## TempDB in SQL Server

* It stores temporary objects, like temporary tables, temporary Stored Procedures and temporary tables to store sorting and so on.
* The dbid of a temp database is 2.
* The recovery model of a temp database is SIMPLE.
* We can't take a backup of a tempdb.
* tempdev and templog are the logical file names of tempdb.
* tempdb.mdf (data file) and templog.ldf are the physical files of a tempdb.
* Same query can be use to see the physical file location of tempdb:

1. SELECT name, physical\_name FROM sys.database\_files;

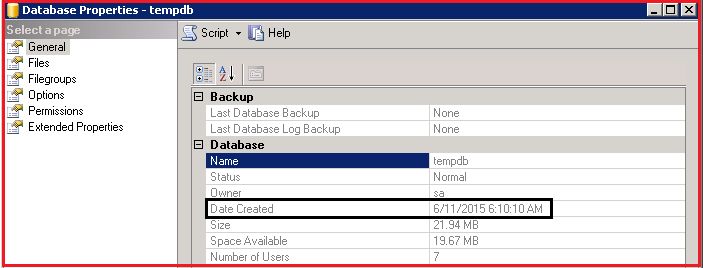
TempDB Table in SQL Server

### Why we can't take a backup of a temp database?

**Answer:** Temp databases, as the name says, are used to do temporary operations, such as tables, Stored Procedures and cursors. Once the operation is over it will be cleared and is minimally logged. A TempDB is recreated everytime SQL is started, so it is always has a clean copy of the database hence backup and restore operations are not allowed for a TempDB.

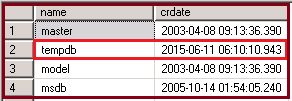
### How you will check to determine if the SQL Server is restarted?

**Answer:** Check the creation date of the tempdb, if it is new it means SQL Server is started.



We can execute the following query also to check the tempdb creation date:

1. select name, crdate from sys.sysdatabases;



**Two Type Temp Table : -**

* Local Temporary Tables
* Global Temporary Tables

**Local Temp Table** [**#**](https://sodocumentation.net/sql-server/topic/5328/use-of-temp-table#local-temp-table)

Will be available till the current connection persists for the user.

Automatically deleted when the user disconnects.

The name should start with # (#temp)

CREATE TABLE #LocalTempTable(

StudentID int,

StudentName varchar(50),

StudentAddress varchar(150))

insert into #LocalTempTable values ( 1, 'Ram','India');

select \* from #LocalTempTable

**Global Temp Table**[**#**](https://sodocumentation.net/sql-server/topic/5328/use-of-temp-table#global-temp-table)

Will start with ## (##temp).

Will be deleted only if user disconnects all connections.

It behaves like a permanent table.

CREATE TABLE ##NewGlobalTempTable(

StudentID int,

StudentName varchar(50),

StudentAddress varchar(150))

Insert Into ##NewGlobalTempTable values ( 1,'Ram','India');

Select \* from ##NewGlobalTempTable

**Dropping temp tables**[**#**](https://sodocumentation.net/sql-server/topic/5328/use-of-temp-table#dropping-temp-tables)

Temp tables must have unique IDs (within the session, for local temp tables, or within the server, for global temp tables). Trying to create a table using a name that already exists will return the following error:

There is already an object named '#tempTable' in the database.

If your query produces temp tables, and you want to run it more than once, you will need to drop the tables before trying to generate them again. The basic syntax for this is:

drop table #tempTable

Trying to execute this syntax before the table exists (e.g. on the first run of your syntax) will cause another error:

Cannot drop the table '#tempTable', because it does not exist or you do not have permission.

To avoid this, you can check to see if the table already exists before dropping it, like so:

IF OBJECT\_ID ('tempdb..#tempTable', 'U') is not null DROP TABLE #tempTable

**Advantages**

Temporary tables behave just like normal ones; you can sort, filter and join them as if they were permanent tables.

Because SQL Server has less logging and locking overheads for temporary tables. they execute more quickly.

You can create a temporary table and insert, delete and update its records without worrying about whether you have sufficient rights to change data in permanent tables, or whether you might be accidentally doing so.