Praveen Madupu +91 98661 30093
Sr SQL Server DBA, Dubai
praveensqldba12@gmail.com

Database Mirroring in SQL Server is a high-availability solution that maintains two copies of a single database—one on the principal server (the primary server) and one on the mirror server (the secondary server). The mirror database is a copy of the principal database and is always in a state of recovery.

This setup allows for automatic failover in certain configurations and minimizes downtime in case of hardware or software failures.

Here's a detailed explanation of **SQL Server Database Mirroring**:

1. Modes of Database Mirroring

Database mirroring operates in three different modes:

1. **High-Safety Mode with Automatic Failover** (Synchronous with Witness):
   - This mode uses synchronous data transfer between the principal and mirror databases.
   - A **witness server** is included to enable automatic failover.
   - The transactions are committed on both the principal and mirror databases before the application is notified.
   - If the principal server fails, the mirror server can automatically take over (automatic failover).
2. **High-Safety Mode without Automatic Failover** (Synchronous without Witness):
   - This mode is similar to the high-safety mode but does **not** use a witness server.
   - Failover must be done manually by a DBA.
   - Data synchronization is still guaranteed as transactions are committed on both the principal and mirror databases before acknowledging success.
3. **High-Performance Mode** (Asynchronous):
   - This mode uses **asynchronous** data transfer.
   - Transactions are committed on the principal database without waiting for confirmation from the mirror database.
   - This improves performance but introduces the risk of data loss because the mirror server may not always be up-to-date.
   - In case of failure, manual failover is required.

2. Key Components of Database Mirroring

1. **Principal Database**:
   - The production database where all write operations occur. It sends transaction log records to the mirror database.
2. **Mirror Database**:
   - A copy of the principal database, maintained in a restoring state. It receives transaction logs and applies them to keep in sync with the principal.
3. **Witness Server** (Optional):

- An optional server used in high-safety mode to enable automatic failover. The witness monitors the health of both the principal and mirror databases.
- If the witness detects that the principal server is down, it allows automatic failover to the mirror database.

3. Database Mirroring Process

The basic flow of database mirroring includes:

1. **Transaction Log Sent from Principal to Mirror**:
   - Whenever a transaction is committed on the principal database, the transaction log is sent to the mirror database.
   - Depending on the mode, the principal either waits for an acknowledgment (in synchronous modes) or proceeds without waiting (in asynchronous mode).
2. **Apply Transaction Logs on the Mirror Database**:
   - The mirror database applies the transaction logs continuously to stay up-to-date with the principal database.
3. **Failover (Manual or Automatic)**:
   - In case of a failure on the principal server, failover can be triggered, and the mirror server can take over as the new principal.
   - If configured with a witness, the failover can occur automatically. Otherwise, the DBA must manually initiate failover.

4. Prerequisites for Database Mirroring

Before setting up database mirroring, you need to meet the following prerequisites:

- **SQL Server Edition**: Database mirroring is supported in **SQL Server Standard** and **Enterprise editions** (but some modes, like High Performance, are Enterprise only).
- **Full Recovery Model**: The principal database must be set to the **Full Recovery Model** to ensure transaction log records can be sent and replayed on the mirror.
- **Network Connectivity**: Reliable and high-bandwidth network connections between the principal, mirror, and witness (if used) servers are essential.
- **Similar Hardware and SQL Server Versions**: Both the principal and mirror servers should have similar hardware and must be running compatible SQL Server versions.

5. Configuring Database Mirroring

The steps to configure database mirroring are:

Step 1: Prepare the Principal and Mirror Databases

- Ensure that the principal database is in **Full Recovery Mode**.
- Take a **full backup** of the principal database and restore it on the mirror server using the `WITH NORECOVERY` option.
- Take a **transaction log backup** from the principal and restore it on the mirror using the `WITH NORECOVERY` option.

Step 2: Set Up Mirroring via SQL Server Management Studio (SSMS)

- In SSMS, connect to the principal server and right-click the database, selecting **Tasks > Mirror**.
- Follow the **Configure Security Wizard**, which will guide you through setting up the endpoints and network connections between the principal and mirror servers.
- If you want automatic failover, you can also configure a **witness server** at this stage.

Step 3: Start Mirroring

- After the configuration, click **Start Mirroring**. The mirror database will begin receiving transaction log records from the principal.

6. Monitoring Database Mirroring

You can monitor the health and status of the database mirroring session through SSMS or use T-SQL queries.

1. **SSMS**: Go to the **Database Properties > Mirroring** tab to view the status of the mirroring session.
2. **T-SQL Query**: The following query provides information about the state of the mirroring session:

```
SELECT
    database_id,
    mirroring_state_desc,
    mirroring_role_desc,
    mirroring_safety_level_desc
FROM sys.database_mirroring;
```

1. **SQL Server Agent Jobs**: SQL Server can alert you to issues in the mirroring setup (e.g., connection failures between the principal and mirror).

7. Failover in Database Mirroring

- **Automatic Failover**:
  - If you are using **High-Safety Mode with a Witness**, failover will occur automatically if the principal server fails.
  - The mirror database becomes the new principal database, and the applications can reconnect.
- **Manual Failover**:
  - In **High-Safety Mode without a Witness** or **High-Performance Mode**, you need to manually initiate failover using the `ALTER DATABASE` command.
  - To manually failover, use the following T-SQL:

```
ALTER DATABASE [YourDatabase] SET PARTNER FAILOVER;
```

8. Advantages of Database Mirroring

1. **Automatic Failover**: In high-safety mode with a witness, database mirroring supports automatic failover, making it highly suitable for critical applications.
2. **Data Redundancy**: It ensures that there is always a copy of your database on a secondary server that can take over quickly in case of failure.
3. **Minimal Data Loss**: In synchronous modes, no committed transaction is lost during failover.
4. **Simple Configuration**: Compared to more complex high-availability solutions (like Always On Availability Groups), database mirroring is relatively simple to set up.
5. **Faster Recovery**: Failover can happen in seconds, minimizing downtime.

9. Limitations of Database Mirroring

1. **End of Support**: Microsoft has deprecated database mirroring in favor of **Always On Availability Groups**, so it may not be supported in future SQL Server versions.
2. **Single Database Scope**: Mirroring works only on a per-database basis, unlike **Always On**, which can protect multiple databases.
3. **Hardware Requirements**: For optimal performance, similar hardware configurations are recommended for both principal and mirror servers.
4. **No Load Balancing**: The mirror server is in a state of recovery and cannot be used for read operations (unlike Always On replicas which can support read-only queries).

10. Scenarios for Using Database Mirroring

- **Critical Applications**: Database mirroring is ideal for applications that require high availability and minimal downtime.
- **Smaller Deployments**: It's simpler to configure than Always On Availability Groups and might be more suitable for smaller SQL Server environments.
- **Cost Considerations**: Although superseded by Always On, database mirroring in SQL Server **Standard Edition** provides a cost-effective high-availability solution.

Conclusion

Database mirroring is a robust high-availability and disaster recovery solution in SQL Server that ensures continuous availability of databases by maintaining a hot standby copy.

However, with its deprecation in favor of Always On Availability Groups, it is advisable to consider the future support and requirements of your SQL Server environment when choosing between the two solutions.