Checklist & Implementation: Monthly Partitioning in SQL Server

Step 1: Identify Partition Function & Scheme

```
-- Identify partition function and scheme for the table
SELECT pf.name AS PartitionFunction, ps.name AS PartitionScheme
FROM sys.indexes i
JOIN sys.partition_schemes ps ON i.data_space_id = ps.data_space_id
JOIN sys.partition_functions pf ON ps.function_id = pf.function_id
WHERE i.object_id = OBJECT_ID('Fdi_transaction_monetory');
```

Step 2: View Current Partition Boundaries

```
-- View existing partition boundaries
SELECT value AS BoundaryValue
FROM sys.partition_range_values
WHERE function_id = (
    SELECT function_id FROM sys.partition_functions WHERE name = 'YourPartitionFunction'
)
ORDER BY value;
```

Step 3: Check Partitioning Column

```
-- Verify the partitioning column used in the table
SELECT c.name AS PartitionColumn
FROM sys.index_columns ic
JOIN sys.columns c ON ic.object_id = c.object_id AND ic.column_id = c.column_id
WHERE ic.object_id = OBJECT_ID('Fdi_transaction_monetory')
  AND ic.partition_ordinal > 0;
```

Step 4: List Existing Filegroups (Optional)

```
-- Check for existing filegroups
SELECT name AS FilegroupName
FROM sys.filegroups
WHERE name LIKE 'FG_%'
ORDER BY name;
```

Step 5: Check Disk Space (Optional)

```
-- Check disk space
EXEC xp_fixeddrives;
```

Step 6: Monitor Active Locks (Optional)

```
-- Check for active locks on the table
SELECT *
FROM sys.dm_tran_locks
```

```
WHERE resource_associated_entity_id = OBJECT_ID('Fdi_transaction_monetory');
```

Step 7: Count Current Partitions

```
-- Count partitions used by the table
SELECT COUNT(*)
FROM sys.partitions
WHERE object_id = OBJECT_ID('Fdi_transaction_monetory');
```

Step 8: Add Monthly Partitions (2026-2050)

```
-- Automatically add monthly partitions up to 2050-12
DECLARE @PartitionFunction SYSNAME = 'pf_FdiTransactionDate';
DECLARE @year INT = 2026, @month INT = 1, @LastDay DATE, @SQL NVARCHAR(MAX);

WHILE @year <= 2050
BEGIN
    WHILE @month <= 12
    BEGIN
        SET @LastDay = EOMONTH(DATEFROMPARTS(@year, @month, 1));
        SET @SQL = 'ALTER PARTITION FUNCTION [' + @PartitionFunction + ']() SPLIT RANGE ('''
                    + CONVERT(CHAR(10), @LastDay, 120) + ''')';
        PRINT @SQL;
        EXEC (@SQL);
        SET @month += 1;
    END
    SET @month = 1;
    SET @year += 1;
END
```