БАНКИ ДАННЫХ

Автор: Емельянов Н. Е.
Правка: Тригуб Н.А.

5.5. Пример объектно-ориентированной БД (XMLDB).
СУБД ИНЕС (для ЕС ЭВМ),
НИКА (для РС)

Цели разработки:

- Объектное проектирование
- Снятие ограничений на размеры
- БД – произвольный граф
- Единый индекс
- Возможность менять схему БД без перезагрузки базы
- Эффективность хранения и доступа

5.5.1. База данных – совокупность двух файлов.

ИмяБД**.dod** *(схема БД)*

ИмяБД**.tree** *(данные)*

- Например. avto.dod и avto.tree

  dod – дерево описания данных (ДОД)

  tree – дерево данных (ДД)

Назначение схемы БД:

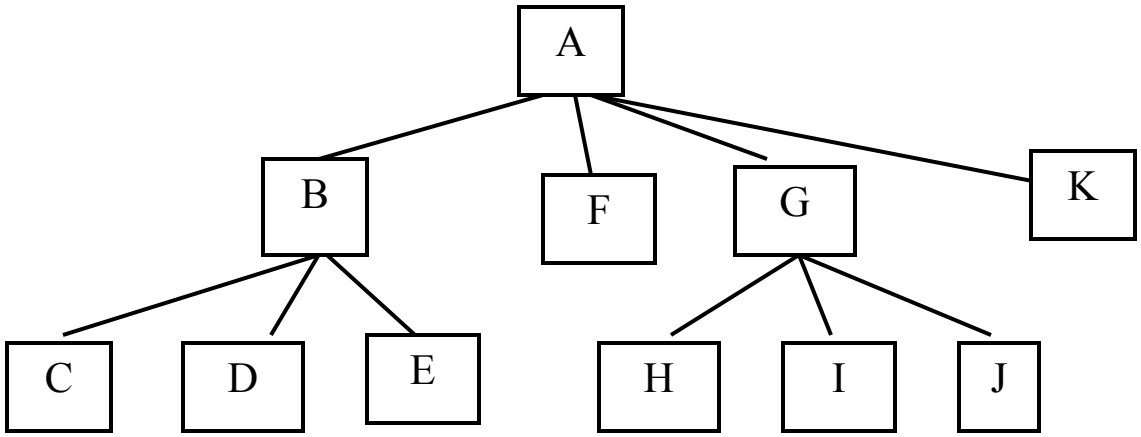
- Шифровка имен.

  Вместо имен вершин (до 256 символов), в дереве данных хранятся шифры (1-2 байта).
- Обеспечение целостности БД.

  Вводить данные в БД можно будет только в соответствие со схемой (структурой объектов и подобъектов и типами терминальных данных).

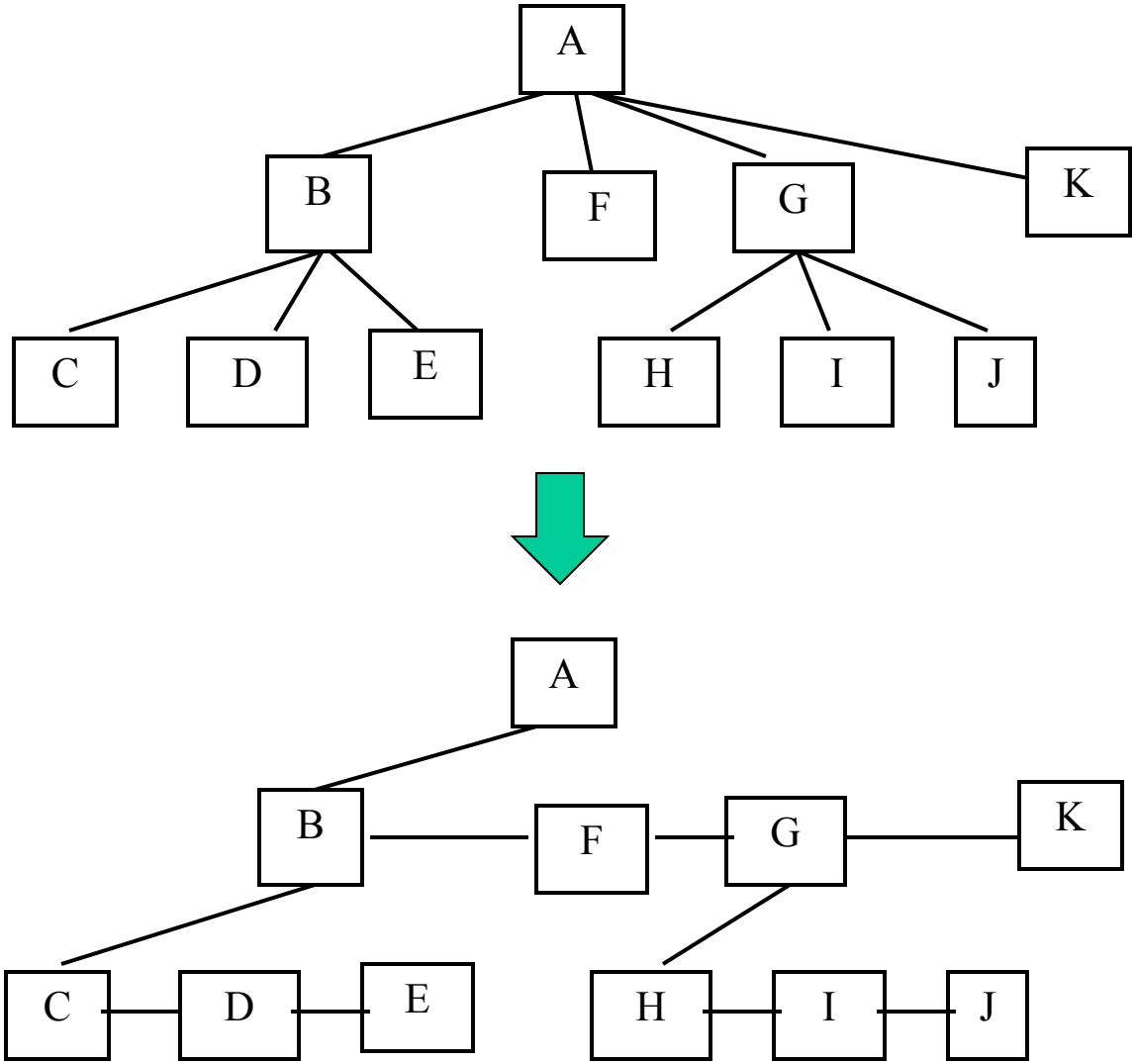
5.5.2. Метод записи деревьев в памяти

Рассмотрим следующую структуру



N-арное дерево

Преобразование N-арной структуры в



Бинарное дерево

Разбивка на страницы

БД разбита на страницы. На каждой странице хранится связный фрагмент дерева. Этот фрагмент превращается из N-арного в бинарный. В каждой вершине хранятся не более двух ссылок: на первую из подчиненных вершин и на следующую из соподчиненных вершин.

Бинарное дерево записывается в память подряд (в порядке левого обхода):

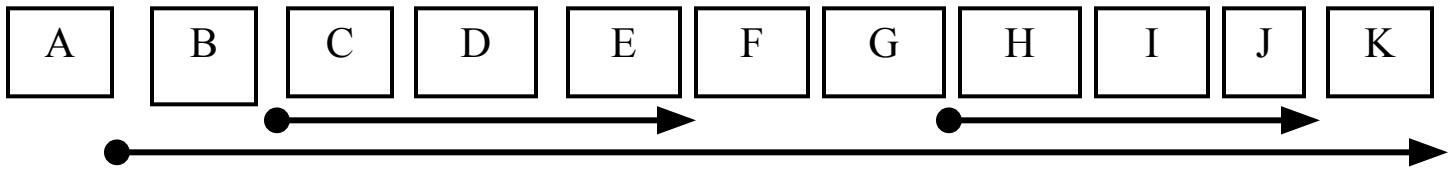
A B C D E F G H I J K

Информация, записанная в каждой вершине

1. Идентификатор данного (шифр)
2. Значение данного, если вершина терминальная
3. Длина данного для терминальных вершин
4. Длина подчиненного поддерева для нетерминальных вершин

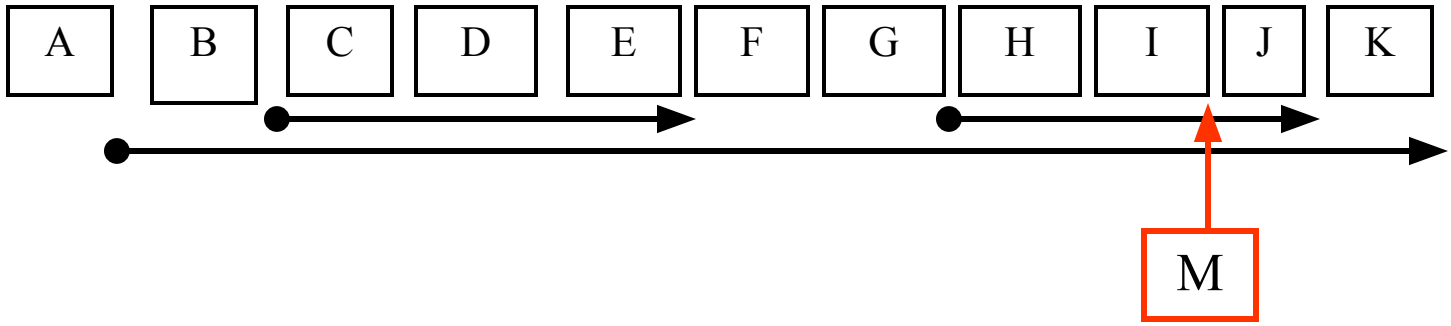
Длина данного - ссылка на следующую вершину дерева в смысле левого обхода.

В каждой не терминальной вершине, вместо данного, хранится длина подчиненного поддерева.



Ввод новых данных

Например, вершины M между I и J (ΔM - длина M).



Если на странице внешней памяти есть сводное место >= ΔM, то сдвигаются данные J и K на ΔM.

В освободившееся место помещается данное M.

Длины поддеревьев A и G увеличиваются на ΔM.

Если на странице свободное место <= ΔM, то включается алгоритм деления страницы.

5.5.3. Метод деления страниц.

При вводе данных в пустую БД открывается 1-я страница, в которую данные записываются в лексикографическом порядке.

Когда емкость 1-й страницы исчерпана, на диске открывается 2-я, и данные распределяются примерно поровну.

Старшие данные остаются на 1-й, младшие - на 2-й.

На 1-й странице организуется справочная, куда выносятся номера страниц и идентификаторы первых данных.

5.5.3. Метод деления страниц (продолжение 1)

Идентификатор следующего вводимого данного сравнивается с идентификаторами справочной.

Вызывается соответствующая страница и новое данное записывается в нее.

.

5.5.3. Метод деления страниц (продолжение 2)

Такая организация памяти позволяет в любое место вставить произвольное количество новых данных.

При переполнении одной из уже существующих страниц открывается следующая страница, куда пересылаются младшие данные; в справочную заносится старший ключ новой страницы.

При переполнении справочной возникает справочная второго уровня, в строках которой хранятся ссылки на справочные первого уровня. В дальнейшем - справочные более высоких уровней.

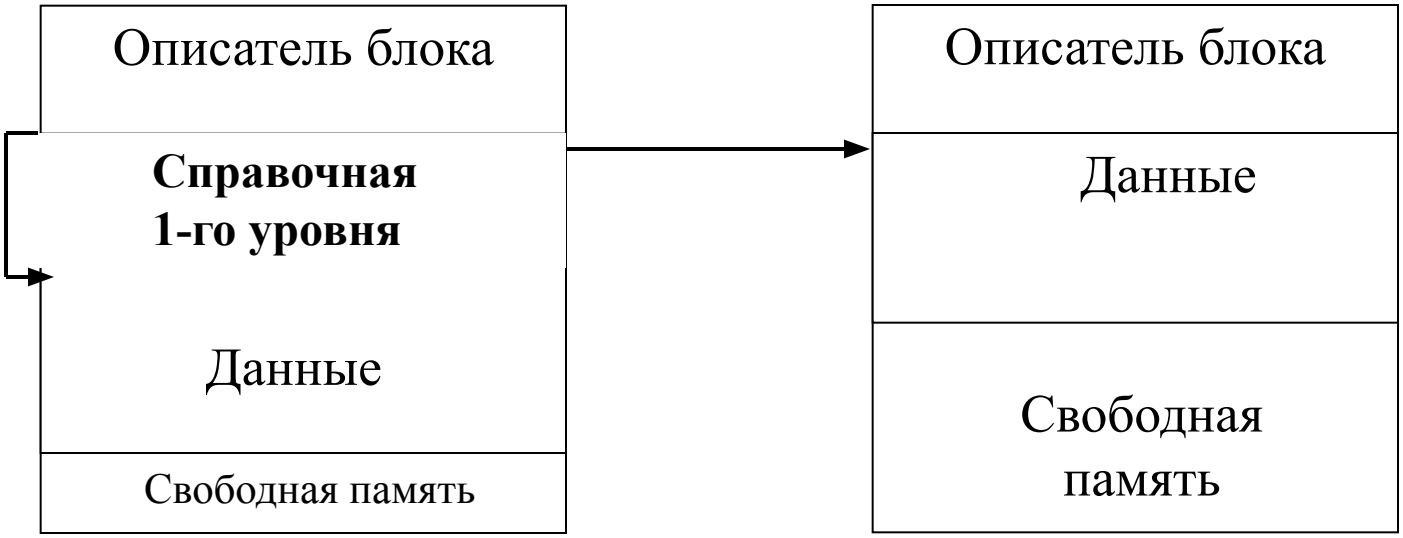
Это обеспечивает балансировку по вертикали.

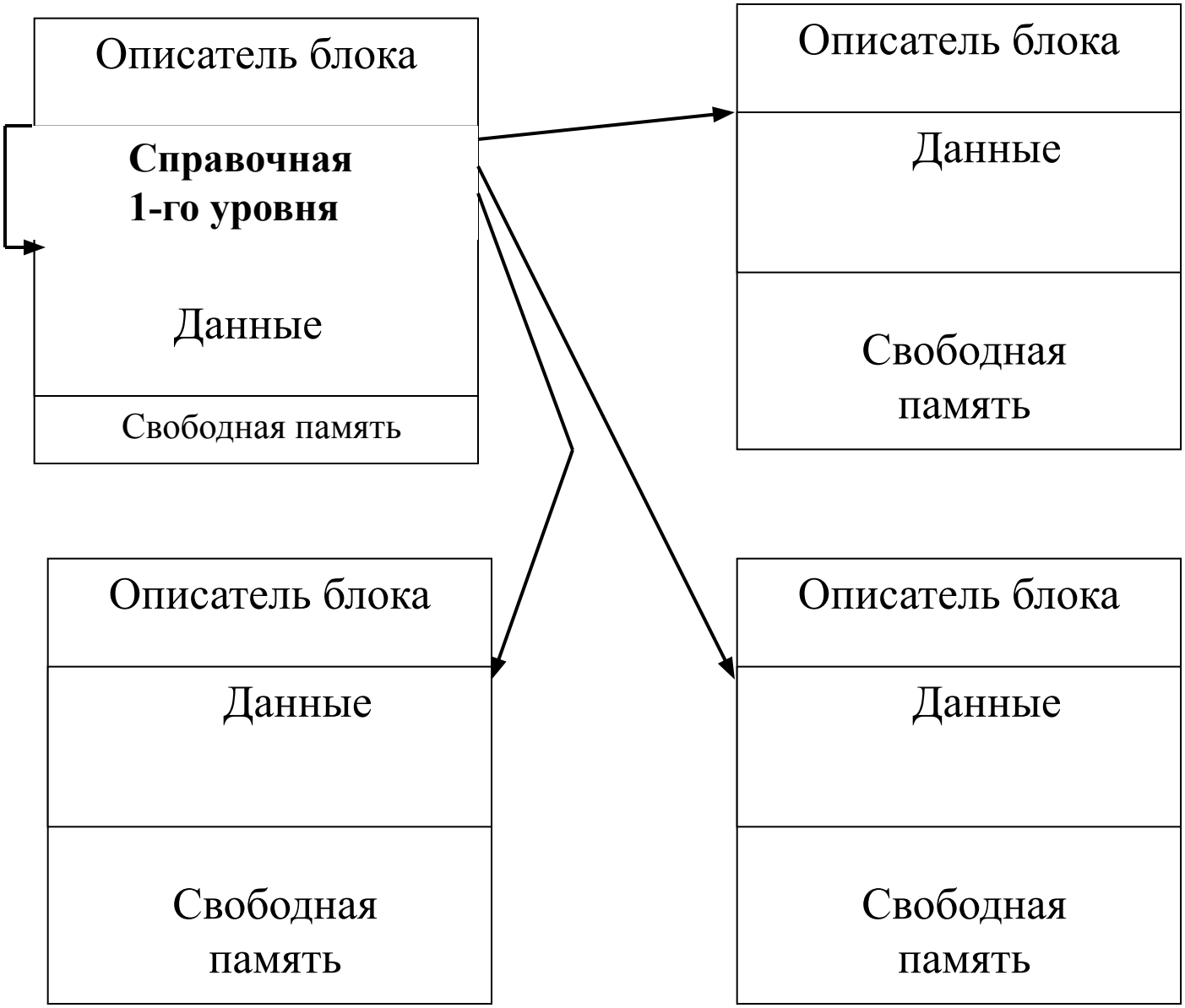
5.5.3. Метод деления страниц (продолжение 3)

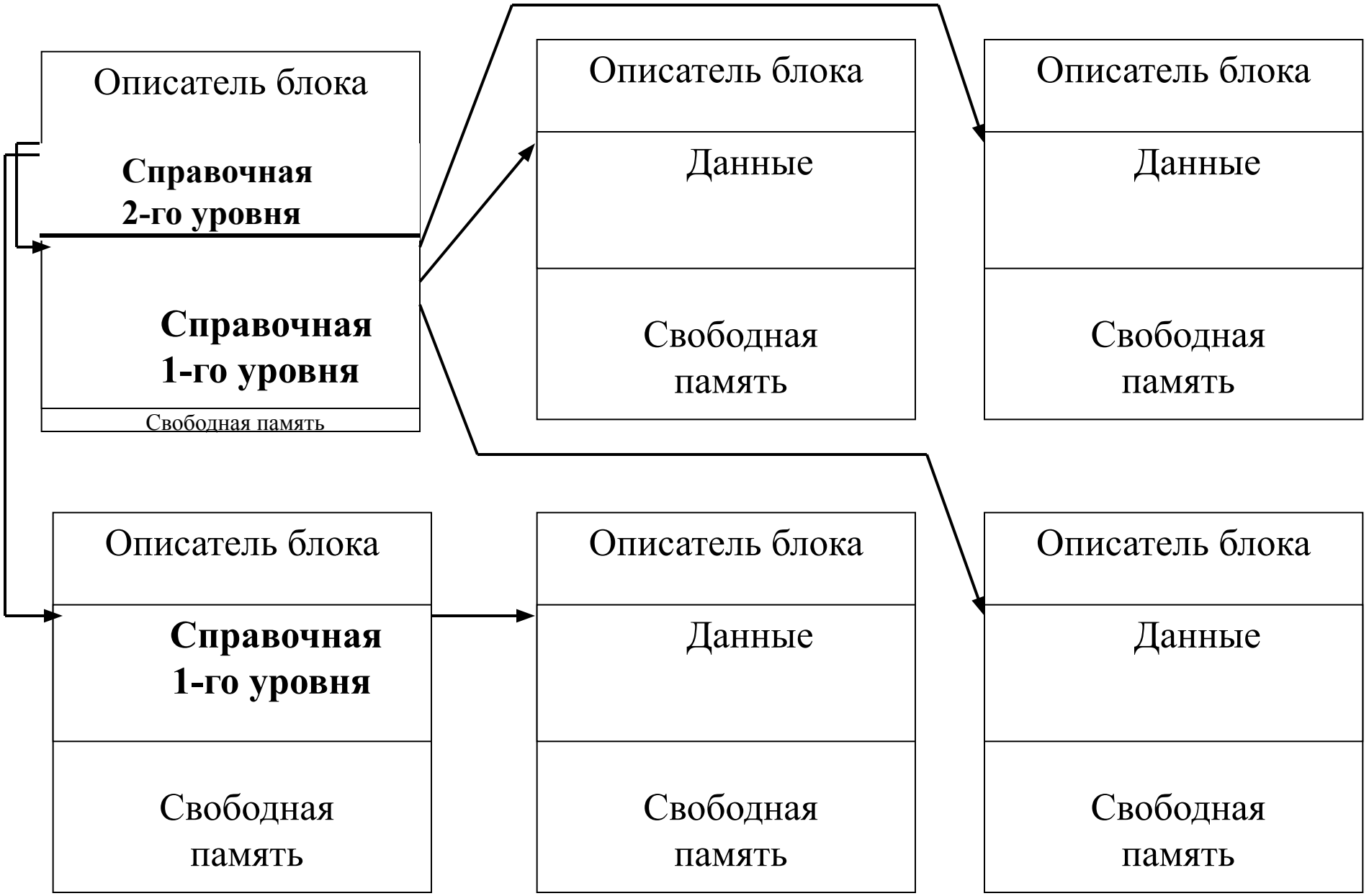
На страницах справочных помещается в среднем примерно одно и тоже число ключей, это приводит к балансировке по горизонтали.

Освободившиеся при удалении данных страницы поступают в резерв свободных, а освободившаяся память внутри страниц - в резерв свободной памяти страницы.


Описатель блока
Данные
Свободная
память


Описатель блока
Справочная
1-го уровня
Данные
Свободная память
Описатель блока
Данные
Свободная
память


Описатель блока
Справочная
1-го уровня
Данные
Свободная память
Описатель блока
Данные
Свободная
память
Описатель блока
Данные
Свободная
память
Описатель блока
Данные
Свободная
память


Описатель блока
Справочная 2-го уровня
Справочная 1-го уровня
Свободная память
Описатель блока
Данные
Свободная память
Описатель блока
Данные
Свободная память
Описатель блока
Справочная 1-го уровня
Свободная память
Описатель блока
Данные
Свободная память
Описатель блока
Данные
Свободная память

5.5.4. Доступ к данным

- **Т.1**. Если известен составной (конкатенированный) ключ, то время доступа минимально.
- **Док.** Доступ проходит за минимальное число обменов, так как идет по многоуровневой логарифмической справочной.

5.5.4. Доступ к данным

- **Т.2.** Если необходим перебор по всей БД или любому поддереву, время доступа минимально.
- **Док.** На каждой странице хранится связное поддерево. После вызова страницы в ОП совершается обход поддерева, страницы вызываются один раз без перевызовов.

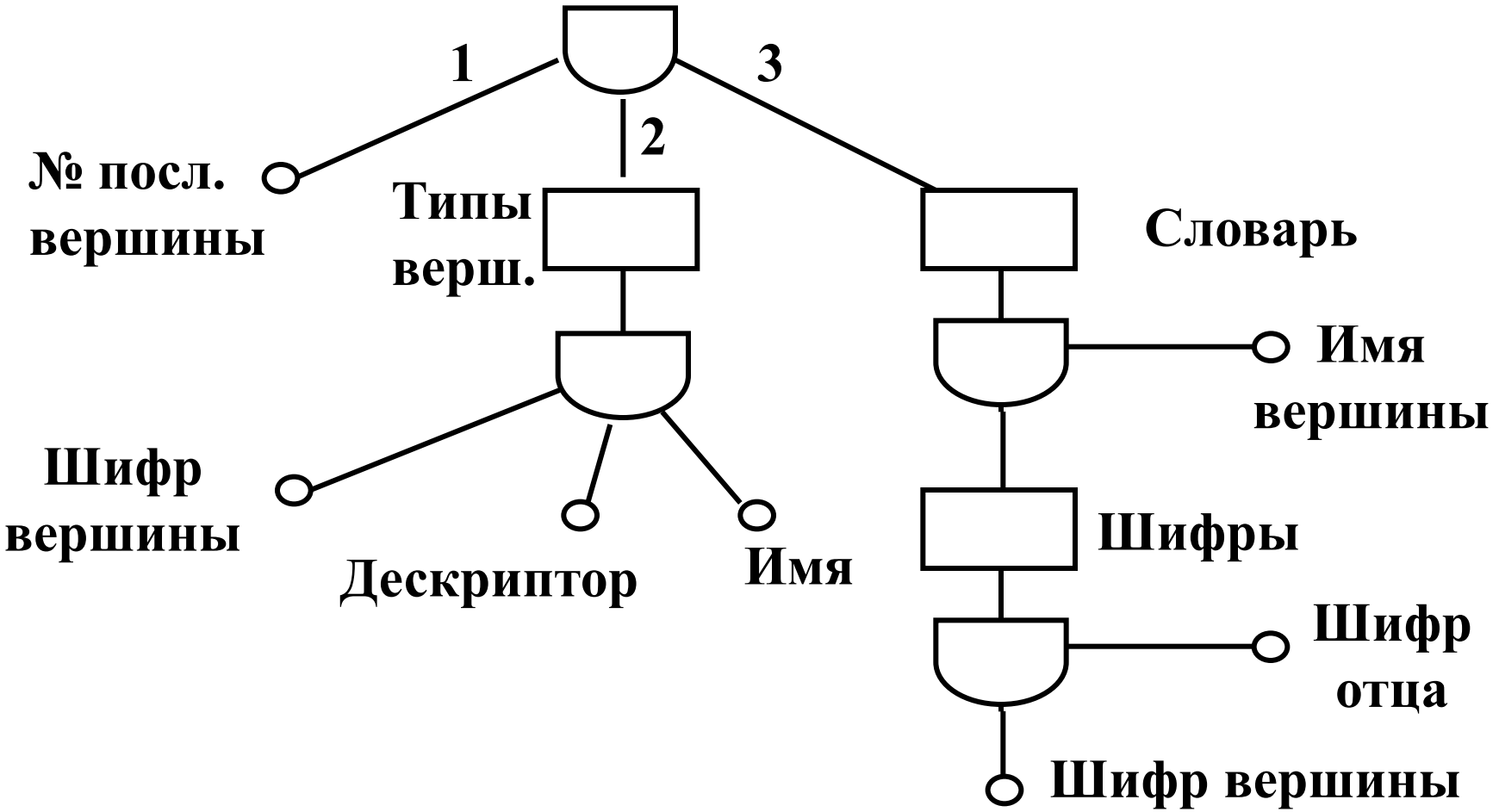
5.5.4. Доступ к данным (объем БД)

- **Т.3.** Объем минимален у слабо заполненных БД и БД со значениями реквизитов разной длины.

- **Док.** Не отводится место под несуществующие реквизиты, реквизит всегда занимает минимальное место +2 байта (шифр и длина). Эффект может достигать нескольких порядков.

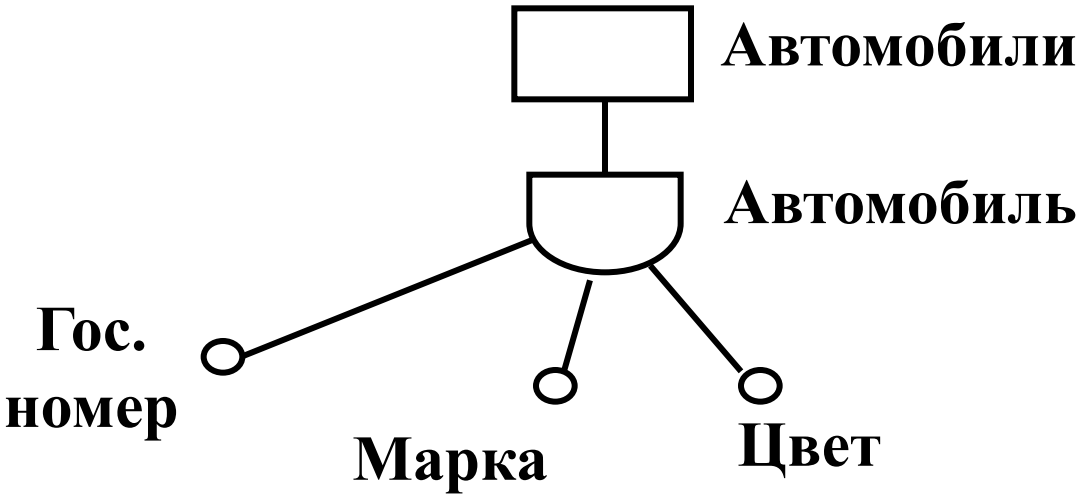
- **Задание.** Привести пример, когда объем реляционной БД меньше?

5.5.5. Структура описания данных

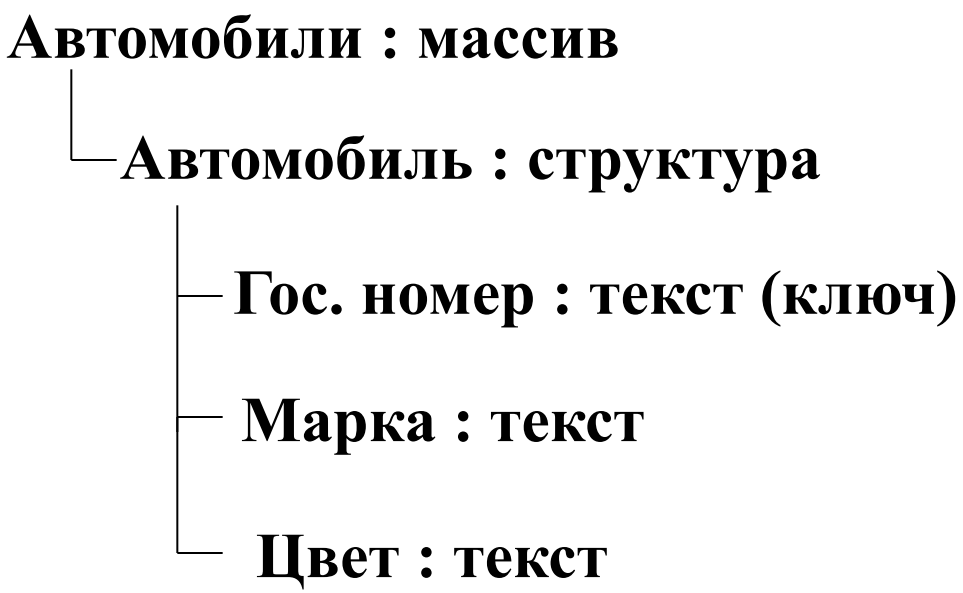


5.5.6. Пример БД

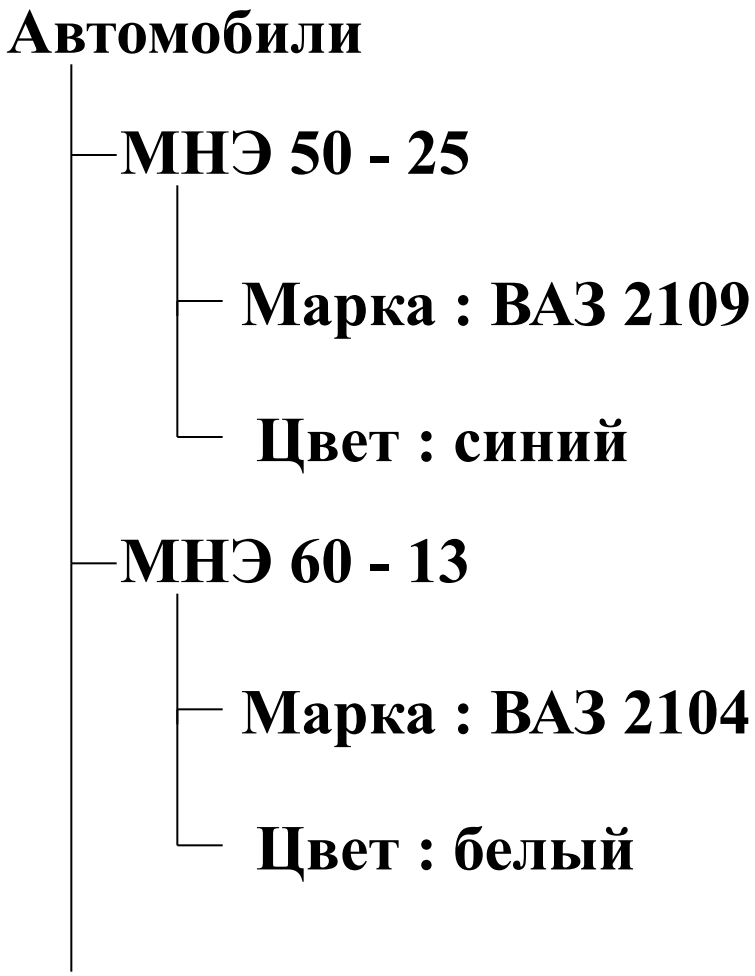
1). Схема БД



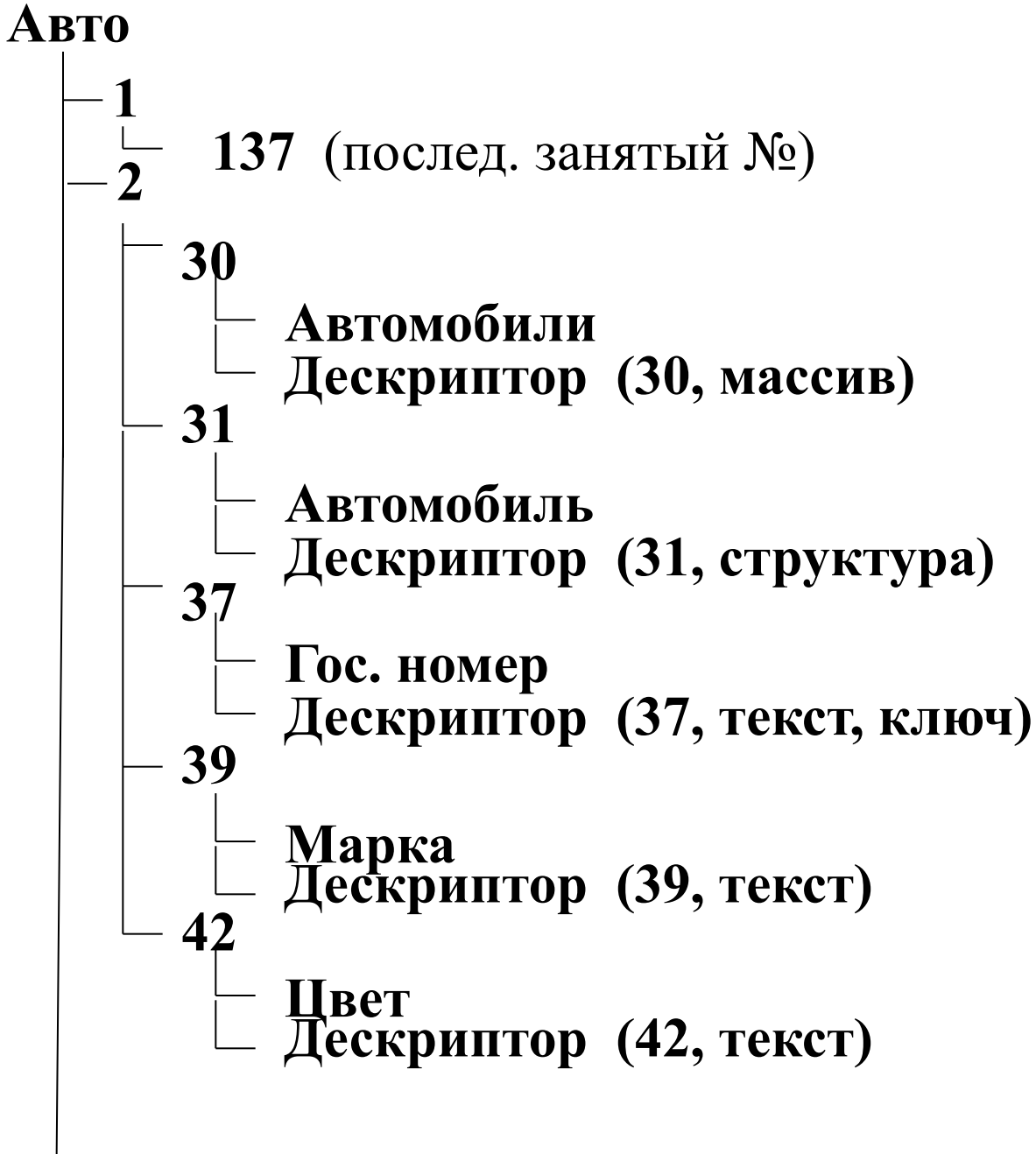
2). Показ Схемы БД на дисплее

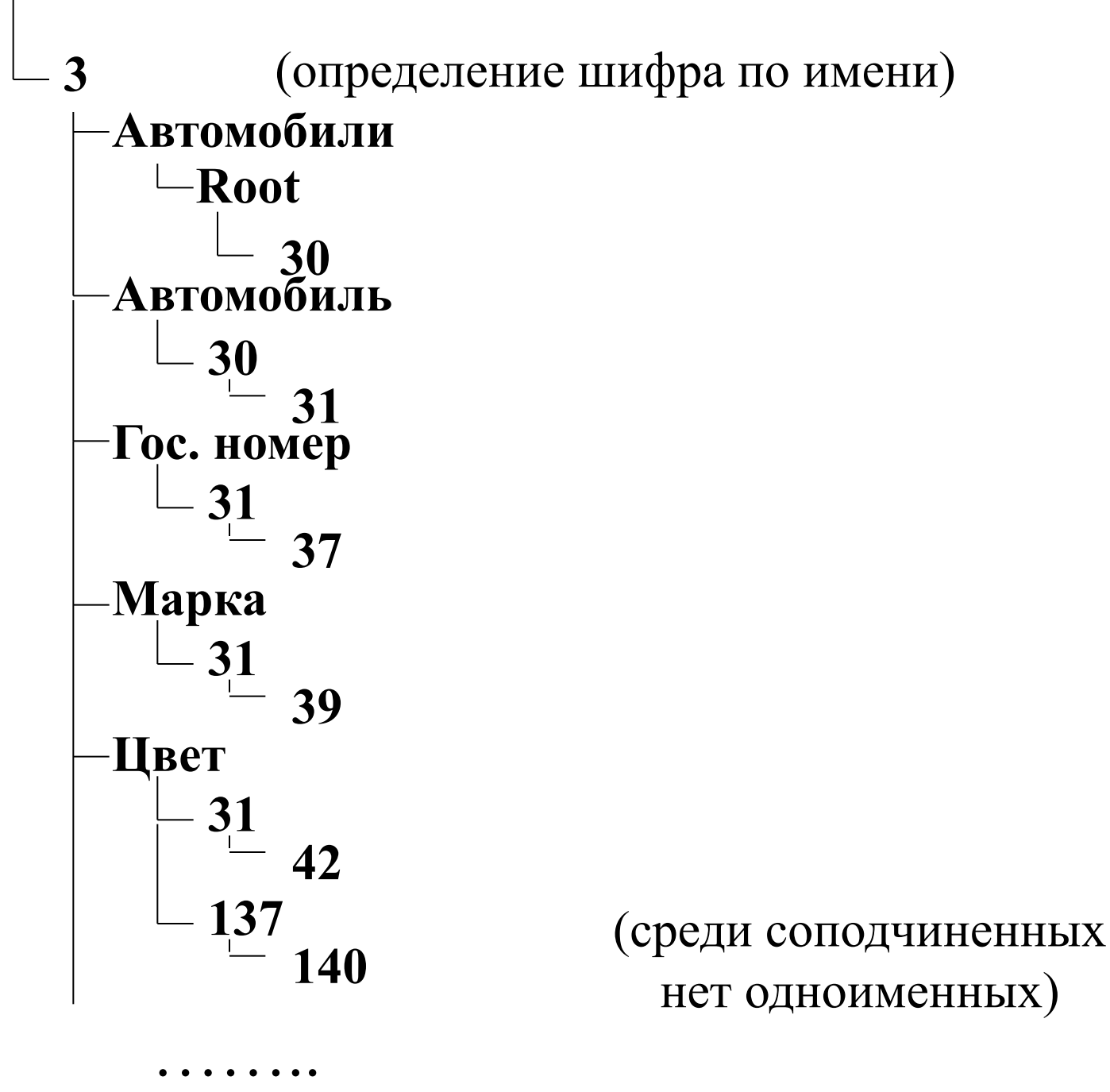


2). Показ самой БД

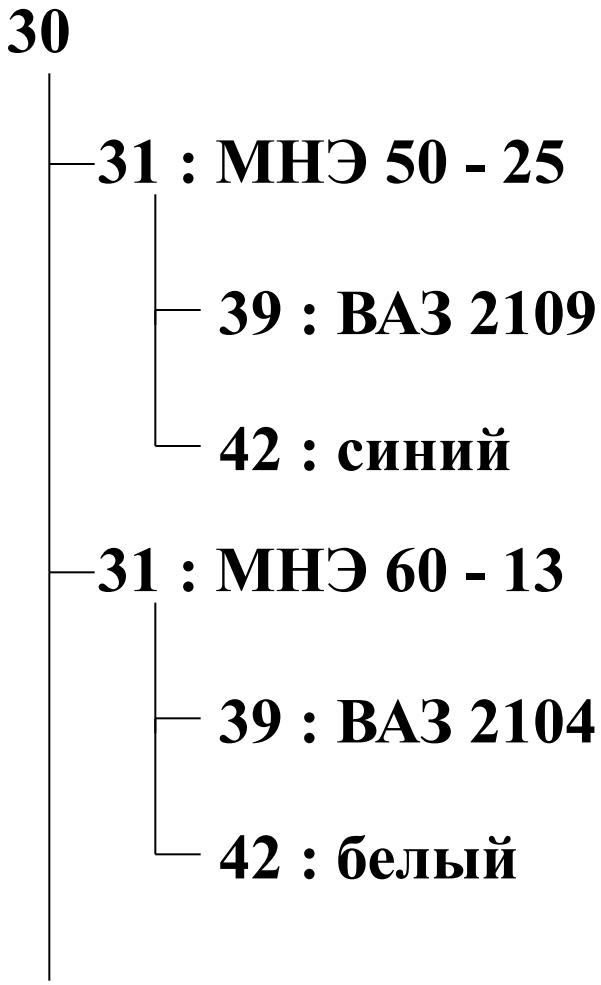


3). Описание данных (Авто.dod)




3
(определение шифра по имени)
Автомобили
Root
30
Автомобиль
30
31
Гос. номер
31
37
Марка
31
39
Цвет
31
42
137
140
(среди соподчиненных
нет одноименных)
. . . . . .

4). Сами данные (Авто.dod)



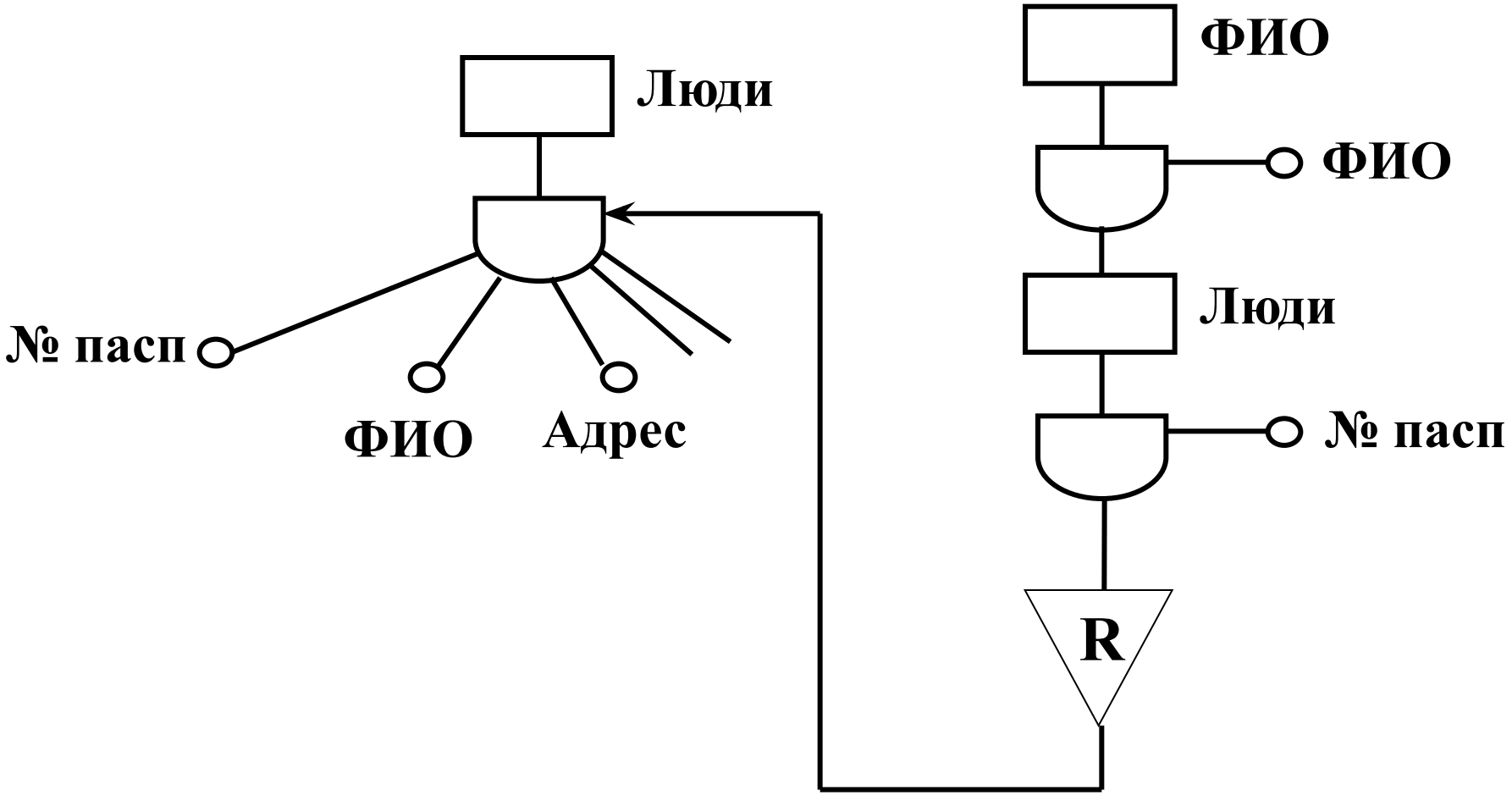
5.5.7. Оптимизация времени доступа

Три подхода к реализации доступа:

- ✔ Интерпретация (dBASE)
- ✔ Трансляция (Clipper)
- ✔ Трансляция при первом обращении (НИКА)

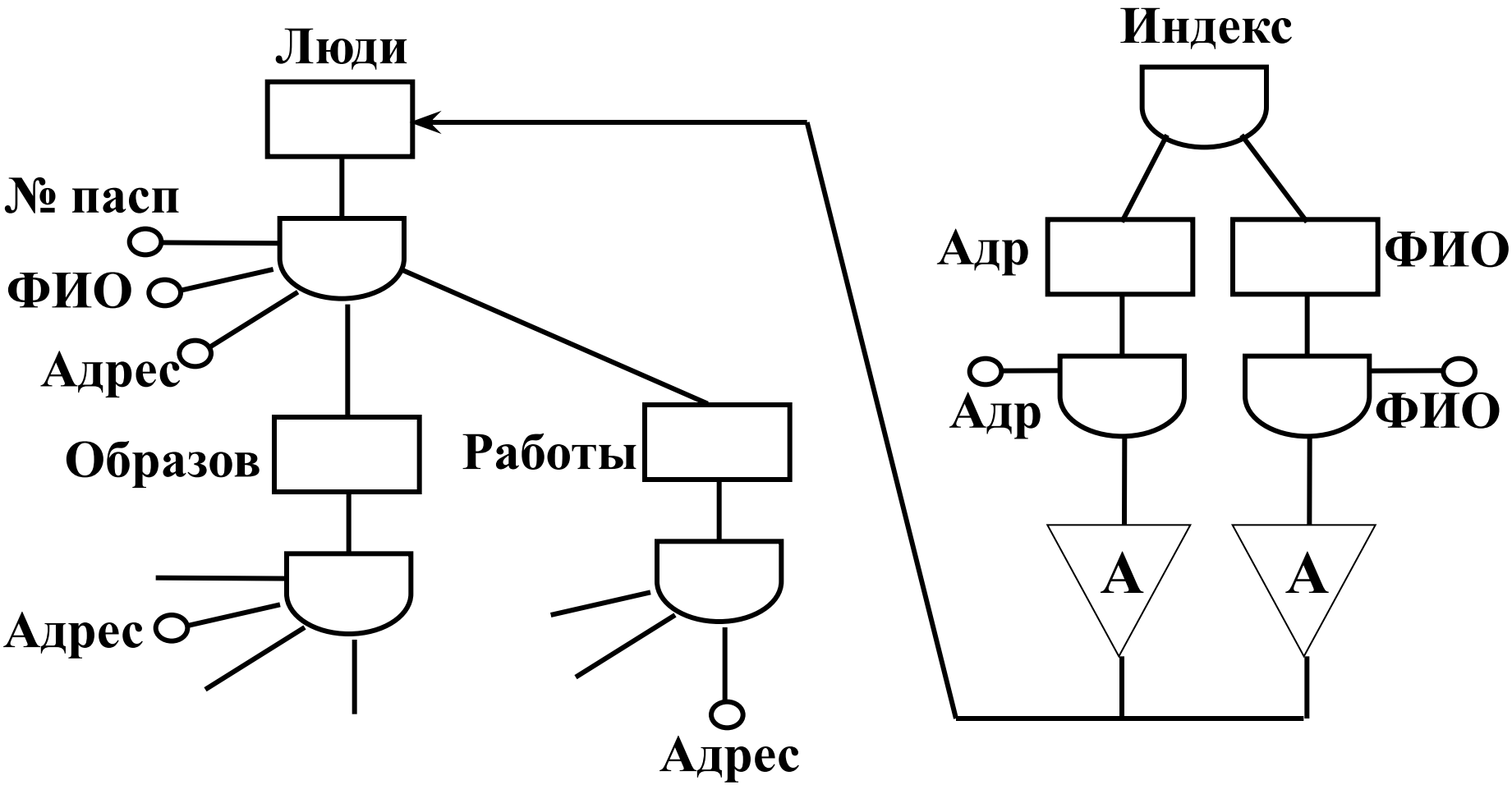
(см. пред. слайд)

5.5.8. Индексация в ООБД и XML DB

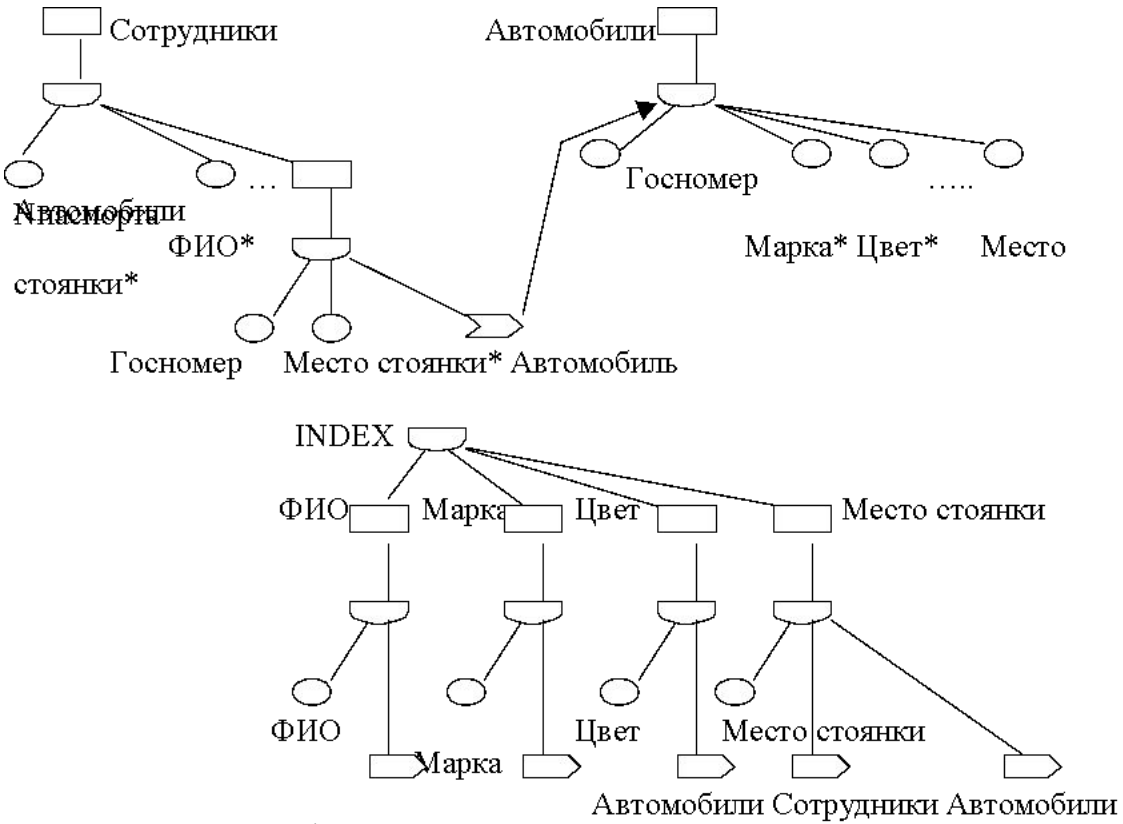


Обычный инверсный вход

Инверсный вход в ООБД



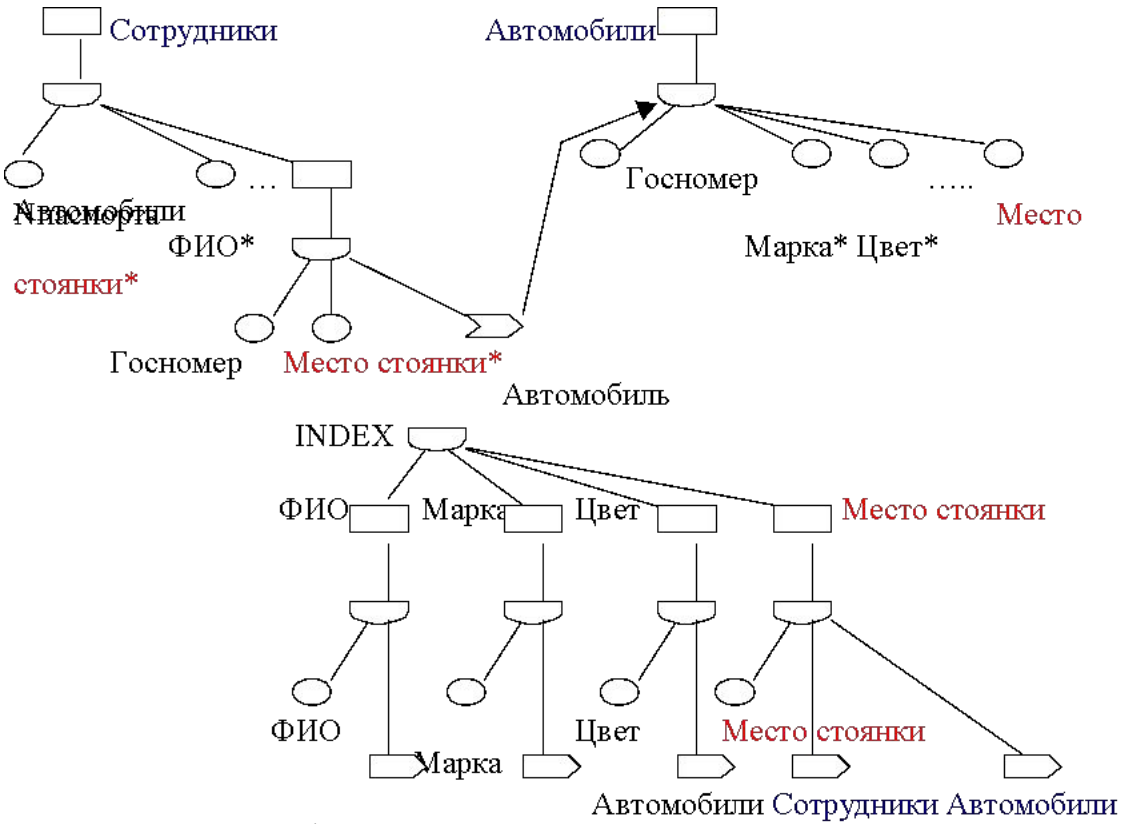
Индекс в СУБД НИКА



Пример индекса СУБД НИКА. Знаком "*" помечены индексируемые данные,

⇨ - ссылка на шаблон, ⇨ - ссылка на значение.

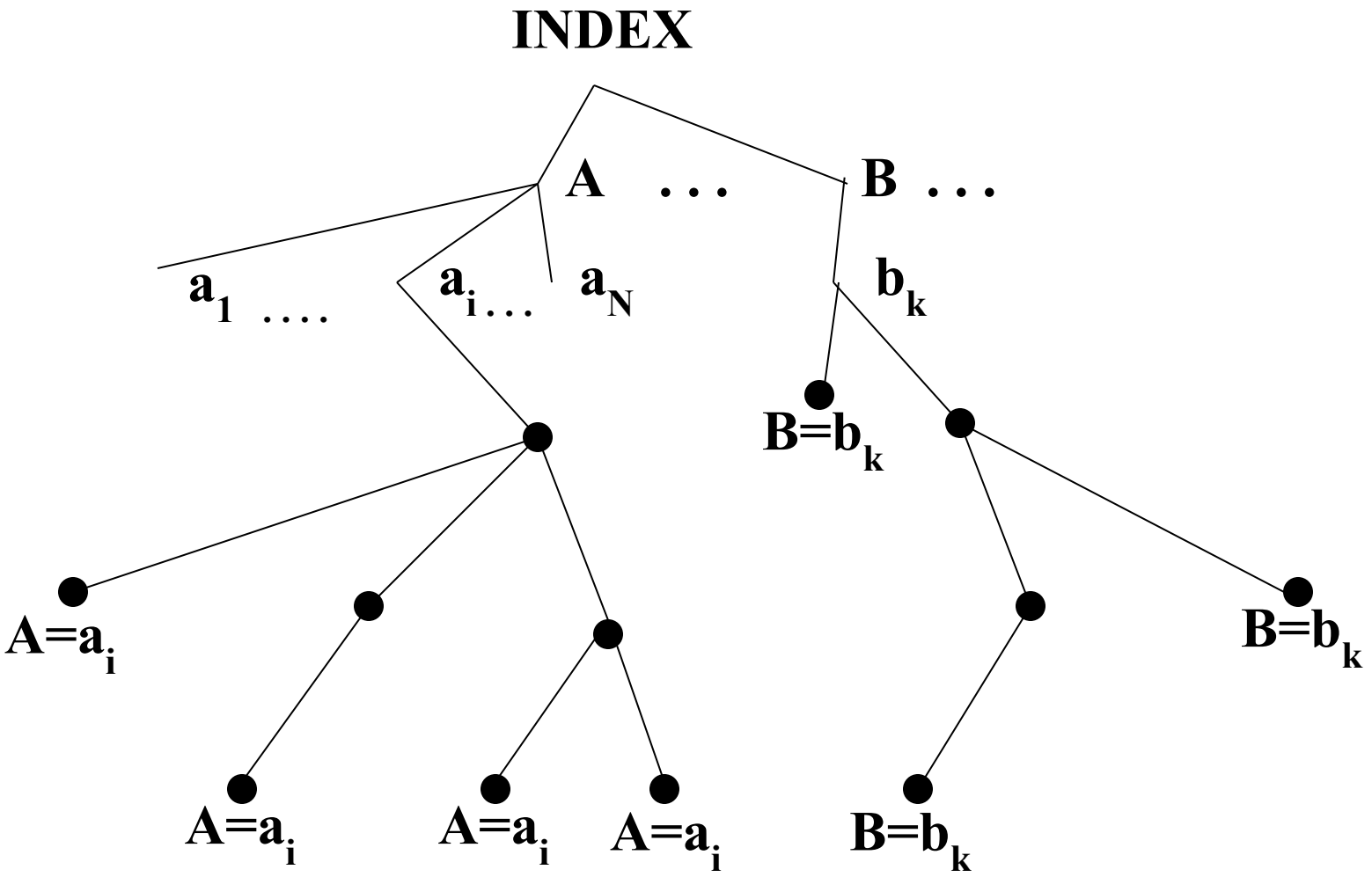
Индекс в СУБД НИКА



Пример индекса СУБД НИКА. Знаком "*" помечены индексируемые данные

- ссылка на шаблон, - ссылка на значение.

Индекс в СУБД НИКА

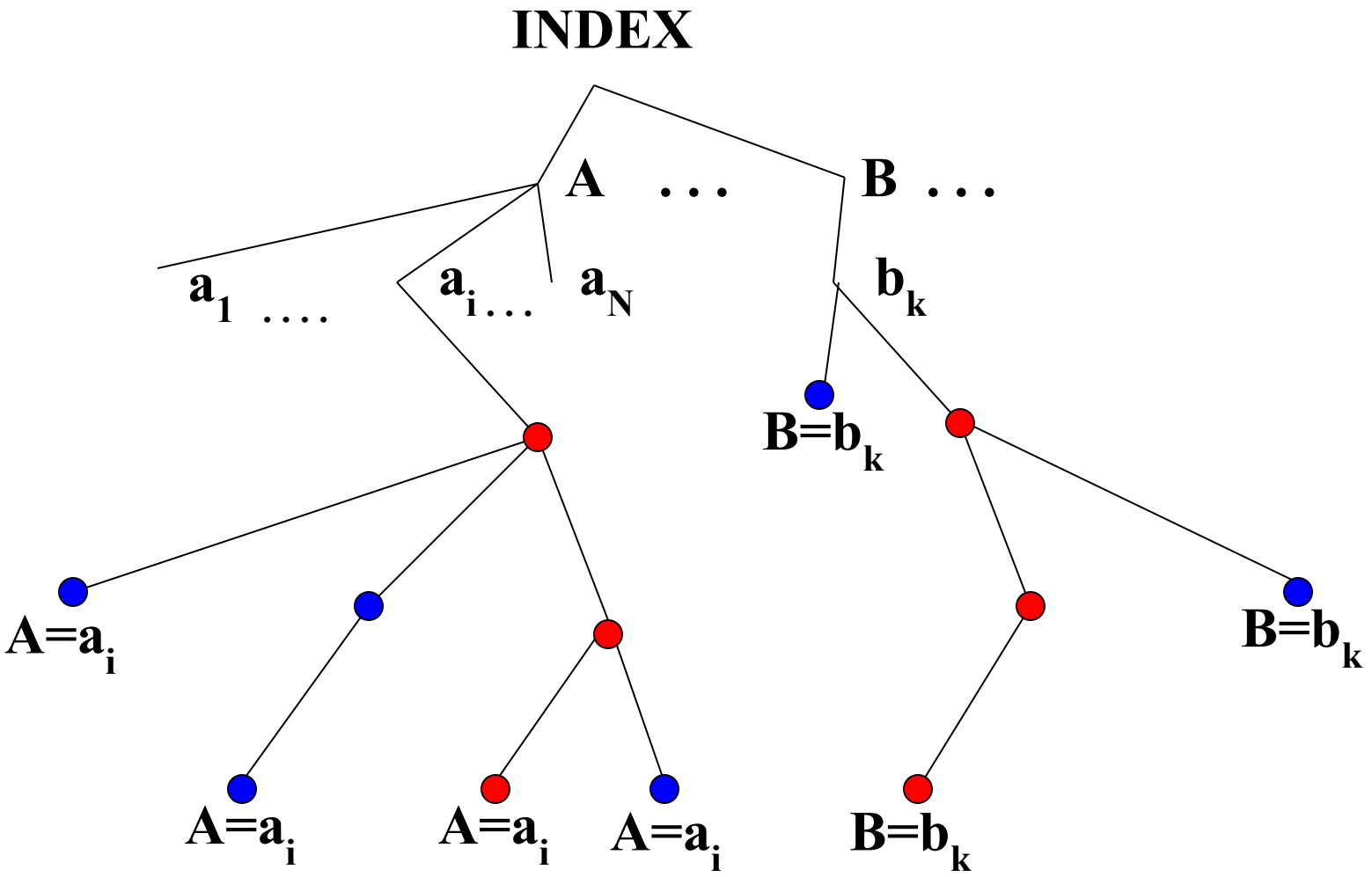


Пример двух веток индекса:

$A = a_i$

$B = b_k$

Индекс в СУБД НИКА



Пример выполнения логических операций:

«И» - ● объекты ,

«ИЛИ» - ● и ● объекты.

- **Теорема.** Такая организация индекса необходима и достаточна для поиска без перебора любых объектов с любыми условиями на реквизиты.