CREATE NONCLUSTERED INDEX IX\_Cities

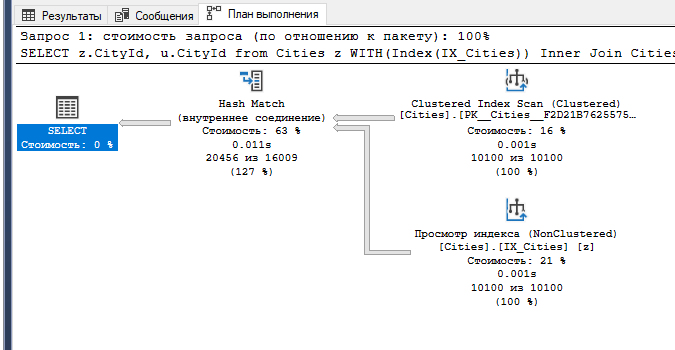
ON Cities (CountryId)

INCLUDE (CityName)

GO

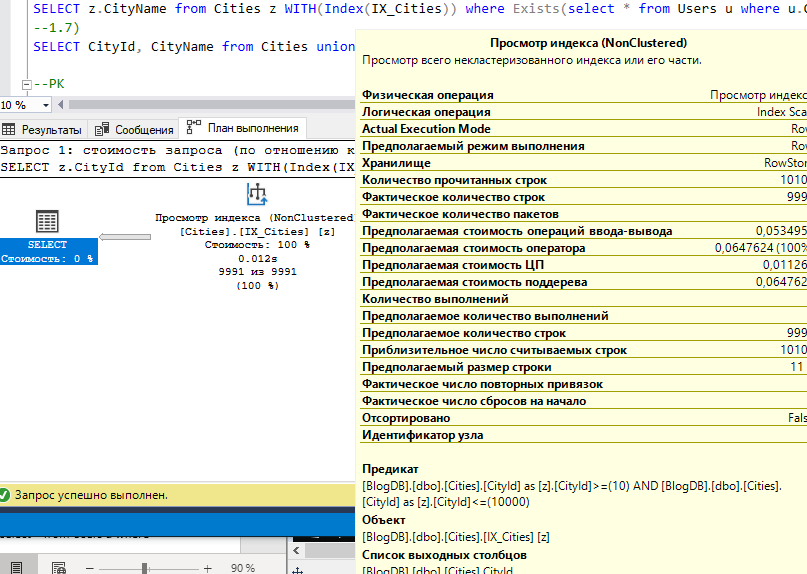
--1.1)

SELECT z.CityId, u.CityId from Cities z WITH(Index(IX\_Cities)) Inner Join Cities u on u.CountryId = z.CountryId



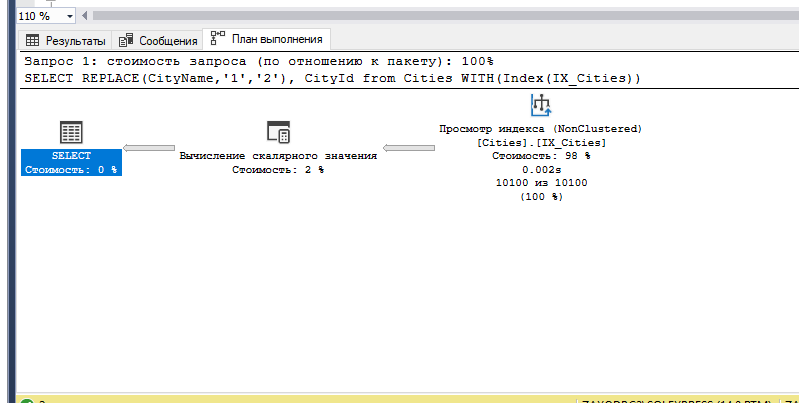
--1.2)

SELECT z.CityId from Cities z WITH(Index(IX\_Cities)) where CityId BETWEEN 10 AND 10000



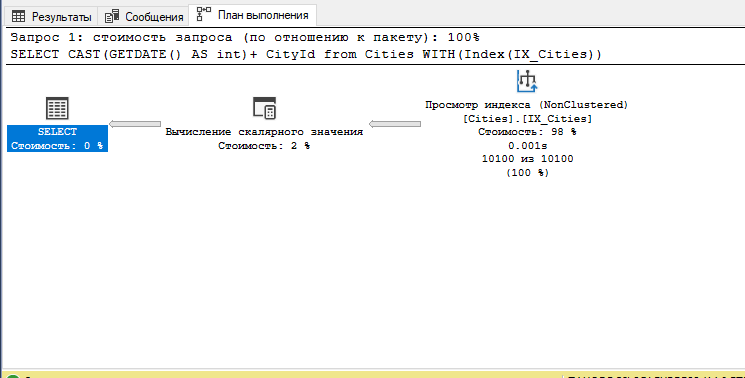
--1.3)

SELECT REPLACE(CityName,'1','2'), CityId from Cities WITH(Index(IX\_Cities))



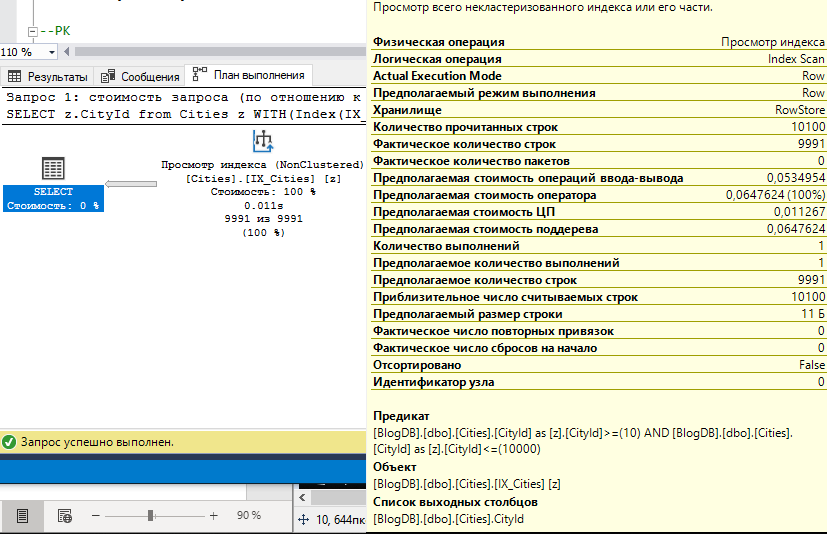
--1.4)

SELECT CAST(GETDATE() AS int)+ CityId from Cities WITH(Index(IX\_Cities))



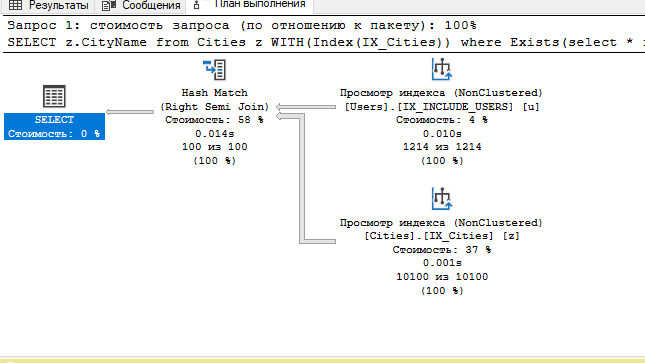
--1.5)

SELECT z.CityId from Cities z WITH(Index(IX\_Cities)) Group By CityId Having z.CityId BETWEEN 10 AND 10000



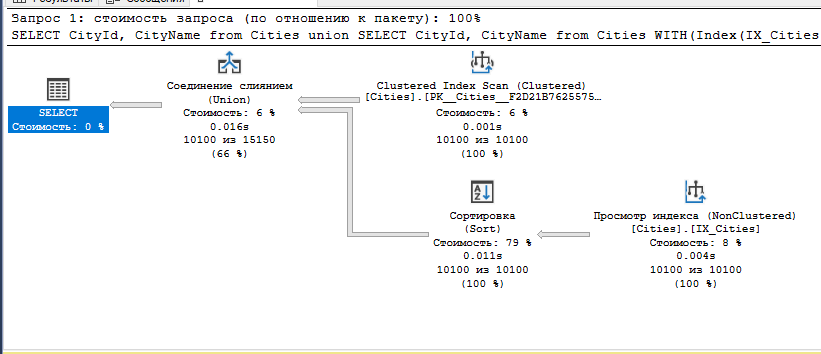
--1.6)

SELECT z.CityName from Cities z WITH(Index(IX\_Cities)) where Exists(select \* from Users u where u.CityId = z.CityId)



--1.7)

SELECT CityId, CityName from Cities union SELECT CityId, CityName from Cities WITH(Index(IX\_Cities))

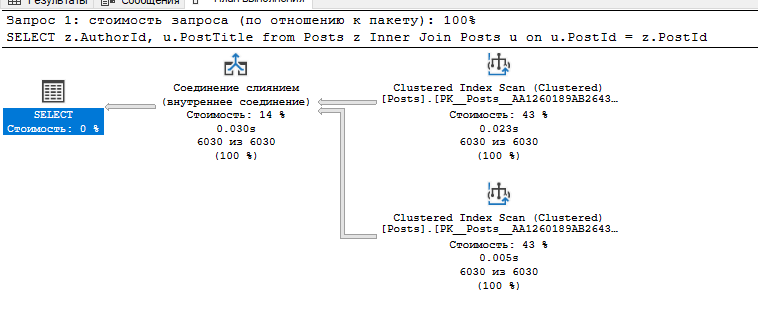


PostId INT IDENTITY(1,1) NOT NULL PRIMARY KEY,

Автоматически сгенерирует Уникальный кластеризованый несоставной индекс

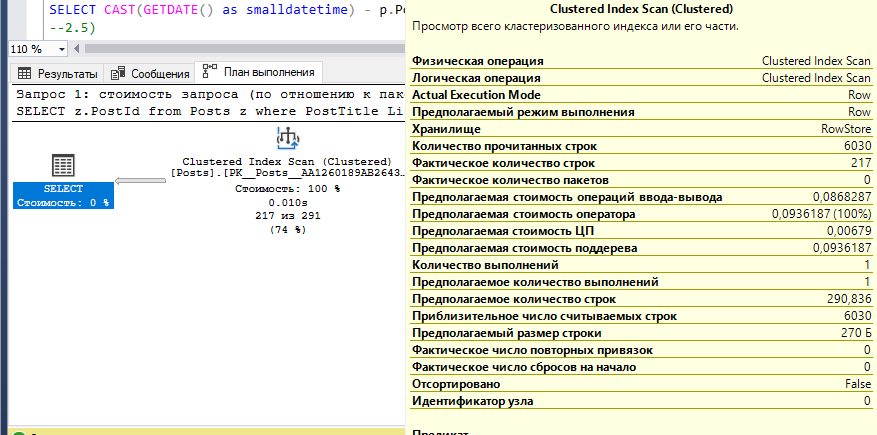
--2.1)

SELECT z.AuthorId, u.PostTitle from Posts z Inner Join Posts u on u.PostId = z.PostId



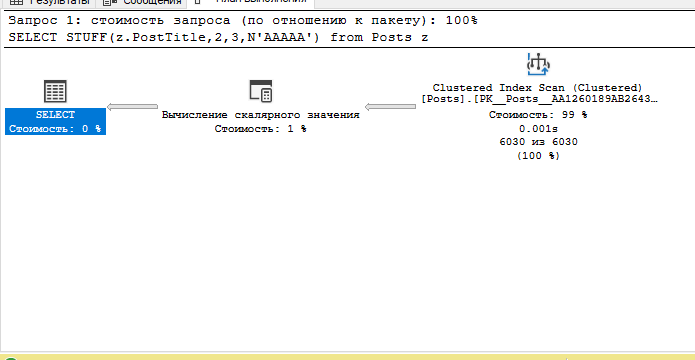
--2.2)

SELECT z.PostId from Posts z where PostTitle Like N'%a'



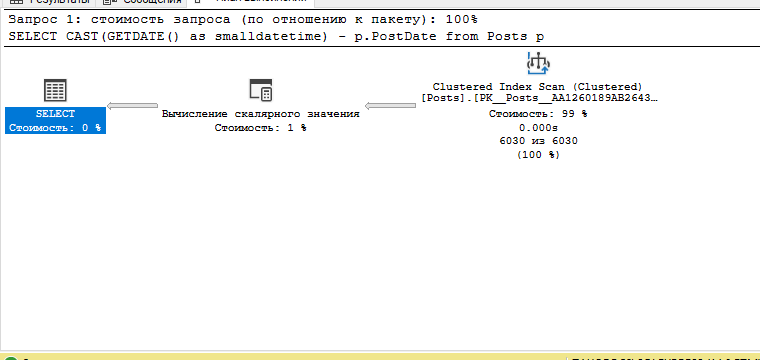
--2.3)

SELECT STUFF(z.PostTitle,2,3,N'AAAAA') from Posts z



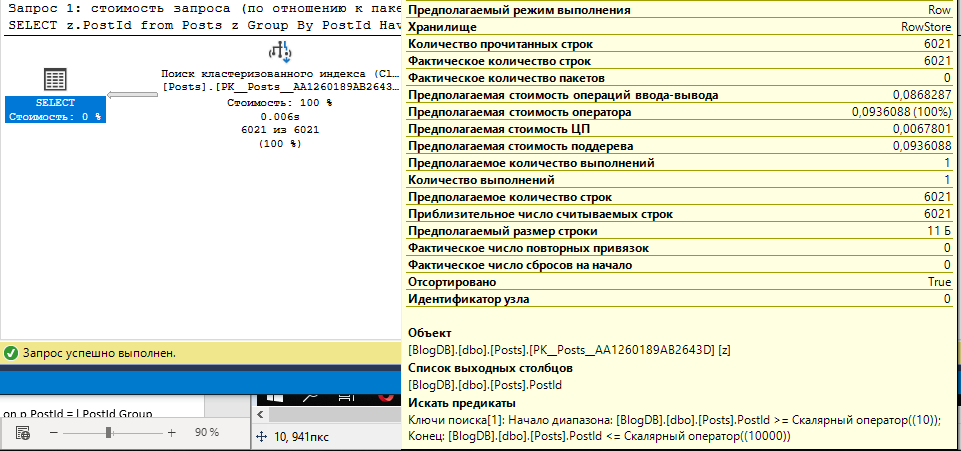
--2.4)

SELECT CAST(GETDATE() as smalldatetime) - p.PostDate from Posts p



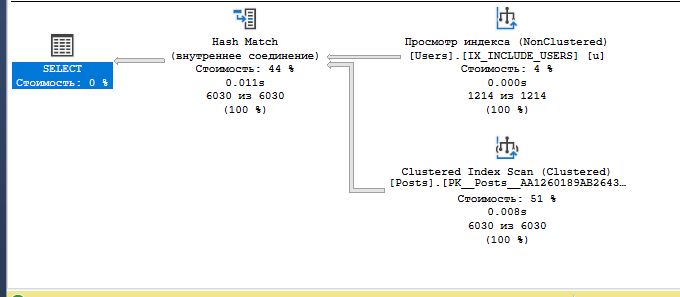
--2.5)

SELECT z.PostId from Posts z Group By PostId Having z.PostId BETWEEN 10 AND 10000



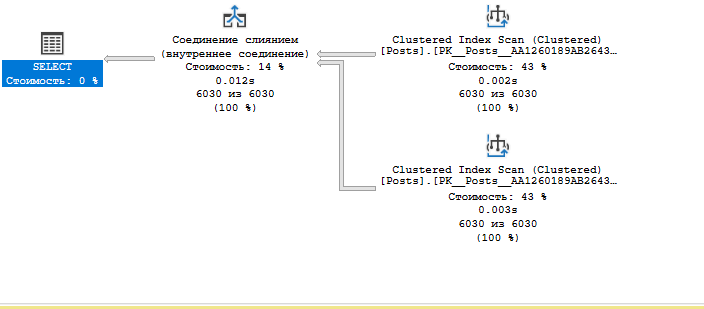
--2.6)

SELECT z.PostTitle from Posts z where Exists(select \* from Users u where u.UserId = z.AuthorId)



--2.7)

SELECT PostId, PostTitle from Posts INTERSECT SELECT PostId, PostTitle from Posts



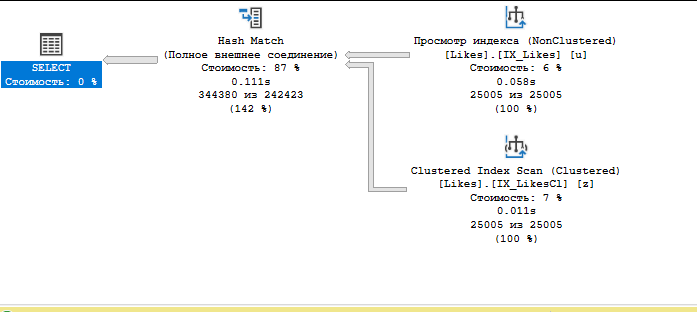
CREATE UNIQUE CLUSTERED INDEX IX\_LikesCl

ON Likes (USERID, POSTID)

Go

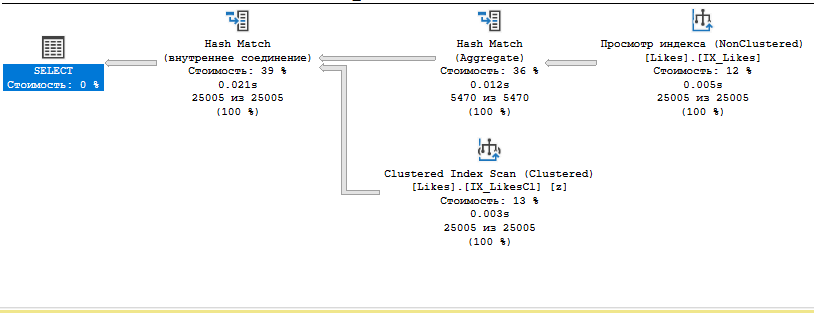
--3.1)

SELECT z.PostId from Likes z WITH(Index(IX\_LikesCl)) Full Outer Join Likes u on u.PostId = z.UserId



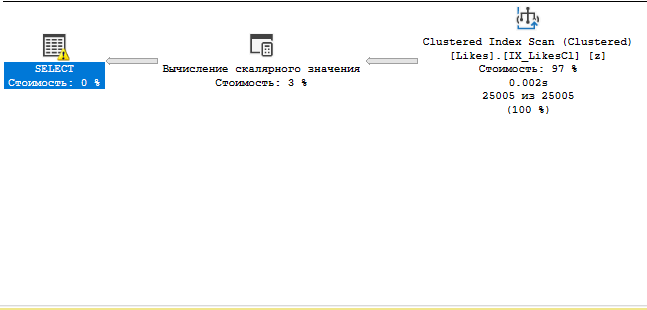
--3.2)

SELECT z.UserId from Likes z WITH(Index(IX\_LikesCl)) where z.PostId = any (Select PostId from Likes)



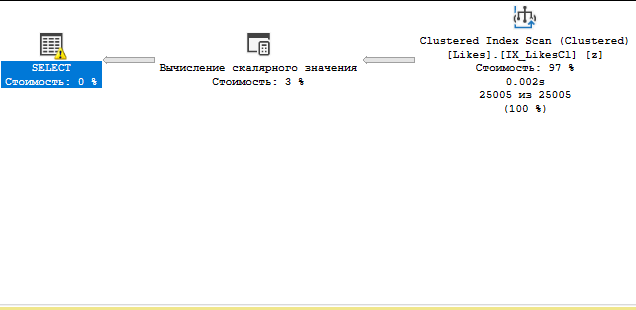
--3.3)

SELECT Substring(N'AAAAA' + Cast(z.PostId as nvarchar),2,4) from Likes z WITH(Index(IX\_LikesCl))



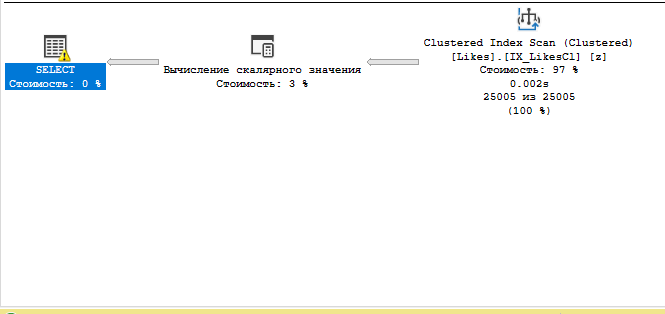
--3.4)

SELECT GETDATE() + l.UserId from Likes l WITH(Index(IX\_LikesCl))



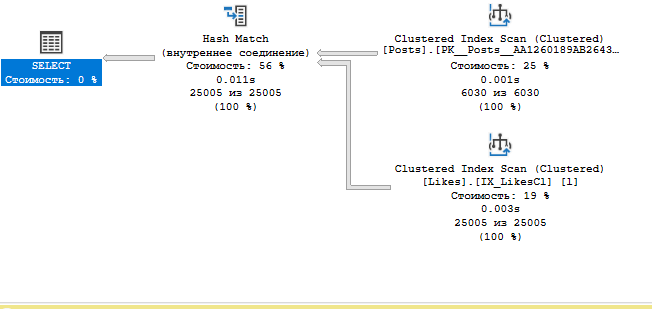
--3.5)

SELECT l.UserId, p.PostTitle from Likes l WITH(Index(IX\_LikesCl)) join Posts p on p.PostId = l.PostId Group By l.UserId, PostTitle Having l.UserId >100



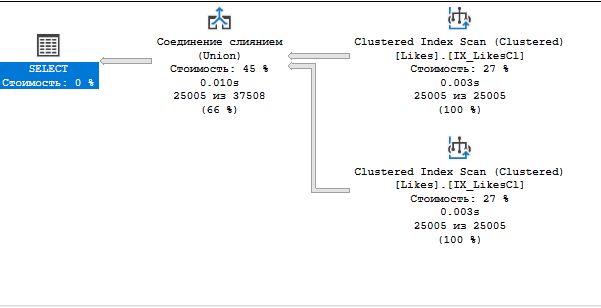
--3.6)

SELECT l.PostId from Likes l WITH(Index(IX\_LikesCl)) where Exists(select \* from Posts p where p.PostId = l.PostId)



--3.7)

SELECT UserId, PostId from Likes WITH(Index(IX\_LikesCl)) UNION SELECT UserId, PostId from Likes WITH(Index(IX\_LikesCl))

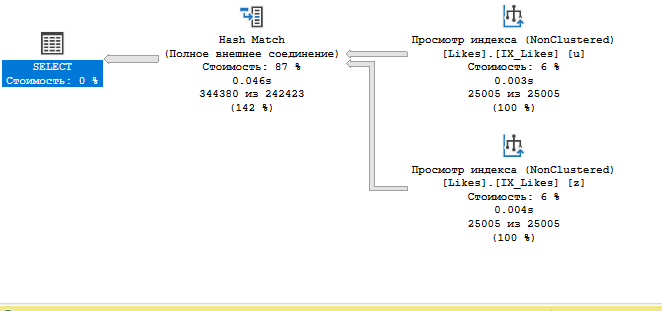


CREATE UNIQUE NONCLUSTERED INDEX IX\_Likes

ON Likes (USERID, POSTID)

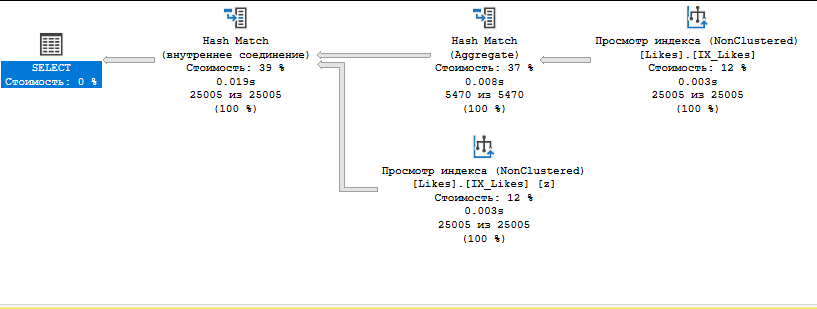
Go

-4.1)



SELECT z.PostId from Likes z WITH(Index(IX\_Likes)) Full Outer Join Likes u on u.PostId = z.UserId

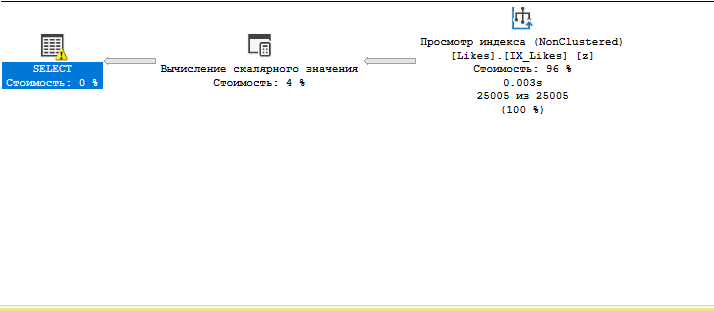
--4.2)



SELECT z.UserId from Likes z WITH(Index(IX\_Likes)) where z.PostId = any (Select PostId from Likes)

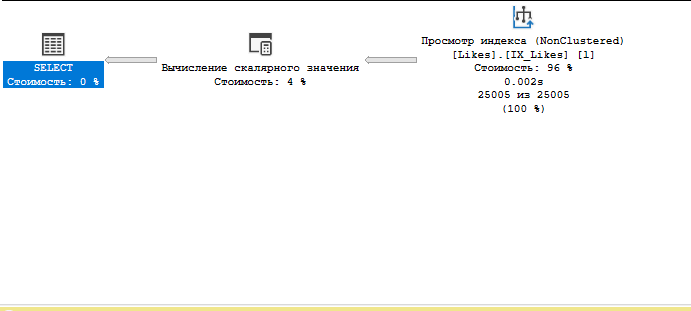
--4.3)

SELECT Substring(N'AAAAA' + Cast(z.PostId as nvarchar),2,4) from Likes z WITH(Index(IX\_Likes))

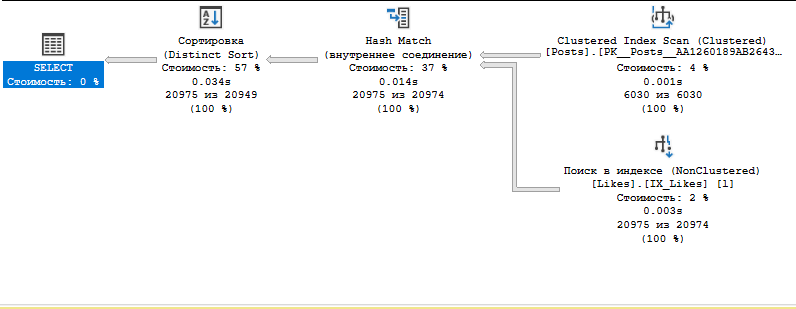


--4.4)

SELECT GETDATE() + l.UserId from Likes l WITH(Index(IX\_Likes))



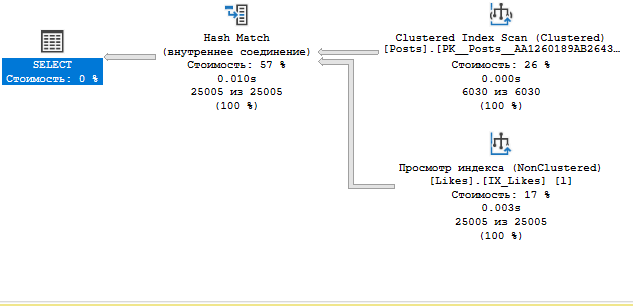
--4.5)



SELECT l.UserId, p.PostTitle from Likes l WITH(Index(IX\_Likes)) join Posts p on p.PostId = l.PostId Group By l.UserId, PostTitle Having l.UserId >100

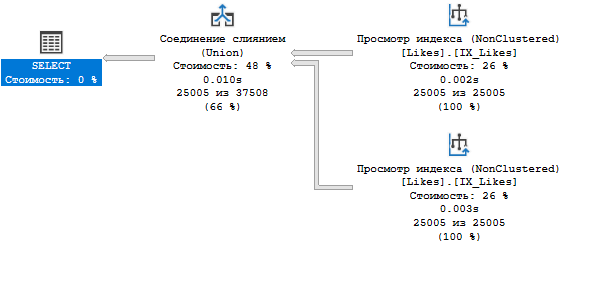
--4.6)

SELECT l.PostId from Likes l WITH(Index(IX\_Likes)) where Exists(select \* from Posts p where p.PostId = l.PostId)



--4.7)

SELECT UserId, PostId from Likes WITH(Index(IX\_Likes)) UNION SELECT UserId, PostId from Likes WITH(Index(IX\_Likes))



CREATE NONCLUSTERED INDEX IX\_Posts\_with\_theme

ON Posts (PostId)

INCLUDE (PostTitle)

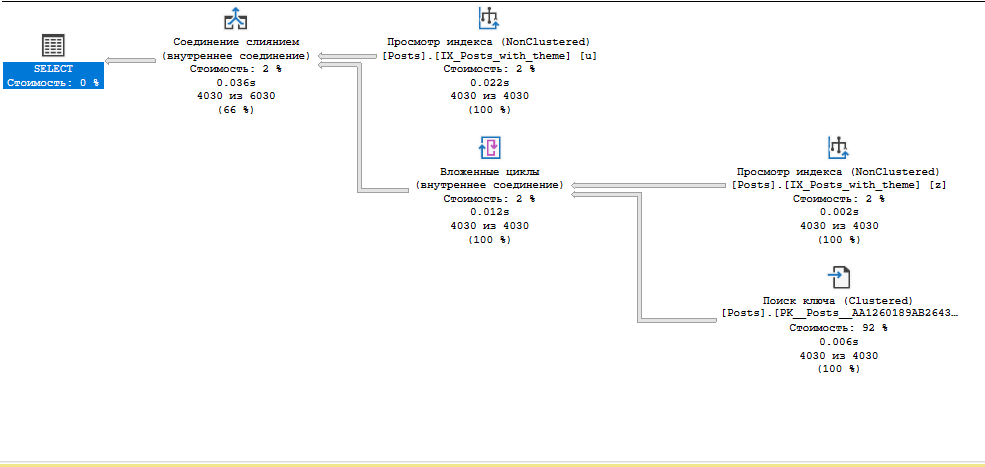
WHERE PostId >2000

GO

--5.1)

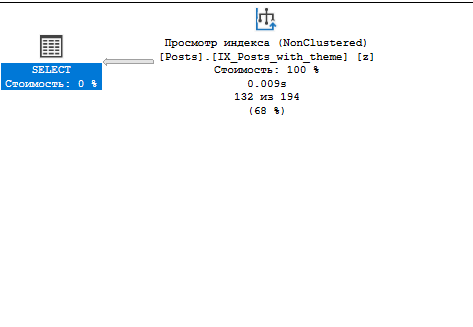
SELECT z.ThemeId, u.PostTitle from Posts z WITH(Index(IX\_Posts\_with\_theme)) Inner Join Posts u WITH(Index(IX\_Posts\_with\_theme)) on u.PostId = z.PostId

WHERE u.PostId >2000 and z.PostId >2000



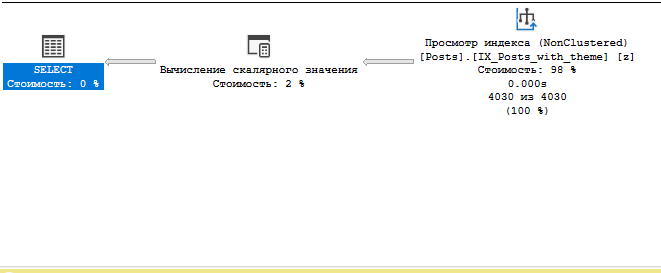
--5.2)

SELECT z.PostId from Posts z WITH(Index(IX\_Posts\_with\_theme)) where PostTitle Like N'%a' and z.PostId >2000



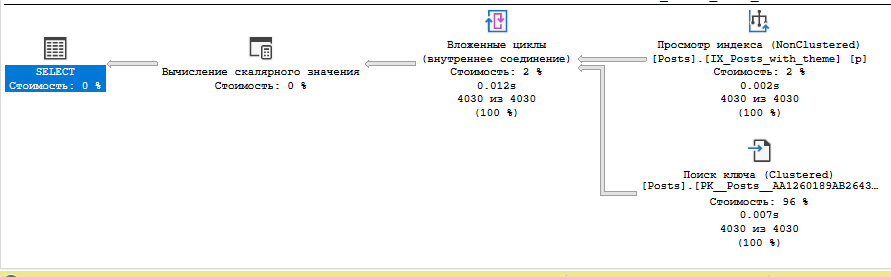
--5.3)

SELECT STUFF(z.PostTitle,2,3,N'AAAAA') from Posts z WITH(Index(IX\_Posts\_with\_theme)) where z.PostId >2000



--5.4)

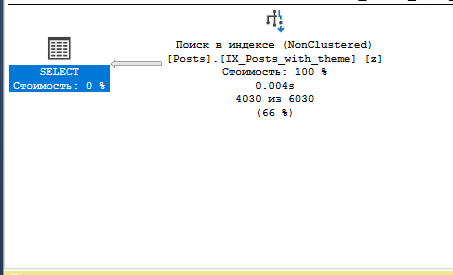
SELECT CAST(GETDATE() as smalldatetime) - p.PostDate from Posts p WITH(Index(IX\_Posts\_with\_theme)) where p.PostId >2000



--5.5)

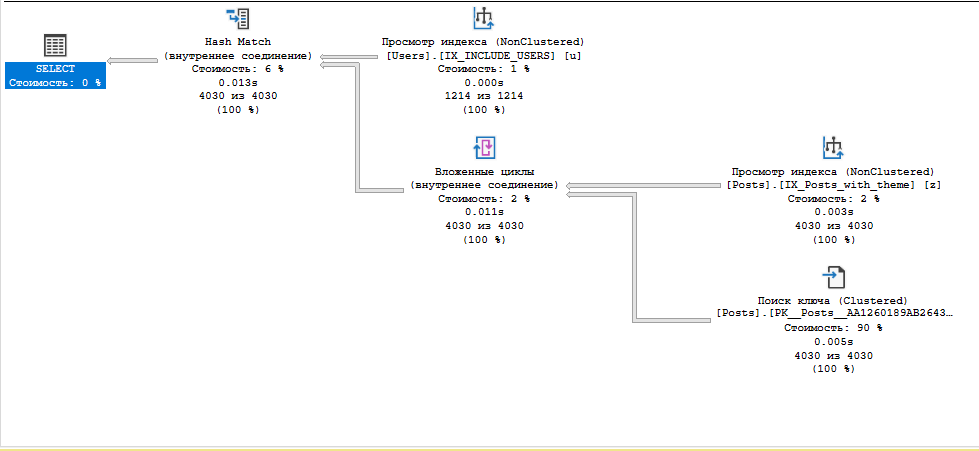
SELECT z.PostId from Posts z WITH(Index(IX\_Posts\_with\_theme)) where z.PostId >2000

Group By PostId Having z.PostId BETWEEN 10 AND 10000



--5.6)

SELECT z.PostTitle from Posts z WITH(Index(IX\_Posts\_with\_theme)) where Exists(select \* from Users u where u.UserId = z.AuthorId) and z.PostId >2000



--5.7)

SELECT PostId, PostTitle from Posts WITH(Index(IX\_Posts\_with\_theme)) where PostId >2000

INTERSECT

SELECT PostId, PostTitle from Posts WITH(Index(IX\_Posts\_with\_theme)) where PostId >2000

