**Лабораторная работа № 9. Создание и применение индексов**

1. С помощью SSMS определить все индексы, которые имеются в БД **UNIVER**. Определить, какие из них являются кластеризованными, а какие некластеризованными. Создать временную локальную таблицу. Заполнить ее данными (не менее 1000 строк). Разработать SELECT-запрос. Получить план запроса и определить его стоимость. Создать кластеризованный индекс, уменьшающий стоимость SELECT-запроса.

EXEC SP\_HELPINDEX 'AUDITORIUM';

EXEC SP\_HELPINDEX 'AUDITORIUM\_TYPE';

EXEC SP\_HELPINDEX 'FACULTY';

EXEC SP\_HELPINDEX 'GROUPS';

EXEC SP\_HELPINDEX 'PROFESSION';

EXEC SP\_HELPINDEX 'PROGRESS';

EXEC SP\_HELPINDEX 'PULPIT';

EXEC SP\_HELPINDEX 'STUDENT';

EXEC SP\_HELPINDEX 'SUBJECT\_T';

EXEC SP\_HELPINDEX 'TEACHER';

CREATE TABLE #EXAMPLE\_1 (

ID int IDENTITY(1,1),

TFIELD varchar(100)

);

SET nocount on;

DECLARE @i int = 0;

WHILE (@i < 1000)

BEGIN

INSERT #EXAMPLE\_1 VALUES(@i + GETDATE());

SET @i = @i + 1;

END;

SELECT \*

FROM #EXAMPLE\_1

WHERE ID BETWEEN 150 AND 250

ORDER BY ID;

checkpoint;

DBCC DROPCLEANBUFFERS;

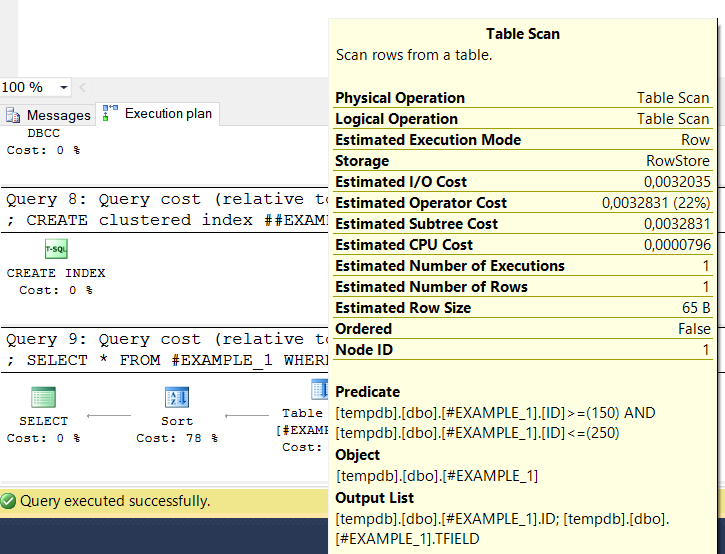
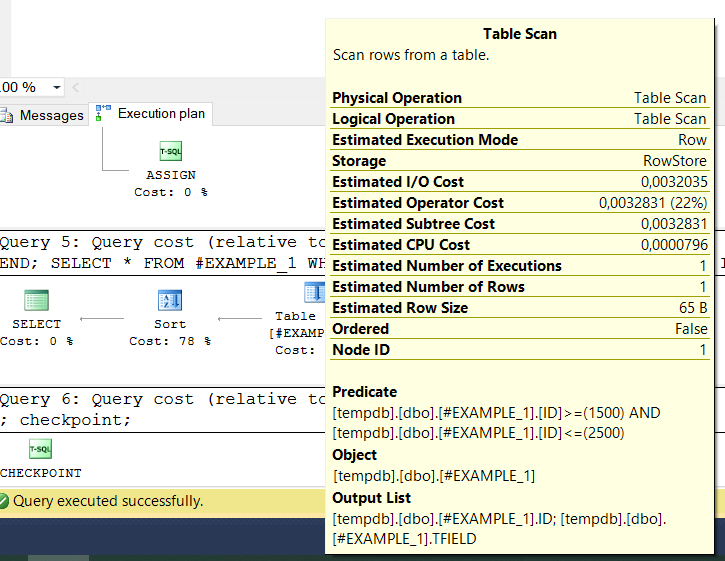
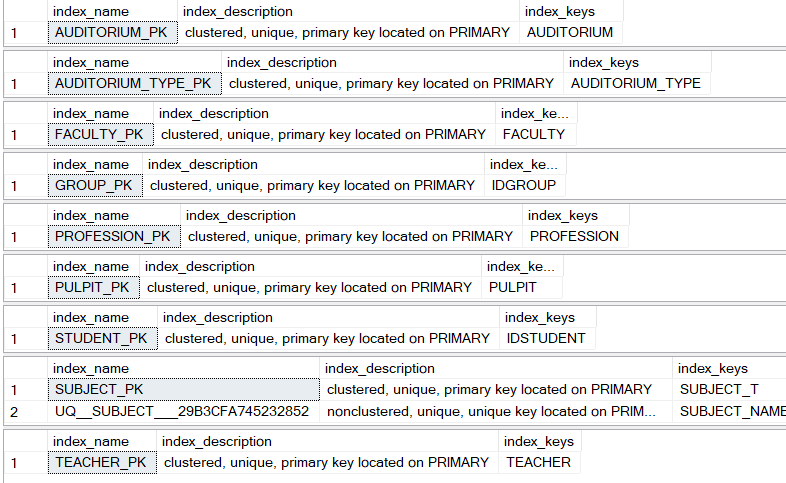
CREATE clustered index ##EXAMPLE\_1\_CL on #EXAMPLE\_1(ID asc);

SELECT \*

FROM #EXAMPLE\_1

WHERE ID BETWEEN 150 AND 250

ORDER BY ID;



1. Создать временную локальную таблицу. Заполнить ее данными (10000 строк или больше). Разработать SELECT-запрос. Получить план запроса и определить его стоимость. Создать *некластеризованный* неуникальный *составной* индекс. Оценить процедуры поиска информации.

GO

CREATE TABLE #EXAMPLE\_2 (

ID int IDENTITY(1,1),

TFIELD varchar(100),

RAND\_NUM int

);

SET nocount on;

DECLARE @i int = 0;

WHILE (@i < 10000)

BEGIN

INSERT #EXAMPLE\_2 VALUES(@i + GETDATE(), floor(300\*RAND()));

SET @i = @i + 1;

END;

SELECT COUNT(\*) [Row count]

FROM #EXAMPLE\_2;

checkpoint;

DBCC DROPCLEANBUFFERS;

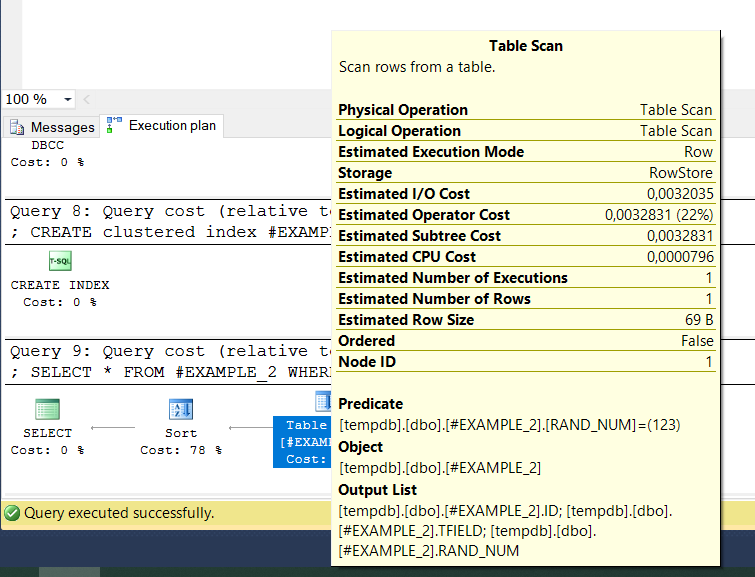
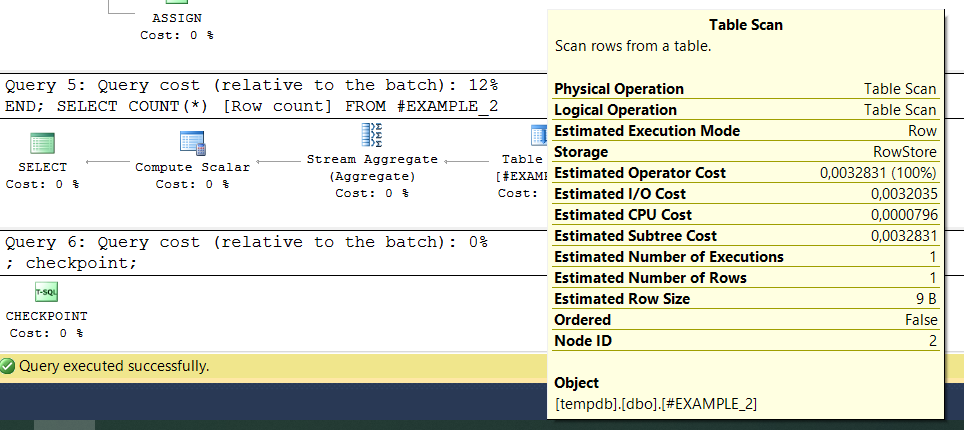
CREATE clustered index #EXAMPLE\_2\_CL on #EXAMPLE\_2(ID asc, TFIELD);

SELECT \*

FROM #EXAMPLE\_2

WHERE RAND\_NUM = 123

ORDER BY ID;



1. Создать временную локальную таблицу. Заполнить ее данными (не менее 10000 строк). Разработать SELECT-запрос. Получить план запроса и определить его стоимость. Создать *некластеризованный индекс покрытия*, уменьшающий стоимость SELECT-запроса.

GO

CREATE TABLE #EXAMPLE\_3 (

ID int IDENTITY(1,1),

TFIELD varchar(100),

RAND\_NUM int

);

SET nocount on;

DECLARE @i int = 0;

WHILE (@i < 10000)

BEGIN

INSERT #EXAMPLE\_3 VALUES(@i + GETDATE(), floor(300\*RAND()));

SET @i = @i + 1;

END;

SELECT TFIELD from #EXAMPLE\_3 where ID between 1 and 100;

checkpoint;

DBCC DROPCLEANBUFFERS;

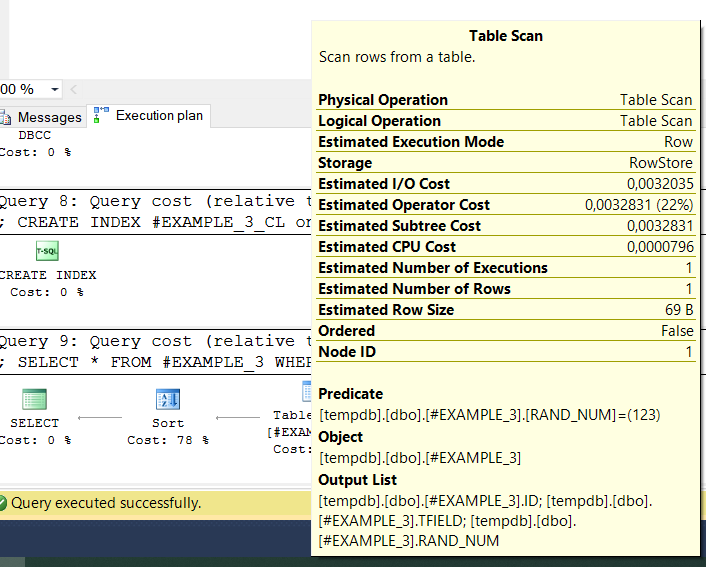
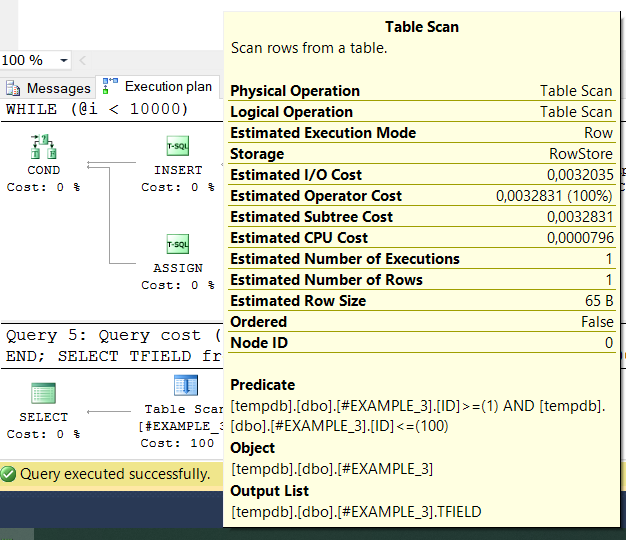
CREATE INDEX #EXAMPLE\_3\_CL on #EXAMPLE\_3(ID) INCLUDE (TFIELD);

SELECT \*

FROM #EXAMPLE\_3

WHERE RAND\_NUM = 123

ORDER BY ID;



1. Создать и заполнить временную локальную таблицу. Разработать SELECT-запрос, получить план запроса и определить его стоимость. Создать *некластеризованный фильтруемый индекс*, уменьшающий стоимость SELECT-запроса.

CREATE TABLE #EXAMPLE\_4 (

ID int IDENTITY(1,1),

TFIELD varchar(100),

RAND\_NUM int

);

SET nocount on;

DECLARE @i int = 0;

WHILE (@i < 20000)

BEGIN

INSERT #EXAMPLE\_4 VALUES(@i + GETDATE(), floor(30000\*RAND()));

SET @i = @i + 1;

END;

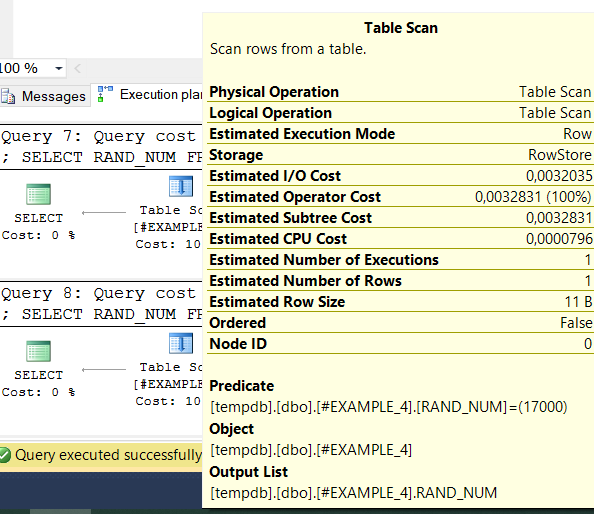
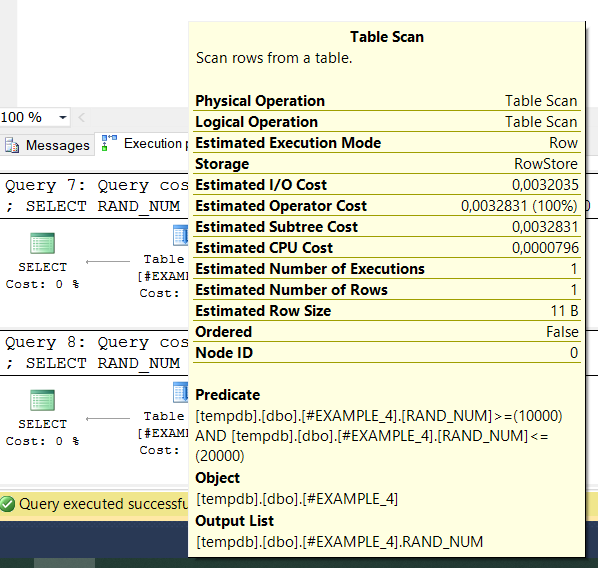
CREATE INDEX #EX\_WHERE ON #EXAMPLE\_4(RAND\_NUM)

WHERE RAND\_NUM >= 10000 AND RAND\_NUM <= 20000;

SELECT RAND\_NUM FROM #EXAMPLE\_4 WHERE RAND\_NUM BETWEEN 5000 AND 19999;

SELECT RAND\_NUM FROM #EXAMPLE\_4 WHERE RAND\_NUM >= 10000 AND RAND\_NUM <= 20000;

SELECT RAND\_NUM FROM #EXAMPLE\_4 WHERE RAND\_NUM = 17000;



1. Заполнить временную локальную таблицу. Создать некластеризованный индекс. Оценить уровень *фрагментации индекса*. Разработать сценарий на T-SQL, выполнение которого приводит к уровню фрагментации индекса выше 90%. Оценить уровень фрагментации индекса. Выполнить процедуру *реорганизации* индекса, оценить уровень фрагментации. Выполнить процедуру *перестройки* индекса и оценить уровень фрагментации индекса.

CREATE TABLE #EXAMPLE\_5 (

ID int IDENTITY(1,1),

TFIELD varchar(100),

RAND\_NUM int

);

SET nocount on;

DECLARE @i int = 0;

WHILE (@i < 20000)

BEGIN

INSERT #EXAMPLE\_5 VALUES(@i + GETDATE(), floor(30000\*RAND()));

SET @i = @i + 1;

END;

SELECT \* FROM #EXAMPLE\_5 where ID between 150 and 200 order by ID;

SELECT name [Индекс], avg\_fragmentation\_in\_percent [Фрагментация(%)]

FROM sys.dm\_db\_index\_physical\_stats(DB\_ID(N'TEMPDB'),

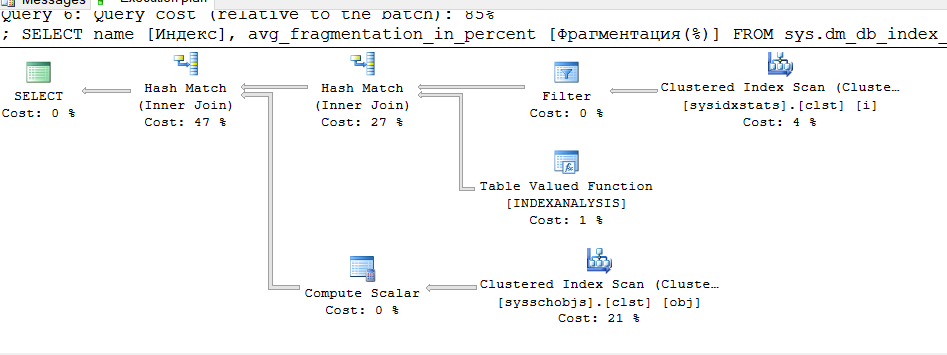
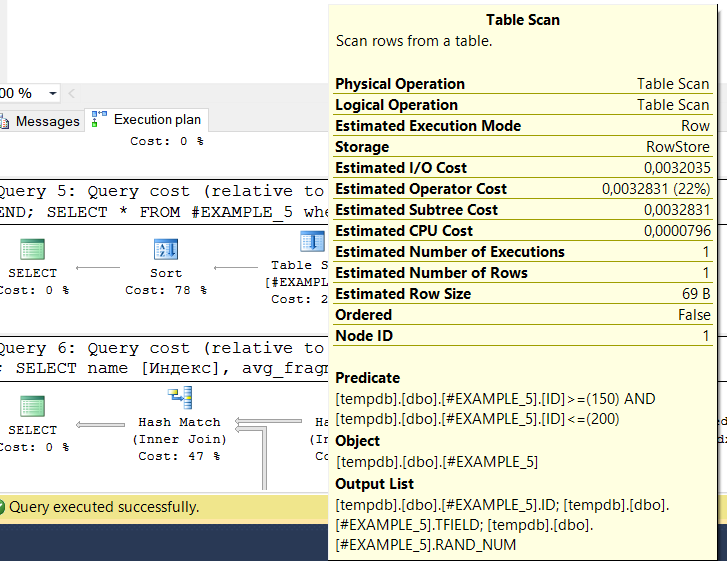
OBJECT\_ID(N'#EXAMPLE\_5'),NULL,NULL,NULL) ss

JOIN sys.indexes ii on ss.index\_id = ii.index\_id and ss.object\_id = ii.object\_id

WHERE name is not null;

ALTER index #EXAMPLE\_UNCL on #EXAMPLE\_5 reorganize;

ALTER index #EXAMPLE\_UNCL on #EXAMPLE\_5 rebuild with (online = off);



1. Разработать пример, демонстрирующий применение параметра FILLFACTOR при создании некластеризованного индекса.

CREATE TABLE #EXAMPLE\_6 (

ID int IDENTITY(1,1),

TFIELD varchar(100),

RAND\_NUM int

);

SET nocount on;

DECLARE @i int = 0;

WHILE (@i < 20000)

BEGIN

INSERT #EXAMPLE\_6 VALUES(@i + GETDATE(), floor(30000\*RAND()));

SET @i = @i + 1;

END;

CREATE index #EX\_TKEY on #EXAMPLE\_6(RAND\_NUM) with (fillfactor = 65);

INSERT top(50) percent INTO #EXAMPLE\_6(TFIELD, RAND\_NUM)

SELECT TFIELD, RAND\_NUM FROM #EXAMPLE\_6;

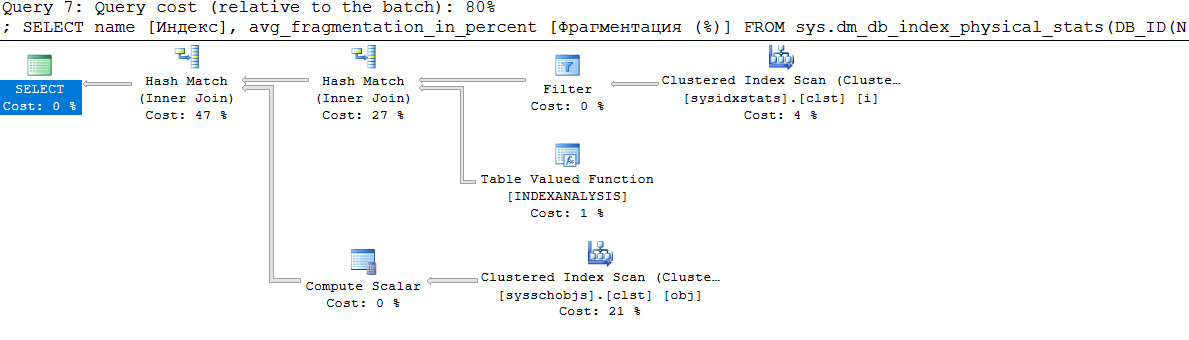
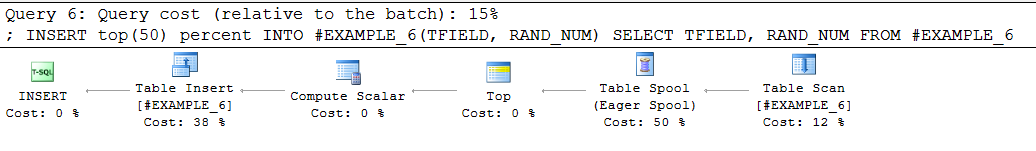
SELECT name [Индекс], avg\_fragmentation\_in\_percent [Фрагментация (%)]

FROM sys.dm\_db\_index\_physical\_stats(DB\_ID(N'TEMPDB'),

OBJECT\_ID(N'#EXFMPLE\_6'), NULL, NULL, NULL) ss JOIN sys.indexes ii

ON ss.object\_id = ii.object\_id and ss.index\_id = ii.index\_id

WHERE name is not null;



1. Создать необходимые индексы и проанализировать планы запросов с использованием этих индексов для таблицы базы данных **X\_MyBASE**.

use Dobriyan\_MyBase;

--- #

EXEC SP\_HELPINDEX 'Groups';

EXEC SP\_HELPINDEX 'Courses';

EXEC SP\_HELPINDEX 'Specialities';

EXEC SP\_HELPINDEX 'Teachers';

--- #

SELECT Number, Students\_Count

FROM Groups

WHERE Students\_Count >= 15 AND Students\_Count <= 20;

CREATE INDEX STUD\_COUNT ON Groups(Students\_Count)

WHERE Students\_Count >= 10 AND Students\_Count <= 20;

SELECT Number, Students\_Count

FROM Groups

WHERE Students\_Count >= 15 AND Students\_Count <= 20;

--- #

CREATE INDEX STUD\_COUNT\_SPECIALITY ON Groups(Students\_Count asc, Speciality);

SELECT Students\_Count, Speciality

FROM Groups;

--- #

SELECT name [Индекс], avg\_fragmentation\_in\_percent [Фрагментация(%)]

FROM sys.dm\_db\_index\_physical\_stats(DB\_ID(N'Dobriyan\_MyBase'),

OBJECT\_ID(N'Groups'),NULL,NULL,NULL) ss

JOIN sys.indexes ii on ss.index\_id = ii.index\_id and ss.object\_id = ii.object\_id

WHERE name is not null;

