(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

1. SSMS

-----------

2. Create and Drop Database

2.1. Using SSMS to Create and Drop Database

2.1.1. Using SSMS to Create Database

2.1.2. Using SSMS to DropDatabase

2.2. Using Query to Create and Drop Database

-----------

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

3.1. CreateTables - Gamer, Gender

3.1.1. CreateTable - Gender

3.1.2. Insert Data to Gender

3.1.3. CreateTable - Gamer and Set Default Constraint

3.1.4. Gamer - Set Check Constraint

3.1.5. Gamer - Referential Integrity constraint (Foreign Key)

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

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

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

3.1.6. Insert Data to Gamer

-----------

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

5. Generate Script to Back up Database  
=======================================================================

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

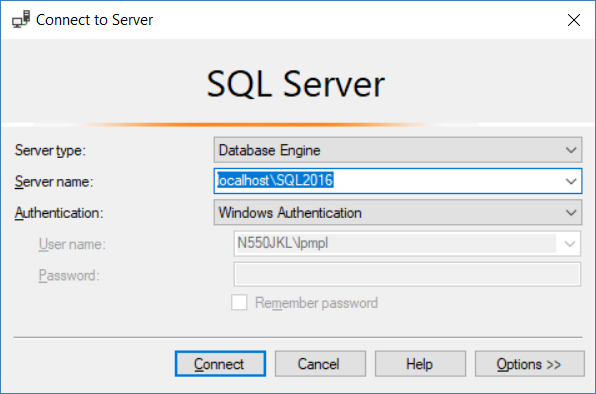
- Insert

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

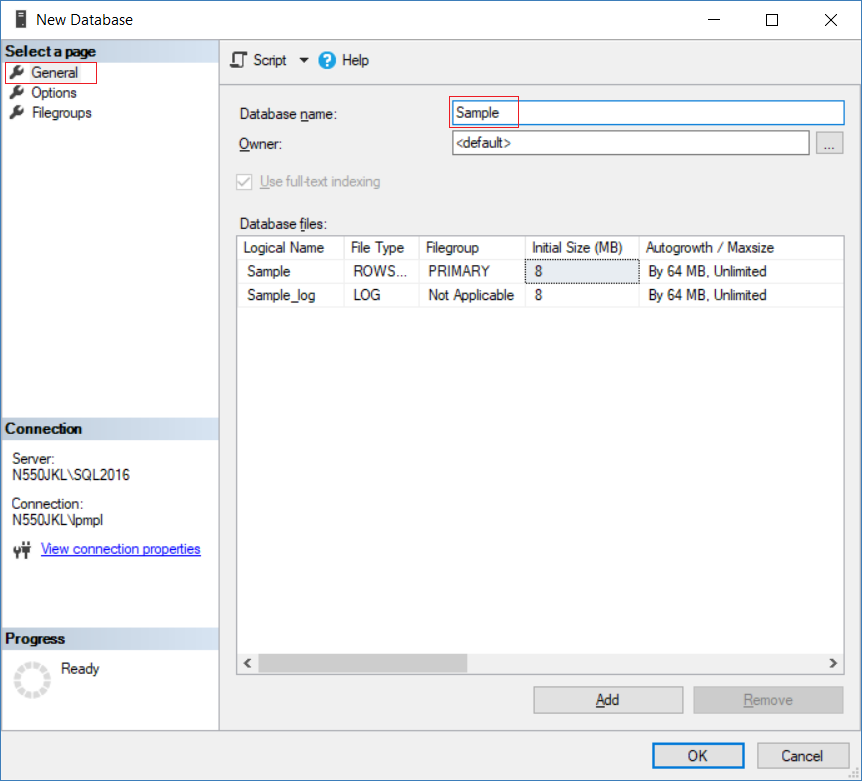
1. Right Click on Databases folder in the Object explorer

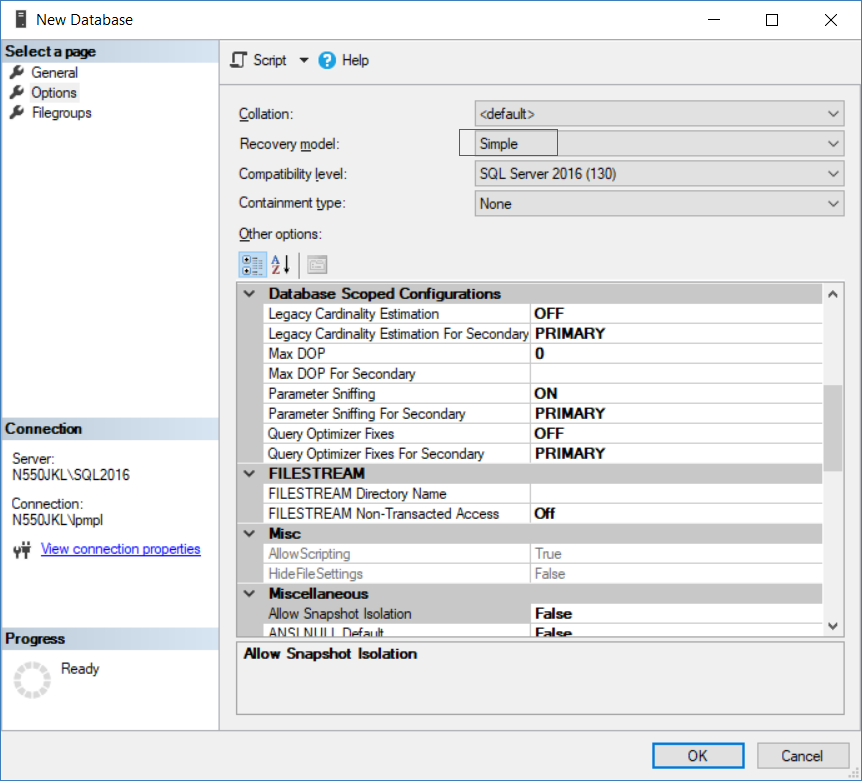
2. Select New Database

3. In the New Database dialog box, enter the Database name and click OK.

Database Name : **Sample**

Recovery Model : **Simple**

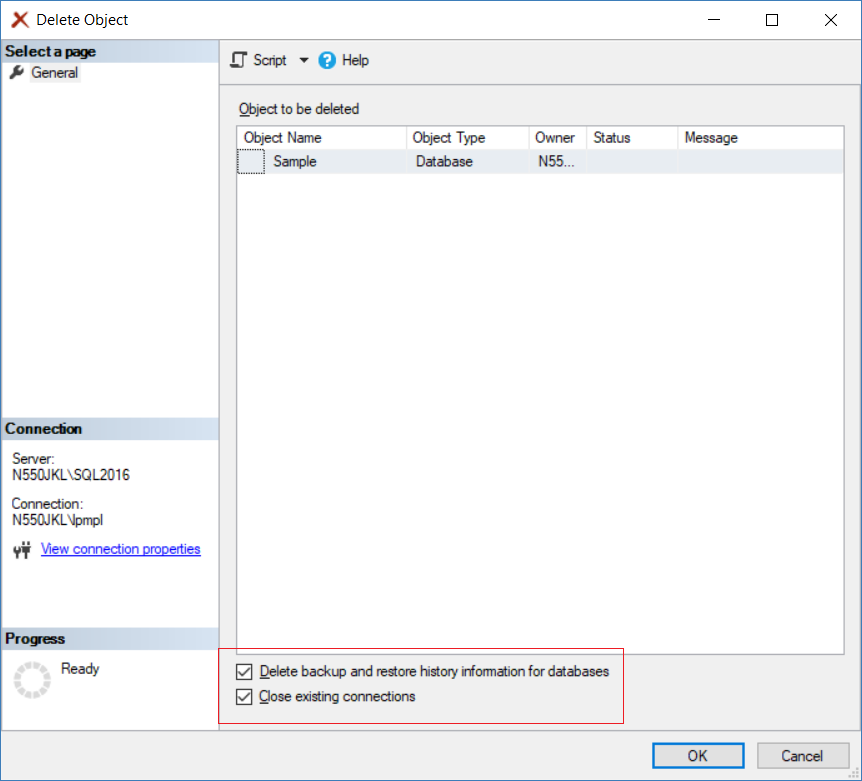




2.1.2. Using SSMS to DropDatabase

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

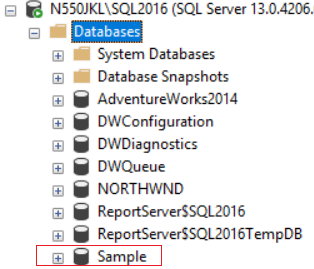
/\*

-- 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

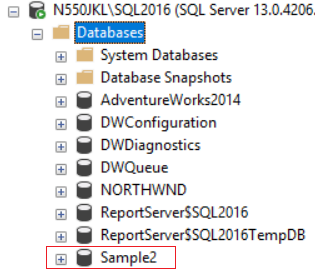
/\*

-- 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';

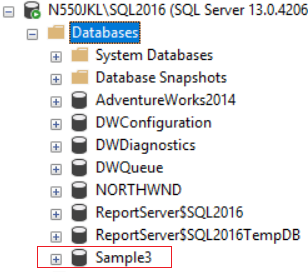
Reference:

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

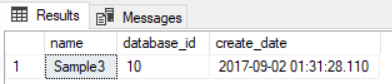
WHERE name = N'Sample3';

GO

/\*

sys.databases is the system database which store all the database list information

\*/



------------------------------------------------------------------------------------------------------------

--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)

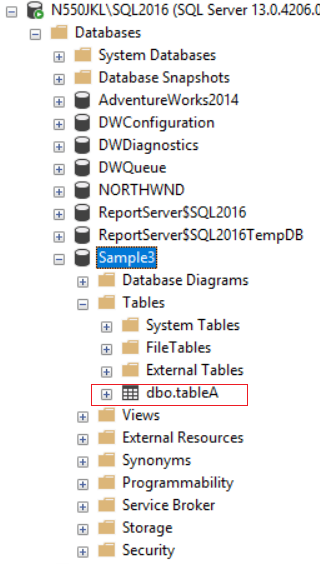
Sometimes a SQL Server database will include one or more .NDF (secondary data files).

-- 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

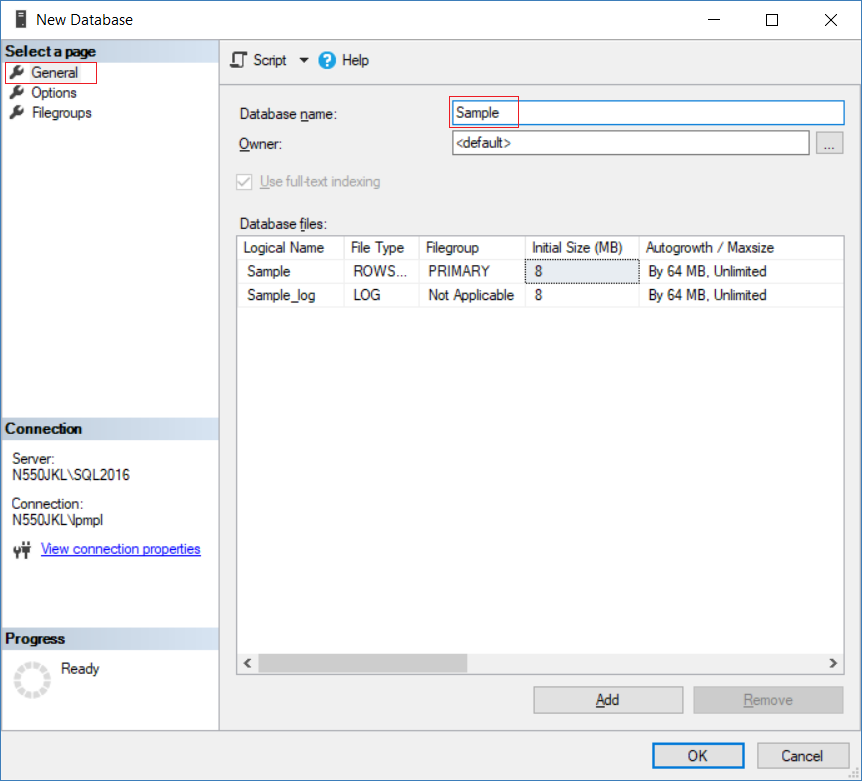
- Foreign Key

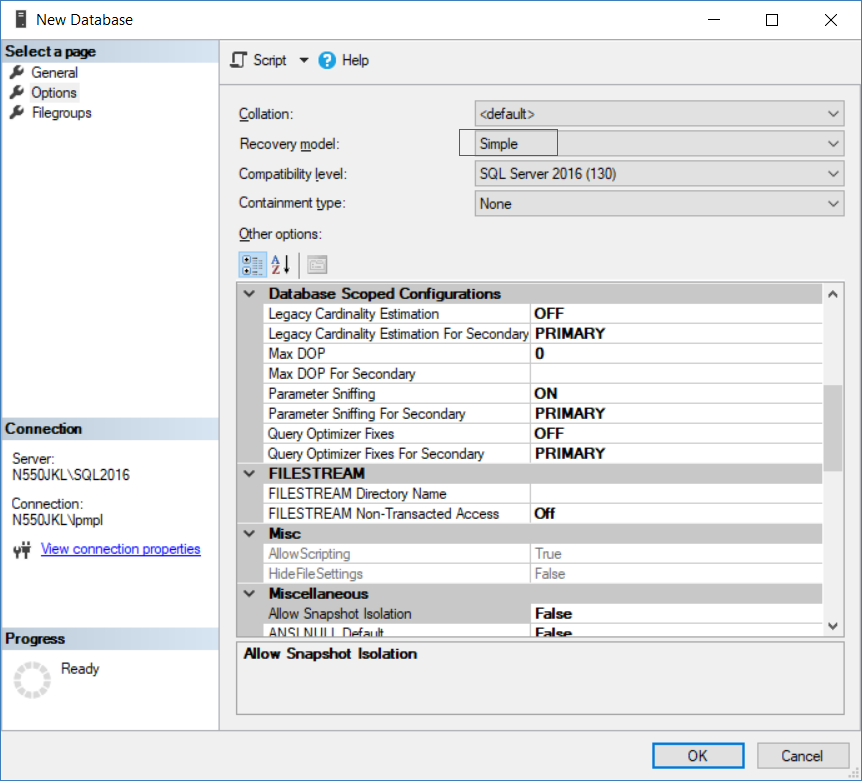
- Insert

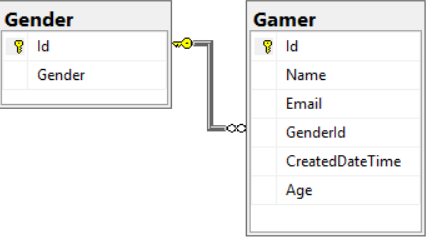
Database --> New Database

Database Name : **Sample**

Recovery Model : **Simple**







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**

-->

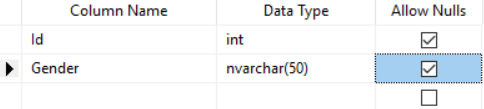
Select Id column --> Right Click

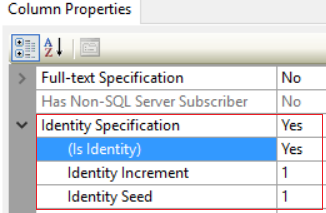
--> Set Primary Key

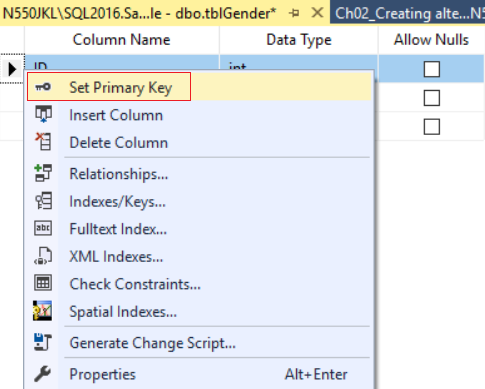
-->

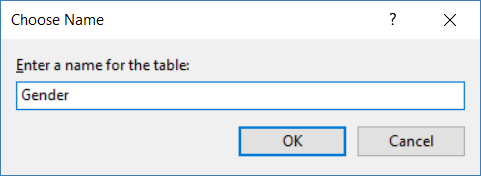
Ctrl + S -->

Table Name:   Gender









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**.

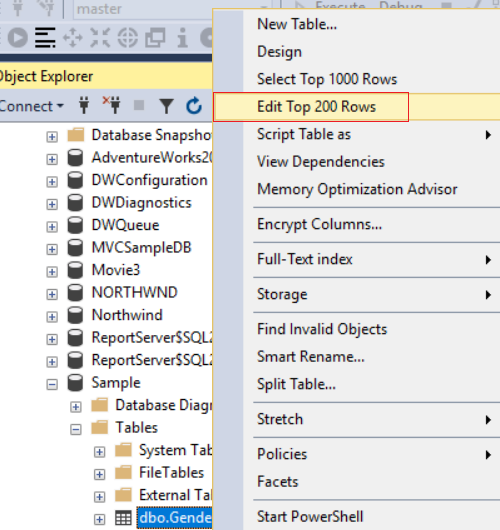
Additionally, I set Id is **Identity Column**.

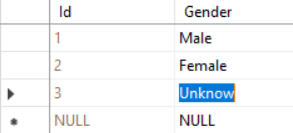
IDENTITY(1,1) means Id will start from 1 (the **first** one is **identity seed**),

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.





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))**

-->

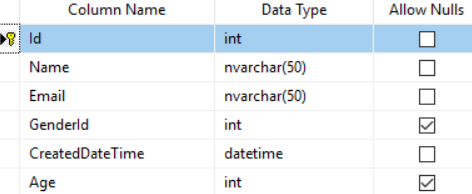
Select Id column --> Right Click

--> Set Primary Key

-->

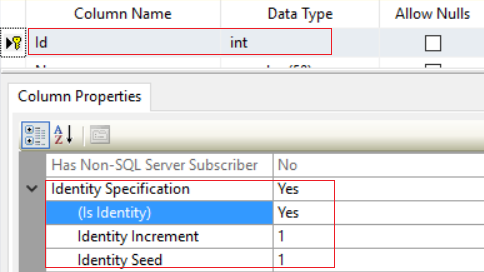
Ctrl + S -->

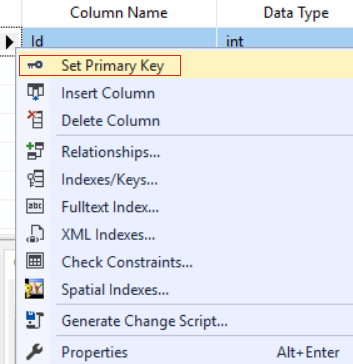
Table Name:   Gamer



-------------------------------------------------

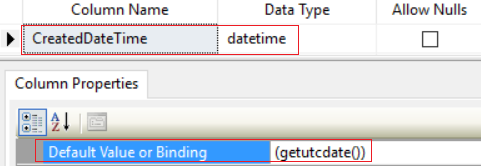
Id column





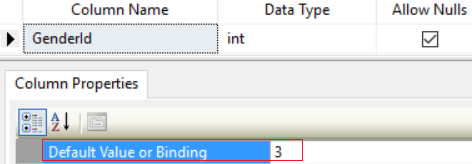
-------------------------------------------------

CreatedDateTime Column

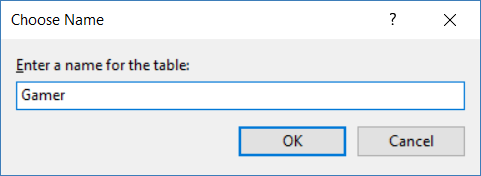


-------------------------------------------------

GenderId Column



-------------------------------------------------



3.1.4. Gamer - Set Check Constraint

Databsae Name --> Tables --> table name --> Constraints --> New Constraint...

--> Add -->

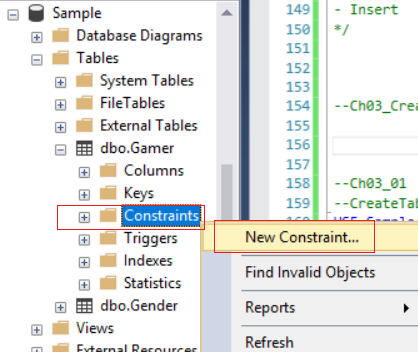
(Name): **CK\_Gamer\_Age**

Expression:    press **..**.

-->    **([Age]>(0) AND [Age]<(150))**--> OK

-->   Close

--> Save the Table Design

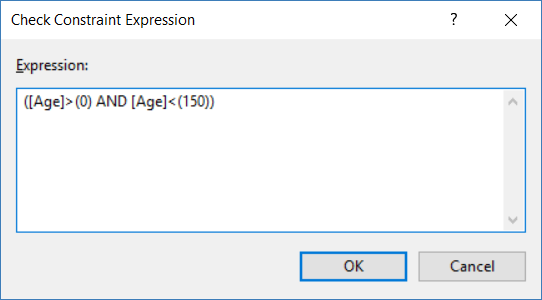


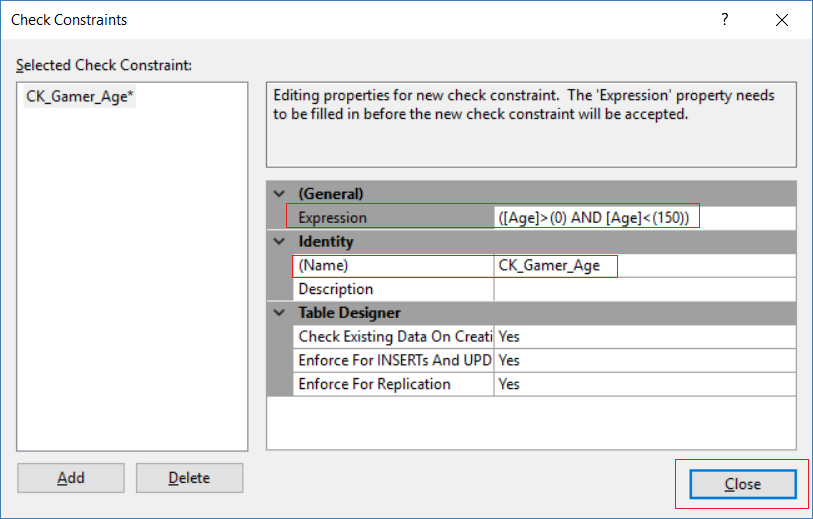
Graphical user interface, application

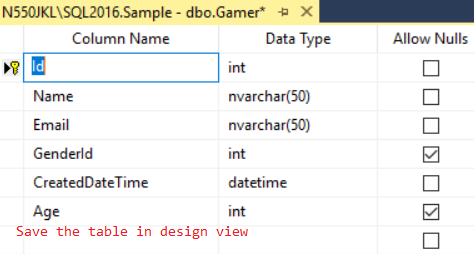
Description automatically generated

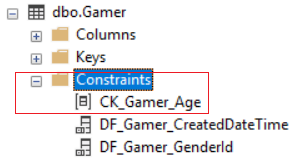
Graphical user interface

Description automatically generated









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

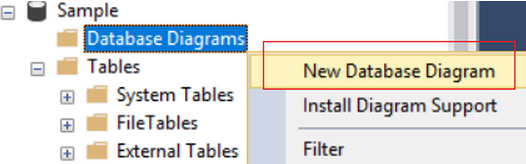
--> Name:  **Diagram\_0**

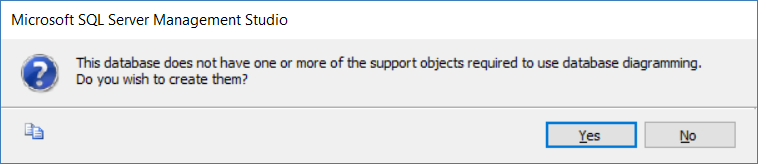
If you want to modify again.

Database Name --> Tables --> Table Name --> Keys --> FK --> Right Click --> Modify -->

**Delete Rule / Update Rule** -->  **No Action / Cascade / Set Null / Set Default**

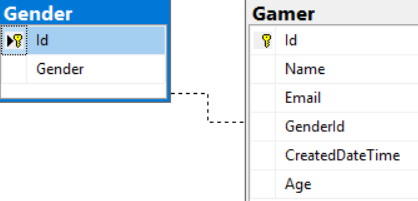
By default, I set to **No Action**.

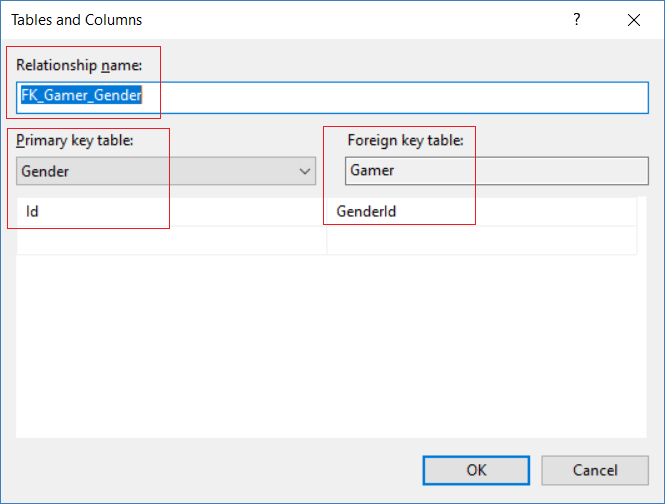


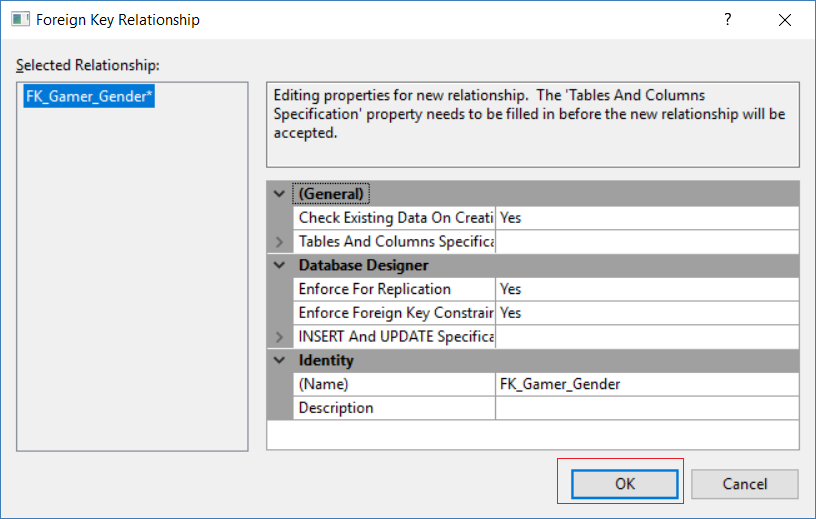


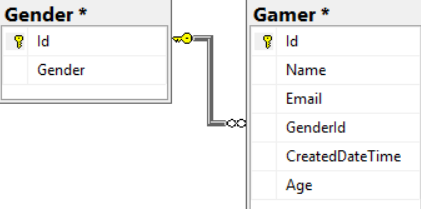
Graphical user interface, application, Word

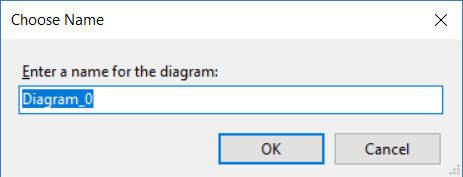
Description automatically generated

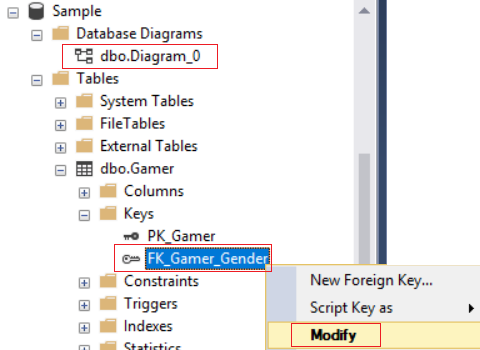


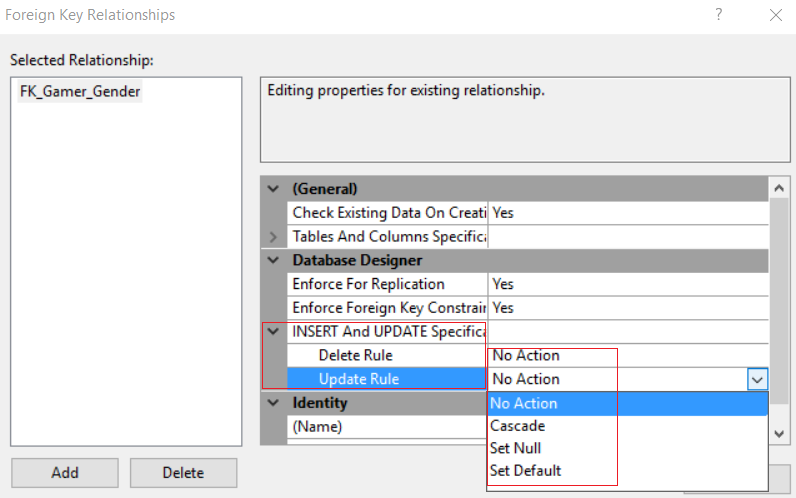












3.1.5.2. Create Referential Integrity constraint (Foreign Key) in Table

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

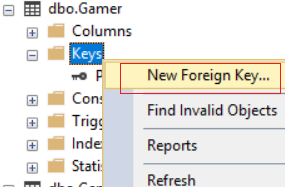
--> In Design View, Ctrl+S    Save the Design

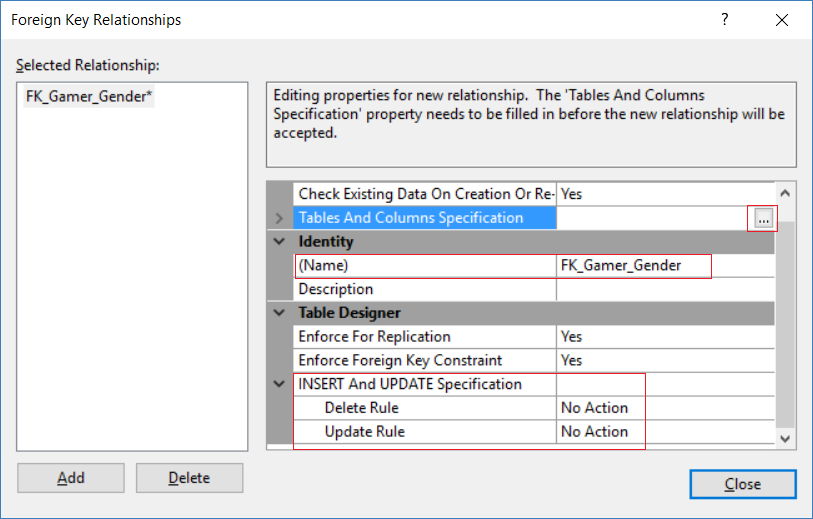
If you want to modify again.

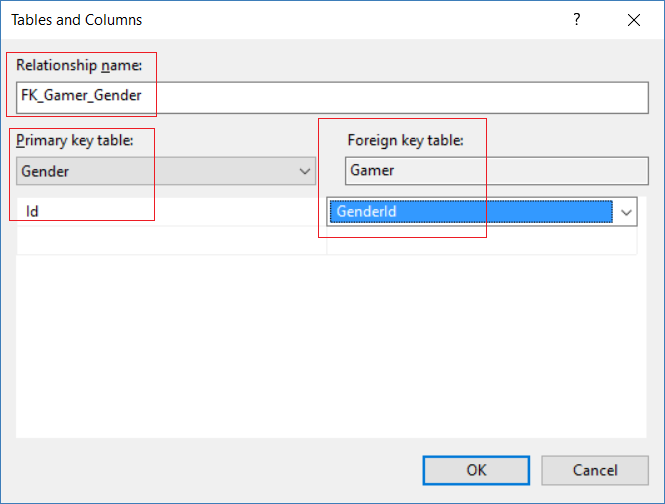
Database Name --> Tables --> Table Name --> Keys --> FK --> Right Click --> Modify -->

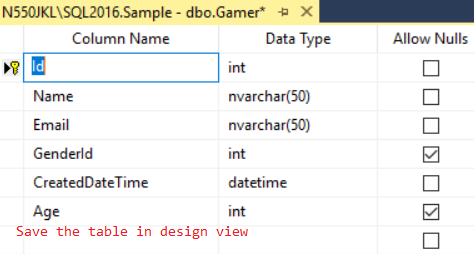
**Delete Rule / Update Rule** -->  **No Action / Cascade / Set Null / Set Default**

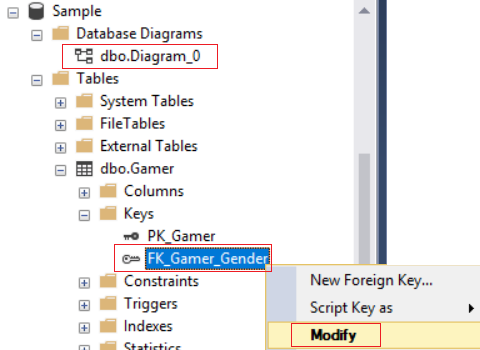
By default, I set to **No Action**.

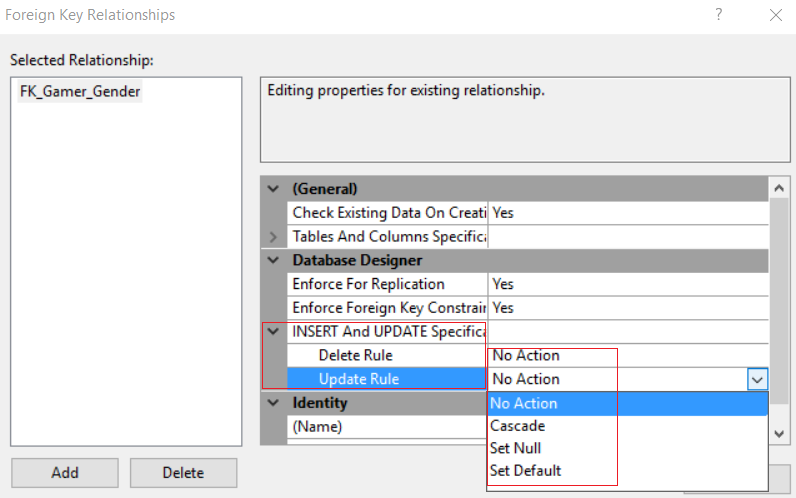












3.1.5.3. Create Referential Integrity constraint (Foreign Key) in Design View

Database Name --> Tables --> Table Name --> Right Click --> Design -->

In the Design View, Select the Column Name --> Right Click --> Relationship

--> Add

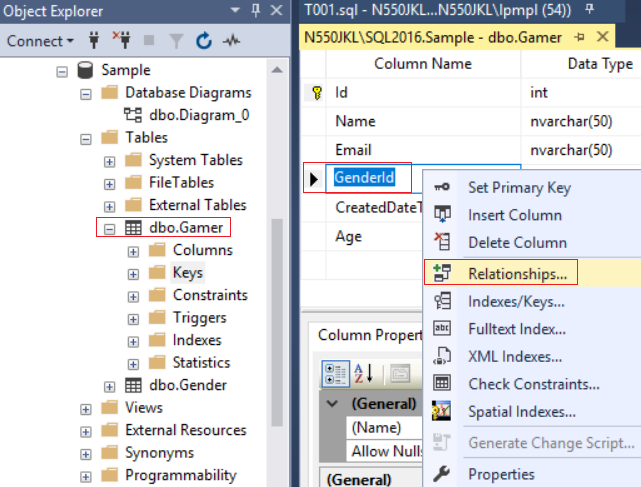
--> **Tables and Column Specification**

It will create **Foreign Key Relationship "FK\_Gamer\_Gender**"

Make sure I connect **[**Gender**].[**Id**]** into **[**Gamer**].[**GenderId**]**

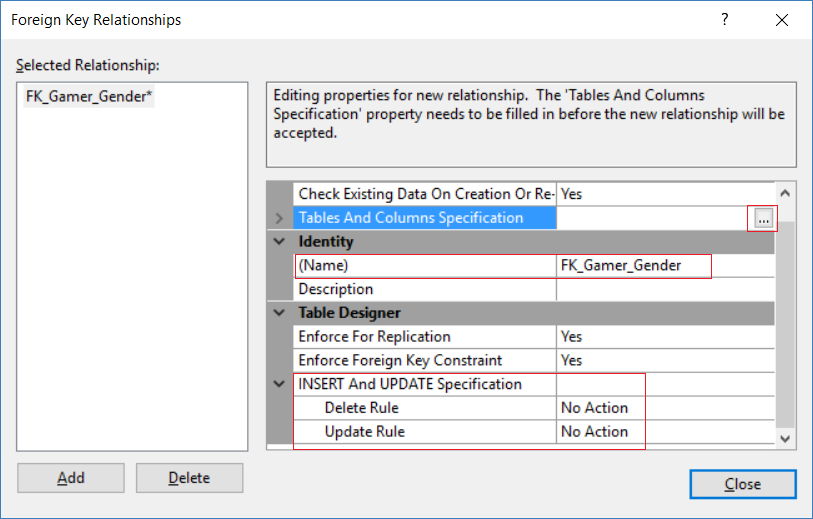
--> Close

--> In Design View, Ctrl + S, Save the Table.



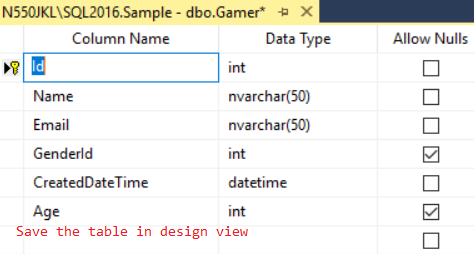
Graphical user interface, application

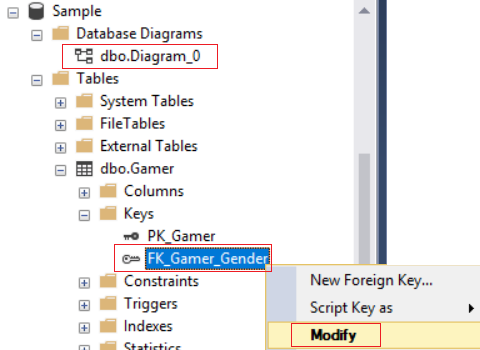
Description automatically generated



Graphical user interface, application, Word

Description automatically generated





3.1.6. Insert Data to Gamer

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

-->

**NULL      Name1**[**1@1.com**](mailto:1@1.com)**1    NULL    21**

**NULL      Name2**[**2@2.com**](mailto:1@1.com)**2    NULL    22**

**NULL      Name3**[**3@3.com**](mailto:1@1.com)**NULL    NULL    23**

**NULL      Name4**[**4@4.com**](mailto:1@1.com)**1    NULL    24**

**NULL      Name5**[**5@5.com**](mailto:1@1.com)**2    NULL    25**

**NULL      Name6**[**6@6.com**](mailto:1@1.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.

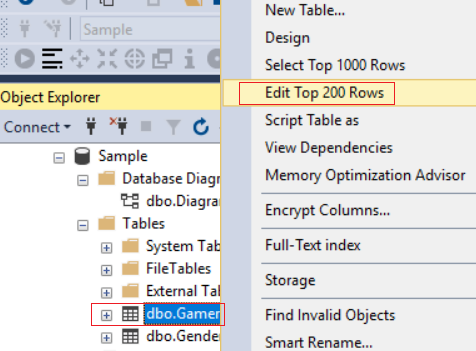
**Default Value** of GenderId is **3**

If I didn't provide any value, it will auto set to 3

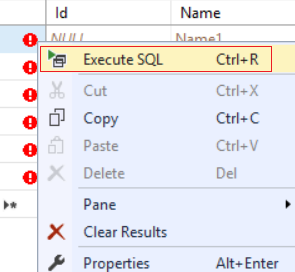
3.

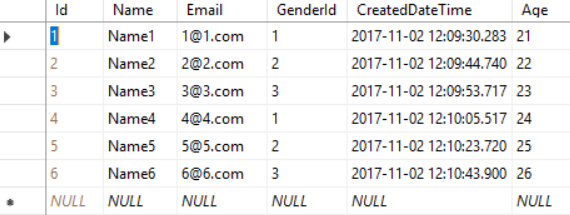
**Default Value** of **CreateDateTime**is **(getutcdate())**

If I didn't provide any value, it will auto set to current UTC Time.









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).

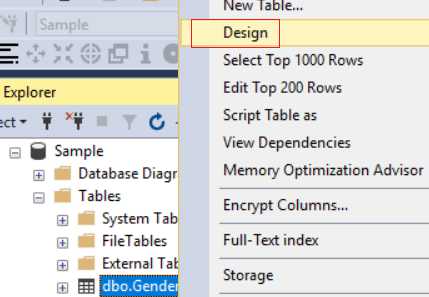
-- ON [PRIMARY]

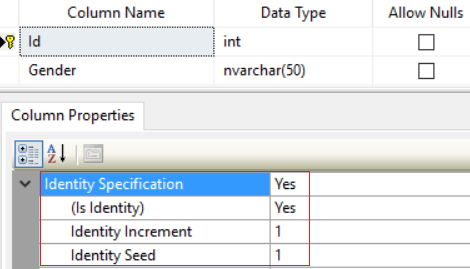
means create this table on the .MDF(primary data file).

\*/

Graphical user interface, application

Description automatically generated





-----------------------------------------------------------------------------------------------------

--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.

\*/

Text, letter

Description automatically generated

Table

Description automatically generated

-----------------------------------------------------------------------------------------------------

--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)

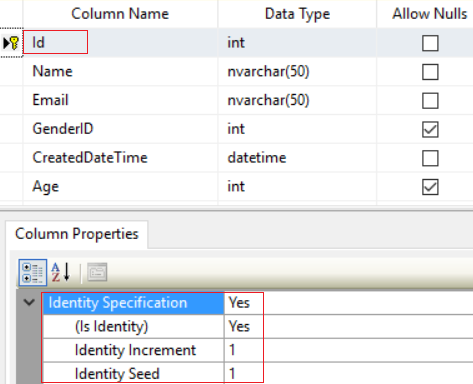
Sometimes a SQL Server database will include one or more .NDF (secondary data files).

-- 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

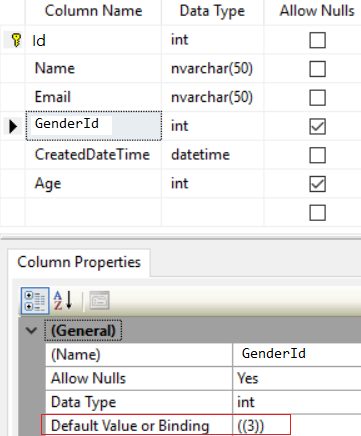
R = Rule (old-style, stand-alone)

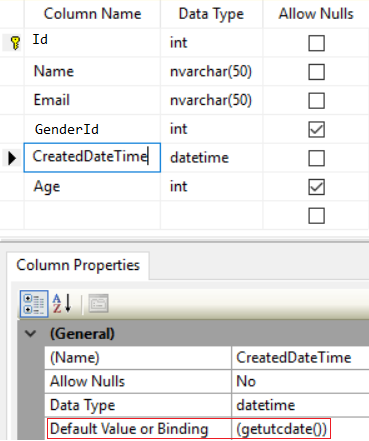
UQ = UNIQUE constraint

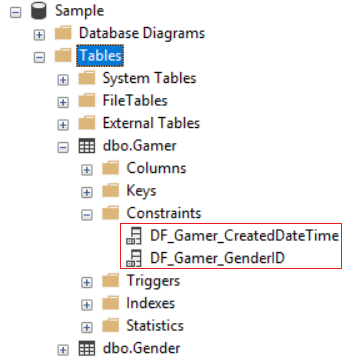
\*/

Text

Description automatically generated with medium confidence







------------------------------------------------------------------------------------------------------

--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](mailto: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](mailto: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

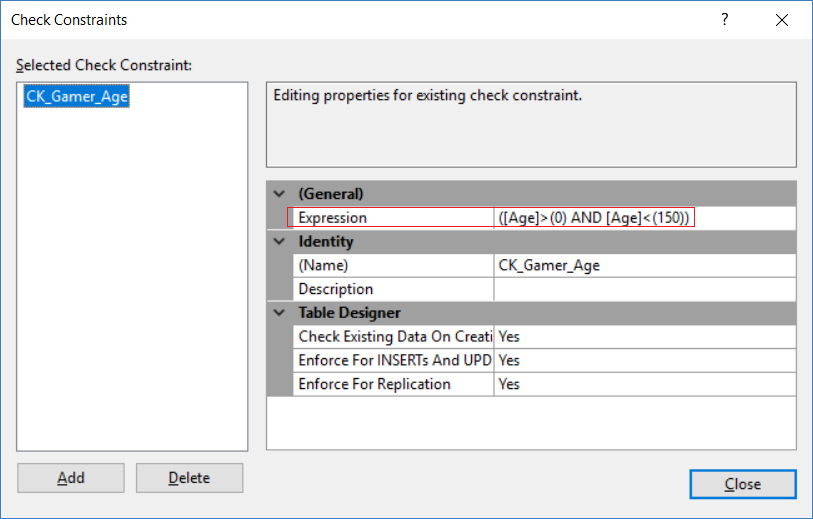
Get all columns from Gamer Table.

\*/



Text

Description automatically generated



------------------------------------------------------------------------------------------------------

--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

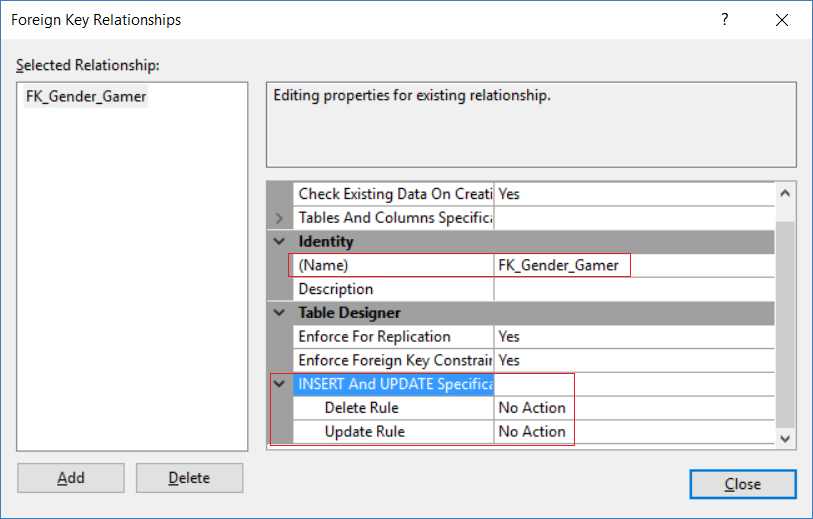
\*/

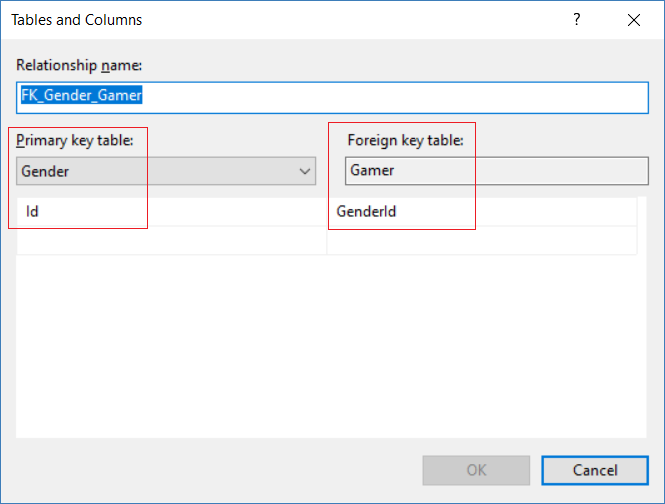
Graphical user interface, diagram

Description automatically generated

Graphical user interface, text

Description automatically generated





------------------------------------------------------------------------------------------------------

--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](mailto: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](mailto: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](mailto: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](mailto:4@4.com)', 1, 24 );

INSERT  Gamer

VALUES  ( N'Name5', N'[5@5.com](mailto: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](mailto: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](mailto: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;

\*/



--====================================================================================================

--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](mailto: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](mailto:7@7.com)', 28 );

GO -- Run the prvious command and begins new batch

SELECT  \*

FROM    Gender;

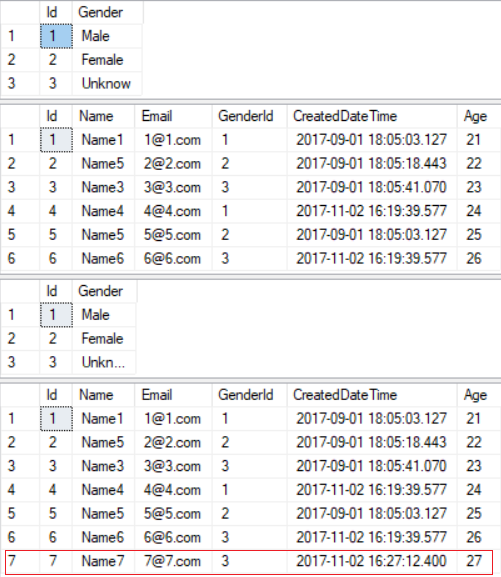
SELECT  \*

FROM    Gamer;

GO -- Run the prvious command and begins new batch

Graphical user interface

Description automatically generated with low confidence



---------------------------------------------------------------------------------------------------------

--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>

\*/



---------------------------------------------------------------------------------------------------------

--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](mailto: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](mailto: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](mailto:7@7.com)';

/\*

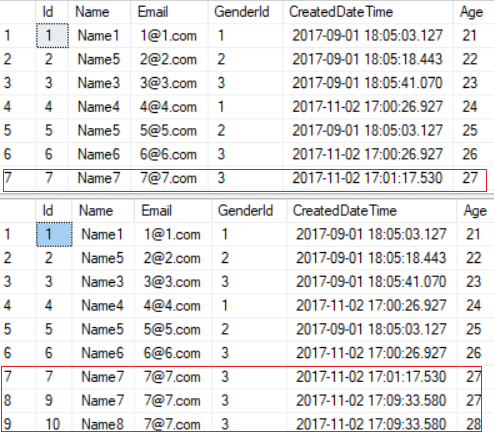
Delete the rows which email is '[7@7.com](mailto:7@7.com)'

\*/

SELECT  \*

FROM    Gamer;

GO -- Run the prvious command and begins new batch





---------------------------------------------------------------------------------------------------------

--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  \*

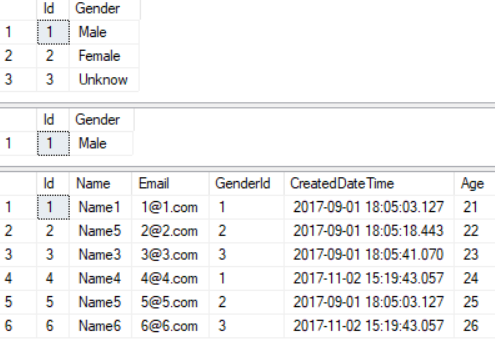
--FROM    Gender

--WHERE   Id = 1;

Get all columns from Gender Table,

Filter the rows where id must be 1.

\*/



---------------------------------------------------------------------------------------------------------

--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';



---------------------------------------------------------------------------------------------------------

--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  \*

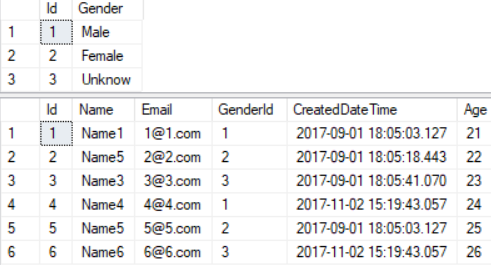
FROM    Gender;

SELECT  \*

FROM    Gamer;

GO -- Run the prvious command and begins new batch





/\*

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

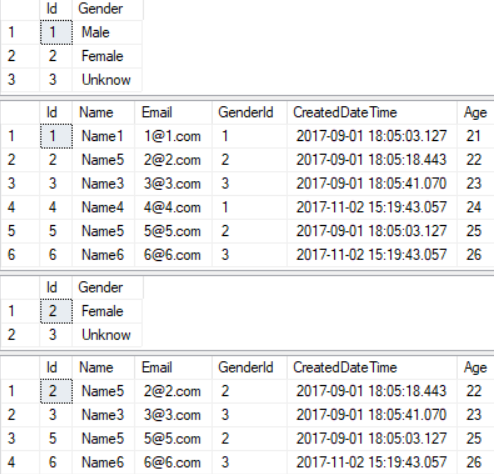
--WHERE   Id = 1;

It also do the following

--DELETE  FROM Gamer

--WHERE   GenderId = 1;

\*/



---------------------------------------------------------------------------------------------------------

--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;

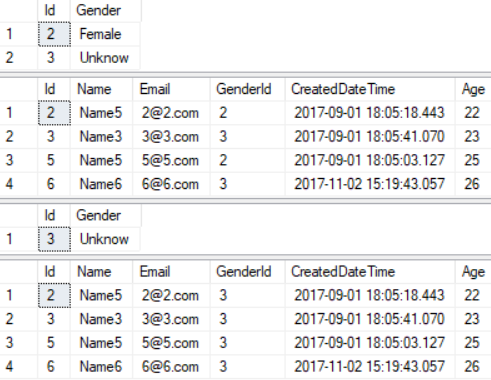
It will also do the following.

--UPDATE Gamer

--SET GenderId = 3

--WHERE GenderId = 2;

\*/



---------------------------------------------------------------------------------------------------------

--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;

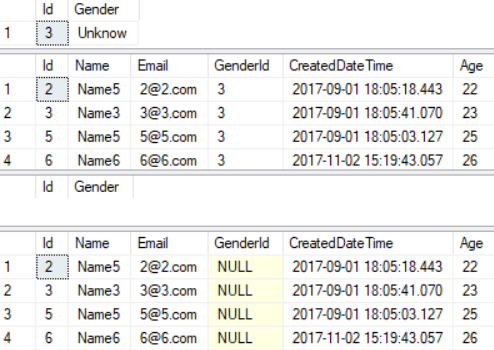
It will also do the following.

--UPDATE Gamer

--SET GenderId = NULL

--WHERE GenderId = 3;

\*/



--====================================================================================================

--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

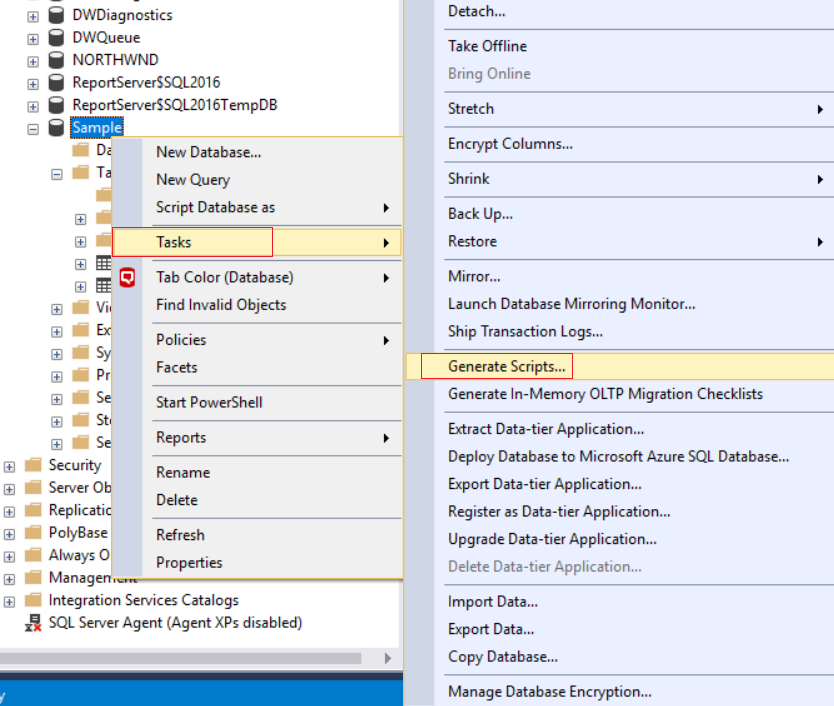
--> Finish

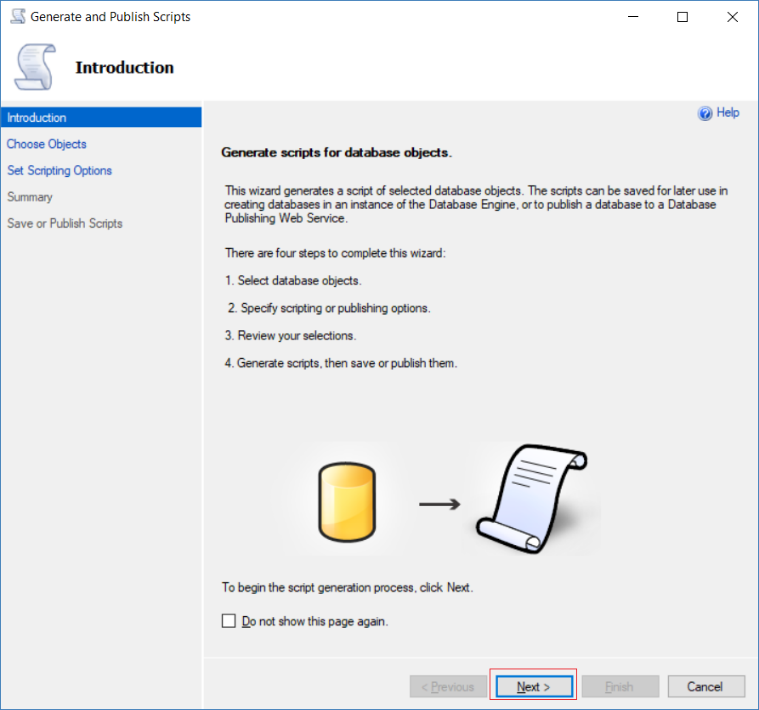
It will create  **script.sql**  file  into  **Document Folder**

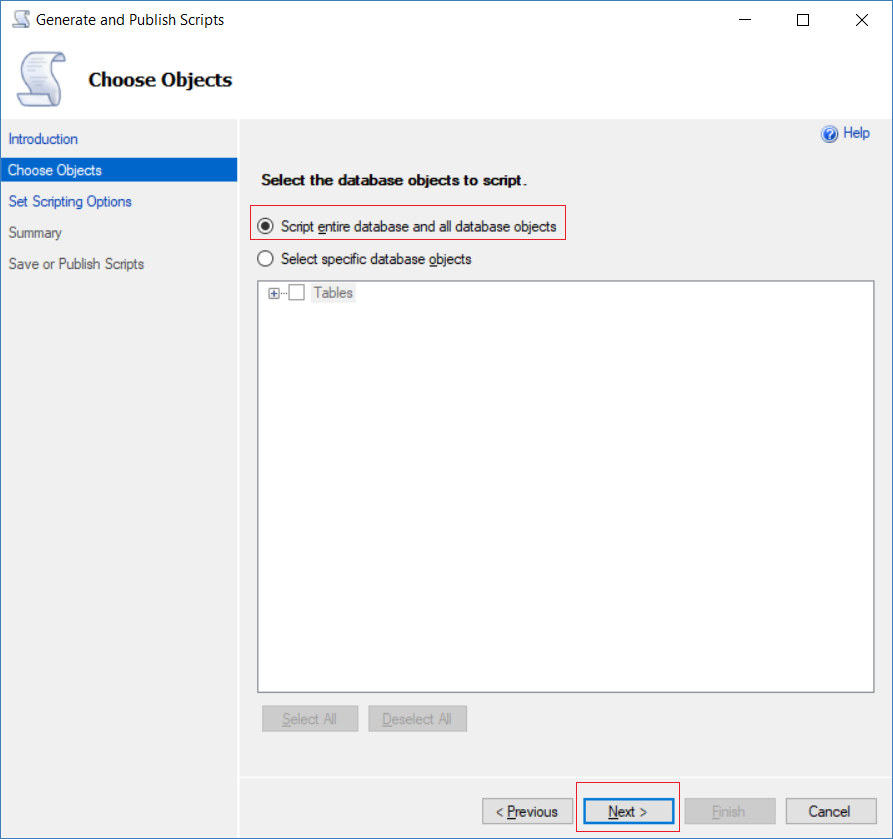
-->

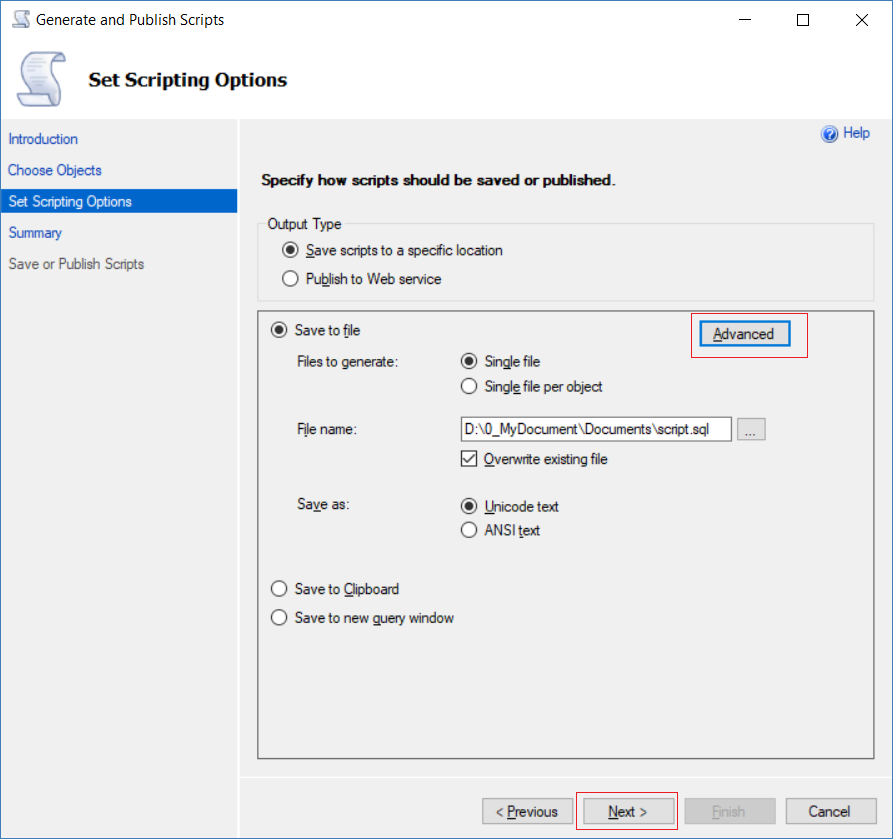
I normally delete create database part.

Please have a look the file,









Graphical user interface, application, table

Description automatically generated

