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:

1) Колонка с именем "ID", так как это "primary key".

2) Колонка с именем "TabNum", так как сделано объявление индекса в инструкции после создания таблицы.

3)Колонка с именем "Email", так как UNIQUE

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:

1) Поиск с использованием индеска хорош, потому что мы выполняем "SELECT" запрос, т.е просто считываем данные, а находить нужные данные гораздо быстрее с помощью индексов, так как их структура B-tree. Еще TabNum - численный тип, что тоже хорошо для запроса.

2) Пробежася по лекции, и вот что мне пришло на ум. TabNum - столбец, который хранит не уникальные значения. Это понижает эффективность использования данного столбца в качестве индекса, так как он может обладать низкой селективностью.

3) Следовательно, надо создавать уникальный индекс, чтобы получать гарантированный прирост производительности.

P.S. В процессе размышления над следующей задачей, пришла мысль, что тут тоже можно сделать покрывающий индекс, чтобы иметь доступ к данным через столбец TabNum, а не через главный кластеризованный ключ.

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:

Никак не повлияет на скорость запроса, так как у нас в таблице уже есть один кластеризованный индекс! Соответственно поиск и сравнение элементов будет проходить по этому индексу, как и до этого, на самом деле. Поэтому надо создать такой индекс, который будет содержать все данные уже в себе (ну, как в кластерном индексе, на листьях содержатся данные, тут должно быть тоже самое, только данные по этому столбцу, тем более, что тут мы просто количество строчек считаем)

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:

Скорость запроса должна увеличиться, так как порядок столбцов в индексе совпадает

с порядком поиска нужных данных и их дальнейшей сортировки. Ничего менять не надо.

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: А вот в этом запросе, как раз-таки осуществляется поиск и сравнение не по главному проиндексированному столбцу, а по зависимому, что не должно быть хорошо.

Поэтому, я бы убрал составной индекс, и просто бы для колонки DateofBirth ввел свой индекс.

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:

Скорость выполнения запроса не изменится или ,даже, должна уменьшиться, так как мы вводим еще один параметр для поиска, а составной индекс не включает в себя этого столбца!

Тут можно добавить соответствующий столбец:

CREATE INDEX idx\_Employees\_Gender\_DateOfBirth

ON Employees (Gender, LastName, DateOfBirth)

И, я бы еще местами поменял Gender и LastName, так как у Gender всего 2 возможных варианта (хотя в наше время все возможно), следовательно этот столбец обладает низкой селективностью, что не есть хорошо.