


-- Задание. Думаем какие запросы у вас будут в базе и добавляем для них индексы.   
 Проверяем, что они используются в запросе.

/\* В рамках данного задания обратим внимание на индексы:

1. Кластеризованные индексы таблиц (в большинстве случаев они же - первичные ключи)
2. Индексы для внешних ключей
3. Индексы XML

Краткое описание и файлы проекта находятся здесь:

<https://github.com/sburovsky/-otus-mssql-2020-02-burovsky/tree/master/Project>

в том числе, скрипты по созданию всех таблиц и индексов:


<https://github.com/sburovsky/-otus-mssql-2020-02-burovsky/tree/master/Project/OtusProjectDDL.sql> 


Схема проекта:

<https://dbdesigner.page.link/EyBwQYZnkuVvwxp6>

В частности, в рамках проекта добавлены индексы:

[PK\_Learning\_Lessons] - кластеризованный индекс для таблицы занятий

[PK\_Learning\_Subjects] - кластеризованный индекс для таблицы тем

[FK\_Learning\_Schedules\_StudentID\_Peoples\_Students] - внешний ключ в таблице   
 расписаний для связи по ключу с таблицей студентов

[IX\_Peoples\_Students\_FullName] - ключ по полному имени студента в таблице студентов

[UQ\_Materials\_Books\_BookID] - кластеризованный индекс для таблицы книг

[PXML\_Learning\_Lessons\_Literature] - первичный XML-индекс в таблице занятий по полю   
 списка литературы

\*/

Use TutorsWorkspace

-- Создадим представление для вывода подробной информации о расписании

Drop view if exists Learning.ScheduleInfo;  
 GO

Create View Learning.ScheduleInfo

AS

Select

Les.LessonName,  
 Subj.SubjectName,  
 Les.LessonID,  
 Subj.SubjectID,  
 Sched.LessonDate,  
 Sched.StudentID,  
 Pupils.FullName AS StudentName

FROM [Learning].[Schedules] AS Sched

INNER JOIN [Learning].[Lessons] AS Les

ON Sched.LessonID = Les.LessonID

INNER JOIN Peoples.Students AS Pupils

ON Sched.StudentID = Pupils.PersonID

LEFT JOIN [Learning].[Subjects] AS Subj

ON Les.SubjectID = Subj.SubjectID

GO

```
-- 1. получим подробную информацию по текущему расписанию студентов по фамилии Некрасов
Declare @StudentName nvarchar(20);
Set @StudentName = N'Некрасов%';
Select
    LessonDate,
    LessonName,
    StudentName,
    SubjectName
FROM Learning.ScheduleInfo
WHERE StudentName Like @StudentName AND LessonDate >= CAST (GETDATE()
AS DATE)
```

/\* в плане запроса IndexSeek1.sqlplan видим Index Seek по индексам:

```
[PK_Learning_Lessons]
[PK_Learning_Subjects]
[FK_Learning_Schedules_StudentID_Peoples_Students]
[IX_Peoples_Students_FullName]
*/
```

-- 2. Получим информацию о литературе по идентификатору занятия из XML- и JSON- полей нескольких таблиц:

```
Declare @LessonID int;
Set @LessonID = 15;
SELECT
    B.ISBN AS ISBN,
    B.Author AS Author,
    B.BookName AS BookName,
    JSON_VALUE([ExtendedInfo], '$.Publishing') AS Publishing,
    JSON_VALUE([ExtendedInfo], '$.PrintRun') AS PrintRun,
    T2.Loc.value('@Chapter', 'int') AS Chapter,
    T2.Loc.value('@Paragraph', 'nvarchar(50)') AS Paragraph,
    T2.Loc.value('@Pages', 'nvarchar(20)') AS Pages

FROM [Learning].[Lessons] as l
CROSS APPLY l.[Literature].nodes('/References/Reference') as T2(Loc)
INNER JOIN Materials.Books AS B
    ON T2.Loc.value('@BookID', 'int') = B.BookID
Where l.LessonID = @LessonID
```

/\* в плане запроса IndexSeek2.sqlplan видим Index Seek по индексам:

```
[PK_Learning_Lessons]
[UQ_Materials_Books_BookID]
[PXML_Learning_Lessons_Literature]
```

Делаем вывод: индексы используются.

```
*/
```