**Step 2 – Create a purge control table**

Paste this in SSMS:

IF OBJECT\_ID('tempdb..#PurgeList') IS NOT NULL DROP TABLE #PurgeList;

CREATE TABLE #PurgeList (

Id INT IDENTITY(1,1) PRIMARY KEY,

SchemaName SYSNAME NOT NULL,

TableName SYSNAME NOT NULL,

ColumnName SYSNAME NOT NULL

);

-- Example entries (replace with your 100 tables and date columns)

INSERT INTO #PurgeList (SchemaName, TableName, ColumnName)

VALUES

('dbo','Orders','OrderDate'),

('dbo','Customers','CreatedAt'),

('dbo','AuditLog','LogDate');

👉 You must fill in your 100 rows with the correct **Schema, Table, Column**.

**Step 3 – Batch purge older than 10 years**

This loop will go through all 100 tables and delete old rows in **10,000 row batches** (so your log won’t explode).

DECLARE @Schema SYSNAME, @Table SYSNAME, @Column SYSNAME, @sql NVARCHAR(MAX);

DECLARE purgeCursor CURSOR LOCAL FAST\_FORWARD FOR

SELECT SchemaName, TableName, ColumnName FROM #PurgeList;

OPEN purgeCursor;

FETCH NEXT FROM purgeCursor INTO @Schema, @Table, @Column;

WHILE @@FETCH\_STATUS = 0

BEGIN

SET @sql = '

DECLARE @BatchSize INT = 10000;

WHILE 1=1

BEGIN

DELETE TOP (@BatchSize)

FROM ' + QUOTENAME(@Schema) + '.' + QUOTENAME(@Table) + '

WHERE ' + QUOTENAME(@Column) + ' < DATEADD(year,-10,GETDATE());

IF @@ROWCOUNT = 0 BREAK;

END';

PRINT 'Purging: ' + @Schema + '.' + @Table;

EXEC sp\_executesql @sql;

FETCH NEXT FROM purgeCursor INTO @Schema, @Table, @Column;

END

CLOSE purgeCursor;

DEALLOCATE purgeCursor;

**⚠️ Safety Notes**

1. **Back up your database first** ✅
2. Test on **1–2 tables manually** with the small batch delete (I can give you that snippet).
3. Run the full loop only when you’re confident.
4. If some tables are referenced by **foreign keys**, purge **child tables first**, otherwise deletes will fail.
5. After purging, run UPDATE STATISTICS and/or rebuild indexes on large tables.