# 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.

***В таблице объявлен индекс: idx\_Employees\_TabNum, индексирующий столбец 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?

***Хорошо:***

* ***потому что индексация по этому столбцу позволяет быстро находить запись с указанным значением поля TabNum = 1***
* ***Использования индекса типа int позволит быстро сравнивать строки***

***Плохо:***

* ***Если в таблице много записей с одинаковым значением поля TabNum. использование индекса не даст ускорения, а может даже замедлить из-за использования дополнительной структуры данных.***
* ***Придется считывать всю строку целиком для получения LastName, FirstName, Email, можно хранить эти данные в самом индексе с помощью include(), если этот запрос того стоит.***

***Что можно сделать:***

* ***Проверить целесообразность использования индексации:***
  1. ***Убедиться в том, что выборка данных по данному полю действительно является критическим узким местом приложения***
  2. ***размер таблицы Employees не превышает определенного порога, после которого использование индекса становится неэффективным.***
  3. ***Индекс на поле TabNum создан и настроен правильно, чтобы запрос мог его использовать.***
* ***Сделать его PERSISTED***
* ***Сделать его кластерным, чтобы не тратить время на чтения данных с диска, а получать их прямиком из индекса***
* ***Сделать его покрывающим, указав в нем через include() LastName, FirstName, Email***

# 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

***Влияние:***

* ***По эффективности уступает TabNum, так как тип datetime2 сложнее в сравнении, чем int, следовательно замедлит, но это все равно эффективнее, чем при отсутствии индексации***
* ***DATEPART() возвращает int, значит сравнение будет проходить так же быстро, как и для TabNum***
* ***Добавление индекса на столбец DateOfBirth позволит ускорить выполнение запроса, так как оптимизатор запросов сможет использовать индекс для поиска строк, соответствующих условию WHERE по столбцу DateOfBirth.***

***Что можно сделать:***

* ***Сделать индекс покрывающим и хранить только одно поле, а не все 27. Count считать по выбранному полю***
* ***В запросе можно заменить операцию DATEPART сравнением дат начала и конца года типа:***

***Или***

# 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

***Влияние:***

* ***Добавление индекса на столбцы LastName и DateOfBirth ускорит выполнение запроса, так как БД сможет быстрее найти нужные записи.***

***Изменение:***

* ***Можно изменить порядок столбцов в индексе на (DateOfBirth, LastName), чтобы БД могла эффективнее использовать его при выполнении запроса.***
* ***Также можно добавить условие на ограничение количества записей, для избежание сканирования всей таблицу:***

***SELECT TOP 1 \****

***FROM Employees***

***WHERE LastName = N'Иванов' AND DateOfBirth IS NOT NULL***

***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'

***Скорость:***

* ***Добавление индекса на столбцы LastName и DateOfBirth в таблице Employees может значительно ускорить выполнение запросов, которые фильтруют данные по этим столбцам. Например, в запросе для нахождения всех сотрудников, у которых фамилия начинается на «А» и которые родились после 01/01/2000 года, SQL Server может использовать индекс для быстрого поиска подходящих строк в таблице.***
* ***Индексация по полю DateOfBirth позволит быстро найти все записи по условия WHERE, скорость запросы увеличится***

***Изменение:***

* ***Можно уменьшить фрагментацию, убрав ненужный LastName.***
* ***Если данные изменяются очень часто, то можно рассмотреть использование индекса с периодической перестройкой или реорганизацией индекса, что менее затратный процесс, чем перестройка.***
* ***Также можно изменить запрос, например, добавив еще один фильтр на поле FirstName, что может уменьшить число записей и ускорить запрос.***

# 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?

***Индексация только по полям Gender, DateOfBirth не даст существенного прироста в скорости, так как базе данных все равно приходится сканировать таблицы, для нахождения строк, удовлетворяющим условию Gender = 'M' и LastName = N'Иванов'***

***Чтобы ускорить запрос нужно добавить в индекс поле LastName***

***CREATE INDEX idx\_Employees\_Gender\_DateOfBirth ON Employees (Gender, DateOfBirth, LastName)***