# 1 задание

# Задача 1

Имеется таблица Employees

CREATE TABLE Employees (

ID int NOT NULL IDENTITY(1,1) primary key,

TabNum int,

LastName nvarchar(50) NOT NULL,

FirstName nvarchar(50) NOT NULL,

Email nvarchar(250) NOT NULL UNIQUE,

DateOfBirth datetime2,

Gender char(1),

-- Еще двадцать столбцов

)

CREATE INDEX idx\_Employees\_TabNum ON Employees (TabNum)

Опишите, какие индексы объявлены в этой таблице.

Решение:

Объявлены индексы неуникальный некластерный индекс TabNum,

(сформированы автоматически) – уникальный кластерный индекс по первичному ключу ID, уникальный некластерный индекс Email

# Задача 2

В результате профилирования работы приложения в рабочей среде вы выяснили, что наиболее часто вызываются запросы вида

SELECT LastName, FirstName, Email

FROM Employees

WHERE TabNum = 1

Почему поиск с использованием индекса по полю TabNum для данного запроса – это хорошо?

Ответ:

- Это прямой доступ к данным согласно условию, нет перебора по всем строкам. То есть вызовется Index Seek  найдет, где TabNum = 1, и прочитает TabNum и ID из каждой строки.

Почему поиск с использованием индекса по полю TabNum для данного запроса – это плохо?

Ответ:

- Потому что далее после index seek вызовется Key Lookup, для поиска оставшихся значения строк, которые отсутствуют в некластеризованном индексе, а именно LastName, FirstName, Email. То есть SQL Server придется обращаться к двум индексам для извлечения всех необходимых данных, что не является эффективным решением.

Что надо сделать, чтобы запрос с использованием индекса гарантированно давал преимущества по сравнению с неиспользованием индекса?

Ответ:

create index idx\_Employees\_TabNum ON Employees (TabNum) include (LastName, FirstName, Email) where TabNum = 1

# Задача 3

Мы добавили индекс

CREATE INDEX idx\_Employees\_DateOfBirth ON Employees (DateOfBirth)

Как он повлияет на скорость исполнения запроса, что можно изменить в индексе или запросе?

SELECT COUNT(\*)

FROM Employees

WHERE DATEPART(YEAR, DateOfBirth) = 1998

Решение:

Замедлит скорость исполнения запроса.

SELECT COUNT(\*)

FROM Employees

WHERE DateOfBirth >= '1998.01.01' and DateOfBirth < '1999.01.01'

# Задача 4

Мы добавили индекс

CREATE INDEX idx\_Employees\_LastName\_DateOfBirth

ON Employees (LastName, DateOfBirth)

Как он повлияет на скорость исполнения запроса, что можно изменить в индексе или запросе?

SELECT TOP 1 \*

FROM Employees

WHERE LastName = N'Иванов'

ORDER BY DateOfBirth DESC

Решение:

Не повлияет.

CREATE INDEX idx\_Employees\_LastName\_DateOfBirth ON Employees (LastName, DateOfBirth desc);

SELECT TOP 1 \*

FROM Employees

WHERE LastName = N'Иванов'

# Задача 5

Мы добавили индекс

CREATE INDEX idx\_Employees\_LastName\_DateOfBirth

ON Employees (LastName, DateOfBirth)

Как он повлияет на скорость исполнения запроса, что можно изменить в индексе или запросе?

SELECT \*

FROM Employees

WHERE DateOfBirth > '2000-01-01'

Решение:

Снизит скорость запроса.

CREATE INDEX idx\_Employees\_DateOfBirth

ON Employees (DateOfBirth)

# Задача 6

Мы добавили индекс

CREATE INDEX idx\_Employees\_Gender\_DateOfBirth

ON Employees (Gender, DateOfBirth)

Для ускорения запроса

SELECT DateOfBirth, COUNT(\*)

FROM Employees

WHERE Gender = 'M'

GROUP BY DateOfBirth

Затем запрос был изменен на

SELECT DateOfBirth, COUNT(\*)

FROM Employees

WHERE Gender = 'M'

AND LastName = N'Иванов'

GROUP BY DateOfBirth

Как это повлияет на скорость исполнения запроса, почему?

Решение:

Снизит скорость запроса, так как last name не в индексе

2 Задание

1нф



3нф из которой можно собрать таблицу из задания

