--TEST 1: we can see below Execution query plan use clustered index scan in to table rate and booking.

CREATE OR ALTER VIEW vBillDrAndrew

AS

SELECT SUM(amount\*nights) AS "total amount"

FROM guest INNER JOIN booking b ON guest.id=b.guest\_id

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

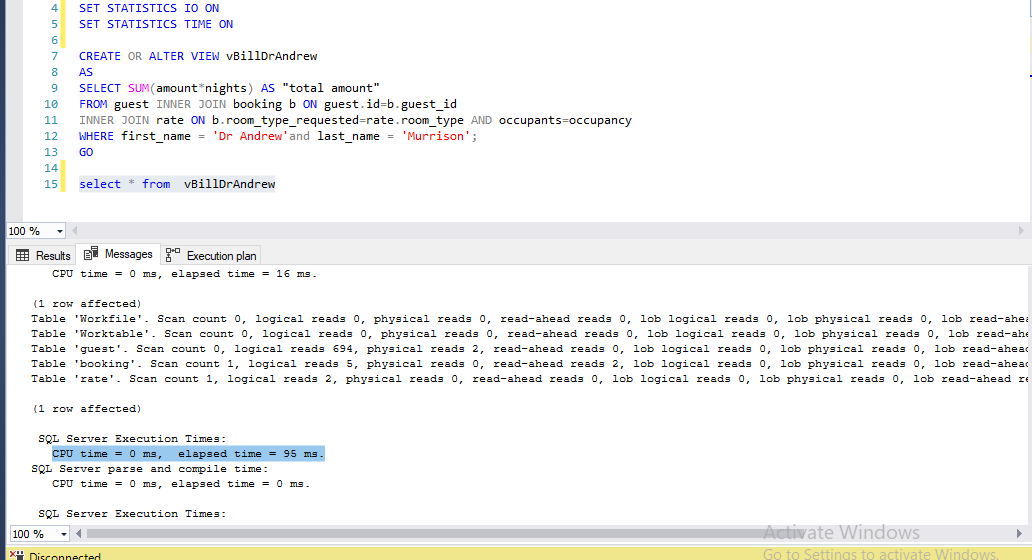
WHERE first\_name = 'Dr Andrew'and last\_name = 'Murrison';

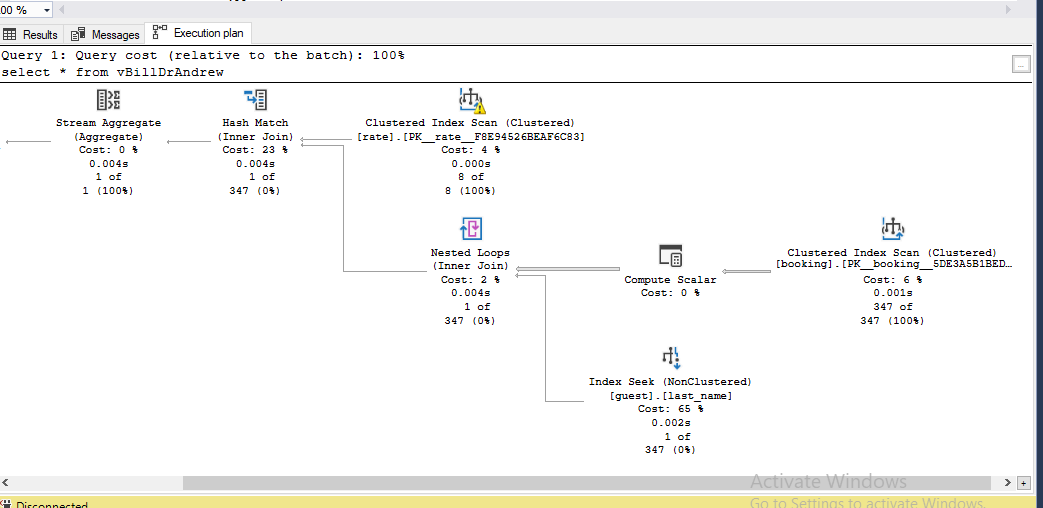
GO

select \* from vBillDrAndrew

SQL Server Execution Times:

CPU time = 0 ms, elapsed time = 95 ms.





--TEST 2 : CREATE NONCLUSTERED INDEX for table booking and rate

CREATE NONCLUSTERED INDEX IX\_nights

ON booking (nights, room\_type\_requested)

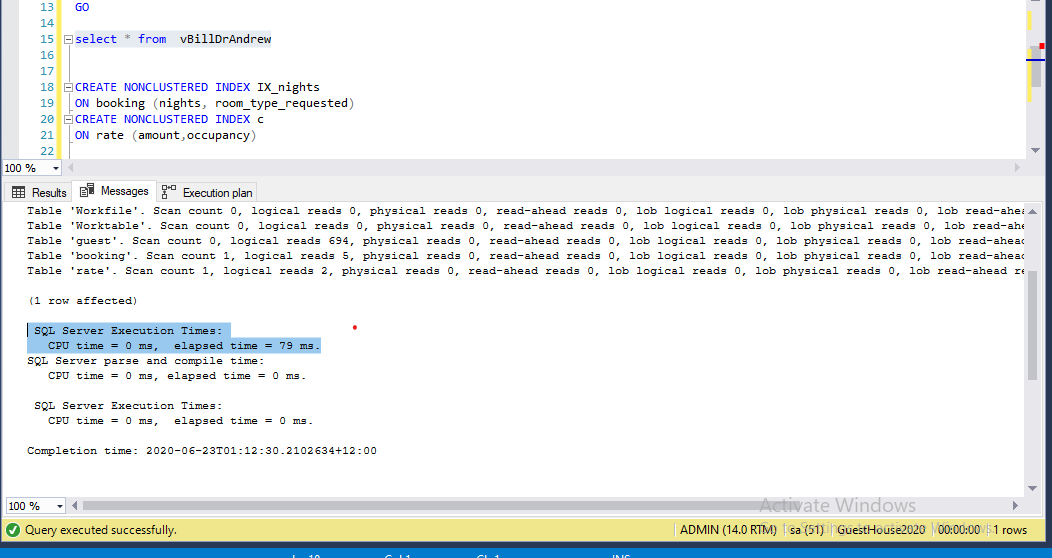
CREATE NONCLUSTERED INDEX c

ON rate (amount,occupancy)

select \* from vBillDrAndrew

SQL Server Execution Times:

CPU time = 0 ms, elapsed time = 79 ms.



CONCLUSION:

In the first test, we can see execution plan using index seek on table cost which is a efficient way in performing query, but it is using clustered index scan for both booking and rate table.

In the 2nd tes, I created non clustered index in both table booking and rate. At the result, the execution time drops from 95s to 79s

So, in this case, creating non clustered index (as above ) is an effective way to make a good query performance.