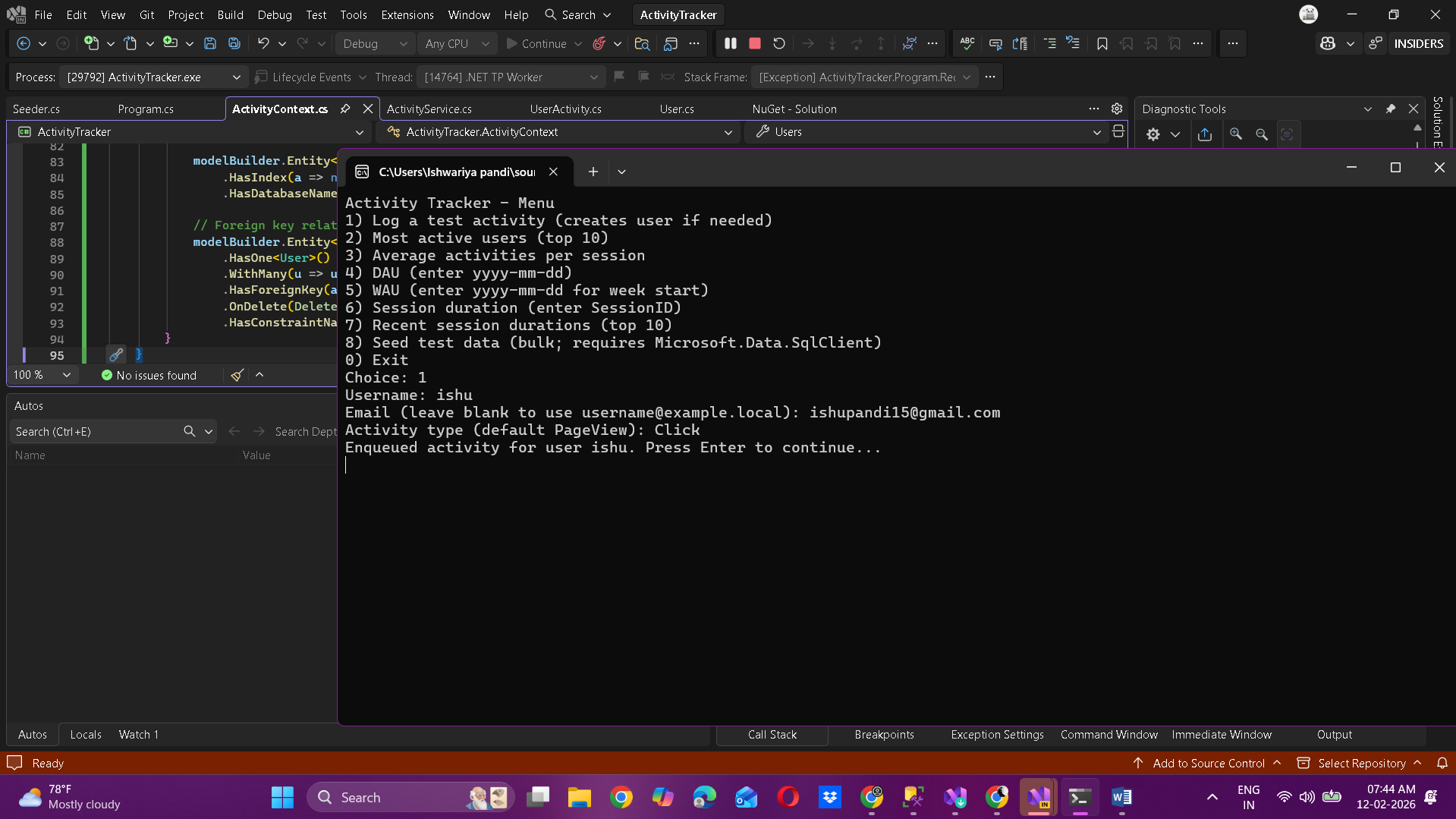
Screenshots of Application Inputs, Outputs, and Corresponding SQL Backend Verification

Choice 1:

  
  
USE UserActivityDB;

GO

SELECT

u.ID AS UserID,

u.Username,

u.Email,

a.ID AS ActivityID,

a.ActivityType,

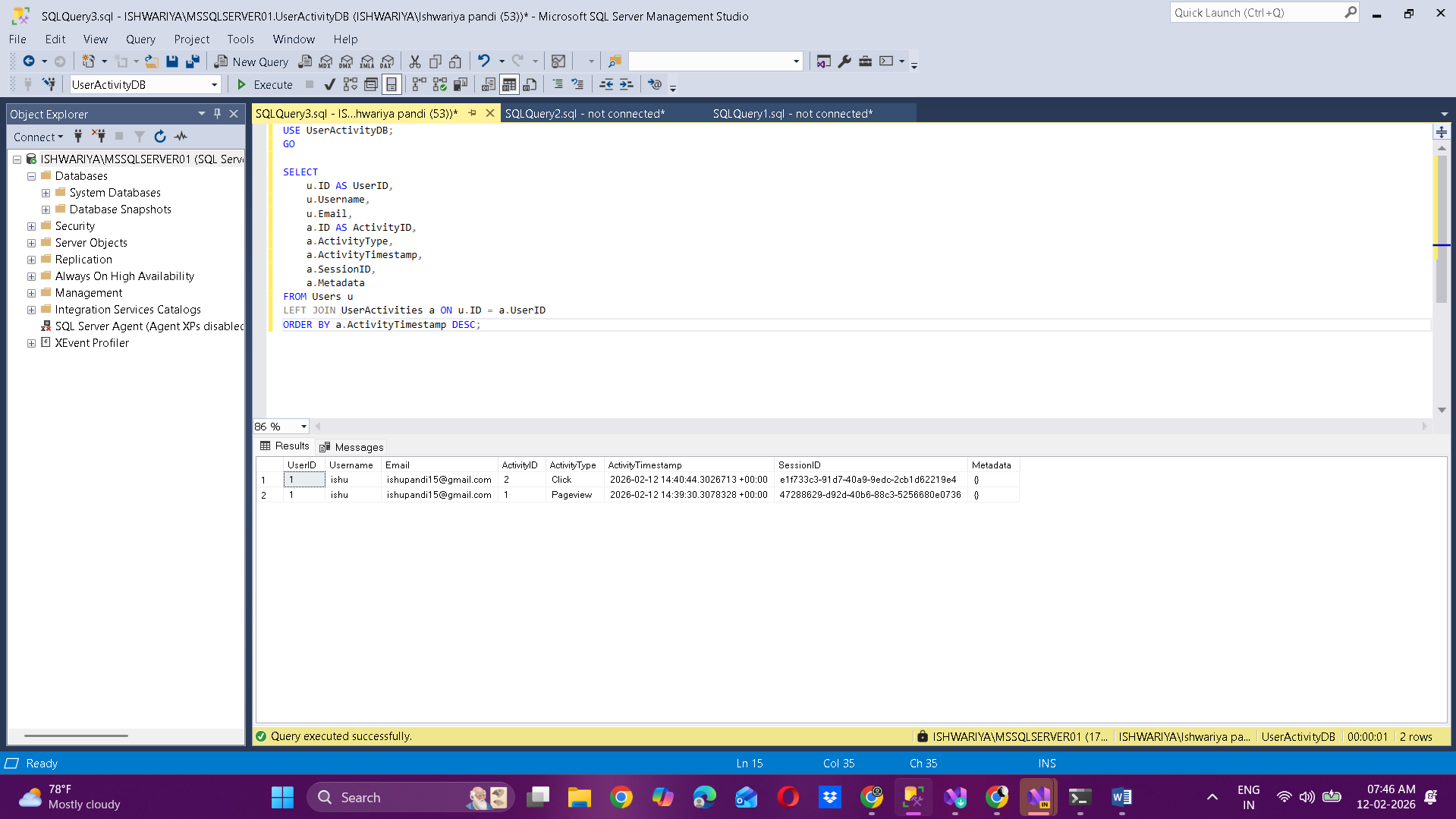
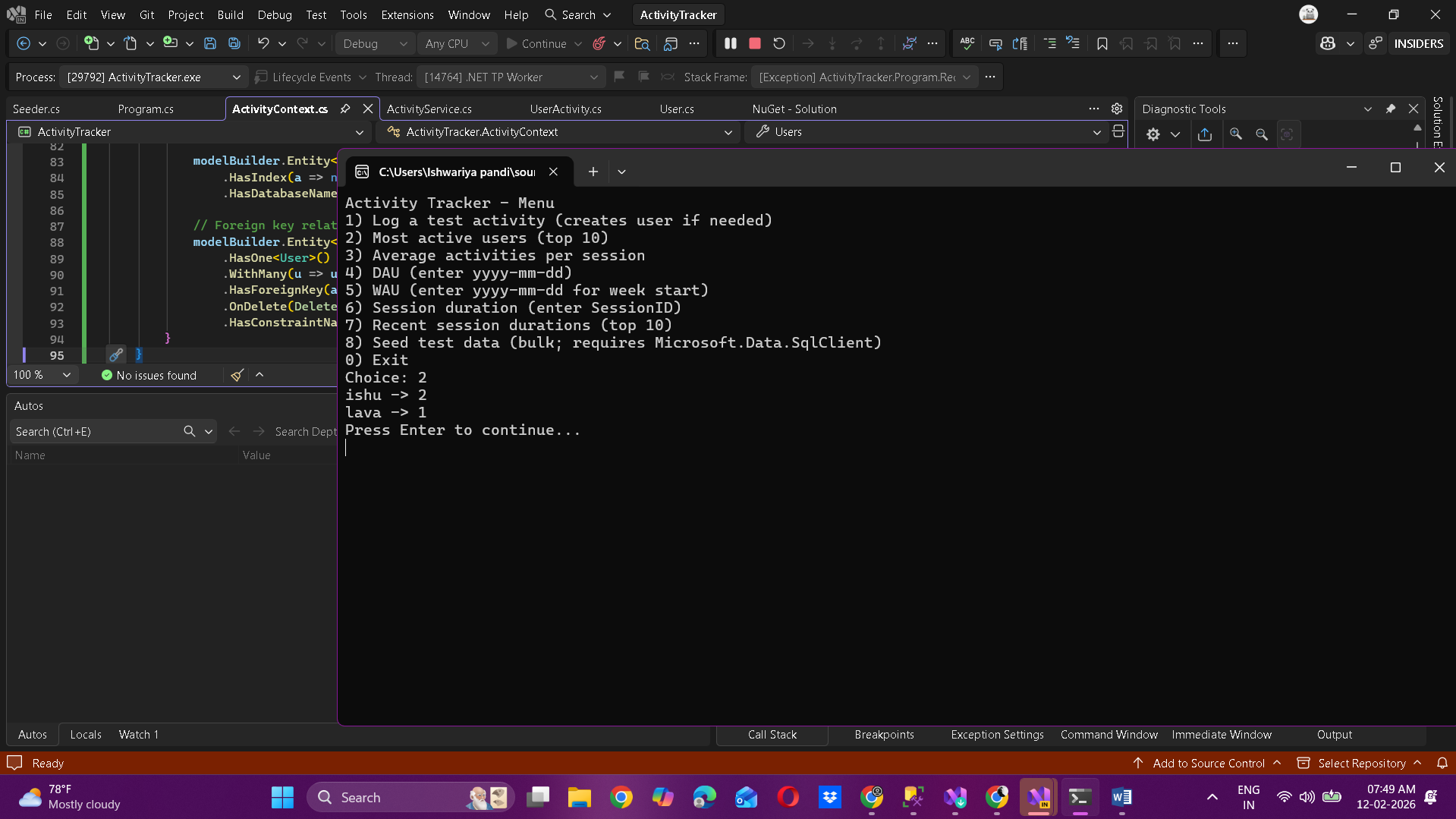
a.ActivityTimestamp,

a.SessionID,

a.Metadata

FROM Users u

LEFT JOIN UserActivities a ON u.ID = a.UserID

ORDER BY a.ActivityTimestamp DESC;  
  
  
  
Choice 2:  
  
  
USE UserActivityDB;

GO

SELECT TOP 10

u.ID AS UserID,

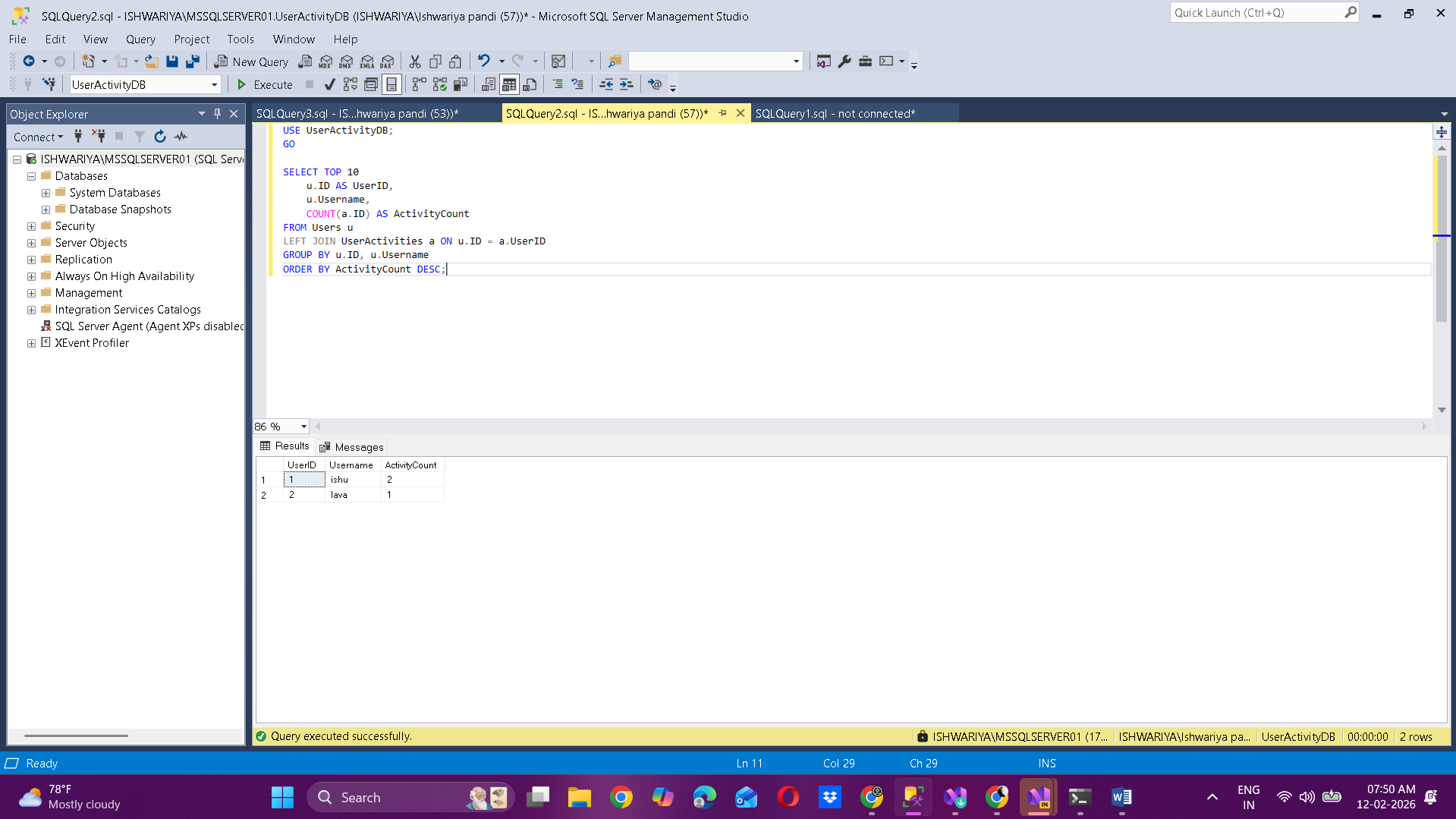
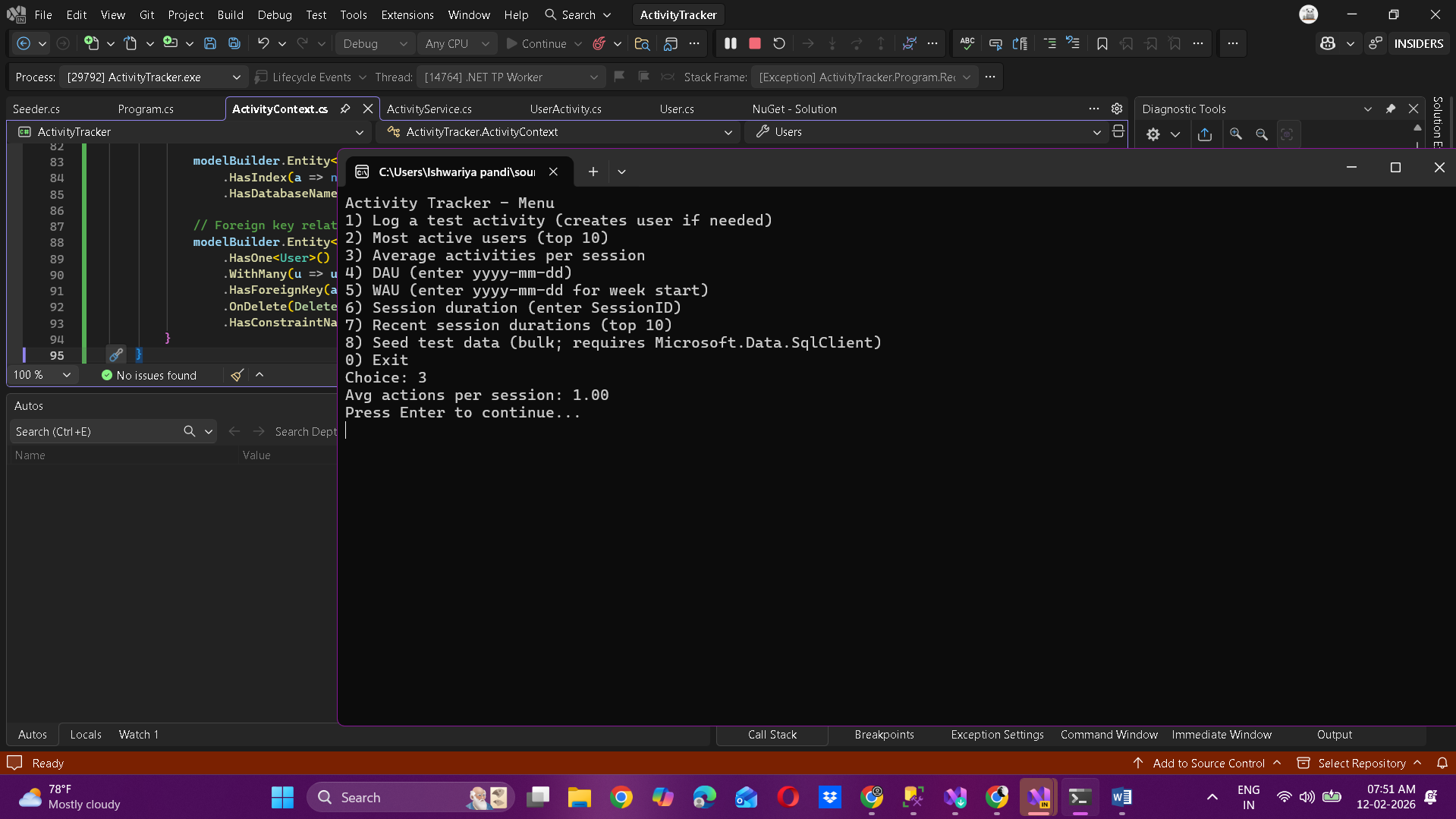
u.Username,

COUNT(a.ID) AS ActivityCount

FROM Users u

LEFT JOIN UserActivities a ON u.ID = a.UserID

GROUP BY u.ID, u.Username

ORDER BY ActivityCount DESC;  
  
  
  
Choice 3:  
  
  
  
USE UserActivityDB;

GO

SELECT

AVG(CAST(SessionCount AS FLOAT)) AS AverageActivitiesPerSession

FROM (

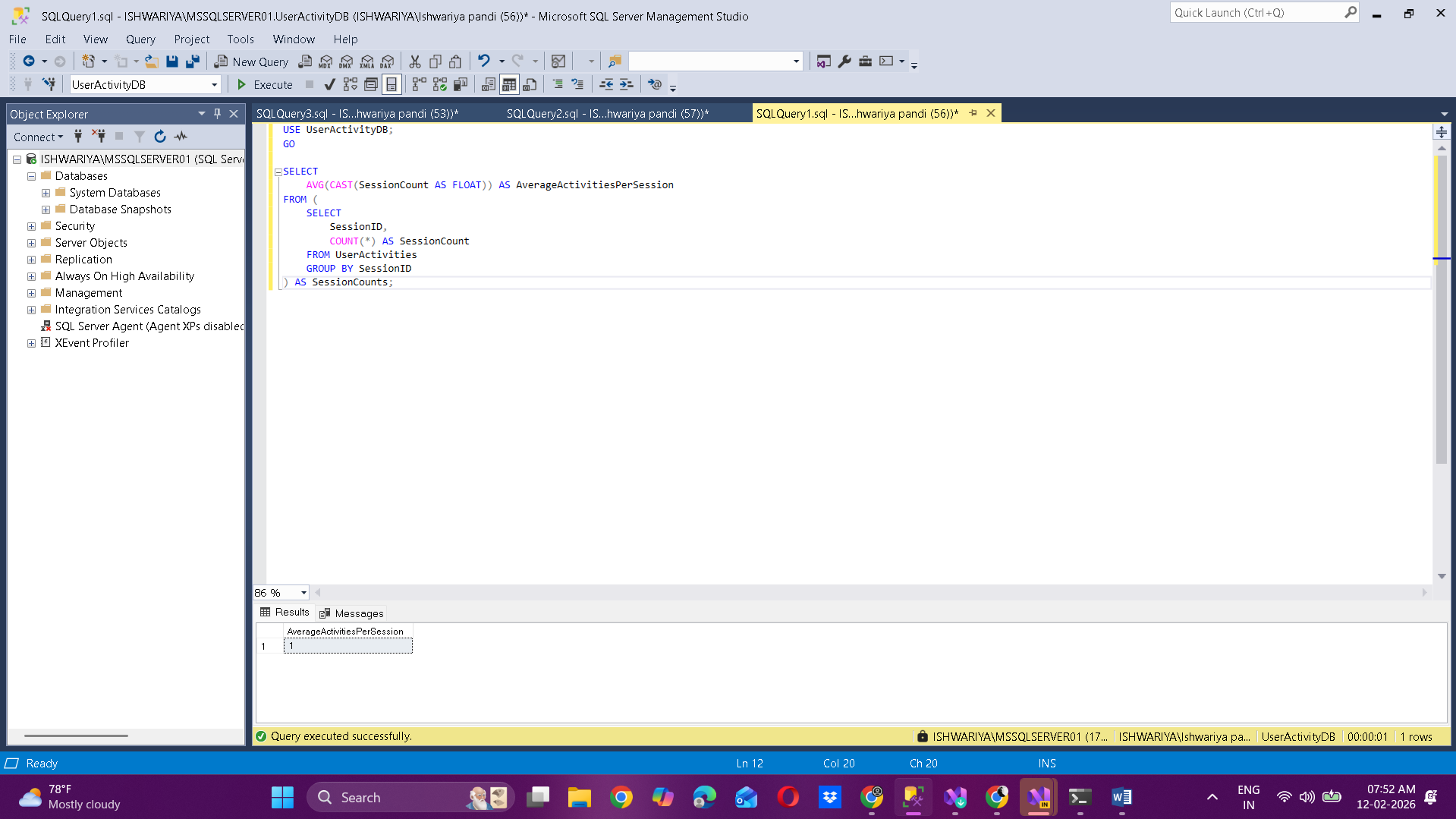
SELECT

SessionID,

COUNT(\*) AS SessionCount

FROM UserActivities

GROUP BY SessionID

) AS SessionCounts;  
  


Choice 4:  
  
  
  
USE UserActivityDB;

GO

DECLARE @TargetDate DATE = '2026-02-12';

;WITH DailyActivity AS

(

SELECT DISTINCT

a.UserID

FROM UserActivities a

WHERE a.ActivityTimestamp >= CAST(@TargetDate AS datetimeoffset)

AND a.ActivityTimestamp < DATEADD(DAY, 1, CAST(@TargetDate AS datetimeoffset))

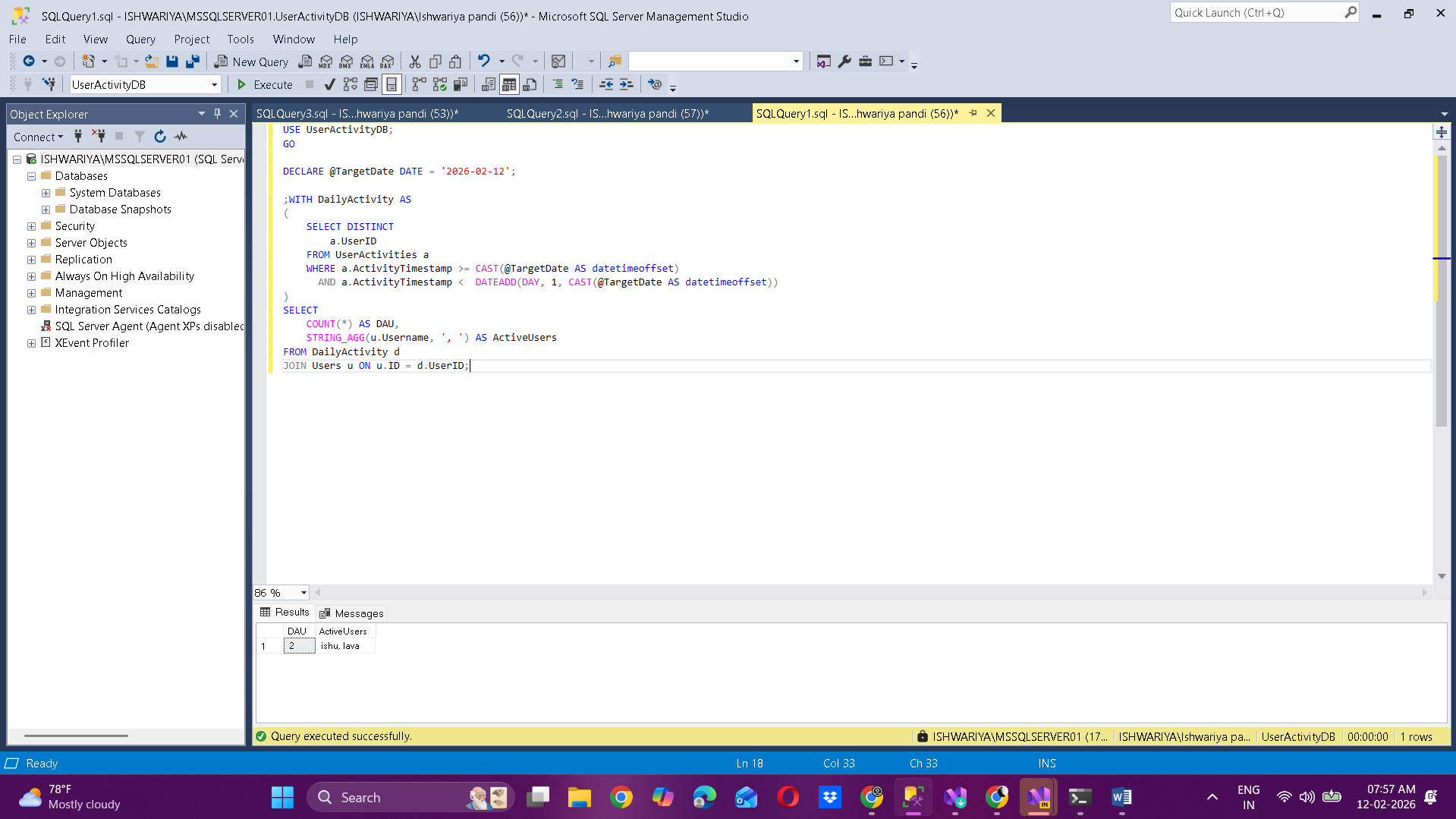
)

SELECT

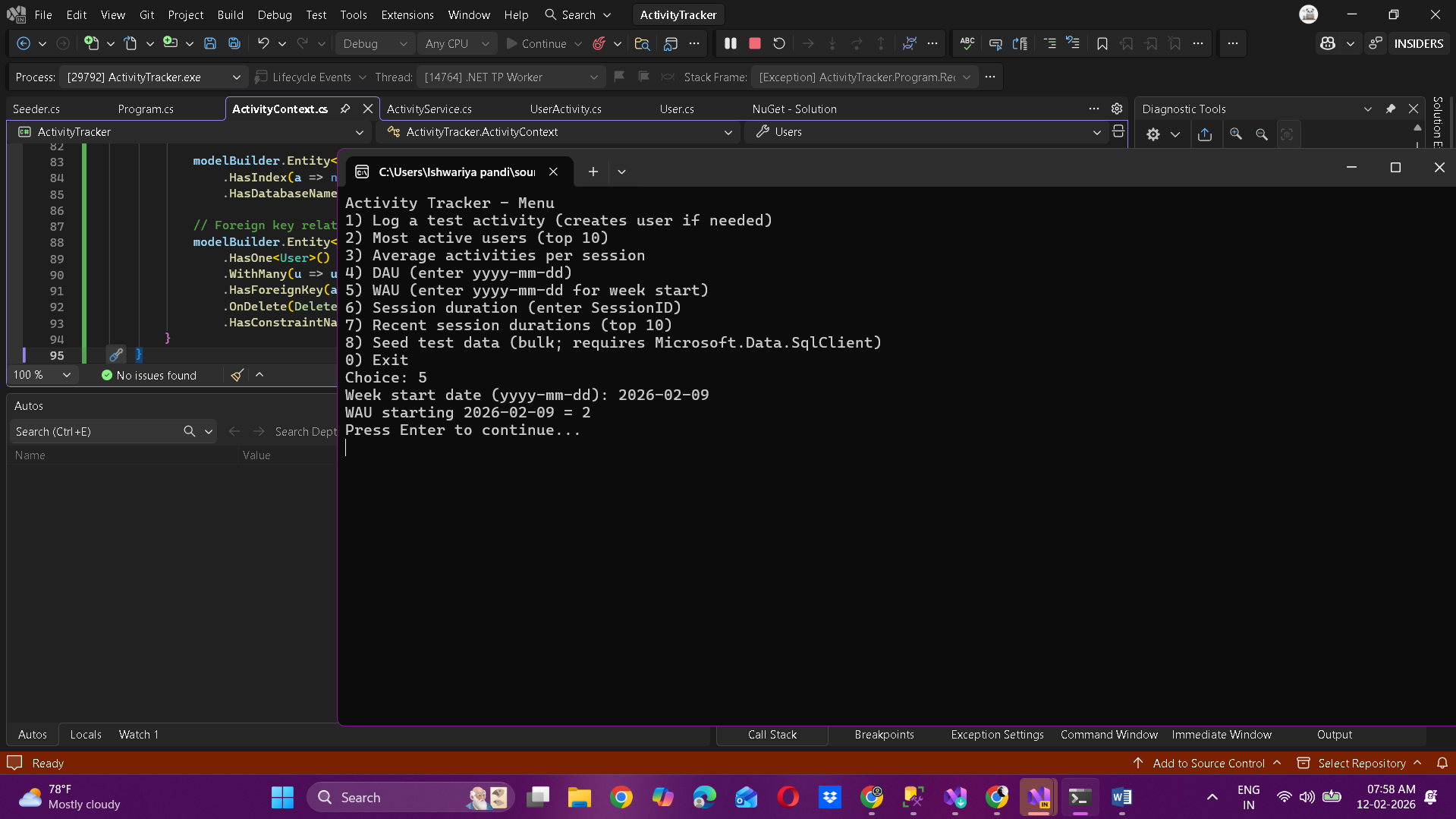
COUNT(\*) AS DAU,

STRING\_AGG(u.Username, ', ') AS ActiveUsers

FROM DailyActivity d

JOIN Users u ON u.ID = d.UserID;  
  


Choice 5:

  
  
USE UserActivityDB;

GO

SET DATEFIRST 1;

DECLARE @InputDate DATE = '2026-02-14'; -- <-- replace as needed

DECLARE @WeekStart DATE = DATEADD(DAY, 1 - DATEPART(WEEKDAY, @InputDate), @InputDate);

DECLARE @WeekEnd DATE = DATEADD(DAY, 6, @WeekStart);

DECLARE @WeekStartDO DATETIMEOFFSET = CAST(@WeekStart AS DATETIMEOFFSET);

DECLARE @WeekEndDO DATETIMEOFFSET = DATEADD(DAY, 1, CAST(@WeekEnd AS DATETIMEOFFSET));

SELECT

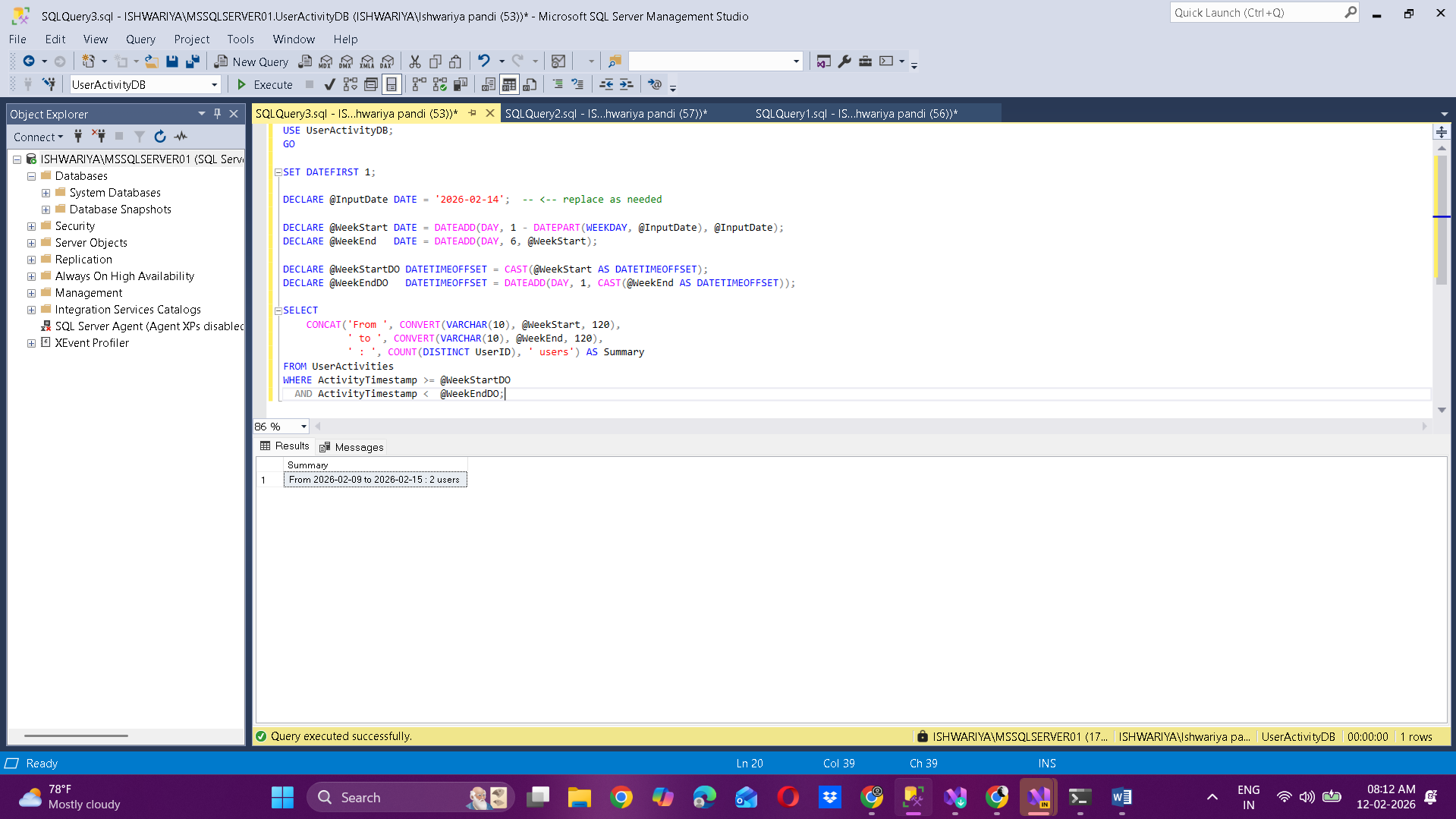
CONCAT('From ', CONVERT(VARCHAR(10), @WeekStart, 120),

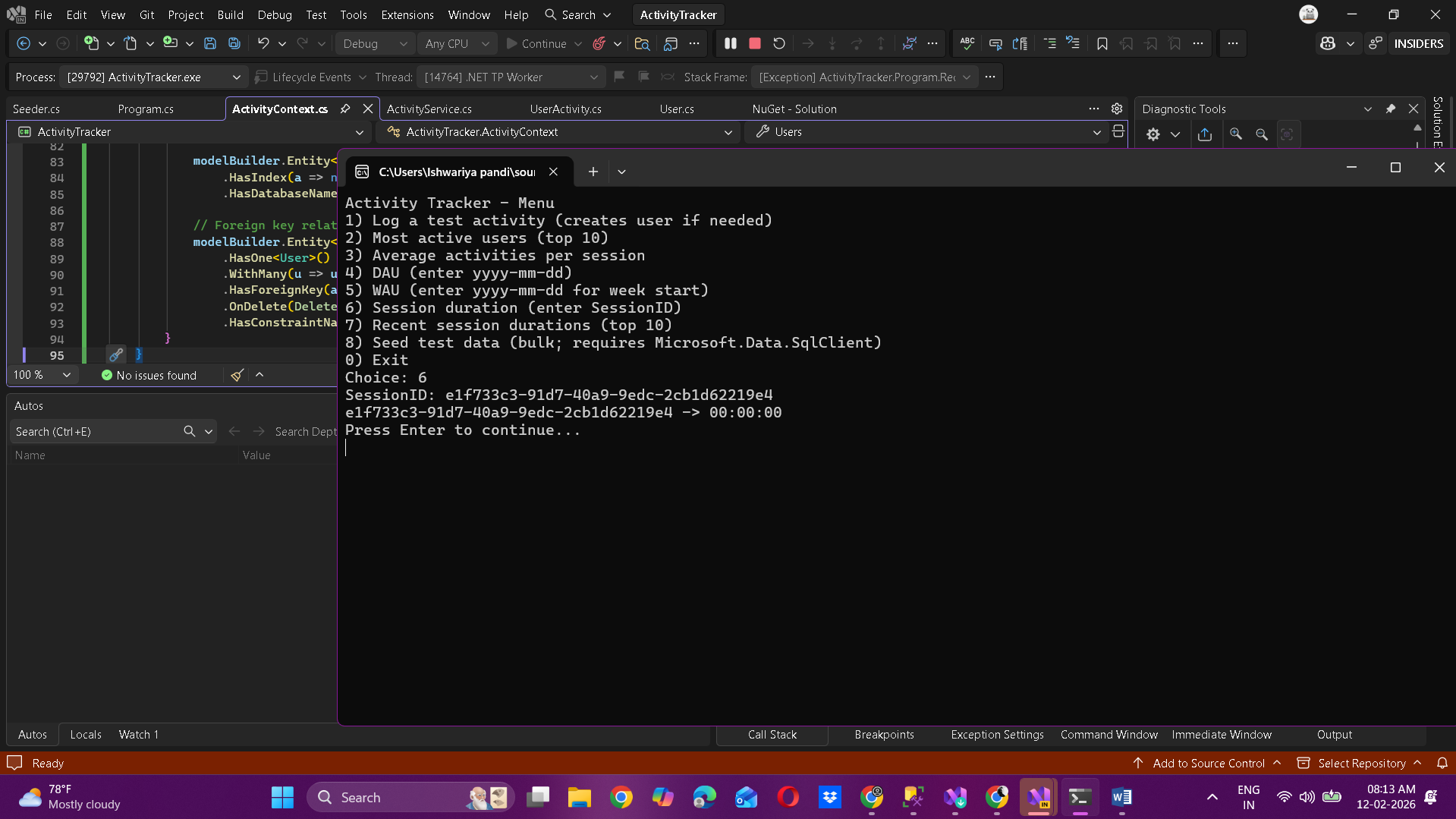
' to ', CONVERT(VARCHAR(10), @WeekEnd, 120),

' : ', COUNT(DISTINCT UserID), ' users') AS Summary

FROM UserActivities

WHERE ActivityTimestamp >= @WeekStartDO

AND ActivityTimestamp < @WeekEndDO;  
  


Choice 6:  
  
  
  
USE UserActivityDB;

GO

-- replace the session id below with the one you saw

DECLARE @SessionId UNIQUEIDENTIFIER = 'e1f733c3-91d7-40a9-9edc-2cb1d62219e4';

SELECT

ID,

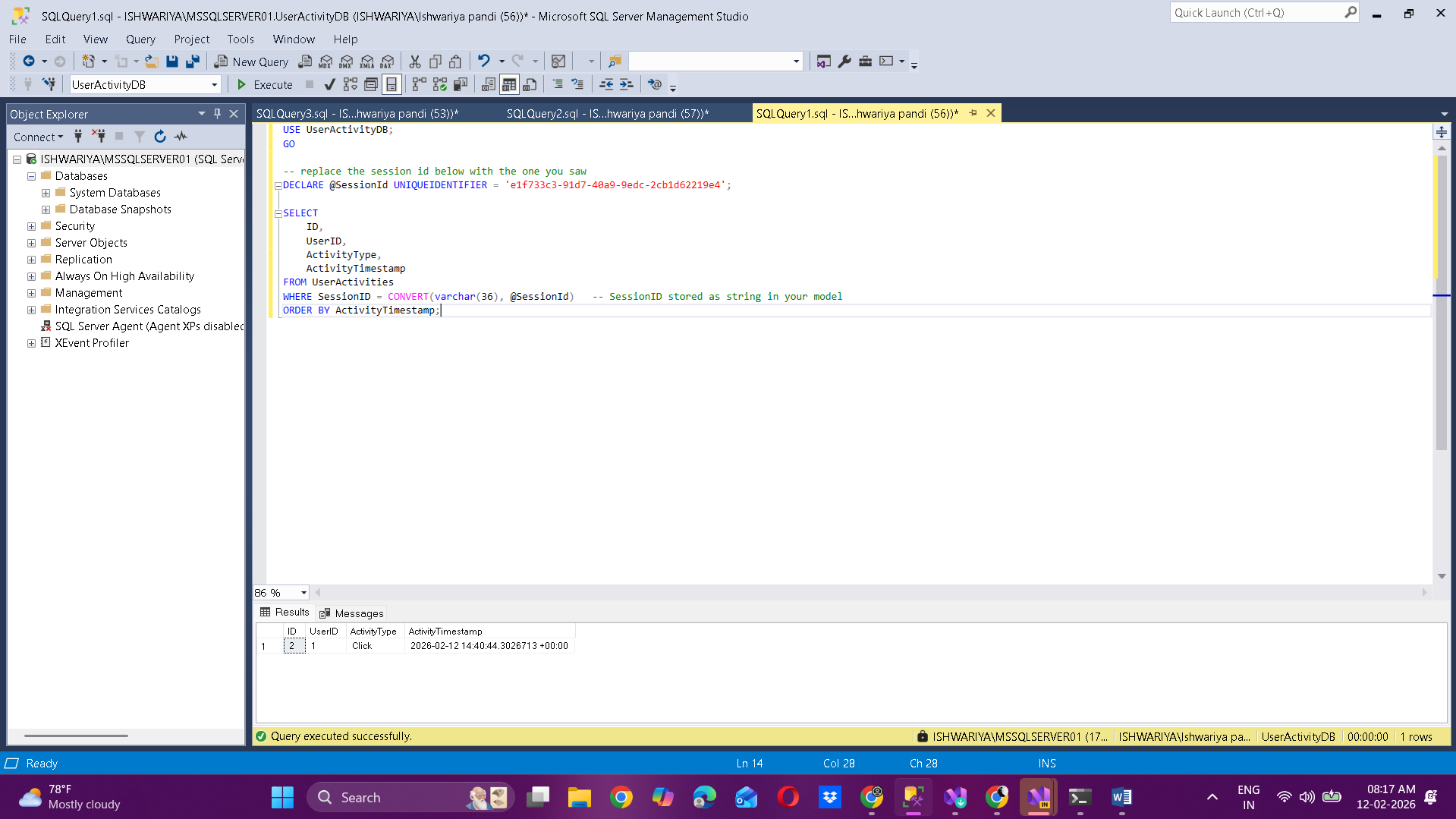
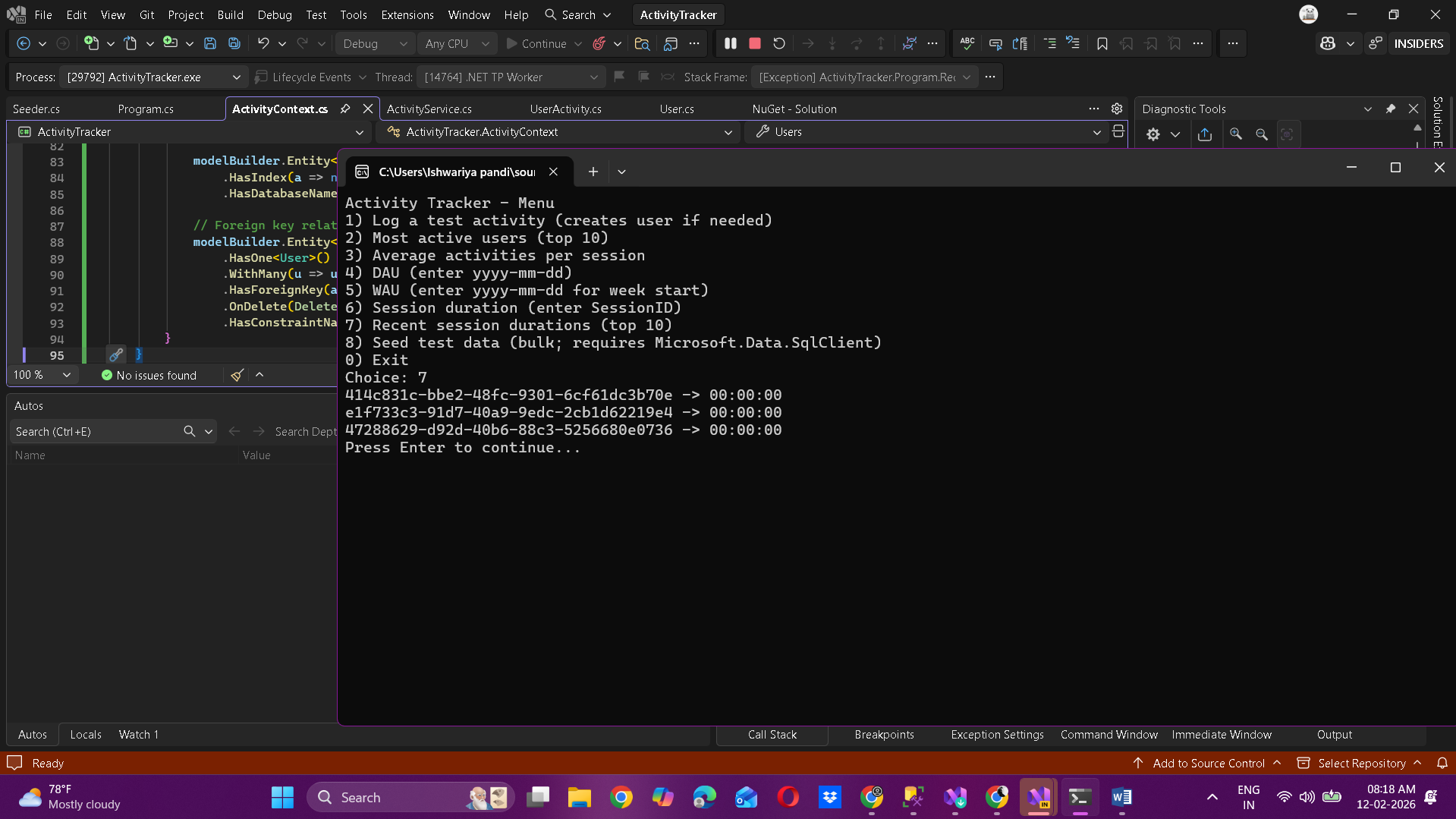
UserID,

ActivityType,

ActivityTimestamp

FROM UserActivities

WHERE SessionID = CONVERT(varchar(36), @SessionId) -- SessionID stored as string in your model

ORDER BY ActivityTimestamp;  
  
  
  
Choice 7:  
  
  
  
SELECT TOP (10)

SessionID,

COUNT(\*) AS ActivityCount,

MIN(ActivityTimestamp) AS StartTime,

MAX(ActivityTimestamp) AS EndTime,

DATEDIFF(SECOND, MIN(ActivityTimestamp), MAX(ActivityTimestamp)) AS DurationInSeconds,

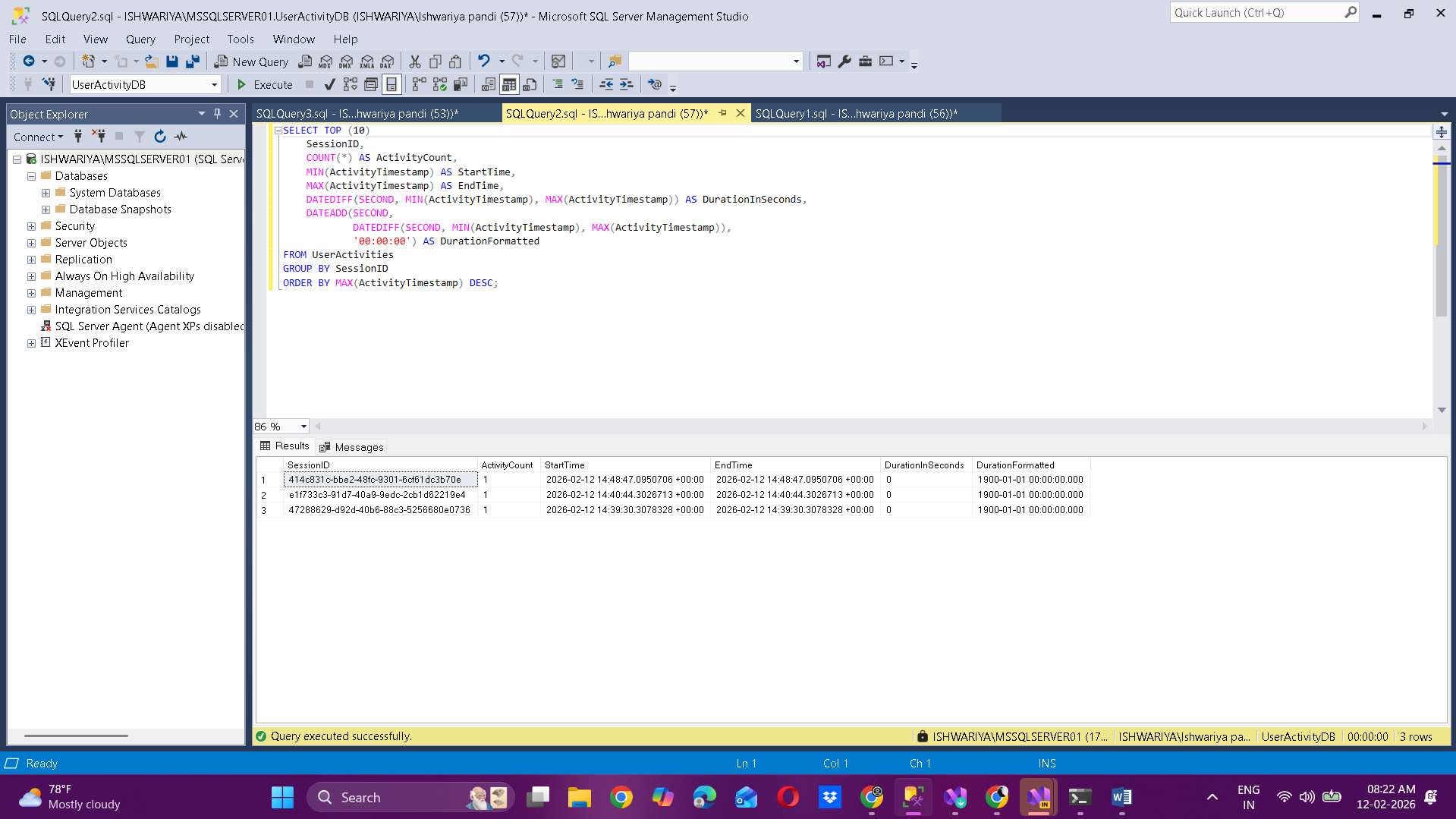
DATEADD(SECOND,

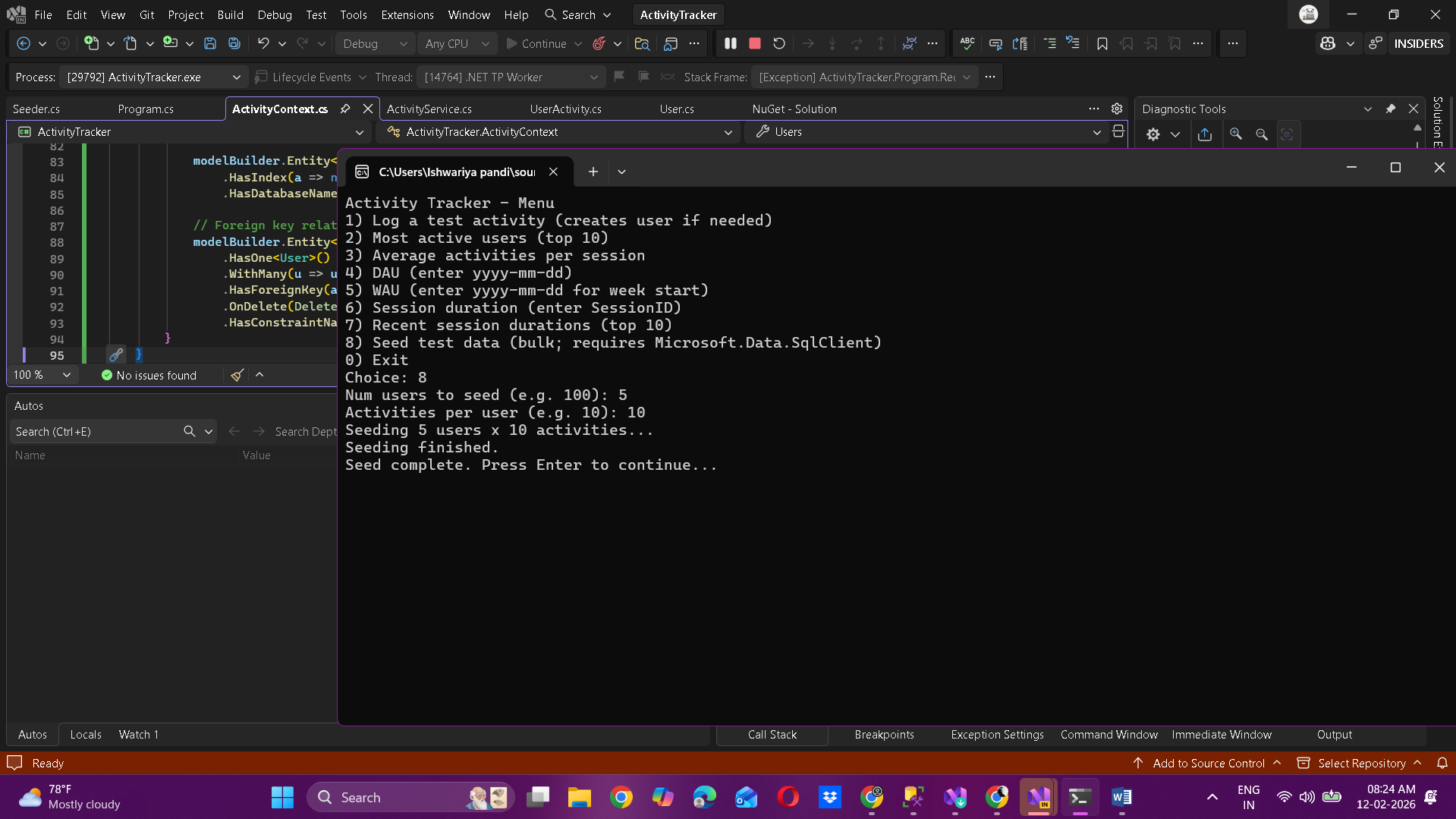
DATEDIFF(SECOND, MIN(ActivityTimestamp), MAX(ActivityTimestamp)),

'00:00:00') AS DurationFormatted

FROM UserActivities

GROUP BY SessionID

ORDER BY MAX(ActivityTimestamp) DESC;  
  


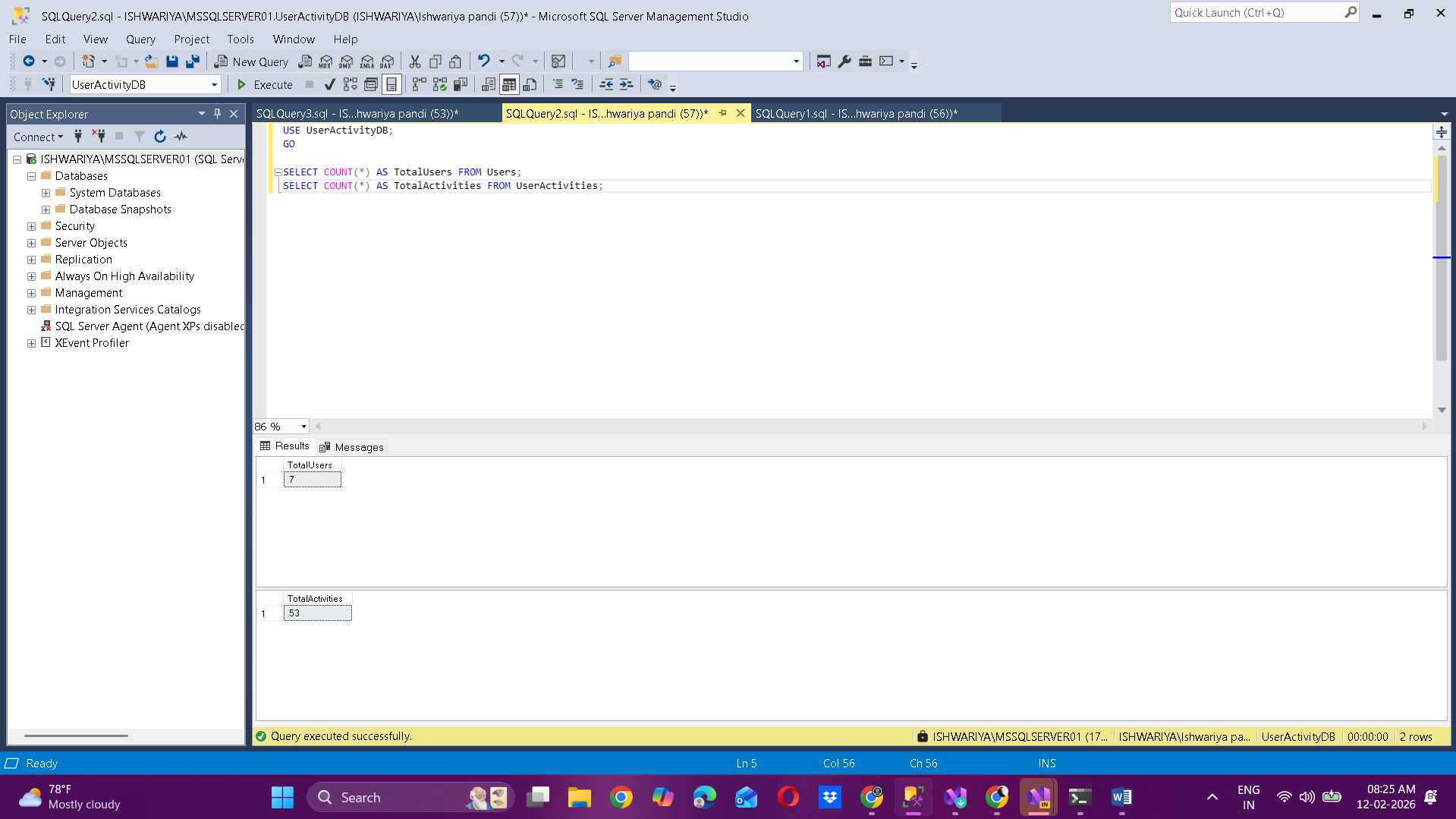
Choice 8:  
  


USE UserActivityDB;

GO

SELECT COUNT(\*) AS TotalUsers FROM Users;

SELECT COUNT(\*) AS TotalActivities FROM UserActivities;

  
  
After seeding:

