Лабораторная №9

Задание 1

Условие:

Определить все индексы, которые имеются в БД **UNIVER**.

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.

Решение:

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'

exec sp\_helpindex 'TEACHER'

CREATE TABLE #New\_table

(

ID int identity(1,1),

STRING varchar(13)

)

DECLARE @iter int = 0;

WHILE @iter < 10000

begin

INSERT INTO #New\_table values (REPLICATE('строка ',2));

SET @iter = @iter + 1;

end;

CREATE CLUSTERED INDEX #EXAMPLE\_CL1 on #New\_table(ID asc);

DROP INDEX #EXAMPLE\_CL1 on #New\_table

SELECT \* FROM #New\_table where ID between 150 and 200 order by ID;

DROP TABLE #New\_table

Результат:

Без индекса:



С индексом:



Задание 2

Условие:

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.

Решение:

create table #table2

(

id int identity(1,1),

stroke varchar(15)

);

drop table #table2

declare @iteration int = 0;

while @iteration < 10000

begin

insert into #table2 values ('stroke')

set @iteration =@iteration + 1;

end

CREATE index #index2 on #table2(id,stroke);

DROP index [#table2].[#index2]

checkpoint;

DBCC DROPCLEANBUFFERS;

select \* from #table2

where #table2.id > 150 and #table2.id<456

order by id

Результат:

Без индекса:  
  
С индексом:



Задание 3

Условие:

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.  
Решение:

CREATE INDEX #index3 on #table2(id) INCLUDE (stroke);

DROP index [#table2].[#index3]

SELECT stroke from #table2 where id between 1 and 10;

Результат:

Без индекса:  
  
С индексом:



Задание 4

Условие:

Создать и заполнить временную локальную таблицу.

Разработать SELECT-запрос, получить план запроса и определить его стоимость.

Создать *некластеризованный фильтруемый индекс*, уменьшающий стоимость SELECT-запроса.  
Решение:

CREATE INDEX #index4 on #table2(id) where(id>=0 and id<=10)

drop INDEX #index4 on #table2

DBCC DROPCLEANBUFFERS; --очистить буферный кэш

select stroke from #table2

where id>=0 and id<=10  
Результат:

Без индекса:



С индексом:



Задание 5

Условие:

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.

Решение:

Use TEMPDB

CREATE TABLE #TASK5

(

INFO NVARCHAR (20),

ITERATOR INT IDENTITY(1,1),

INDEX\_ INT

)

DROP TABLE #TASK5

DECLARE @X INT =0;

WHILE @X <= 100000

BEGIN

INSERT INTO #TASK5(INFO,INDEX\_)

VALUES ('Строка' + CAST(@X AS NVARCHAR),FLOOR(20000\*RAND()))

SET @X +=1;

END

CREATE INDEX #TASK5\_KEY ON #TASK5(INDEX\_)

DROP INDEX #TASK5\_KEY ON #TASK5

SELECT NAME [Индекс], AVG\_FRAGMENTATION\_IN\_PERCENT [Фрагментация (%)]

FROM SYS.DM\_DB\_INDEX\_PHYSICAL\_STATS(DB\_ID(N'TEMPDB'),

OBJECT\_ID(N'#TASK5'), 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;

INSERT TOP(10000) #TASK5(INDEX\_ ,INFO) SELECT INDEX\_, INFO FROM #TASK5

ALTER INDEX #TASK5\_KEY ON #TASK5 REORGANIZE

ALTER INDEX #TASK5\_KEY ON #TASK5 REBUILD WITH (ONLINE = OFF)

Результат:

Индекс:  


Реорганизация:



Ребилд:



Задание 6

Условие:

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

Решение:

go

CREATE TABLE #TASK6

(

INFO NVARCHAR (20),

ITERATOR INT IDENTITY(1,1),

INDEX\_ INT

)

DECLARE @X INT =0;

WHILE @X <= 100000

BEGIN

INSERT INTO #TASK6(INFO,INDEX\_)

VALUES ('СТРОКА' + CAST(@X AS NVARCHAR),FLOOR(20000\*RAND()))

SET @X +=1;

END

CREATE INDEX #TASK6\_TKEY ON #TASK6(INDEX\_) WITH FILLFACTOR = 65

INSERT TOP(100) PERCENT #TASK6(INDEX\_, INFO)

SELECT INDEX\_, INFO FROM #TASK6

SELECT NAME [Индекс], AVG\_FRAGMENTATION\_IN\_PERCENT [Фрагментация (%)]

FROM SYS.DM\_DB\_INDEX\_PHYSICAL\_STATS(DB\_ID(N'TEMPDB'),

OBJECT\_ID(N'#TASK5'), 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;

DROP INDEX #TASK6\_TKEY ON #TASK6

Результат:

