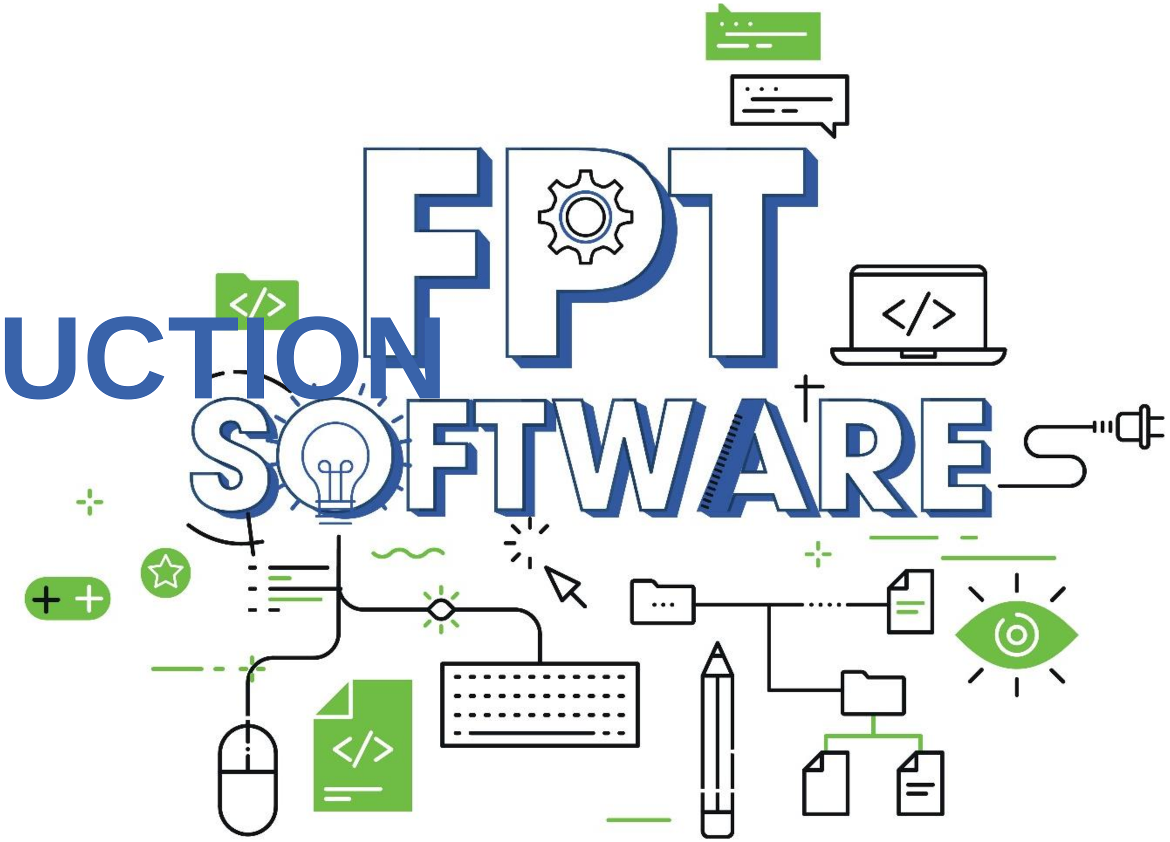
SQL SERVER INTRODUCTION

Fsoft Academy



Lesson Objectives

- *Understand SQL Server and the data types supported by SQL Server*
- *Able to create and modify databases, create tables and constraints on tables*
- *Distinguish between types of Constraints on the table*

Agenda

SQL Server

1. Introduction
   - ✓ What is SQL Server
   - ✓ History SQL Server
   - ✓ SQL Server Edition
   - ✓ Importance of SQL Server Instance
   - ✓ Summary
2. SQL Server Database : Create, Alter, Drop, Backup / Restore
3. SQL Server Data Types
4. SQL Server Table: CREATE, ALTER, DROP
5. SQL Server Keys, Constraints

Section 1

Introduction

Introduction

What is SQL Server ?

SQL SERVER is a relational database management system (RDBMS) developed by Microsoft.

- ✓ SQL Server supports **ANSI SQL**, which is the standard **SQL** (Structured Query Language) language. However, SQL Server comes with its own implementation of the SQL language, T-SQL (Transact-SQL).
- ✓ **T-SQL** is a Microsoft propriety Language known as **Transact-SQL.** It provides further capabilities of declaring variable, exception handling, stored procedure, etc.

TCL ko phải là T-SQL

Version History SQL Server

Microsoft and Sybase released version 1.0 in 1989.

However, the partnership between these two ended in the early 1990s.

Microsoft maintained ownership rights to the name SQL Server.

Since the 1990s, subsequent versions of SQL Server have been released including SQL Server 2000, 2005, 2008, 2012, 2014, 2016, 2017, and 2019

Introduction

SQL Server Editions

- ✓ **SQL Server Enterprise:** It is used in the high end, large scale and mission Critical business. It provides High-end security, Advanced Analytics, Machine Learning, etc.
- ✓ **SQL Server Standard:** It is suitable for Mid-Tier Application and Data marts. It includes basic reporting and analytics.
- ✓ **SQL Server WEB:** It is designed for a low total-cost-of-ownership option for Web hosters. It provides scalability, affordability, and manageability capabilities for small to large scale Web properties.
- ✓ **SQL Server Developer:** It is similar to an enterprise edition for the non-production environment. It is mainly used for build, test, and demo.
- ✓ **SQL Server Express:** It is for small scale applications and free to use.

Importance of SQL Server Instances

The following are the advantages of SQL Server instances:

1. For installation of different versions on one machine
2. For cost reduction
3. For maintenance of development, production and test environments separately
4. For reducing temporary database problems
5. For separating security privileges
6. For maintaining a standby server

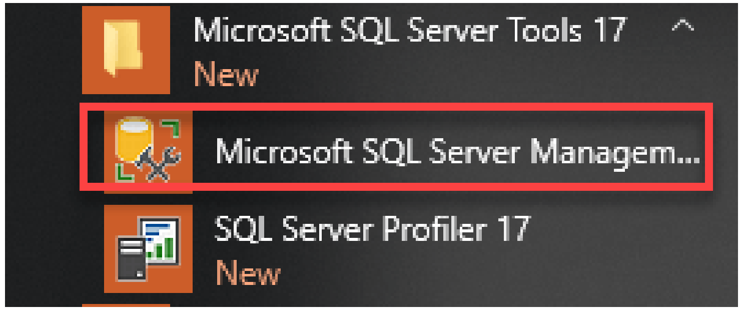
Introduction

- Summary
  - ✓ SQL Server is defined as a relational database management system (RDBMS) developed by Microsoft
  - ✓ T-SQL means Transact-SQL, a propriety Language by Microsoft
  - ✓ Microsoft and Sybase released version 1.0 in 1989
  - ✓ Various Editions of SQL Server are **Enterprise**, **Standard**, **Web**, **Developer**, and **Express**
  - ✓ You can run multiple instances of SQL Server the same on the same machine.

SQL Server Database

- **First**, launch the Microsoft SQL Server Management Studio from the **Start** menu

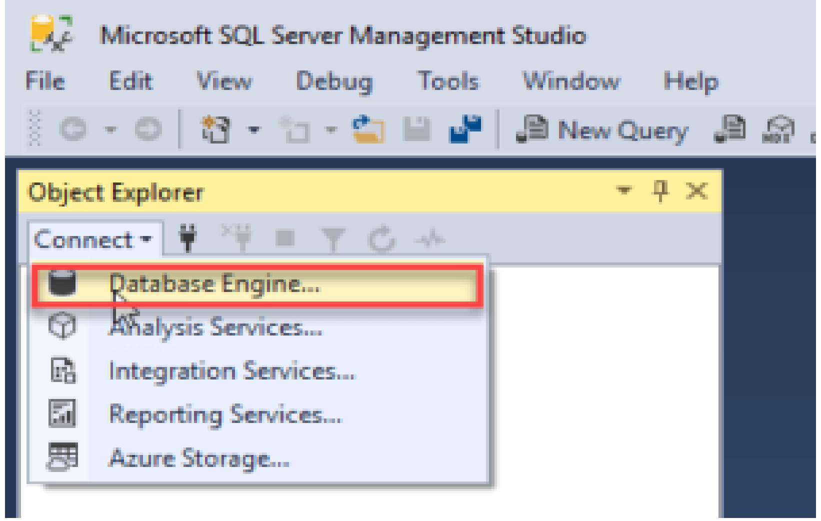
✓ **Connect to SQL Server**



Microsoft SQL Server Tools 17
New
Microsoft SQL Server Managem...
SQL Server Profiler 17
New

[1]

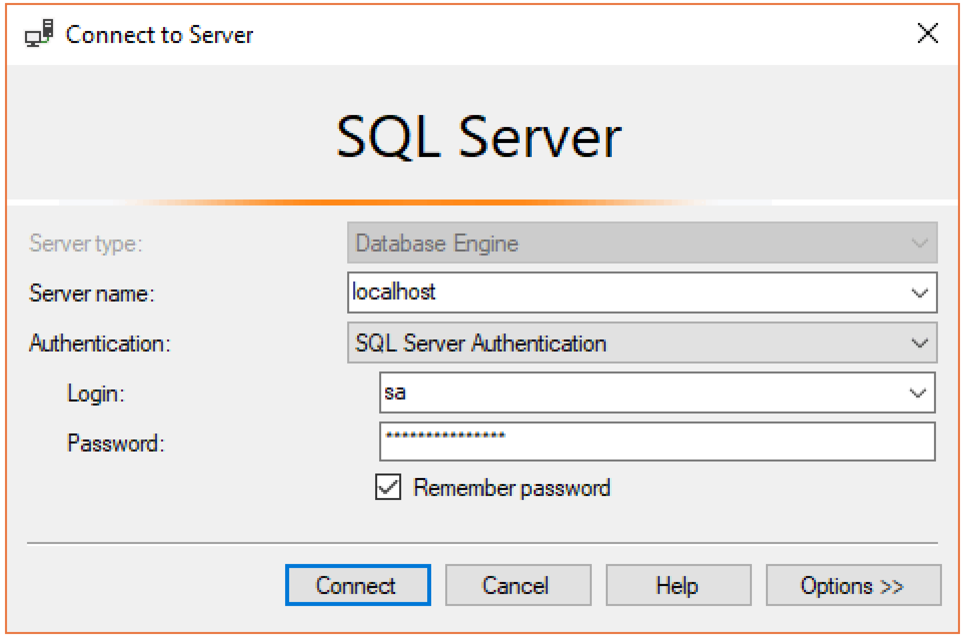
✓ Next, from the **Connect** menu under the **Object Explorer**, choose the **Database Engine**…



[2]

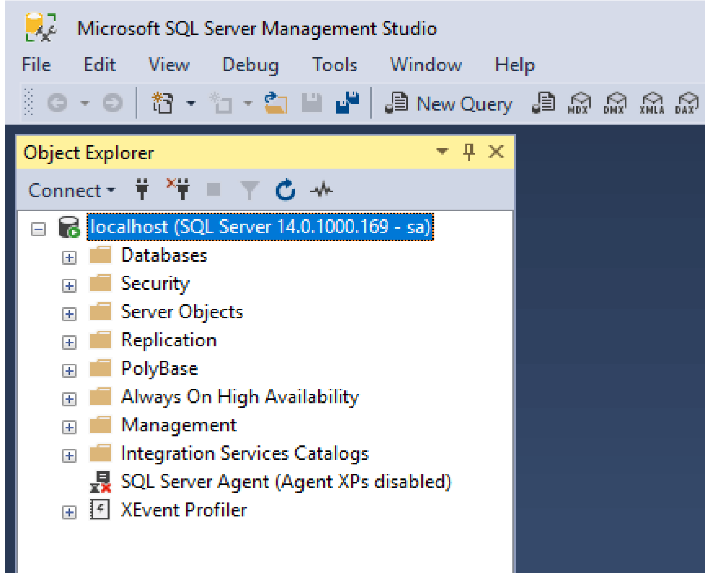
SQL Server Database

- If the connection is established successfully, then you will see the following **Object Explorer** panel:



Server type: Database Engine
Server name: localhost
Authentication: SQL Server Authentication
Login: sa
Password: ****************
☑ Remember password

[3]


Microsoft SQL Server Management Studio
File Edit View Debug Tools Window Help
New Query
Object Explorer
Connect

localhost (SQL Server 14.0.1000.169 - sa)
- Databases
- Security
- Server Objects
- Replication
- PolyBase
- Always On High Availability
- Management
- Integration Services Catalogs
- SQL Server Agent (Agent XPs disabled)
- XEvent Profiler

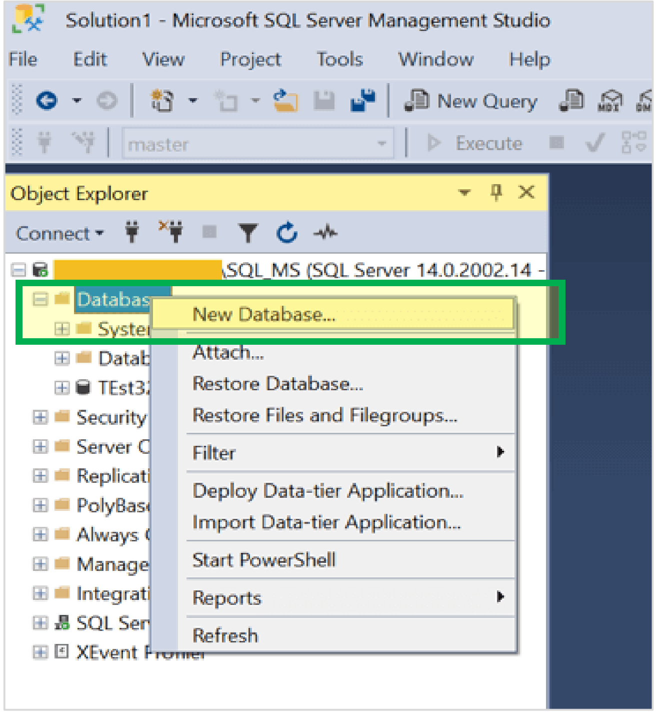
[4]

SQL Server Database

Create Database

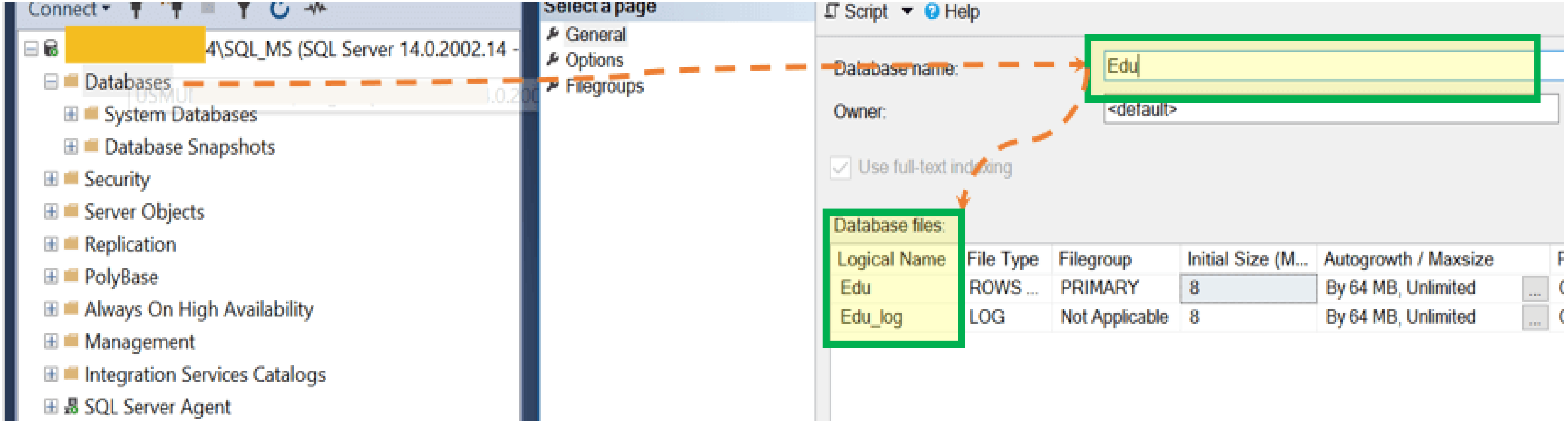
- There are 2 ways to create Database in SQL server.
  1. SQL Server Management Studio
  2. Transact-SQL

1. SQL Server Management Studio



[1.1]

SQL Server Database



[redacted] 4\SQL_MS (SQL Server 14.0.2002.14 -

- Databases
  - System Databases
  - Database Snapshots
- Security
- Server Objects
- Replication
- PolyBase
- Always On High Availability
- Management
- Integration Services Catalogs
- SQL Server Agent

Database name: Edu

Owner: <default>

☑ Use full-text indexing

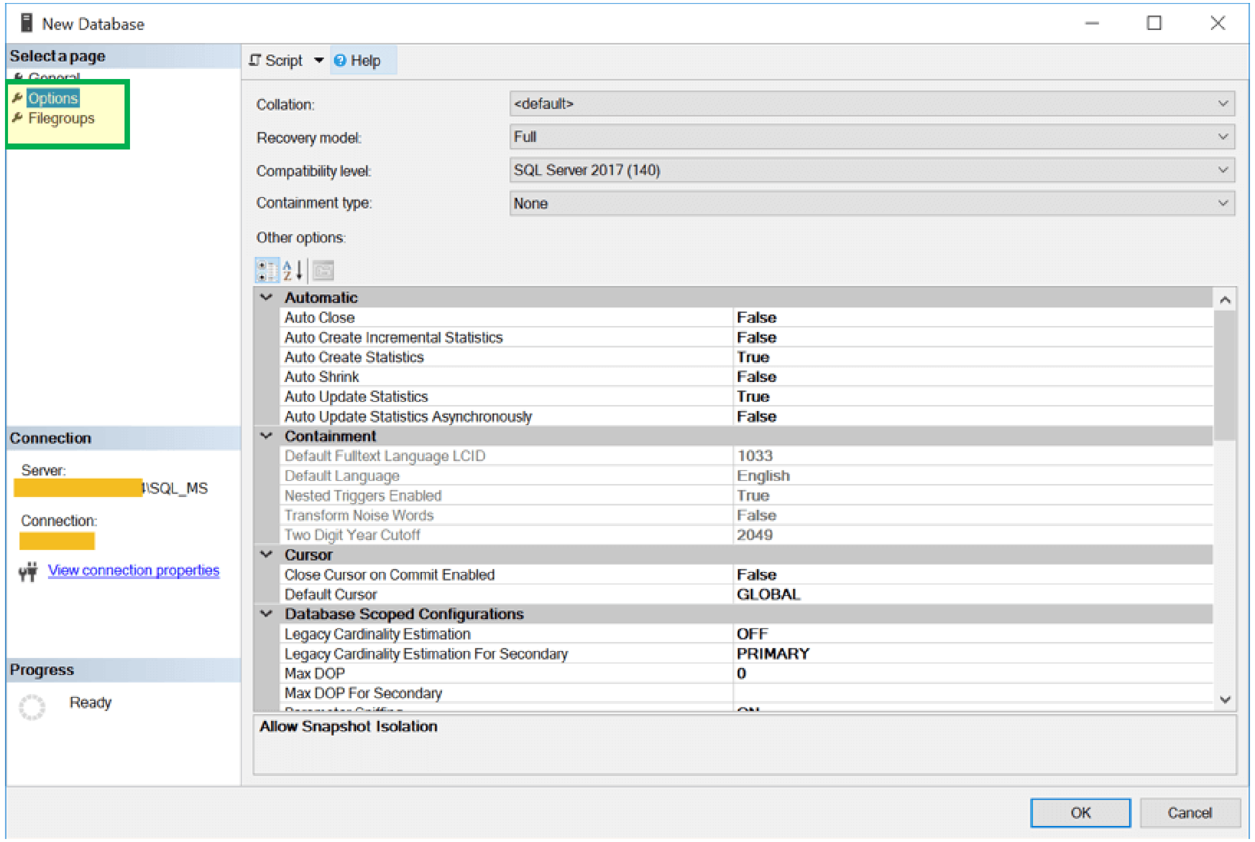
Database files:

Logical Name	File Type	Filegroup	Initial Size (M...	Autogrowth / Maxsize
Edu	ROWS ...	PRIMARY	8	By 64 MB, Unlimited
Edu_log	LOG	Not Applicable	8	By 64 MB, Unlimited

[1.2]

SQL Server Database

[1.3]



New Database

Select a page
- General
- Options
- Filegroups

Script ▾ | Help

Collation: <default>

Recovery model: Full

Compatibility level: SQL Server 2017 (140)

Containment type: None

Other options:

Automatic	
Auto Close	False
Auto Create Incremental Statistics	False
Auto Create Statistics	True
Auto Shrink	False
Auto Update Statistics	True
Auto Update Statistics Asynchronously	False
Containment	
Default Fulltext Language LCID	1033
Default Language	English
Nested Triggers Enabled	True
Transform Noise Words	False
Two Digit Year Cutoff	2049
Cursor	
Close Cursor on Commit Enabled	False
Default Cursor	GLOBAL
Database Scoped Configurations	
Legacy Cardinality Estimation	OFF
Legacy Cardinality Estimation For Secondary	PRIMARY
Max DOP	0
Max DOP For Secondary	

Allow Snapshot Isolation

Connection

Server:
[redacted]\SQL_MS

Connection:
[redacted]

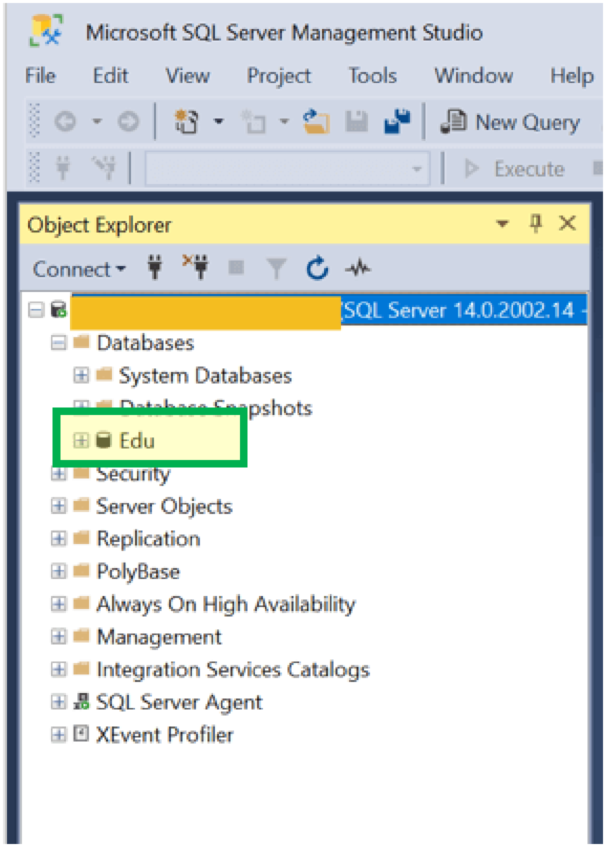
View connection properties

Progress

Ready

OK | Cancel

[Optional – For Advance Setting]



Microsoft SQL Server Management Studio

- [redacted] SQL Server 14.0.2002.14
  - Databases
    - System Databases
    - Database Snapshots
    - Edu
  - Security
  - Server Objects
  - Replication
  - PolyBase
  - Always On High Availability
  - Management
  - Integration Services Catalogs
  - SQL Server Agent
  - XEvent Profiler

[1.4]

SQL Server Database

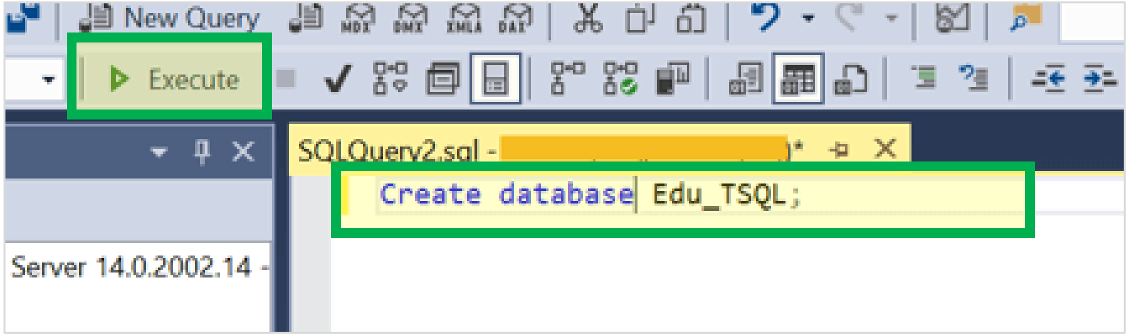
2. Transact-SQL

Syntax:

```
CREATE DATABASE <Database_name>
```

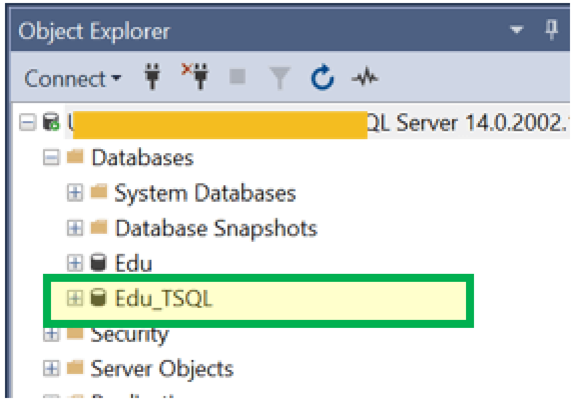
Query:

```
CREATE DATABASE [Edu_TSQL_file]
```

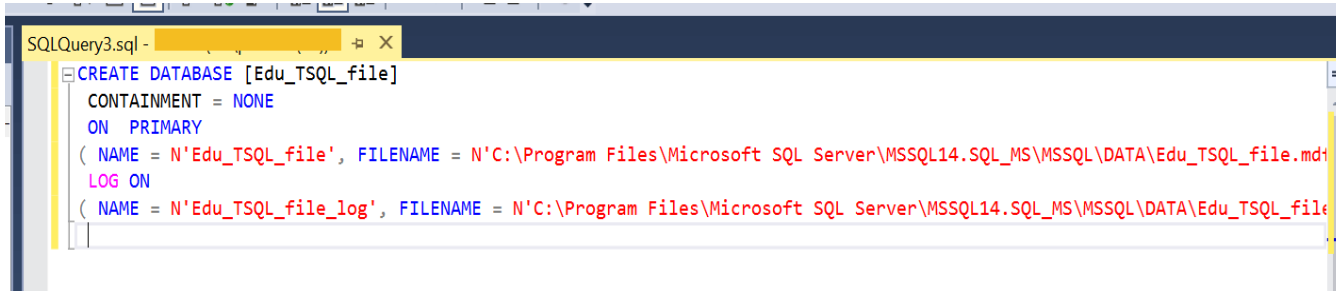


```
Create database Edu_TSQL;
```

[2.1]



[2.2]



```
CREATE DATABASE [Edu_TSQL_file]
 CONTAINMENT = NONE
 ON  PRIMARY
( NAME = N'Edu_TSQL_file', FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL14.SQL_MS\MSSQL\DATA\Edu_TSQL_file.mdf
 LOG ON
( NAME = N'Edu_TSQL_file_log', FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL14.SQL_MS\MSSQL\DATA\Edu_TSQL_file
```

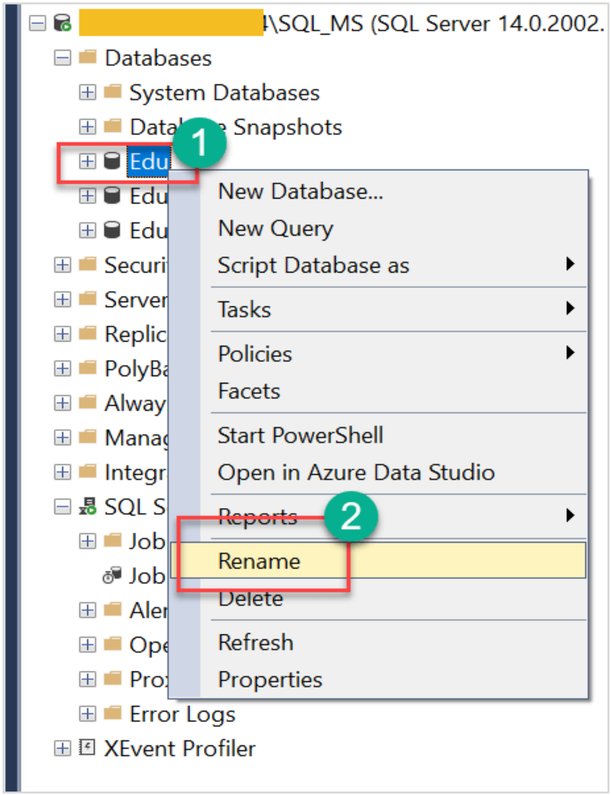
[2.3]

SQL Server Database

Alter Database

- There are 2 ways to alter Database in SQL server.
  1. SQL Server Management Studio
  2. Transact-SQL

1. SQL Server Management Studio



SQL Server Database

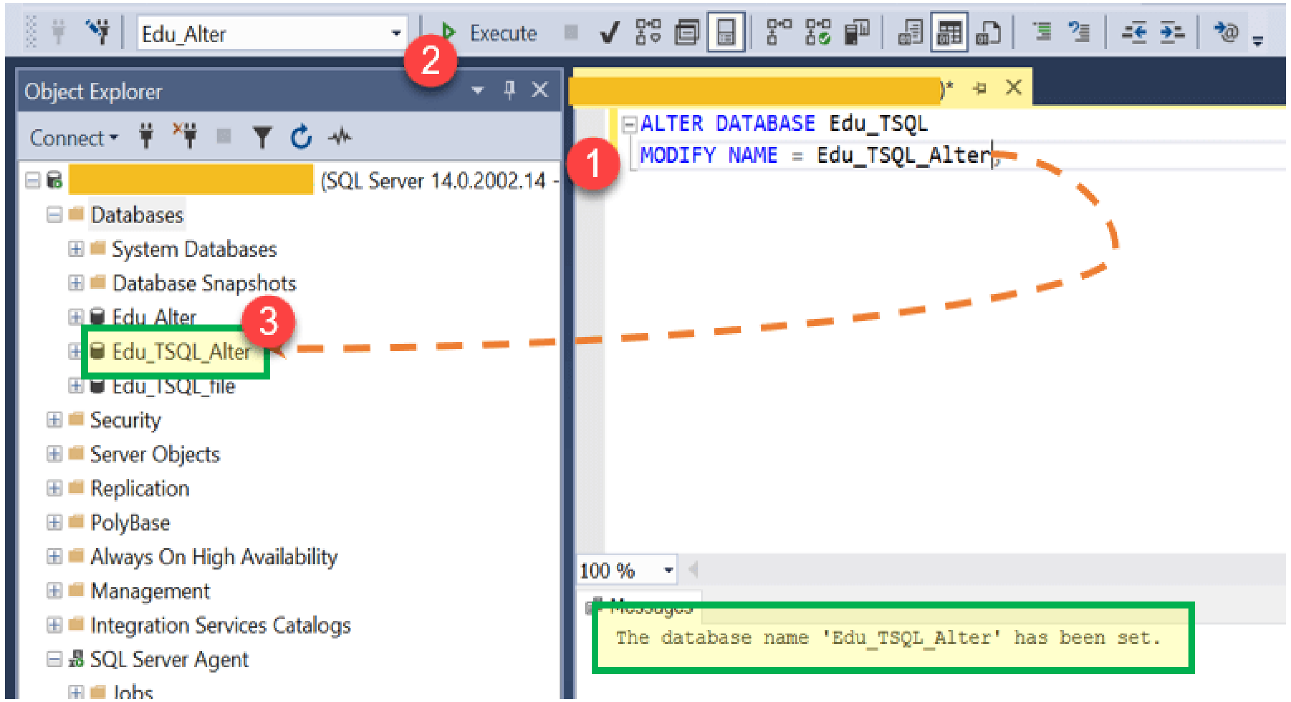
2. Transact-SQL

Syntax:

```
ALTER DATABASE <Databse_name>
MODIFY NAME = <New Name>
```

Query:

```
ALTER DATABASE Edu_TSQL
MODIFY NAME = Edu_TSQL_Alter
```



```
ALTER DATABASE Edu_TSQL
MODIFY NAME = Edu_TSQL_Alter;
```

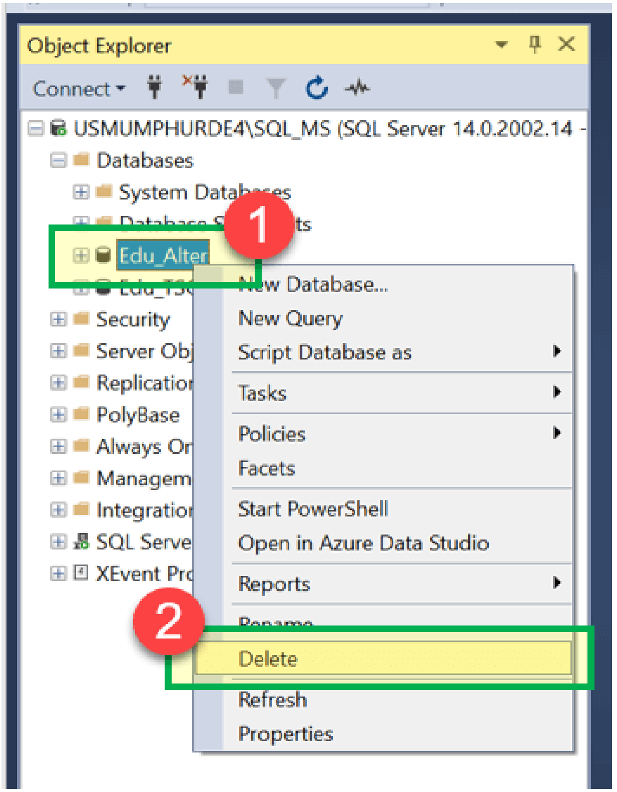
```
The database name 'Edu_TSQL_Alter' has been set.
```

SQL Server Database

Delete Database

- There are 2 ways to Delete Database in SQL server.

1. SQL Server Management Studio



2. Transact-SQL.

Syntax:

DROP DATABASE <Databse_name>

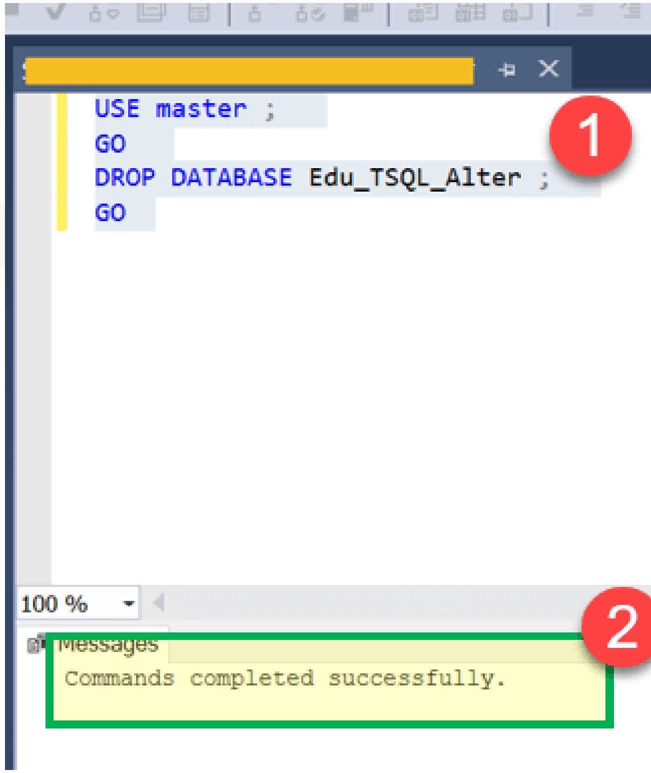
Query:

USE master;

GO

DROP DATABASE Edu_TSQL_Alter;

GO

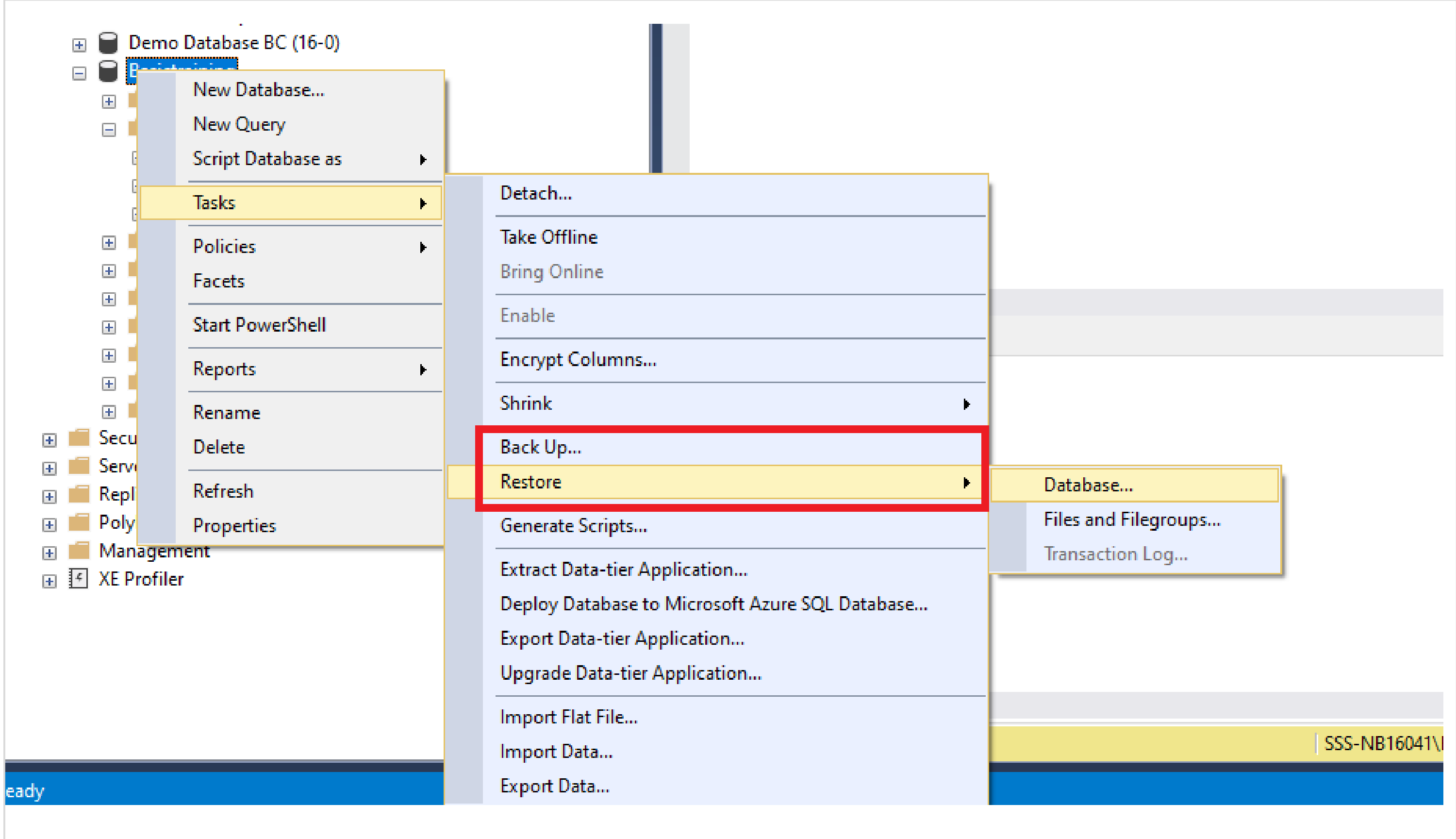


```
USE master ;
GO
DROP DATABASE Edu_TSQL_Alter ;
GO
```

```
Commands completed successfully.
```

SQL Server Database

- Backup / Restore Database



Section 2

SQL Server Data Types

Ms SQL Server Data Types

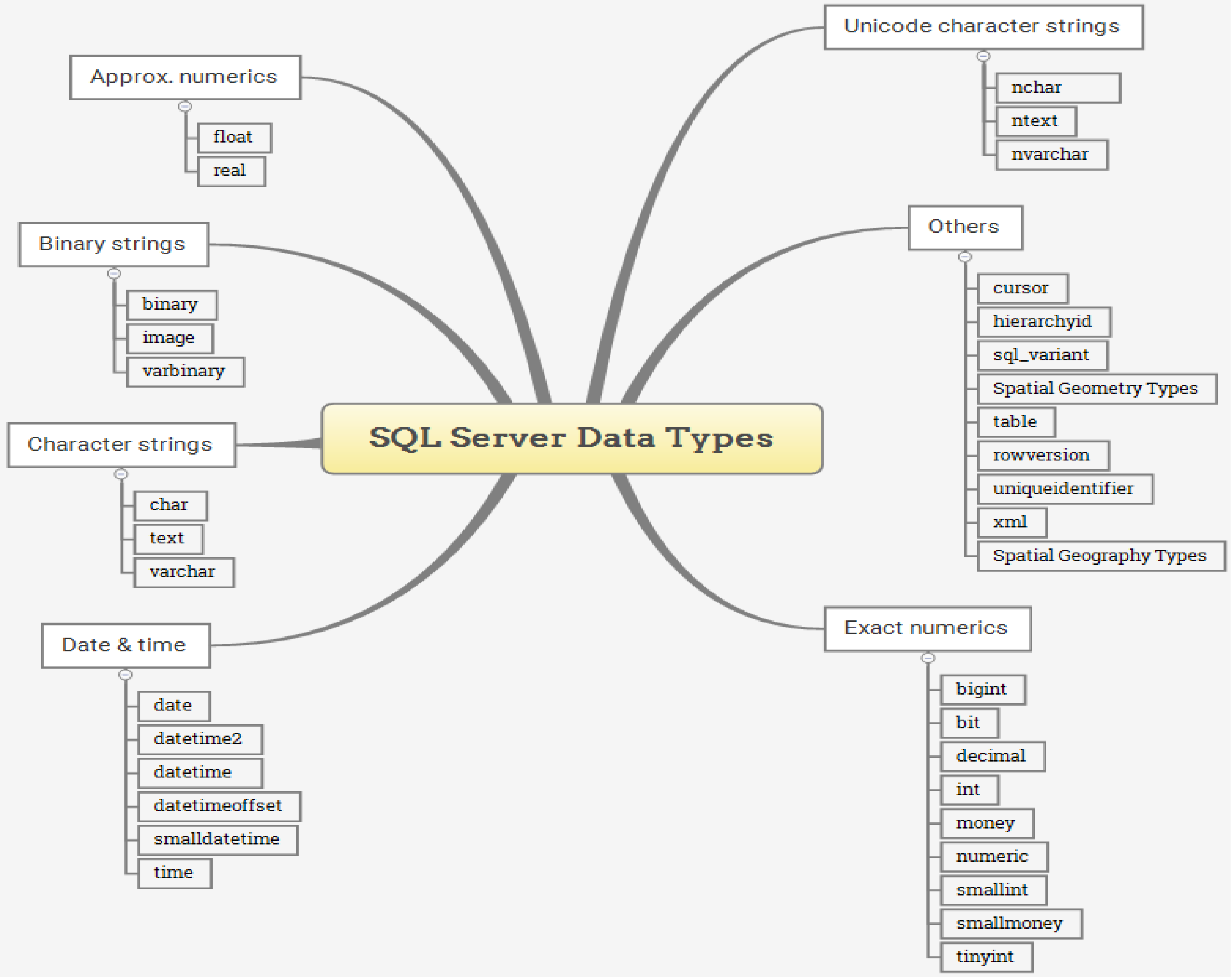
Student:

- ✓ Name
- ✓ Birthday
- ✓ Gender
- ✓ Address
- ✓ Marks…

Storage

What type of data each of field ???.............

SQL Server Data Types



SQL Server Data Types

- Approx. numerics
  - float
  - real
- Binary strings
  - binary
  - image
  - varbinary
- Character strings
  - char
  - text
  - varchar
- Date & time
  - date
  - datetime2
  - datetime
  - datetimeoffset
  - smalldatetime
  - time
- Unicode character strings
  - nchar
  - ntext
  - nvarchar
- Others
  - cursor
  - hierarchyid
  - sql_variant
  - Spatial Geometry Types
  - table
  - rowversion
  - uniqueidentifier
  - xml
  - Spatial Geography Types
- Exact numerics
  - bigint
  - bit
  - decimal
  - int
  - money
  - numeric
  - smallint
  - smallmoney
  - tinyint

Example :

```
DECLARE @Datatype_Date DATE = '2030-01-01'
PRINT @Datatype_Date
```

```
DECLARE @Datatype_Binary BINARY(2) = 12;
PRINT @Datatype_Binary
```

```
DECLARE @Datatype_Char VARCHAR(30) = 'This is Character Datatype'

PRINT @Datatype_Char
```

```
DECLARE @Datatype_Float FLOAT(24) = 22.1234;

PRINT @Datatype_Float
```

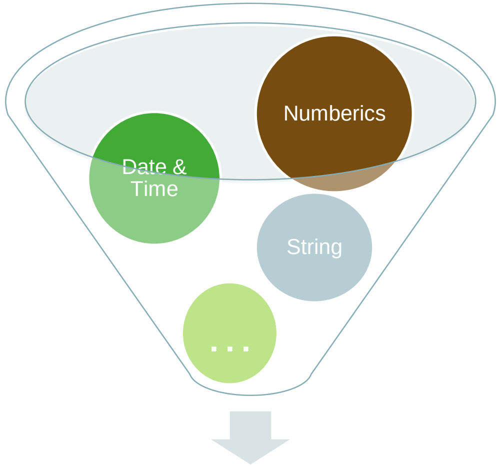
```
DECLARE @Datatype_Decimal DECIMAL (3,2) = 2.31;

PRINT @Datatype_Decimal

-- (P,S) : P : precision , S : scale
```

Ms SQL Server Data Types

- **Data types in SQL Server are organized into the following categories: *NULL is default value for most data type*:**
  - ✓ Exact Numerics
  - ✓ Approximate Numerics
  - ✓ Date and Time
  - ✓ Character Strings
  - ✓ Unicode Character Strings
  - ✓ Binary Strings
  - ✓ Other Data Types



Data types

In SQL Server, based on their storage characteristics, some data types are designated as belonging to the following groups:

- ✓ Large value data types: **varchar(max)**, and **nvarchar(max)**
- ✓ Large object data types: **text**, **ntext**, **image**, **varbinary(max)**, and **xml**

Exact Numbers

Exact-number data types that use **integer data**. To save space in the database, use the **smallest data type** that can reliably contain all possible values.

1/ Interger-based data type

Data type	Range	Range of values	Size
bigint	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807	-2^63 to 2^63-1	8 Bytes
int	-2,147,483,648 to 2,147,483,647	-2^31 to 2^31-1	4 Bytes
smallint	-32,768 to 32,767	-2^15 to 2^15-1	2 Bytes
tinyint	0 to 255	2^0-1 to 2^8-1	1 Byte

Exact Numbers

Exact decimal-based data type

Data type	Size	Range of values
decimal(p,s)	5 - 17 Bytes (depending on precision)	- Varies based on precision setting. - Maximum values are -10^38 +1 through 10^38 -1
(p is the maximum number of all digits (both sides of the decimal point), s is the maximum number of digits after the decimal point)		
numeric(p,s)	…	Identical to Decimal type
smallmoney	4 Bytes	- 214,748.3648 to 214,748.3647
money	8 Bytes	-922,337,203,685,477.5808 To 922,337,203,685,477.5807

Approximate Numerics

Data type	Size	Range of values
float	8 Bytes	- 1.79E+308 to 1.79E+308
float(n)	Depends on the value of n	
	If 1 ≤ n ≤ 24: 4 Bytes (Precision: 7 digits)	4 Bytes: - 3.40E + 38 to 3.40E + 38 8 Bytes: - 1.79E+308 to 1.79E+308
	If 25 ≤ n ≤ 53: 8 Bytes (Precision: 15 digits)	
real	4 Bytes	- 3.40E + 38 to 3.40E + 38

Note: *SQL Server treats n as one of two possible values. If **1**<=n<=**24**, n is treated as **24**. If **25**<=n<=**53**, n is treated as **53**.*

Date and Time

Data Type	Description	Example
date	Stores dates between January 1, 0001, and December 31, 9999	2008-01-15
datetime	Stores dates and times between January 1, 1753, and December 31, 9999, with an accuracy of 3.33 milliseconds precision 1/300 second	2008-01-15 09:42:16.142
datetime2	Stores date and times between January 1, 0001, and December 31, 9999, with an accuracy of 100 nanoseconds	2008-01-15 09:42:16.1420221
datetimeoffset ~ datetime2	Similar to the datetime2 data type, but also expects an offset designation of –14:00 to +14:00	2008-01-15 09:42:16.1420221 +05:00
smalldatetime	Stores dates and times between January 1, 1900, and June 6, 2079, with an accuracy of 1 minute	2008-01-15 09:42:00
time	Stores times with an accuracy of 100 nanoseconds	09:42:16.1420221

Character Strings

Character data types that are either fixed-size, **char**, or variable-size, **varchar**.

Non-Unicode string data types:

Data type	Description
char(n)	- Fixed-length - Maximum length of 8,000 characters (1 ≤ n ≤ 8000)
varchar(n)	- Variable-length - Maximum of 8,000 characters (1 ≤ n ≤ 8000)
varchar(max) MAX là 1 từ khóa	- Variable-length - Maximum length of 2,147,483,647 characters
text	- Variable-length - Maximum length of 2,147,483,647 characters - Use varchar(max) instead

Unicode Character Strings

Character data types that are either fixed-size, **nchar**, or variable-size, **nvarchar**

Unicode string data types are “double width”:

Data type	Description
nchar(n)	- Fixed-length - Maximum specified length is 4,000 characters (1≤ n ≤ 4000)
nvarchar(n)	- Variable-length - Maximum specified length is 4,000 characters (1≤ n ≤ 4000)
nvarchar(max)	- Variable-length - Maximum length of 1,073,741,823 characters
ntext	- Variable-length - Maximum length of 1,073,741,823 characters

Binary Strings

Binary data types of either fixed length or variable length.

Data type	Description
binary	- Fixed-length binary data vd: binary(4) có độ dài là 4 byte thì có thể chứa dữ liệu "0x12345678" - Maximum length of 8,000 bytes
varbinary	- Variable length binary data - Maximum length of 8,000 bytes.
image	- Variable length binary data - Maximum length of 2,147,483,647 bytes.

Other Data Types

Data Type	Description
hierarchyid	Special data type that maintains hierarchy positioning information
uniqueidentifier	Stores a database-wide unique number that gets updated every time a row gets updated
sql_variant	Stores values of various SQL Server-supported data types, except text, ntext, and timestamp
xml	Stores XML data. You can store xml instances in a column or a variable (SQL Server 2005 only).
table	Stores a result set for later processing
rowversion	Is a data type that exposes automatically generated, unique binary numbers within a database. The storage size is 8 bytes.

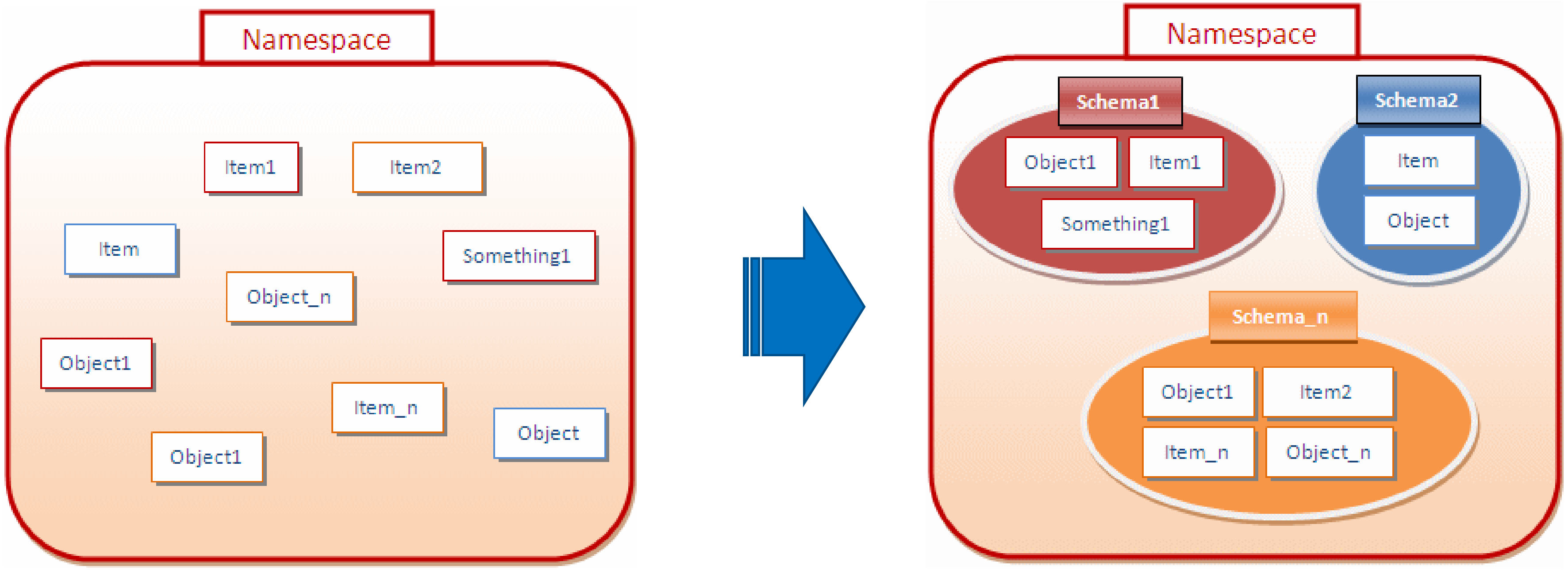
Setion 3

SQL Server Table

Schema Object

Schema là một cách tổ chức các đối tượng trong cơ sở dữ liệu như bảng (tables), views, stored procedures, và các đối tượng khác vào các nhóm.

- A **namespace** can have objects inside



- To further control and manage the objects inside of a namespace, you can put them in *sub-groups* called **schemas**.

Schema Object

- **Schema default:**
  - **dbo** is default schema in every database
  - Ex: SalesOrderDetail, HumanResources.Department
  - [**linked-server**].[**DBName**].[**SchemaName**].[**Objectname**]
- **Schema as:**
  - naming boundaries
  - security boundaries

SQL Server Table (Create, Alter, Drop)

A. CREATE TABLE

We can Create a table in the following ways:

1.T-SQL: Create a New Table by defining all columns and its data type.

2.T-SQL: Create New Table using an existing table

3.Using Table Designer

Interesting Facts!

✓ We can also store big files like **.xml** in a column as **BLOB**, **CLOB** datatype.

✓ Delete can roll back, but Drop cannot be rollback.

1. T-SQL: Create a New Table by defining all

Syntax:

```
CREATE TABLE tableName (
              column_1 datatype [ NULL | NOT NULL ],
              column_2 datatype [ NULL | NOT NULL ], ...

);
```

Query:

```
CREATE TABLE COURSE (
              COURSE_ID INT,
              COURSE_NAME VARCHAR(10)
)
```

SQL Server Table

2. T-SQL: Create New Table using an existing

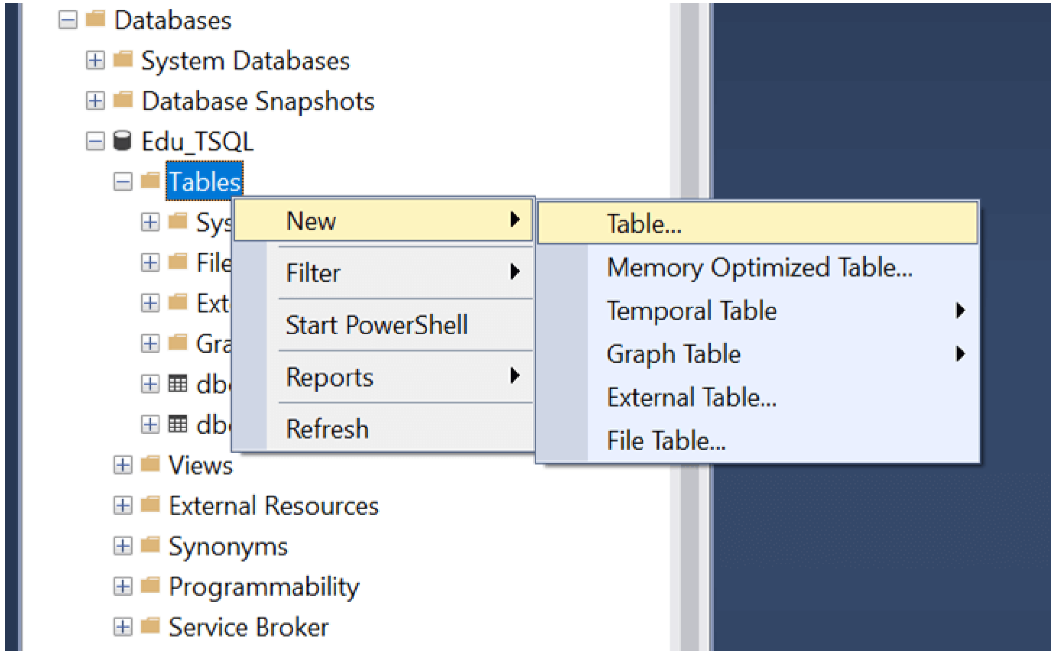
Syntax:

```
SELECT (Column 1, …) INTO <New Table name>
FROM <Old Table name>;
```

Query:

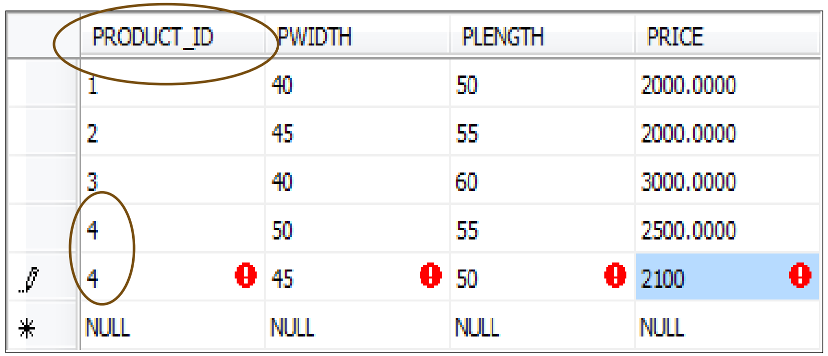
```
SELECT COURSE_NAME
INTO COURSE_NAMES FROM COURSE;
```

3. Using Table Designer



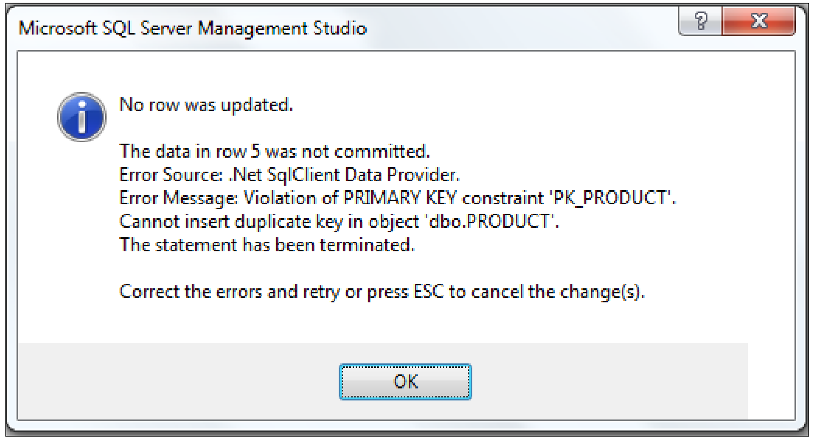
SQL Server Table: Identity

Primary key



PRODUCT_ID	PWIDTH	PLENGTH	PRICE
1	40	50	2000.0000
2	45	55	2000.0000
3	40	60	3000.0000
4	50	55	2500.0000
4	45	50	2100
NULL	NULL	NULL	NULL

Error



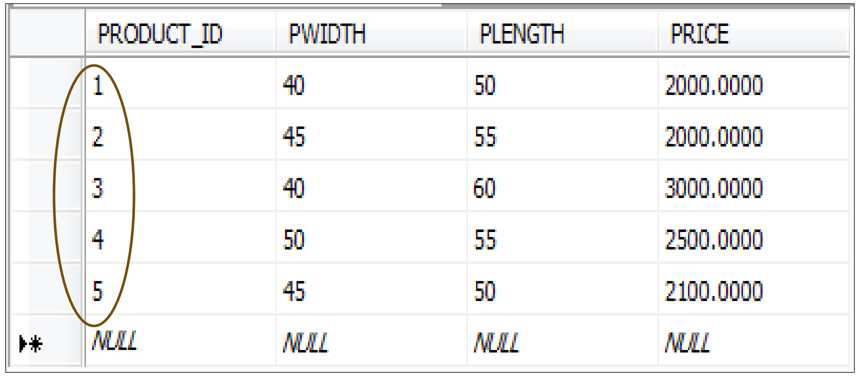
Microsoft SQL Server Management Studio

No row was updated.

The data in row 5 was not committed.
Error Source: .Net SqlClient Data Provider.
Error Message: Violation of PRIMARY KEY constraint 'PK_PRODUCT'.
Cannot insert duplicate key in object 'dbo.PRODUCT'.
The statement has been terminated.

Correct the errors and retry or press ESC to cancel the change(s).

Auto increment



PRODUCT_ID	PWIDTH	PLENGTH	PRICE
1	40	50	2000.0000
2	45	55	2000.0000
3	40	60	3000.0000
4	50	55	2500.0000
5	45	50	2100.0000
NULL	NULL	NULL	NULL

SQL Server Table: Identity

- **Identity has:**
  - ✓A seed: is the initial value
  - ✓An increment: is the value by which we need to skip to fetch the next value
- Seed Increment **For example:**
  - Identity(1,2) will generate sequence numbers 1,3,5,7…

1
2
3
4
5
…

Identity(1,1)

1
4
7
10
13
…

Identity(1,3)

1
3
5
7
9
…

Identity(1,2)

SQL Server Table

B. ALTER TABLE

There are **two ways** to **Alter Table** in **SQL server.**
1. T-SQL: Alter Table by adding new columns.
2. Using Table designer

Below is the syntax to **Alter** table

1. T-SQL: Alter Table by adding new columns.

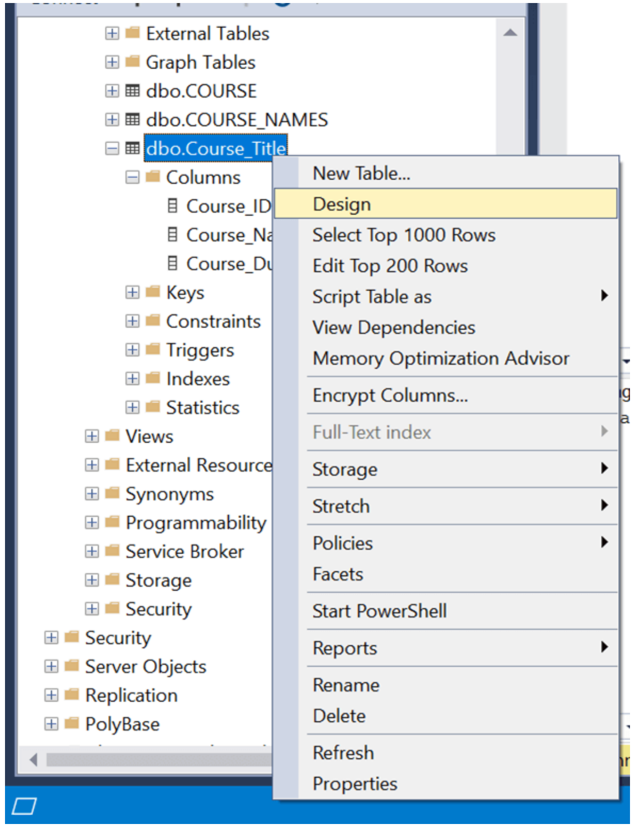
Syntax:

```
ALTER TABLE <Table name>
ADD Column1 datatype, Column2 datatype;
```

Example:

```
ALTER TABLE dbo.Course_Title
ADD Course_Duration VARCHAR(20);
```

2. Using Table designer



SQL Server Table

C. DROP TABLE

We delete the table when it is not required anymore.

There are **two ways** to **Drop Table in SQL server.**

1. Using SQL Server Management Studio.

2. **T-SQL:** Delete Table.

1. T-SQL: Delete Table.

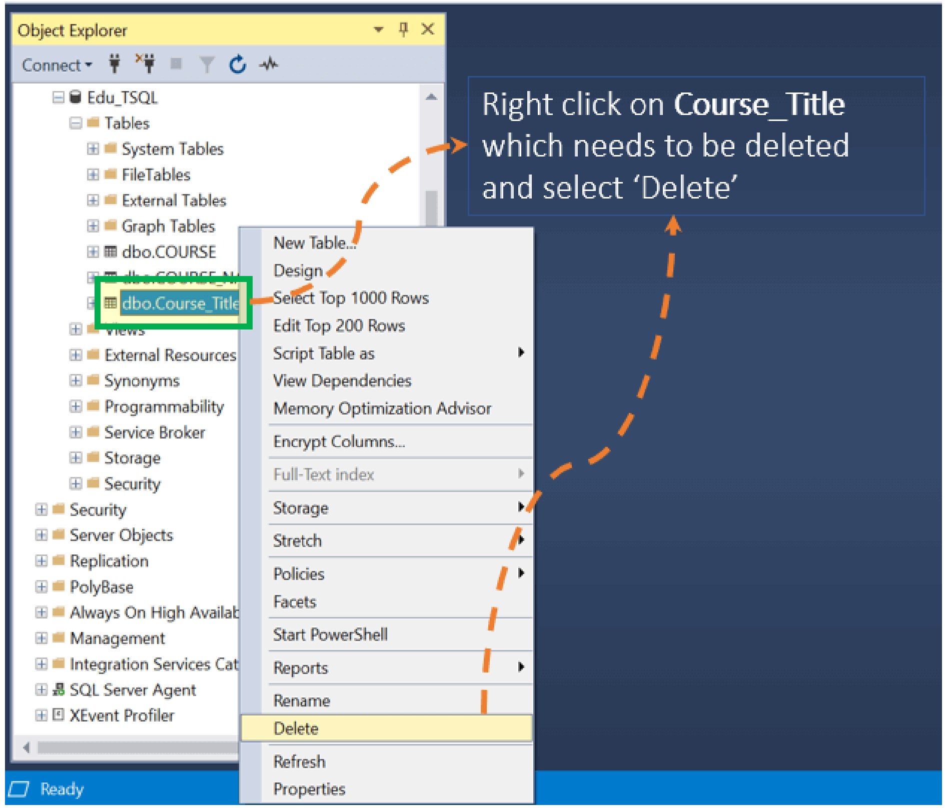
Syntax:

```
DROP TABLE <tableName>;
```

Query:

```
DROP TABLE COURSE_NAMES;
```

2. Using SQL Server Management Studio.



Right click on **Course_Title** which needs to be deleted and select 'Delete'

SQL Constraints Scope

- SQL constraints can be applied at:

  - table level và column level

  - **Table level**
    - ✓ Are declared independently from the column definition
    - ✓ declare table-level constraints at the end of the CREATE TABLE statement
  - **Column level:**
    - ✓ Are declared when define columns for the table.
    - ✓ It is applied particularly to the column where it attached to

SQL Server : Keys, Constraint

a. Keys
   - Primary Key
   - Foreign Key

b. Constraint
   - Unique Constraint
   - Check Constraint

c. Indexes

a1. Primary Key

T-SQL: Create a Primary key while creating a New Table.

Syntax:

```
CREATE TABLE <Table_Name> (
            Column1 datatype,
            Column2 datatype,
            CONSTRAINT <Name> PRIMARY KEY (Column name) .
);
```

Example:

```
CREATE TABLE COURSE_TSQL_PK (
            COURSE_ID INT NOT NULL,
            COURSE_NAME VARCHAR(20),
            CONSTRAINT PK PRIMARY KEY (COURSE_ID)
) ;
```

SQL Server : Keys, Constraint

a. Keys
   - Primary Key
   - Foreign Key

b. Constraint
   - Unique Constraint
   - Check Constraint

a1. Primary Key

T-SQL: Add a Primary key to existing table using Alter Table

Syntax:

```
ALTER TABLE tableName
ADD CONSTRAINT constraintName PRIMARY KEY (column_1, column_2, ... column_n);
```

Example:

```
ALTER TABLE students
ADD CONSTRAINT students_pk PRIMARY KEY (admission);
```

Interesting Facts!

✓ The Primary key can be a combination of multiple columns. This combination is known as the **Composite primary key**.

✓ The Primary key can have a maximum of **16 columns.**

SQL Server : Keys, Constraint

Composite Key

- A composite key is made by the combination of **two or more columns** in a table that can be used to uniquely identify each row in the table when the columns are combined uniqueness of a row is guaranteed.
- **Example**:

```
CREATE TABLE Student(
        rollNumberINT,
        student_nameVARCHAR(30),
        classVARCHAR(30),
        sectionVARCHAR(1),
        mobileVARCHAR(10),
        PRIMARY KEY (rollNumber, mobile)
);
```

SQL Server : Keys, Constraint

a2. Foreign Key: Used to define relationships between tables in the database.

- We can create a **Foreign Key** in SQL server in **2 ways:**

  1. SQL Server Management Studio
  2. T-SQL

- **Parent Table:** Say, we have an existing Parent table as 'Course.' COURSE_ID and COURSE_NAME are two columns with ***COURSE_ID*** as Primary Key.
- **Child Table:** We need to create the second table as a child table. 'COURSE_ID' and 'COURSE_STRENGTH' as two columns. However, ***'COURSE_ID***' shall be Foreign Key.

SQL Server : Keys, Constraint

a2. Foreign Key

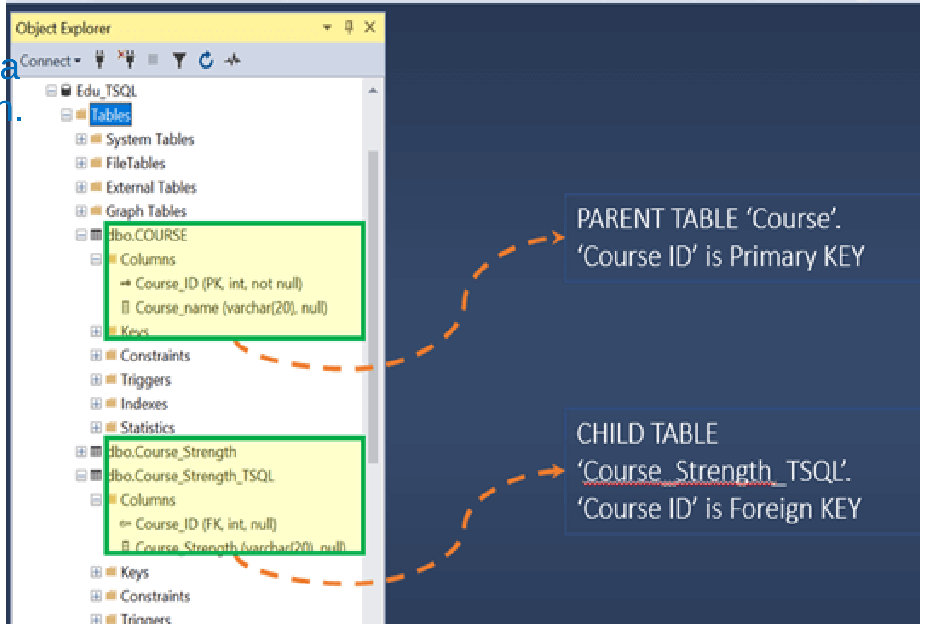
set null: Khi một hàng trong bảng cha bị xóa hoặc cập nhật, các cột trong bảng con tham chiếu đến hàng đó sẽ được đặt thành NULL.
Lưu ý: Các cột trong bảng con phải được khai báo là NULL.

Syntax:

```
CREATE TABLE childTable (
        column_1 datatype [ NULL |NOT NULL ],
        column_2 datatype [ NULL |NOT NULL ], ...
        CONSTRAINT fkey_name
        FOREIGN KEY (child_column1, child_column2, ... child_column_n)
        REFERENCES parentTable (parent_column1, parent_column2,
        ... parent_column_n)
        [ ON DELETE { NO ACTION |CASCADE |SET NULL |SET DEFAULT } ]
        [ ON UPDATE { NO ACTION |CASCADE |SET NULL |SET DEFAULT } ]
);
```

NO ACTION Không cho phép xóa hoặc cập nhật 1 hàng trong bảng cha nếu nó có foreignkey tham chiếu đến.

Khi một hàng trong bảng cha bị xóa hoặc cập nhật, các cột trong bảng con tham chiếu đến hàng đó sẽ được đặt về giá trị mặc định (default) đã định nghĩa trước



PARENT TABLE 'Course'.
'Course ID' is Primary KEY

CHILD TABLE
'Course_Strength_TSQL'.
'Course ID' is Foreign KEY

Query:

```
CREATE TABLE Course_Strength_TSQL (
        COURSE_ID INT,
        COURSE_STRENGTH VARCHAR(20)
        CONSTRAINT FK FOREIGN KEY (Course_ID) REFERENCES COURSE (Course_ID)
)
```

SQL Server : Keys, Constraint

a2. Foreign Key

Using ALTER TABLE

Syntax:

```
ALTER TABLE ChildTable
ADD CONSTRAINT fkey_name FOREIGN KEY (child_column1, child_column2, ... child_column_n)
REFERENCES parentTable (parent_column1, parent_column2, ... parent_column_n);
```

Query:

```
ALTER TABLE Department
ADD CONSTRAINT fkey_student_admission FOREIGN KEY (admission)
REFERENCES students (admission);
```

SQL Server : Keys, Constraint

b1. Unique Constraint

Create unique Contraint - Using a CREATE TABLE statement

```
CREATE TABLE table_name (
        column1 datatype [ NULL | NOT NULL ],
        column2 datatype [ NULL | NOT NULL ], ...
        CONSTRAINT constraint_name UNIQUE (uc_col1, uc_col2, ... uc_col_n)
);
```

```
CREATE TABLE Employees (
        employee_id INT PRIMARY KEY,
        employee_number INT NOT NULL,
        last_name VARCHAR(50) NOT NULL,
        first_name VARCHAR(50),
        salary MONEY,
        CONSTRAINT employees_unique UNIQUE (employee_number)
);
```

Create unique contraint - Using an ALTER TABLE statement

```
ALTER TABLE table_name
ADD CONSTRAINT constraint_name
UNIQUE (column1, column2, ... column_n);
```

```
ALTER TABLE Employees
ADD CONSTRAINT employees_unique
UNIQUE (employee_number);
```

Drop Unique Constraint

```
ALTER TABLE table_name DROP CONSTRAINT constraint_name;
```

SQL Server : Keys, Constraint

b2. Check Constraint

Using a CREATE TABLE statement

```
CREATE TABLE table_name (
        column1 datatype [ NULL | NOT NULL ],
        column2 datatype [ NULL | NOT NULL ], ...
        CONSTRAINT constraint_name
        CHECK [ NOT FOR REPLICATION ] (column_name condition)
);
```

```
CREATE TABLE employees (
        employee_id INT NOT NULL,
        last_name VARCHAR(50) NOT NULL,
        first_name VARCHAR(50), salary MONEY,
        CONSTRAINT check_employee_id
        CHECK (employee_id BETWEEN 1 and 10000)
);
```

Using an ALTER TABLE statement

```
ALTER TABLE table_name
ADD CONSTRAINT constraint_name CHECK (column_name condition);
```

```
ALTER TABLE employees
ADD CONSTRAINT check_last_name
CHECK (last_name IN ('Smith', 'Anderson', 'Jones'));
```

Drop a Check Constraint

```
ALTER TABLE table_name DROP CONSTRAINT constraint_name;
```

PRIMARY KEY and UNIQUE KEY

chính là UNIQUE constraint
ví dụ như cột email

Parameter	PRIMARY KEY	UNIQUE KEY
Basic	Used to serve as a unique identifier for each row in a table.	Uniquely determines a row which isn't primary key.
NULL value acceptance	Cannot accept NULL values.	Can accepts NULL values.
Number of keys that can be defined in the table	Only one primary key	More than one unique key
Index	Creates clustered index	Creates non-clustered index
Auto Increment	A Primary key supports auto increment value.	A unique key does not supports auto increment value.
Modification	We cannot change or delete values stored in primary keys.	We can change unique key values.

chỉ có 1 clustered index trên mỗi bảng, còn non-clustered index thì có thể có nhiều!

TRUNCATE statement

- TRUNCATE statement is a Data Definition Language (DDL) operation that is used to mark the extents of a table for deallocation (empty for reuse).
- The result of this operation quickly removes all data from a table, typically bypassing a number of integrity enforcing mechanisms.
- It was officially introduced in the SQL:2008 standard.
- The TRUNCATE TABLE mytable statement is logically (though not physically) equivalent to the DELETE FROM mytable statement (without a WHERE clause).

TRUNCATE statement

- Removes all rows in a table.
- Table structure and its columns, constraints, indexes, …remain.
  - ✓ Resets the identity value.
  - ✓ Releases the memory used.

DROP vs TRUNCATE

- Truncate is normally ultra-fast and its ideal for deleting data from a temporary table.
- Truncate preserves the structure of the table for future use, unlike drop table where the table is deleted with its full structure.
- Table or Database deletion using DROP statement cannot be rolled back, so it must be used wisely.

SUMMARY

1. Introduction
   - ✓ What is SQL Server
   - ✓ History SQL Server
   - ✓ SQL Server Edition
   - ✓ Importance of SQL Server Instance
   - ✓ Summary
2. SQL Server Database : Create, Alter, Drop, Backup / Restore
3. SQL Server Data Types
4. SQL Server Table: CREATE, ALTER, DROP
5. SQL Server Keys, Constraints

Reference

- https://www.techonthenet.com/sql_server/
- https://www.guru99.com/ms-sql-server-tutorial.html
- https://www.sqlservertutorial.net/?s=constraint

THANK YOU!