

# Лекция №10 03.11.23

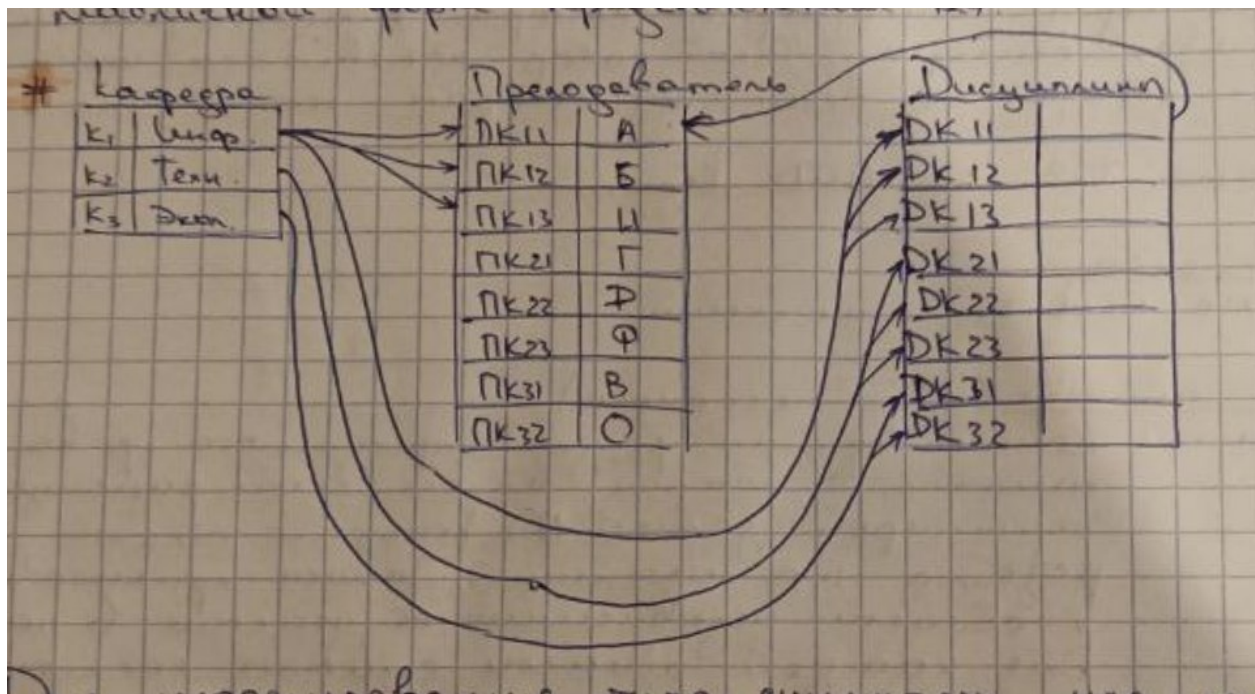
## Сетевая модель данных (продолжение)

**Сетевая база данных** состоит из набора экземпляров определенного типа записей и набора экземпляров определенного типа связей. Для сетевой модели разработан стандарт организации **CODASYL**.

Были введены понятия:

1. **Сетевая схема** - логическая организация всей базы данных
2. **Подсхема** - часть БД, представляемая пользователю или приложению
3. **Язык управления данными**
4. **Язык определения данных**

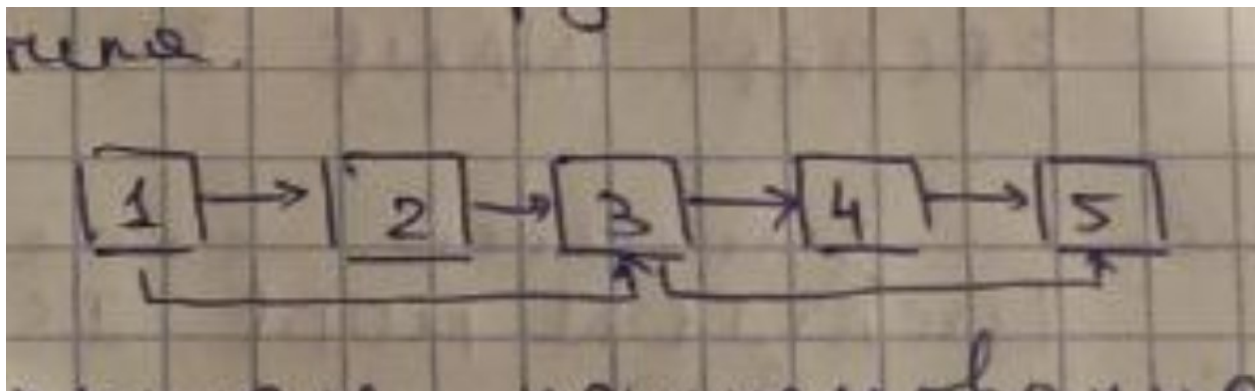
На графической диаграмме сетевой БД вершина графа используется для интерпретации типа сущности, а дуги для интерпретации типов связей. Сетевые МД относятся к классу графо-табличных МД и базируются на использовании графовой и табличной форм представления и организации данных.



Для моделирования типа сущностей используется тип группы (набор экземпляров группы одного типа). Для моделирования экземпляра сущности используется экземпляр группы. Для моделирования типа связи между двумя сущностями используется групповое отношение (набор экземпляров одного типа).

Структура данных в модели:

1. **Элемент данных** - наименьшая поименованная единица данных (аналог поля)
2. **Агрегат данных** - поименованная совокупность элементов данных внутри записи
3. **Запись** - поименованная совокупность элементов данных и агрегатов
4. **Набор** - поименованная совокупность записей, образующих двухуровневую иерархическую структуру - каждый тип набора представляет собой отношение между 2 или несколькими типами записей
5. **БД** - поименованная совокупность экземпляров записей различного типа, содержащая ссылки между записями. Описание структуры БД задается ее схемой



Терминология:

1. Элемент данных
2. Группа - агрегат-запись
3. Групповое отношение. Используется для задания связей между группами
4. БД

В основе создания структур сетевых БД лежат следующие правила:

1. БД может содержать любое количество типов записей и типов наборов

2. Между 2 типами записей может быть определено любое количество типов наборов
3. Тип записи может быть одновременно и владельцем и членом нескольких различных типов наборов
4. Модели реализуют связи **1:M** и **1:1**. Кроме того, в моделях существуют сингулярные наборы. Владелец такого набора - система. Такой набор объявляется, чтобы получить доступ ко всем экземплярам этого типа

## Описание схем на CODASYL

Описание схемы БД и МДД на **CODASYL** :

1. Статья схемы:

**SCHEMA NAME IS <имя\_схемы>**

2. Статья области (1 или несколько):

**Область** - поименованный раздел адресуемого пространства памяти, в котором выполняется размещение экземпляров записей. Каждая область разбита на страницы. Записи приписываются к области независимо от их участия в наборах. Для каждой области данная статья состоит из **AREA IS <имя\_области>**.

3. Статья записей:

**RECORD NAME IS <имя\_записи>**

Способ выбора страниц для записей:

**LOCATION NAME IS <способ\_выбора>**

Способы выбора:

1. **DIRECT** - Номер страницы области будет поступать из программы
2. **CALC <имя\_процедуры>** - задаются, если предполагается, что специальная процедура вычисляет значение ключа.
3. **VIA SET** - если необходимо, чтобы запись оказалась физически ближе к экземпляру набора, в которых она должна быть помещена.
4. **SYSTEM** - если не предполагается использовать рассмотренные выше варианты

`WITH IN <имя_области> AREA` внутренняя структура записи.

Каждая данная записи описывается предложением, которое начинается с номера уровня, затем следует имя данного и тип.

#### 4. Статья набора:

`SET NAME IS <имя_набора>`

Задается имя, затем тип записей владельца:

`OWNER IS {<имя_записи_владельца> SYSTEM}`

Далее способ включения экземпляров записей в экземпляры олицетворения типа набора:

`ORDER IS {SORTED; NEXT; LAST; FIRST}`

Описывается запись член набора `MEMBER IS <имя_записи> {AUTOMATIC, MANUAL}`

`{MANDATORY; OPTIONAL}`, где способы включения: `AUTOMATIC` -

автоматический, `MANUAL` - ручной; Опции: `MANDATORY` - обязательное, `OPTIONAL` - необязательное.

`KEY IS <имя_данного>` - индикаторы текущего состояния.

В начале выполнения имеют нулевое значение. Установка требуемых значений индикаторов можно выполнить оператором `FIND` и поместить через `STORE`. Установленное значение может использоваться далее операторами языка манипулирования данными. Кроме индикаторов вводятся регистры

БД (состояние БД, имя области БД, имя набора БД, имя записи БД).

## Язык манипулирования БД CODASYL

Команды языка `CODASYL`:

1. `READY` (Готовность) - открытие области БД
2. `FINISH` (Завершить работу) - закрытие области БД
3. `FIND` (Поиск) - найти - определяет место расположения экземпляра записи
4. `GET` (Получить) - выборка текущей записи процесса
5. `STORE` (Заполнить) - помещение данных в БД
6. `MODIFY` (Изменить) - обновляет текущую запись

7. **ERASE** (Стереть) - удаление экземпляра записи по соответствующему индикатору текущей записи
8. **CONNECT** (Присоединить) - включение текущей записи в набор
9. **DISCONNECT** (Отделить) - исключение текущей записи из набора

Выполнение операторов ЯМД может завершиться неудачно, поэтому в программе необходим анализ результатов выполнения операторов.

## Графовая модель данных

**Граф** - набор вершин и ребер и взаимосвязи между ними.

**Объекты** - узлы.

Способы, которыми эти объекты соединены между собой - **взаимосвязи**.

Графовые БД позволяют хранить сущности (моделируются узлами) и отношения между ними. Сущности моделируются узлами, которые имеют свойства. Узел интерпретируется как экземпляр объекта приложения. Отношения моделируются ребрами, которые могут иметь свойства. Ребра имеют направления, это позволяет находить требуемые шаблоны среди узлов.

Система управления графовыми БД поддерживает весь **CRUD-процесс** (методы **CREATE**, **READ**, **EDIT** и **DELETE**).

Графовые БД поддерживают систему транзакций реального времени **OLTP**.

## Графовая СУБД Neo4j