Microsoft® SQL Server™ 2000

Backup/Restore Architecture

The backup and restore components of Microsoft® SQL Server™ 2000 allow you to create a copy of a database. This copy is stored in a location protected from the potential failures of the server running the instance of SQL Server. If the server running the instance of SQL Server fails, or if the database is somehow damaged, the backup copy can be used to re-create, or restore, the database.

SQL Server 2000 provides these sophisticated backup and restore capabilities:

- Options for how a database is backed up and restored:
  - A full database backup is a full copy of the database.
  - A transaction log backup copies only the transaction log.
  - A differential backup copies only the database pages modified after the last full database backup.
  - A file or filegroup restore allows the recovery of just the portion of a database that was on the failed disk.

  These options allow backup and restore processes to be tailored to how critical the data in the database is. Noncritical databases that can be easily re-created from some other source may have no backups, other databases may have simple backups that can re-create the database to the night before a failure, and critical databases may have sophisticated backups that will restore the database right up to the point of failure.

- Control with the BACKUP and RESTORE statements.

  Users can execute the BACKUP and RESTORE statements directly from applications, Transact-SQL scripts, stored procedures, and triggers. It is more common, however, to use SQL Server Enterprise Manager to define a backup schedule, and then let SQL Server Agent run the backups automatically according to the schedule. The Database Maintenance Plan Wizard can be used to define and schedule a full set of backups for each database. This fully automates the backup process, requiring minimal or no operator action.

- Maintenance of a set of backup history tables in the **msdb** database.

  The backup history tables record the backups for each database. If a database has to be restored, the **Restore Database** dialog box in SQL Server Enterprise Manager presents the user with a list of all the backups available for the database. The **Restore Database** dialog box also has logic to display which set of the backups in the history can be used to restore the database in the shortest possible time. When the dialog box is displayed, the backups needed to restore the database are checked. If a user knows that one of the backups is not available (for example, if a tape cartridge was damaged or lost), the user can deselect that backup, and SQL Server Enterprise Manager calculates a new restore process. When the user agrees with the restore process, SQL Server Enterprise Manager restores the database, prompting for tapes as needed.

- Backups that can be performed while the database is in use, allowing backups to be made of systems that must run continuously.

  The backup processing and internal data structures of SQL Server 2000 are structured so that backups maximize their rate of data transfer with minimal effect on transaction throughput.

- Fast data transfer rates for backup and restore operations, making SQL Server 2000 capable of supporting very large databases (VLDB).

  The data structures in SQL Server 2000 databases and the backup and restore algorithms support high data transfer rates for backup and restore operations. SQL Server backup and restore operations can also run in parallel against multiple backup files or tape drives, which further increases the backup and restore data transfer rates.

- RESTORE statement re-creates the database automatically if necessary.

  This eliminates the need to execute a separate CREATE DATABASE or CREATE DATABASE FOR LOAD statement if the database does not exist at the time the RESTORE statement is executed.

- Interrupted backup and restore operations started near the point of the interruption when restarted.

- Verification of a SQL Server 2000 backup before an attempt to restore the database. This includes verifying that the collation of the database is supported by the instance of SQL Server.

Backup and restore processes should be planned together. The administrators must first determine the criticality of the data in the database. They must determine if it is acceptable to just restore the database to a point such as the night before the failure, or if the database must be restored to a point as close as possible to the time of failure. They must also determine how long the database can be unavailable, whether it must be brought back online as quickly as possible, or if it does not need to be restored immediately.

After the restore requirements are determined, the administrators can then plan a backup process that maintains a set of backups that will meet the restore requirements. The administrators can choose the backup processes that can be performed with the minimum effect on the system as it runs, yet still meet the restore requirements. Based on the resource requirements, the administrators also choose the recovery model for the database. The recovery model balances logging overhead against the criticality of fully recovering the data. The recovery models are:

- Full

  The data is critical and must be recoverable to the point of failure. All data modifications are logged. All SQL Server 2000 recovery options are available.

- Bulk-logged

  Certain bulk operations (bulk copy operations, SELECT INTO, text processing) can be replayed if necessary, so these operations are not fully logged. Can only recover to the end of the last database or log backup.

- Simple

  All data modifications made since last backup are expendable, or can be redone. Lowest logging overhead, but cannot recover past the end of the last backup.