# 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.

ОТВЕТ:

Объявлен один некластеризованный индекс – idx\_Employees\_TabNum в столбце TabNum таблицы Employees

# 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?

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

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

ОТВЕТ:

Хорошо, потому, что будет давать ускорять исполнение запроса, в случае большого количества данных

Плохо, потому, что в случае небольшого количества данных в таблице не будет давать преимущества в поиске, может даже его замедлить

Для получения постоянного преимущества можно создать частичный индекс, плюс такого действия заключается в том, что он позволит избежать индексирования распространенных значений, таким образом – уменьшить размер индекса, а значит и ускорить запросы, использующие индекс

# 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

ОТВЕТ:

Индекс по дате не даст преимущества.

Функция DATEPART возвращает целое число, представляющее часть от типа date.

Чтобы найти строки, подходящие под условие WHERE необходимо выполнить сканирование кластерного индекса и вычислить значение функции.

ЧТО МОЖНО ИЗМЕНИТЬ:

Задать границы для дат в условии, где нужное значение будет находиться в некотором интервале. Добавить эти границы в условие индекса:

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

WHERE DateOfBirth >= ‘1998.01.01’ AND DateOfBirth < ‘1999.01.01’;

Изменить сам запрос:

SELECT COUNT(\*)

FROM Eemployees

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

ОТВЕТ:

Индекс повысит скорость исполнения запроса, так как все поля запроса содержатся в индексе.

ЧТО МОЖНО ИЗМЕНИТЬ:

Можно добавить условие LastName=N’Иванов’ в индекс, также можно добавить модификатор сортировки DESC в индекс после поля DateOfBirth

РЕЗУЛЬТАТ ИЗМЕНЕНИЯ:

CREATE INDEX idx\_Employees\_LastName\_DateOfBith

ON Employees(LastName, DateOfBirth DESC)

WHERE LastName = N’Иванов’;

SELECT TOP 1 \*

FROM Employees

WHERE LastName = N’Иванов’

ORDER BY DateOfBirth DESC

# 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'

ОТВЕТ:

Данный индекс является некорректным и не даст преимуществ, так как он не будет использоваться при выполнении запроса, он включает неиспользуемое поле на обслуживание.

ЧТО МОЖНО ИЗМЕНИТЬ:

Необходимо убрать поле LastName из конструкции создания индекса, также – сделать

Фильтрованный индекс с условием DateOfBirth > ‘2000-01-01’

РЕЗУЛЬТАТ ИЗМЕНЕНИЯ:

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

WHERE DateOfBirth > ‘2000-01-01’;

SELECT \*

FROM Employees

WHERE DateOfBirth > ‘2000-01-01’

# 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?

ОТВЕТ: Данный индекс является некорректным и не даст преимуществ в скорости, так как он не будет использоваться при выполнении запроса, он не включает используемое поле – LastName

Также, для получения полных преимуществ в скорости, необходимо добавить фильтрацию по плою - Gender

ЧТО МОЖНО ИЗМЕНИТЬ:

Необходимо добавить в конструкцию создания индекса поле LastName, а также – сделать фильтрацию LastName = N’Иванов’ и Gender = ‘M’

РЕЗУЛЬТАТ ИЗМЕНЕНИЯ

CREATE INDEX idx\_Employees\_Gender\_DateOfBirth\_LastName

ON Employees (Gender, DateOfBirth, LastName)

WHERE Gender = ‘M’ AND LastName = N’Иванов’;