CREATE TABLE Logs(

Id int NOT NULL IDENTITY,

LogDate datetime,

MsgText nvarchar(300),

CONSTRAINT PK\_Logs\_Id PRIMARY KEY (Id)

)

GO

ALTER PROC usp\_Add1MilionLogs

AS

DECLARE @counter int

SET @counter = 1;

WHILE(@counter < 10000000)

BEGIN

DECLARE @Date datetime

SET @Date =

DATEADD(month, CONVERT(varbinary, newid()) % (50 \* 12), getdate())

INSERT INTO Logs (LogDate, MsgText)

VALUES(@Date, 'dsd')

SET @counter = @counter + 1;

END

GO

EXEC usp\_Add1MilionLogs

CHECKPOINT; DBCC DROPCLEANBUFFERS;

----------------------------TASK 1--------------------------------

SELECT l.LogDate

FROM Logs l

WHERE YEAR(l.LogDate) < 2012 AND YEAR(l.LogDate) > 2001

----------------------------TASK 2--------------------------------

CREATE INDEX IDX\_Logs\_LogDate ON Logs(LogDate)

DROP INDEX Logs.IDX\_Logs\_LogDate

SELECT l.LogDate

FROM Logs l

WHERE YEAR(l.LogDate) < 2012 AND YEAR(l.LogDate) > 2001

----------------------------TASK 3--------------------------------

DROP INDEX Logs.IDX\_Logs\_LogDate

CREATE INDEX IDX\_Logs\_MsgText ON Logs(MsgText)

SELECT l.MsgText

FROM Logs l

WHERE YEAR(l.LogDate) < 2012 AND YEAR(l.LogDate) > 2001

-------------------------TASK 4 --------------------------------

CREATE DATABASE PartitioningDB;

USE PartitioningDB;

CREATE TABLE Logs(

LogId int NOT NULL AUTO\_INCREMENT,

LogDate datetime,

LogText nvarchar(100),

CONSTRAINT PK\_Logs\_LogId PRIMARY KEY(LogId, LogDate)

) PARTITION BY RANGE(YEAR(LogDate))(

PARTITION p0 VALUES LESS THAN (1990),

PARTITION p2 VALUES LESS THAN (2000),

PARTITION p3 VALUES LESS THAN (2010),

PARTITION p4 VALUES LESS THAN MAXVALUE

);

CALL PopulateDB()

SELECT \* FROM Logs PARTITION (p0)

USE DatabasePerformance;

SELECT \* FROM Logs

WHERE LogDate < '1990-01-01'

USE PartitioningDB

SELECT \* FROM Logs PARTITION(p0)

WHERE LogDate < '1990-01-01'