

Лекция №9 27.10.23

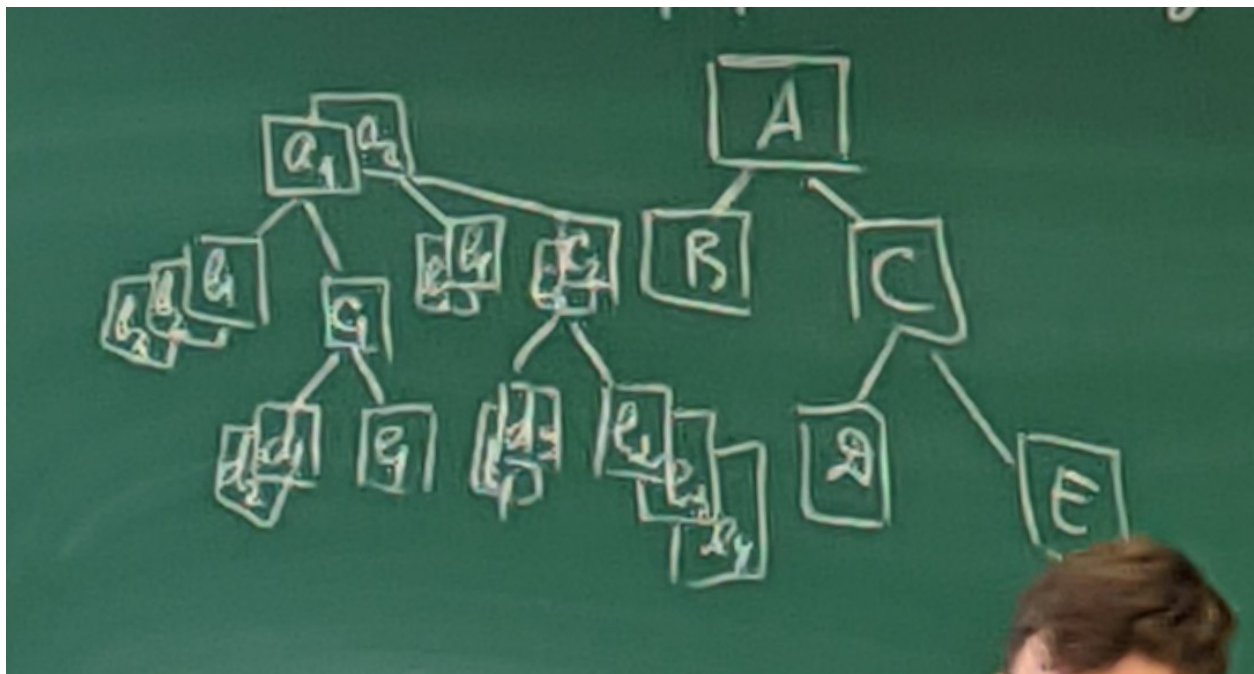
Иерархическая модель данных (продолжение)

Схема иерархической базы данных

Схема иерархической модели - это совокупность отдельных деревьев, каждое дерево в рамках модели называется физической базой данных.

Каждая физическая иерархическая база данных удовлетворяет условиям:

1. В каждой физической базе данных существует 1 корневой сегмент
2. Каждый логически исходный сегмент может быть связан с произвольным количеством подменённых сегментов
3. Каждый подменённый сегмент может быть связан только с 1 исходным сегментом



Пример физической записи:

Запись 1

Запись 2

a1, b1, b2, b3, c1, d1, d2, e1	a1, b4, b5, c2, c3, d3, d4, e2, e3, e4
--------------------------------	----------------------------------------

Описание иерархической базы данных

Физические записи в иерархической модели отличаются по длине и структуре. В рамках иерархической модели выделяют языки - **DDL** , **DML** .

Описание начинается с определения базы данных - оператор определения бд **DBD** `Name<имя>, ACCESS = <тип доступа>` . Где `ACCESS = <тип доступа>` - взаимосвязь физических записей. Их разделяют на:

- **HSAM** - иерархически последовательный метод
- **HISAM** - иерархически индексно последовательный метод
- **EDAM** - иерархически прямой метод
- **HIDAM** - иерархически индексно прямой метод
- **INDEX** - индексный метод

После описания определения базы данных идёт описание наборов данных - там задаётся устройство хранения базы данных и область переполнения. Далее идёт описание типов сегментов. Описание сегментов всегда начинается с описания корневого сегмента:

```
SEGM NAME = <имя сегмента> BYTES = <размер байтов>
FREQ = <> -- среднее кол-во экземпляров данного сегмента
PARENT = <> -- имя родительского сегмента
```

Далее идет описание полей:

```
FIELD NAME = <название поля>
START = <номер байта с которого он начинается>
BYTES = <размер поля>
TYPE = <тип данных>
```

В иерархической модели могут быть только 3 типа данных:

- **X** - шестнадцатеричный
- **P** - десятичный

- **C** - СИМВОЛЬНЫЙ

В **FIELD NAME** можно указать признак последовательности **SEQ**.

Всё заканчивается вызовом процедуры генерации:

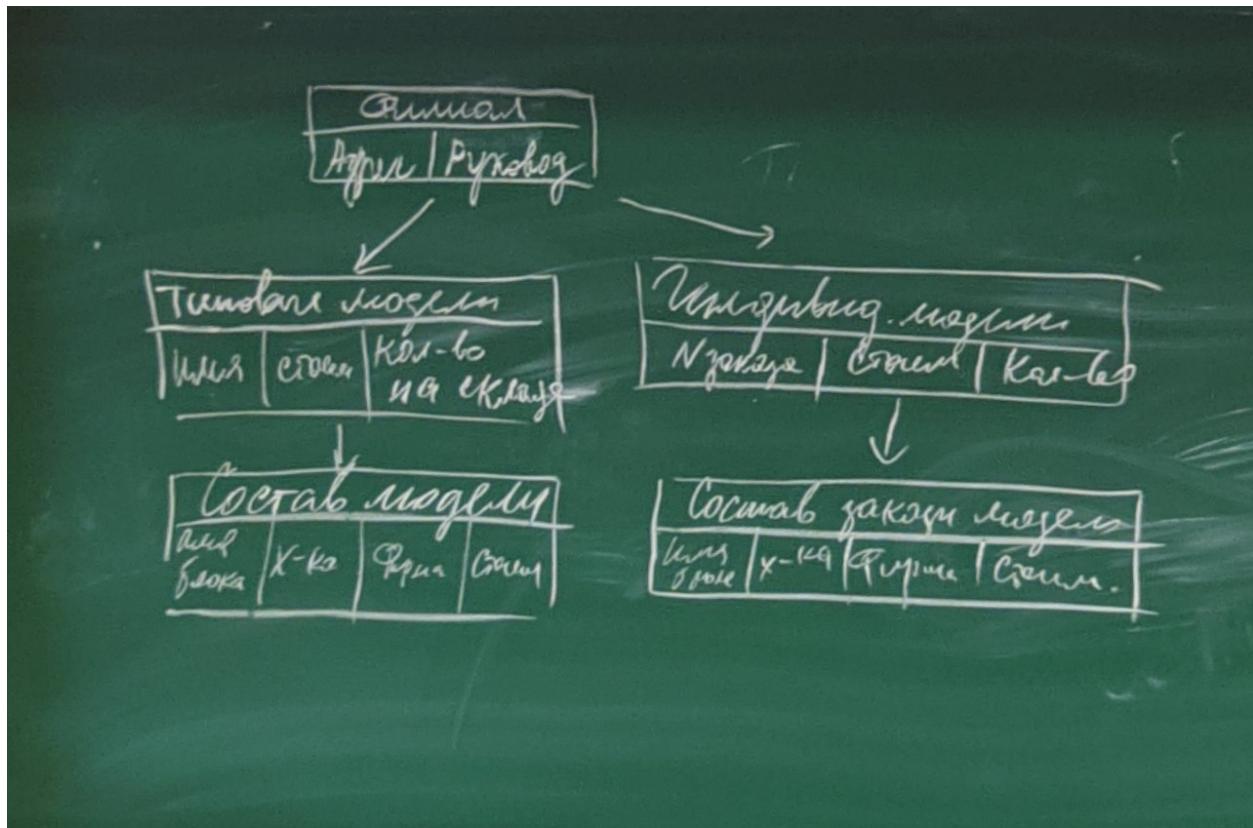
```
DBD GEN
FINISH
END
```

В системе может быть несколько баз данных - все они описываются таким образом.

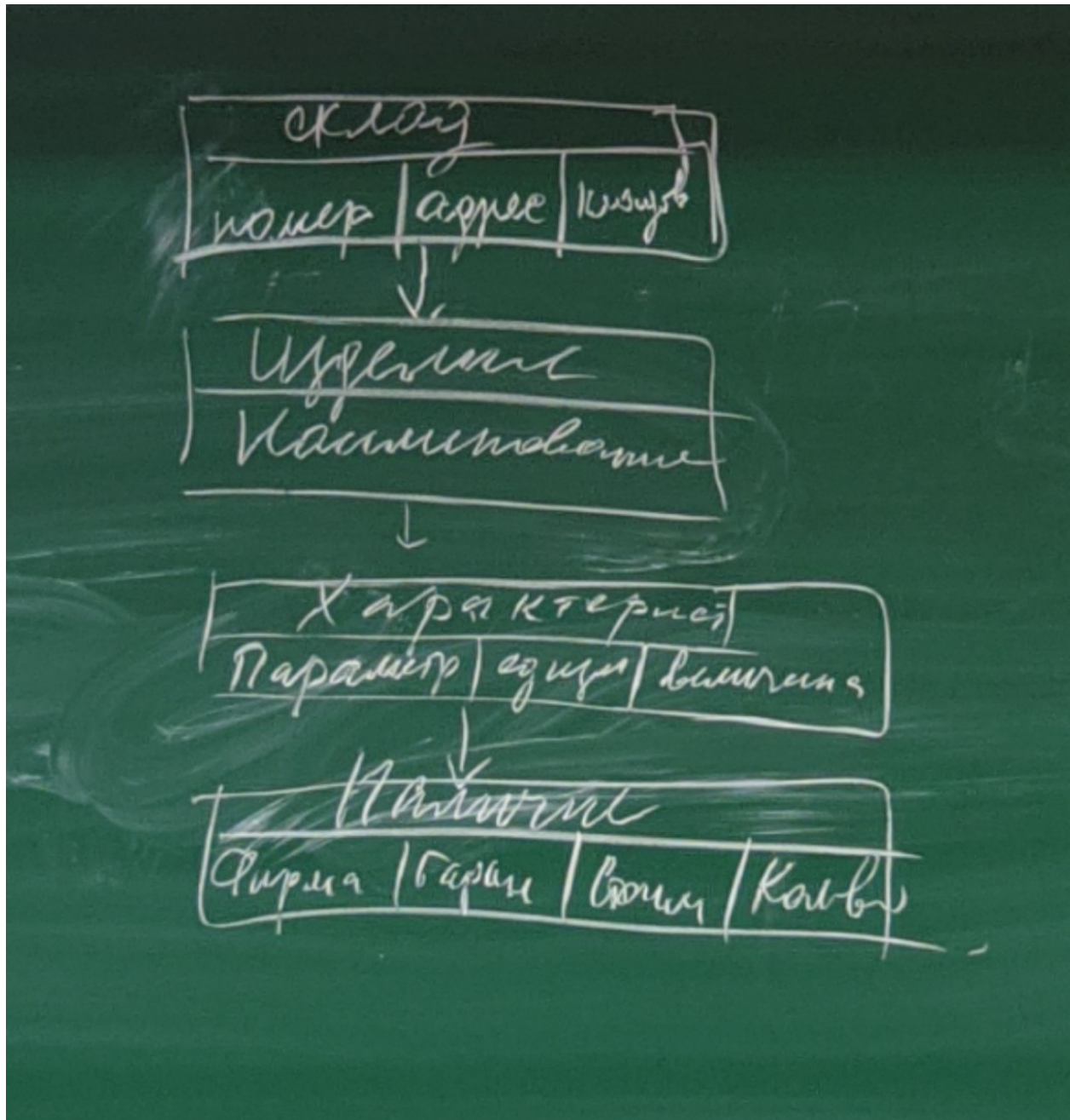
Внешняя модель и навигационные операции

Внешняя модель - совокупность поддеревьев в физической базе данных с которыми работает данный пользователь. Каждый подграф обязательно должен содержать корневой тип сегмента.

Пример физической иерархической базы данных:



Необходимо построить 2 дерево:



Так физически будет выглядеть наша база данных.

В иерархической модели базы данных для доступа к базе данных у пользователя должна быть сформирована специальная среда, которая поддерживает в явном виде навигационные операции. Для этого в ней должны храниться:

- шаблоны всех записей логических баз данных
- указатели на текущий экземпляр сегмента данного типа

Все операции можно разделить на 3 группы:

1. Операторы поиска данных

- `GET UNIQUE... WHERE...`
- `GET NEXT... WHERE...`

2. Операторы поиска данных с возможностью модификации

- `GET HOLD UNIQUE... WHERE...`
- `GET HOLD NEXT... WHERE...`

3. Операторы модификации данных

- `DELETE`
- `UPDATE`
- `INSERT <имя сегмента>`

Способ перемещения от одного сегмента к другому - навигационный способ.

Пример такой иерархической модели - СУБД ИНЕС.

Структура иерархической базы данных

Субд Инес имеет следующую структуру данных:

1. **Терминальная вершина** - наименьшая поименованная единица данных. Для неё должен быть определён тип значения.

```
<Имя терминальной вершины> ::= <тип значения>
ФИО ::= text
```

2. **Структура** - поименованная двухуровневая древовидная иерархическая конструкция.

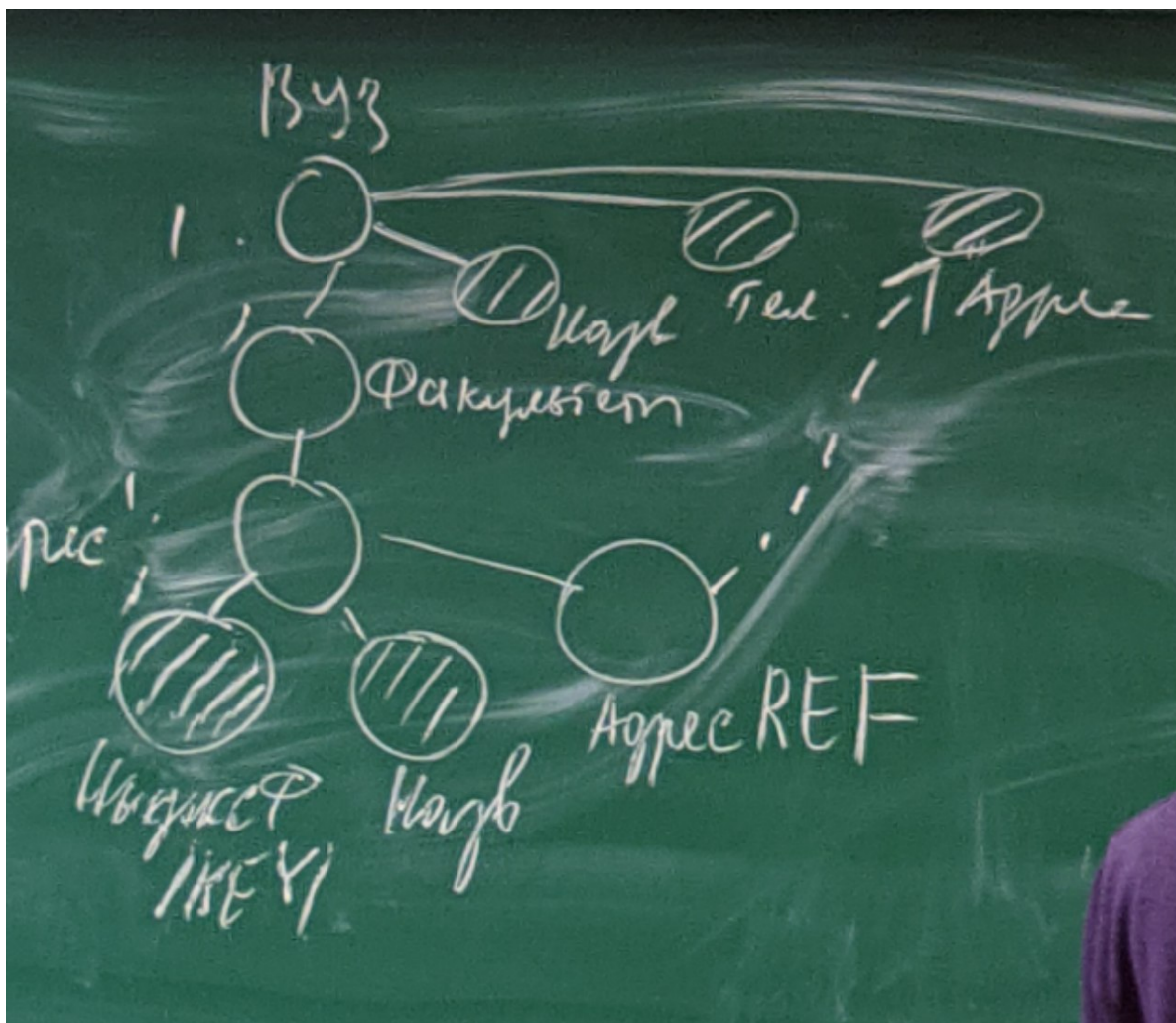
```
<имя вершины структуры> ::= STRUCT(подмененные вершины)
ВУЗ ::= STRUCT(Название, Телефон, Адрес)
```

3. **Массив** - поименованная трехуровневая Древовидная иерархическая конструкция, реализуемая в базе данных n экземпляров однотипных подчинённых структур, имеющая в составе подчинённых терминальных структур первичный ключ. Подчинённых терминальных структур не именуется.

```
<имя вершины>::=ARK(  
    имя первичного ключа |KEY|,  
    имена подчинённой структуры  
)  
ФАКУЛЬТЕТЫ::=ARC(  
    индекс Ф/KEY  
    Название, Адрес  
)
```

4. **Адресная ссылка** - поименованная конструкция для реализации сетевых связей в древовидном графе. Хранит адрес вершины, в которой хранятся данные.

```
<имя вершины адресной ссылки>::=REF'путь доступа от корневой вершины до структуры с данными'  
АдресREF::=REF'вуз.Адрес'
```

5. **База данных (дерево данных)** - поименованная Древовидная иерархическая конструкция с одним экземпляром корневой вершины, то есть совокупность терминальных вершин, структур, массивов, адресных ссылок.

- Никакой потомок не может существовать без родителя.
- Для представления данных используют только листья.
- Типы данных.
- Максимальное количество уровней иерархии в базе данных - 64.

Вывод

Иерархическая база данных хорошо интерпретируется файловой системой. Эта модель ориентирована на запросы, направленные вниз по иерархии. Запросы, направленные вверх по иерархии, более сложные по реализации. Чтобы преобразовать в древовидную модель данных в графическую модель СУБД ИНЕС применяют дублирование и использование адресных ссылок.

Сетевая модель данных

Сетевая модель данных - логическая модель данных, состоящая из элементов данных, записей и связей типа 1:1, 1:M, M:1, M:M.