

# Индексация в СУБД

# Понятие индекса

Основная проблема в СУБД – это поиск нужных данных за минимальное время

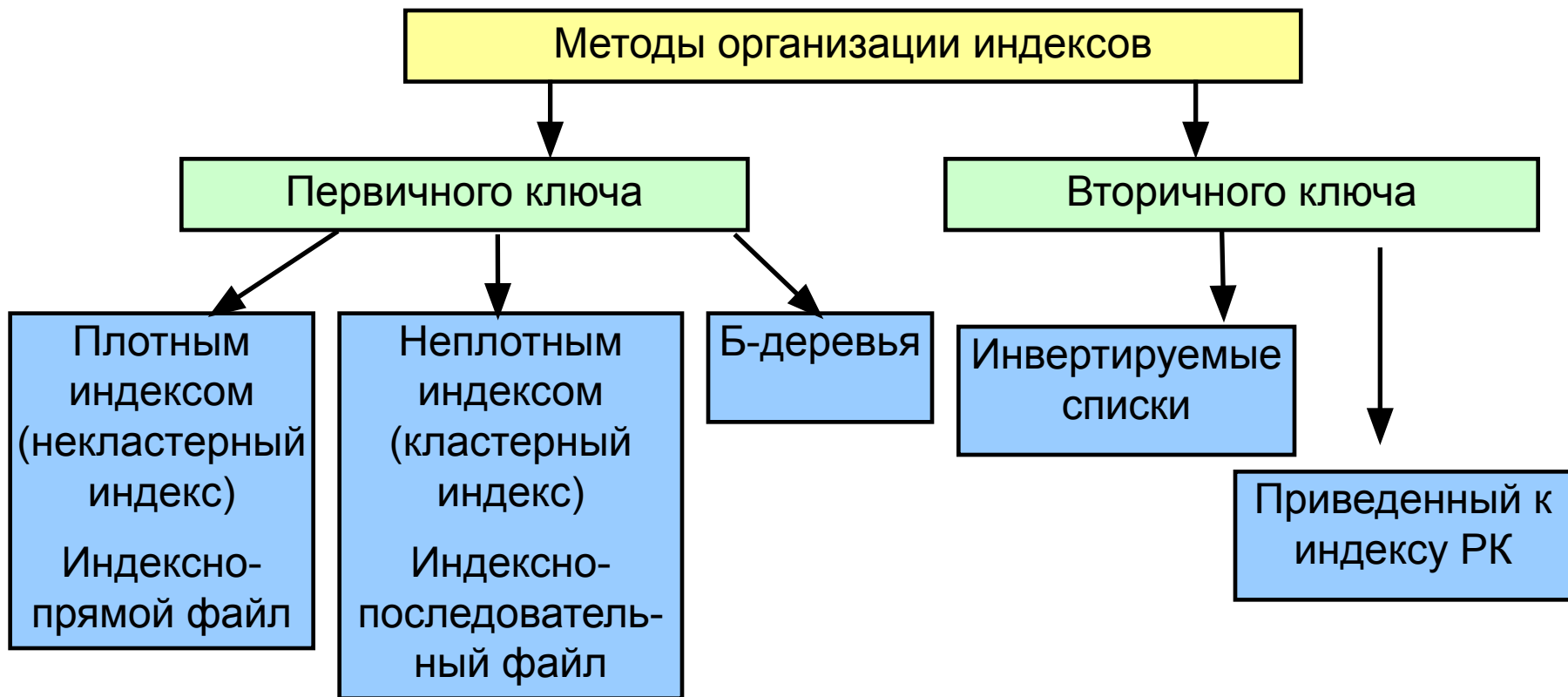
Быстрый поиск может быть выполнен в случае, если данные отсортированы.

Сортировка данных в таблице невозможна, т.к. критериев поиска может быть несколько.

Поэтому для таблицы с данными создаются специальные таблицы для каждого критерия поиска, которые называются **ИНДЕКСЫ**

**Индекс – это структура данных для быстрого поиска записей в таблице по значению ключа**

# Методы организации индекса



# Плотный индекс

## Талица индекса

Содержит индексные записи в отсортированном порядке, расположенные **в блоках**, имеющих изначально свободное пространство.

## Основная таблица

Содержит последовательность записей одинаковой длины в произвольном порядке.

## Индексная запись

<b>Индексный ключ</b>
<b>Номер записи в основном файле</b>

Поиск данных осуществляется по индексному ключу в индексной таблице методом бинарного поиска **по блокам**.

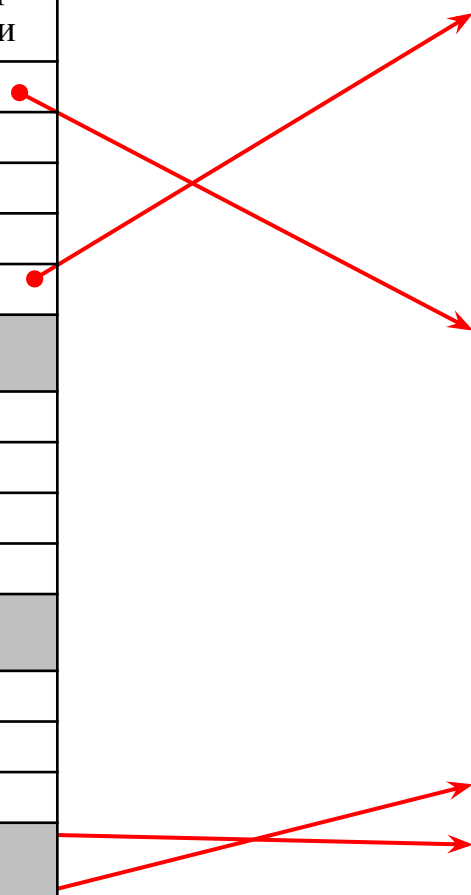
# Плотный индекс

## Таблица индекса

	КодСотр	Номер записи
Блок 1	03	6
	06	5
	63	10
	66	4
	73	1
Блок 2	75	2
	80	3
	81	12
	83	7
Блок 3	84	11
	90	8
	91	9
Блок 4	92	14
	98	13

## Основная таблица

КодСотр	Фамилия	отдел	Адрес
73	Иванов	04	г.Минск ...
75	Степанов	04	г.Минск ...
80	Кустов	21	г.Минск ...
66	Чернов	22	г.Минск ...
06	Трусов	14	г.Минск ...
03	Терпухов	12	г.Минск ...
83	Павлов	04	г.Минск ...
90	Мухина	12	г.Минск ...
91	Артемьев	21	г.Минск ...
63	Васильев	22	г.Минск ...
84	Васильева	04	г.Минск ...
81	Уваров	04	г.Минск ...
98	Шишкин	21	г.Минск ...
92	Удалов	22	г.Минск ...



# Плотный индекс

## 2-я таблица индекса

Фамилия	Номер записи
Артемьев	9
Васильев	10
Васильева	11
Иванов	1
Кустов	3
Мухина	8
Павлов	7
Степанов	2
Терпухов	6
Трусов	5
Уваров	12
Удалов	14
Чернов	4
Шишкин	13

Блок 1

Блок 2

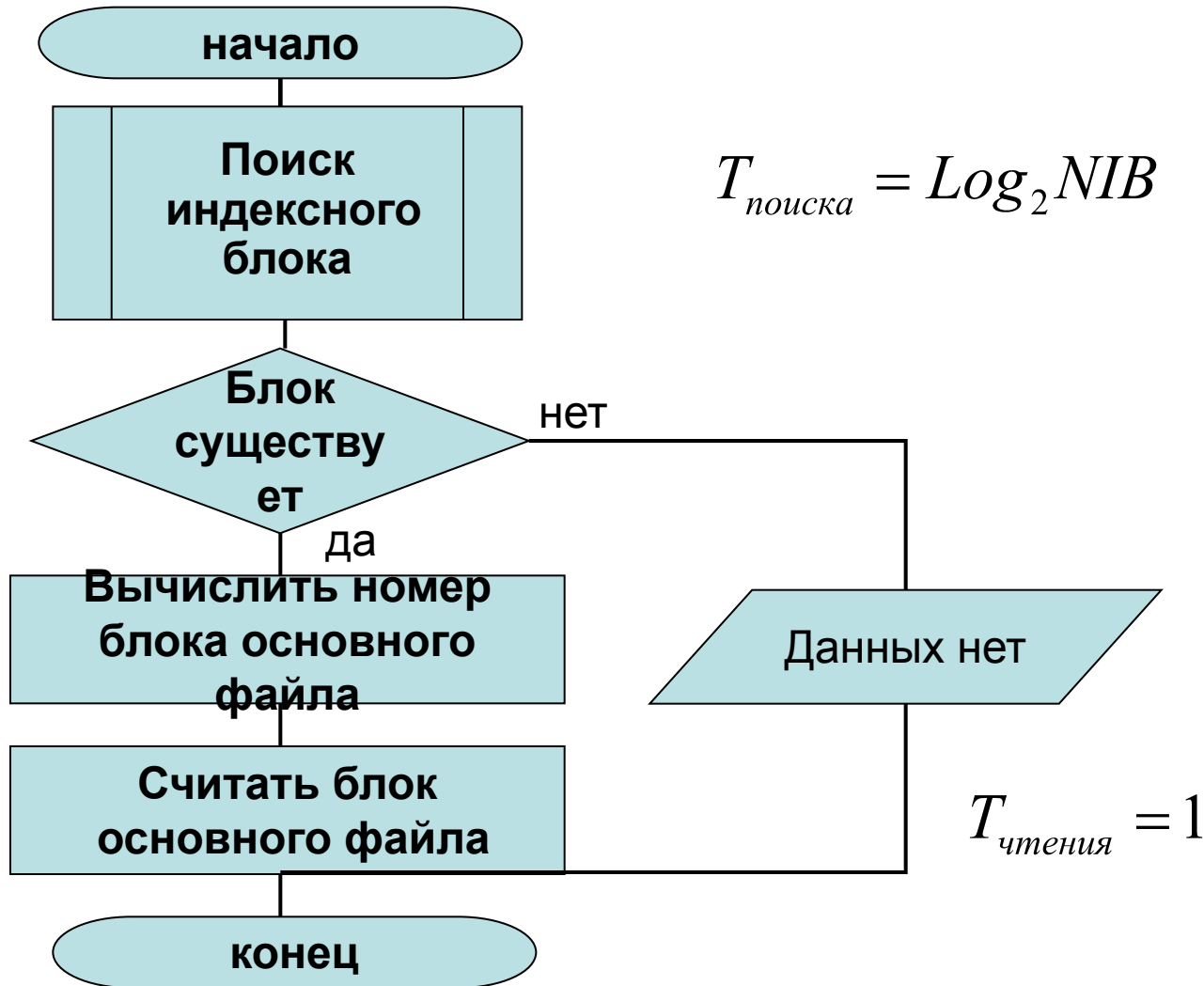
Блок 3

## Основная таблица

КодСотр	Фамилия	отдел	Адрес
73	Иванов	04	г.Минск ...
75	Степанов	04	г.Минск ...
80	Кустов	21	г.Минск ...
66	Чернов	22	г.Минск ...
06	Трусов	14	г.Минск ...
03	Терпухов	12	г.Минск ...
83	Павлов	04	г.Минск ...
90	Мухина	12	г.Минск ...
91	Артемьев	21	г.Минск ...
63	Васильев	22	г.Минск ...
84	Васильева	04	г.Минск ...
81	Уваров	04	г.Минск ...
98	Шишкин	21	г.Минск ...
92	Удалов	22	г.Минск ...

# Плотный индекс

## Алгоритм поиска данных



# Плотный индекс

## Алгоритм добавления записи



$$T_{\text{записи}} = 1$$

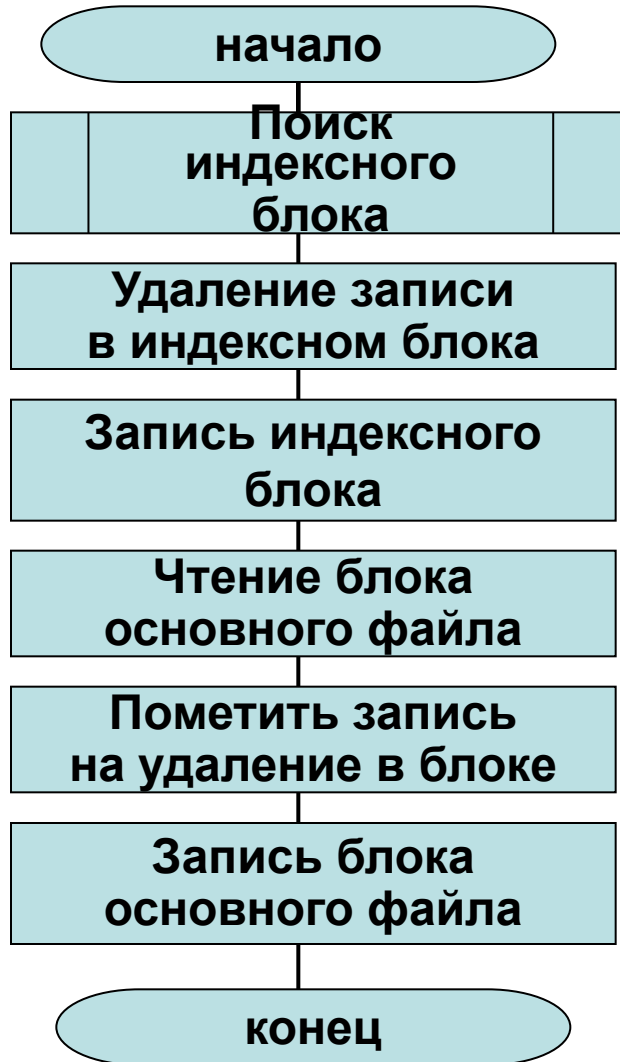
$$T_{\text{поиска}} = \text{Log}_2 NIB$$

$$T_{\text{записи}} = 1$$



# Плотный индекс

## Алгоритм удаления записи



$$T_{поиска} = \log_2 NIB$$

$$T_{записи} = 1$$

$$T_{чтения} = 1$$

$$T_{записи} = 1$$

# Плотный индекс

Оценка времени на выполнение основных операций  
(в максимальном количестве обращений к диску)

без использования индекса

$$T_{\text{поиска}} = \frac{N}{LB / LZ}$$

$$T_{\text{добав}} = 1$$

$$T_{\text{удал}} = T_{\text{поиск}} + 1$$

с использованием индекса

$$T_{\text{поиска}} = \text{Log}_2\left(\frac{N}{LB / LI}\right) + 1$$

$$T_{\text{добав}} = T_{\text{поиск}} + 1$$

$$T_{\text{удал}} = T_{\text{поиск}} + 2$$

где N – количество записей в основной таблице;

LB – размер блока;

LZ- длина записи основной таблицы

где LI– длина записи в индексной таблице;

# Неплотный индекс

## Таблица индекса

Содержит индексные записи в отсортированном порядке, расположенные в блоках.

## Основная таблица

Содержит последовательность записей одинаковой длины в отсортированном порядке по индексному ключу, расположенные в блоках, имеющих изначально свободное пространство.

## Индексная запись

<b>Индексный ключ</b>
<b>Номер блока в основном файле</b>

Поиск данных осуществляется по индексному ключу в индексном файле методом бинарного поиска по блокам.

# Неплотный индекс

Таблица индекса

КодСотр	Номер блока
03	1
73	2
83	3
98	4

Блок 1

Блок 2

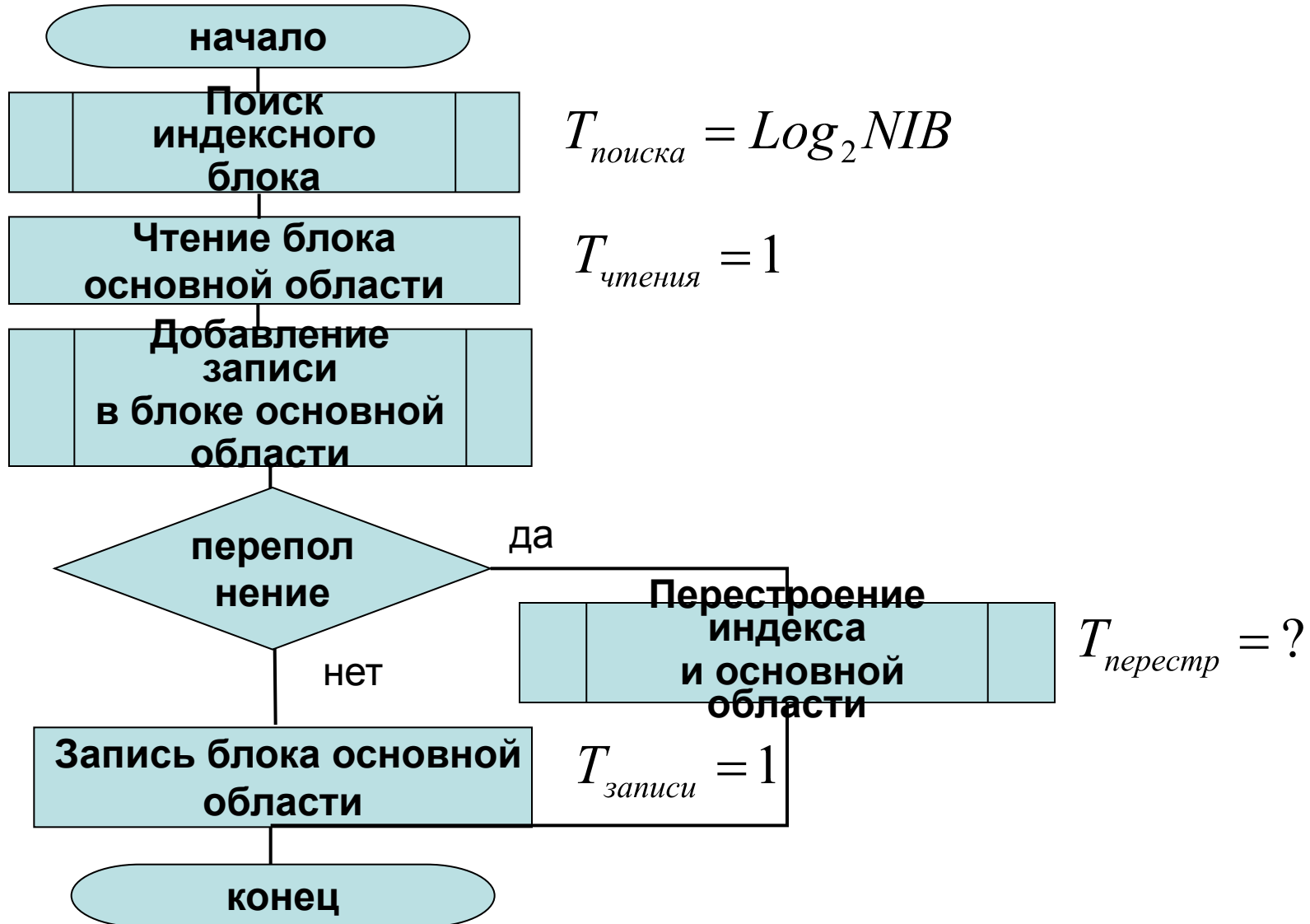
Блок 3

Основная таблица

КодСотр	Фамилия	отдел	Адрес
03	Терпухов	12	г.Минск ...
06	Трусов	14	г.Минск ...
63	Васильев	22	г.Минск ...
66	Чернов	22	г.Минск ...
73	Иванов	4	г.Минск ...
75	Степанов	4	г.Минск ...
80	Кустов	21	г.Минск ...
81	Уваров	4	г.Минск ...
83	Павлов	4	г.Минск ...
84	Васильева	4	г.Минск ...
90	Мухина	12	г.Минск ...
91	Артемьев	21	г.Минск ...
92	Удалов	22	г.Минск ...

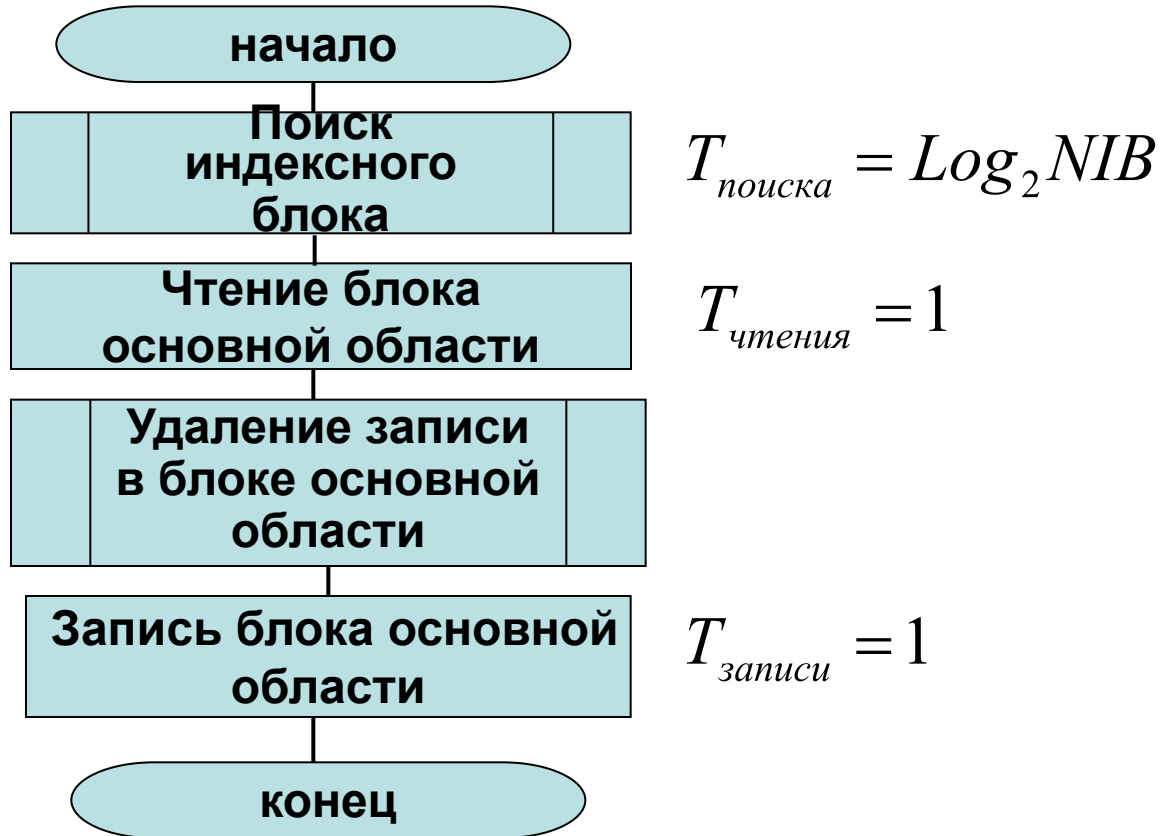
# Неплотный индекс

## Алгоритм добавления записи



# Неплотный индекс

## Алгоритм удаления записи



# Неплотный индекс

Оценка времени на выполнение основных операций  
(в максимальном количестве обращений к диску)

$$T_{\text{поиска}} = \text{Log}_2\left(\frac{N \times LZ \times LI}{LB^2}\right) + 1$$

$$T_{\text{добав}} = T_{\text{поиск}} + 1$$

$$T_{\text{удал}} = T_{\text{поиск}} + 1$$

где  $N$  – количество записей в основной таблице;

$LB$  – размер блока;

$LZ$ - длина записи основной таблицы

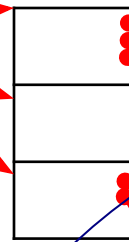
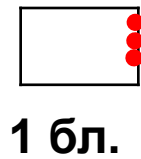
где  $LI$ – длина записи в индексной таблице;

# Индекс Б-дерево

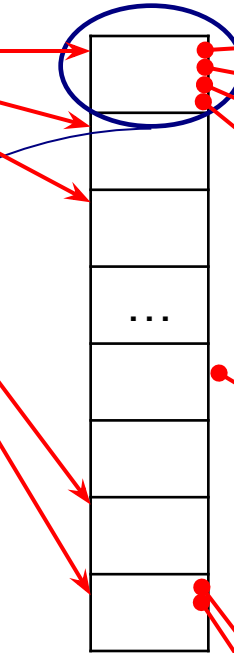
4 уровень      3 уровень      2 уровень      1 уровень

$$NIB_i = \frac{NIB_{i-1} \times LI}{LB}$$

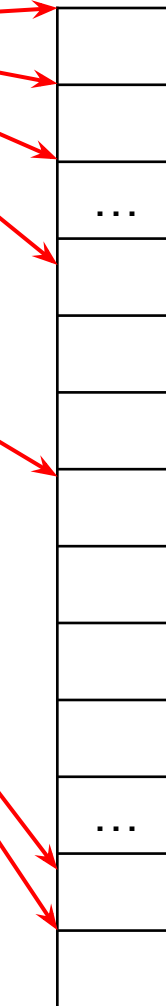
для  $i > 1$



$$172/73 = 3 \text{ бл.}$$



$$12500/73 = 172 \text{ бл.}$$



100000  
записей

12500 бл.

КодСотр	Номер блока
03	1
73	2
83	3
98	4

Со 2-го уровня тип индекса – неплотный индекс



# Индекс Б-дерево

Оценка времени на выполнение основных операций  
(в максимальном количестве обращений к диску)

$$T_{\text{поиска}} = NLI$$

$$T_{\text{добав}} = T_{\text{поиск}} + 1$$

$$T_{\text{удал}} = T_{\text{поиск}} + 1$$

где  $NLI$  – количество уровней индексов.

# Инвертируемые списки

## Индексная таблица 1-го уровня

Содержит индексные записи в отсортированном порядке, расположенные в блоках, имеющих изначально свободное пространство.

## Индексная таблица 2-го уровня

Содержит цепочку номеров записей в отсортированном порядке, расположенных в блоках.

## Основная таблица

Содержит последовательность записей одинаковой длины в произвольном порядке.

## Индексная запись

Индексный ключ
----------------

Номер блока в индексном файле 2-го уровня
---

## Индексная запись

Номер блока в основном файле
---------------------------------

Поиск данных осуществляется по индексному ключу в индексной таблице (файле) 1-го уровня методом бинарного поиска по блокам, с дальнейшим переходом к цепочке записей, расположенных в индексной таблице (файле) 2-го уровня.



# Рекомендации по созданию индексов

## Факторы, определяющие «хороший» индекс

- число столбцов в индексе не более 4-5 (максимум 16 столбцов общей суммой не более 900 байт))

- в индекс включаются не часто обновляемые столбцы

- создание индексов по FK

- создание индексов по столбцам из выражения WHERE часто выполняемых запросов

- не использовать поиски с предикатом LIKE

## Факторы, определяющие «плохой» индекс

- по столбцам, входящих в индекс, не выполняются поиски строк

- слишком много столбцов в индексе

- слишком мало записей в индексируемой таблице

- слишком много индексов

# Оператор создания индекса

СУБД всегда создает индекс для первичного ключа таблицы

Для создания индексов для других полей используется оператор SQL

```
CREATE [ UNIQUE ] INDEX имя_индекса  
ON имя_таблицы (имя_столбца [ ASC | DESC ] [,...])
```

Пример. Создать индекс для таблицы ЗАКАЗ по вторичному ключу

```
CREATE INDEX Заказ_FK_инд  
ON Заказ (MFR, КодТов)
```

В каждой СУБД оператор создания индексов содержит дополнительные предложения специфические для каждой СУБД.

# Оператор создания индекса

## Оператор для создания индексов в T - SQL

**CREATE [ UNIDUE ]**

Создание уникального индекса

**[ CLASTERED | NOCLASTERED ]**

Создание неплотного или плотного индекса

**INDEX *имя\_индекса***

**ON {*имя\_таблицы* | *имя\_представления*}**

**(*имя\_столбца* [ ASC | DESC ] [,...])**

По возрастанию или убыванию

**WITH**

**[ < параметры > [ ,...n ] ]**

**[ ON *имя\_файла\_группы* ]**

Расположение индекса в файлах ОС БД

Резервирование на каждой странице индекса свободного пространства

< параметры > - это

**{ PAD\_INDEX |**

**FILLFACTOR = %запол. |**

Степень заполнения свободного пространства

**IGNORE\_DUP\_KEY |**

Поведение сервера при появлении дублей строк

**DROP\_EXISTING |**

Поведение сервера при наличии индекса

**STATISTICS\_NORECOMPUTE |**

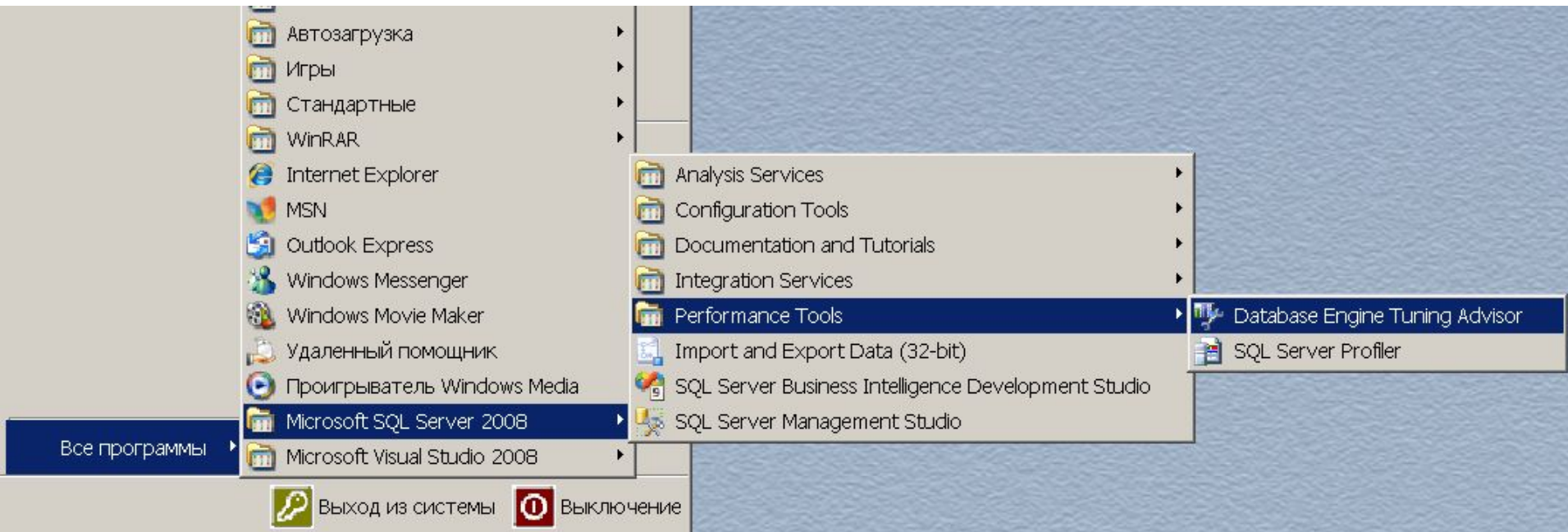
**SORT\_IN\_TEMPDB ...**

...

}

# Инструмент настройки индексов

## Database Engine Tuning Advisor (DTA)



**DTA проводит анализ БД по файлу рабочей нагрузки (workload file) и позволяет определить какие индексы следует поменять или усовершенствовать.**

**Используется DTA через значительный интервал времени работы БД.**