SQL Server System Databases

The value of SQL Server system databases cannot be underestimated due to the significant amount of information residing in these databases.

So basically understanding of the SYSTEM DATABASE's is very important as a Database Administrator.

System Databases

SQL Server includes the following system databases:

1) Master Database

Master is the most important database in SQL Server.

The **master** database records all of the system level information for a SQL Server system. The master database is at the heart of SQL Server, and if it becomes corrupted, there is a very good chance that SQL Server will not work correctly.

The master database contains the following crucial information:

- All logins, or roles, to which the user IDs belong.
- Every system configuration setting (for example, data sorting information, security implementation, default language).
- The names of and information about the databases within the server.
- The location of databases.

- How SQL Server is initialized.
- **Specific system tables holding the following information (this list is not**

REPORT THIS AD

exhaustive):.

- How the cache is used.
- Which character sets are available.
- A list of the available languages.
- System error and warning messages.
- Special SQL Server objects called *assemblies* (tables within every database that deal with SQL Server objects and therefore are not specific to the master database)
- The master database is the security guard of SQL Server, and it uses the preceding information to ensure that everything is kept in check.
- Master database is used to store information of all the databases on the SQL server.
- The server cannot start if the database is not configured properly.

Note

It is crucial that you regularly back up the master database. Ensure that doing so is part of your backup strategy.

Restrictions

The following operations cannot be performed on the **master** database:

- Adding files or filegroups.
- Changing collation. The default collation is the server collation.
- Changing the database owner. **master** is owned by **sa**.
- Creating a full-text catalog or full-text index.

- Creating triggers on system tables in the database.
- Dropping the database.
- Dropping the **guest** user from the database.
- Enabling change data capture.
- Participating in database mirroring.
- Removing the primary filegroup, primary data file, or log file.
- Renaming the database or primary filegroup.
- Setting the database to OFFLINE.
- Setting the database or primary filegroup to READ_ONLY.

Recommendations

When you work with the **master** database, consider the following recommendations:

- Always have a current backup of the **master** database available.
- Back up the **master** database as soon as possible after the following operations:
  - Creating, modifying, or dropping any database
  - Changing server or database configuration values
  - Modifying or adding logon accounts
- Do not create user objects in **master**. If you do, **master** must be backed up more frequently.
- Do not set the TRUSTWORTHY option to ON for the **master** database.

2) MODEL Database

- The **model** database is used as the template for all databases created on a system. When a CREATE DATABASE statement is issued, the first part of the database is

created by copying in the contents of the **model** database, then the remainder of t he new database is filled with empty pages.

- Because **tempdb** is created every time SQL Server is started, the **model** database must always exist on a SQL Server system.
- The entire contents of the **model** database, including database options, are copied to the new database. Some of the settings of **model** are also used for creating a new **tempdb** during start up, so the **model** database must always exist on a SQL Server system.

Additional Information:

- When a CREATE DATABASE statement is issued, the first part of the database is created by copying in the contents of the **model**database. The rest of the new database is then filled with empty pages.
- User defined tables, stored procedures, user defined data types, etc can be created i n the Model database and will exist in all future user defined databases.
- The database configurations such as the recovery model for the Model database are applied to future user defined databases
- If you modify the **model** database, all databases created afterward will inherit those changes. For example, you could set permissions or database options, or add objects such as tables, functions, or stored procedures. File properties of the **model** database are an exception, and are ignored except the initial size of the data file. The default initial size of the model database data and log file is 8 MB.
- Understand which database options can be changed.

Restrictions

The following operations cannot be performed on the **model** database:

- Adding files or filegroups.

- Changing collation. The default collation is the server collation.
- Changing the database owner. **model** is owned by **sa**.
- Dropping the database.
- Dropping the **guest** user from the database.
- Enabling change data capture.
- Participating in database mirroring.
- Removing the primary filegroup, primary data file, or log file.
- Renaming the database or primary filegroup.
- Setting the database to OFFLINE.
- Setting the primary filegroup to READ_ONLY.
- Creating procedures, views, or triggers using the WITH ENCRYPTION option. The encryption key is tied to the database in which the object is created. Encrypted objects created in the **model** database can only be used in **model**.

3) MSDB Database

- It stores information related to database backups, DTS packages, Replication, SQL Server Agent information, SQL Server jobs.
- The **msdb** database is used by SQL Server Agent for scheduling alerts and jobs and by other features such as SQL Server Management Studio, Service Broker and Database Mail.
- SQL Server automatically maintains a complete online backup-and-restore history within tables in **msdb**. This information includes the name of the party that performed the backup, the time of the backup, and the devices or files where the backup is stored. SQL Server Management Studio uses this information to propose a plan for restoring a database and applying any transaction log backups. Backup events for all databases are recorded even if they were created with custom

applications or third-party tools. For example, if you use a Microsoft Visual Basic application that calls SQL Server Management Objects (SMO) objects to perform backup operations, the event is logged in the **msdb** system tables, the Microsoft Windows application log, and the SQL Server error log.

- By default, **msdb** uses the simple recovery model.

Important

After any operation that updates **msdb**, such as backing up or restoring any database, we recommend that you back up **msdb**. For more information, see Back Up and Restore of System Databases (SQL Server).

Restrictions

The following operations cannot be performed on the **msdb** database:

- Changing collation. The default collation is the server collation.
- Dropping the database.
- Dropping the **guest** user from the database.
- Enabling change data capture.
- Participating in database mirroring.
- Removing the primary filegroup, primary data file, or log file.
- Renaming the database or primary filegroup.
- Setting the database to OFFLINE.
- Setting the primary filegroup to READ_ONLY.

4) TEMPDB Database

tempdb is a global resource; the temporary tables and stored procedures for all users c onnected to the system are stored there.

Tempdb is used to hold:

- Temporary **user objects** that are explicitly created, such as: global or local temporary tables and indexes, temporary stored procedures, table variables, Tables returned in table-valued functions, or cursors.
- **Internal objects** that are created by the database engine. These include:
  - Work tables to store intermediate results for spools, cursors, sorts, and temporary large object (LOB) storage.
  - Work files for hash join or hash aggregate operations.
  - Intermediate sort results for operations such as creating or rebuilding indexes (if SORT_IN_TEMPDB is specified), or certain GROUP BY, ORDER BY, or UNION queries.
- Operations within **tempdb** are minimally logged so that transactions can be rolled back.
- **tempdb** is re-created every time SQL Server is started so that the system always starts with a clean copy of the database.
- Backup and restore operations are not allowed on **tempdb**.
- Because temporary tables and stored procedures are dropped automatically on dis connect, and no connections are active when the system is shut down, there is nev er anything in **tempdb** to be saved from one session of SQL Server to another.
- By default, **tempdb** autogrows as needed while SQL Server is running. Unlike oth er databases, however, it is reset to its initial size each time the database engine is started.
- If the size defined for **tempdb** is small, part of your system processing load may b e taken up with autogrowing **tempdb** to the size needed to support your workloa

d each time to restart SQL Server. You can avoid this overhead by using ALTER DATABASE to increase the size of **tempdb**.

Restrictions

The following operations cannot be performed on the **tempdb** database:

- Adding filegroups.
- Backing up or restoring the database.
- Changing collation. The default collation is the server collation.
- Changing the database owner. **tempdb** is owned by **sa**.
- Creating a database snapshot.
- Dropping the database.
- Dropping the **guest** user from the database.
- Enabling change data capture.
- Participating in database mirroring.
- Removing the primary filegroup, primary data file, or log file.
- Renaming the database or primary filegroup.
- Running DBCC CHECKALLOC.
- Running DBCC CHECKCATALOG.
- Setting the database to OFFLINE.
- Setting the database or primary filegroup to READ_ONLY.

5) RESOURCE Database

- The Resource database is a read-only database that contains all the system objects that are included with SQL Server. The ID of the Resource database is always 32767.
- SQL Server system objects, such as sys.objects(system Views), are physically persisted in the Resource database, but they logically appear in the sys schema of every database.
- The Resource database does not contain user data or user metadata.
- The physical file names of the Resource database are: mssqlsystemresource.mdf and mssqlsystemresource.ldf.
- These files are located in : Program FilesMicrosoft SQL ServerMSSQL11.MSSQLBinn.

Physical Properties of Resource

- The physical file names of the Resource database are mssqlsystemresource.mdf and mssqlsystemresource.ldf.
- These files are located in <*drive*>:\Program Files\Microsoft SQL Server\MSSQL<version>.<*instance_name*>\MSSQL\Binn\ and should not be moved.
- Each instance of SQL Server has one and only one associated mssqlsystemresource.mdf file, and instances do not share this file.

REPORT THIS AD

The Resource database makes upgrading to a new version of SQL Server an easier and faster procedure. In earlier versions of SQL Server, upgrading required dropping and creating system objects. Because the Resource database file contains all system objects, an upgrade is now accomplished simply by copying the single Resource database file to the local server.

Backing Up and Restoring the Resource Database

SQL Server cannot back up the Resource database. You can perform your own file-based or a disk-based backup by treating the mssqlsystemresource.mdf file as if it were a binary (.EXE) file, rather than a database file, but you cannot use SQL Server to restore your backups. Restoring a backup copy of mssqlsystemresource.mdf can only be done manually, and you must be careful not to overwrite the current Resource database with an out-of-date or potentially insecure version.