# 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)

Ответ:

В данной таблице будет три индекса.

1. Индекс, который будет автоматически создан из-за использования PRIMARY KEY.
2. Индекс, автоматически созданный из-за ограничения UNIQUE.
3. Вручную созданный индекс «idx\_Employees\_TabNum»

# 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. В данном случае этот вариант хорош из-за того, что мы используем селект запрос для целочисленного типа, который будет работать быстрее вместе с индексом, за счет своей структуры данных (B-дерево).
2. Так как TubNum допускает использование неуникальных значений, эффективность использования индекса снижается, ввиду низкой плотности выборки.
3. Нужно сделать TubNum уникальным.

# 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:

Из-за того, что у нас в таблице уже есть один кластеризованный индекс, добавления этого индекса никак не ускорит скорость запроса, т. к. поиск эл-тов будет проходить по тому индексу.

Изменение:

Можно попробовать создать индекс, который будет содержать все данные в себе.

# 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:

В этом случае, за счет того, что порядок столбцов в индексе совпадает с порядком поиска и сортировки, скорость запроса увеличится.

# 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:

В этом случае порядок поиска происходит уже не проиндексированному столбцу, из-за чего скорость не должна увеличиться.

Изменение:

Стоит убрать составной индекс, и ввести для столбца DateBirth 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:

Из-за того, что мы вводим еще один параметр для поиска, а составной индекс не включает этого столбца, то скорость не увеличится. Возможно, что она даже уменьшится.

Изменение:

Стоит изменить команду для создания индекса и добавить туда соответствующий столбец LastName.

CREATE INDEX idx\_Employees\_Gender\_DateOfBirth

ON Employees (Gender, LastName, DateOfBirth)