Учреждение образования “БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ”

Лабораторная работа №9. Создание и применение индексов

Выполнил:

студент 2 курса, 7 группы

Бобрович Глеб Сергеевич

Минск 2022ПроТТеоре

|  |  |  |
| --- | --- | --- |
| **Задание** |  | |
| 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. | | **Задание 1**  USE ЛР9;  exec sp\_helpindex 'PULPIT'  exec sp\_helpindex 'SUBJECT' --перечень индексов, связ. с д-й таблицей  exec sp\_helpindex 'FACULTY'  exec sp\_helpindex 'TEACHER' |
| 2.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. | | **Задание 2**  USE ЛР9;  create table #QUERY2LR9  ( ID int,  FIELD varchar(20)  )  SET nocount on; --не вывод сообщения о вводе строк  DECLARE @i int = 0;  while @i < 100  begin  insert #QUERY2LR9(ID, FIELD)  values(FLOOR(RAND()\*10000), REPLICATE('FIELD',2));  SET @i = @i + 1;  end  select \* from #QUERY2LR9 where ID between 1500 and 5000 order by ID  checkpoint; --фиксация БД  DBCC DROPCLEANBUFFERS; --очистить буферный кэш  CREATE clustered index #QUERY2LR9\_cl on #QUERY2LR9(ID asc)  drop table #QUERY2LR9 |
| 3.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. | | **Задание 3**  USE ЛР9;  create table #QUERY3LR9  (  ID int,  FIELD varchar(20),  cc int identity(1,1)  )  SET nocount on;  DECLARE @k int = 0;  while @k < 100  begin  insert #QUERY3LR9(ID, FIELD)  values(FLOOR(RAND()\*30000), REPLICATE('julia',3) );  SET @k = @k + 1;  end  select \* from #QUERY3LR9 where cc >500 and ID between 1500 and 5000  CREATE index #QUERY3LR9\_nonclu\_2 on #QUERY3LR9(ID) INCLUDE(cc)  select CC from #QUERY3LR9 where ID > 500  --drop table #QUERY3LR9 |
| 4.Создать и заполнить временную локальную таблицу.  Разработать SELECT-запрос, получить план запроса и определить его стоимость.  Создать *некластеризованный фильтруемый индекс*, уменьшающий стоимость SELECT-запроса. | | **Задание 4**  tempdb;  SELECT ID from #QUERY3LR9 where ID between 5000 and 19999;  SELECT ID from #QUERY3LR9 where ID>15000 and ID < 20000  SELECT ID from #QUERY3LR9 where ID=17000  CREATE index #temp\_table\_WHERE on #QUERY3LR9(ID) where (ID>15000 and  ID < 20000);  drop index #QUERY3LR9\_WHERE on #QUERY3LR9 |
| 5.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. | | **Задание 5**  use tempdb  CREATE index #QUERY3LR9\_ID on #QUERY3LR9(ID);  SELECT name [Индекс],  avg\_fragmentation\_in\_percent [Фрагментация(%)]  FROM sys.dm\_db\_index\_physical\_stats(DB\_ID(N'TEMPDB'),  OBJECT\_ID(N'#QUERY3LR9'), 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) #QUERY3LR9(ID, FIELD) select ID, FIELD from #QUERY3LR9;  ALTER index #QUERY3LR9\_ID on#QUERY3LR9 reorganize;  ALTER index #QUERY3LR9\_ID on #QUERY3LR9 rebuild with (online = off);  drop index #QUERY3LR9\_ID on #QUERY3LR9 |
| 6.Разработать пример, демонстрирующий применение параметра FILLFACTOR при создании некластеризованного индекса. | | **Задание 6**  USE ЛР9;  CREATE index #QUERY3LR9\_ID1 on #QUERY3LR9(ID)with (fillfactor = 65);  INSERT top(50)percent into #QUERY3LR9(ID, FIELD) select IND, FIELD from #QUERY3LR9;  use tempdb  SELECT name [Индекс], avg\_fragmentation\_in\_percent [Фрагментация(%)]  FROM sys.dm\_db\_index\_physical\_stats(DB\_ID(N'TEMPDB'), OBJECT\_ID(N'#temp\_table\_2'), 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 #QUERY3LR9\_ind1 on #QUERY3LR9 |