

Модели данных

Понятие «данные» в концепции баз данных — это набор конкретных значений, параметров, характеризующих объект, условие, ситуацию или любые другие факторы.

Примеры данных: Петров Николай Степанович, \$30 и т. д. Данные не обладают определенной структурой, данные становятся информацией тогда, когда пользователь задает им определенную структуру

Модель данных — это некоторая абстракция, которая, будучи приложима к конкретным данным, позволяет пользователям и разработчикам трактовать их уже как информацию, т. е. сведения, содержащие не только данные, но и взаимосвязь между ними

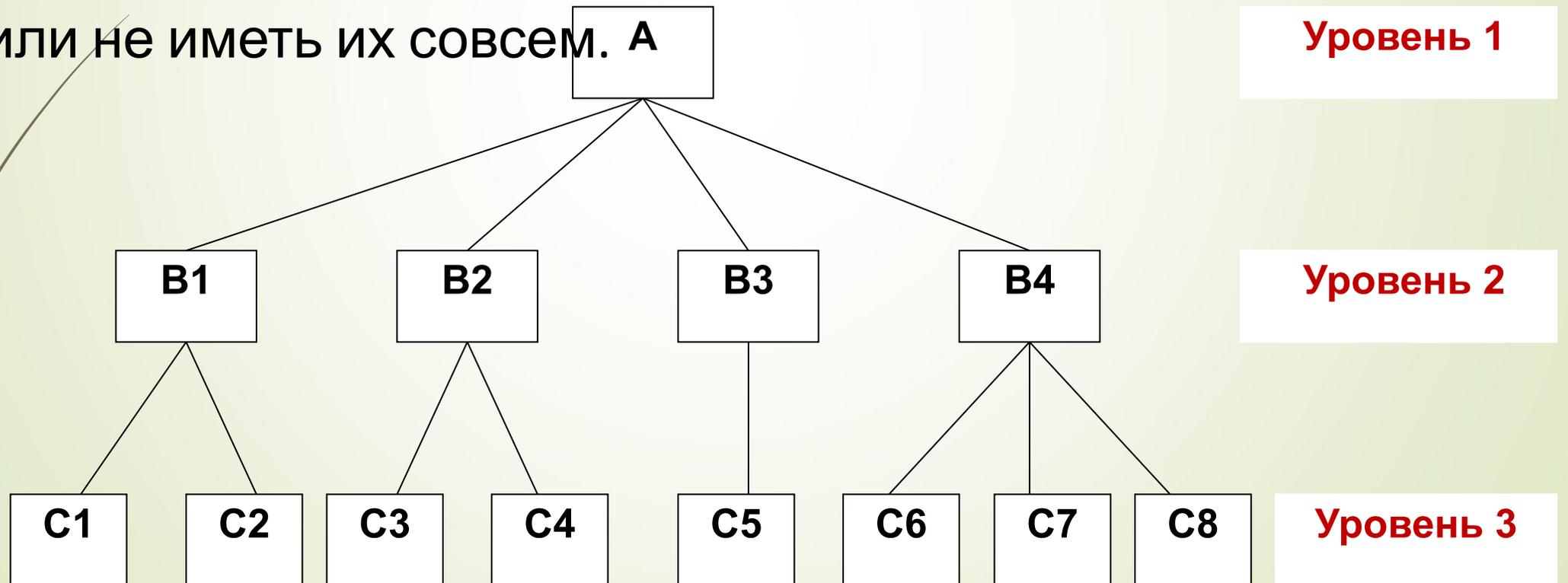
Модель данных – совокупность структур данных и операций их обработки (логическая структура данных в БД).

1. иерархическая модель
2. сетевая
3. реляционная
4. постреляционная
5. многомерная
6. объектно-ориентированная

1.1 Иерархическая модель

3

Под иерархической моделью данных понимается модель, объединяющая записи, хранимые в общей древовидной структуре с одним корневым типом записи, который имеет несколько подчиненных типов записи или не имеет совсем. Каждый подчиненный тип записи также может иметь несколько подчиненных типов или не иметь их совсем. А



1.1 Иерархическая модель

4

Основной структурой, поддерживающей иерархическое представление информации, является дерево.

Основные понятия:

- ✓ уровень
- ✓ узел (элемент)
- ✓ связь

Узел — представляет объект предметной области. Объекты могут иметь как одинаковый тип, так и являться разными сущностями.

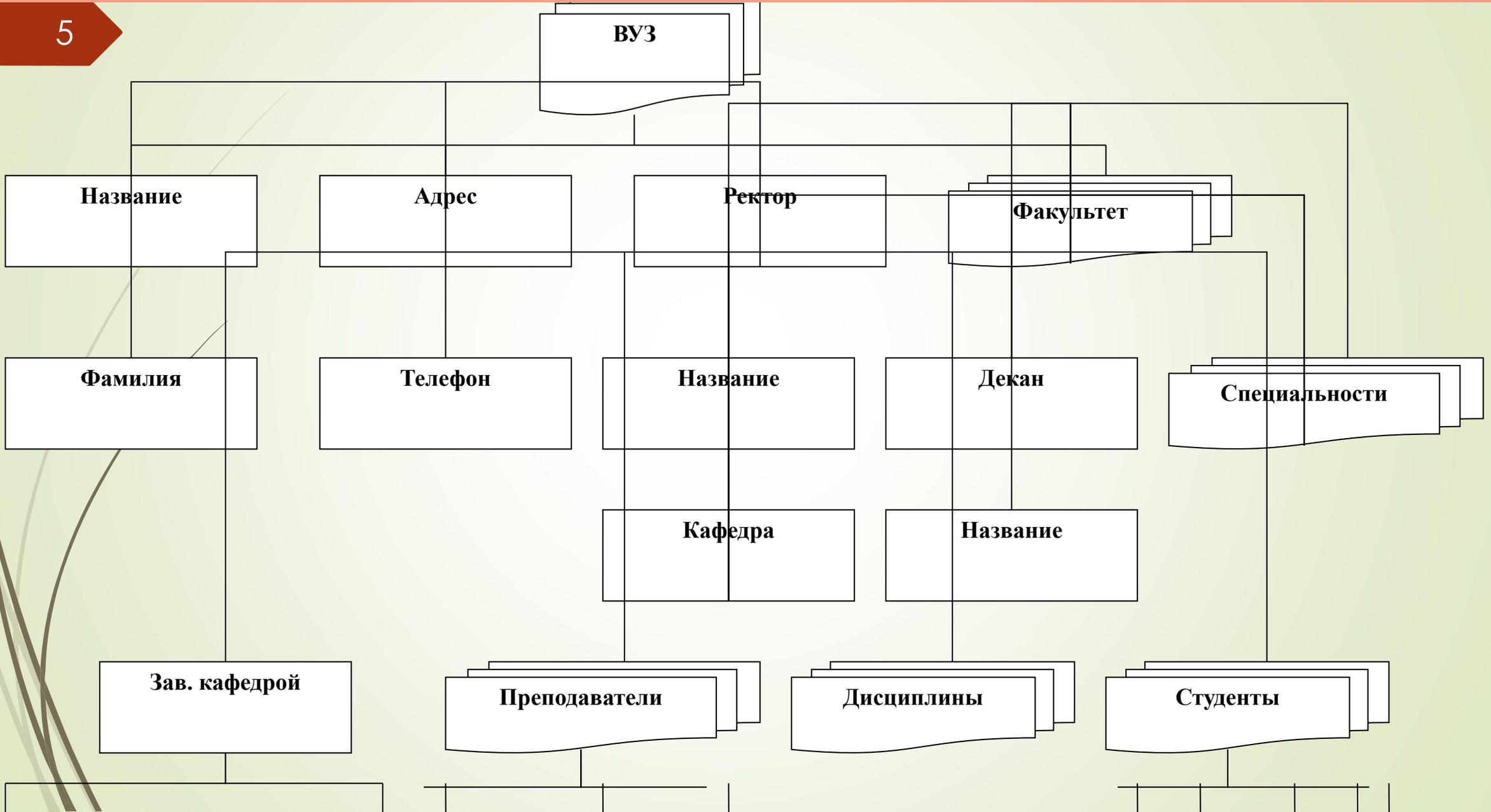
Иерархическая структура естественным образом поддерживает связи типа «один ко многим» (1: M) и типа «один к одному» (1:1).

Основные правила контроля целостности:

- ✓ Узел-потомок не может существовать без узла-родителя, а у некоторых родителей может не быть узлов-потомков.
- ✓ К каждой записи (узлу БД) существует только один путь от корневой записи.

1.1 Иерархическая модель

5



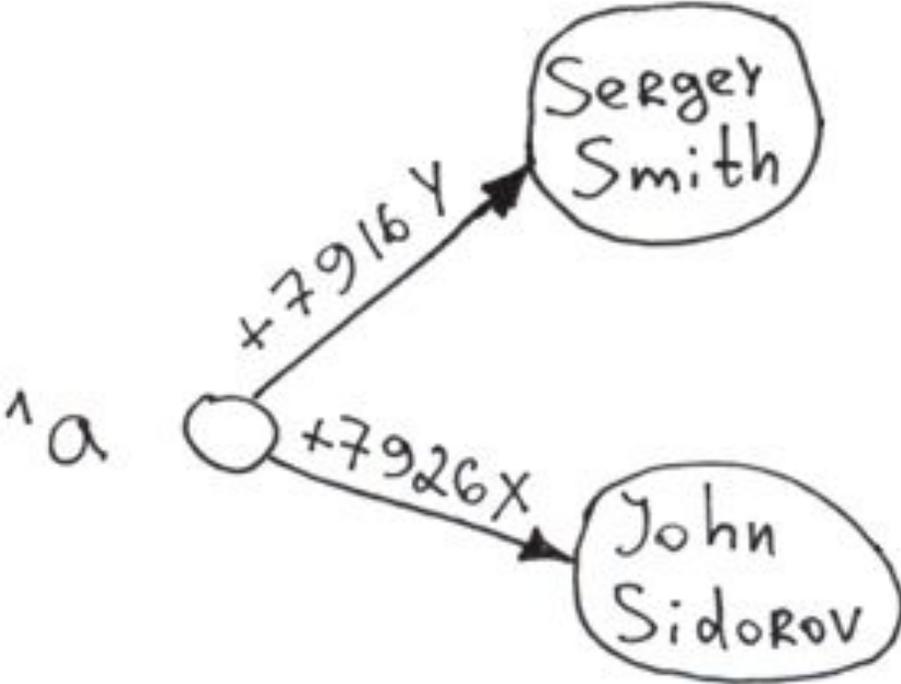
1.1 Иерархическая модель

6

Основные операции манипулирования иерархически организованными данными:

1. Поиск указанного экземпляра БД
2. Переход от одного дерева к другому
3. Переход от одной записи к другой внутри дерева
4. Вставка новой записи в указанную позицию
5. Удаление текущей записи и т. д.

1.1 Иерархическая модель



Самая сильная сторона иерархических структур — это скорость вставки новых узлов. Данные там всегда проиндексированы. Их обход как на одном уровне, так и вглубь дерева, всегда быстр.

```
Set ^a("+7926X") = "John Sidorov"
```

```
Set ^a("+7916Y") = "Sergey Smith"
```

Варианты структур при использовании глобалов

1. ОДИН УЗЕЛ БЕЗ ВЕТВЕЙ

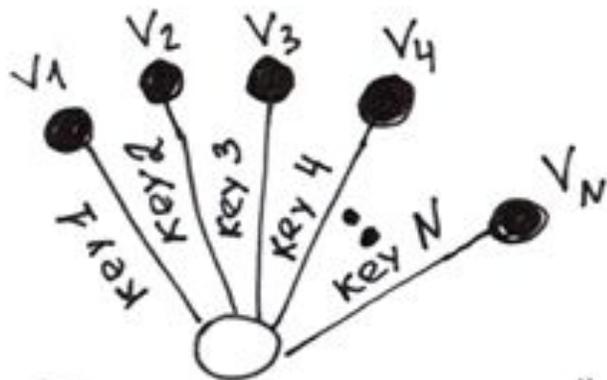


1.1 Иерархическая модель

8

2. ОДНА ВЕРШИНА И МНОЖЕСТВО ВЕТВЕЙ

Вообще — это классическая key-value база. А если в качестве значения мы будем сохранять кортеж значений, то получим самую обыкновенную таблицу с первичным ключом.



База "key-value"
на глобале

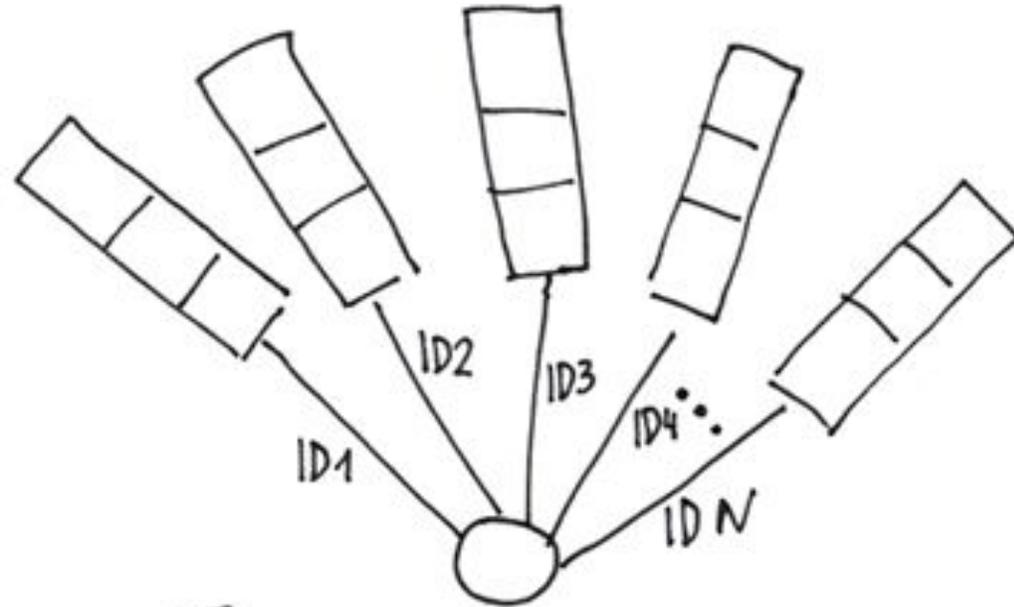
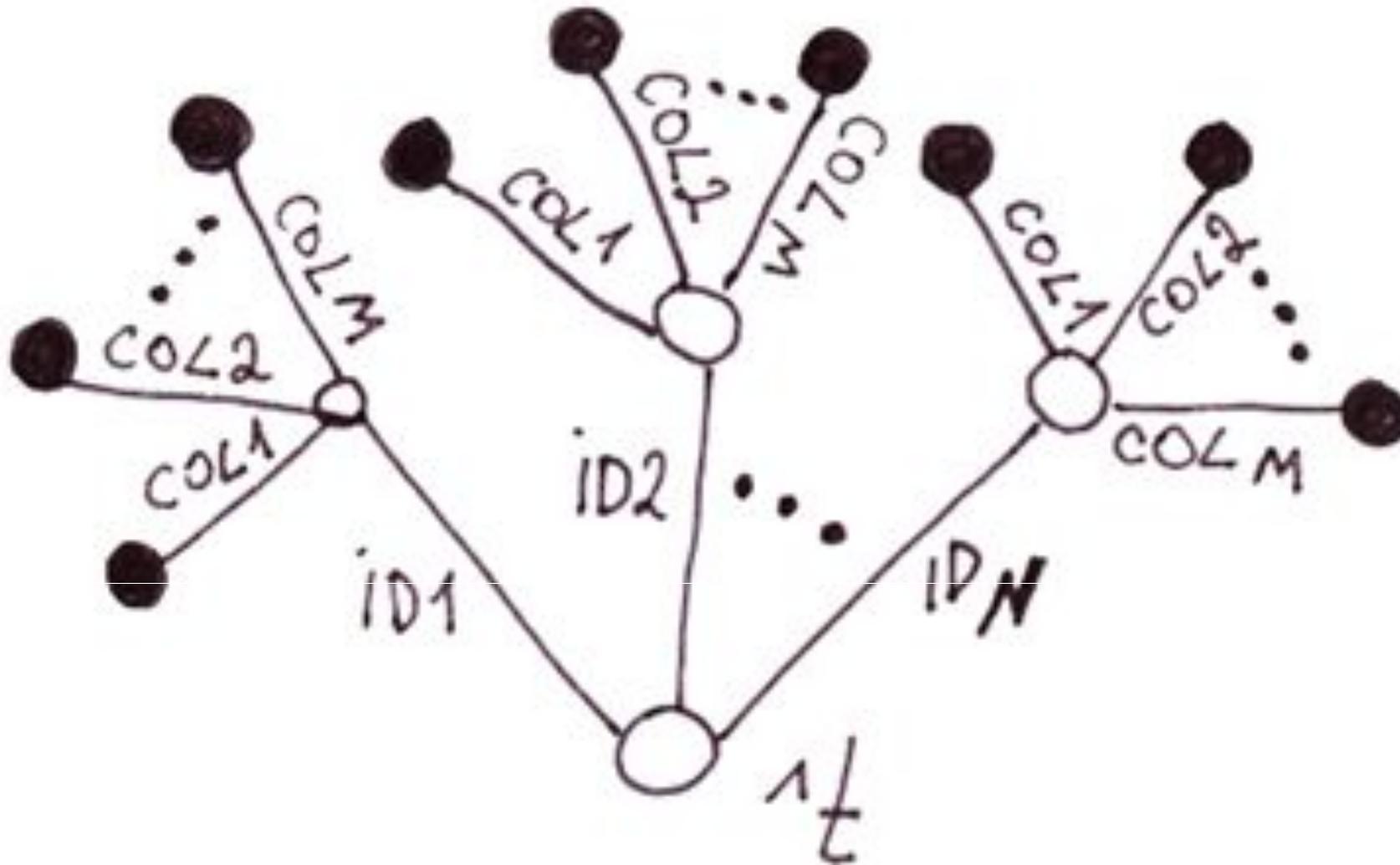


Таблица на глобале.
В качестве значения
сохраняется строка

1.1 Иерархическая модель

9

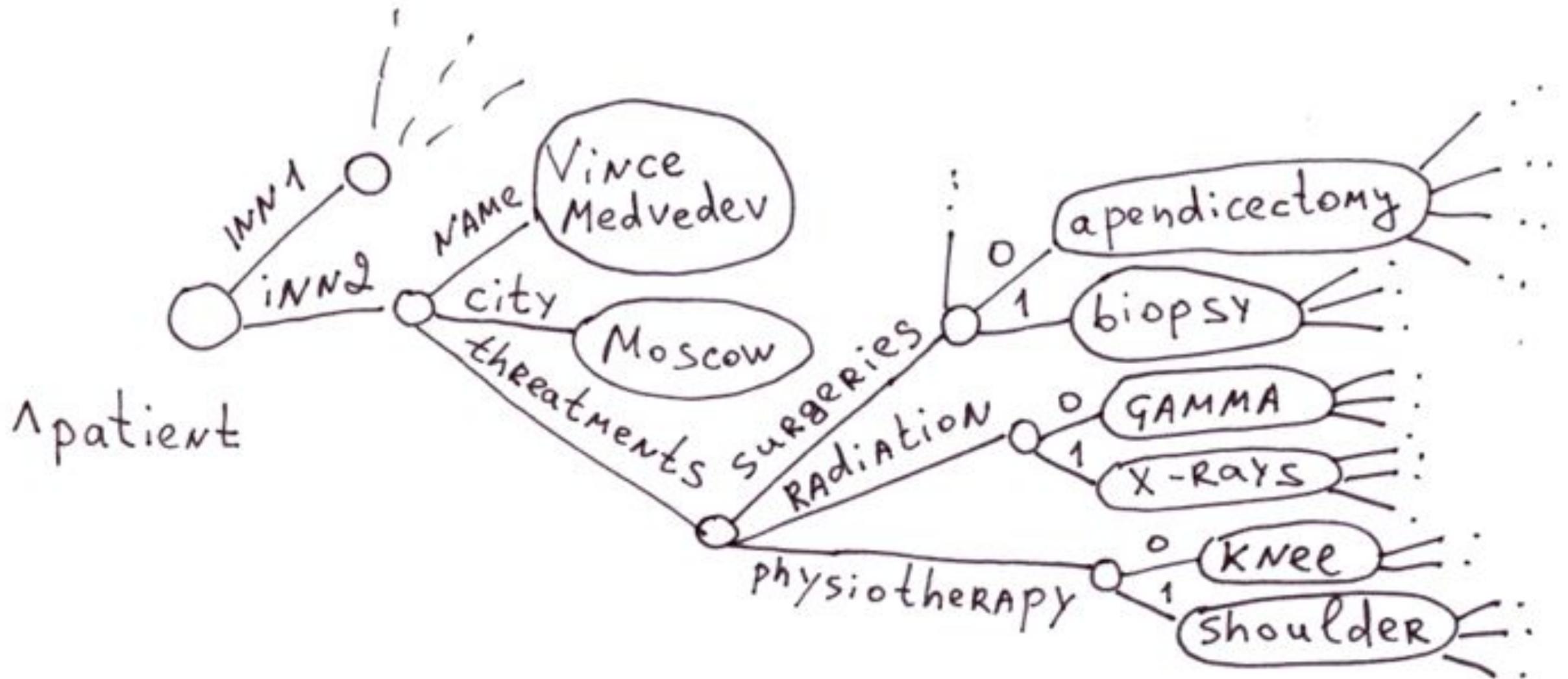
3 ДВУХУРОВНЕВОЕ ДЕРЕВО, У КАЖДОГО УЗЛА ВТОРОГО УРОВНЯ ФИКСИРОВАННОЕ ЧИСЛО ВЕТВЕЙ



1.1 Иерархическая модель

10

4 Объекты с подобъектами



1.1 Иерархическая модель

11

5 Иерархические документы: XML, JSON

Самый простой способ раскладки XML на глобалы, это когда в узлах храним атрибуты тэгов. А если будет нужен быстрый доступ к атрибутам тэгов, то мы можем их вынести в отдельные ветви.

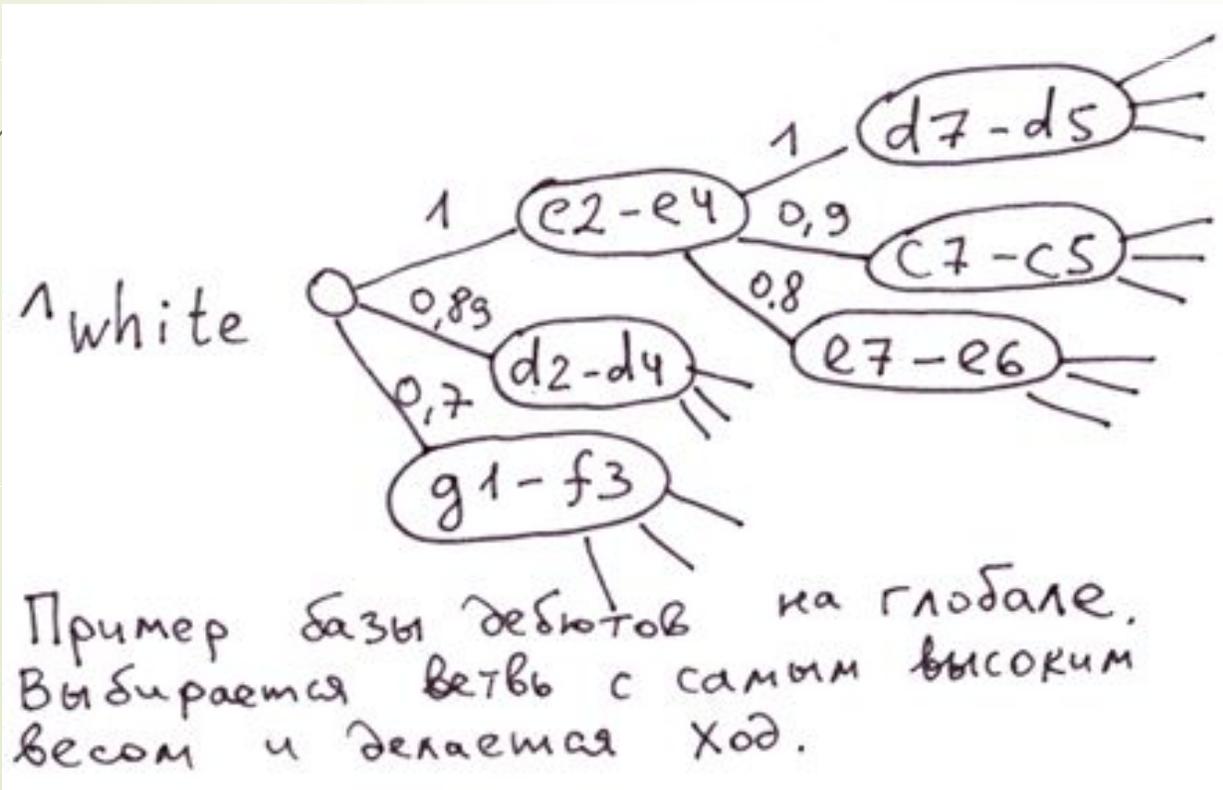


1.1 Иерархическая модель

12

6 Одинаковые структуры, связанные иерархическими отношениями

База дебютов. Можно в качестве значения индекса узла глобала использовать оценку силы хода. Тогда, чтобы выбрать самый сильный ход достаточно будет выбрать ветвь с наибольшим весом. В глобале все ветви на каждом уровне будут отсортированы по силе хода.



1.1 Иерархическая модель

13

6 Одинаковые структуры, связанные иерархическими отношениями

Структура офисов продаж, структура людей в МЛМ. В узлах можно хранить некие кеширующие значения отражающие характеристики всего поддерева. Например, объём продаж данного поддерева. В любой момент мы можем получить цифру, отражающую достижения любой ветви.



В каждом узле можно хранить интегральные показатели

1.1 Иерархическая модель

СКОРОСТЬ

Вставка [с автоматической сортировкой на каждом уровне], [индексированием по главному ключу]

Удаление поддеревьев

Объекты с массой вложенных свойств, к которым нужен индивидуальный доступ

Иерархическая структура с возможностью обхода дочерних ветвей с любой, даже не существующей

Обход поддеревьев в глубину

УДОБСТВО ДААННЫХ

ОБРАБОТКИ/ПРЕДСТАВЛЕНИЯ

Объекты/сущности с огромным количеством необязательных [и/или вложенных] свойств/сущностей

Безсхемные данные (schema-less). Когда часто могут появляться новые свойства и исчезать старые.

Нужно создать нестандартную БД. Базы путей и деревья решений. Когда пути удобно представлять в виде дерева.

Удаление иерархических структур без использования рекурсии

1.1 Иерархическая модель

15

При вставке информации (команда Set) автоматически происходят 3 вещи:

1 Сохранение данных на диск.

2 Индексация. То что в скобках выступает ключом (в англоязычной литературе — «subscript»), а справа от равно — значением («node value»).

3 Сортировка. Данные сортируются по ключу. В дальнейшем при обходе массива первым элементом станет «Sergey Smith», а вторым «John Sidorov».

1.1 Иерархическая модель

16

При работе с обобщенной древовидной структурой используются два метода доступа ко всем узлам (типам записей) внутри дерева.

Один метод начинается с доступа к корню с последующей обработкой всего дерева с доступом к поддеревьям в порядке слева направо. Это так называемый **прямой порядок обхода** дерева (pre-order traversal) или нисходящий порядок обхода.

Другой метод начинается с доступа к самым нижним узлам с постепенным нисходящим переходом от одного поддерева к другому слева направо и с завершением обработки в корне. Этот метод называют **обратным порядком обхода** дерева (post-order traversal) или **восходящим порядком обхода**. В информационных структурах чаще всего используется нисходящий порядок обхода, поскольку самые важные данные, как правило, располагаются на самых высоких уровнях древовидной

1.1 Иерархическая модель

17

Достоинства

- ✓ эффективное использование памяти ЭВМ;
- ✓ хорошие показатели времени выполнения основных операций над данными.

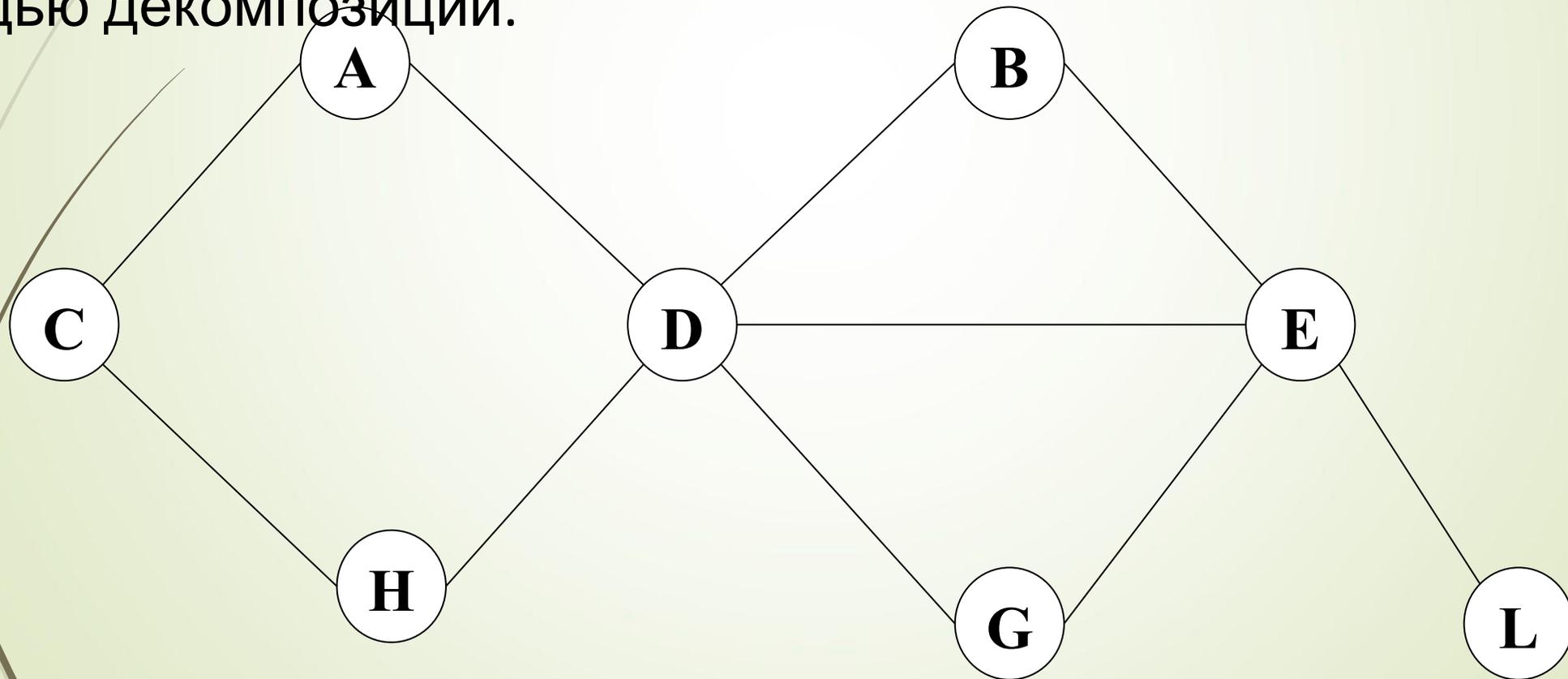
Недостатки

- ✓ отсутствуют механизмы поддержания целостности связи между записями различных деревьев;
- ✓ невозможность хранения экземпляров записей, которые не имеют никаких родительских записей.
- ✓ громоздкость иерархической модели для обработки информации с достаточно сложными логическими связями;
- ✓ сложность понимания для обычного пользователя;
- ✓ трудность моделирования связей типа «многие ко многим» (M:N)

1.2 Сетевая модель

18

Под **сетевой моделью данных** понимается модель, состоящая из записей, элементов данных и связей типа «один ко многим» (1:M), установленных между записями. При этом связи типа «многие ко многим» (M:N) и рекурсивные связи поддерживаются с помощью декомпозиции.



1.2 Сетевая модель

19

Позволяет отображать разнообразные связи элементов данных в виде произвольного графа, тем самым обобщая иерархическую модель.

Структура называется **сетевой**, если в отношениях между данными структурный элемент может иметь более одного исходного.

При тех же основных понятиях (узел, элемент, связь) в сетевой структуре каждый элемент может быть связан с каждым.

Тип связь определяется для двух типов записей – предка и потомка.

Переменные типа связь являются экземплярами связи.

Сетевая БД состоит из набора записей и набора соответствующих данных.

На формирование связей ограничения не накладываются.

1.2 Сетевая модель

20

В качестве базовой физической структуры данных выступает сеть, в которой записи связаны друг с другом в один набор с помощью указателей. Записи могут содержать встроенные в них указатели.

Под логической структурой данных понимается набор, в котором один тип записи-родителя может быть связан со многими типами записей-потомков. Сложные сети создаются с помощью типов наборов.

Поддержка целостности на уровне ссылок для связей типа «родитель-потомок» обеспечивается средствами СУБД с использованием правил вставки и сохранения для структуры наборов.

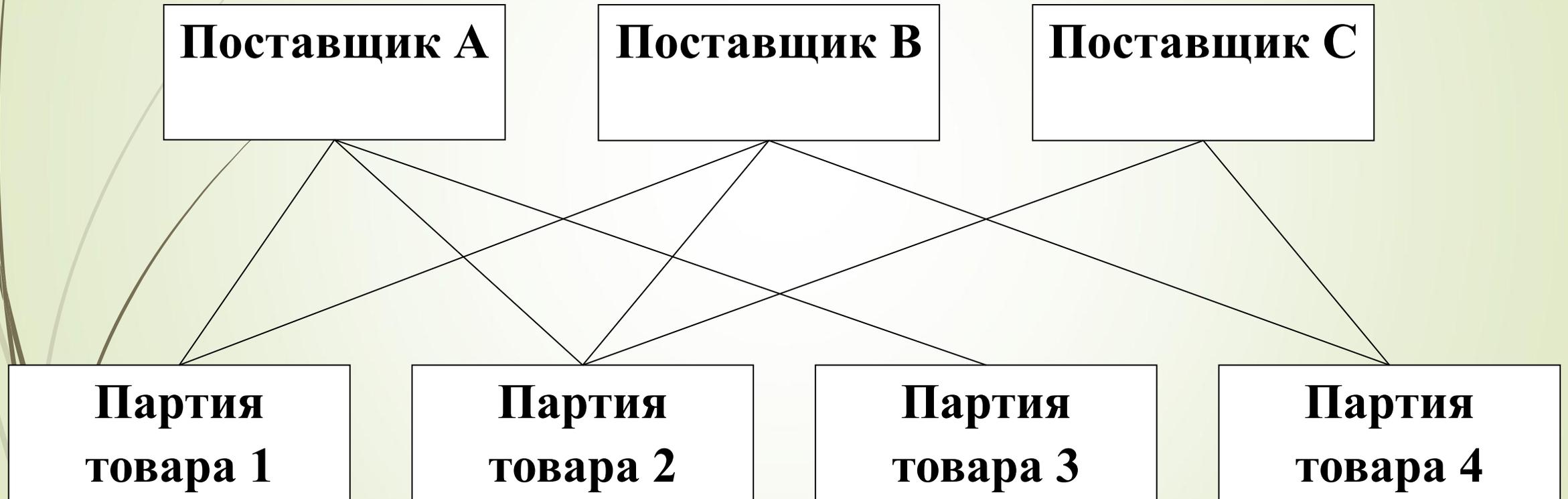
1.2 Сетевая модель

21



1.2 Сетевая модель

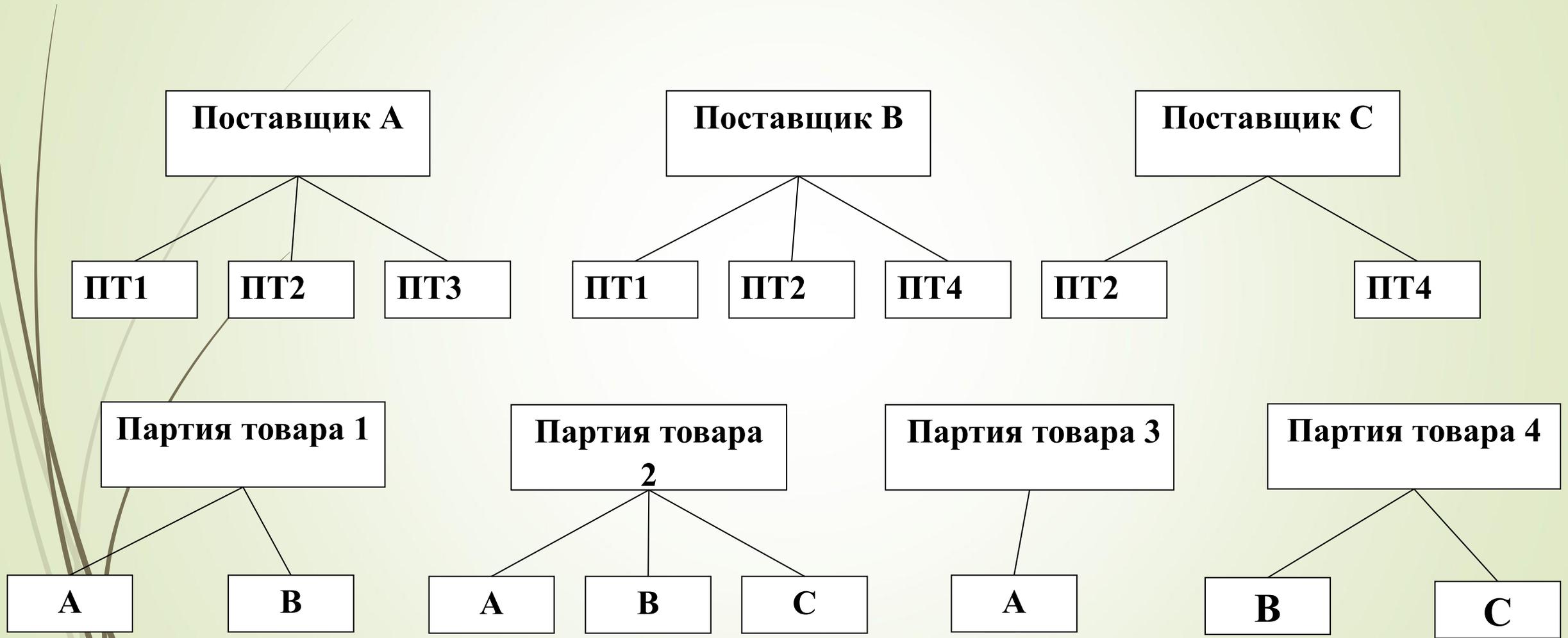
22



1.2 Сетевая модель

23

Пример эквивалентных древовидных структур



Основные операции манипулирования данными:

1. поиск записей в БД;
2. переход от предка к первому потомку;
3. переход от потомка к первому предку;
4. создание новой записи;
5. удаление или обнуление текущей записи;
6. включение (исключение) записи в связь.

Достоинства

- ✓ возможность эффективной реализации по показателям затрат памяти и оперативной обработки;
- ✓ большие возможности по сравнению с иерархической моделью в допустимости образования новой связи.

Недостатки

- ✓ высокая сложность и жесткость схемы БД;
- ✓ сложность понимания и выполнения обработки информации для обычного пользователя;
- ✓ ослаблен контроль целостности связей вследствие допустимости установки программных связей.

Реляционная модель

состоит из трех частей*:

- ✓ Структурной части.
- ✓ Целостной части.
- ✓ Манипуляционной части.

Структурная часть описывает, какие объекты рассматриваются реляционной моделью. Постулируется, что единственной структурой данных, используемой в реляционной модели, являются нормализованные n -арные отношения.

* Крис Дейт. Введение в системы баз данных.

1.3 Реляционная модель

27

Целостная часть описывает ограничения специального вида, которые должны выполняться для любых отношений в любых реляционных базах данных. Это целостность сущностей и целостность внешних ключей.

Манипуляционная часть описывает два эквивалентных способа манипулирования реляционными данными - реляционную алгебру и реляционное исчисление.

1.3 Реляционная модель

28

Таблица отражает тип объекта реального мира (сущность), а каждая ее строка (кортеж) – конкретный объект.

Например, таблица «Сотрудники отдела» содержит сведения обо всех сотрудниках отдела, каждая ее строка – набор значений атрибутов конкретного сотрудник.

Значения конкретного атрибута выбираются из домена (domain) – множества всех возможных значений атрибута объекта. Имя столбца должно быть уникальным в таблице.

| № зач. КНИЖКИ | Фамилия | Имя | Отчество | Группа |
|------------------|---------|------|----------|--------|
| 16493 | Сергеев | Петр | Иванович | М-31 |
| 16593 | Петрова | Анна | Игоревна | М-32 |

1.3 Реляционная модель

29

Строкам таблицы соответствуют кортежи (экземпляры объектов), а столбцам соответствуют атрибуты отношений (свойства объектов). Реляционная модель поддерживает связи типа «один к одному» и «один ко многим». Связи типа «многие ко многим» и рекурсивные связи поддерживаются с помощью декомпозиции.

| Номер заказа | Наименование | Дата заказа | Дата поставки | Наименование детали | Цена | Количество | Итого |
|--------------|--------------|-------------|---------------|---------------------|------|------------|-------|
| | | | | | | | |

| Номер заказа | Наименование | Дата заказа | Дата поставки | Итого |
|--------------|--------------|-------------|---------------|-------|
| | | | | |

| Номер заказа | Наименование детали | Цена | Количество |
|--------------|---------------------|------|------------|
| | | | |

1.3 Реляционная модель

30

Достоинства

- ✓ простота представления данных (таблица);
- ✓ минимальная избыточность данных, что достигается путем нормализации таблиц;
- ✓ независимость приложений пользователя от данных (при включении или удалении таблиц);
- ✓ возможность изменения состава атрибутивных отношений;
- ✓ отсутствие необходимости описывать схемы данных (в иерархических и сетевых моделях — надо).

Недостатки

- ✓ нормализация таблиц приводит к значительной фрагментации данных, а при решении задач, как правило, их необходимо объединять.

1.3 Реляционная модель

31

Доминирование реляционной модели в современных СУБД обусловлено рядом причин, в числе которых:

- наличие развитой теории реляционной модели данных, которая поддерживается теоретическими исследованиями в большей степени по сравнению с другими моделями;
- наличие аппарата приведения к реляционной других моделей данных;
- поддержка реляционной моделью специальных средств ускоренного доступа к информации;
- возможность манипулирования данными без необходимости знания конкретной физической организации базы данных во внешней памяти;
- наличие стандартизованного высокоуровневого языка запросов к

1.3 Реляционная модель

32

Обычно различают три класса СУБД, обеспечивающих работу иерархических, сетевых и реляционных моделей. Можно прогнозировать появление новых классов, связанных с интенсивными разработками в области баз знаний (БЗ) и объектно-ориентированной инфотехнологий.

Каждая из указанных моделей обладает характеристиками, делающими ее наиболее удобной для конкретных приложений.

Для иерархических и сетевых СУБД их структура часто не может быть изменена после ввода данных, тогда как для реляционных СУБД структура может изменяться в любое время.

С другой стороны, для больших баз данных, структура которых остается длительное время неизменной, и постоянно работающих с ними приложений с интенсивными потоками запросов именно иерархические и сетевые СУБД могут оказаться наиболее эффективными решениями, т. к. они могут обеспечивать более быстрый доступ к информации, чем реляционные СУБД.

1.4 Постреляционная модель

33

INVOICES (накладные)

| INVNO (номер накладной) | CASTNO (номер покупателя) |
|--------------------------------|----------------------------------|
| 0373 | 8723 |
| 8374 | 8232 |
| 7364 | 8723 |

INVOICES_ITEMS (накладные_товары)

| INVNO | GOODS | QTY |
|--------------|--------------|------------|
| 0373 | Сыр | 3 |
| 0373 | Рыба | 2 |
| 8374 | Лимонад | 1 |
| 8374 | Сок | 6 |
| 8374 | Печенье | 2 |
| 8364 | Йогурт | 1 |

1.4 Постреляционная модель

34

Классическая реляционная модель предполагает неделимость данных хранящихся в полях записей таблицы (1 нормальная форма).

Постреляционная модель снимает ограничения неделимости данных. Она **допускает многозначные поля**, значения которых состоят из подзначений.

Набор значений многозначных полей считается самостоятельной таблицей.

IVOICES

| INVNO | CASTNO | GOODS | QTY |
|-------|--------|---------|-----|
| 0373 | 8723 | Сыр | 3 |
| | | Рыба | 2 |
| 8374 | 8232 | Лимонад | 1 |
| | | Сок | 6 |
| | | Печенье | 2 |
| 7364 | 8723 | Йогурт | 1 |

1.4 Постреляционная модель

35

Для выполнения постреляционного запроса должны быть реализованы соответствующие механизмы выборки.

Постреляционная модель поддерживает ассоциированные многозначные поля (множественные группы).

Совокупность ассоциированных полей называется **ассоциацией**.

| Предприятие | Количество служащих ж/м | | Средний возраст ж/м | | Средняя зарплата ж/м | |
|-------------|-------------------------|------|---------------------|----|----------------------|-----|
| | ж | м | ж | м | ж | м |
| СевНТУ | 1300 | 1000 | 35 | 48 | 290 | 470 |
| МГУ | 300 | 500 | 37 | 42 | 500 | 600 |

1.4 Постреляционная модель

36

Достоинства

- ✓ высокая наглядность представления информации;
- ✓ более эффективное хранение данных;
- ✓ при обработке данных не требуется выполнять операцию соединения двух таблиц.

Недостатки:

- ✓ сложность решения проблемы обеспечения целостности и непротиворечивости данных (ограничения целостности накладываются на процедурном уровне. Для описания функций контроля значений в полях создаются специальные процедуры: коды конверсии, коды корреляции, автоматически вызываемые во время обращения к данным).

1.5 Многомерная модель данных

37

Существует 2 направления в развитии ИС:

1. системы оперативной или транзакционной обработки информации (*весьма эффективны реляционные модели*);
2. системы аналитической обработки информации (системы поддержки принятия решений) (*эффективны многомерные СУБД*).

Многомерные СУБД являются узкоспециализированными системами, предназначенными для интерактивной аналитической обработки информации.

Основные понятия:

- 1. Агрегируемость данных** — возможность рассмотрения информации на разных уровнях ее обобщения.
- 2. Историчность данных** — предполагает обеспечение высокого уровня неизменности данных и их взаимосвязи, а также обязательность привязки данных ко времени.
- 3. Прогнозируемость данных** — подразумевает задание функции прогнозирования и применение ее к различным временным интервалам.

1.5 Многомерная модель данных

39

Пример:

а) реляционное представление данных

| Модель | Месяц | Объем |
|-----------|--------|-------|
| «Жигули» | Июнь | 12 |
| «Жигули» | Июль | 24 |
| «Жигули» | Август | 5 |
| «Москвич» | Июнь | 2 |
| «Москвич» | Июль | 18 |
| «Волга» | Июль | 19 |

б) многомерное представление

| Модель | Июнь | Июль | Август |
|-----------|------|------|--------|
| «Жигули» | 12 | 24 | 5 |
| «Москвич» | 2 | 18 | № |
| «Волга» | № | 19 | № |

1.5 Многомерная модель данных

40

Измерение (Dimension) – множество однотипных данных, образующих одну из граней многомерного объекта (гиперкуба).

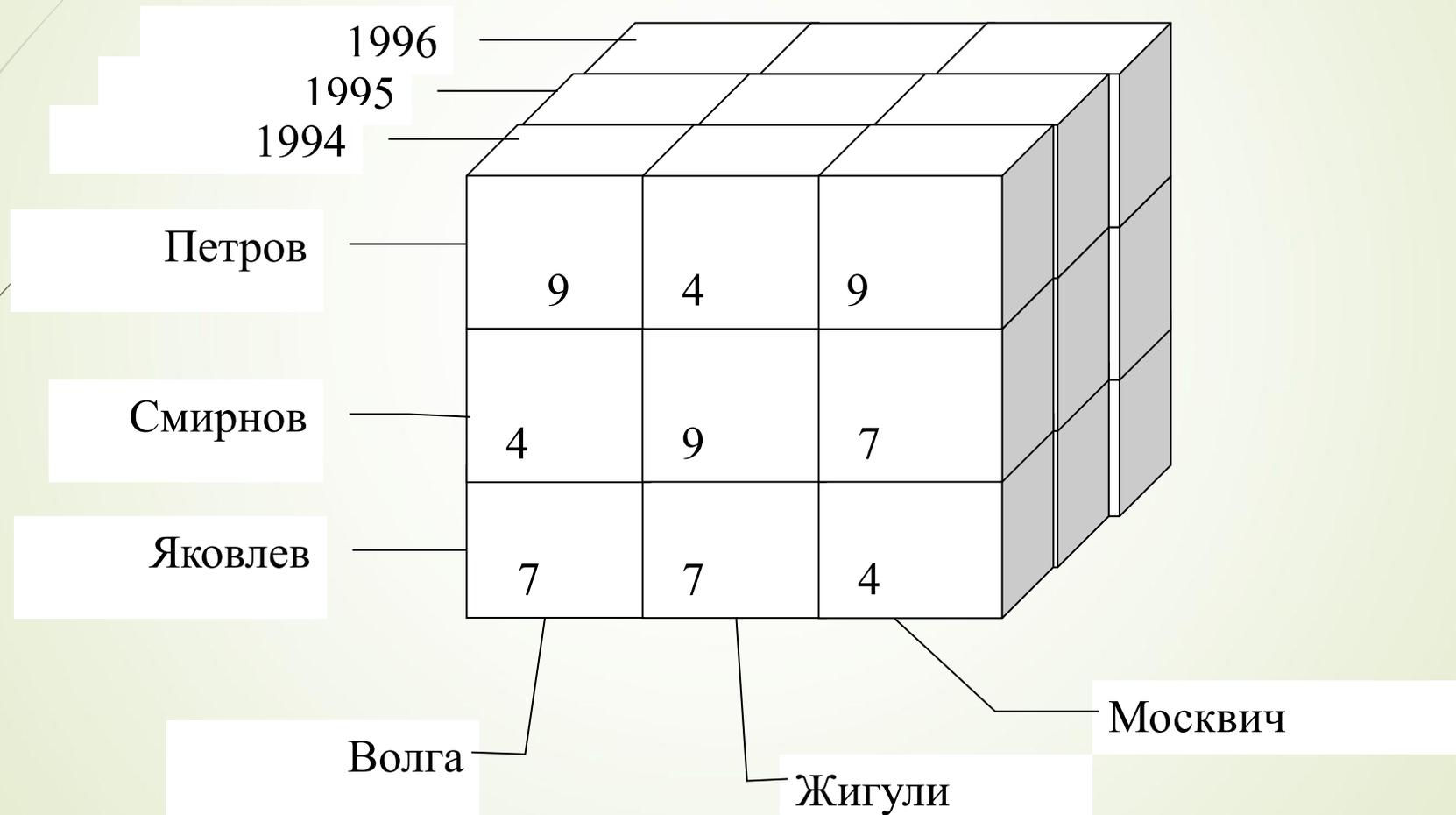
Ячейка гиперкуба (Cell) – поле, значение которого однозначно определяется фиксированным набором измерений (чаще всего числовой тип). В гиперкуб введены измерения и существует **иерархия снизу вверх**.

Операции

1. **Срез** (Slice) – выделение подмножества гиперкуба, полученного в результате фиксации одного или нескольких измерений.
2. **Вращение** (Rotate) – изменение порядка измерений при визуальном представлении информации.
3. **Агрегация** – переход к более общему представлению из гиперкуба.
4. **Детализация** – переход к более детальному представлению из гиперкуба.

1.5 Многомерная модель данных

41



Достоинства

- ✓ Многомерная модель обладает большей наглядностью и информативностью, чем информационная модель.
- ✓ Удобство и эффективность аналитической обработки больших объемов данных, связанных со временем.

Недостатки

- ✓ Громоздкость для простейших задач обычной оперативной обработки информации.

1.6 Объектно-ориентированная модель

43

| БИБЛИОТЕКА | | |
|------------|--------|----------|
| СВОЙСТВО | ТИП | значение |
| район | String | |
| АБОНЕНТ | Class | |
| КАТАЛОГ | Class | |
| ВЫДАЧА | Class | |
| билет | abs | |
| номер | abs | |

| АБОНЕНТ | | |
|---------|--------|----------|
| билет | string | 0015 |
| имя | string | Васильев |
| адрес | string | Мира, 3 |

| ВЫДАЧА | | |
|--------|--------|---------|
| билет | string | 0015 |
| номер | string | 02867 |
| дата | string | 9.01.97 |

| КАТАЛОГ | | |
|----------|--------|-------------|
| isbn | string | 3.217.00628 |
| удк | string | 651306 |
| название | string | БД |
| автор | string | Дейт |
| КНИГА | class | |

| КНИГА | | |
|----------|--------|-------|
| номер | string | 02694 |
| стеллаж | string | 7 |
| название | string | 1 |

1.6 Объектно-ориентированная модель

44

1. Структура объектно-ориентированной БД графически представима в виде дерева, узлами которого являются объекты.
2. Объектом является любой экземпляр любой сущности. Это значит, что имеется возможность идентифицировать отдельные записи БД и определить функции их обработки.
3. Свойства объекта описываются некоторым стандартным типом (например, `string`) или типом, конструированным пользователем (`class`).
4. Каждый объект-экземпляр класса является потомком того объекта, в котором он определен как свойство, и наследует свойство родителя.

1.6 Объектно-ориентированная модель

45

5. Родовые отношения между объектами в БД образуют связную иерархию объектов и реализуются путем задания соответствующих ссылок.
6. Логическая основа БД похожа на иерархическую. Основное отличие – в способах манипулирования данными. Для выполнения действий над данными используются методы того или иного объекта (их логическая ориентация скрыта в описании класса объекта). Эти классы могут иметь разный набор свойств.
7. Поиск в объектно-ориентированной БД состоит в выяснении сходства между объектом, задаваемым пользователем (цель), и объектами, хранящимися в БД.

1.6 Объектно-ориентированная модель

46

Достоинства

- ✓ Возможность отображения информации о сложных взаимосвязях объектов (иерархические).
- ✓ Обеспечение более высокого уровня абстракции при манипулировании данными за счет объектно-ориентированных механизмов инкапсуляции, наследования, полиморфизма.
- ✓ Возможность идентификации отдельных кортежей.

Недостатки

- ✓ Высокая понятийная сложность.
- ✓ Низкая скорость выполнения запросов.