





ТЕОРЕТИКО- ГРАФОВЫЕ МОДЕЛИ ДАННЫХ



1. Иерархическая модель данных




Первая иерархическая СУБД IMS (Information Management System), фирмы IBM.




Поле данных - минимальная неделимая
единица данных, доступная пользователю с
помощью СУБД.

Сегмент - *запись*

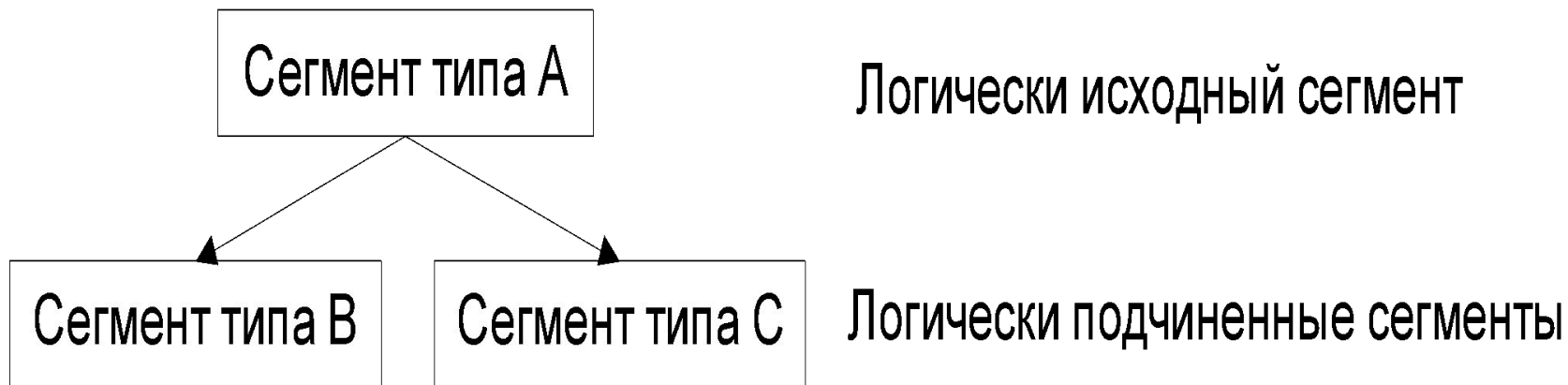


Тип сегмента — это поименованная совокупность типов элементов данных, в него входящих.

Экземпляр сегмента образуется из конкретных значений полей или элементов данных, в него входящих.



Ключом называется набор элементов данных, однозначно идентифицирующих экземпляр сегмента.



Пример иерархических связей между сегментами




Схема иерархической БД представляет собой совокупность отдельных деревьев, каждое дерево в рамках модели называется *физической базой данных*.

Каждая физическая БД удовлетворяет следующим иерархическим ограничениям:

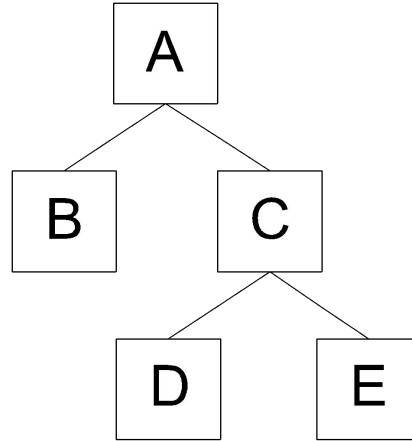
- в каждой физической БД существует один корневой сегмент, т.е. сегмент, у которого нет логически исходного (родительского) типа сегмента;
- каждый логически исходный сегмент может быть связан с произвольным числом логически подчиненных сегментов;
- каждый логически подчиненный сегмент может быть связан только с одним логически исходным (родительским) сегментом.



а)

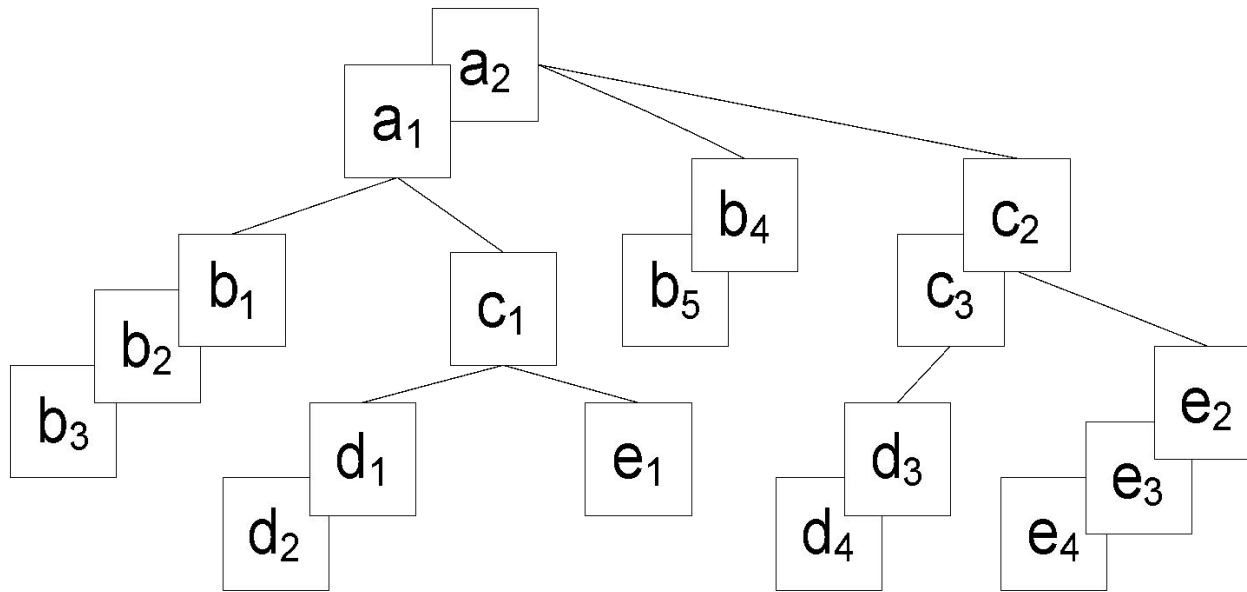


б)



Пример структуры иерархического дерева

Набор всех экземпляров сегментов, подчиненных одному экземпляру корневого сегмента, называется *физической записью*.



Пример двух экземпляров дерева

a₁ b₁ b₂ b₃ c₁ d₁ d₂ e₁	a₂ b₄ b₅ c₂ d₃ d₄ e₂ e₃ e₄
Запись 1	Запись 2



Схема



Экземпляры схемы




**Язык описания данных
иерархической модели
(DDL, Data Definition Language)**

Способ доступа определяет способ организации взаимосвязи физических записей.

- 1) HSAM (*hierarchical sequential access method*) — иерархически последовательный метод;
- 2) HISAM (*hierarchical index sequential access method*) — иерархически индексно-последовательный метод;
- 3) HDAM (*hierarchical direct access method*) — иерархически прямой метод;
- 4) HIDAM (*hierarchical index direct access method*) — иерархически индексно-прямой метод;
- 5) INDEX — индексный метод.

Внешние модели

Внешняя модель представляет собой совокупность поддеревьев для физических БД, с которыми работает данный пользователь. Каждый подграф внешней модели в обязательном порядке должен содержать корневой тип сегмента соответствующий физической БД концептуальной модели.



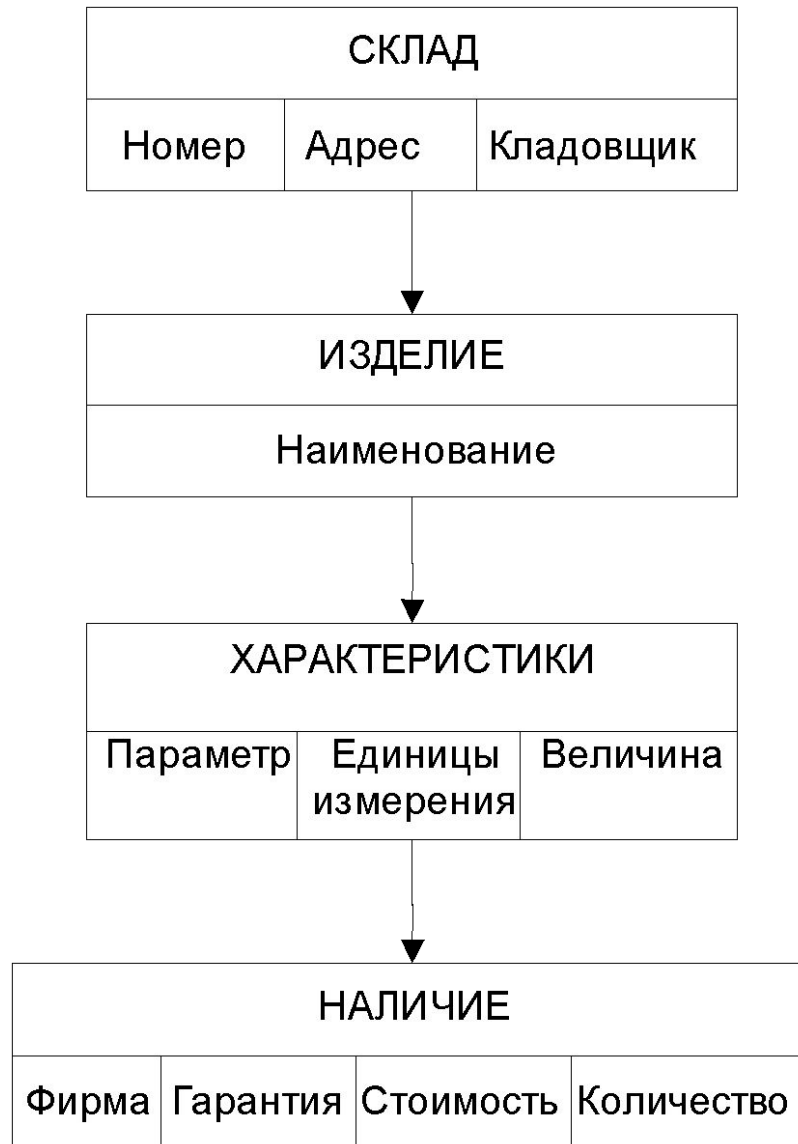
Представление внешней модели называется **логической БД** и определяется совокупностью блоков связи данного приложения с физическими БД, входящими в концептуальную схему БД.

Блок связи — *PCB* (*program communication bloc*)
— описывает связь с одной физической
БД.

Совокупность блоков PCB образует полное
внешнее представление данного
приложения — **блок спецификации
программ** (*PSB, program specifying block*).



Пример иерархической БД





Язык манипулирования данными в иерархических БД (DML, Data Manipulation Language)

1. Операторы поиска данных.

- 1). GET UNIQUE <имя сегмента> WHERE
<список поиска>
- 2). GET NEXT <имя сегмента> WHERE
<список аргументов поиска>
- 3). GET NEXT <имя сегмента> WITHIN
PARENT [where <дополнительные
условия>]

2. Операторы поиска данных с возможностью модификации.

Используются операторы поиска данных, но в синтаксис добавляется слово HOLD:

```
GET HOLD UNIQUE <имя сегмента>  
WHERE <список поиска>
```

3. Операторы модификации данных.

1). DELETE

2). UPDATE

3). INSERT < имя сегмента >

Преимуществами иерархической модели:

- развитые средства управления данными во внешней памяти на низком уровне;
- возможность построения вручную эффективных прикладных систем;
- возможность эффективного использования памяти.


Недостатки иерархической модели :

- асимметрия поиска по симметричным запросам;
- зависимость поиска от соответствия иерархической структуры существующим связям в предметной области;
- низкий уровень языка запросов и манипулирования данными;

- трудность реализации «дружественных» интерфейсов пользователя;
- аномалии вставки, удаления и обновления;
- дублируемость данных;
- трудно реализовывать гибкие механизмы защиты данных, целостности, непротиворечивости.




2. СЕТЕВАЯ МОДЕЛЬ ДАННЫХ



Стандарт сетевой модели впервые был определен в 1975 году организацией CODASYL, которая определила базовые понятия модели и формальный язык описания.

Базовыми объектами модели являются:

- элемент данных;
- агрегат данных;
- запись;
- набор данных.



Элемент данных - минимальная
информационная единица, доступная
пользователю с использованием СУБД.

(то же, что и в иерархической модели)

Агрегат данных соответствует следующему уровню обобщения в модели.

В модели определены агрегаты двух типов: агрегат типа **вектор** и агрегат типа **повторяющаяся группа**.

Адрес			
Город	Улица	дом	квартира

Зарплата	
Месяц	Сумма

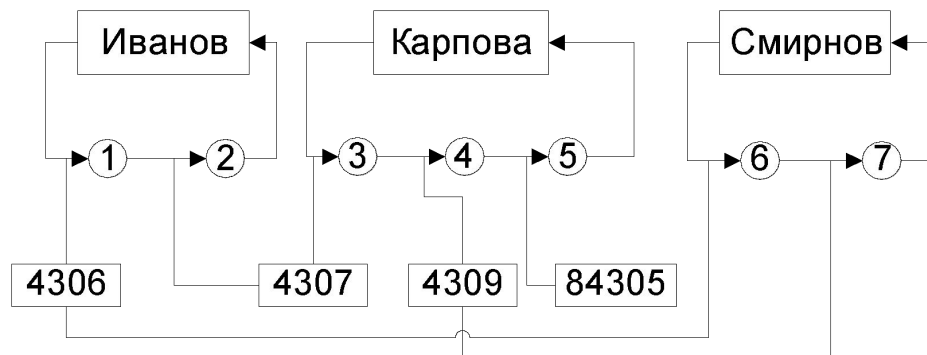
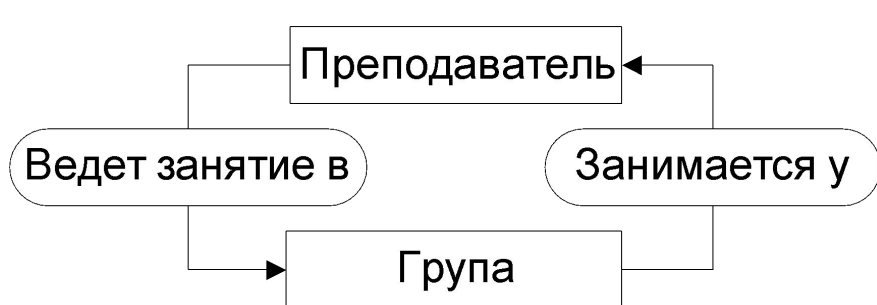
Записью называется совокупность агрегатов или элементов данных, моделирующая некоторый класс объектов реального мира.

(соответствует понятию «сегмент» в иерархической модели. Для записи, так же как и для сегмента, вводятся понятия типа записи и экземпляра записи)

Набором называется двухуровневый граф, связывающий отношением «один-ко-многим» два типа записи.



Преподаватель	Группа	День недели	№ пары	Аудитория	Дисциплина
Иванов	4306	Понедельник	1	22-13	кид
Иванов	4307	Понедельник	2	22-13	кид
Карпова	4307	Вторник	2	22-14	БЗ и ЭС
Карпова	4309	Вторник	4	22-14	БЗиЭС
Карпова	84305	Вторник	1	22-14	БД
Смирнов	4306	Вторник	3	23-07	ГВП
Смирнов	4309	Вторник	4	23-07	ГВП






Язык описания данных в сетевой модели




Язык описания данных имеет разделы:

- описание БД – области размещения;
- описания записей – элементов и агрегатов (каждого в отдельности);
- описания наборов (каждого в отдельности).



Язык манипулирования данными в сетевой модели

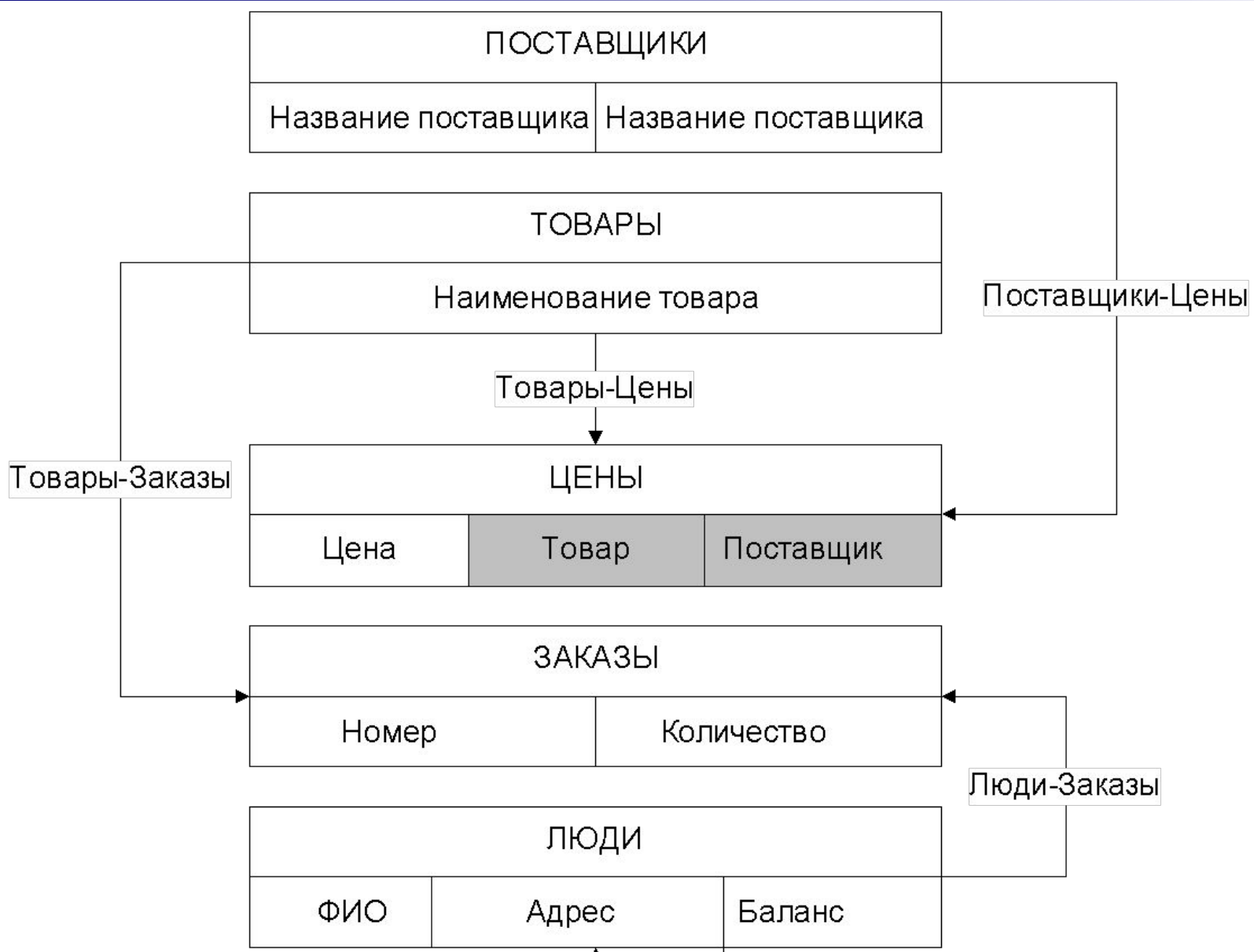
Навигационные операции осуществляют перемещение по БД путем прохождения по связям, которые поддерживаются в схеме БД. В этом случае результатом является новый единичный объект, который получает статус *текущего объекта*.



Операции модификации осуществляют добавление новых экземпляров отдельных типов записей и наборов, удаление экземпляров записей и наборов, модификацию отдельных составляющих внутри конкретных экземпляров записей.

указатели текущего состояния:

- текущая запись процесса (код или ключ последней записи, с которой работала данная программа);
- текущая запись типа записи (для каждого типа записи ключ последней записи с которой работала программа);
- текущая запись типа набор (для каждого набора с владельцем T1 и членом T2 указывается, T1 или T2 были последней обрабатываемой записью).



Контрольные вопросы

- Определите иерархическую структуру данных.
- Каковы операции манипулирования иерархической структурой данных. Приведите примеры их использования.
- Определите сетевую структуру данных.
- Приведите примеры операций сетевой структуры данных. Приведите примеры их использования.