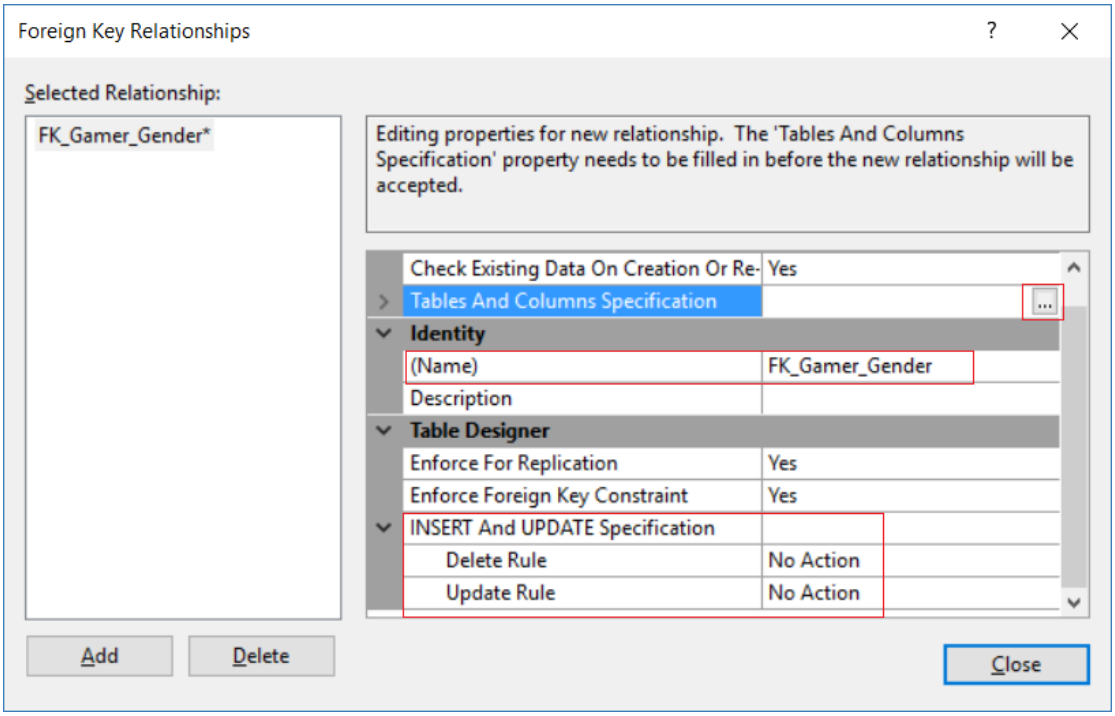


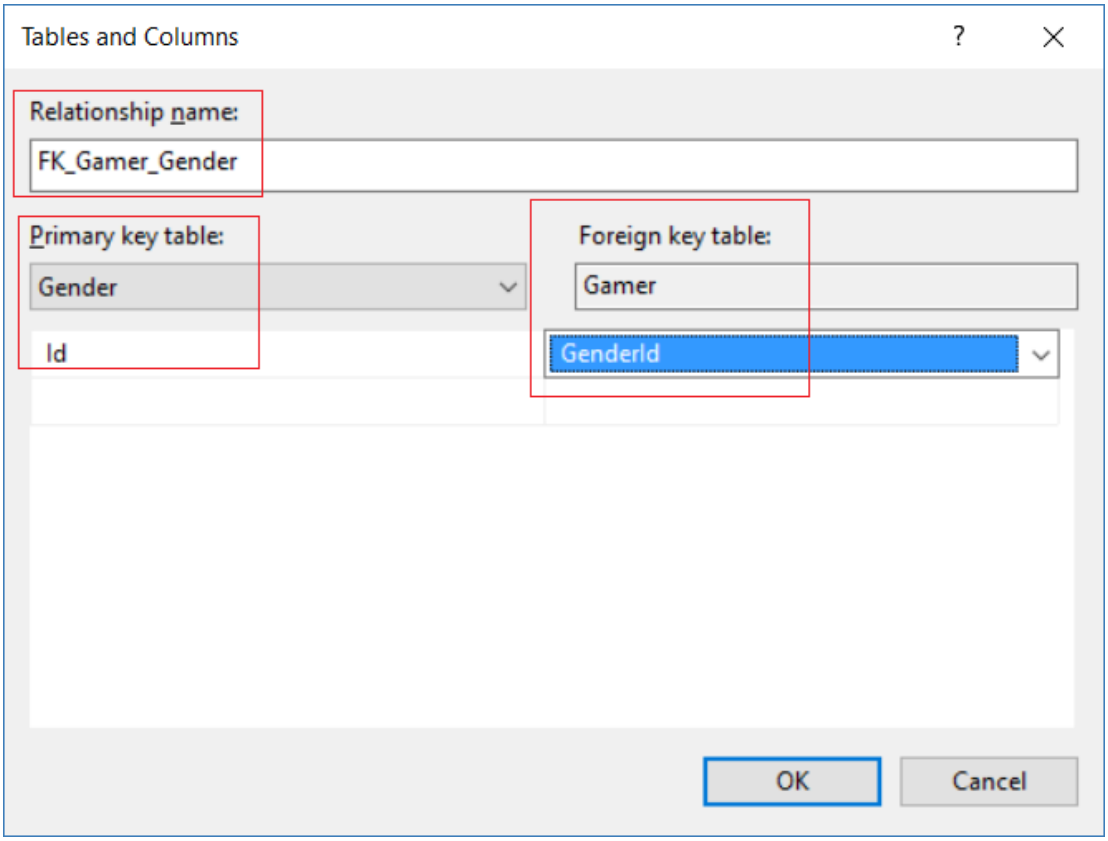
3.1.5.2. Create Referential Integrity constraint (Foreign Key) in Table

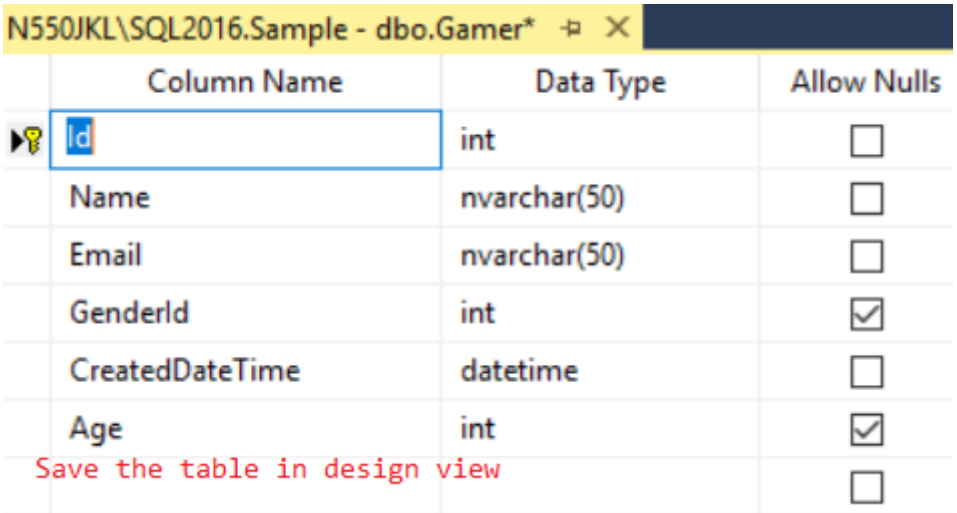
Table Name --> Keys --> Right Click --> New Foreign key
-->
Name:
FK_Gamer_Gender
Tables And Columns Specification --> Select ...
-->
Set Gender , Id --> Map to Gamer, GenderId --> OK
--> Close
--> In Design View, Ctrl+S Save the Design

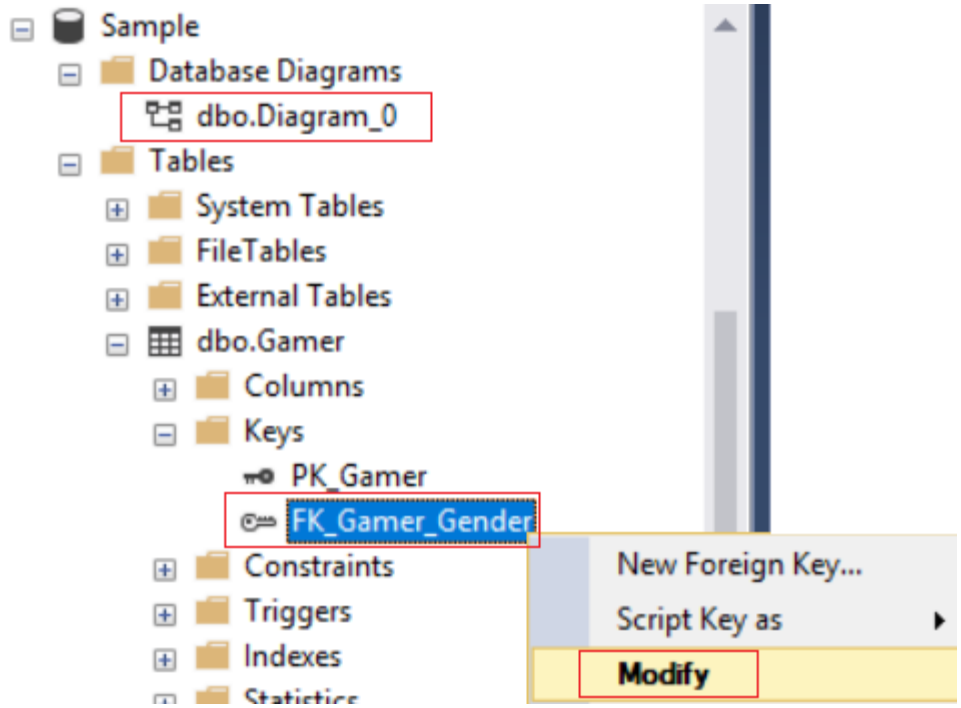
If you want to modify again.
Database Name --> Tables --> Table Name --> Keys --> FK --> Right Click --> Modify -->
Delete Rule / Update Rule --> **No Action / Cascade / Set Null / Set Default**
By default, I set to **No Action**.

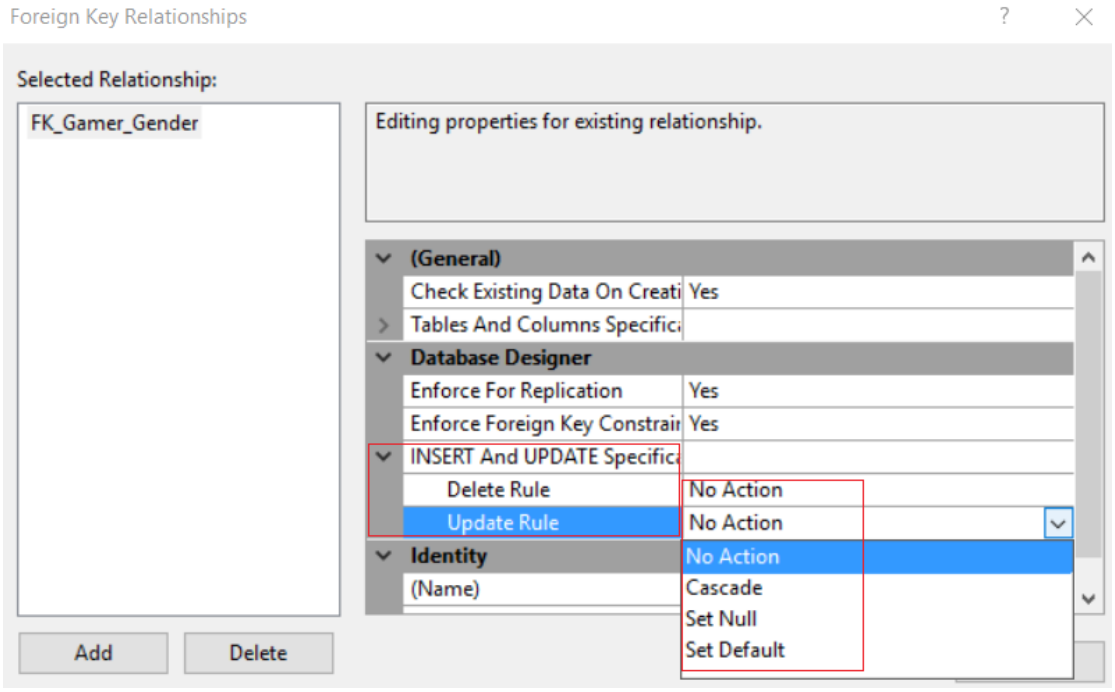





Tables and Columns
Relationship name:
FK_Gamer_Gender
Primary key table:
Gender
Foreign key table:
Gamer
Id
GenderId
OK
Cancel

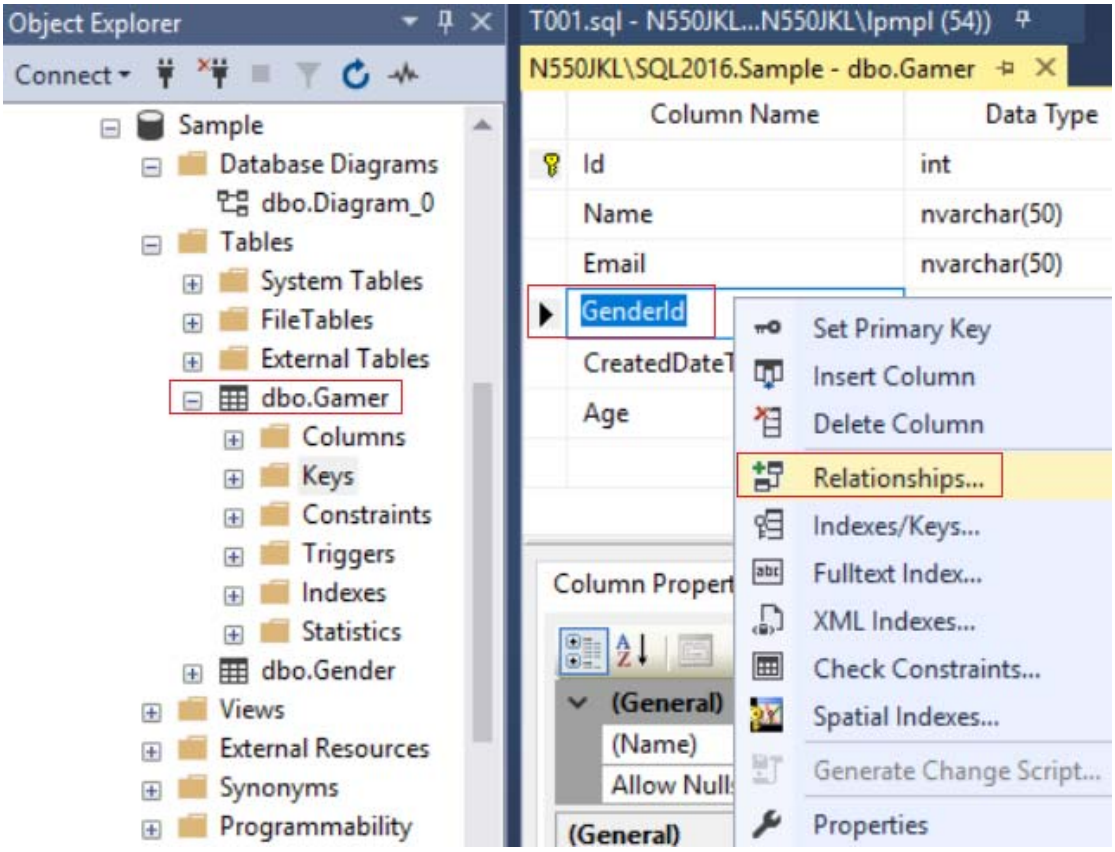

N550JKL\SQL2016.Sample - dbo.Gamer*
Column Name
Data Type
Allow Nulls
Id
int
Name
nvarchar(50)
Email
nvarchar(50)
GenderId
int
CreatedDateTime
datetime
Age
int
Save the table in design view

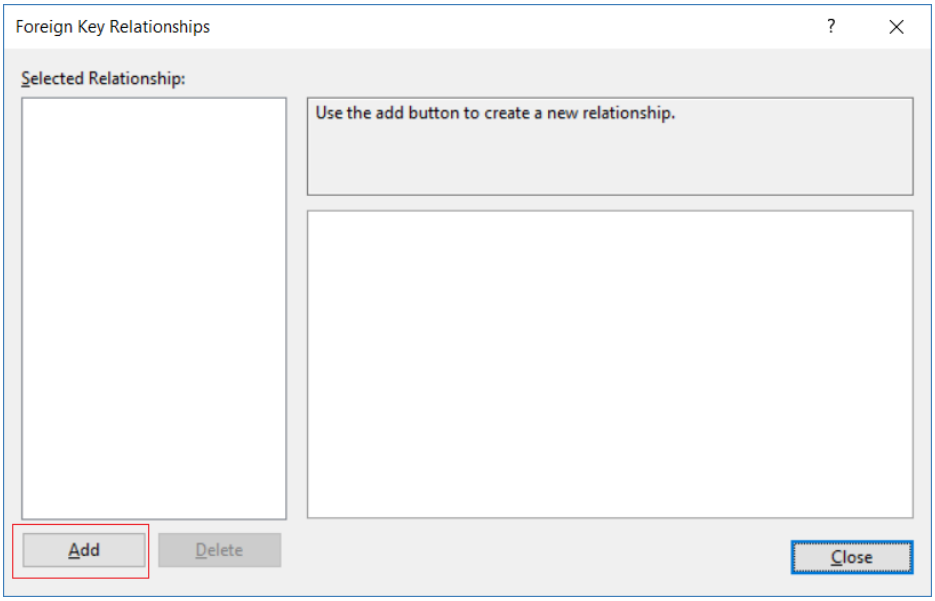


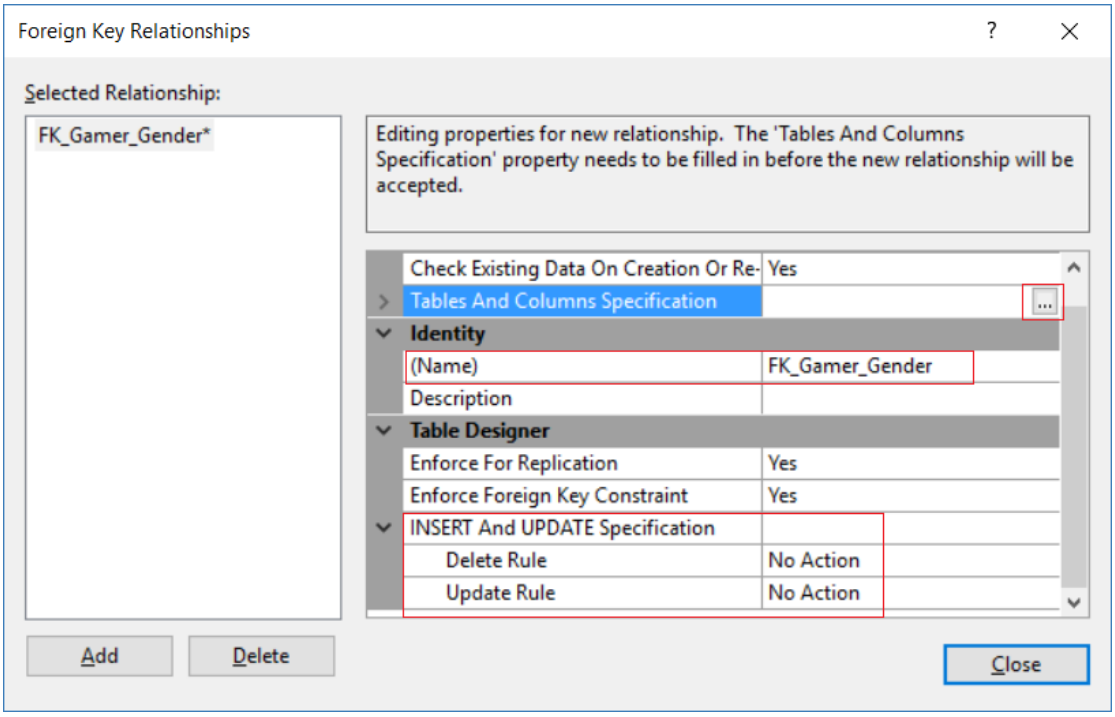


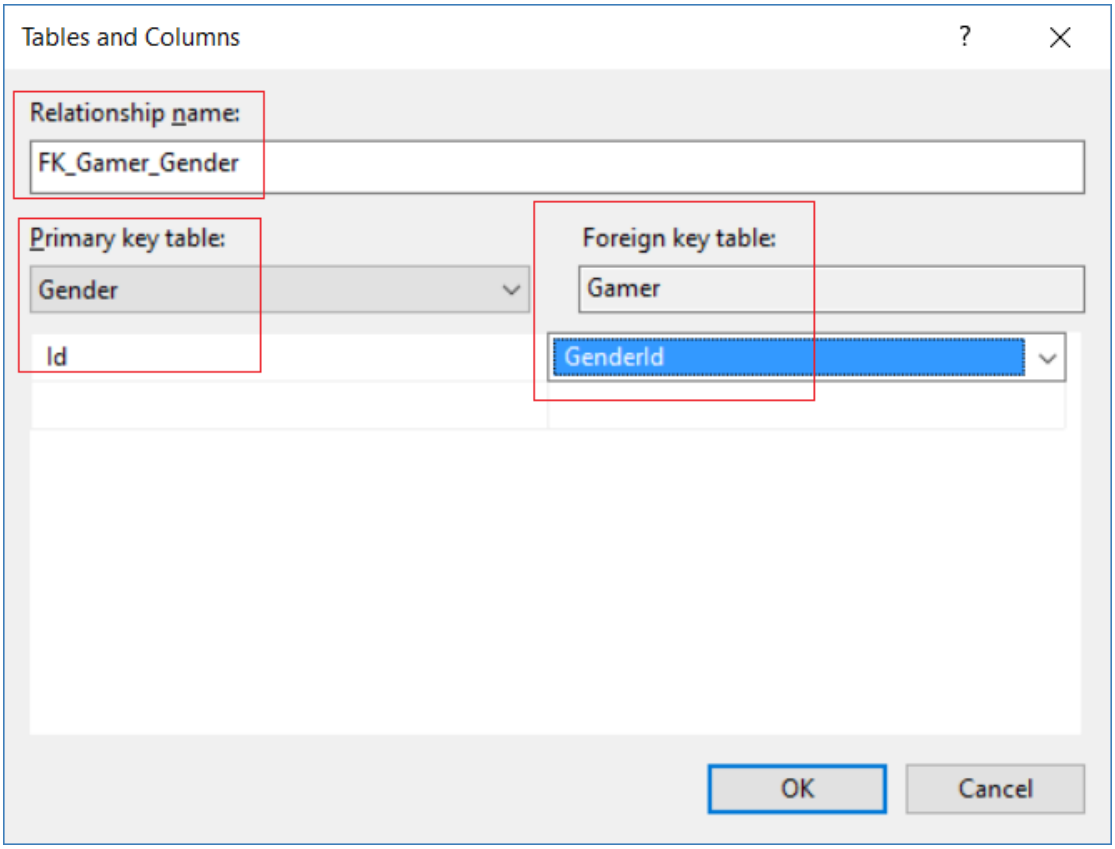
3.1.5.3. Create Referential Integrity constraint (Foreign Key) in Design View

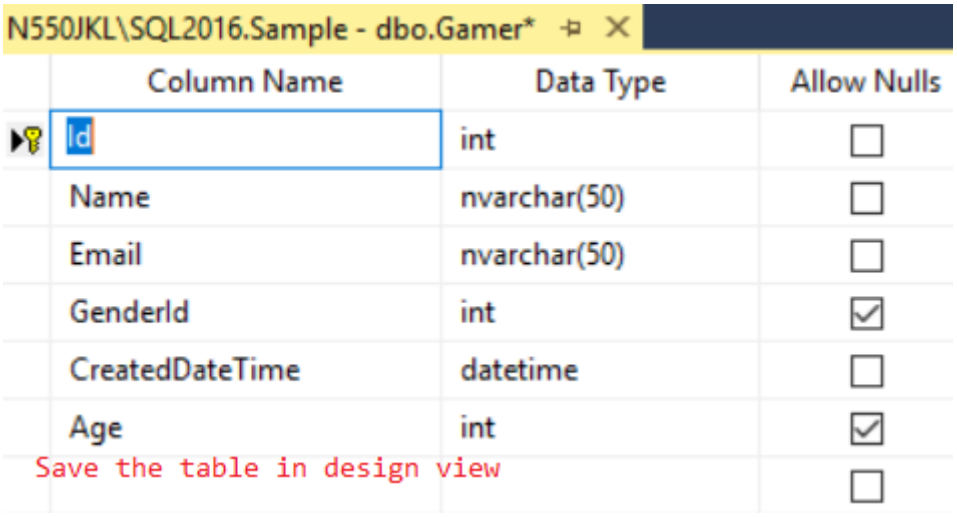
Database Name --> Tables --> Table Name --> Right Click --> Design -->
In the Design View, Select the Column Name --> Right Click --> Relationship
--> Add
--> **Tables and Column Specification**
It will create **Foreign Key Relationship "FK_Gamer_Gender"**
Make sure I connect **[**Gender**].[**Id**]** into **[**Gamer**].[**GenderId**]**
--> Close
--> In Design View, Ctrl + S, Save the Table.

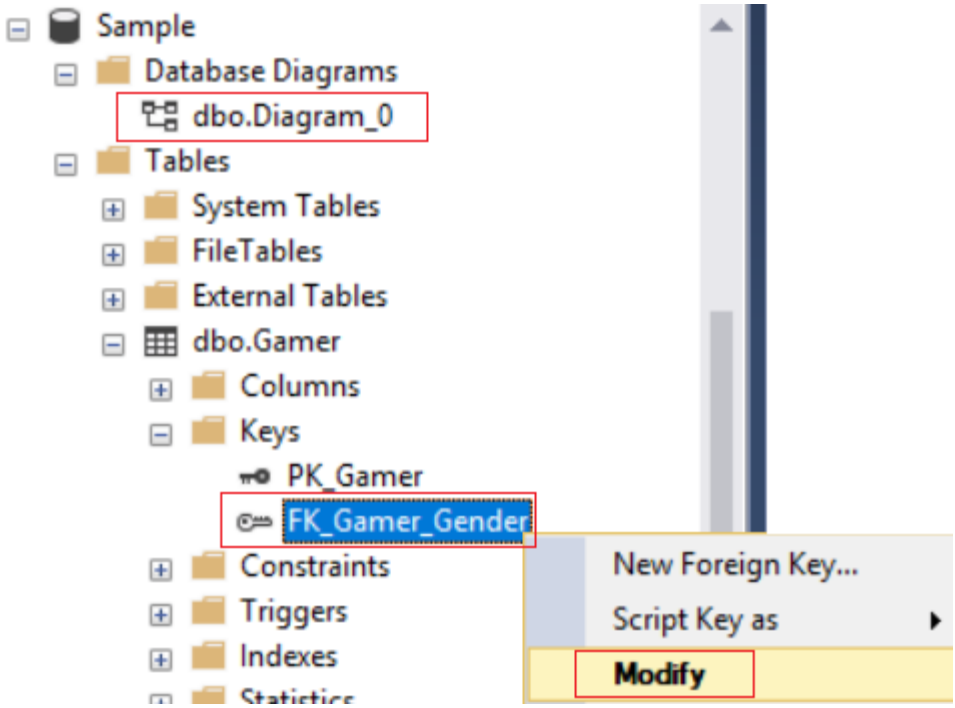

Object Explorer
Connect
Sample
Database Diagrams
dbo.Diagram_0
Tables
System Tables
FileTables
External Tables
dbo.Gamer
Columns
Keys
Constraints
Triggers
Indexes
Statistics
dbo.Gender
Views
External Resources
Synonyms
Programmability
T001.sql - N550JKL...N550JKL\Ipmpl (54))
N550JKL\SQL2016.Sample - dbo.Gamer
Column Name
Data Type
Id
int
Name
nvarchar(50)
Email
nvarchar(50)
GenderId
CreatedDateT
Age
Column Propert
(General)
(Name)
Allow Null
(General)
Set Primary Key
Insert Column
Delete Column
Relationships...
Indexes/Keys...
Fulltext Index...
XML Indexes...
Check Constraints...
Spatial Indexes...
Generate Change Script...
Properties


Foreign Key Relationships
Selected Relationship:
Use the add button to create a new relationship.
Add
Delete
Close


Foreign Key Relationships
Selected Relationship:
FK_Gamer_Gender*
Editing properties for new relationship. The 'Tables And Columns Specification' property needs to be filled in before the new relationship will be accepted.
Check Existing Data On Creation Or Re-
Yes
Tables And Columns Specification
Identity
(Name)
FK_Gamer_Gender
Description
Table Designer
Enforce For Replication
Yes
Enforce Foreign Key Constraint
Yes
INSERT And UPDATE Specification
Delete Rule
No Action
Update Rule
No Action
Add
Delete
Close


Tables and Columns
Relationship name:
FK_Gamer_Gender
Primary key table:
Gender
Id
Foreign key table:
Gamer
GenderId
OK
Cancel





3.1.6. Insert Data to Gamer

Database Name --> Tables --> Table Name --> Right Click --> Edit top 200 rows -->

-->

NULL Name1 1@1.com 1 NULL 21
NULL Name2 2@2.com 2 NULL 22
NULL Name3 3@3.com NULL NULL 23
NULL Name4 4@4.com 1 NULL 24
NULL Name5 5@5.com 2 NULL 25
NULL Name6 6@6.com NULL NULL 26

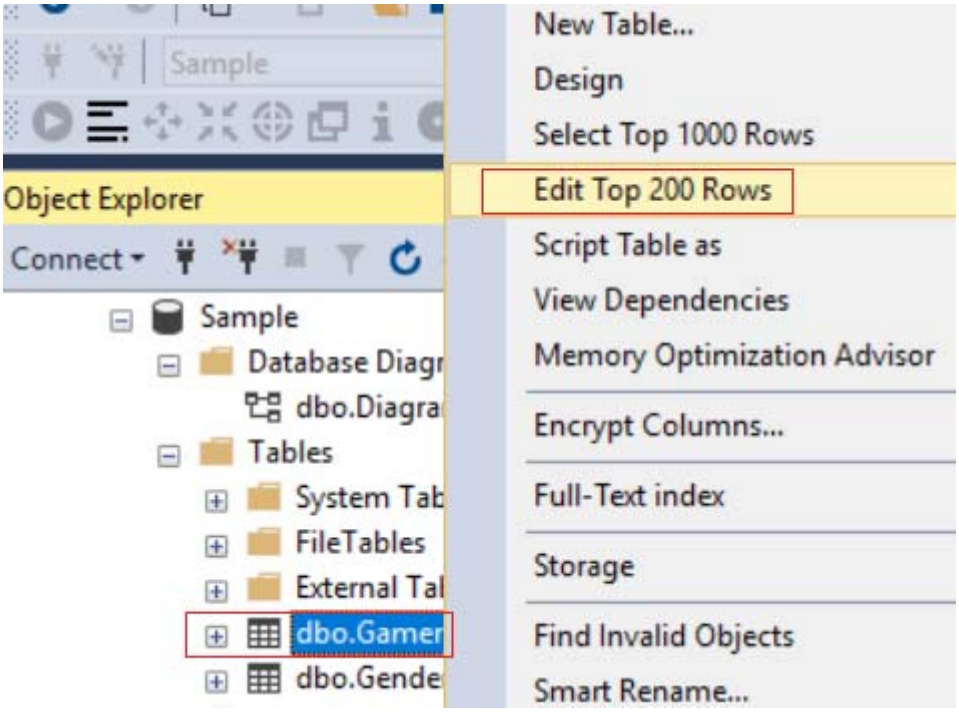
-->

Explanation:

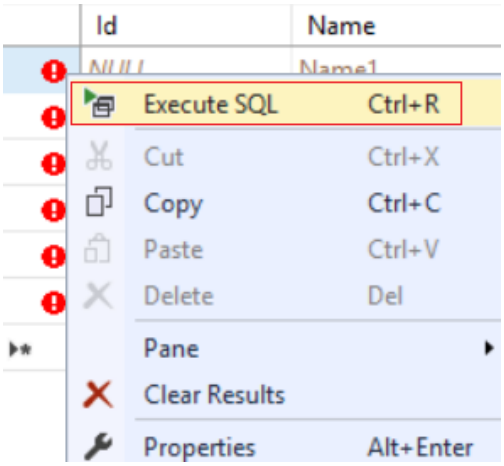
1.

[Id**] [int]** IDENTITY**(1,1) Primary Key NOT NULL,**

When I created table, I set Id is **INT** type,
and I also set Id is **Primary Key**.
Additionally, I set Id is **Identity Column**.
IDENTITY(1,1) means Id will start from 1 (the **first** one is **identity seed**),
and then increase 1 (the **second** one is **identity increment**).
We do not have to set value for **INT Type Identity Column**, because it will be auto generated by SQL server.
Thus, when we insert new data, I only have to provide the rest of data except Id.
2.
Default Value of GenderId is **3**
If I didn't provide any value, it will auto set to 3
3.
Default Value of **CreateDateTime** is **(getutcdate())**
If I didn't provide any value, it will auto set to current UTC Time.



Id	Name	Email	GenderId	CreatedDateTi...	Age
NULL	Name1	1@1.com	1	NULL	21
NULL	Name2	2@2.com	2	NULL	22
NULL	Name3	3@3.com	3	NULL	23
NULL	Name4	4@4.com	1	NULL	24
NULL	Name5	5@5.com	2	NULL	25
NULL	Name6	6@6.com	3	NULL	26



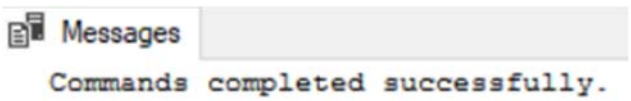
Id	Name	Email	GenderId	CreatedDateTime	Age
1	Name1	1@1.com	1	2017-11-02 12:09:30.283	21
2	Name2	2@2.com	2	2017-11-02 12:09:44.740	22
3	Name3	3@3.com	3	2017-11-02 12:09:53.717	23
4	Name4	4@4.com	1	2017-11-02 12:10:05.517	24
5	Name5	5@5.com	2	2017-11-02 12:10:23.720	25
6	Name6	6@6.com	3	2017-11-02 12:10:43.900	26
NULL	NULL	NULL	NULL	NULL	NULL

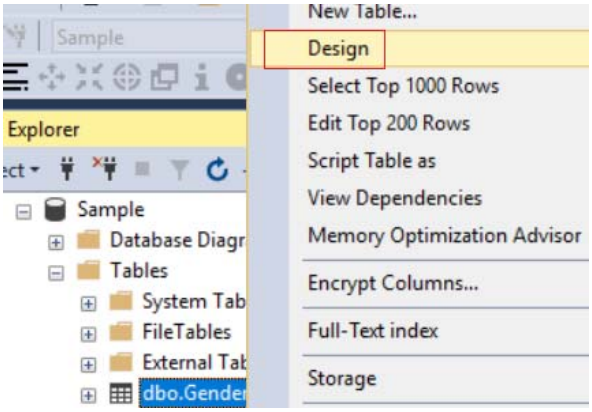
4. Using Query to Create Tables -> Set Default Constraint -> Set Check Constraint -> Set Referential Integrity constraint (Foreign Key)

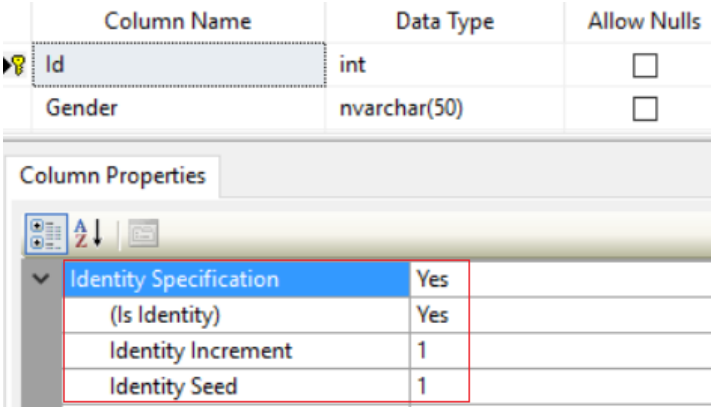
What to learn
- Create Table
- Default Constraint
- Check Constraint
- Identity Column
- Primary Key
- Foreign Key
- Insert

```
------------------------------------------------------------------------------------------------------------
---
--T001_02_00
--Create or ReCreate Database.
USE master;
-- be sure that you're not on the database you want to delete
GO -- Run the prvious command and begins new batch
IF ( EXISTS ( SELECT    [name] ,
                        database_id ,
```

```
                        create_date
              FROM      sys.databases
              WHERE     name = N'Sample' ) )
    BEGIN
        ALTER DATABASE [Sample] SET SINGLE_USER WITH ROLLBACK IMMEDIATE;
        DROP DATABASE [Sample];
    END;
GO -- Run the previous command and begins new batch
CREATE DATABASE [Sample];
GO -- Run the previous command and begins new batch
USE [Sample];
GO -- Run the prvious command and begins new batch

-----------------------------------------------------------------------------------------------------
--T001_02_01
--CreateTable - Gender
CREATE TABLE Gender
(
  Id INT IDENTITY(1, 1)
        PRIMARY KEY
        NOT NULL ,
  --Id INT IDENTITY(1,1)
  --       NOT NULL ,
  [Gender] [NVARCHAR](50) NOT NULL ,
  --CONSTRAINT [PK_Gender] PRIMARY KEY CLUSTERED ( [Id] ASC )
  --  WITH ( PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
  --         ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON ) ON [PRIMARY]
)
    ON
[PRIMARY];
GO -- Run the prvious command and begins new batch
/*
1.
There are 2 ways to set the primary Key
1.1.
--Id INT IDENTITY(1, 1) PRIMARY KEY NOT NULL,
1.2.
--[Id] [INT] IDENTITY(1, 1) NOT NULL ,
--CONSTRAINT [PK_Gender] PRIMARY KEY CLUSTERED ( [Id] ASC )
--WITH ( PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
--        IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
--        ALLOW_PAGE_LOCKS = ON ) ON [PRIMARY]
2.
[Id] [int] IDENTITY(1,1) NOT NULL,
It means Id is the Primary Key and the type is int.
Id will start from 1 (the first one is identity seed),
and then increase 1 (the second one is identity increment)
3.
-- ON [PRIMARY]
When you create database, SQL server will generate
one .MDF(primary data file) and one .LDF(log file)
Sometimes a SQL Server database will include one or more .NDF (secondary data files).
-- ON [PRIMARY]
means create this table on the .MDF(primary data file).
*/
```








```
--------------------------------------------------------------------------------------------------------
--T001_02_02
--Insert Data to Gender
INSERT  Gender
VALUES  ( N'Male' );
SET IDENTITY_INSERT Gender ON;
INSERT  Gender
        ( Id, Gender )
VALUES  ( 2, N'Female' );
INSERT  [dbo].Gender
        ( Id, Gender )
VALUES  ( 3, N'Unknow' );
SET IDENTITY_INSERT Gender OFF;
GO -- Run the prvious command and begins new batch

SELECT *
FROM Gender;
GO -- Run the prvious command and begins new batch
/*
1.
--INSERT  [dbo].[Gender]
--VALUES  ( N'Male' );
You do not have to provide value for identity column
because it is auto generated.
2.
You do not have to provide value for identity column
because it is auto generated.
If you want to provide value for identity column,
then you have to set IDENTITY_INSERT is ON.
--SET IDENTITY_INSERT [TableName] ON;
--INSERT ...
--SET IDENTITY_INSERT [TableName] OFF;
3.
--SELECT *
--FROM Gender;
```

```
* means all columns
Get all Columns from Gender Table.
*/
```

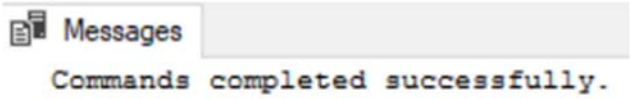
Messages

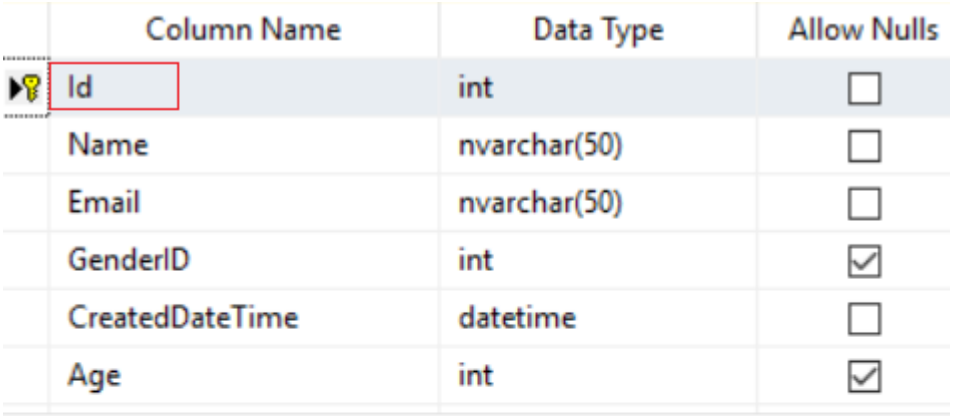
```
(1 row affected)

(1 row affected)

(1 row affected)
```

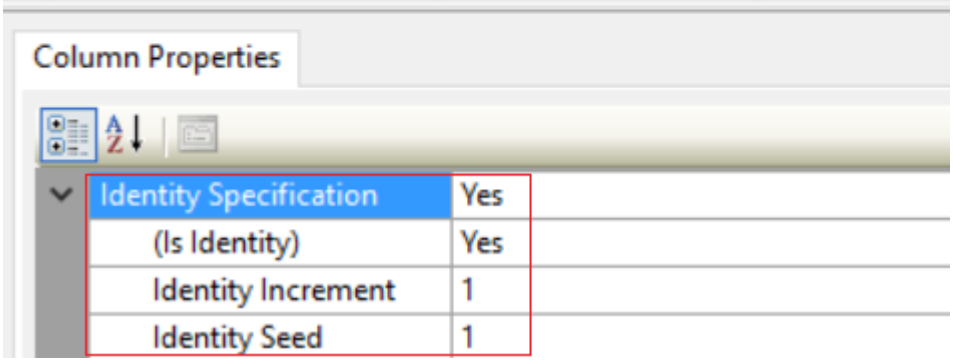
	Id	Gender
▶	1	Male
	2	Female
	3	Unknow
*	NULL	NULL

```
-----------------------------------------------------------------------------------------------------
--T001_02_03
--CreateTable - Gamer
CREATE TABLE Gamer
(
  Id INT IDENTITY(1, 1)
        NOT NULL ,
      --Id INT IDENTITY(1, 1)
      --           PRIMARY KEY
      --           NOT NULL ,
  [Name] NVARCHAR(50) NOT NULL ,
  Email NVARCHAR(50) NOT NULL ,
  GenderId INT NULL ,
  CreatedDateTime DATETIME NOT NULL ,
  Age INT NULL ,
 CONSTRAINT [PK_Gamer_1] PRIMARY KEY CLUSTERED ( [Id] ASC )
   WITH ( PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
          ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON ) ON [PRIMARY]
)
   ON
[PRIMARY];
GO -- Run the prvious command and begins new batch
/*
1.
There are 2 ways to set the primary Key
1.1.
--Id INT IDENTITY(1, 1) PRIMARY KEY NOT NULL,
1.2.
--[Id] [INT] IDENTITY(1, 1) NOT NULL ,
--CONSTRAINT [PK_Gamer_1] PRIMARY KEY CLUSTERED ( [Id] ASC )
--    WITH ( PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
--           IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
--           ALLOW_PAGE_LOCKS = ON ) ON [PRIMARY]
2.
[Id] [int] IDENTITY(1,1) NOT NULL,
```

```
It means Id is the Primary Key and the type is int.
Id will start from 1 (the first one is identity seed),
and then increase 1 (the second one is identity increment)
3.
-- ON [PRIMARY]
When you create database, SQL server will generate
one .MDF(primary data file) and one .LDF(log file)
Sometimes a SQL Server database will include one or more .NDF (secondary data files).
-- ON [PRIMARY]
means create this table on the .MDF(primary data file).
*/
```








```
-----------------------------------------------------------------------------------------------
--T001_02_04
--dbo.Gamer - Default Constraint
-------------------------------------------------------------------------
--T001_02_04_01
--Altering an existing column to add a default constraint.
ALTER TABLE Gamer
ADD  CONSTRAINT DF_Gamer_GenderId
DEFAULT ((3)) FOR [GenderId];
ALTER TABLE Gamer
ADD  CONSTRAINT [DF_Gamer_CreatedDateTime]
DEFAULT (GETUTCDATE()) FOR [CreatedDateTime];
GO -- Run the prvious command and begins new batch
-------------------------------------------------------------------------
--T001_02_04_02
--Adding a new column, with default value, to an existing table
ALTER TABLE Gamer
ADD GenderId2 INT NULL
```

```
CONSTRAINT DF_Gamer_GenderId2 DEFAULT ((3));
GO -- Run the prvious command and begins new batch
--------------------------------------------------------------------------
--T001_02_04_03
--Check the default constraint.
SELECT  *
FROM    sys.objects
WHERE   type_desc LIKE '%CONSTRAINT'
        AND OBJECT_NAME(object_id) = 'DF_Gamer_GenderId2';
GO -- Run the prvious command and begins new batch
--------------------------------------------------------------------------
--T001_02_04_04
--delete the default constraint if it exists.
IF OBJECT_ID('DF_Gamer_GenderId2', 'D') IS NOT NULL
    BEGIN
        ALTER TABLE Gamer
        DROP CONSTRAINT DF_Gamer_GenderId2;
    END;
GO -- Run the prvious command and begins new batch
/*
Constraint Object Types:
C = CHECK constraint
D = DEFAULT (constraint or stand-alone)
F = FOREIGN KEY constraint
PK = PRIMARY KEY constraint
R = Rule (old-style, stand-alone)
UQ = UNIQUE constraint
*/
--------------------------------------------------------------------------
--T001_02_04_05
--Delete the column
ALTER TABLE Gamer
DROP COLUMN GenderId2;
/*
1.
Default Constraint Syntax1:
--ALTER TABLE {TableName}
--ADD  CONSTRAINT {DFConstraintName}
--DEFAULT {DefaultValue} FOR {ColumnName};
Altering an existing column to add a default constraint.
In TableName,
Add a default constraint called DFConstraintName,
The default value of ColumnName is DefaultValue.
When the column has DEFAULT CONSTRAINT,
then we do not have to provide value for the column.
1.1.
E.g.
--ALTER TABLE Gamer
--ADD  CONSTRAINT DF_Gamer_GenderId
--DEFAULT ((2)) FOR [GenderId];
In Gamer Table,
Add a default constraint called DF_Gamer_GenderId,
The default value of GenderId Column is 2.
1.2.
E.g.
--ALTER TABLE Gamer
--ADD  CONSTRAINT [DF_Gamer_CreatedDateTime]
--DEFAULT (GETUTCDATE()) FOR [CreatedDateTime];
In Gamer Table,
Add a default constraint called DF_Gamer_CreatedDateTime,
The default value of CreatedDateTime Column is GETUTCDATE().
-------------------------------------
2.
```

```
Default Constraint Syntax2:
--ALTER TABLE { TableName }
--ADD { ColumnName } { DataType } { NULL | NOT NULL }
--CONSTRAINT { DFConstraintName } DEFAULT { DefaultValue }
Adding a new column, with default value, to an existing table.
In TableName,
Add a new column called ColumnName,
its type is DataType,
which can be NULL | NOT NULL
Add a default constraint called DFConstraintName,
The default value of ColumnName is DefaultValue.
When the column has DEFAULT CONSTRAINT,
then we do not have to provide value for the column.
2.1.
E.g.
--ALTER TABLE Gamer
--ADD GenderId2 INT NULL
--CONSTRAINT DF_Gamer_GenderId2 DEFAULT ((3))
In Gamer Table,
Add a new column called "GenderId2",
its type is "INT",
which can be NULL.
Add a default constraint called "DF_Gamer_GenderId2",
The default value of "GenderId2" is "3".
-------------------------------------
3.
Drop Default Constraint Syntax:
--ALTER TABLE {TableName}
--DROP CONSTRAINT {DFConstraintName}
In {TableName} TABLE
Drop the constraint called DFConstraintName.
3.1.
E.g.
--ALTER TABLE Gamer
--DROP CONSTRAINT DF_Gamer_GenderId2
In Gamer TABLE
Drop the constraint called DF_Gamer_GenderId2.
-------------------------------------
4.
Drop Column Syntax:
--ALTER TABLE {TableName}
--DROP COLUMN {ColumnName};
In {TableName} TABLE
Drop the column called {ColumnName}
4.1.
E.g.
--ALTER TABLE Gamer
--DROP COLUMN GenderId2;
In Gamer TABLE
Drop the column called GenderId2
-------------------------------------
5.
--IF OBJECT_ID('DF_Gamer_GenderId2', 'D') IS NOT NULL
Reference:
https://stackoverflow.com/questions/2499332/how-to-check-if-a-constraint-exists-in-sql-server
Constraint Object Types:
C = CHECK constraint
D = DEFAULT (constraint or stand-alone)
F = FOREIGN KEY constraint
PK = PRIMARY KEY constraint
R = Rule (old-style, stand-alone)
UQ = UNIQUE constraint
*/
```

Messages

```
Commands completed successfully.
```

Column Name	Data Type	Allow Nulls
Id	int	☐
Name	nvarchar(50)	☐
Email	nvarchar(50)	☐
GenderId	int	☑
CreatedDateTime	datetime	☐
Age	int	☑
		☐

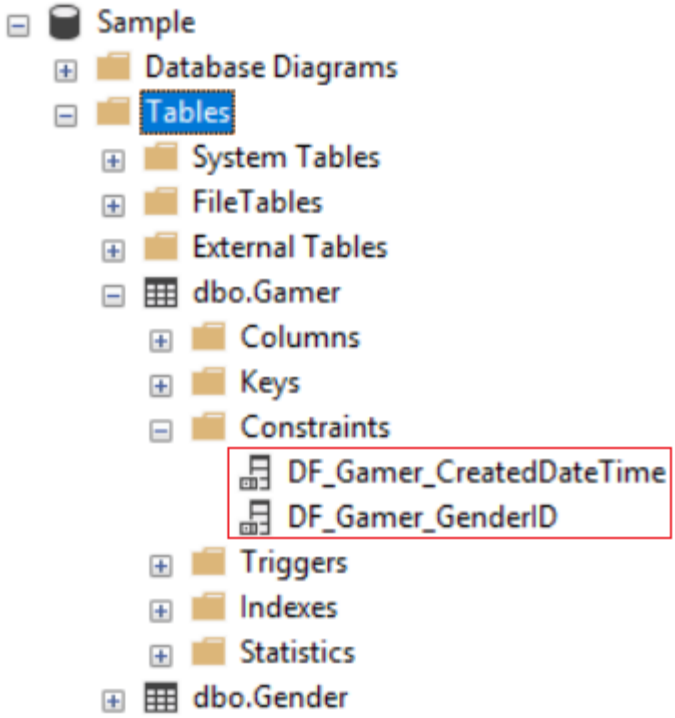
Column Properties

(General)	
(Name)	GenderId
Allow Nulls	Yes
Data Type	int
Default Value or Binding	((3))

Column Name	Data Type	Allow Nulls
Id	int	☐
Name	nvarchar(50)	☐
Email	nvarchar(50)	☐
GenderId	int	☑
CreatedDateTime	datetime	☐
Age	int	☑
		☐

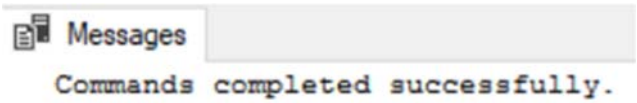
Column Properties

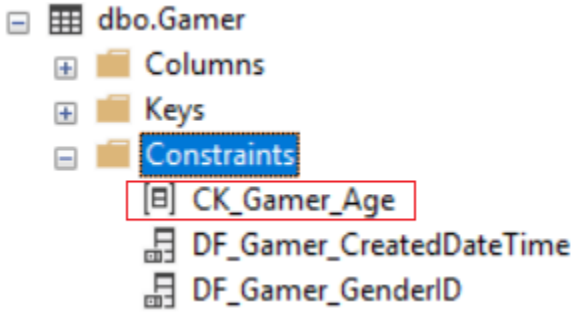
(General)	
(Name)	CreatedDateTime
Allow Nulls	No
Data Type	datetime
Default Value or Binding	(getutcdate())

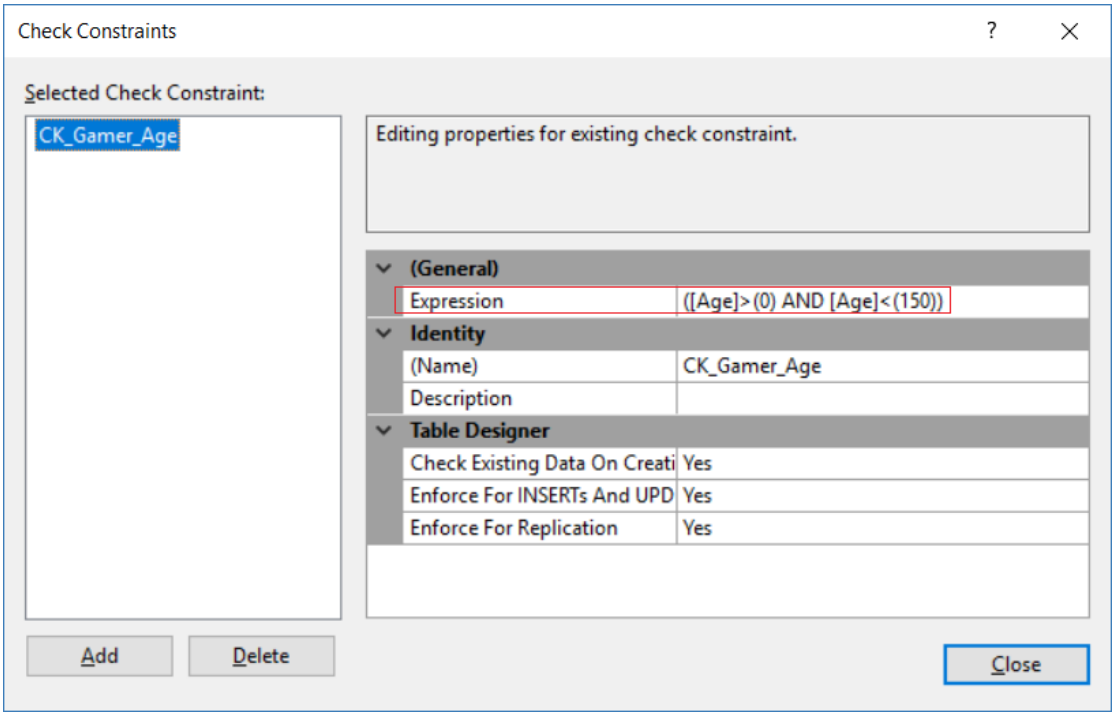



```
--------------------------------------------------------------------------------------------------------
--T001_02_05
--Gamer - Check Constraint
--------------------------------------------------------------------------------
--T001_02_05_01
--ALTER TABLE [dbo].[Gamer]  WITH CHECK ADD  CONSTRAINT [CK_Gamer_Age] CHECK  (([Age]>(0) AND
[Age]<(150)));
--ALTER TABLE [dbo].[Gamer] CHECK CONSTRAINT [CK_Gamer_Age];
--GO -- Run the prvious command and begins new batch
--------------------------------------------------------------------------------
--T001_02_05_02
--Add Check constraint
ALTER TABLE Gamer
ADD CONSTRAINT CK_Gamer_Age CHECK (Age > 0 AND Age < 150);
GO -- Run the prvious command and begins new batch
INSERT  Gamer
        ( [Name], [Email], [Age] )
VALUES  ( N'NameAA', N'AA@AA.com', 180 );
--Fail to insert, because of the check constraint.
SELECT  *
FROM    Gamer;
--------------------------------------------------------------------------------
--T001_02_05_03
--Get the information of the check constraint.
SELECT  *
FROM    INFORMATION_SCHEMA.TABLE_CONSTRAINTS
WHERE   CONSTRAINT_NAME = 'CK_Gamer_Age';
--------------------------------------------------------------------------------
--T001_02_05_04
--https://stackoverflow.com/questions/2499332/how-to-check-if-a-constraint-exists-in-sql-server
--Create or Recreate Check constraint
IF ( EXISTS ( SELECT    *
              FROM      INFORMATION_SCHEMA.TABLE_CONSTRAINTS
              WHERE     CONSTRAINT_NAME = 'CK_Gamer_Age' ) )
   BEGIN
       ALTER TABLE Gamer
```

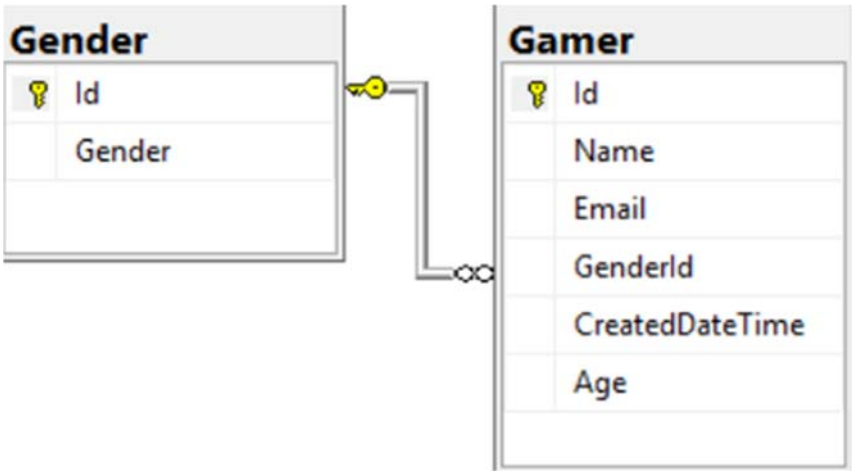
```
        DROP CONSTRAINT CK_Gamer_Age;
    END;
GO -- Run the previous command and begins new batch
ALTER TABLE Gamer
ADD CONSTRAINT CK_Gamer_Age CHECK (Age > 0 AND Age < 150);
GO -- Run the prvious command and begins new batch
/*
1.
Add Check Constraint Syntax:
--ALTER TABLE {TableName}
--ADD CONSTRAINT {CKConstraintName} CHECK {Condition};
In {TableName} Table,
Add a check constraint called {CKConstraintName},
It must fullfill the {Condition}.
1.1.
E.g.
--ALTER TABLE Gamer
--ADD CONSTRAINT CK_Gamer_Age CHECK (Age > 0 AND Age < 150);
In Gamer Table,
Add a check constraint called CKConstraintName,
The value in Age Column must be between 0 to 150.
2.
--INSERT  Gamer
--        ( [Name], [Email], [Age] )
--VALUES  ( N'NameAA', N'AA@AA.com', 180 );
Because the Age must be between 0 to 150.
this will fail.
3.
ALTER TABLE {TableName}
DROP CONSTRAINT {CKConstraintName};
In {TableName} Table,
Drop the check constraint called {CKConstraintName},
3.1.
E.g.
--ALTER TABLE Gamer
--DROP CONSTRAINT CK_Gamer_Age;
In Gamer Table,
Drop the check constraint called CK_Gamer_Age,
4.
--SELECT  *
--FROM    Gamer;
* means all columns
Get all columns from Gamer Table.
*/
```



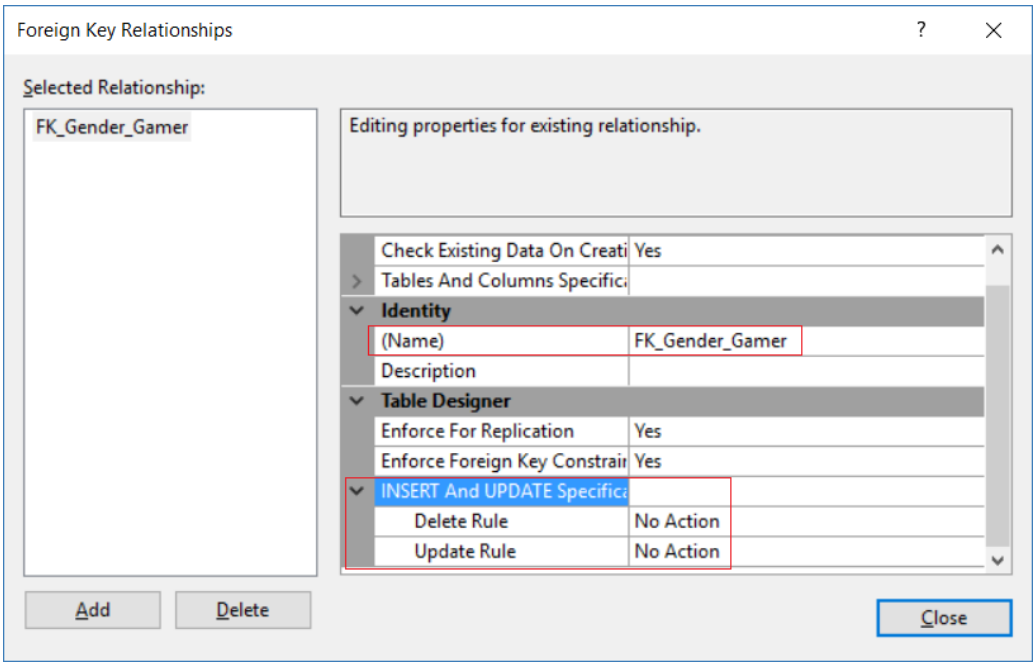


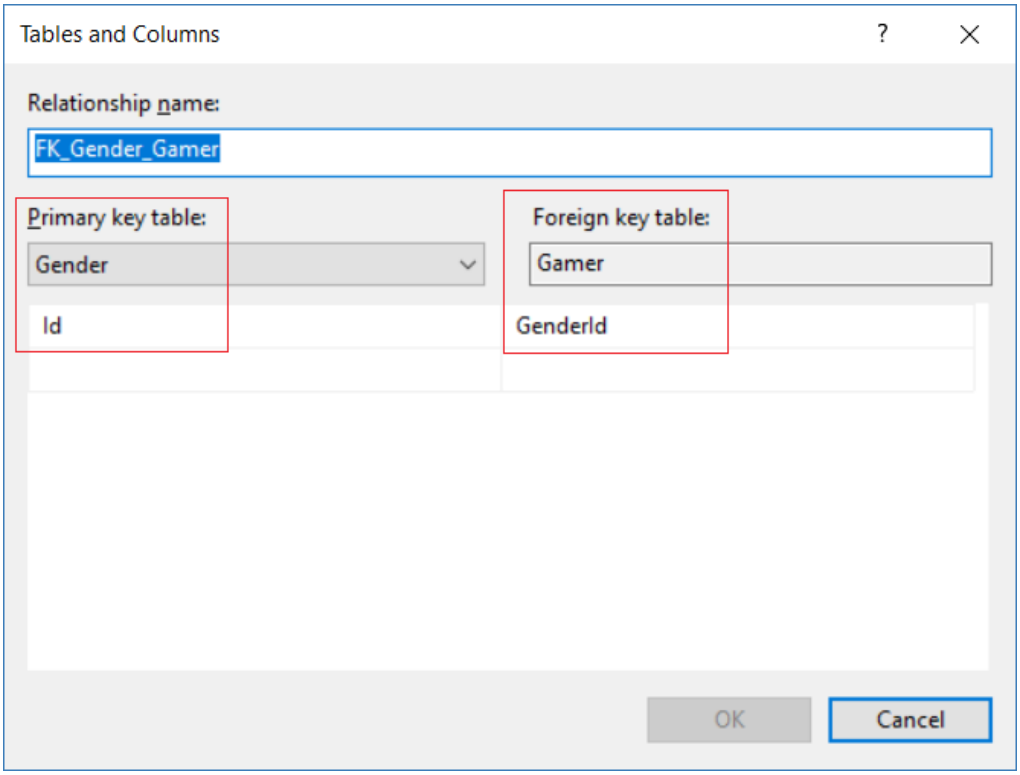



```
--------------------------------------------------------------------------------------------------------
--T001_02_06
--Gamer - Referential Integrity constraint (Foreign Key)
--------------------------------------------------------------------------------
--T001_02_06_01
--ALTER TABLE [dbo].[Gamer]  WITH CHECK ADD  CONSTRAINT [FK_Gender_Gamer] FOREIGN KEY([Id])
--REFERENCES [dbo].[Gamer] ([Id])
--ALTER TABLE [dbo].[Gamer] CHECK CONSTRAINT [FK_Gender_Gamer]
--GO -- Run the prvious command and begins new batch
--------------------------------------------------------------------------------
--T001_02_06_02
ALTER TABLE Gamer ADD CONSTRAINT FK_Gender_Gamer
FOREIGN KEY (GenderId) REFERENCES Gender(Id)
ON DELETE NO ACTION;
GO -- Run the prvious command and begins new batch
/*
1.
--ALTER TABLE Gamer ADD CONSTRAINT FK_Gender_Gamer
--FOREIGN KEY (GenderId) REFERENCES Gender(Id)
--ON DELETE NO ACTION;
1.1.
Create a FOREIGN KEY CONSTRAINT "FK_Gender_Gamer" in order to
connect the  [Gamer].[GenderId]   column into  [Gender].[Id]
Foreign keys are used to enforce database integrity.
The values that you enter into the foreign key column,
has to be one of the values contained in the table it points to.
1.2.
You may delete
--ON DELETE NO ACTION;
because, Foreign key is "ON DELETE NO ACTION" by default setting.
This means when you delete valueA in Gender Table,
If the valueA is still used in Gamer table,
then do nothing which means
valueA can not be deleted if it is still used in other table.
2.
--ALTER TABLE Gamer ADD CONSTRAINT FK_Gender_Gamer
--FOREIGN KEY (GenderId) REFERENCES Gender(Id);
This is easier way to create CHECK CONSTRAINT
*/
```


Gender
Id
Gender
Gamer
Id
Name
Email
GenderId
CreatedDateTime
Age


dbo.Gamer
Columns
Keys
PK_Gamer_1
FK_Gender_Gamer


Foreign Key Relationships
Selected Relationship:
FK_Gender_Gamer
Editing properties for existing relationship.
Check Existing Data On Creati Yes
Tables And Columns Specifica
Identity
(Name) FK_Gender_Gamer
Description
Table Designer
Enforce For Replication Yes
Enforce Foreign Key Constrair Yes
INSERT And UPDATE Specifica
Delete Rule No Action
Update Rule No Action
Add
Delete
Close




```
--------------------------------------------------------------------------------------------------------
--T001_02_07
--INSERT Data to Gamer
SET IDENTITY_INSERT [Gamer] ON;
INSERT  Gamer
        ( Id ,
          [Name] ,
          Email ,
          GenderId ,
          CreatedDateTime ,
          Age
        )
VALUES  ( 1 ,
          N'Name1' ,
          N'1@1.com' ,
          1 ,
          CAST(N'2017-09-01T18:05:03.127' AS DATETIME) ,
          21
        );
INSERT  [dbo].[Gamer]
        ( Id ,
          [Name] ,
          Email ,
          GenderId ,
          CreatedDateTime ,
          Age
        )
VALUES  ( 2 ,
          N'Name5' ,
          N'2@2.com' ,
          2 ,
          CAST(N'2017-09-01T18:05:18.443' AS DATETIME) ,
          22
```

```
        );
INSERT  [dbo].[Gamer]
        ( Id ,
          [Name] ,
          Email ,
          GenderId ,
          CreatedDateTime ,
          Age
        )
VALUES  ( 3 ,
          N'Name3' ,
          N'3@3.com' ,
          3 ,
          CAST(N'2017-09-01T18:05:41.070' AS DATETIME) ,
          23
        );
SET IDENTITY_INSERT [dbo].[Gamer] OFF;
INSERT  Gamer
        ( [Name], Email, GenderId, Age )
VALUES  ( N'Name4', N'4@4.com', 1, 24 );
INSERT  Gamer
VALUES  ( N'Name5', N'5@5.com', 2,
          CAST(N'2017-09-01T18:05:03.127' AS DATETIME), 25 );
INSERT  Gamer
        ( [Name], [Email], [Age] )
VALUES  ( N'Name6', N'6@6.com', 26 );
GO -- Run the prvious command and begins new batch
SELECT  *
FROM    Gamer;
GO -- Run the prvious command and begins new batch
/*
1.
--INSERT  Gamer
--        ( [Name], [Email], [Age] )
--VALUES  ( N'Name6', N'6@6.com', 26 );
1.1.
You do not have to provide value for identity column [Id]
because it is auto generated.
When we create the table, we set the id is IDENTITY(1,1)
--Id INT IDENTITY(1,1) NOT NULL,
It means Id is the Primary Key and the type is int.
Id will start from 1 (the first one is identity seed),
and then increase 1 (the second one is identity increment)
1.2.
Previously, We set the default constraint,
thus, we do not have to provide value for [GenderId] , [CreatedDateTime].
The default value of [GenderId] is 3
The default value of [CreatedDateTime] is GETUTCDATE()
1.3.
--ALTER TABLE Gamer
--ADD CONSTRAINT CK_Gamer_Age CHECK (Age > 0 AND Age < 150);
Previously, We set the Check constraint,
The Age must be between 0 to 150.
2.
You do not have to provide value for identity column
because it is auto generated.
If you want to provide value for identity column,
then you have to set IDENTITY_INSERT is ON.
--SET IDENTITY_INSERT [TableName] ON;
--INSERT ...
--SET IDENTITY_INSERT [TableName] OFF;
```

*/

	Id	Name	Email	GenderId	CreatedDateTime	Age
1	1	Name1	1@1.com	1	2017-09-01 18:05:03.127	21
2	2	Name5	2@2.com	2	2017-09-01 18:05:18.443	22
3	3	Name3	3@3.com	3	2017-09-01 18:05:41.070	23
4	4	Name4	4@4.com	1	2017-11-02 15:11:38.817	24
5	5	Name5	5@5.com	2	2017-09-01 18:05:03.127	25
6	6	Name6	6@6.com	3	2017-11-02 15:11:38.817	26

```
--=============================================================================================================
--T001_03
--Unique Key Constraint
-------------------------------------------------------------------------------------------------------------
--T001_03_01
--Add the unique constraint
ALTER TABLE Gamer
ADD CONSTRAINT UQ_Gamer_Email UNIQUE(Email);
--Email must be unique
SELECT  *
FROM    Gender;
SELECT  *
FROM    Gamer;
GO -- Run the prvious command and begins new batch
INSERT  Gamer
        ( [Name], [Email], [Age] )
VALUES  ( N'Name7', N'7@7.com', 27 );
--insert Name8 will be fail, because Email must be unique.
INSERT  Gamer
        ( [Name], [Email], [Age] )
VALUES  ( N'Name8', N'7@7.com', 28 );
GO -- Run the prvious command and begins new batch
SELECT  *
FROM    Gender;
SELECT  *
FROM    Gamer;
GO -- Run the prvious command and begins new batch
```

```
(3 rows affected)

(6 rows affected)

(1 row affected)
Msg 2627, Level 14, State 1, Line 714
Violation of UNIQUE KEY constraint 'UQ_Gamer_Email'. Cannot insert duplicate key in object 'dbo.Gamer'. The duplicate key value is (7@7.com).
The statement has been terminated.

(3 rows affected)

(7 rows affected)
```

	Id	Gender
1	1	Male
2	2	Female
3	3	Unknow

	Id	Name	Email	GenderId	CreatedDateTime	Age
1	1	Name1	1@1.com	1	2017-09-01 18:05:03.127	21
2	2	Name5	2@2.com	2	2017-09-01 18:05:18.443	22
3	3	Name3	3@3.com	3	2017-09-01 18:05:41.070	23
4	4	Name4	4@4.com	1	2017-11-02 16:19:39.577	24
5	5	Name5	5@5.com	2	2017-09-01 18:05:03.127	25
6	6	Name6	6@6.com	3	2017-11-02 16:19:39.577	26

	Id	Gender
1	1	Male
2	2	Female
3	3	Unkn...

	Id	Name	Email	GenderId	CreatedDateTime	Age
1	1	Name1	1@1.com	1	2017-09-01 18:05:03.127	21
2	2	Name5	2@2.com	2	2017-09-01 18:05:18.443	22
3	3	Name3	3@3.com	3	2017-09-01 18:05:41.070	23
4	4	Name4	4@4.com	1	2017-11-02 16:19:39.577	24
5	5	Name5	5@5.com	2	2017-09-01 18:05:03.127	25
6	6	Name6	6@6.com	3	2017-11-02 16:19:39.577	26
7	7	Name7	7@7.com	3	2017-11-02 16:27:12.400	27

```
-----------------------------------------------------------------------------------------------------------
--T001_03_02
--Get the information of the unique constraint
SELECT  *
FROM    INFORMATION_SCHEMA.TABLE_CONSTRAINTS
WHERE   CONSTRAINT_NAME = 'UQ_Gamer_Email';
/*
Reference:
https://stackoverflow.com/questions/2499332/how-to-check-if-a-constraint-exists-in-sql-server
*/
```

	CONSTRAINT_CATALOG	CONSTRAINT_SCHEMA	CONSTRAINT_NAME	TABLE_CATALOG	TABLE_SCHEMA	TABLE_NAME	CONSTRAINT_TYPE	IS_DEFERRABLE	INITIALLY_DEFERRED
1	Sample	dbo	UQ_Gamer_Email	Sample	dbo	Gamer	UNIQUE	NO	NO

```
-----------------------------------------------------------------------------------------------------------
--T001_03_03
--Drop the unique constraint.
ALTER TABLE Gamer
DROP CONSTRAINT UQ_Gamer_Email;
SELECT  *
FROM    Gamer;
GO -- Run the prvious command and begins new batch
INSERT  Gamer
        ( [Name], [Email], [Age] )
VALUES  ( N'Name7', N'7@7.com', 27 );
--insert Name8 will be fail, because Email must be unique.
INSERT  Gamer
        ( [Name], [Email], [Age] )
VALUES  ( N'Name8', N'7@7.com', 28 );
GO -- Run the prvious command and begins new batch
```

```
SELECT  *
FROM    Gamer;
GO -- Run the prvious command and begins new batch

DELETE Gamer
WHERE Email = N'7@7.com';
/*
Delete the rows which email is '7@7.com'
*/
SELECT  *
FROM    Gamer;
GO -- Run the prvious command and begins new batch
```

	Id	Name	Email	GenderId	CreatedDateTime	Age
1	1	Name1	1@1.com	1	2017-09-01 18:05:03.127	21
2	2	Name5	2@2.com	2	2017-09-01 18:05:18.443	22
3	3	Name3	3@3.com	3	2017-09-01 18:05:41.070	23
4	4	Name4	4@4.com	1	2017-11-02 17:00:26.927	24
5	5	Name5	5@5.com	2	2017-09-01 18:05:03.127	25
6	6	Name6	6@6.com	3	2017-11-02 17:00:26.927	26
7	7	Name7	7@7.com	3	2017-11-02 17:01:17.530	27

	Id	Name	Email	GenderId	CreatedDateTime	Age
1	1	Name1	1@1.com	1	2017-09-01 18:05:03.127	21
2	2	Name5	2@2.com	2	2017-09-01 18:05:18.443	22
3	3	Name3	3@3.com	3	2017-09-01 18:05:41.070	23
4	4	Name4	4@4.com	1	2017-11-02 17:00:26.927	24
5	5	Name5	5@5.com	2	2017-09-01 18:05:03.127	25
6	6	Name6	6@6.com	3	2017-11-02 17:00:26.927	26
7	7	Name7	7@7.com	3	2017-11-02 17:01:17.530	27
8	9	Name7	7@7.com	3	2017-11-02 17:09:33.580	27
9	10	Name8	7@7.com	3	2017-11-02 17:09:33.580	28

	Id	Name	Email	GenderId	CreatedDateTime	Age
1	1	Name1	1@1.com	1	2017-09-01 18:05:03.127	21
2	2	Name5	2@2.com	2	2017-09-01 18:05:18.443	22
3	3	Name3	3@3.com	3	2017-09-01 18:05:41.070	23
4	4	Name4	4@4.com	1	2017-11-02 17:00:26.927	24
5	5	Name5	5@5.com	2	2017-09-01 18:05:03.127	25
6	6	Name6	6@6.com	3	2017-11-02 17:00:26.927	26

```
-----------------------------------------------------------------------------------------------------------
--T001_03_04
--Create or Recreate the unique constraint
IF ( EXISTS ( SELECT    *
              FROM      INFORMATION_SCHEMA.TABLE_CONSTRAINTS
              WHERE     CONSTRAINT_NAME = 'UQ_Gamer_Email' ) )
   BEGIN
```

```
        ALTER TABLE Gamer
        DROP CONSTRAINT UQ_Gamer_Email;
    END;
GO -- Run the previous command and begins new batch
ALTER TABLE Gamer
ADD CONSTRAINT UQ_Gamer_Email UNIQUE(Email);
--Email must be unique
/*
1.
Add Unique constraint Syntax:
--ALTER TABLE {TableName}
--ADD CONSTRAINT {UQConstraintName} UNIQUE({ColumnName});
In {ColumnName} column of {TableName} table,
Add a unique constraint called {UQConstraintName}.
1.1.
E.g.
--ALTER TABLE Gamer
--ADD CONSTRAINT UQ_Gamer_Email UNIQUE(Email);
In Email column of Gamer table,
Add a unique constraint called UQ_Gamer_Email.
2.
Drop Unique constraint Syntax:
--ALTER TABLE {TableName}
--DROP CONSTRAINT {UQConstraintName};
In {TableName} table,
Delete the unique constraint called {UQConstraintName}.
2.1.
E.g.
--ALTER TABLE Gamer
--DROP CONSTRAINT UQ_Gamer_Email;
In Gamer table,
Delete the unique constraint called UQ_Gamer_Email.
*/



--=============================================================================================================
--T001_04
--Foreign Key
/*
What to learn
- Foreign Key Constraint
- No Action/Cascade/Set NULL/SetDefault
*/



-------------------------------------------------------------------------------------------------------------
--T001_04_01
SELECT  *
FROM    Gender;
SELECT  *
FROM    Gender
WHERE   Id = 1;
SELECT  *
FROM    Gamer;
GO -- Run the prvious command and begins new batch
/*
1.
* means all columns
1.1.
--SELECT  *
--FROM    Gender;
Get all columns from Gender Table.
1.2.
```

```
--SELECT  *
--FROM    Gamer;
Get all columns from Gamer Table.
1.3.
--SELECT  *
--FROM    Gender
--WHERE   Id = 1;
Get all columns from Gender Table,
Filter the rows where id must be 1.
*/
```

	Id	Gender
1	1	Male
2	2	Female
3	3	Unknow

	Id	Gender
1	1	Male

	Id	Name	Email	GenderId	CreatedDateTime	Age
1	1	Name1	1@1.com	1	2017-09-01 18:05:03.127	21
2	2	Name5	2@2.com	2	2017-09-01 18:05:18.443	22
3	3	Name3	3@3.com	3	2017-09-01 18:05:41.070	23
4	4	Name4	4@4.com	1	2017-11-02 15:19:43.057	24
5	5	Name5	5@5.com	2	2017-09-01 18:05:03.127	25
6	6	Name6	6@6.com	3	2017-11-02 15:19:43.057	26

```
-------------------------------------------------------------------------------------------------------------
--T001_04_02
--Get the information of the foreign key constraint called "FK_Gender_Gamer"
SELECT  *
FROM    INFORMATION_SCHEMA.REFERENTIAL_CONSTRAINTS
WHERE    CONSTRAINT_NAME = 'FK_Gender_Gamer';
```

	CONSTRAINT_CATALOG	CONSTRAINT_SCHEMA	CONSTRAINT_NAME	UNIQUE_CONSTRAINT_CATALOG	UNIQUE_CONSTRAINT_SCHEMA	UNIQUE_CONSTRAINT_NAME	MATCH_OPTION	UPDAT
1	Sample	dbo	FK_Gender_Gamer	Sample	dbo	PK__Gender__3214EC0771C447F4	SIMPLE	NO AC

```
-------------------------------------------------------------------------------------------------------------
--T001_04_03
--ON DELETE NO ACTION;  this is the default setting of the foreign key constraint.
--Create or Recreate the foreign key constraint
--Delete the the foreign key constraint if it exists
IF ( EXISTS ( SELECT    *
              FROM      INFORMATION_SCHEMA.REFERENTIAL_CONSTRAINTS
              WHERE      CONSTRAINT_NAME = 'FK_Gender_Gamer' ) )
    BEGIN
        ALTER TABLE Gamer
        DROP CONSTRAINT FK_Gender_Gamer;
    END;
GO -- Run the previous command and begins new batch
--Create the foreign key constraint
ALTER TABLE Gamer ADD CONSTRAINT FK_Gender_Gamer
```

```
FOREIGN KEY (GenderId) REFERENCES Gender(Id)
ON DELETE NO ACTION;
GO -- Run the prvious command and begins new batch
-- Delete Rule is No Action
DELETE  FROM Gender
WHERE   Id = 1;
SELECT  *
FROM    Gender;
SELECT  *
FROM    Gamer;
GO -- Run the prvious command and begins new batch
```

```
Msg 547, Level 16, State 0, Line 762
The DELETE statement conflicted with the REFERENCE constraint "FK_Gender_Gamer". The conflict occurred in database "Sample", table "dbo.Gamer", column 'GenderId'.
The statement has been terminated.
```

	Id	Gender
1	1	Male
2	2	Female
3	3	Unknow

	Id	Name	Email	GenderId	CreatedDateTime	Age
1	1	Name1	1@1.com	1	2017-09-01 18:05:03.127	21
2	2	Name5	2@2.com	2	2017-09-01 18:05:18.443	22
3	3	Name3	3@3.com	3	2017-09-01 18:05:41.070	23
4	4	Name4	4@4.com	1	2017-11-02 15:19:43.057	24
5	5	Name5	5@5.com	2	2017-09-01 18:05:03.127	25
6	6	Name6	6@6.com	3	2017-11-02 15:19:43.057	26

```
/*
1.
---- Delete Rule is No Action
--DELETE  FROM Gender
--WHERE   Id = 1;
1.1.
Output as the following
--Msg 547, Level 16, State 0, Line 764
--The DELETE statement conflicted with the REFERENCE constraint "FK_Gender_Gamer".
--The conflict occurred in database "Sample", table "dbo.Gamer", column 'GenderId'.
--The statement has been terminated.
1.2.
We can not do delete the Gender with id is 1,
because the id 1 gender value is currently still used in Gamer Table.
1.3.
You may delete
--ON DELETE NO ACTION;
because, Foreign key is "ON DELETE NO ACTION" by default setting.
This means when you delete valueA in Gender Table,
If the valueA is still used in Gamer table,
then do nothing which means
valueA can not be deleted if it is still used in other table.
*/
```

```
--T001_04_04
--ON DELETE CASCADE;
--Create or Recreate the foreign key constraint
```

```
--Delete the the foreign key constraint if it exists
IF ( EXISTS ( SELECT    *
              FROM      INFORMATION_SCHEMA.REFERENTIAL_CONSTRAINTS
              WHERE     CONSTRAINT_NAME = 'FK_Gender_Gamer' ) )
    BEGIN
        ALTER TABLE Gamer
        DROP CONSTRAINT FK_Gender_Gamer;
    END;
GO -- Run the previous command and begins new batch
--Create the foreign key constraint
ALTER TABLE Gamer ADD CONSTRAINT FK_Gender_Gamer
FOREIGN KEY (GenderId) REFERENCES Gender(Id)
ON DELETE CASCADE;
GO -- Run the prvious command and begins new batch
SELECT  *
FROM    Gender;
SELECT  *
FROM    Gamer;
GO -- Run the prvious command and begins new batch
-- Delete Rule is Cascade
DELETE  FROM Gender
WHERE   Id = 1;
SELECT  *
FROM    Gender;
SELECT  *
FROM    Gamer;
GO -- Run the prvious command and begins new batch
/*
1.
---- Delete Rule is No Action
--DELETE  FROM Gender
--WHERE   Id = 1;
When we delete the the Gender id 1 row from the Gender Table.
It also delete the Gender id 1 row from Gamer Table.
1.1.
Because of the delete rule is Cascade
When
--DELETE  FROM Gender
--WHERE   Id = 1;
It also do the following
--DELETE  FROM Gamer
--WHERE   GenderId = 1;
*/
```

	Id	Gender
1	1	Male
2	2	Female
3	3	Unknow

	Id	Name	Email	GenderId	CreatedDateTime	Age
1	1	Name1	1@1.com	1	2017-09-01 18:05:03.127	21
2	2	Name5	2@2.com	2	2017-09-01 18:05:18.443	22
3	3	Name3	3@3.com	3	2017-09-01 18:05:41.070	23
4	4	Name4	4@4.com	1	2017-11-02 15:19:43.057	24
5	5	Name5	5@5.com	2	2017-09-01 18:05:03.127	25
6	6	Name6	6@6.com	3	2017-11-02 15:19:43.057	26

	Id	Gender
1	2	Female
2	3	Unknow

	Id	Name	Email	GenderId	CreatedDateTime	Age
1	2	Name5	2@2.com	2	2017-09-01 18:05:18.443	22
2	3	Name3	3@3.com	3	2017-09-01 18:05:41.070	23
3	5	Name5	5@5.com	2	2017-09-01 18:05:03.127	25
4	6	Name6	6@6.com	3	2017-11-02 15:19:43.057	26

```
-----------------------------------------------------------------------------------------------------------
--T001_04_05
--ON DELETE SET DEFAULT;
--Create or Recreate the foreign key constraint
--Delete the the foreign key constraint if it exists
IF ( EXISTS ( SELECT    *
              FROM      INFORMATION_SCHEMA.REFERENTIAL_CONSTRAINTS
              WHERE     CONSTRAINT_NAME = 'FK_Gender_Gamer' ) )
    BEGIN
        ALTER TABLE Gamer
        DROP CONSTRAINT FK_Gender_Gamer;
    END;
GO -- Run the previous command and begins new batch
--Create the foreign key constraint
ALTER TABLE Gamer ADD CONSTRAINT FK_Gender_Gamer
FOREIGN KEY (GenderId) REFERENCES Gender(Id)
ON DELETE SET DEFAULT;
GO -- Run the prvious command and begins new batch
SELECT  *
FROM    Gender;
SELECT  *
FROM    Gamer;
GO -- Run the prvious command and begins new batch
-- Delete Rule is DEFAULT
DELETE  FROM Gender
WHERE   Id = 2;
SELECT  *
FROM    Gender;
```

```
SELECT  *
FROM    Gamer;
GO -- Run the prvious command and begins new batch
/*
1.
---- Delete Rule is DEFAULT
--DELETE  FROM Gender
--WHERE   Id = 2;
Previously, We set the default constraint of GenderId column in Gamer table.
The default value of [Gamer].[GenderId] is 3.
When we delete the Id 2 row from Gender Table.
It will set the GenderId in Gamer table to default value which is 3.
1.1.
That means when we do the following.
--DELETE  FROM Gender
--WHERE   Id = 2;
It will also do the following.
--UPDATE Gamer
--SET GenderId = 3
--WHERE GenderId = 2;
*/
```

	Id	Gender
1	2	Female
2	3	Unknow

	Id	Name	Email	GenderId	CreatedDateTime	Age
1	2	Name5	2@2.com	2	2017-09-01 18:05:18.443	22
2	3	Name3	3@3.com	3	2017-09-01 18:05:41.070	23
3	5	Name5	5@5.com	2	2017-09-01 18:05:03.127	25
4	6	Name6	6@6.com	3	2017-11-02 15:19:43.057	26

	Id	Gender
1	3	Unknow

	Id	Name	Email	GenderId	CreatedDateTime	Age
1	2	Name5	2@2.com	3	2017-09-01 18:05:18.443	22
2	3	Name3	3@3.com	3	2017-09-01 18:05:41.070	23
3	5	Name5	5@5.com	3	2017-09-01 18:05:03.127	25
4	6	Name6	6@6.com	3	2017-11-02 15:19:43.057	26

```
--------------------------------------------------------------------------------------------------------------
--T001_04_06
--ON DELETE SET NULL;
--Create or Recreate the foreign key constraint
--Delete the the foreign key constraint if it exists
IF ( EXISTS ( SELECT    *
              FROM      INFORMATION_SCHEMA.REFERENTIAL_CONSTRAINTS
              WHERE     CONSTRAINT_NAME = 'FK_Gender_Gamer' ) )
    BEGIN
        ALTER TABLE Gamer
        DROP CONSTRAINT FK_Gender_Gamer;
    END;
GO -- Run the previous command and begins new batch
--Create the foreign key constraint
ALTER TABLE Gamer ADD CONSTRAINT FK_Gender_Gamer
```

```
FOREIGN KEY (GenderId) REFERENCES Gender(Id)
ON DELETE SET NULL;
GO -- Run the prvious command and begins new batch
SELECT  *
FROM    Gender;
SELECT  *
FROM    Gamer;
GO -- Run the prvious command and begins new batch
-- Delete Rule is SET NULL
DELETE  FROM Gender
WHERE   Id = 3;
SELECT  *
FROM    Gender;
SELECT  *
FROM    Gamer;
GO -- Run the prvious command and begins new batch
/*
1.
---- Delete Rule is SET NULL
--DELETE  FROM Gender
--WHERE   Id = 3;
When we delete the Id 3 row from Gender Table.
It will set the GenderId in Gamer table to NULL value.
1.1.
That means when we do the following.
--DELETE  FROM Gender
--WHERE   Id = 3;
It will also do the following.
--UPDATE Gamer
--SET GenderId = NULL
--WHERE GenderId = 3;
*/
```

	Id	Gender
1	3	Unknow

	Id	Name	Email	GenderId	CreatedDateTime	Age
1	2	Name5	2@2.com	3	2017-09-01 18:05:18.443	22
2	3	Name3	3@3.com	3	2017-09-01 18:05:41.070	23
3	5	Name5	5@5.com	3	2017-09-01 18:05:03.127	25
4	6	Name6	6@6.com	3	2017-11-02 15:19:43.057	26

	Id	Gender

	Id	Name	Email	GenderId	CreatedDateTime	Age
1	2	Name5	2@2.com	NULL	2017-09-01 18:05:18.443	22
2	3	Name3	3@3.com	NULL	2017-09-01 18:05:41.070	23
3	5	Name5	5@5.com	NULL	2017-09-01 18:05:03.127	25
4	6	Name6	6@6.com	NULL	2017-11-02 15:19:43.057	26

```
--=============================================================================================================
--T001_05
--Clean up
--Drop Table if it exists
IF ( EXISTS ( SELECT    *
              FROM      INFORMATION_SCHEMA.TABLES
              WHERE     TABLE_NAME = 'Gamer' ) )
   BEGIN
       TRUNCATE TABLE Gamer;
```

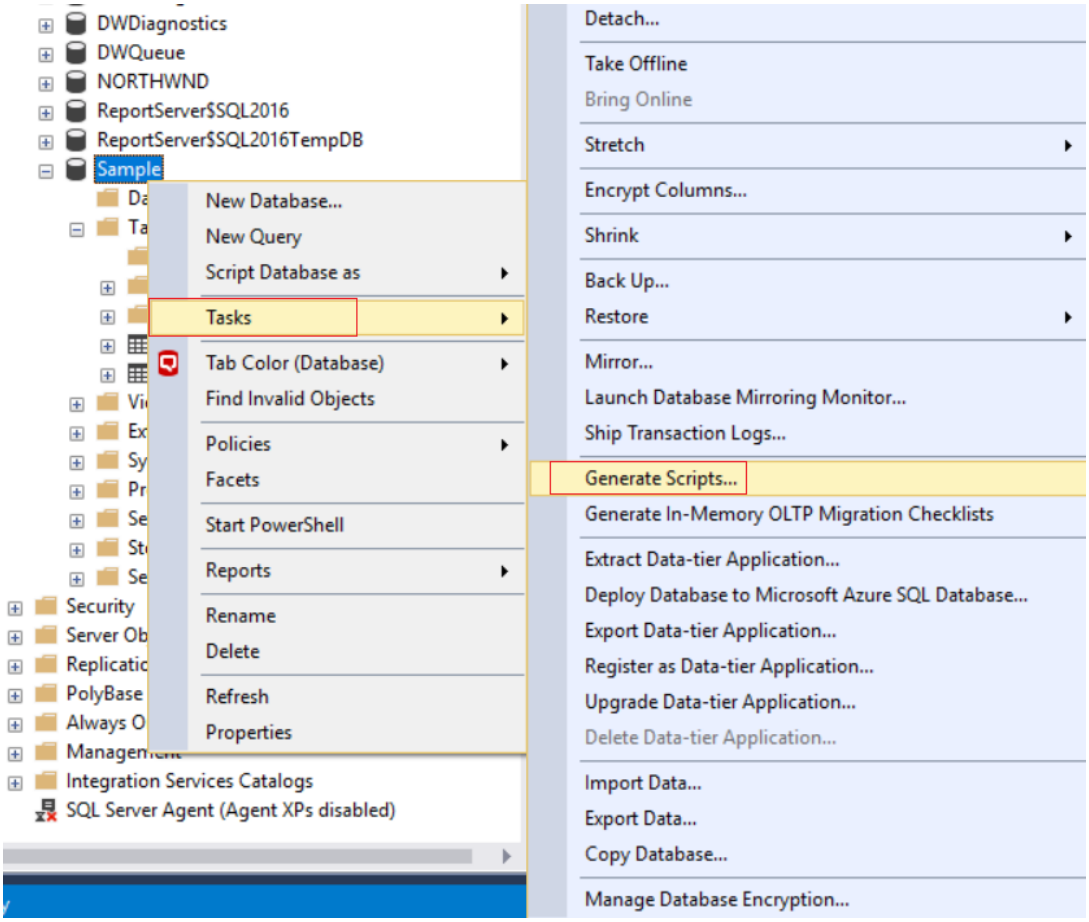
```
        DROP TABLE Gamer;
    END;
GO -- Run the previous command and begins new batch
--Drop Table if it exists
IF ( EXISTS ( SELECT    *
              FROM      INFORMATION_SCHEMA.TABLES
              WHERE      TABLE_NAME = 'Gender' ) )
    BEGIN
        TRUNCATE TABLE Gender;
        DROP TABLE Gender;
    END;
GO -- Run the previous command and begins new batch
/*
1.
-- TRUNCATE TABLE dbo.tblPerson2;
and
--DELETE  dbo.tblPerson2
are both doing the same thing to delete every data in the table.
However, TRUNCATE TABLE is better
because TRUNCATE TABLE will delete the data and clean up the space.
DELETE will delete the data without clean up the space.
It is more possible to cause data fragmentation.
2.
--DROP TABLE Gender;
Delete the table.
*/
```

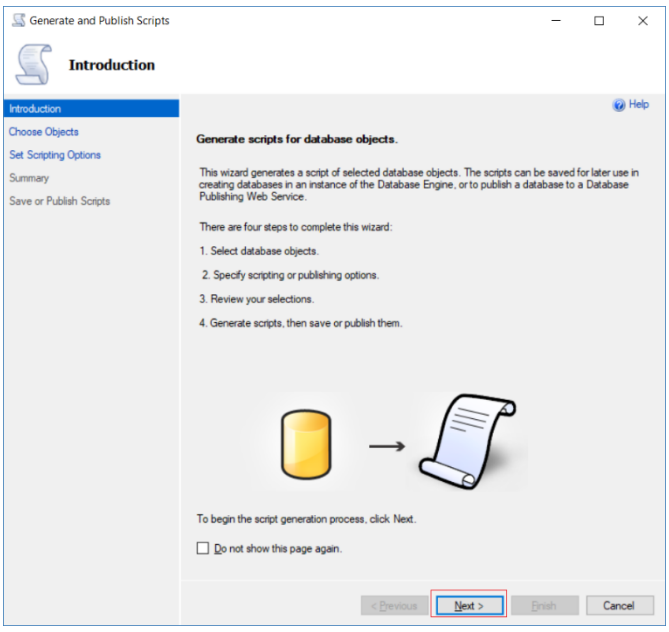
5. Generate Script to Back up Database

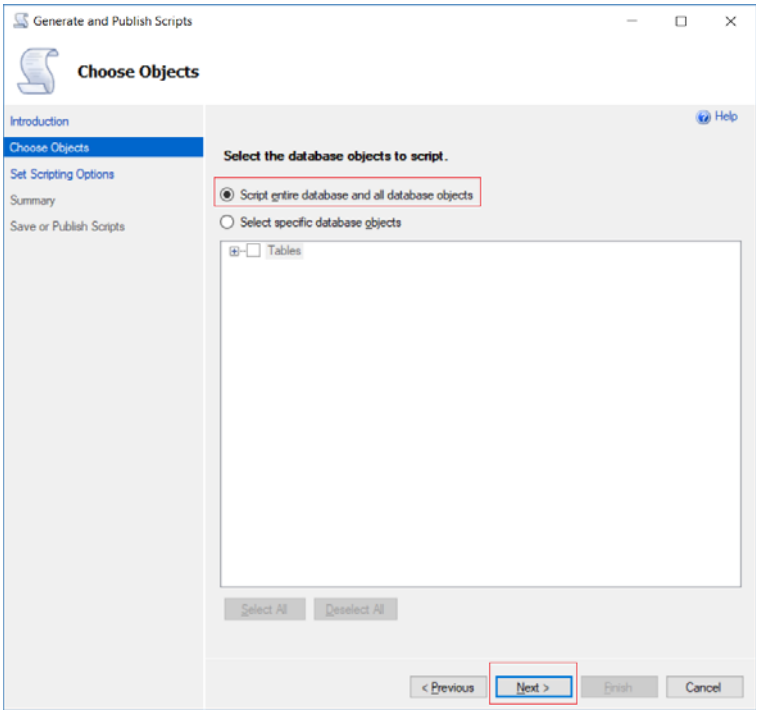
Database Name --> Right Click --> Tasks --> Generate Scripts -->
Next -->
Select
Script entire database and all database objects
--> Next
--> Advanced
--> **Types of data to script : Schema and data**
--> Next
--> Next
--> Finish

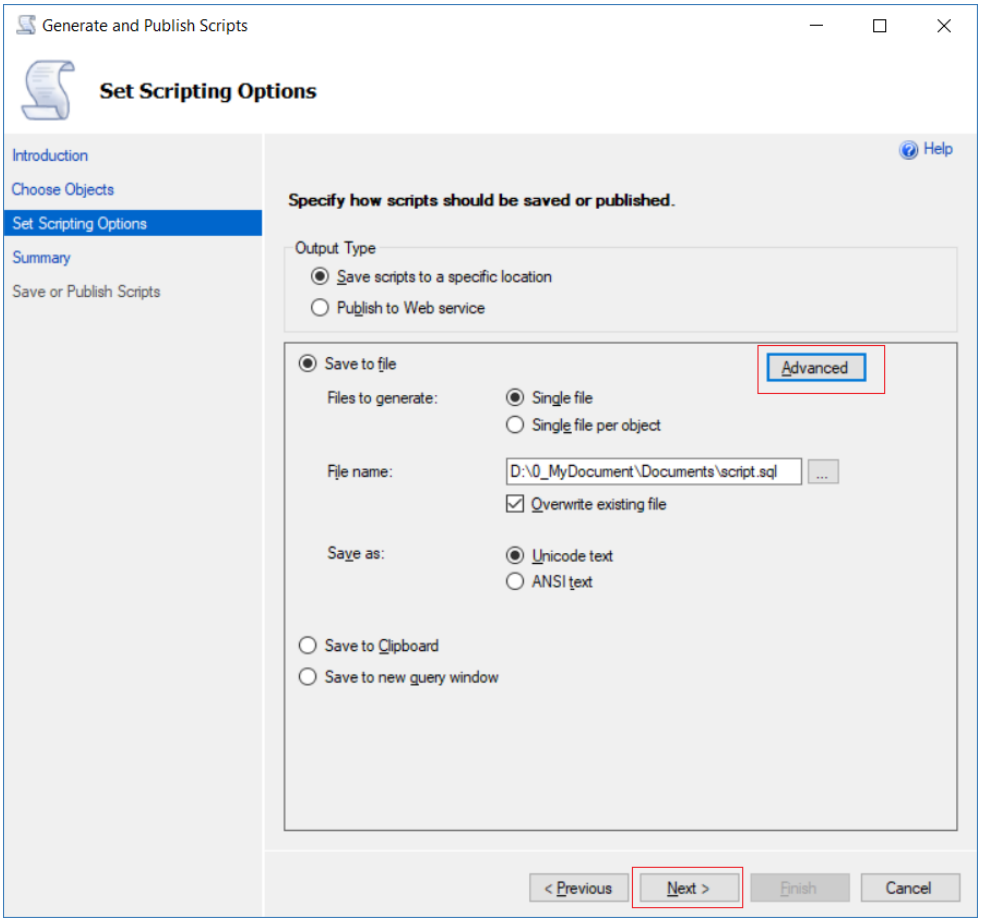
It will create **script.sql** file into **Document Folder**

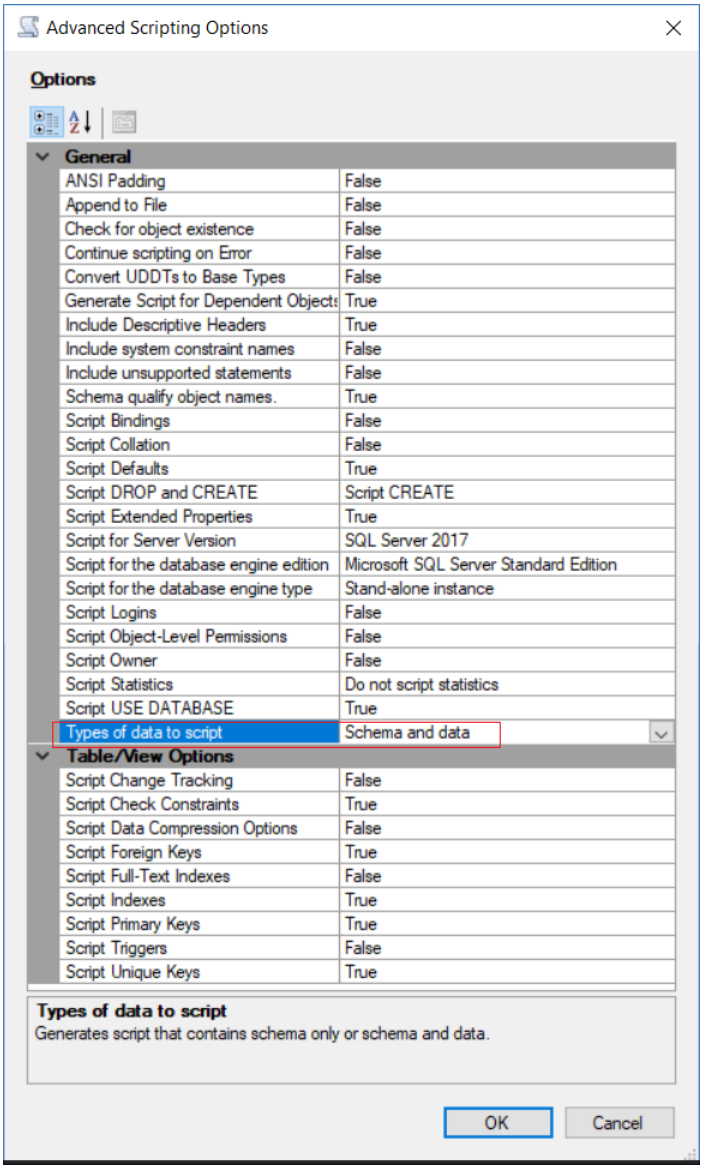
-->
I normally delete create database part.
Please have a look the file,

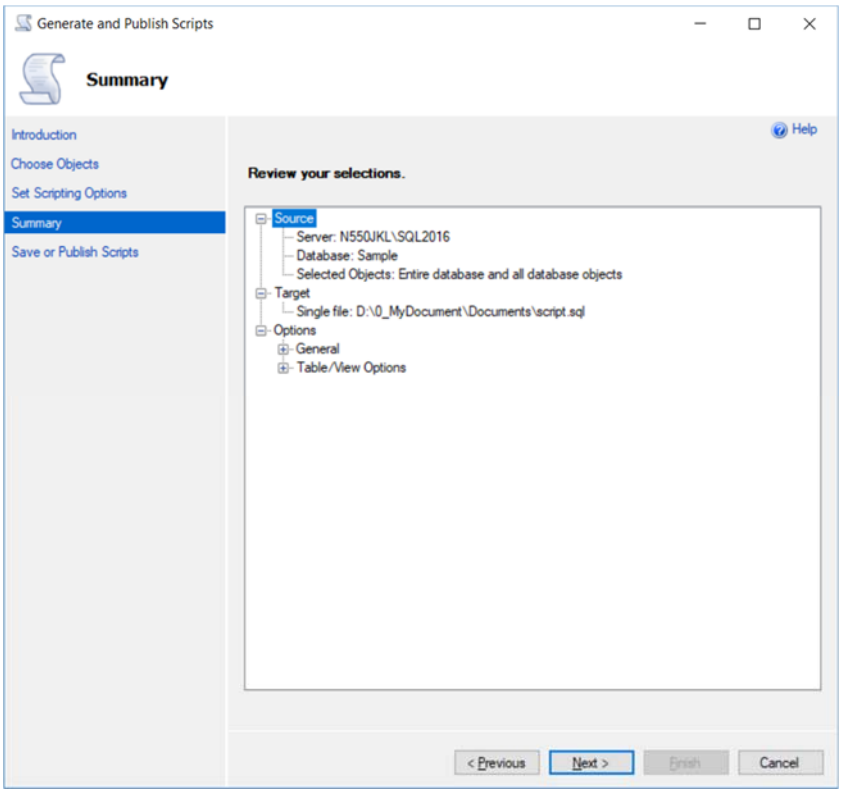

DWDiagnostics
DWQueue
NORTHWND
ReportServer$SQL2016
ReportServer$SQL2016TempDB
Sample
New Database...
New Query
Script Database as
Tasks
Tab Color (Database)
Find Invalid Objects
Policies
Facets
Start PowerShell
Reports
Rename
Delete
Refresh
Properties
Security
Integration Services Catalogs
SQL Server Agent (Agent XPs disabled)
Detach...
Take Offline
Bring Online
Stretch
Encrypt Columns...
Shrink
Back Up...
Restore
Mirror...
Launch Database Mirroring Monitor...
Ship Transaction Logs...
Generate Scripts...
Generate In-Memory OLTP Migration Checklists
Extract Data-tier Application...
Deploy Database to Microsoft Azure SQL Database...
Export Data-tier Application...
Register as Data-tier Application...
Upgrade Data-tier Application...
Delete Data-tier Application...
Import Data...
Export Data...
Copy Database...
Manage Database Encryption...


Generate and Publish Scripts
Introduction
Introduction
Choose Objects
Set Scripting Options
Summary
Save or Publish Scripts
Help
Generate scripts for database objects.
This wizard generates a script of selected database objects. The scripts can be saved for later use in creating databases in an instance of the Database Engine, or to publish a database to a Database Publishing Web Service.
There are four steps to complete this wizard:
1. Select database objects.
2. Specify scripting or publishing options.
3. Review your selections.
4. Generate scripts, then save or publish them.
To begin the script generation process, click Next.
Do not show this page again.
< Previous
Next >
Finish
Cancel


Generate and Publish Scripts
Choose Objects
Introduction
Choose Objects
Set Scripting Options
Summary
Save or Publish Scripts
Help
Select the database objects to script.
Script entire database and all database objects
Select specific database objects
Tables
Select All
Deselect All
< Previous
Next >
Finish
Cancel


Generate and Publish Scripts
Set Scripting Options
Introduction
Choose Objects
Set Scripting Options
Summary
Save or Publish Scripts
Help
Specify how scripts should be saved or published.
Output Type
Save scripts to a specific location
Publish to Web service
Save to file
Advanced
Files to generate:
Single file
Single file per object
File name:
D:\0_MyDocument\Documents\script.sql
Overwrite existing file
Save as:
Unicode text
ANSI text
Save to Clipboard
Save to new query window
< Previous
Next >
Finish
Cancel


Advanced Scripting Options
Options
General
ANSI Padding False
Append to File False
Check for object existence False
Continue scripting on Error False
Convert UDDTs to Base Types False
Generate Script for Dependent Objects True
Include Descriptive Headers True
Include system constraint names False
Include unsupported statements False
Schema qualify object names. True
Script Bindings False
Script Collation False
Script Defaults True
Script DROP and CREATE Script CREATE
Script Extended Properties True
Script for Server Version SQL Server 2017
Script for the database engine edition Microsoft SQL Server Standard Edition
Script for the database engine type Stand-alone instance
Script Logins False
Script Object-Level Permissions False
Script Owner False
Script Statistics Do not script statistics
Script USE DATABASE True
Types of data to script Schema and data
Table/View Options
Script Change Tracking False
Script Check Constraints True
Script Data Compression Options False
Script Foreign Keys True
Script Full-Text Indexes False
Script Indexes True
Script Primary Keys True
Script Triggers False
Script Unique Keys True
Types of data to script
Generates script that contains schema only or schema and data.
OK
Cancel


Generate and Publish Scripts
Summary
Introduction
Choose Objects
Set Scripting Options
Summary
Save or Publish Scripts
Help
Review your selections.
Source
Server: N550JKL\SQL2016
Database: Sample
Selected Objects: Entire database and all database objects
Target
Single file: D:\0_MyDocument\Documents\script.sql
Options
General
Table/View Options
< Previous
Next >
Finish
Cancel