

6. Индексы реляционной БД

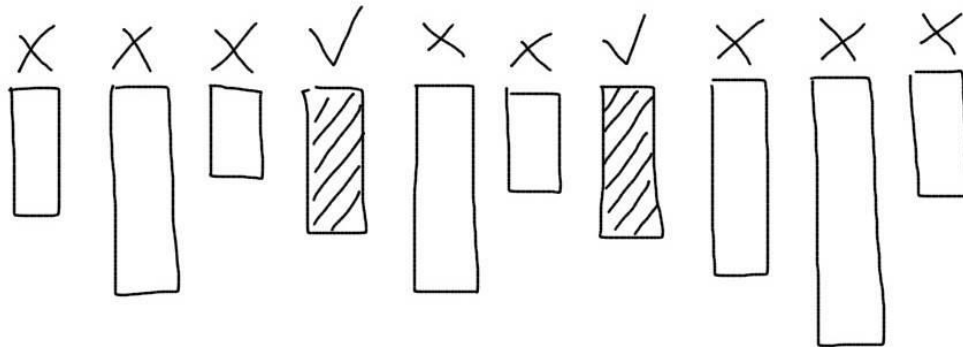
к.т.н., доцент кафедры ИиСП

Лучинин
Захар Сергеевич

Проблема производительности

SELECT * FROM user **WHERE** age = 29

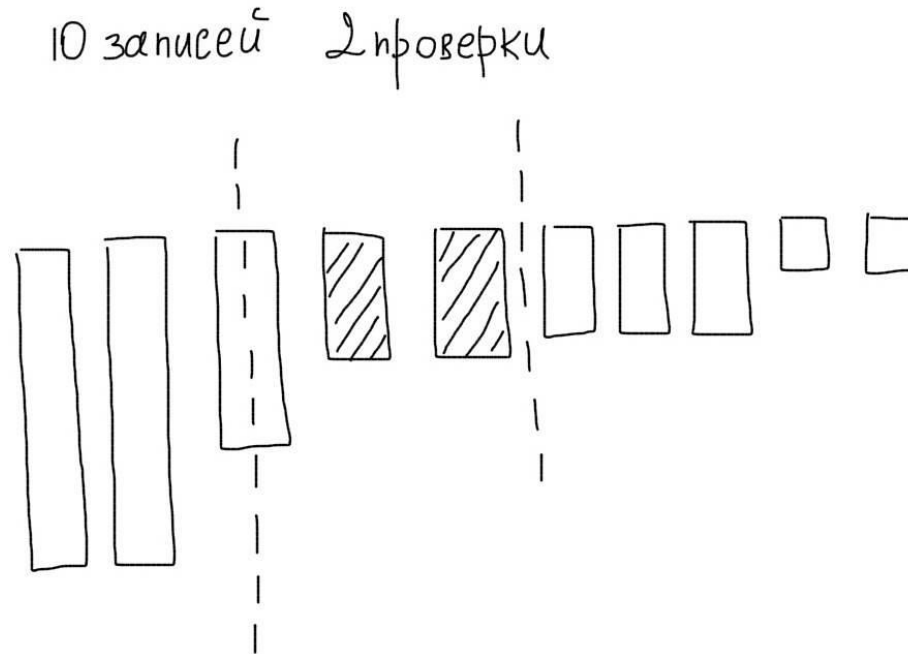
10 записей 10 проверок



Две проблемы при чтении данных:

- Низкая скорость чтения файлов из-за расположения блоков в разных частях диска (фрагментация).
- Большое количество операций сравнения для поиска нужных данных.

Индекс

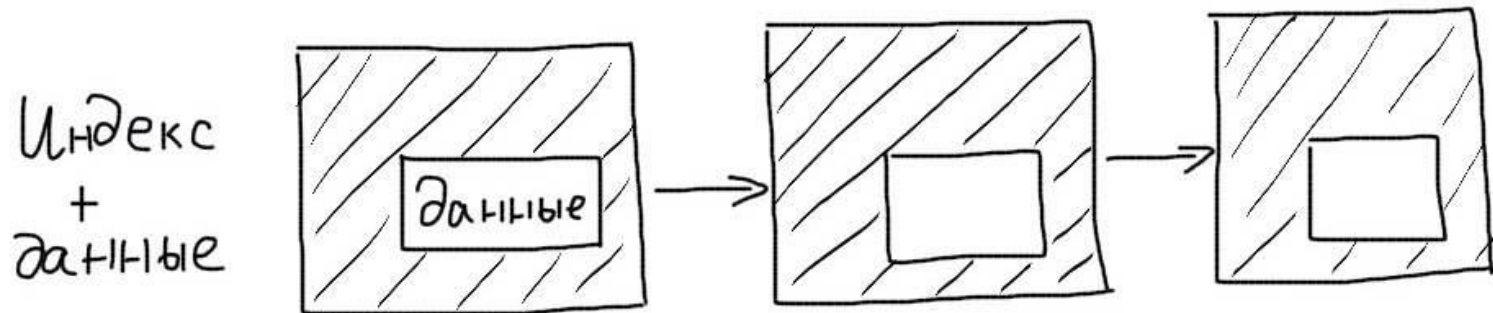


Индекс — объект базы данных, создаваемый с целью повышения производительности поиска данных.

Кластеризованный индекс

Кластеризованный индекс — индекс, при котором значения индекса хранятся вместе с данными, им соответствующими.

Существует только один **кластеризованный индекс** для каждой таблицы, так как строки данных могут храниться в единственном порядке.



Первичный ключ

Первичный ключ (Primary Key) — индекс, который является идентификатором записей в таблице.

- В таблице возможно наличие только одного ограничения по первичному ключу.
- Все столбцы с ограничением PRIMARY KEY должны иметь признак NOT NULL.

```
CREATE TABLE [dbo].[author](  
    [id_author] [int] IDENTITY(1,1) NOT NULL,  
    [first_name] [nvarchar](50) NULL,  
    [last_name] [nvarchar](100) NOT NULL,  
CONSTRAINT [PK_author] PRIMARY KEY CLUSTERED  
(  
    [id_author] ASC  
)
```

Куча

- Если у таблицы нет кластерного индекса, таблица является **кучей**. Записи не имеют определенного порядка хранения.
- При выполнении запроса к таблице, у которой нет индексов, выполняется **сканирование таблицы**. SQL Server не знает, что в таблице существует только одна запись, удовлетворяющая условию и будет просматривать все записи.

```
CREATE TABLE [dbo].[author_heap](  
    [first_name] [nvarchar](50) NOT NULL,  
    [last_name] [nvarchar](100) NOT NULL,  
    [date_of_birth] [datetime] NOT NULL  
)
```

Некластеризованный индекс

В отличие от кластеризованного индекса, листья **некластеризованного индекса** содержат столбцы, по которым определен данный индекс, и указатель на строки с реальными данными в таблице.

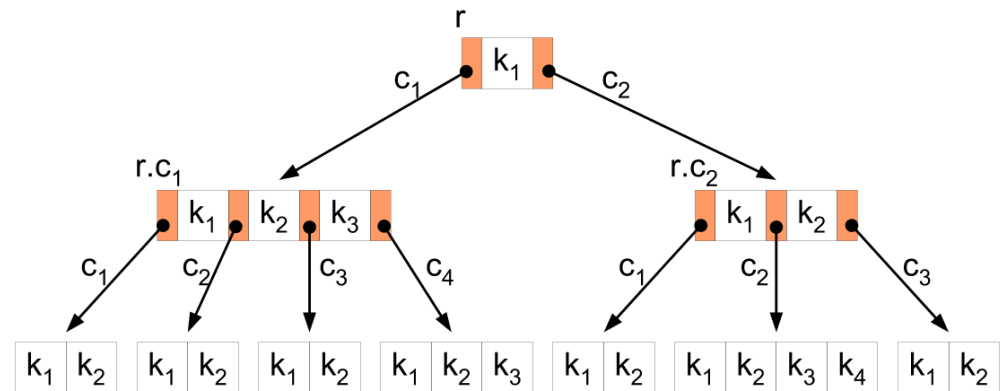
id	name	weight
1	Den	29
2	Alyona	15
3	Vova	89
4	Petro	12



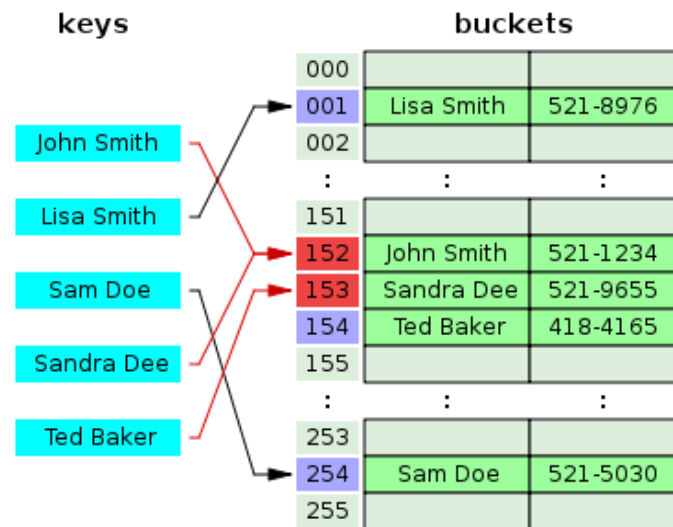
weight	index
12	4
15	2
29	1
89	3

Типы индексов по структуре

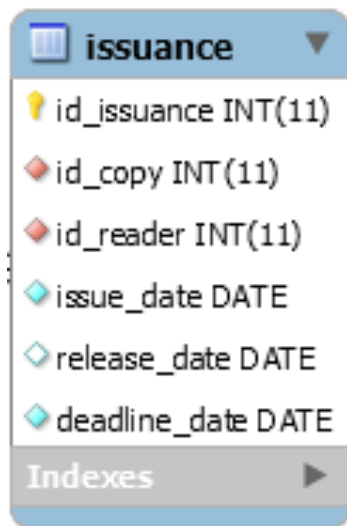
В-дерево - структура данных, дерево поиска. С точки зрения внешнего логического представления, сбалансированное, сильно ветвистое дерево. Часто используется для хранения данных во внешней памяти.



Хеш-таблица - структура данных, реализующая интерфейс ассоциативного массива, она позволяет хранить пары (ключ, значение).



Некластеризованный индекс



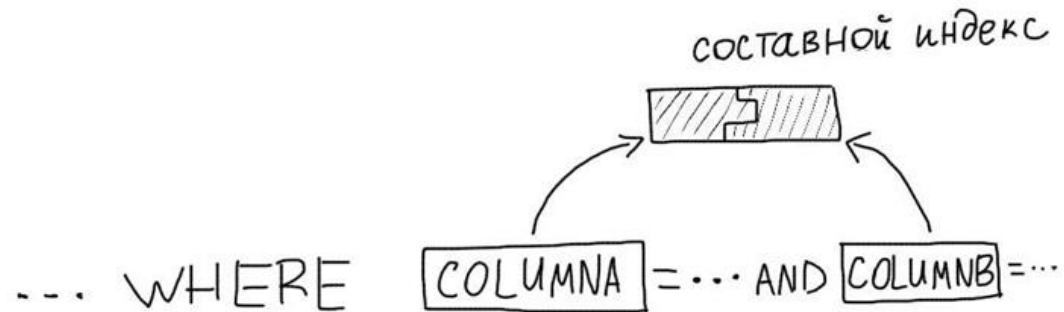
Какие некластеризованные индексы необходимо создать и почему?

```
CREATE NONCLUSTERED INDEX [IX_issuance_id_copy] ON [dbo].[issuance]
(
    [id_copy] ASC
)
```

```
CREATE NONCLUSTERED INDEX [IX_issuance_id_reader] ON [dbo].[issuance]
(
    [id_reader] ASC
)
```

Составной индекс

Составной индекс — это индекс, построенный по нескольким колонкам. Такие индексы имеет смысл использовать, когда в одном запросе фигурируют более одной колонки.

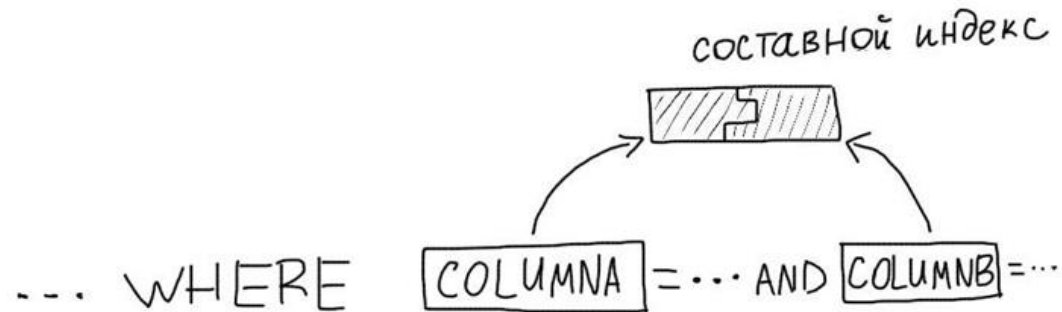


Необходимо создать составной индекс для запроса:

```
SELECT * FROM issuance WHERE id_copy = 394 AND id_reader = 94
```

Составной индекс

Составной индекс — это индекс, построенный по нескольким колонкам. Такие индексы имеет смысл использовать, когда в одном запросе фигурируют более одной колонки.



```
CREATE NONCLUSTERED INDEX [IX_issuance_id_reader_id_copy] ON  
[dbo].[issuance]  
(  
    [id_reader] ASC,  
    [id_copy] ASC  
)
```

При наличии составного индекса нет смысла создавать индекс по `id_reader`

Составной индекс

Все работает точно так же, как и для обычного индекса. Но для значений используются значения всех входящих колонок сразу.

id	name	weight	gender
1	Den	29	male
2	Alyona	15	female
3	Vova	89	male
4	Petro	12	male

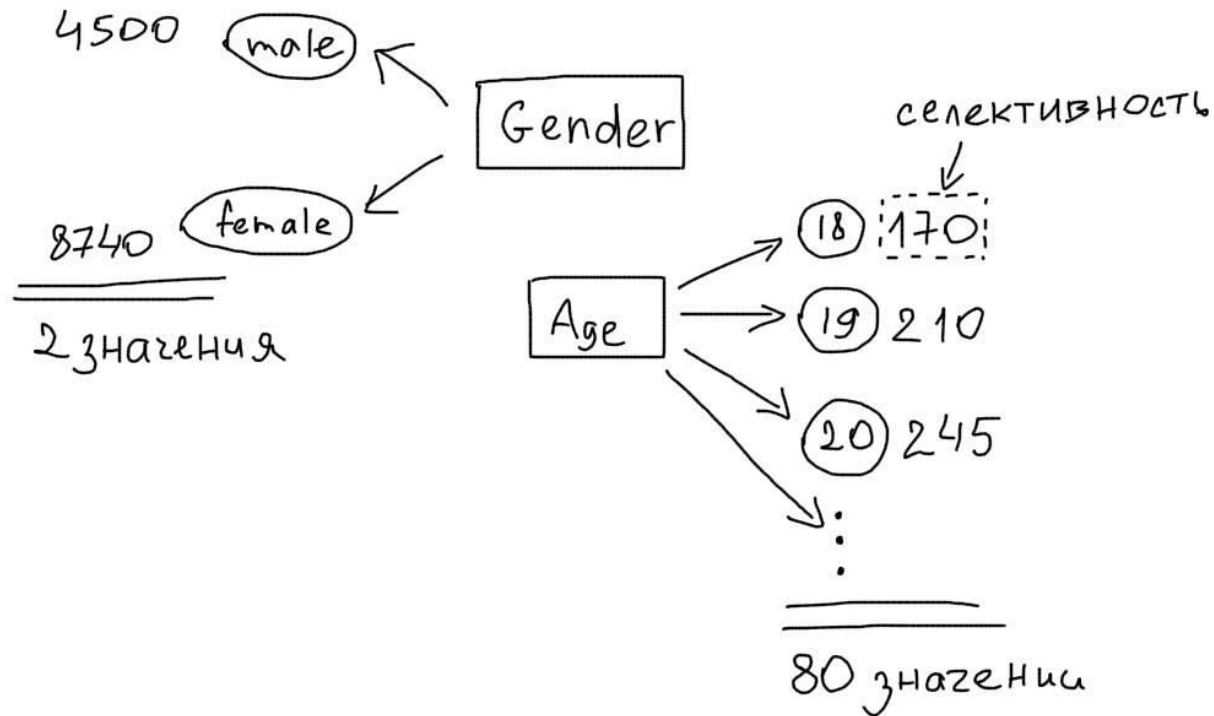


weight_gender	index
12male	4
15female	2
29male	1
89male	3

Очередность колонок в индексе будет играть большую роль.

Селективность индексов

`SELECT * FROM [user] WHERE age = 29 AND gender = 1`



Уникальный индекс

- Необходимо создавать уникальный индекс для колонок, значения в которых должны быть уникальными по всей таблице.

```
SELECT * FROM [user] WHERE email = 'hello@gmail.com'
```

- При поиске данных, SQL остановится после обнаружения первого соответствия. В случае обычного индекса будет обязательно проведена еще одна проверка (следующего значения в индексе).

```
CREATE UNIQUE NONCLUSTERED INDEX [IU_user_email] ON  
[dbo].[user]  
(  
    [email] ASC  
)
```

Покрывающий индекс

Покрывающий индекс - индекс, который позволяет конкретному запросу сразу получить все необходимые данные с листьев индекса без дополнительных обращений к записям самой таблицы.

```
CREATE NONCLUSTERED INDEX [IX_issuance_id_reader-id_copy] ON  
[dbo].[issuance]  
(  
    [id_reader] ASC,  
    [id_copy] ASC  
)
```

```
SELECT id_copy, id_reader FROM issuance WHERE id_reader = 394
```

Индекс с включенными столбцами

- Включенные столбцы хранятся в индексе, но не происходит переиндексация при их изменении

```
SELECT id_copy, id_reader, issue_date FROM issuance WHERE  
id_reader = 394
```

```
CREATE NONCLUSTERED INDEX [IX_issuance_id_reader-id_copy] ON  
[dbo].[issuance]  
(  
    [id_reader] ASC,  
    [id_copy] ASC  
)  
INCLUDE ([issue_date])
```


Полнотекстовый индекс

В отличие от оператора LIKE, такой тип поиска предусматривает создание соответствующего полнотекстового индекса, который представляет собой своеобразный словарь упоминаний слов в полях.

Поисковые характеристики полнотекстового индекса:

- Одно или несколько конкретных слов или фраз (простое выражение)
- Слова, начинающиеся заданным текстом, или фразы с такими словами (префиксные выражения)
- Словоформы конкретного слова (производное выражение)
- Слова или фразы, находящиеся рядом с другими словами или фразами (выражения с учетом расположения)
- Синонимические формы конкретного слова (тезаурус)
- Слова или фразы со взвешенными значениями (взвешенное выражение)

Ограничения индексов

Не будут использоваться индексы

Использование символа подстановки в начале условия поиска исключает для СУБД возможность использования индекса

```
SELECT * FROM book WHERE name LIKE '%Honor'
```

При чтении большого количества кортежей СУБД не использует индекс

```
SELECT * FROM book WHERE id_book > 10
```

Ограничения индексов

Не будут использоваться индексы

Для одной таблицы в запросе может быть использован только один индекс

```
CREATE NONCLUSTERED INDEX [IX_issuance_id_reader] ON  
[dbo].[issuance]  
(  
    [id_reader] ASC  
)
```

```
CREATE NONCLUSTERED INDEX [IX_issuance_id_copy] ON  
[dbo].[issuance]  
(  
    [id_copy] ASC  
)
```

```
SELECT id_copy, id_reader FROM issuance WHERE id_copy = 394  
AND id_reader = 4839
```

Ограничения индексов

Не будут использоваться индексы

Нет возможности использовать индекс при поиске по `id_copy`

```
CREATE NONCLUSTERED INDEX [IX_issuance_id_reader-id_copy] ON  
[dbo].[issuance]  
(  
    [id_reader] ASC,  
    [id_copy] ASC  
)
```

```
SELECT id_copy, id_reader FROM issuance WHERE id_copy = 394
```

Именованние индексов

Именованние индексов таблицы должно подчиняться следующим правилам:

«Имя индекса» = «Префикс индекса» + «_» + наименование таблицы + «_» + перечисление имен столбцов таблицы, участвующих в построении индекса, разделенных символом “-”.

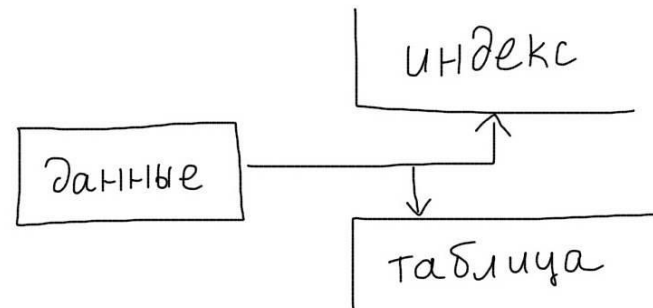
«Префикс индекса» может принимать одно из следующих значений:

- IX = обычный индекс;
- IU = уникальный индекс.

```
CREATE NONCLUSTERED INDEX [IX_issuance_id_copy] ON [dbo].[issuance]  
(  
    [id_copy] ASC  
)
```

Недостатки индексов

- Индексы занимают дополнительное место на диске и в оперативной памяти.
- Замедляются операции вставки, обновления и удаления записей.



Рекомендации при планировании стратегии индексирования

- Для таблиц которые часто обновляются используйте как можно меньше индексов.
- Если таблица содержит большое количество данных, но их изменения незначительны, тогда используйте столько индексов, сколько необходимо.
- Для кластеризованных индексов старайтесь использовать настолько короткие поля насколько это возможно. Наилучшим образом будет применение кластеризованного индекса на столбцах с уникальными значениями.

Рекомендации при планировании стратегии индексирования

- Уникальность значений в столбце влияет на производительность индекса.
- Для составного индекса возьмите во внимание порядок столбцов в индексе.
- По возможности, используйте покрывающие индексы
- Не создавайте индексы, которые не используются
- Создайте некластеризованный индекс на столбцах которые часто используются в ваших запросах в качестве условий поиска в WHERE и соединения в JOIN.
- Старайтесь вставлять или модифицировать в одном запросе как можно больше строк, а не делать это в несколько одиночных запросов.