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

A **"Suspect" database** in SQL Server occurs when the database fails to start due to issues such as corruption, lack of disk space, missing log files, or failed recovery processes. When a database is marked as "Suspect," users cannot access it, and it requires intervention to bring it back online.

Below are the steps to fix a **DB Suspect** issue in SQL Server, along with a detailed example.

Common Causes of Suspect Databases

- **Corruption** in the database.
- **Disk issues** (e.g., disk space full, disk failure).
- **Missing or corrupted transaction log files**.
- **Inconsistent system shutdowns**, power failures, or crashes.
- SQL Server service **crashes** or improper shutdowns during critical operations.

Steps to Fix a Database in Suspect Mode

Step 1: Set the Database to Emergency Mode

Emergency mode allows read-only access to the database and provides an opportunity to troubleshoot and fix the issue.

```
ALTER DATABASE [YourDatabaseName] SET EMERGENCY;
```

Example:

```
ALTER DATABASE [MyDB] SET EMERGENCY;
```

This step sets the database to **Emergency Mode**, which allows you to access the database in a limited capacity.

Step 2: Set the Database to Single-User Mode

Single-user mode ensures that only one connection can access the database at a time. This is important because multiple users can interfere with the recovery process.

```
ALTER DATABASE [YourDatabaseName] SET SINGLE_USER WITH ROLLBACK IMMEDIATE;
```

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

Example:

```
ALTER DATABASE [MyDB] SET SINGLE_USER WITH ROLLBACK IMMEDIATE;
```

This will forcibly close any active connections to the database and switch it to **Single-User Mode**.

Step 3: Run DBCC CHECKDB to Assess the Integrity of the Database

Now that the database is in Emergency Mode, run **DBCC CHECKDB** to check for corruption and assess the damage. This step will help determine if the database is recoverable or if there is significant damage.

```
DBCC CHECKDB ([YourDatabaseName]);
```

Example:

```
DBCC CHECKDB ([MyDB]);
```

This command checks the physical and logical integrity of all the objects in the specified database.

- **If there is no corruption**, proceed to the next step.
- **If there is corruption**, the command will provide details of the affected objects. Depending on the severity, you may need to perform repairs using one of the repair options.

Step 4: Repair the Database

If corruption is detected, you can attempt to repair the database using **DBCC CHECKDB** with one of the following repair options:

- **REPAIR_REBUILD**: This option fixes issues related to non-clustered indexes.
- **REPAIR_ALLOW_DATA_LOSS**: This option is more aggressive and may cause some data loss, but it can recover the database in situations where other options fail.

Run the repair command based on the severity of the corruption. Start with the less aggressive **REPAIR_REBUILD**, and if that doesn't fix the problem, use **REPAIR_ALLOW_DATA_LOSS**.

```
DBCC CHECKDB ([YourDatabaseName], REPAIR_REBUILD);
```

If the REPAIR_REBUILD doesn't resolve the issue, use:

```
DBCC CHECKDB ([YourDatabaseName], REPAIR_ALLOW_DATA_LOSS);
```

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

Example:

- The **REPAIR_ALLOW_DATA_LOSS** option may result in some data being deleted to maintain database integrity. **Back up the database before running this command.**

Step 5: Set the Database Back to Multi-User Mode

After repairing the database, return it to **Multi-User Mode** so that normal operations can resume.

```
ALTER DATABASE [YourDatabaseName] SET MULTI_USER;
```

Example:

```
ALTER DATABASE [MyDB] SET MULTI_USER;
```

Step 6: Bring the Database Online

Now, bring the database online using the following command:

```
ALTER DATABASE [YourDatabaseName] SET ONLINE;
```


Example:

```
ALTER DATABASE [MyDB] SET ONLINE;
```

At this point, the database should be online and fully accessible.

If the repair was successful, the database will now function normally.

Step 7: Backup the Database

After fixing the suspect issue, **immediately back up the database** to ensure that you have a safe point to recover from in the future.

```
BACKUP DATABASE [YourDatabaseName] TO DISK = 'C:\Backup\YourDatabaseName.bak';
```

Example:

```
BACKUP DATABASE [MyDB] TO DISK = 'C:\Backup\MyDB.bak';
```

This is important to prevent future data loss if the same issue arises again.

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

-- 1. Set the database to Emergency Mode

```
ALTER DATABASE [MyDB] SET EMERGENCY;
```

-- 2. Set the database to Single-User Mode

```
ALTER DATABASE [MyDB] SET SINGLE_USER WITH ROLLBACK IMMEDIATE;
```

-- 3. Run DBCC CHECKDB to check for corruption

```
DBCC CHECKDB ([MyDB]);
```

-- 4. If corruption is detected, attempt a repair

```
DBCC CHECKDB ([MyDB], REPAIR_REBUILD);  → With No Data Loss

DBCC CHECKDB ([MyDB], REPAIR_ALLOW_DATA_LOSS); → With Data Loss
```

-- 5. Set the database back to Multi-User Mode

```
ALTER DATABASE [MyDB] SET MULTI_USER;
```


-- 6. Bring the database back online

```
ALTER DATABASE [MyDB] SET ONLINE;
```

-- 7. Backup the database immediately after recovery

```
BACKUP DATABASE [MyDB] TO DISK = 'C:\Backup\MyDB.bak';
```

Considerations When Dealing with Suspect Databases

- **Data Loss**: If you need to use **REPAIR_ALLOW_DATA_LOSS**, it's important to recognize that some data might be lost during the repair. This should be a last resort.
- **Backup First**: Always try to **back up the database** before performing any repair operations, especially when using **REPAIR_ALLOW_DATA_LOSS**.
- **Hardware Issues**: Investigate hardware issues if you frequently encounter suspect database issues. Disk corruption or disk space issues can lead to such problems.
- **Error Logs**: Always review the SQL Server **error logs** to understand the underlying issue that caused the database to become suspect in the first place.

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

Preventing a Suspect Database

To avoid databases going into suspect mode, consider the following preventive measures:

- **Regular Backups**: Ensure regular backups are scheduled and successful.
- **Disk Maintenance**: Ensure the disk has sufficient space and is regularly checked for errors (e.g., using `chkdsk`).
- **Monitoring**: Set up alerts to monitor SQL Server performance and logs for early warnings of potential issues.
- **Consistent Power Supply**: Use Uninterruptible Power Supplies (UPS) to avoid improper shutdowns.

Summary

Fixing a database in **Suspect mode** requires careful attention and proper troubleshooting.

By following the steps above, you can attempt to recover the database while minimizing data loss.

Always make sure you have a reliable backup system in place to avoid critical data loss.

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