**‘AccessBank DBA Internship Program’**

**Project 11: Creating indexes on a table or how to tune query**

While monitoring Database Server, you found 2 queries which are very expensive. Try to tune those queries with the help of indexes.

SELECT [first\_name]

FROM [HR].[dbo].[employees]

ORDER BY [first\_name] asc

1. **sort operation (**ORDER BY [first\_name] asc**) is an expensive operation. You need to create an index which will avoid using** ORDER BY [first\_name] asc **therefore making that query less expensive.**

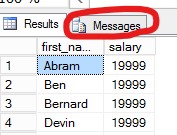
SELECT [first\_name],[salary]

FROM [HR].[dbo].[employees]

WHERE [salary] < 60000

1. **This query consumes a lot of memory. You can find out by running the following command in your current session and look at messages tab:**

**SET STATISTICS IO ON**

****

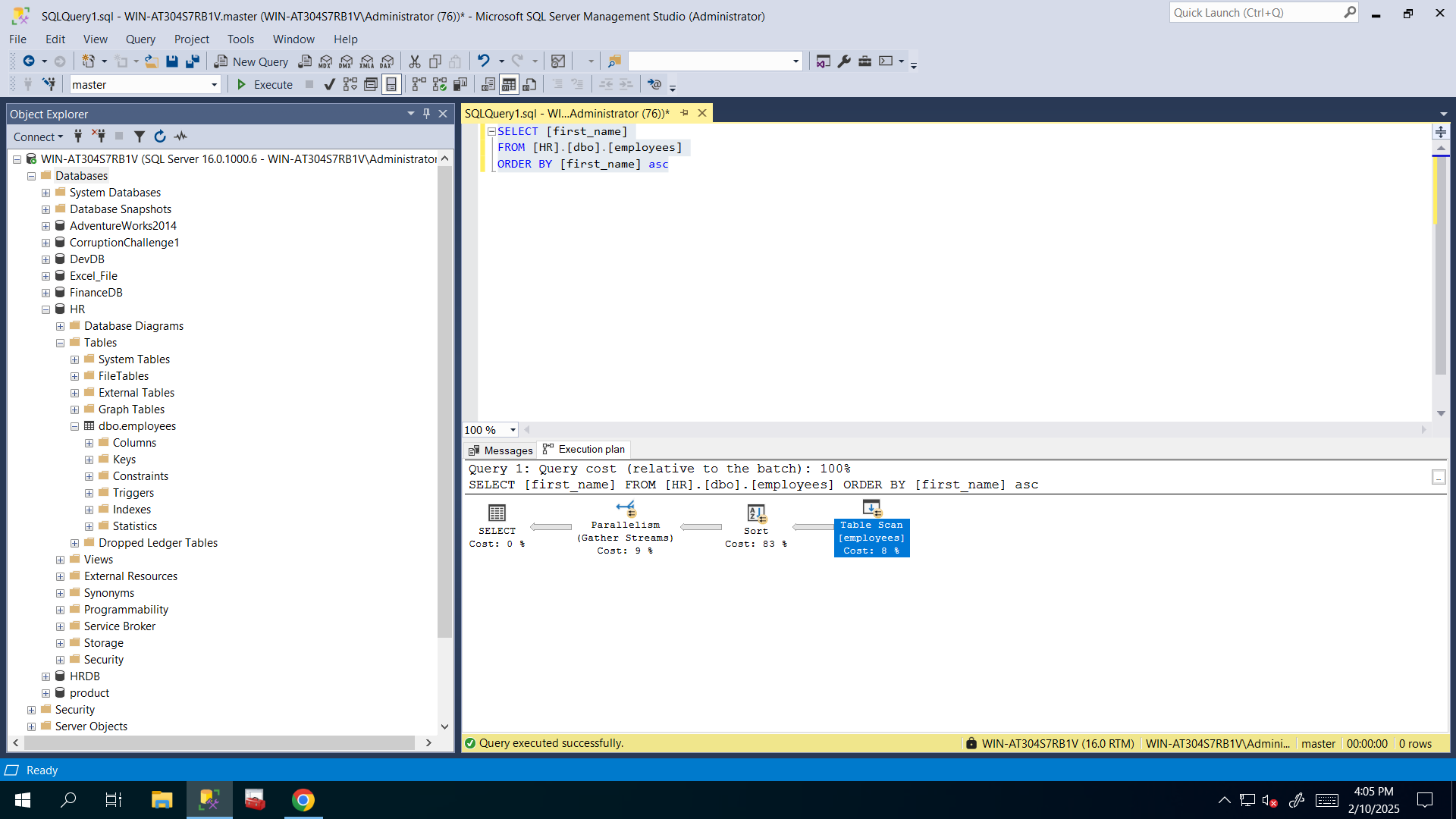
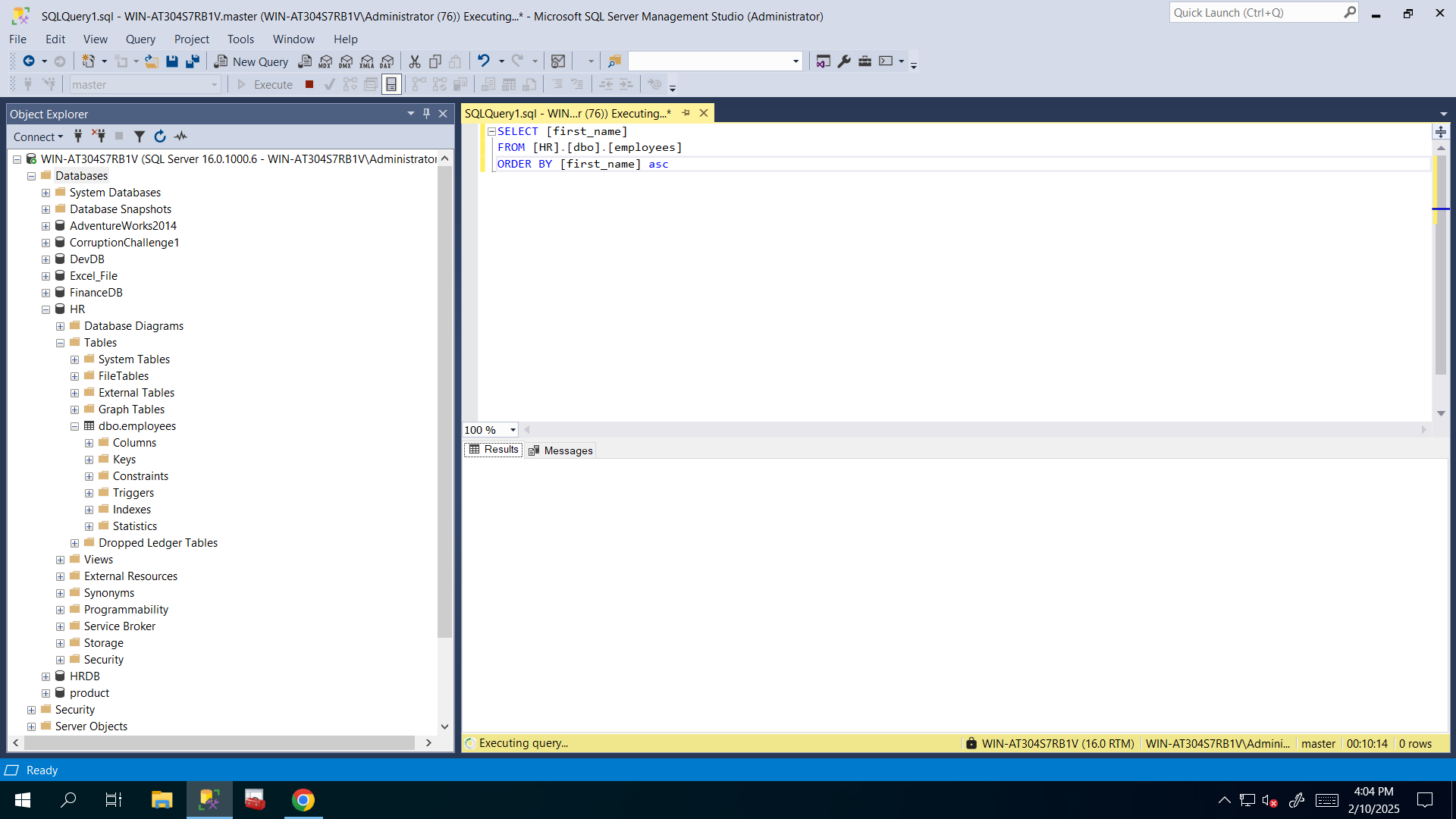
**As you can see 14575 pages were read from memory which means 14575 pages \* 8 KB (each page) = 116600 KB~114 MB.**

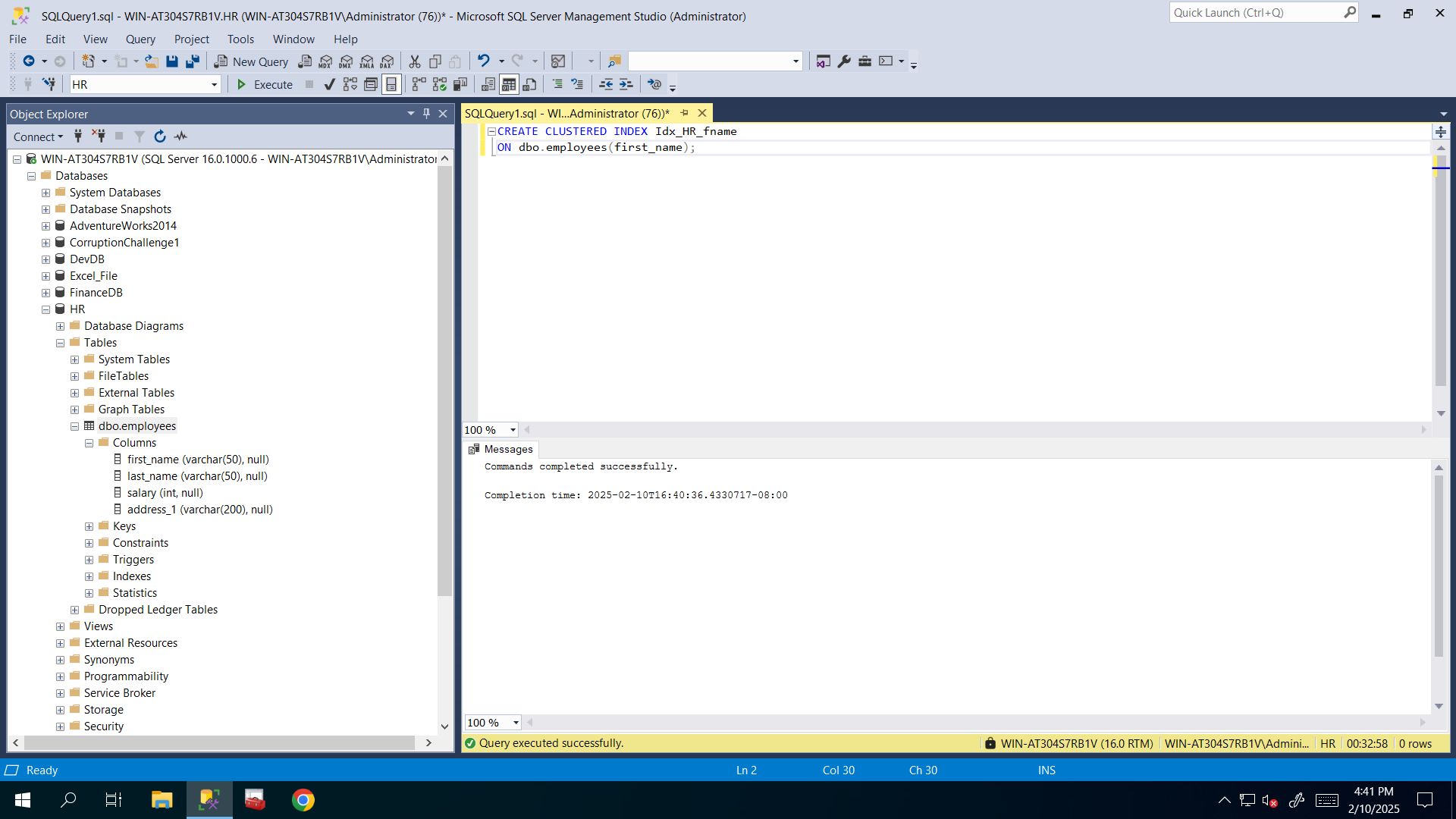
**You need to create a nonclustered index which will make this query read less data from memory and by this way saving space for other data.**

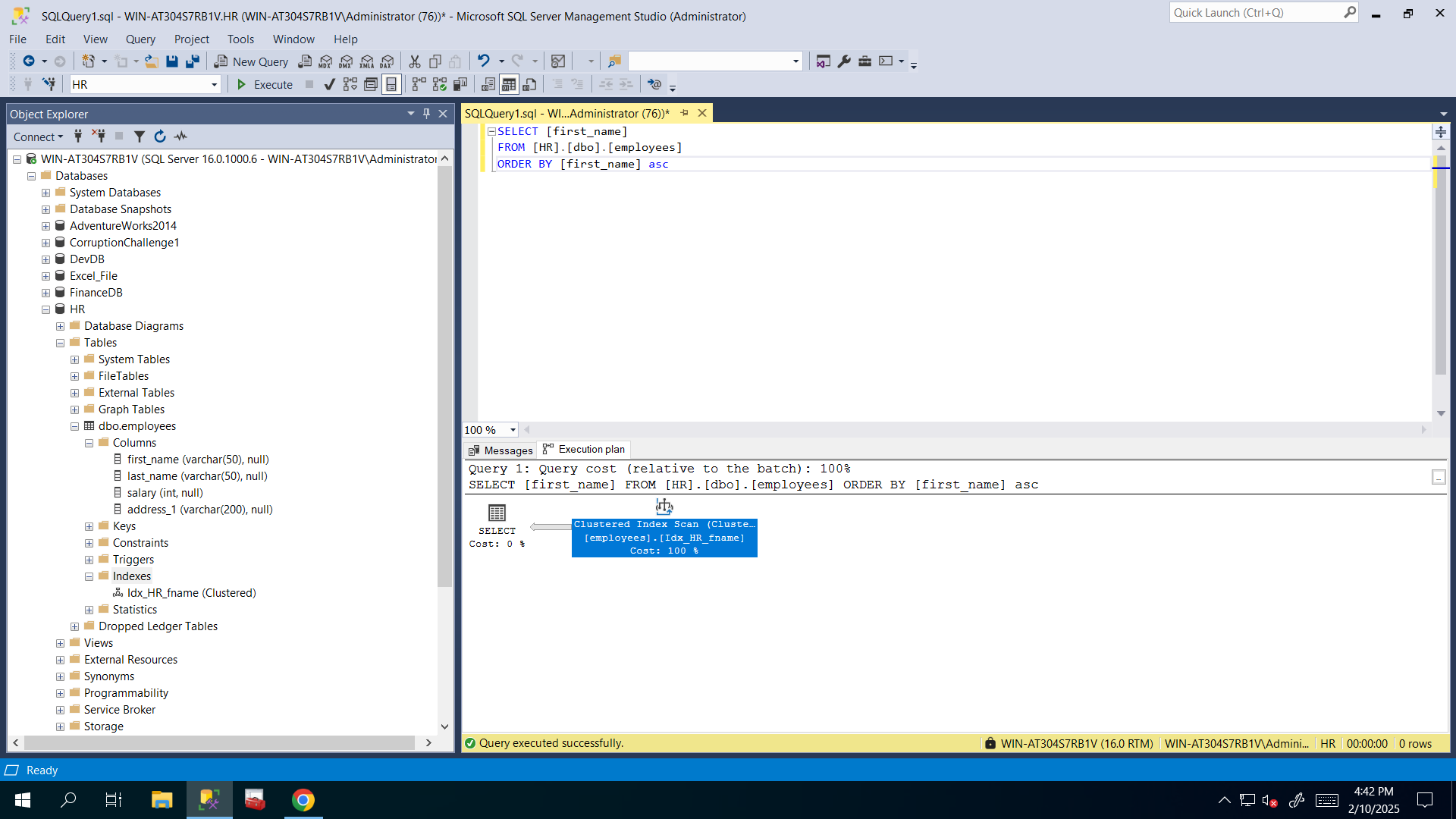
Note:

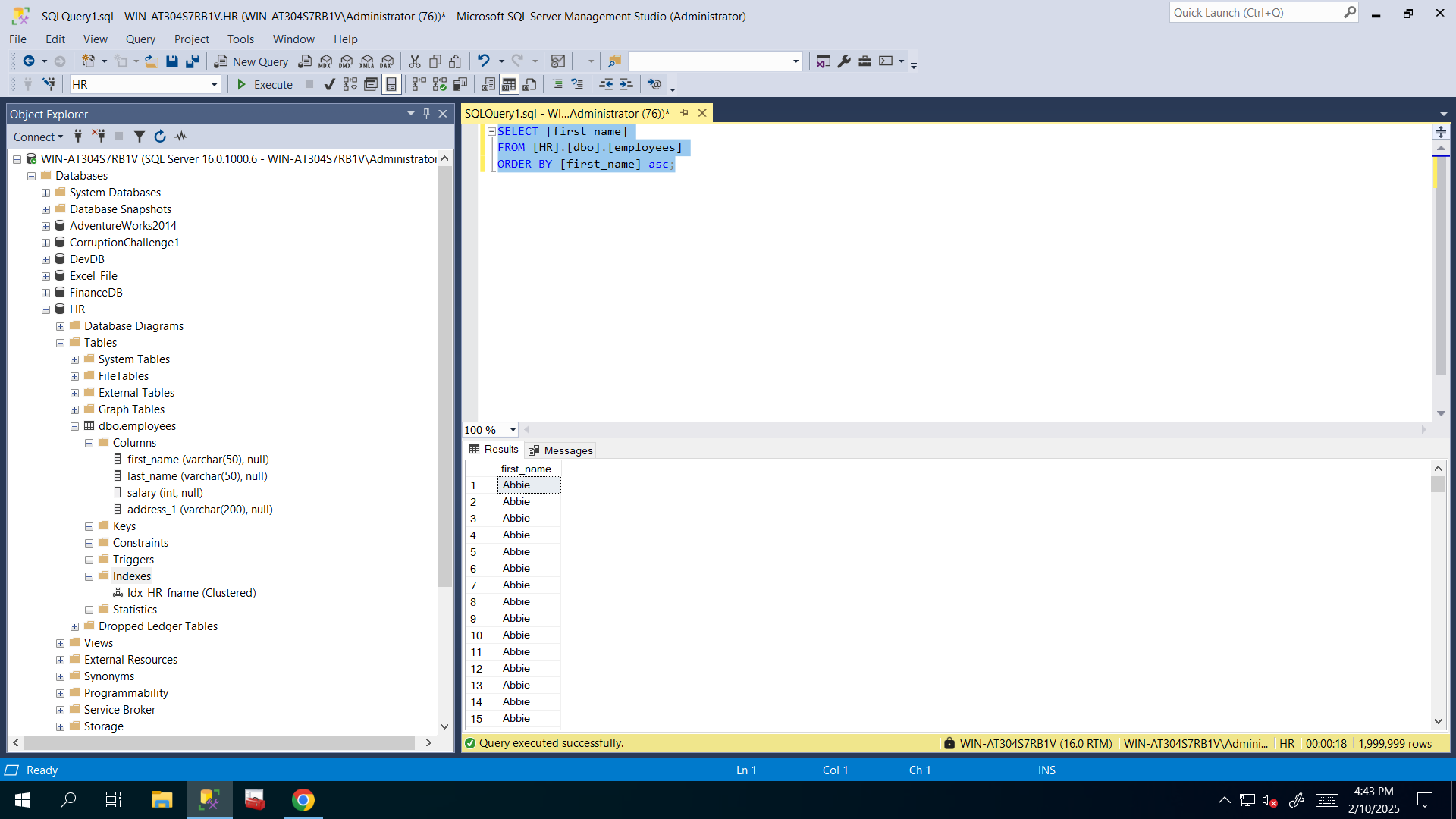
SQL Data Generator

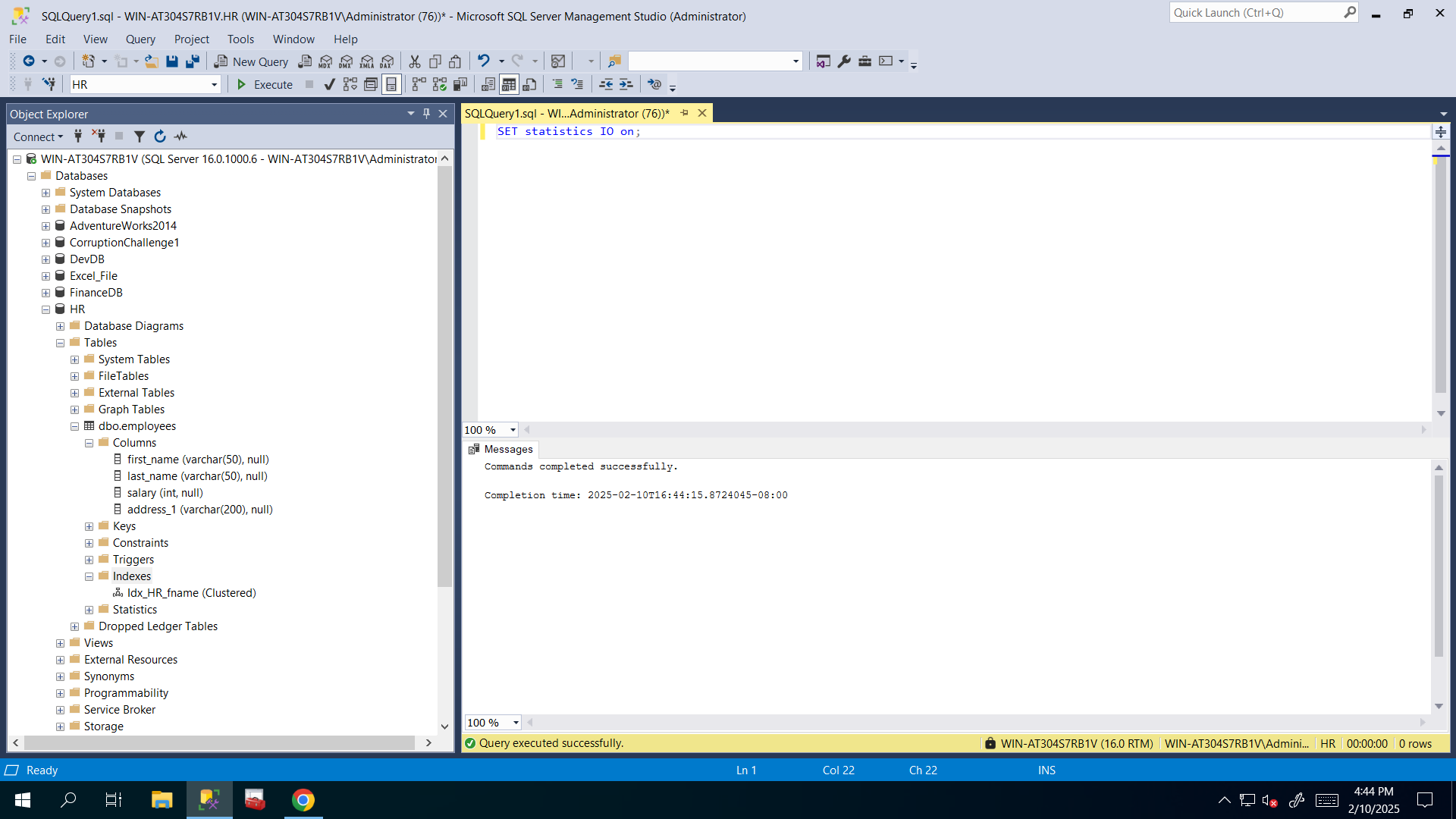
<https://sql-data-generator.software.informer.com/3.1/>

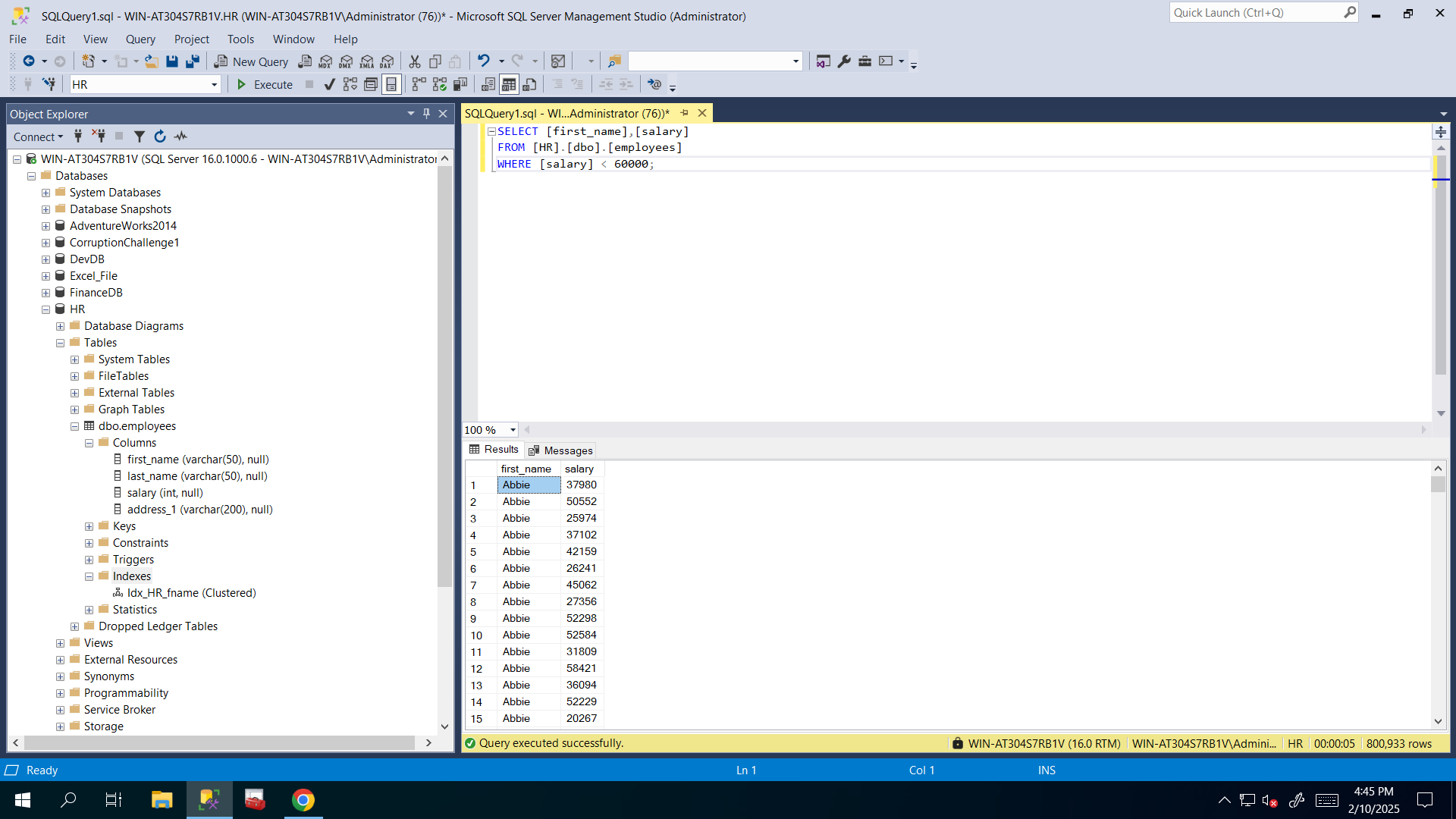
**In first query, without index it takes over 10 minutes to execute that query which means it’s really expensive so I need to create index to tune that query and I used Display Estimated Execution plan to see it really uses Table Scan**:

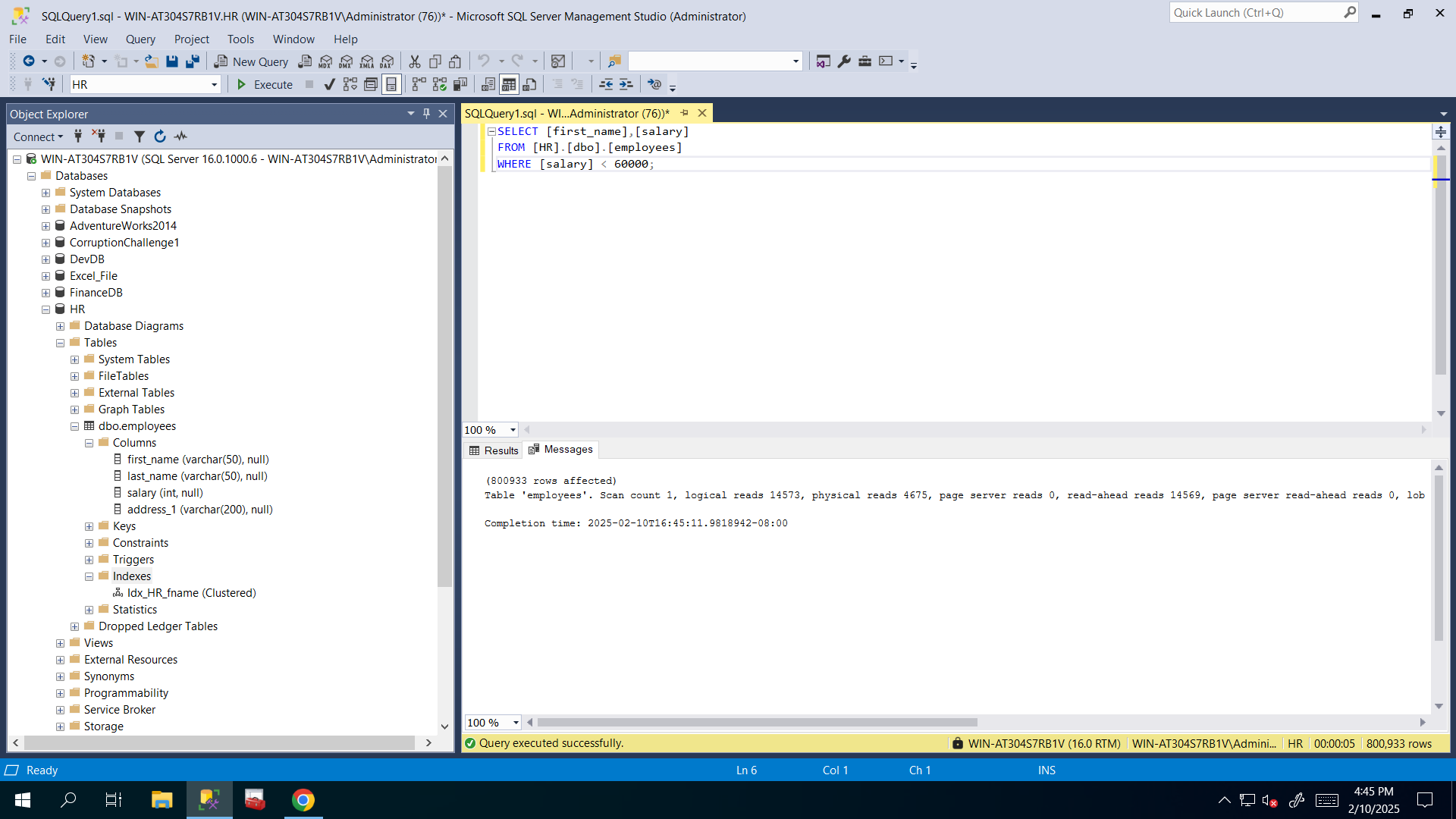
**So I created clustered index for that column which is used in Order BY clause and why I did choose that? Because every table can have maximum one clustered index and clustered index is more beneficial, efficient and fast in Order By statements if that column is used to create clustered index**:

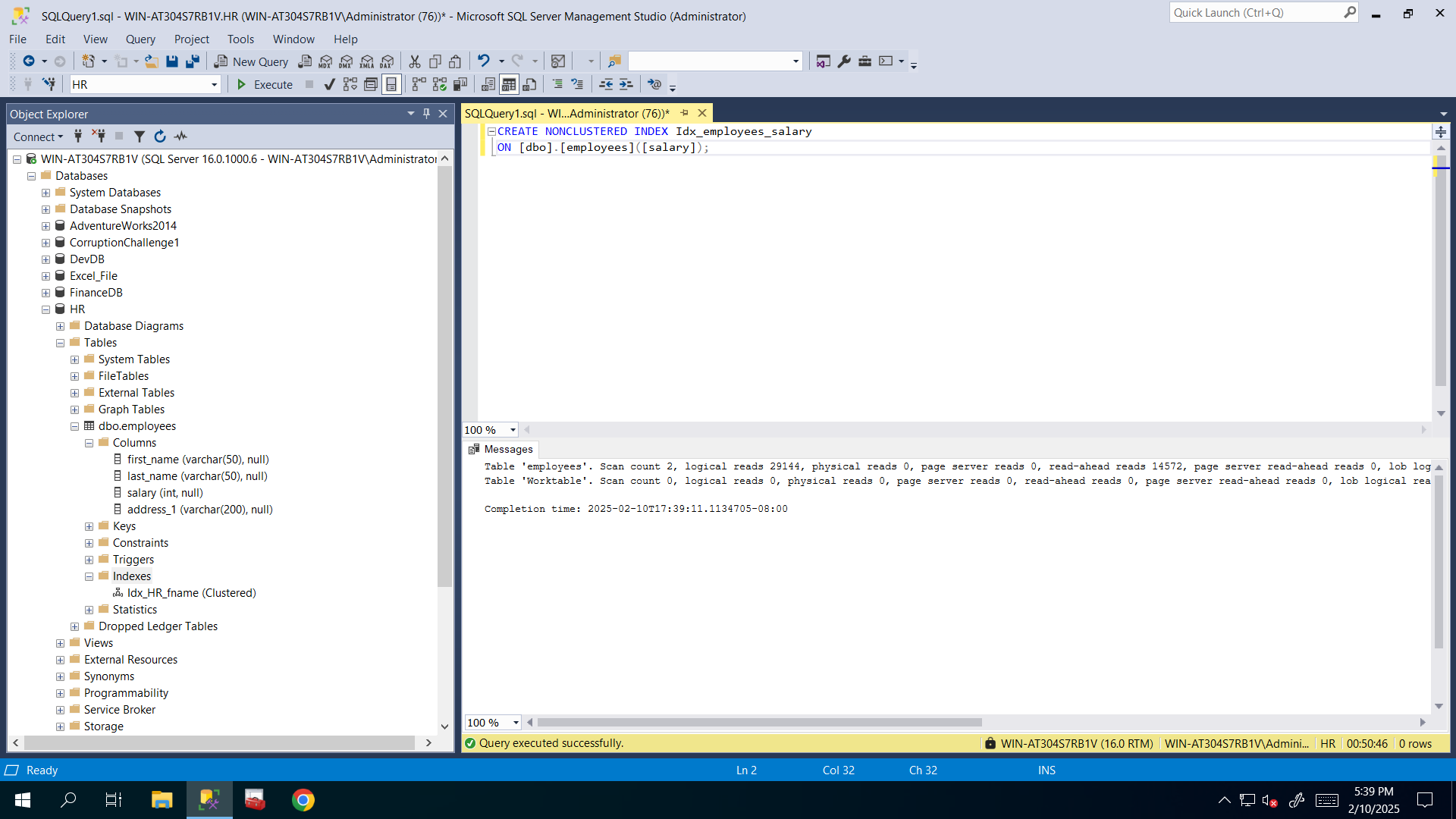
**And again by using Display Estimated Execution plan I checked to see, whether that query is executed by using clustered index or not**:

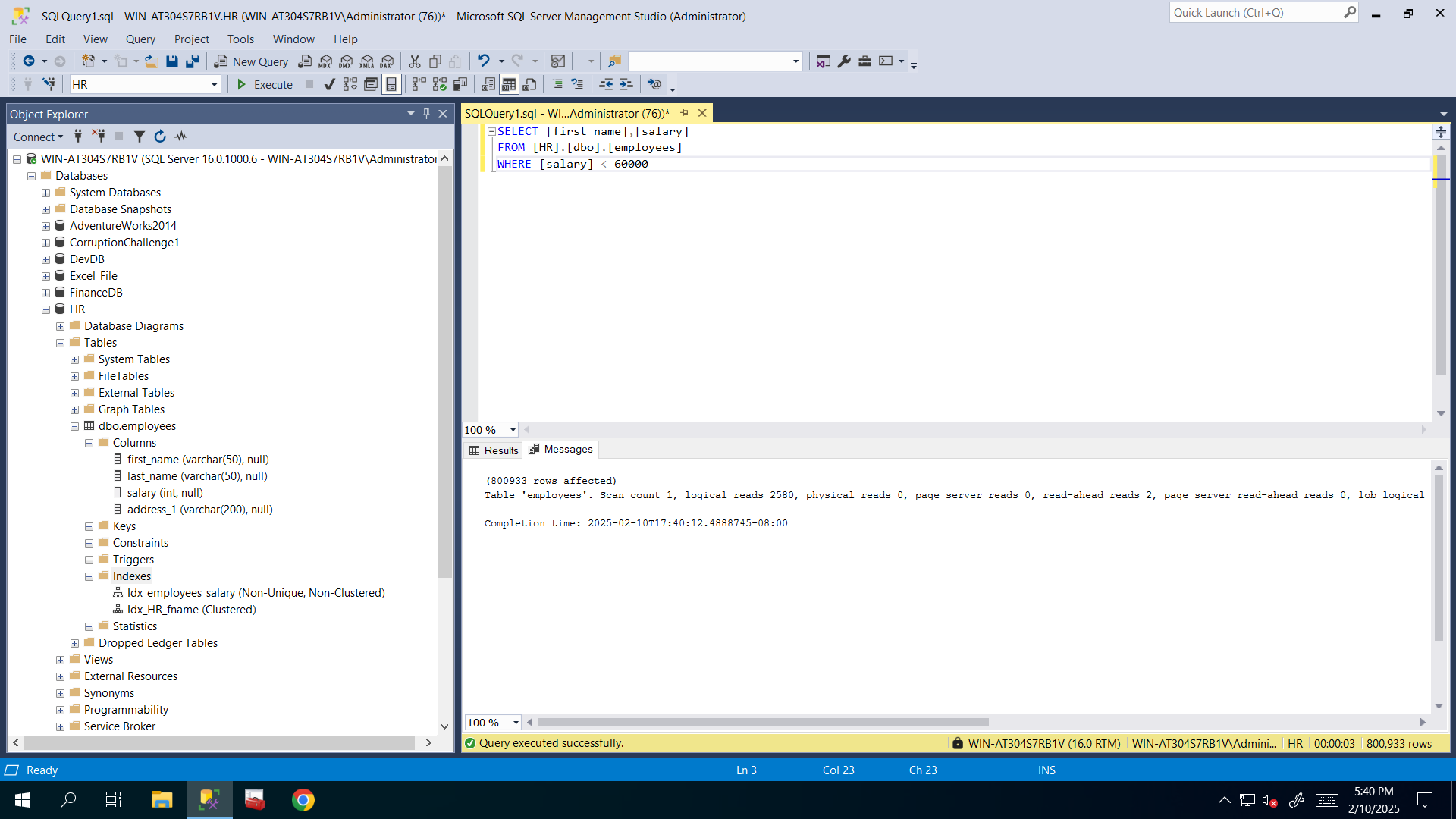
**And it took just 18 seconds to execute that query**:

**Now let’s run the following command to see how many pages were read in second query**:

**Let’s run the second query to see if it’s true**:

**And as you can see it really reads 14573 pages that’s a lot of memory so I need to create nonclustered index on that salary column**:

**So I created nonclustered index on salary column by using following query and I called it Idx\_employees\_salary for naming conventions (prefix+table name + column name):**

**Now it just reads 2580 pages rather than 14573:**