--7. Calculate the total bill for booking 5128 including extras.

Test 1

CREATE OR ALTER VIEW vTotalBill5128

AS

SELECT SUM (rate.amount\*b.nights) +

(SELECT SUM(e.amount)

FROM extra e

WHERE e.booking\_id=b.booking\_id) AS "total bill for booking 5128"

FROM rate INNER JOIN booking b

ON b.occupants=rate.occupancy AND b.room\_type\_requested=rate.room\_type

WHERE b.booking\_id=5346

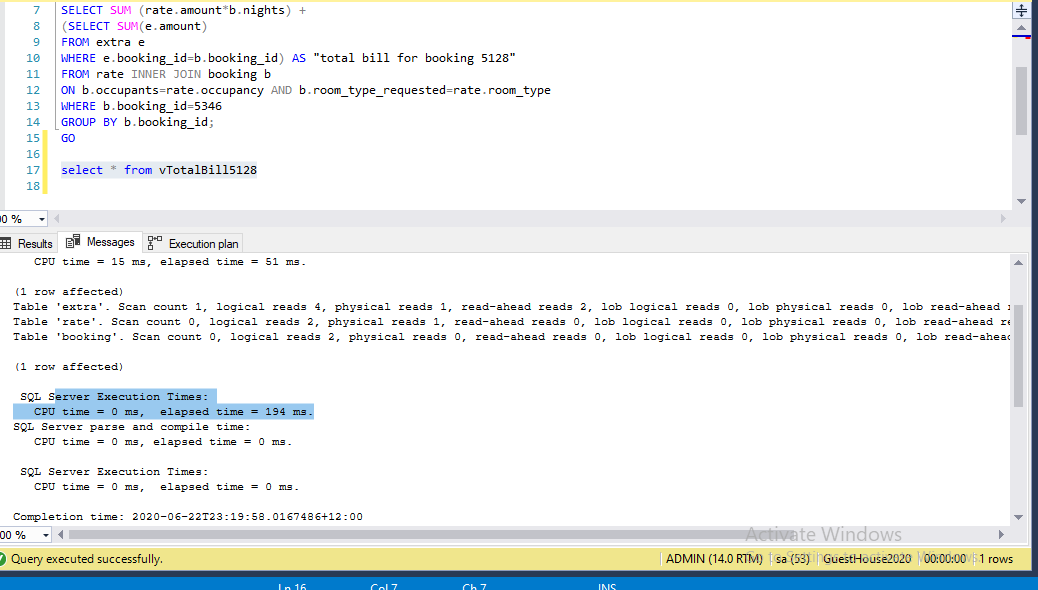
GROUP BY b.booking\_id;

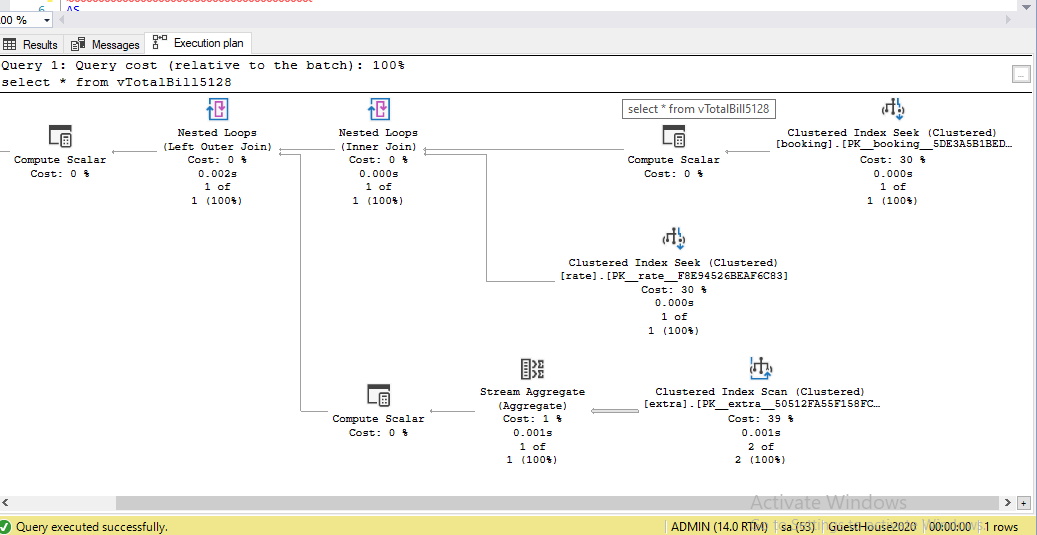
GO

select \* from vTotalBill5128

SQL Server Execution Times:

CPU time = 0 ms, elapsed time = 194 ms





TEST 2 : create nonclustered index as below . This make significant change in execution time : elapsed time drop from 194 s to 68s

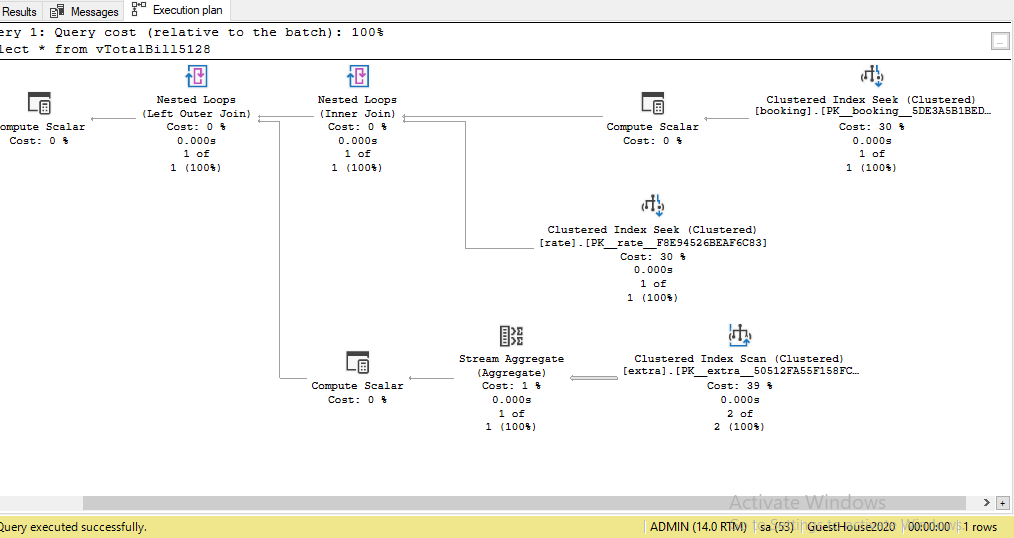
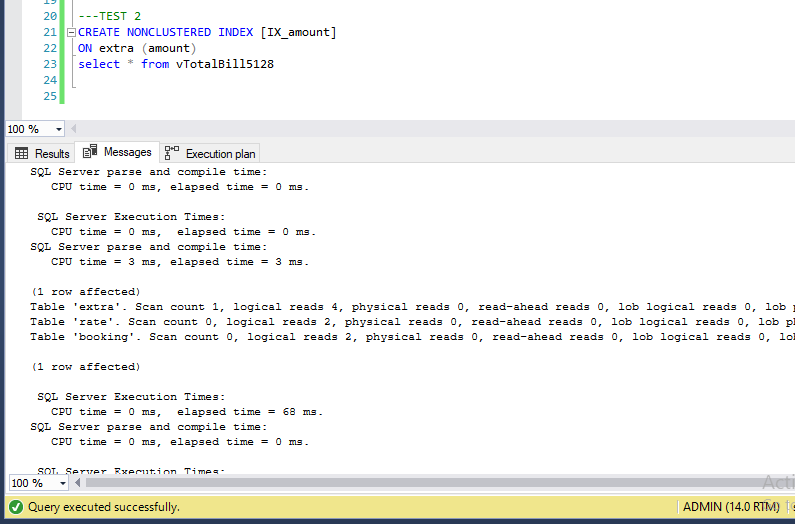
CREATE NONCLUSTERED INDEX [IX\_amount]

ON extra (amount)

select \* from vTotalBill5128

SQL Server Execution Times:

CPU time = 0 ms, elapsed time = 68 ms.



CONCLUSION

In query plan 1 (test 1 ) , It is very good that SQL use clustered index seek to scan table rate and booking, however , we can see the execution plan use clustered index to scan on table extra. This lead to good time of execution : 194s

In query plan 2 (test 2), I created nonclustered index for extra table, with index key is amount (as showing above), this make execution time drop significantly to more than half time of the first test. So, query plan 2 is an effective way.