Программирование и безопасность баз данных мобильных систем

Лекция 8

SQLite
Индексы

Индекс

- Что такое индекс?
- Каким образом применяется?
- В каких случаях создается автоматически?

Индекс

- Что получаем при добавлении индекса?
- Чем за это платим?
- Для каких объектов можно создать индекс?
- Для каких столбцов создаем индекс?

Индекс

- Что такое простой индекс?
- Что такое составной индекс?
- Что такое уникальный / не уникальный индекс?
- Какой порядок значений (ASC, DESC)?
- Что такое индекс покрытия?
- Что такое фильтрующий индекс?

Индекс

- Интеграция индекса с таблицей:
  - кластеризованные
  - некластеризованные

Эффективность индекса

- Эффективность индекса оценивает оптимизатор запросов, основываясь на характеристиках запроса:
  - Селективность
  - Плотность
  - Распределение значений

Селективность

- Селективность – отношение числа выбираемых записей к общему числу записей
- Чем выше селективность – чем больше записей выбирается – тем хуже

Плотность

- Плотность – отношение числа дубликатов значений к общему числу значений
- Лучшая плотность – уникальные значения

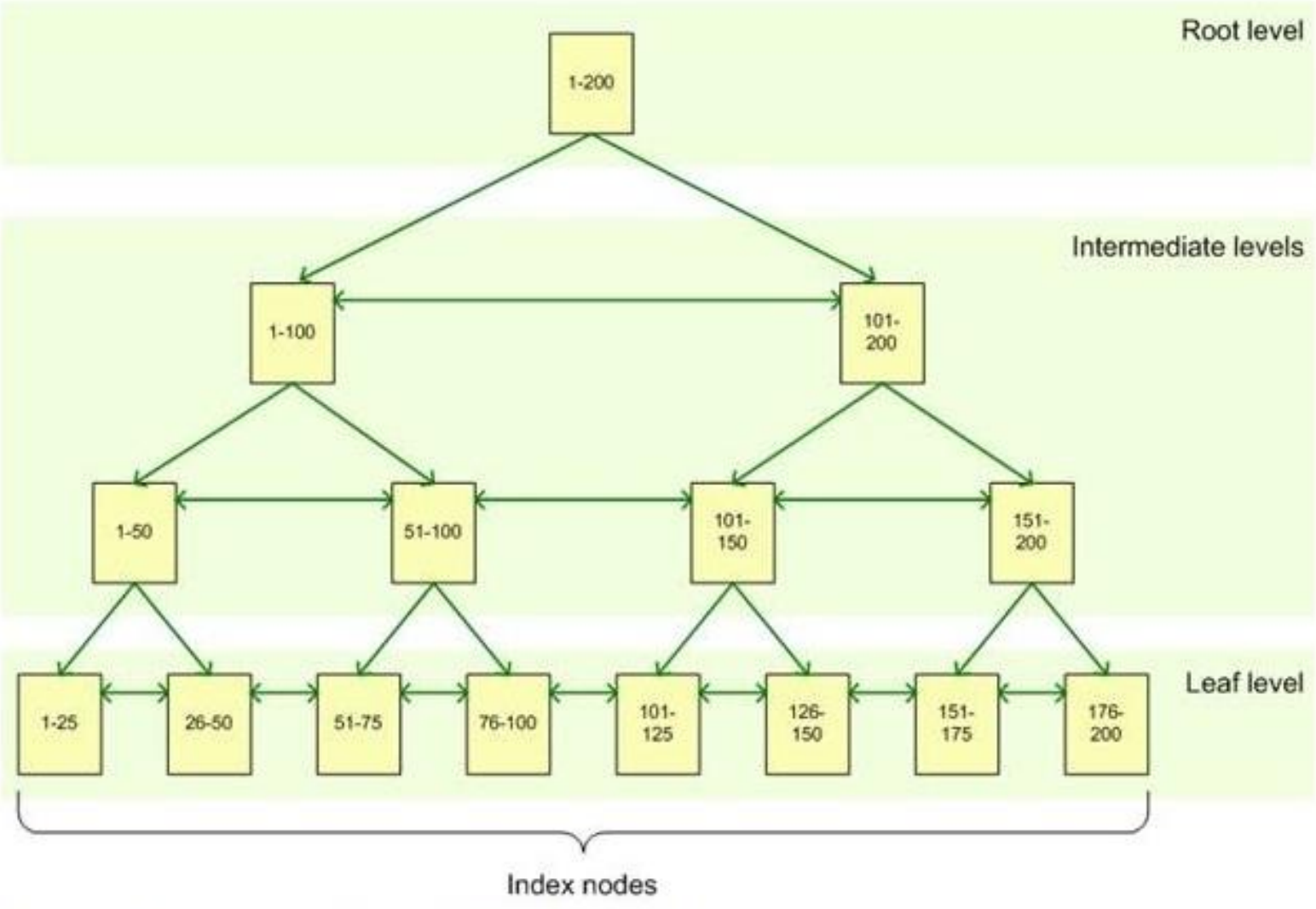
Распределение значений

- Распределение значений – показатель, как значения ключей индекса распределены по всему интервалу возможных значений
- Больше слов на букву А, чем на букву Й

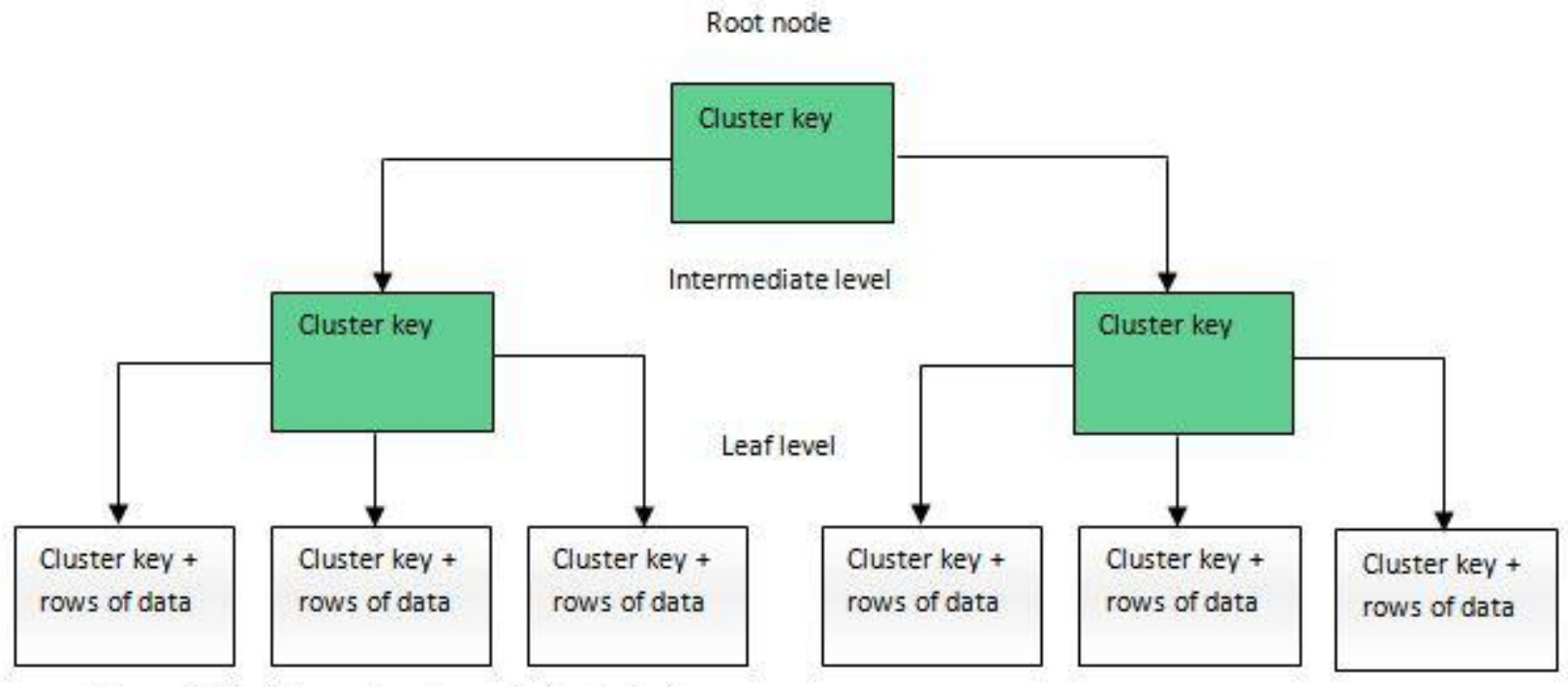
Индекс

- Структура индекса:
  - деревья (tree-index)
    - частичные
    - функциональные
  - двоичные таблицы (bitmap, Oracle)
  - пространственные индексы (spatial)
  - полнотекстовые индексы (full text)
  - XML-индексы
  - колоночные индексы (для OLAP-приложений)

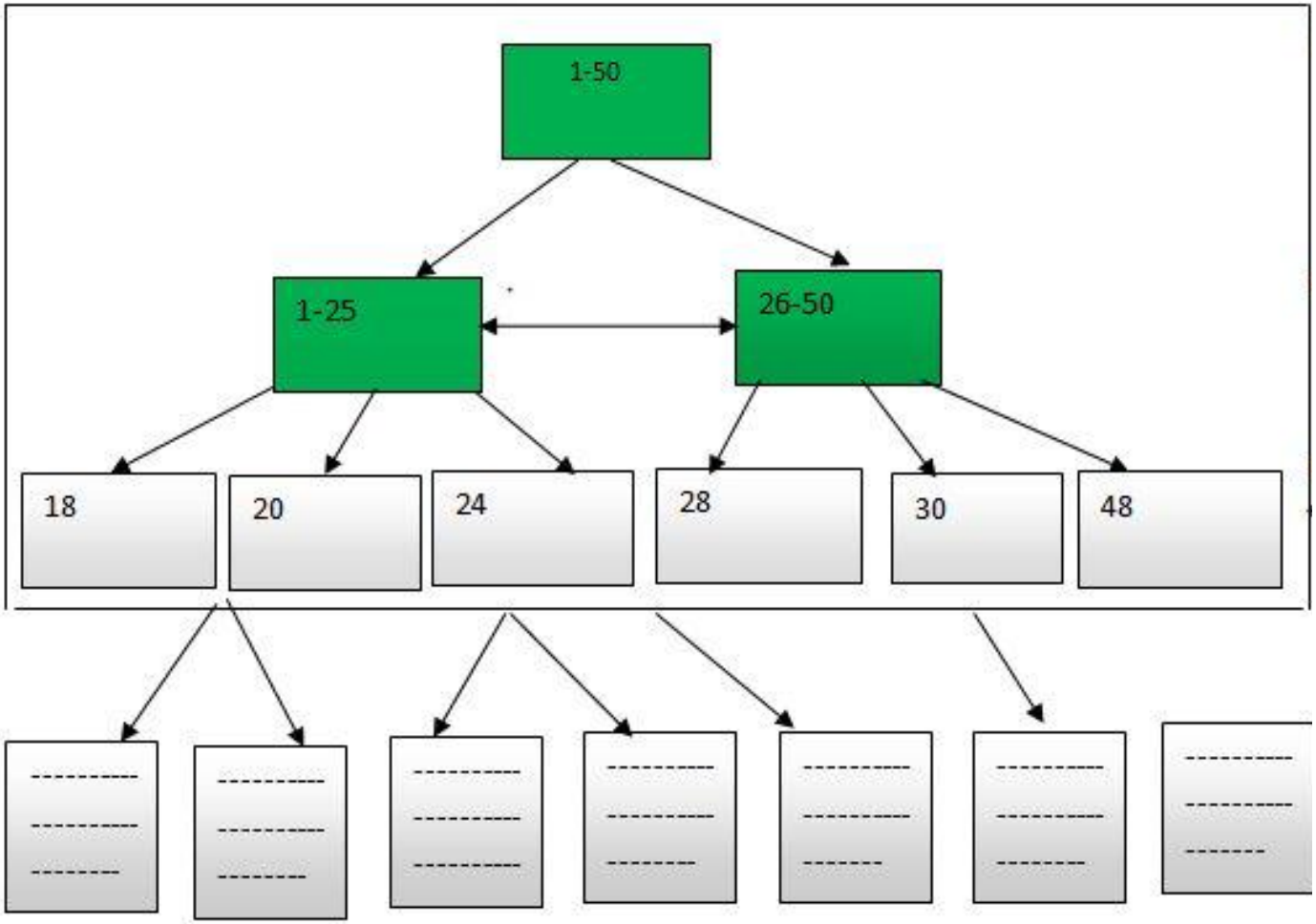
B-tree индекс



B-tree кластеризованный индекс



B-tree некластеризованный индекс

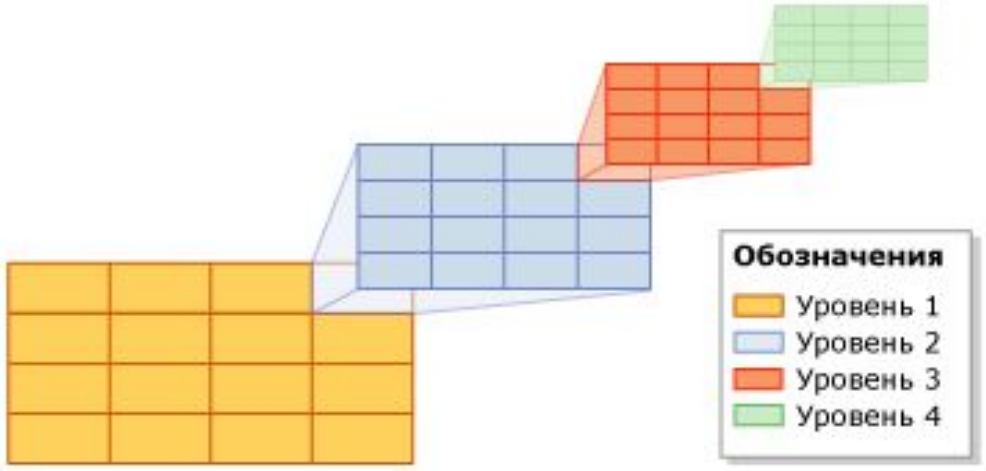


Битовые индексы в Oracle

- Битовый индекс создает битовые карты для каждого возможного значения столбца, где каждому биту соответствует строка, а значение бита 1 (0) означает, что соответствующая строка содержит (не содержит) индексируемое значение
- Предназначен для индексирования столбцов с набором значений
- Не подходит для таблиц с частым обновлением
- Хорошо подходят для хранилищ данных

Пространственные индексы MS SQL Server

```
CREATE SPATIAL INDEX i_spatial_shape
ON market(shape)
USING GEOMETRY_GRID
WITH (BOUNDING_BOX = (xmin=0, ymin=0, xmax=500, ymax=200),
GRIDS = (LOW, LOW, MEDIUM, HIGH),
PAD_INDEX = ON);
```



Ключевое слово	Конфигурация сетки	Число ячеек
LOW	4X4	16
MEDIUM	8X8	64
HIGH	16X16	256

Полнотекстовые индексы MS SQL Server

1	Полнотекстовый индекс строится на основании
2	Пространственный индекс используется
3	Битовый индекс применяется
4	Кластеризованный и некластеризованный индекс

1	полнотекстовый
2	пространственный
3	битовый
4	кластеризованный
4	некластеризованный
1, 2, 3, 4	индекс
1	строится
2	применяется

Полнотекстовые индексы MS SQL Server

- Средства разбиения по словам и парадигматические модули
- Списки стоп-слов
- Файлы тезауруса
- Фильтры

XML индексы в MS SQL Server

- Первичный XML-индекс:
  - Индексируются все теги, значения и пути
  - Используется для возвращения скалярных значений или поддеревьев
- Вторичные XML-индексы:
  - FOR PATH —по структуре документа
  - FOR VALUE —по значениям элементов и атрибутов столбца XML
  - FOR PROPERTY —для поиска по свойствам
- Не могут быть составными
- Не могут быть кластеризованными

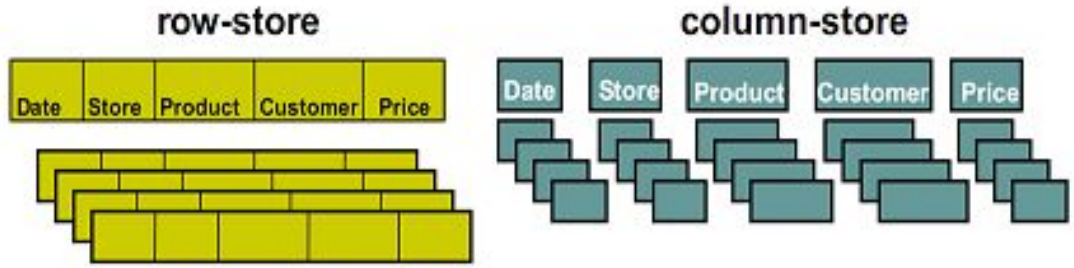
Индексы XML в MS SQL Server

```
CREATE PRIMARY XML INDEX index_xml_column ON xmltab(xml_column);


CREATE XML INDEX i_xmlcolumn_path ON xmltab(xml_column)
USING XML INDEX index_xml_column FOR PATH;
```

Колоночные индексы в MS SQL Server

- Введены с версии 2012
- Данные хранятся по столбцам
- В индекс включаются столбцы, по которым будет производиться поиск
- Позволяют получить значительный выигрыш в производительности для больших массивов данных



Индекс

- Что такое фрагментация индекса?
- В чем заключается обслуживание индекса?

Индекс

- Как принимается решение о применении индекса?
- Каковы критерии принятия решения?
- Что такое hints?

Поиск и сортировка в запросах в SQLite

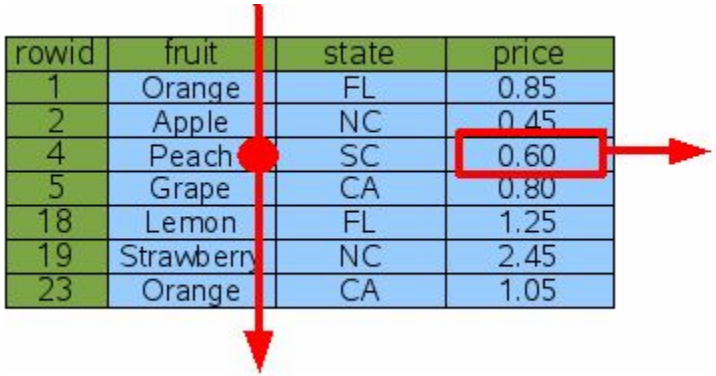
- Полный скан таблицы
- Поиск по RowId
- Поиск по индексу (поиск RowId, переход по RowId)
- Поиск по нескольким условиям (И)
- Поиск по нескольким условиям (ИЛИ)
- Покрывающий индекс (включены все поля поиска)

Поиск и сортировка в запросах в SQLite

- Полный скан таблицы

rowid	fruit	state	price
1	Orange	FL	0.85
2	Apple	NC	0.45
4	Peach	SC	0.60
5	Grape	CA	0.80
18	Lemon	FL	1.25
19	Strawberry	NC	2.45
23	Orange	CA	1.05

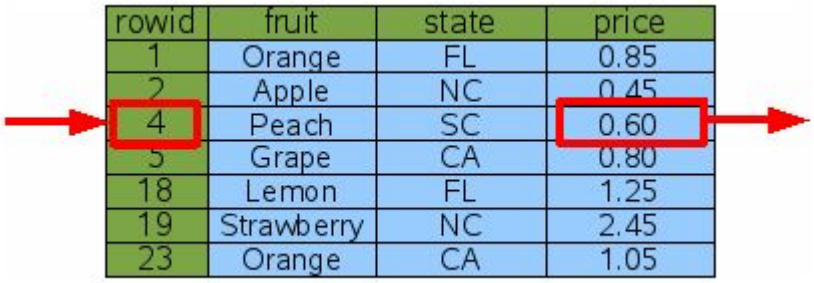
```
SELECT price FROM fruitsforsale WHERE fruit='Peach';
```



Поиск и сортировка в запросах в SQLite

- Поиск по RowId

```
SELECT price FROM fruitsforsale WHERE rowid=4;
```



rowid	fruit	state	price
1	Orange	FL	0.85
2	Apple	NC	0.45
4	Peach	SC	0.60
5	Grape	CA	0.80
18	Lemon	FL	1.25
19	Strawberry	NC	2.45
23	Orange	CA	1.05

Поиск и сортировка в запросах в SQLite

- Поиск по индексу (поиск Rowld, переход по Rowld)

```
CREATE INDEX Idx1 ON fruitsforsale(fruit);
```

fruit	rowid
Apple	2
Grape	5
Lemon	18
Orange	1
Orange	23
Peach	4
Strawberry	19

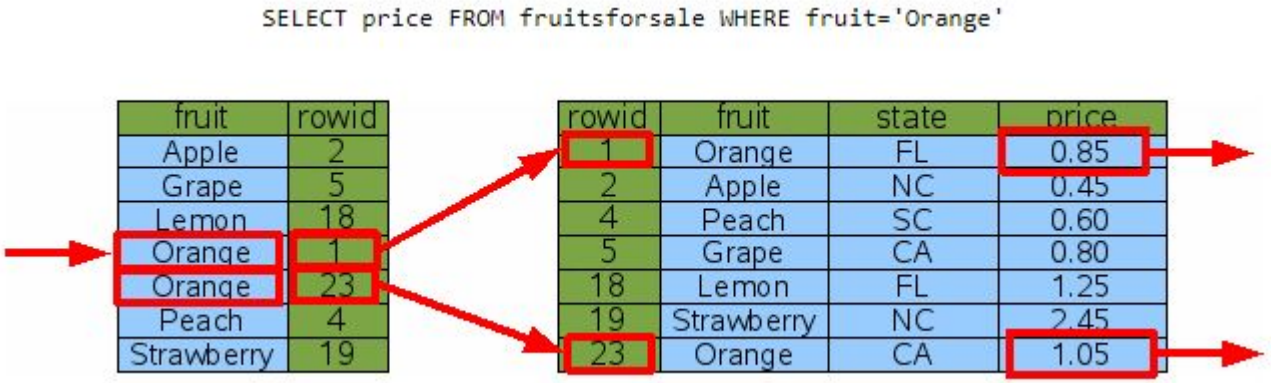
```
SELECT price FROM fruitsforsale WHERE fruit='Peach';
```

fruit	rowid
Apple	2
Grape	5
Lemon	18
Orange	1
Orange	23
Peach	4
Strawberry	19

rowid	fruit	state	price
1	Orange	FL	0.85
2	Apple	NC	0.45
4	Peach	SC	0.60
5	Grape	CA	0.80
18	Lemon	FL	1.25
19	Strawberry	NC	2.45
23	Orange	CA	1.05

Поиск и сортировка в запросах в SQLite

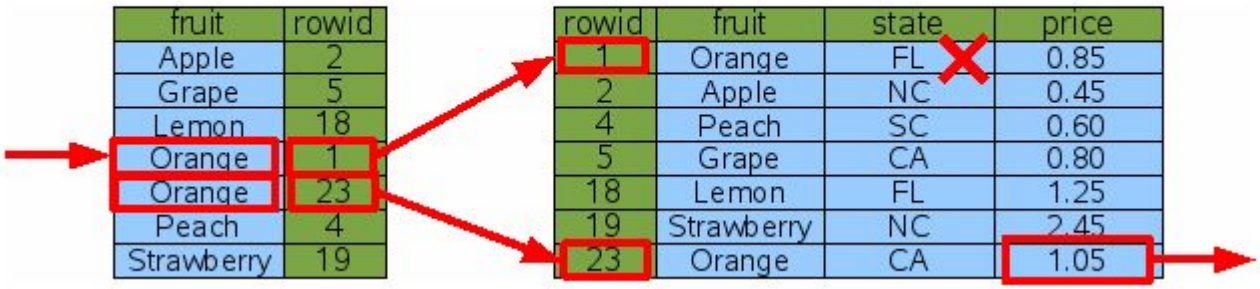
- Поиск по индексу (поиск Rowld, переход по Rowld)



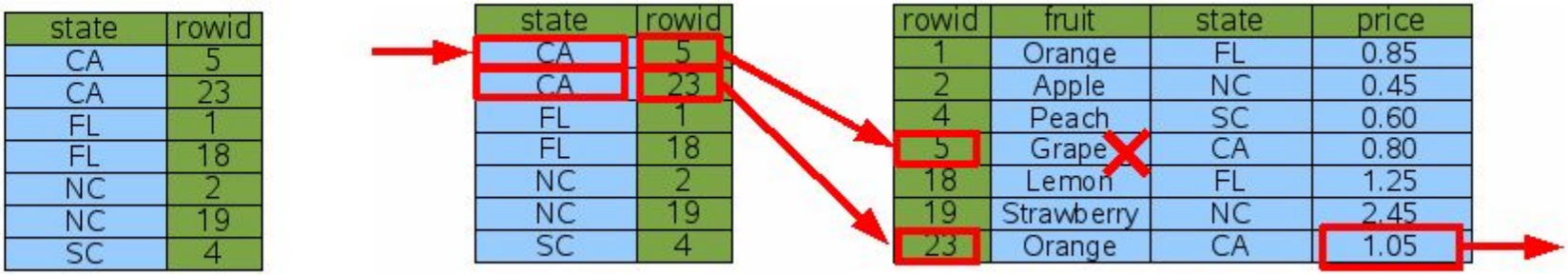
Поиск и сортировка в запросах в SQLite

- Поиск по нескольким условиям (И)

```
SELECT price FROM fruitsforsale WHERE fruit='Orange' AND state='CA'
```




```
CREATE INDEX Idx2 ON fruitsforsale(state);
```



Поиск и сортировка в запросах в SQLite

- Поиск по нескольким условиям (И)

```
CREATE INDEX Idx3 ON FruitsForSale(fruit, state);
```

fruit	state	rowid
Apple	NC	2
Grape	CA	5
Lemon	FL	18
Orange	CA	23
Orange	FL	1
Peach	SC	4
Strawberry	NC	19

```
SELECT price FROM fruitsforsale WHERE fruit='Orange' AND state='CA'
```

fruit	state	rowid
Apple	NC	2
Grape	CA	5
Lemon	FL	18
Orange	CA	23
Orange	FL	1
Peach	SC	4
Strawberry	NC	19

rowid	fruit	state	price
1	Orange	FL	0.85
2	Apple	NC	0.45
4	Peach	SC	0.60
5	Grape	CA	0.80
18	Lemon	FL	1.25
19	Strawberry	NC	2.45
23	Orange	CA	1.05

```
SELECT price FROM fruitsforsale WHERE fruit='Peach'
```

fruit	state	rowid
Apple	NC	2
Grape	CA	5
Lemon	FL	18
Orange	CA	23
Orange	FL	1
Peach	SC	4
Strawberry	NC	19

rowid	fruit	state	price
1	Orange	FL	0.85
2	Apple	NC	0.45
4	Peach	SC	0.60
5	Grape	CA	0.80
18	Lemon	FL	1.25
19	Strawberry	NC	2.45
23	Orange	CA	1.05

Поиск и сортировка в запросах в SQLite

- Покрывающий индекс (включены все поля поиска)

```
CREATE INDEX Idx4 ON FruitsForSale(fruit, state, price);
```

fruit	state	price	rowid
Apple	NC	0.45	2
Grape	CA	0.80	5
Lemon	FL	1.25	18
Orange	CA	1.05	23
Orange	FL	0.85	1
Peach	SC	0.60	4
Strawberry	NC	2.45	19

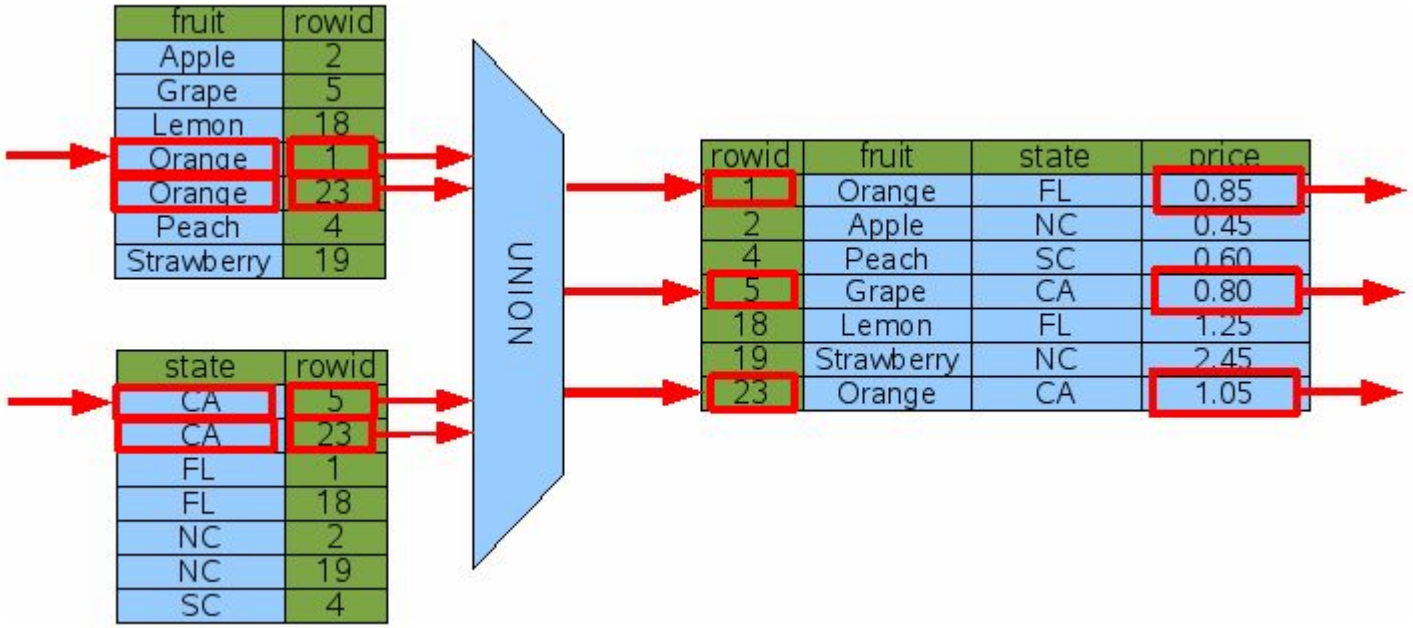
```
SELECT price FROM fruitsforsale WHERE fruit='Orange' AND state='CA';
```

fruit	state	price	rowid
Apple	NC	0.45	2
Grape	CA	0.80	5
Lemon	FL	1.25	18
Orange	CA	1.05	23
Orange	FL	0.85	1
Peach	SC	0.60	4
Strawberry	NC	2.45	19

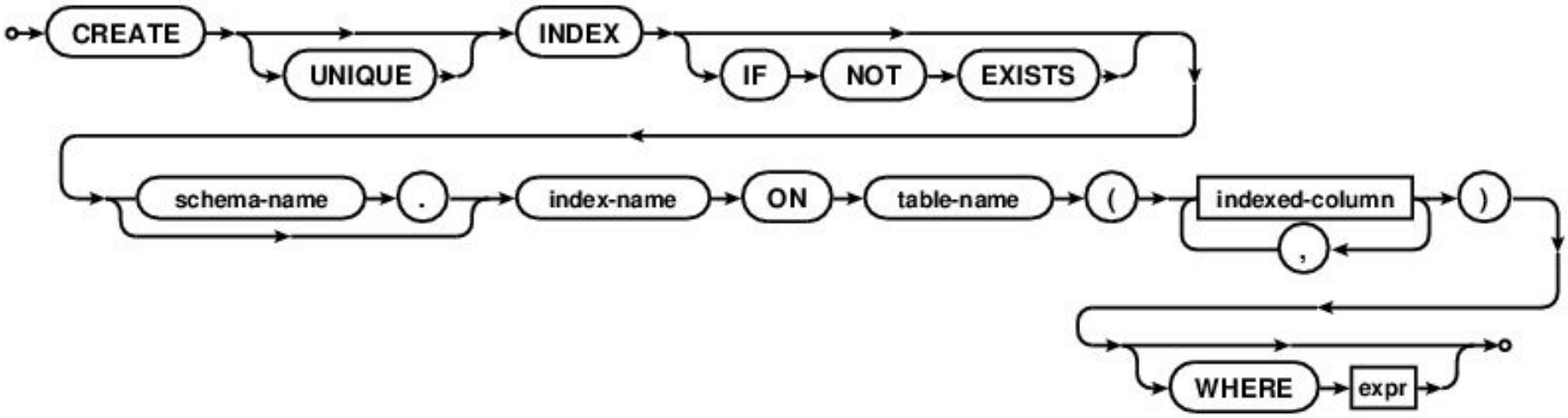
Поиск и сортировка в запросах в SQLite

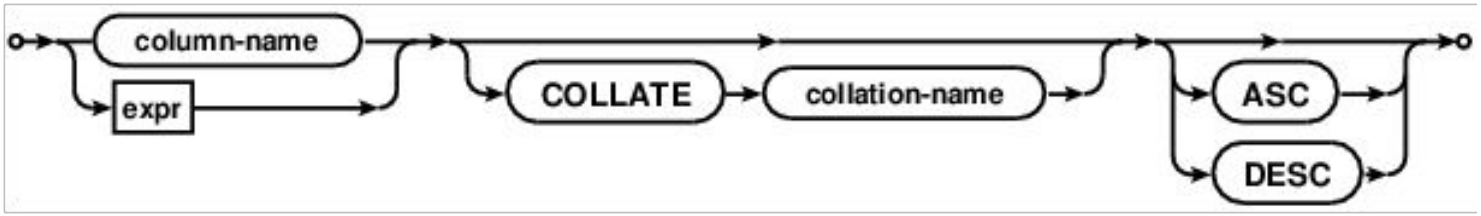
- Поиск по нескольким условиям (ИЛИ)

```
SELECT price FROM FruitsForSale WHERE fruit='Orange' OR state='CA';
```

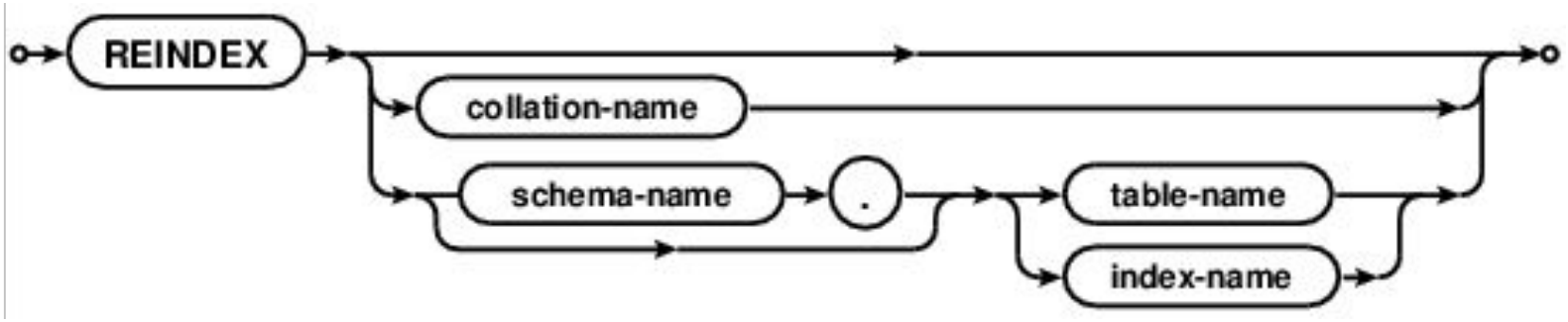


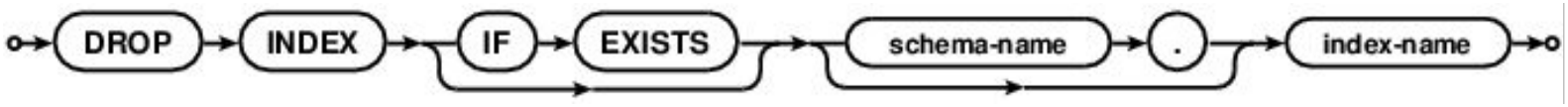
Создание индекса





Пересоздание и удаление индекса





Именование индекса

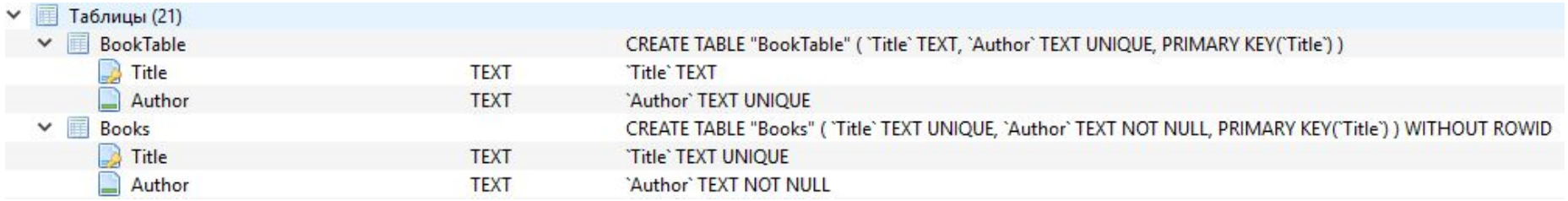
- Стандартный префикс idx_
- Имя таблицы
- Имя столбца (столбцов)

Создание индекса

- Индекс строится на столбцах одной таблицы
- На представлении индекс построить нельзя
- На виртуальной таблице индекс построить нельзя
- Нет ограничений на количество индексов для одной таблицы
- Количество столбцов в индексе ограничено
- Если используется ключевое слово UNIQUE дублирование записей индекса не допускается

Автоматическое создание индекса

- При указании UNIQUE и PRIMARY KEY
- Не могут быть удалены DROP INDEX
- Показаны в sqlite_master
- Индекс по RowID не показывается в sqlite_master



Автоматическое создание индекса

```
1  select * from sqlite_master where type = 'index';
2
```

	type	name	tbl_name	rootpage	sql
1	index	sqlite_autoindex_Table_B_1	Table_B	13	NULL
2	index	sqlite_autoindex_Table_R_1	Table_R	15	NULL
3	index	sqlite_autoindex_search_table_fts_segdir_1	search_table_fts_segdir	21	NULL
4	index	idx_Customers_Cu	Customers	17	CREATE UNIQUE INDEX `idx_Customers_Cu` ON `Cust...
5	index	sqlite_autoindex_BookTable_1	BookTable	25	NULL
6	index	sqlite_autoindex_BookTable_2	BookTable	31	NULL

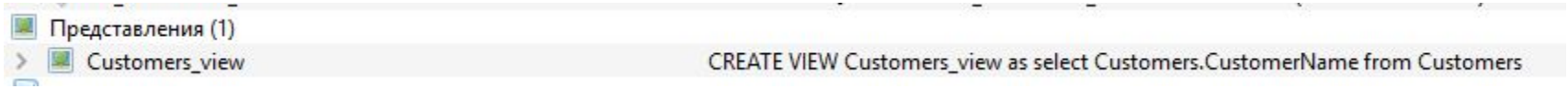
```
drop index sqlite_autoindex_BookTable_1;
```

```
index associated with UNIQUE or PRIMARY KEY constraint cannot be dropped: drop index sqlite_autoindex_BookTable_1;
```

```
reindex  sqlite_autoindex_BookTable_1;
```

```
Запрос успешно выполнен: reindex  sqlite_autoindex_BookTable_1; (заняло 44мс)
```

Индекс на представление




```
CREATE UNIQUE INDEX `idx_Customers_Cu` ON `Customers_view` (
    `CustomerName`
);
```

```
views may not be indexed: CREATE UNIQUE INDEX `idx_Customers_Cu` ON `Customers_view` (
          `CustomerName`
);
```

```
sqlite> .limit
              length 1000000000
           sql_length 1000000000
               column 2000
           expr_depth 1000
      compound_select 500
             vdbe_op 250000000
        function_arg 127
            attached 10
 like_pattern_length 50000
      variable_number 999
        trigger_depth 1000
       worker_threads 0
```

Индексы WHERE

- Частичный индекс - это индекс над подмножеством строк таблицы
- Например, частичный индекс может опускать записи, для которых индексируемый столбец NULL
- Назначение - уменьшение файла базы данных

Индексы WHERE

- Если в индексе используется OR, то в запросе, использующем индекс, может быть указано одно из условий
- Если в индексе используется точное равенство (=a), то запрос будет использовать индекс при таком же условии (between a and a – не подойдет)
- Если в индексе указано IS NOT NULL, то в запросе, использующем <> индекс не применяется

```
CREATE INDEX `idx_Goods_Descr_Price` ON `Goods` (
    `GoodDescr`,
    `Price`
) WHERE GoodDescr is not null or Price > 0;
```

Составные индексы

- Используются для ускорения выполнения запросов по нескольким полям
- Порядок расположения столбцов важен!

Индексы

```
CREATE INDEX `idx_Orders_Date_IdCustomer` ON `Orders` (
    `OrderDate` ASC,
    `idCustomer`    ASC
) WHERE OrderDate > '2016-12-31';
```

```
Запрос успешно выполнен: CREATE INDEX `idx_Orders_Date_IdCustomer` ON `Orders` (
            `OrderDate` ASC,
            `idCustomer`            ASC
) WHERE OrderDate > '2016-12-31'; (заняло 0мс)
```

Индексы с указанием порядка сортировки

- Указание порядка хранения значений ключей оправдано для запроса с предложением ORDER BY

```
create index `idx_Goods_Descr` on 'Goods' ('GoodDescr' desc);
```

```
Запрос успешно выполнен: create index `idx_Goods_Descr` on 'Goods' ('GoodDescr' desc); (заняло 37мс)
```

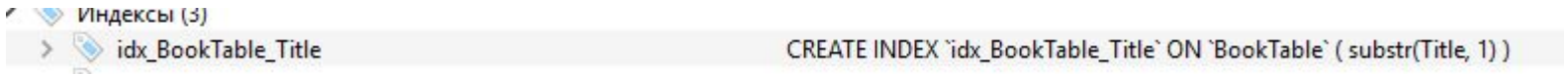
Индексы

- COLLATE определяет последовательность сортировки для текстовых записей

```
CREATE INDEX `idx_Orders_Details` ON `Orders` (
    `OrderDetails`  collate rtrim ASC);
```

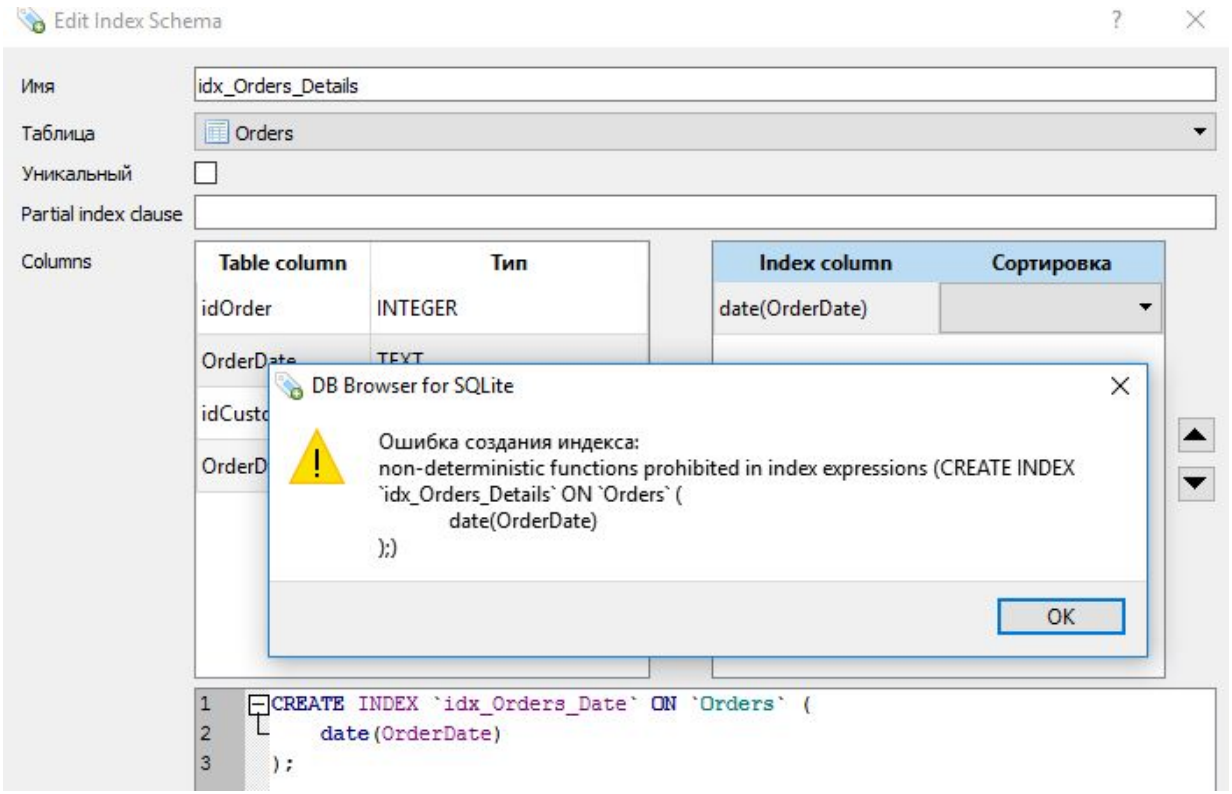
Индексы в выражениях

- В индексе можно использовать выражение от столбца/набора столбцов



Индексы в выражениях

- Выражения в индексе не могут ссылаться на другие таблицы и использовать подзапросы и функции, результат которых может измениться



Оптимизация запросов

- Используются только для оперативного анализа
- ANALYZE
- EXPLAIN
- EXPLAIN QUERY PLAN

ANALYZE

- Сбор статистики о таблицах и индексах
- Хранится в sqlite_stat1
- Оптимизатор запросов может получить доступ к информации и использовать ее, чтобы помочь улучшить выбор планирования запросов

ANALYZE

```
select * from sqlite_stat1;
```

```
analyze;

analyze Goods;

analyze idx_Goods_Descr_Price;
```

	tbl	idx	stat
1	search_table_fts_stat	NULL	1
2	search_table_fts_content	NULL	5
3	Table_B	sqlite_autoindex_Table_B_1	4 1
4	Table4	NULL	3
5	search_table_fts_docsize	NULL	5
6	Table5	NULL	2
7	search_table	NULL	5
8	Table_R	sqlite_autoindex_Table_R_1	3 1
9	Customers	idx_Customers_Cu	7 1
10	Goods	idx_Goods_Descr_Price	9 1 1
11	Goods	NULL	9
12	Orders	idx_Orders	5 1 1
13	Orders	NULL	7
14	OrderDetails	NULL	33
15	Table1	NULL	7
16	search_table_fts_segdir	sqlite_autoindex_search_table_fts_segdir_1	1 1 1
17	Table2	NULL	5
18	Table3	NULL	5

EXPLAIN

- Пошаговое исполнение оператора
- Оператор при этом не выполняется

```
1  CREATE TABLE `Books` (
2      `Title` TEXT UNIQUE,
3      `Author`    TEXT NOT NULL,
4      PRIMARY KEY(`Title`)
5  ) WITHOUT ROWID;
```

```
explain
INSERT INTO Books VALUES('War and Peace','Tolstoy L.N.');
```

	addr	opcode	p1	p2	p3	p4	p5	comment
1	0	Init	0	16	0		00	NULL
2	1	OpenWrite	1	27	0	k(1,)	00	NULL
3	2	Null	0	1	0		00	NULL
4	3	String8	0	2	0	War and Peace	00	NULL
5	4	String8	0	3	0	Tolstoy L.N.	00	NULL
6	5	HaltIfNull	1299	2	2	Books.Title	01	NULL
7	6	HaltIfNull	1299	2	3	Books.Author	01	NULL
8	7	Affinity	2	2	0	BB	00	NULL
9	8	SCopy	2	5	0		00	NULL
10	9	SCopy	3	6	0		00	NULL
11	10	MakeRecord	5	2	4		00	NULL
12	11	NoConflict	1	13	5	1	00	NULL
13	12	Halt	1555	2	0	Books.Title	02	NULL
14	13	IdxInsert	1	4	0		11	NULL
15	14	Close	1	0	0		00	NULL
16	15	Halt	0	0	0		00	NULL
17	16	Transaction	0	1	63	0	01	NULL
18	17	TableLock	0	27	1	Books	00	NULL
19	18	Goto	0	1	0		00	NULL

EXPLAIN QUERY PLAN

- Показывает, каким образом будет проводится поиск в таблице

```
9   explain query plan select GoodDescr from Goods;
10
```

	selectid	order	from	detail
1	0	0	0	SCAN TABLE Goods

```
CREATE INDEX `idx_Goods_Descr_Price` ON `Goods` (
    `GoodDescr`,
    `Price`
) WHERE GoodDescr is not null or Price > 0;
```

```
9   explain query plan
10  select GoodDescr from Goods where GoodDescr ='Computer';
11
```

	selectid	order	from	detail
1	0	0	0	SEARCH TABLE Goods USING COVERING INDEX idx_Goods_Descr_Price (GoodDescr=?)

PRAGMA

- INDEX_INFO (indexname) – возвращает одну строку для каждого столбца ключа в названном индексе
- INDEX_LIST (tablename) – возвращает одну строку для каждого индекса, связанного с данной таблицей
- INDEX_XINFO (indexname) – возвращает информацию о каждом столбце в индексе

PRAGMA

```
1  pragma index_list (Goods);
```

	seq	name	unique	origin	partial
1	0	idx_Goods_Descr_Price	0	c	1

```
CREATE INDEX `idx_Orders` ON `Orders` ( `OrderDate` ASC, `idCustomer` ASC ) WHERE OrderDate>'2016-12-31'
```

```
1  pragma index_info (idx_Orders);
```

	seqno	cid	name
1	0	1	OrderDate
2	1	2	idCustomer

```
1  pragma index_xinfo (idx_Orders);
```

	seqno	cid	name	desc	coll	key
1	0	1	OrderDate	0	BINARY	1
2	1	2	idCustomer	0	BINARY	1
3	2	-1	*NULL*	0	BINARY	0

Итого

- Что такое индекс?
- Каким образом применяется?
- В каких случаях создается автоматически?
- Что такое плотность и селективность запроса?
- Что получаем при добавлении индекса?
- Плюсы и минусы индекса?
- Для каких столбцов создаем индекс?
- Как обслуживать индексы?

Индекс

- Что такое простой индекс?
- Что такое составной индекс?
- Что такое уникальный / не уникальный индекс?
- Какой порядок значений (ASC, DESC)?
- Что такое индекс покрытия?
- Что такое фильтрующий индекс?
- Что такое кластеризованный / некластеризованный индекс?
- Какие виды индексов есть в SQLite?

Вопросы?