DBMS LAB # 1

MS SQL Server Installation & Operations

# SQL Server 2019 Installation and Commands Introduction

# Server Editions of SQL Server 2019

|  |  |
| --- | --- |
| Enterprise (64-bit and 32-bit) | The premium offering, SQL Server 2019 Enterprise edition delivers comprehensive high-end datacenter capabilities with blazing-fast performance, unlimited virtualization, and end-to-end business intelligence - enabling high service levels for mission-critical workloads and end user access to data insights. |
| Business Intelligence (64-bit and 32-bit) | SQL Server 2019 Business Intelligence edition delivers comprehensive platform empowering organizations to build and deploy secure, scalable and manageable BI solutions. It offers exciting functionality such as browser based data exploration and visualization; powerful data mash-up capabilities, and enhanced integration management. |
| Standard (64-bit and 32-bit) | SQL Server 2019 Standard edition delivers basic data management and business intelligence database for departments and small organizations to run their applications and supports common development tools for on-premise and cloud - enabling effective database management with minimal IT resources. |

## Breadth Editions of SQL Server 2019

Breadth editions of SQL Server are engineered for specific customer scenarios and are offered FREE or at a very nominal cost. The following table describes the breadth editions of SQL Server.

|  |  |
| --- | --- |
| Developer (64-bit and 32-bit) | SQL Server 2019 Developer edition lets developers build any kind of application on top of SQL Server. It includes all the functionality of Enterprise edition, but is licensed for use as a development and test system, not as a production server. SQL Server Developer is an ideal choice for people who build and test applications. |
| Express (64-bit and 32-bit) editions | SQL Server 2019 Express edition is the entry-level, free database and is ideal for learning and building desktop and small server data-driven applications. It is the best choice for independent software vendors, developers, and hobbyists building client applications. If you need more advanced database features, SQL Server Express can be seamlessly upgraded to other higher end versions of SQL Server. SQL Server Express LocalDB, a lightweight version of Express that has all of its programmability features, yet runs in user mode and has a fast, zero-configuration installation and a short list of prerequisites. |

# SQL Server Components

| **Server components** | **Description** |
| --- | --- |
| SQL Server Database Engine | SQL Server Database Engine includes the Database Engine, the core service for storing, processing, and securing data, replication, full-text search, tools for managing relational and XML data, and the Data Quality Services (DQS) server. |
| Analysis Services | Analysis Services includes the tools for creating and managing online analytical processing (OLAP) and data mining applications. |
| Reporting Services | Reporting Services includes server and client components for creating, managing, and deploying tabular, matrix, graphical, and free-form reports. Reporting Services is also an extensible platform that you can use to develop report applications. |
| Integration Services | Integration Services is a set of graphical tools and programmable objects for moving, copying, and transforming data. It also includes the Data Quality Services (DQS) component for Integration Services. |
| Master Data Services | Master Data Services (MDS) is the SQL Server solution for master data management. MDS can be configured to manage any domain (products, customers, accounts) and includes hierarchies, granular security, transactions, data versioning, and business rules, as well as an Add-in for Excel that can be used to manage data. |

| **Management tools** | **Description** |
| --- | --- |
| SQL Server Management Studio | SQL Server Management Studio is an integrated environment to access, configure, manage, administer, and develop components of SQL Server. Management Studio lets developers and administrators of all skill levels use SQL Server. |
| SQL Server Configuration Manager | SQL Server Configuration Manager provides basic configuration management for SQL Server services, server protocols, client protocols, and client aliases. |
| SQL Server Profiler | SQL Server Profiler provides a graphical user interface to monitor an instance of the Database Engine or Analysis Services. |
| Database Engine Tuning Advisor | Database Engine Tuning Advisor helps create optimal sets of indexes, indexed views, and partitions. |
| Data Quality Client | Provides a highly simple and intuitive graphical user interface to connect to the DQS server, and perform data cleansing operations. It also allows you to centrally monitor various activities performed during the data cleansing operation. |
| SQL Server Data Tools | SQL Server Data Tools provides an IDE for building solutions for the Business Intelligence components: Analysis Services, Reporting Services, and Integration Services.  (Formerly called Business Intelligence Development Studio).  SQL Server Data Tools also includes "Database Projects", which provides an integrated environment for database developers to carry out all their database design work for any SQL Server platform (both on and off premise) within Visual Studio. Database developers can use the enhanced Server Explorer in Visual Studio to easily create or edit database objects and data, or execute queries. |
| Connectivity Components | Installs components for communication between clients and servers, and network libraries for DB-Library, ODBC, and OLE DB. |

## Installation of SQL Server:

Installing SQL Server 2019 Express Edition

To install Microsoft SQL Server 2019 Express Edition, follow these steps:

Run the setup file through the administrative account. The Microsoft SQL Server 2019 Installation Center will begin.

1. Select the “New Installation” Feature.
2. In this Feature Selection Wizard Select all of the features and click the Next button.
3. In this Instance Configuration select the **Default Instance** option and click Next button.
4. Click **Next**.
5. In this **Database Engine Configuration** select the **Windows Authentication Mode** and click Next button.
6. Installation will progress and setup will complete installation automatically. Just wait and watch.
7. From the Setup Complete window, click **Close**.

Now go to the Start menu and Search SQL Server Management Studio. A new window will be opened. Connect to the server.

Now place the following sample code in the query window, run it and see the output:

SELECT getdate(); -- Selects the current (server) date and time.

On the left side you will see databases named master etc.

On clicking any of the database, you’ll see default tables in that database. You can also right click on any of the table and select ‘return all rows’ to see the entire values in the table.

But you have to create your own database with your own name.

Run the following query by pressing F5 key:

CREATE DATABASE WXYZ; -- Creates a database named WXYZ;

Remember…!! SQL is not case sensitive.

Are you able to see the database created…???

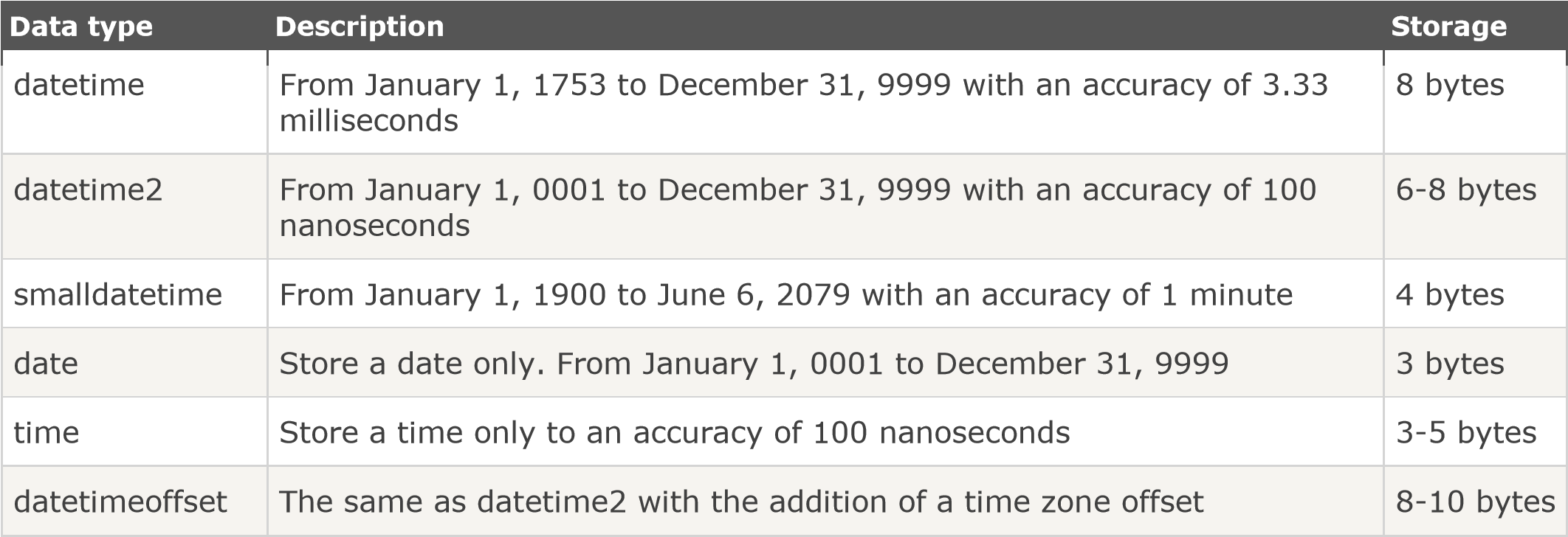
If no, then refresh the services again…!!

SQL Server Data Types 

**String types:**

|  |  |  |
| --- | --- | --- |
| **Data type** | **Description** | **Storage** |
| tinyint | Allows whole numbers from 0 to 255 | 1 byte |
| smallint | Allows whole numbers between -32,768 and 32,767 | 2 bytes |
| int | Allows whole numbers between -2,147,483,648 and 2,147,483,647 | 4 bytes |
| bigint | Allows whole numbers between -9,223,372,036,854,775,808 and 9,223,372,036,854,775,807 | 8 bytes |
| decimal(p,s) | Fixed precision and scale numbers. | 5-17 bytes |

|  |  |
| --- | --- |
| **Data type** | **Description** |
| sql\_variant | Stores up to 8,000 bytes of data of various data types, except text, ntext, and timestamp |
| uniqueidentifier | Stores a globally unique identifier (GUID) |
| xml | Stores XML formatted data. Maximum 2GB |
| cursor | Stores a reference to a cursor used for database operations |
| table | Stores a result-set for later processing |



1. **Commands Introduction**

SQL can be divided into four parts: The Data Manipulation Language (DML), the Data Definition Language (DDL), Data Control Language (DCL) and Transaction control language (TCL).

The DDL part of SQL permits database tables to be created or deleted. It also define indexes (keys), specify links between tables, and impose constraints between tables.

The most important DDL statements in SQL are:

* **CREATE DATABASE** - creates a new database
* **ALTER DATABASE** - modifies a database
* **CREATE TABLE** - creates a new table
* **ALTER TABLE** - modifies a table
* **DROP TABLE** - deletes a table
* **CREATE INDEX** - creates an index (search key)
* **DROP INDEX** - deletes an index

Examples of DML statements are:

* **SELECT** - extracts data from a database
* **UPDATE** - updates data in a database
* **DELETE** - deletes data from a database
* **INSERT INTO** - inserts new data into a database

Now create the table in the above created database using the CREATE TABLE command:

Syntax:

|  |  |
| --- | --- |
| CREATE TABLE | *table\_name* |

( *column\_name1 data\_type*(*size*), *column\_name2 data\_type*(*size*), *column\_name3 data\_type*(*size*),

....

); --// CREATE TABLE is the keyword.

Suppose if you want to create the following table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Reg\_No** | **Courses** | **Course\_Code** | **Offered\_By** |

Run the query and see the results.

CREATE TABLE Student

(

Name varchar(255),

Reg\_No varchar(255),

Courses varchar(255),

Course\_Code int,

Offered\_by varchar(255)

)

;

Was the table created named STUDENT….???

In order to verify the results run the following query:

SELECT \*

FROM Student;

Was the table displayed…???