**‘AccessBank DBA Internship Program’**

**Project 12: Index Maintenance**

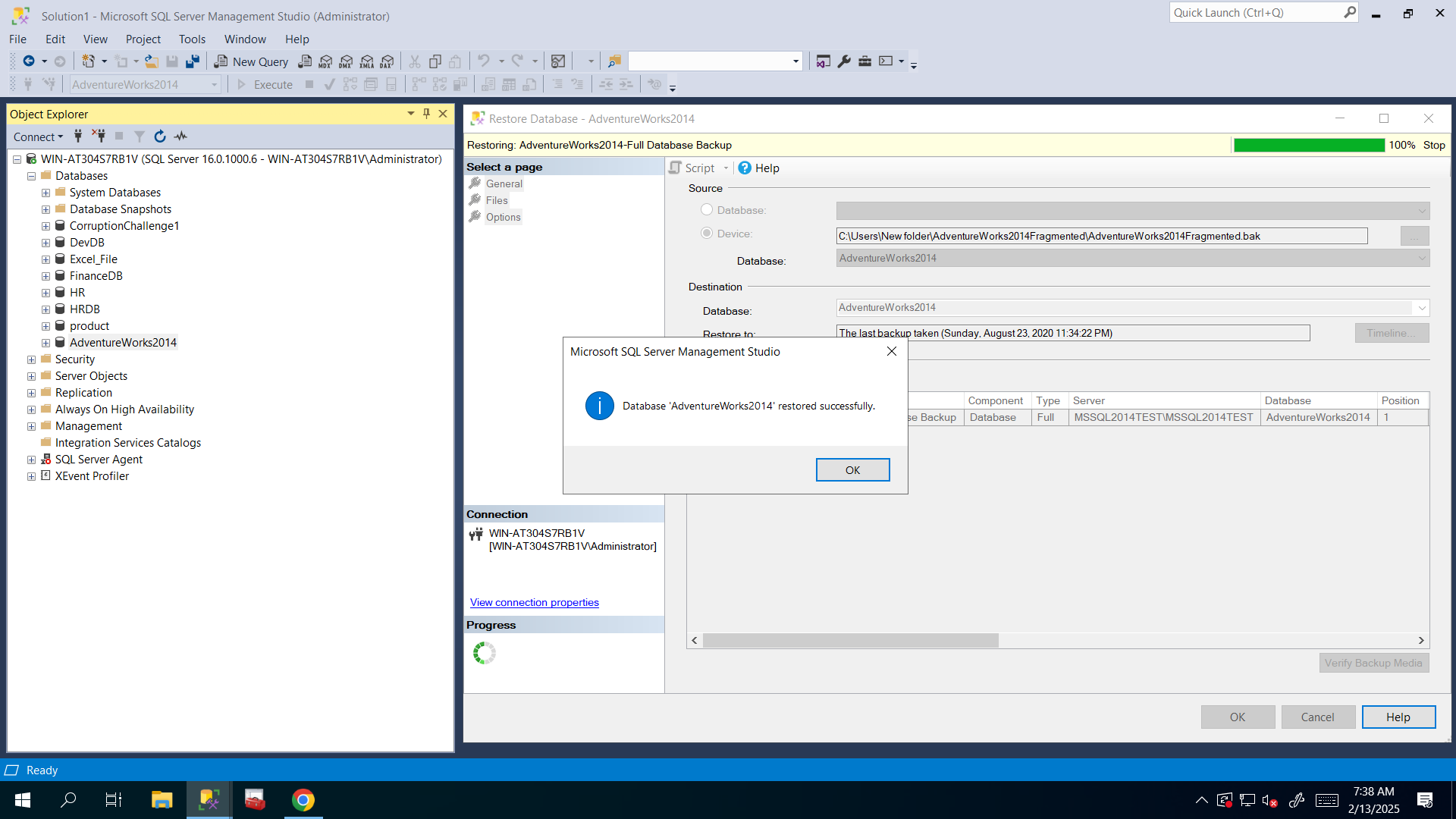
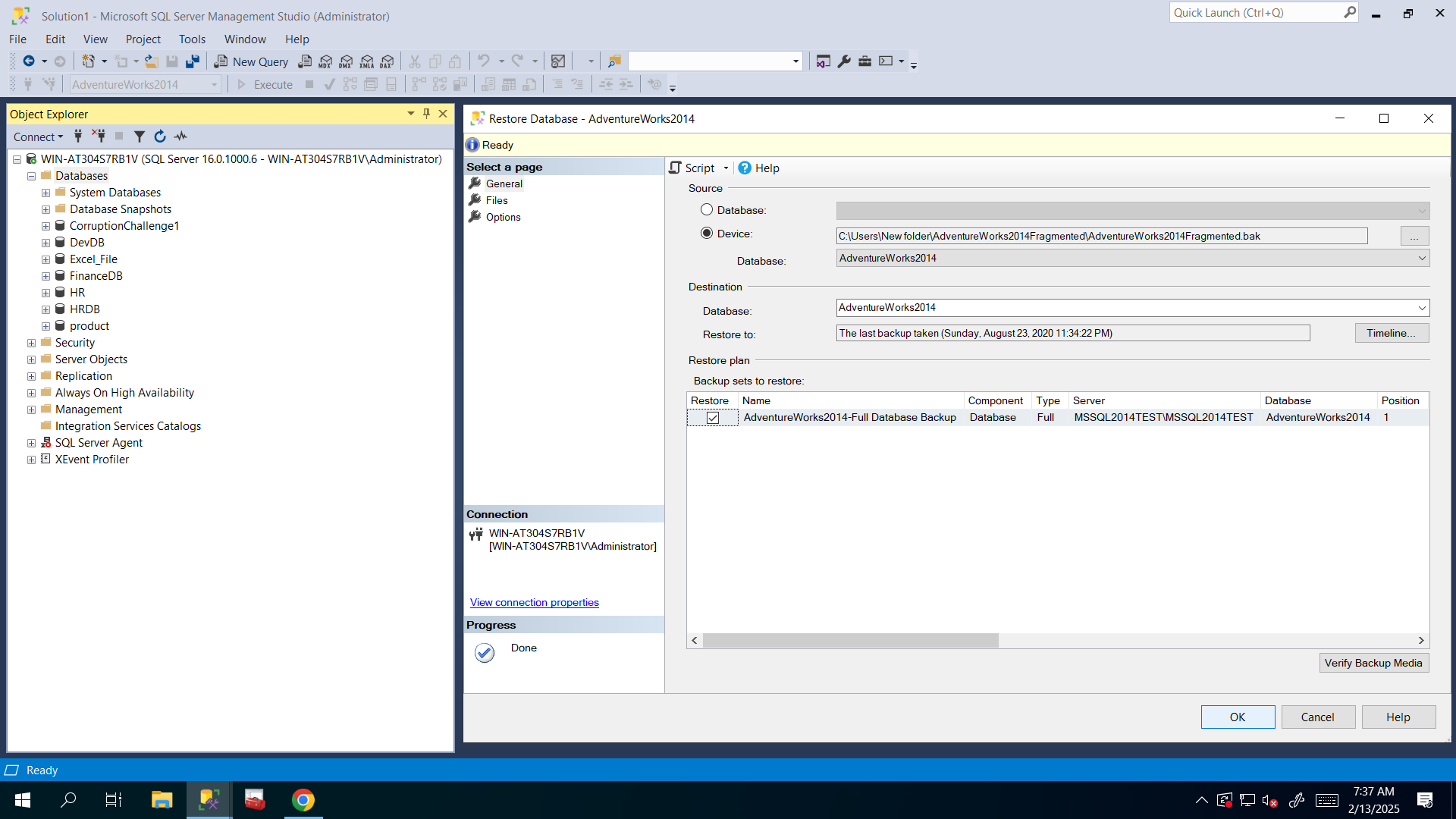
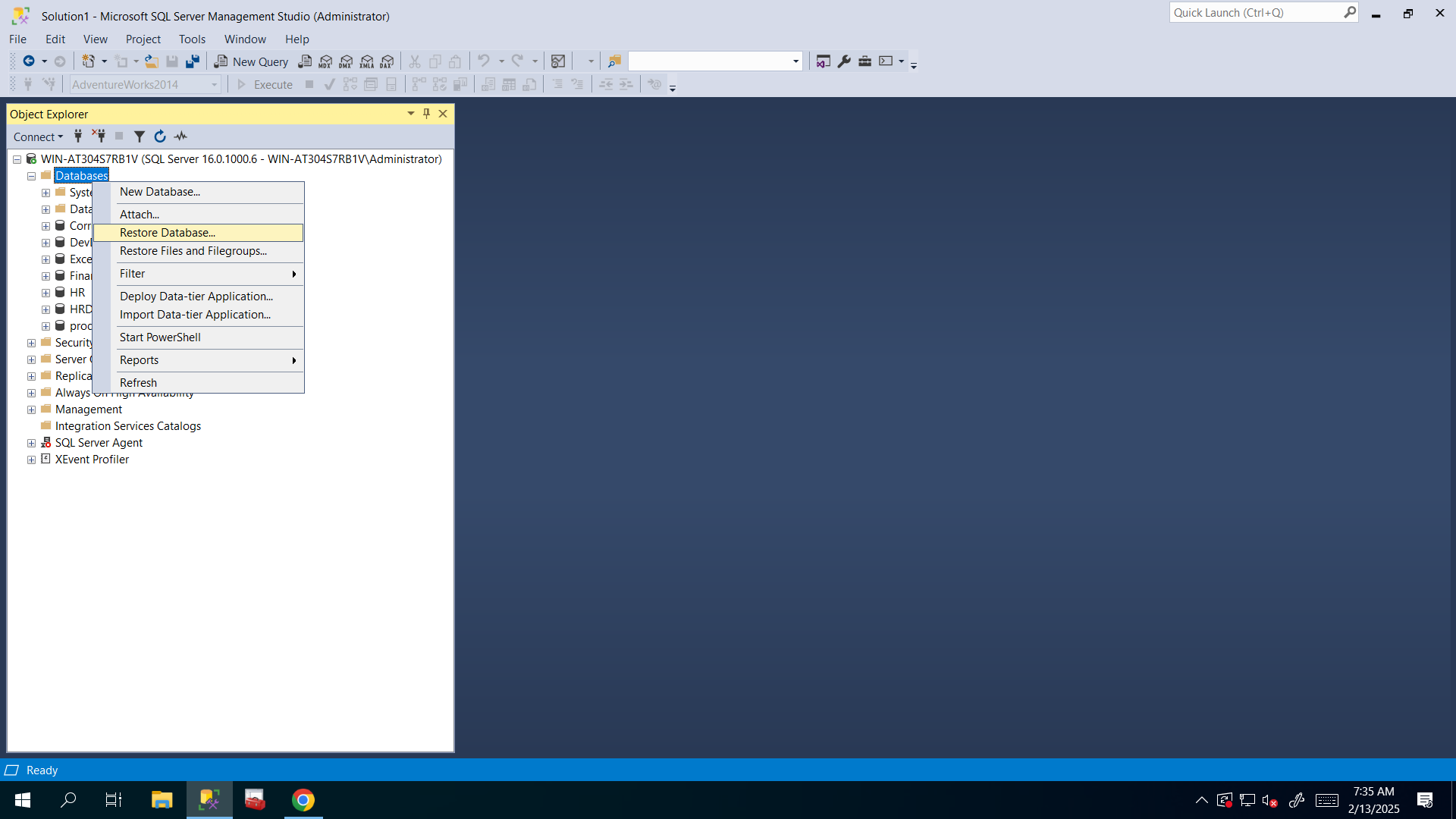
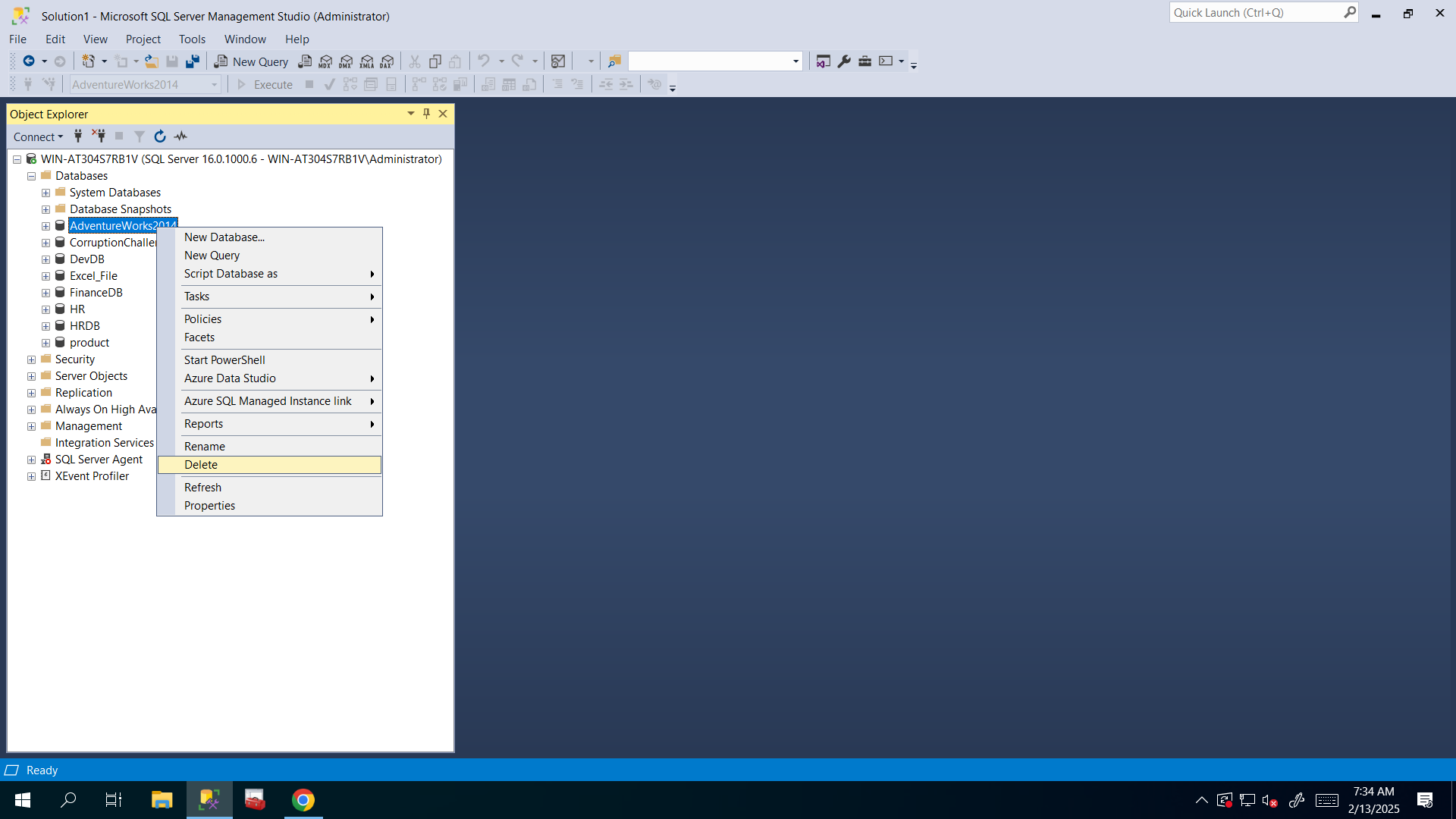
Recently you got emails from employees about slow performance while querying reports. While checking that database health, you found out that the table which employees try to query has highly fragmented indexes. Your plan is to find all indexes with at least 1000 pages which have index fragmentation higher than 5%. Decide which indexes must be rebuilt and which indexes must be reorganized. Finally, perform index maintenance via T-SQL.

Perform the following tasks in order to simulate this scenario:

1. **Remove AdventureWorks2014 database from your server. Restore the following modified database:**

<https://drive.google.com/file/d/1nPRKAY_L7vzugJysr8cGUWp1hYTufnDO/view?usp=sharing>

**I deleted AdventureWorks2014 database and restored it by using that backup file:**



1. Use the following query to find all fragmented indexes in that database. Then **make some changes to the code to filter indexes which contain at least 1000 pages and with index fragmentation higher than 5%.**

USE AdventureWorks2014

GO

SELECT

OBJECT\_SCHEMA\_NAME(b.OBJECT\_ID) AS SchemaName,

OBJECT\_NAME(b.OBJECT\_ID) AS TableName,

b.index\_id AS IndexID,

b.name AS IndexName,

a.avg\_fragmentation\_in\_percent AS PercentFragment,

a.fragment\_count AS TotalFrags,

a.avg\_fragment\_size\_in\_pages AS PagesPerFrag,

a.page\_count AS NumPages,

8 \* SUM(c.used\_pages) AS 'Indexsize(KB)'

FROM sys.dm\_db\_index\_physical\_stats(DB\_ID(), NULL, NULL, NULL, NULL) AS a

JOIN sys.indexes AS b ON a.object\_id = b.object\_id

JOIN sys.partitions AS p ON p.OBJECT\_ID = b.OBJECT\_ID AND p.index\_id = b.index\_id

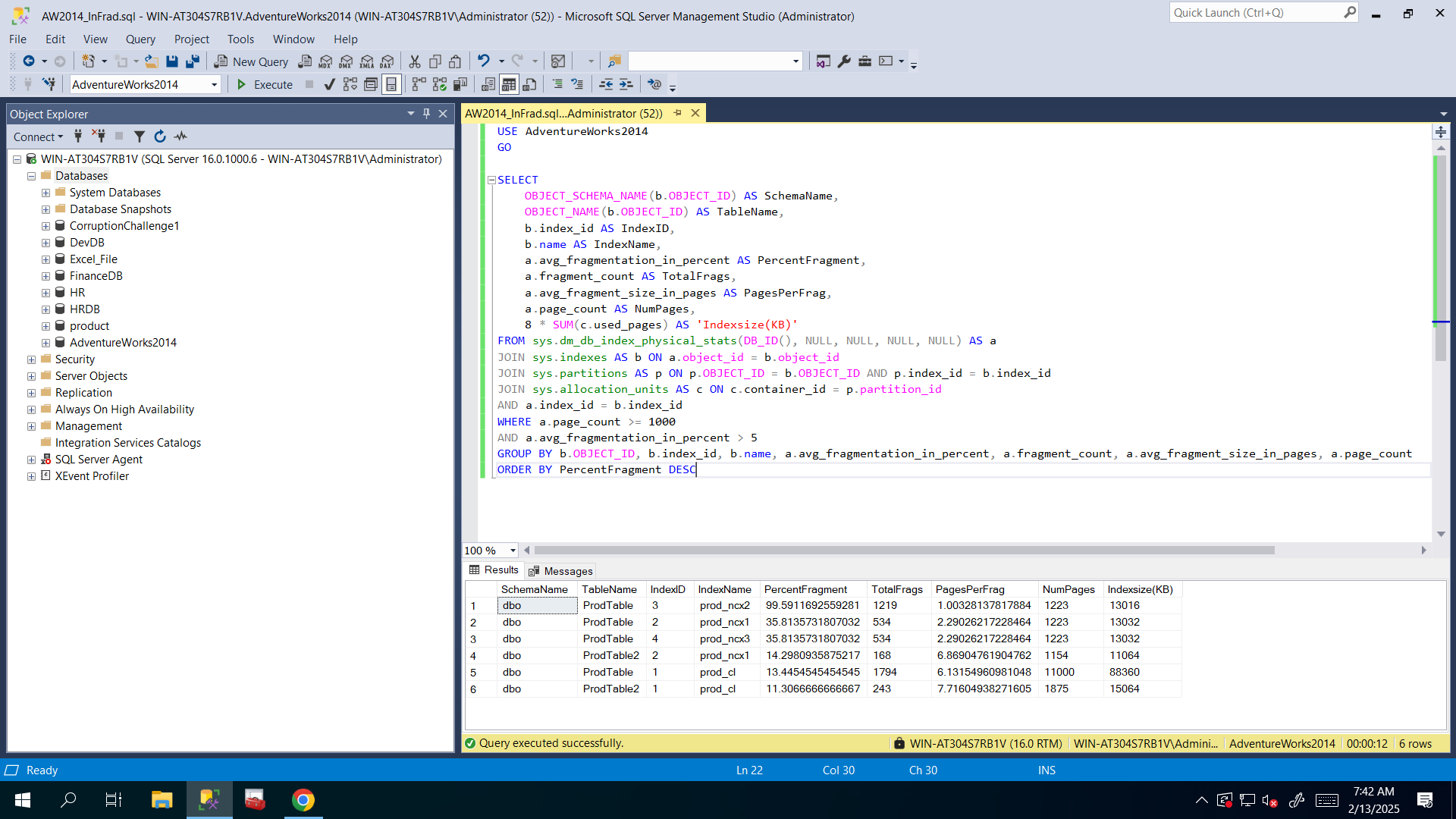
JOIN sys.allocation\_units AS c ON c.container\_id = p.partition\_id

AND a.index\_id = b.index\_id

GROUP BY b.OBJECT\_ID,b.index\_id,b.name, a.avg\_fragmentation\_in\_percent, a.fragment\_count, a.avg\_fragment\_size\_in\_pages,a.page\_count

ORDER BY PercentFragment DESC

**I just added 2 lines of code and used WHERE clause to filter out needed indexes:**



1. Which indexes should be rebuilt?

**I think first three ones should be rebuilt: prod\_ncx1, prod\_ncx2 and prod\_ncx3 indexes from dbo schema and ProdTable because their fragmentation percentage is more than 30%.**

1. Which indexes should be reorganized?

**I think last three ones should be reorganized: prod\_ncx1, prod\_cl indexes from dbo schema and ProdTable table and prod\_cl index from dbo schema and ProdTable2 table because their fragmentation percentage is less than 30% and more than 5% (I filtered like that).**

1. Before running index rebuild/reorganize operations pay attention to log file size using the following query:

USE AdventureWorks2014

GO

SELECT DB\_NAME() AS DbName,

name AS FileName,

type\_desc,

size/128.0 AS CurrentSizeMB,

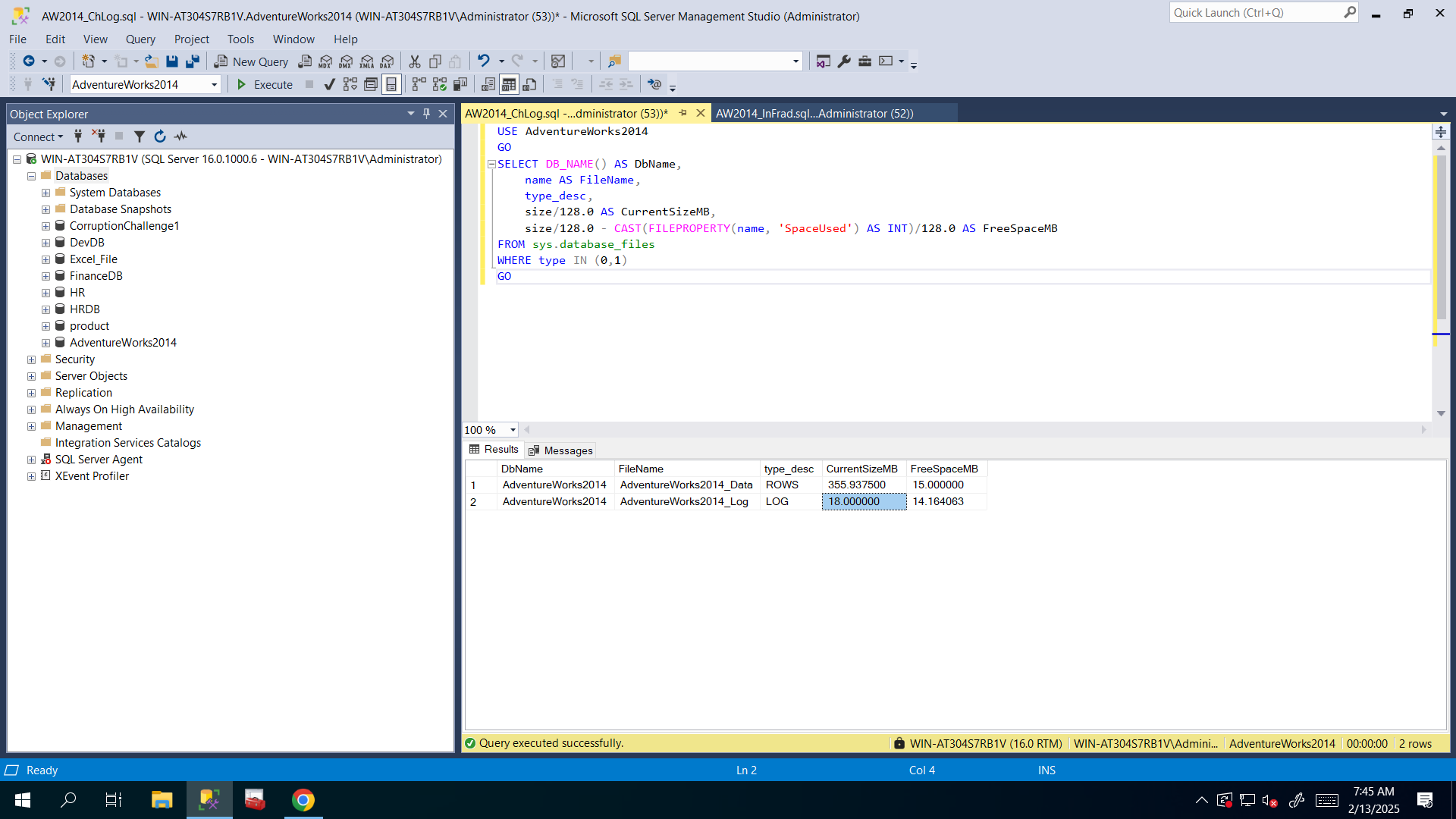
size/128.0 - CAST(FILEPROPERTY(name, 'SpaceUsed') AS INT)/128.0 AS FreeSpaceMB

FROM sys.database\_files

WHERE type IN (0,1)

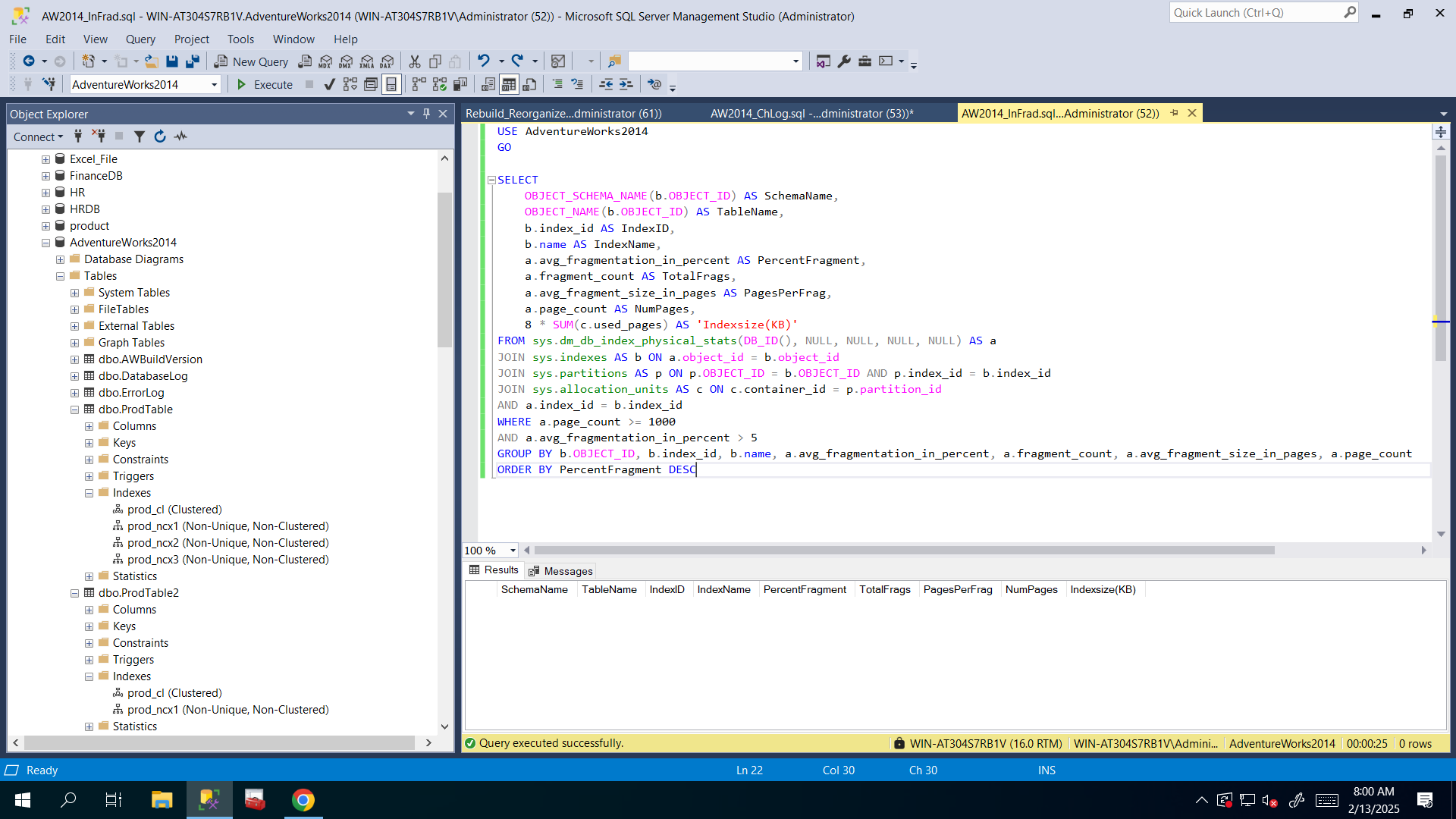
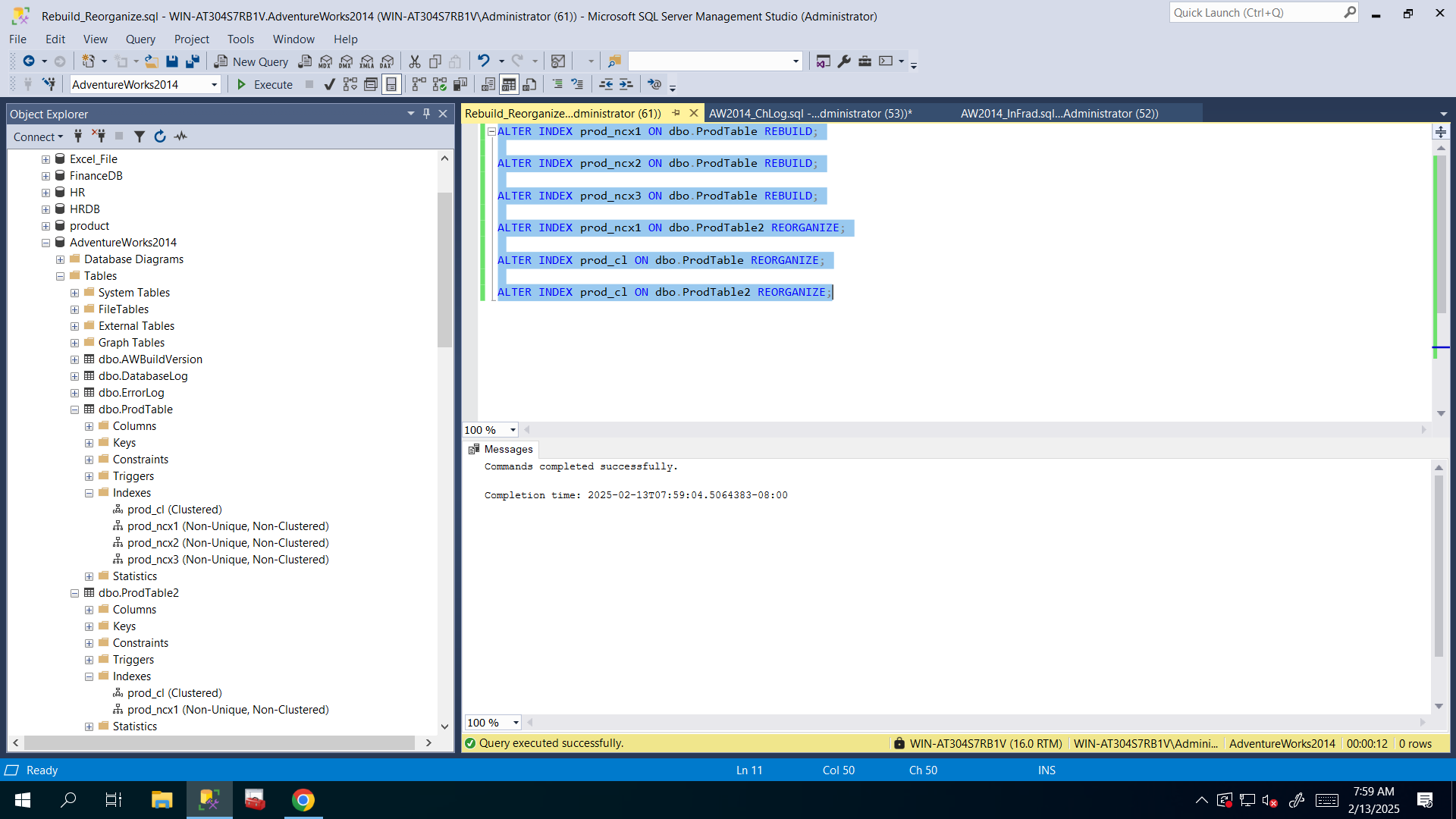
GO

**I executed above query and got that Log file’s size is 18MB**:



1. Perform index rebuild and index reorganize operations. Then re-run the above query to make sure that no fragmented indexes left in the database.

**I used structure “ALTER INDEX <index\_name> ON <schema\_name>.<table\_name> REBUILD/REORGANIZE” and executed that query to perform rebuild or reorganize operations:**



1. Run the blow query again and check log file size:

USE AdventureWorks2014

GO

SELECT DB\_NAME() AS DbName,

name AS FileName,

type\_desc,

size/128.0 AS CurrentSizeMB,

size/128.0 - CAST(FILEPROPERTY(name, 'SpaceUsed') AS INT)/128.0 AS FreeSpaceMB

FROM sys.database\_files

WHERE type IN (0,1)

GO

**Now I checked again Log file’s size and it’s really a lot increased and now it’s 290MB rather than 18MB because of these rebuild and reorganize operations(modifications, changes, and etc.):**

