FACULTY OF TECHNOLOGY
UNIVERSITY OF COLOMBO
SRI LANKA

Database Management Systems -II

Lecture 6: Transact - SQL

Department of Information and Communication Technology

Outline of the syllabus

- Week 1 – Introduction, Revision of Database Design Process and ER
- Week 2 – Enhanced Entity Relationship (EER) -1
- Week 3 – EER Specialization/Generalization Hierarchies, Lattices & Shared Subclasses
- Week 4 – Functional Dependences and Normalization
- Week 5 – Relation Algebra (Revised) and Transact -SQL
- Week 6 – Stored Procedures and Functions
- Week 7 – Data Storages – Disks and Files

Outline of the syllabus

- Week 8 – Database Connectivity
- Week 9 – Mid Semester Exam
- Week 10 – File Organization and Indexes
- Week 11 – Query processing
- Week 12 - Physical Database Design and Turning Overview
- Week 13– Transaction and Concurrency Control
- Week 14 – Crash Recovery

In this lecture you will learn

- SQL - Review
- Fundamentals of SQL Server
- SQL and MySQL
- Transact-SQL Datatypes
- TempDB
- Transaction Basics

SQL: Review

- Structured Query Language (SQL) was initially called SEQUEL (for Structured English QUEry Language) in 1970's.
- It was developed for an experimental relational database system called System R
- A joint effort between ANSI (American National Standard Institute) and ISO (International Standards Organization) led to a standard version of SQL in 1986 (SQL1, SQL-86, etc.)
- Major revisions have been proposed and SQL2 (also called SQL-92) has subsequently been developed
- SQL3 extends SQL to include object-oriented concepts (recent standard)
- SQL is a platform-independent query language.

SQL Further divide

- **DDL(Data Definition Language):** DDL is used to specify the relational database schema, which is usually hidden from the database users. Some DDL commands are **CREATE, DROP, ALTER, TRUNCATE**, etc.

- **DML(Data Manipulation Language):** DML enables a database user to access or modify the data stored in the database that is organized by appropriate data models. Some of the DML commands are **SELECT, INSERT, UPDATE, DELETE**, etc.

- **DCL(Data Control Language):** DCL is a language used to control access to data stored in the database. Some of the DCL commands are **GRANT, REVOKE**, etc.

- **TCL(Transaction Control Language):** TCL deals with the transactions within the relational database. Some of the TCL commands are **COMMIT, ROLLBACK, SAVEPOINT**, etc.

MS-SQL Server

- The SQL Server is developed by Microsoft in the year 1989 for commercial purposes.
- SQL Server is proprietary software or an RDBMS tool that executes the SQL statements.
- Instead of the SQL, MS-SQL Server uses **Transact-SQL** which is almost same as SQL with slight variations in the queries.
- **Transact-SQL** is a Microsoft product and is an extension of SQL language.
- The SQL Server is platform-dependent, different software is available for different platforms.
- SQL Server is supported by the Microsoft Windows and Linux operating systems.
- The latest version of the SQL Server is 15.0, which is released in the year 2019.
- SQL Server is easy to use and provides better functionalities and user-friendly operations.

Difference Between SQL Server and SQL

- SQL Server is a Database Engine. Oracle is also a Database Engine
- SQL is a Query Language
- Whenever we perform database operations, we think that SQL itself executes those operations. But it is not so.
- SQL is a query language that is used to write the database queries, which are executed by some database engines like SQL Server.

SQL and MySQL

- SQL is a language used for operating different relational databases, MySQL boasts of being the first open-source relational database in the early 90s.
- SQL is a query language, whereas MySQL is a relational database that uses SQL to query a database.
- Use SQL to access, update, and manipulate the data stored in a database. MySQL is a database that stores the existing data in a database in an organized manner.
- SQL is used for writing queries for databases, MySQL facilitates data storing, modifying, and management in a tabular format

SQL and MySQL (Contd.)

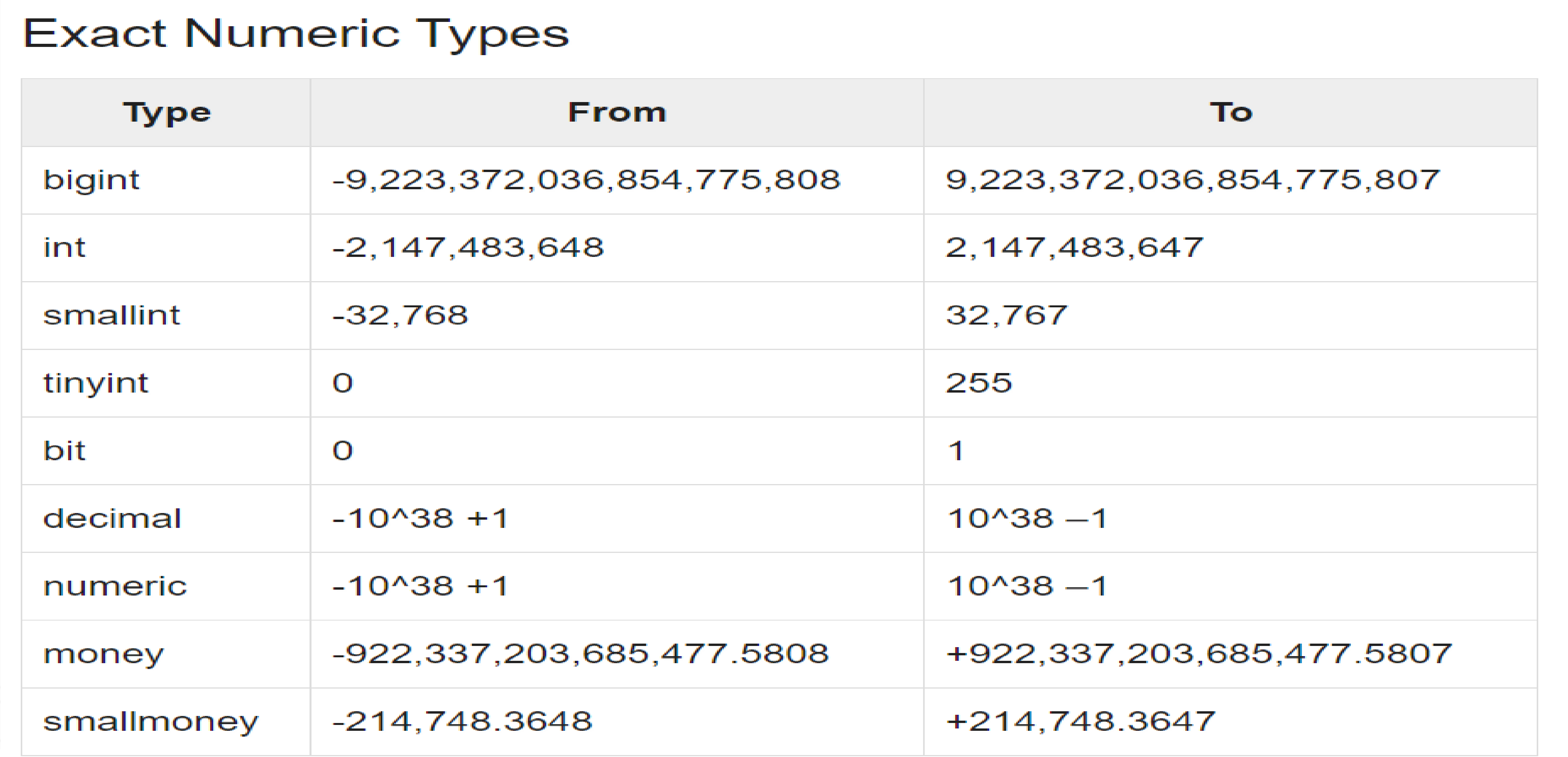
- SQL follows a standard format wherein the basic syntax and commands used for DBMS and RDBMS remain pretty much the same, whereas MySQL receives frequent updates.
- SQL supports a single storage engine, but MySQL supports multiple storage engines and also plug-in storage engines. Thus, MySQL is more flexible.
- *In* MySQL, the server blocks the database during a data backup session, minimizing data corruption chances when switching from one MySQL version to another.
- In terms of data security, the SQL server is much more secure than the MySQL server. In SQL, external processes (like third-party apps) cannot access or manipulate the data directly. While in MySQL, one can easily manipulate or modify the database files during run time using binaries.

SQL and Ttansact-SQL

- SQL is mainly used to issue or execute a single query with insert/update/delete/select. User can not be able to execute multiple statements at a same time.
- T-SQL is used to do the different transactional activities in SQL server programming. User can use multiple programming techniques while using TSQL.
- TSQL provides all the functionality of SQL but with some added extras.
- Database companies are allowed to add their own extensions to the rigid specifications of standard SQL. TSQL is an example for this.
- SQL is open-source. T-SQL is developed and owned by Microsoft.
- SQL statements are executed one at a time, also known as "non-procedural." T-SQL executes statements in a "procedural" way, meaning that the code will be processed as a block, logically and in a structured order.

T-SQL Data types

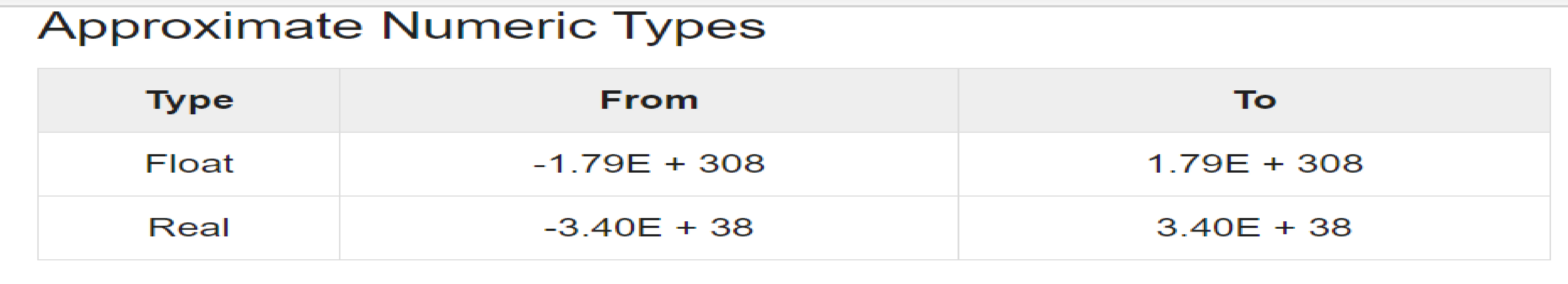
SQL Server offers seven categories including other category of data types for use.



Exact Numeric Types

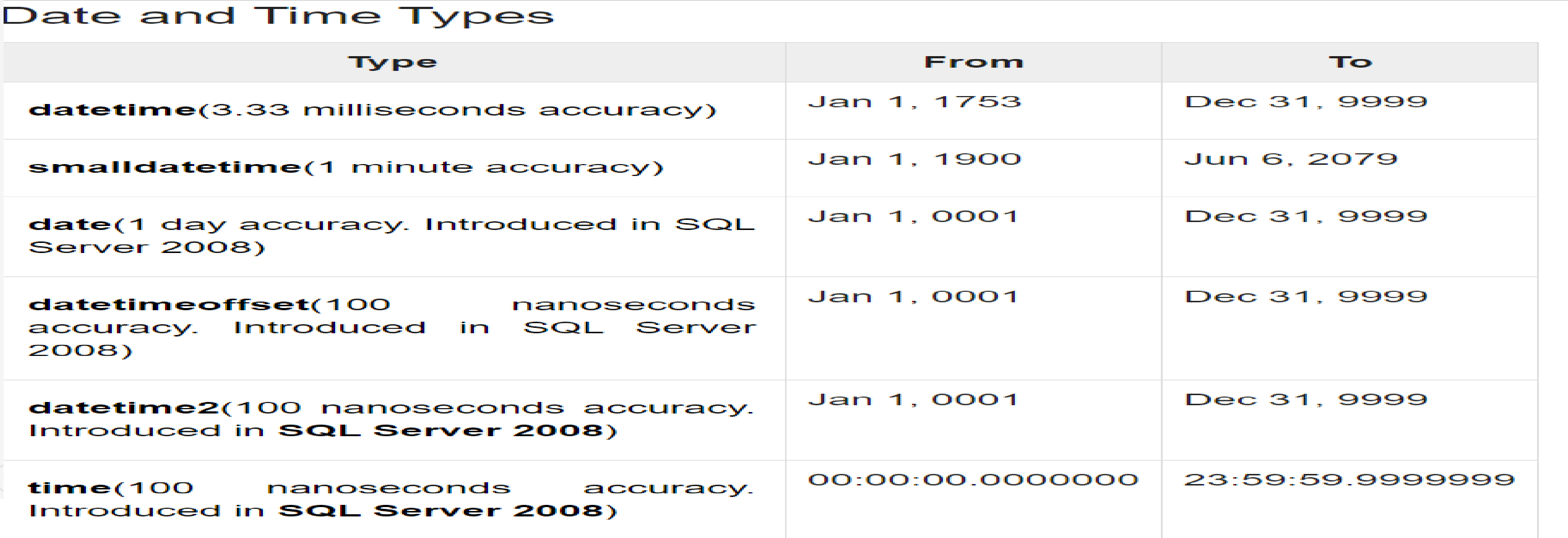
Type	From	To
bigint	-9,223,372,036,854,775,808	9,223,372,036,854,775,807
int	-2,147,483,648	2,147,483,647
smallint	-32,768	32,767
tinyint	0	255
bit	0	1
decimal	-10^38 +1	10^38 –1
numeric	-10^38 +1	10^38 –1
money	-922,337,203,685,477.5808	+922,337,203,685,477.5807
smallmoney	-214,748.3648	+214,748.3647

T-SQL Data types



Approximate Numeric Types

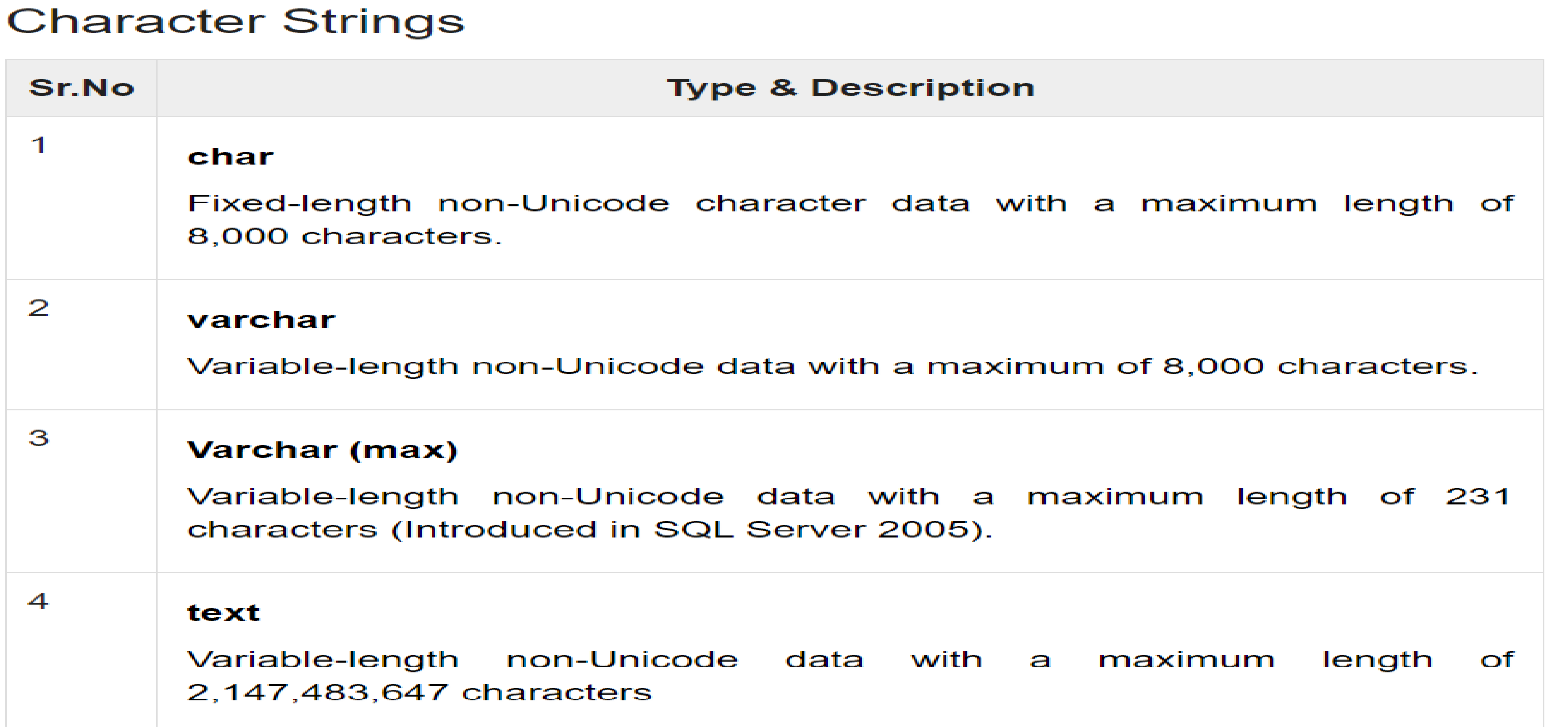
Type	From	To
Float	-1.79E + 308	1.79E + 308
Real	-3.40E + 38	3.40E + 38



Date and Time Types

Type	From	To
datetime(3.33 milliseconds accuracy)	Jan 1, 1753	Dec 31, 9999
smalldatetime(1 minute accuracy)	Jan 1, 1900	Jun 6, 2079
date(1 day accuracy. Introduced in SQL Server 2008)	Jan 1, 0001	Dec 31, 9999
datetimeoffset(100 nanoseconds accuracy. Introduced in SQL Server 2008)	Jan 1, 0001	Dec 31, 9999
datetime2(100 nanoseconds accuracy. Introduced in **SQL Server 2008**)	Jan 1, 0001	Dec 31, 9999
time(100 nanoseconds accuracy. Introduced in **SQL Server 2008**)	00:00:00.0000000	23:59:59.9999999

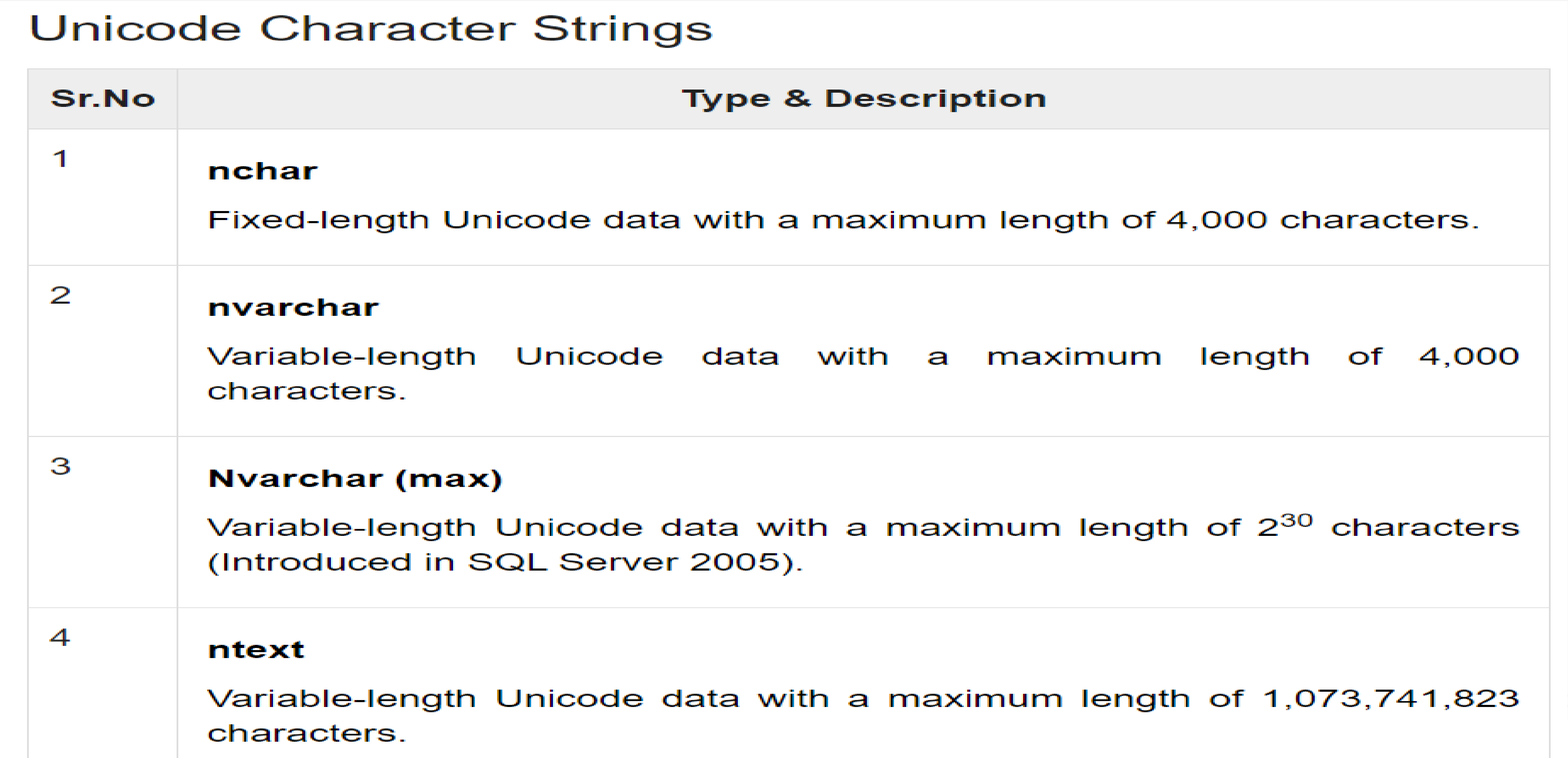
T-SQL Data types



Character Strings

Sr.No	Type & Description
1	**char** Fixed-length non-Unicode character data with a maximum length of 8,000 characters.
2	**varchar** Variable-length non-Unicode data with a maximum of 8,000 characters.
3	**Varchar (max)** Variable-length non-Unicode data with a maximum length of 231 characters (Introduced in SQL Server 2005).
4	**text** Variable-length non-Unicode data with a maximum length of 2,147,483,647 characters

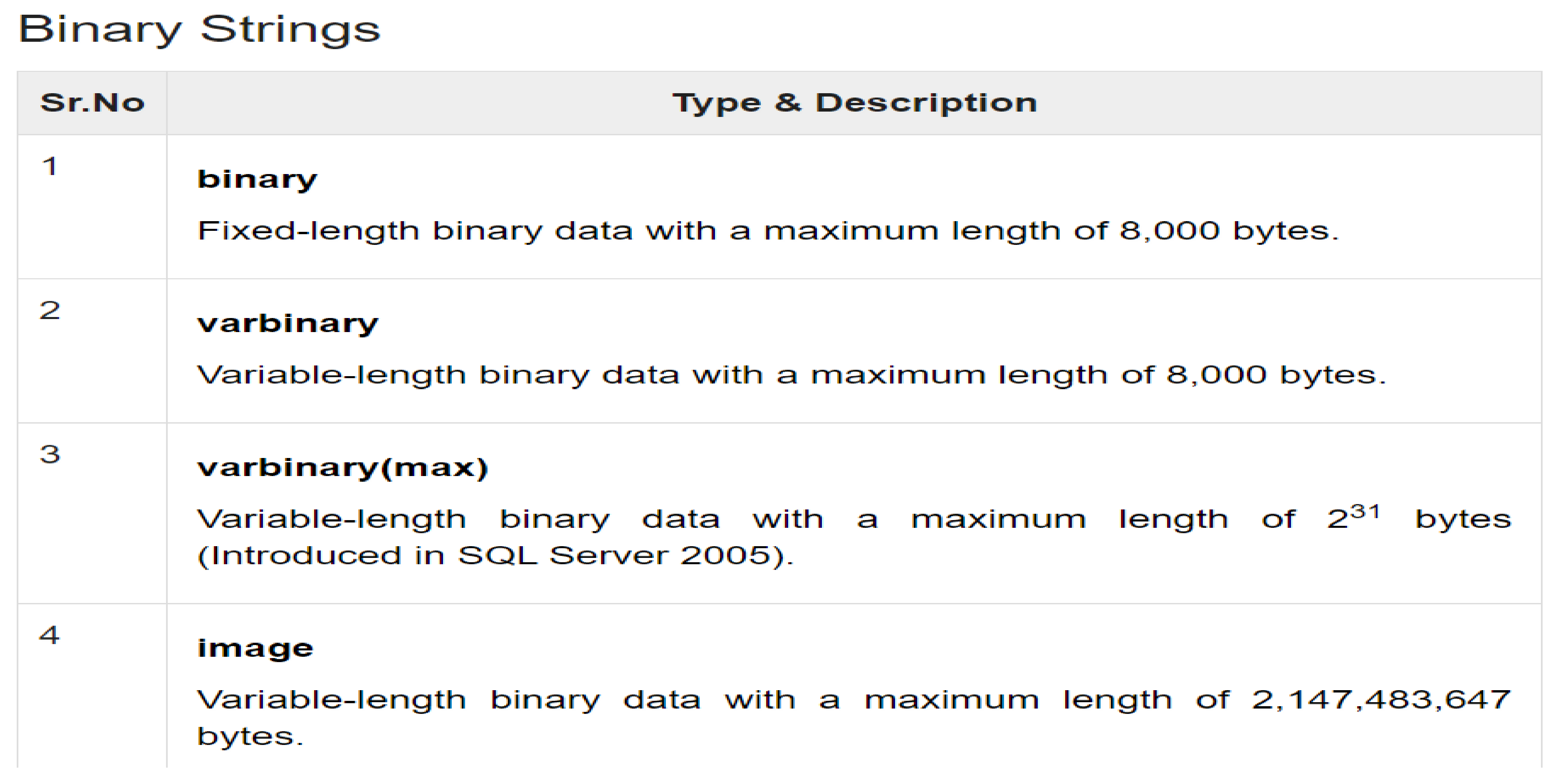
T-SQL Data types



Unicode Character Strings

Sr.No	Type & Description
1	**nchar** Fixed-length Unicode data with a maximum length of 4,000 characters.
2	**nvarchar** Variable-length Unicode data with a maximum length of 4,000 characters.
3	**Nvarchar (max)** Variable-length Unicode data with a maximum length of 2^{30} characters (Introduced in SQL Server 2005).
4	**ntext** Variable-length Unicode data with a maximum length of 1,073,741,823 characters.

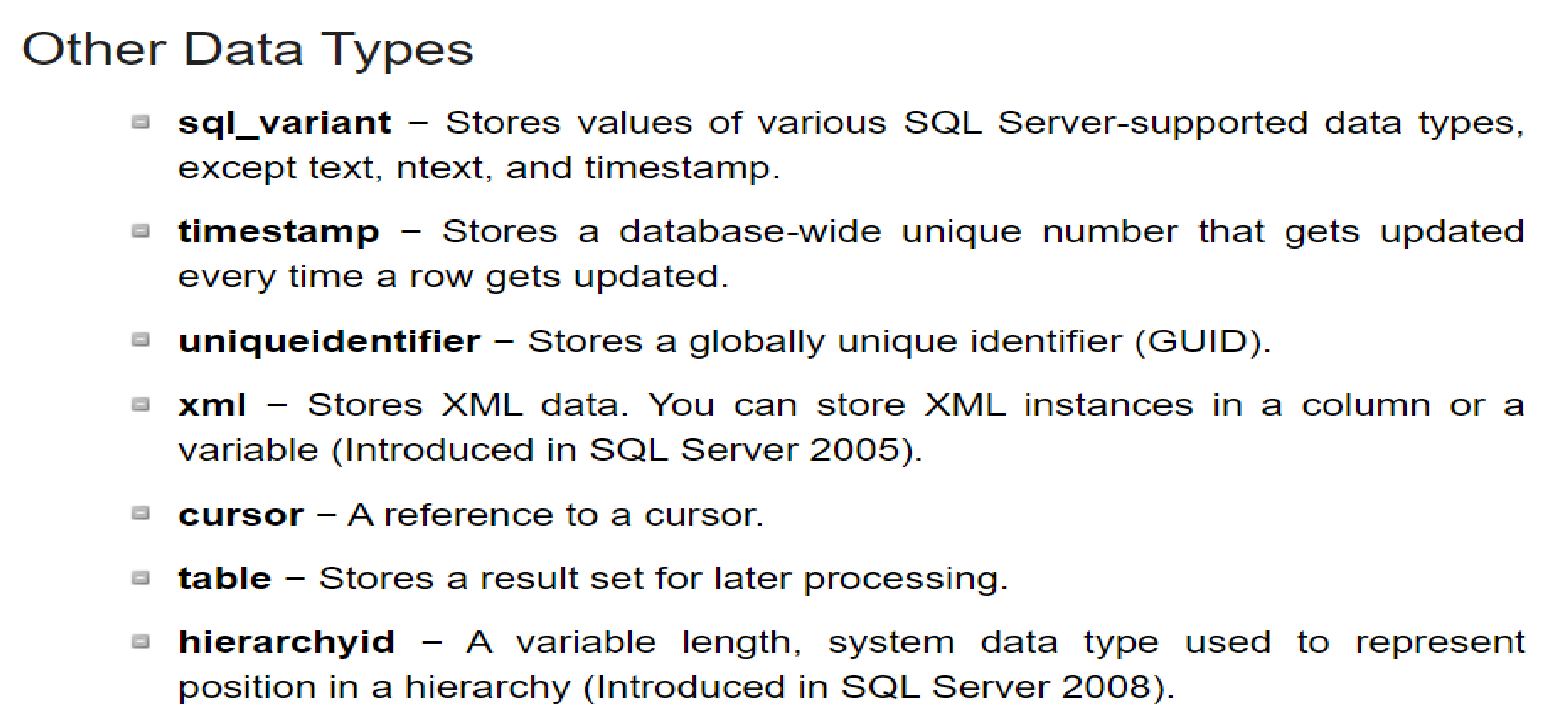
T-SQL Data types



Binary Strings

Sr.No	Type & Description
1	**binary** Fixed-length binary data with a maximum length of 8,000 bytes.
2	**varbinary** Variable-length binary data with a maximum length of 8,000 bytes.
3	**varbinary(max)** Variable-length binary data with a maximum length of 2^{31} bytes (Introduced in SQL Server 2005).
4	**image** Variable-length binary data with a maximum length of 2,147,483,647 bytes.

T-SQL Data types



Other Data Types

- **sql_variant** – Stores values of various SQL Server-supported data types, except text, ntext, and timestamp.
- **timestamp** – Stores a database-wide unique number that gets updated every time a row gets updated.
- **uniqueidentifier** – Stores a globally unique identifier (GUID).
- **xml** – Stores XML data. You can store XML instances in a column or a variable (Introduced in SQL Server 2005).
- **cursor** – A reference to a cursor.
- **table** – Stores a result set for later processing.
- **hierarchyid** – A variable length, system data type used to represent position in a hierarchy (Introduced in SQL Server 2008).

T-SQL - Stored Procedures - Tutorialspoint

T-SQL

Create Table

```
USE tempDB;
GO
    CREATE TABLE Students
    (ID int,
    Name varchar(255),
    Birthday date,
    City varchar(500));
GO
```

Insert into Table

```
USE tempDB;
GO
       insert into
       Students(id,Name,Birthday,City)
       values
       (1,'Emma','1994-01-01','New York’),
       (2,'Daniel','1995-06-08','Chicago’),
       (3,'Joseph','1996-10-11','Dallas’),
       (4,'Jennifer','1997-03-15','Los Angeles’),
       (5,'Debra','1998-09-05','Dallas') ;
GO
```

T-SQL

Create Temporary Table

```
USE tempDB;
GO
    CREATE TABLE # Course
    (ID int,
    Name varchar(255);
GO
```

Insert into Temporary Table

```
USE tempDB;
GO
       insert into #Course(ID, Name)
       values
       (1,’SQL’),
       (2,’Transact-SQL’);
GO
```

TempDB

What is the purpose of SQL Server TempDB?

- SQL Server comes installed **with four system databases** by default. They are master, model, msdb, and TempDB.
- **TempDB** is a database that has many functions within SQL Server, but it is rarely called explicitly.
- It has so many functions that it is often one of the busiest, if not the busiest database on most SQL Server instance.
- One of the functions of TempDB is to act something like a page or swap file would at the operating system level.
- If a SQL Server operation is too large to be completed in memory or if the initial memory grant for a query is too small, the operation can be moved to disk in TempDB.

TempDB (Contd.)

What is the purpose of SQL Server TempDB?

- Another function of TempDB is to store temporary tables. Anyone who has created a temporary table in T-SQL using a pound or hash prefix (#) or the double pound/hash prefix (##) has created an object in TempDB as this is where those are stored.
- When a trigger is executing the inserted and deleted virtual tables are stored in TempDB.
- SQL Server cannot run (except in very extreme recovery scenarios and for very short periods) without a TempDB.
- TempDB is a mostly blank database and there is nothing particularly unique about any edition or patch level of SQL Server.
- TempDB cannot be dropped, detached, taken offline, or renamed. Attempting any of these operations will return an error. There is no reason that this should ever be attempted as this is a critical system database.

[SQL Server TempDB Tutorial (mssqltips.com)]()

Transaction Basics

- **Concepts**
  - Atomicity
  - Consistency
  - Isolation
  - Durability
- **T-SQL Commands**
  - BEGIN TRANSACTION – Begin transaction
  - COMMIT TRANSACTION – End transaction
  - ROLLBACK TRANSACTION – Cancel transaction

Transaction Basics… (contd.)

- Transferring Rs. 1000 from Account 1234 to Account 2345

```
Begin Transaction

        update Account
        set balance = balance - 1000
        where account_no = ‘1234’

        update Account
        set balance = balance + 1000
        where account_no = ‘2345’

Commit transaction
```

Programming in T-SQL

- Similar to a programming language, certain extensions have been made in SQL to program simple server-side logic.
- Some of the statements include:
  - Variables
  - Selection conditions

    IF (…)… ELSE …
  - Looping

    WHILE (…)

T-SQL: Variables

- A Transact-SQL local variable is an object that can hold a single data value of a specific type.
- Variables in batches and scripts are typically used:
  - As a counter either to count the number of times a loop is performed or to control how many times the loop is performed
  - To hold a data value to be tested by a control-of-flow statement
  - To save a data value to be returned by a stored procedure return code.

T-SQL: Variables… (contd.)

- The DECLARE statement initializes a Transact-SQL variable by:
  - Assigning a name. The name must have a single @ as the first character.
  - Assigning a system-supplied or user-defined data type and a length. For numeric variables, a precision and scale are also assigned.
  - Setting the value to NULL.

- For example…
  - DECLARE @DeptName VARCHAR(20)

T-SQL: Variables… (contd.)

- When a variable is first declared, its value is set to NULL.

- To assign a value to a variable, use the SET statement.

- For example,
  - SET @DeptName = ‘Sales’

T-SQL: Variables… (contd.)

- Now we can use the variable in our script…
- For example...

```
SELECT building
 FROM Dept
 WHERE deptname = @DeptName
```

T-SQL: Variables… (contd.)

- A variable can also have a value assigned by being referenced in a **select** list.

-
  ```
  DECLARE @EmpIDVariable INT
      SELECT @EmpIDVariable =
              MAX(EmployeeID)
      FROM Employees
  ```

T-SQL: IF statement

- Imposes conditions on the execution of a Transact-SQL statement.
- IF (SELECT AVG(price) FROM titles) < $15

```
BEGIN
    PRINT ‘Inside the IF statement’
    PRINT ‘ Multiple statements have to be encoded in BEGIN – END’
END
ELSE
    PRINT 'Average title price is more than $15.'
```

T-SQL: WHILE statement

- Sets a condition for the repeated execution of an SQL statement or statement block.
- The statements are executed repeatedly as long as the specified condition is true.
- The execution of statements in the WHILE loop can be controlled from inside the loop with the BREAK and CONTINUE keywords.

T-SQL: WHILE statement… (contd.)

- BREAK

  Causes an exit from the innermost WHILE loop. Any statements appearing after the END keyword, marking the end of the loop, are executed.

- CONTINUE

  Causes the WHILE loop to restart, ignoring any statements after the CONTINUE keyword.

T-SQL: WHILE statement… (contd.)

❖ Example…

```
WHILE (SELECT AVG(price) FROM titles) < $30
BEGIN
        UPDATE titles
        SET price = price * 2 SELECT MAX(price) FROM titles

        IF (SELECT MAX(price) FROM titles) > $50
                BREAK
        ELSE
                CONTINUE
END
PRINT 'Too much for the market to bear'
```

End of the Lecture – 6
Thank You