

Nr	SQL	Första körning	Andra körning	Kommentar och jämförelse
1	SELECT * FROM dbo.Medlem	21689 ms	20751 ms	
2	SELECT ENAMN, FNAMN, POSTNR, ORT FROM dbo.Medlem	5326 ms	4750 ms	Betydligt snabbare än första queryn då det är många kolumner och värden som inte tas med.
3	SELECT * FROM dbo.Medlem ORDER BY ENAMN	24106 ms	23373 ms	Clustered Index Scan tar 2% av operator cost, och sort 98%. Sortering gör att det tar mycket längre tid att få ut precis samma värden som i test 1.
4	SELECT * FROM dbo.Medlem ORDER BY ENAMN, FNAMN	24667 ms	24041 ms	Samma som föregående, men nu ännu lite längre tid.
5	SELECT ENAMN, FNAMN, POSTNR, ORT FROM dbo.Medlem ORDER BY ENAMN	6692 ms	6427 ms	Clustered Index Scan tar 35% av operator cost, och sort 65%. Tidigare tog Clustered Index Scan 100%
6	SELECT ENAMN, FNAMN, POSTNR, ORT FROM dbo.Medlem ORDER BY ENAMN, FNAMN	6167 ms	6041 ms	Execution plan visar samma resultat som föregående test. Det tog lite kortare tid att genomföra denna query, förmodligen eftersom programmet redan visat samma fält sorterade efter enamn precis innan och lagrat detta.
7	SELECT * FROM dbo.Medlem WHERE ORT = 'Kalmar'	214 ms	208 ms	Jag trodde först att jag gjort något fel när det gick så fort jämfört med alla tidigare tester. Men det var ändå inte fullt så många poster som matchade heller.
8	SELECT * FROM dbo.Medlem WHERE ORT LIKE 'Kal%'	268 ms	245 ms	Jämfört med test 7 blev denna långsammare, dock finns det fler poster som matchar.
9	SELECT * FROM dbo.Medlem WHERE ORT LIKE 'Kal%' AND ENAMN LIKE '%son%' ORDER BY Fastnr	175 ms	165 ms	Sort är 4% av operator cost, och Clustered Index Scan är 96%. Operationen tar kortare tid än test 8 och 7 som är liknande, men det är förmodligen bara för att de testerna har körts innan och svaren lagrats.
10	SELECT * FROM dbo.Medlem WHERE ANNDATUM >= '20100101'	74 ms	69 ms	Jag antar bara att året som menas är 2010, då det är det sista året där anndatum är specificerat.
11	SELECT COUNT(ANNDATUM), ORT FROM dbo.Medlem WHERE ANNDATUM > '0' GROUP BY ORT	160 ms	157 ms	Aggregatfunktionen tar 4% av operator cost för att räkna ihop fälten i det här fallet.
12	SELECT ORT FROM dbo.Medlem GROUP BY ORT ORDER BY ORT	361 ms	364 ms	Aggregatfunktionen tar 12% av operator cost för att bara ta med ortnamnet en gång. Operationen tog lite längre tid andra gången men det är så fruktansvärt lite så det är egentligen samma tid.
13	SELECT * FROM dbo.Medlem WHERE Telenr1 != Telenr2	18743 ms	18981 ms	Det tog lite längre tid att köra queryn andra gången på detta testet med, och jag har ingen logisk anledning varför.