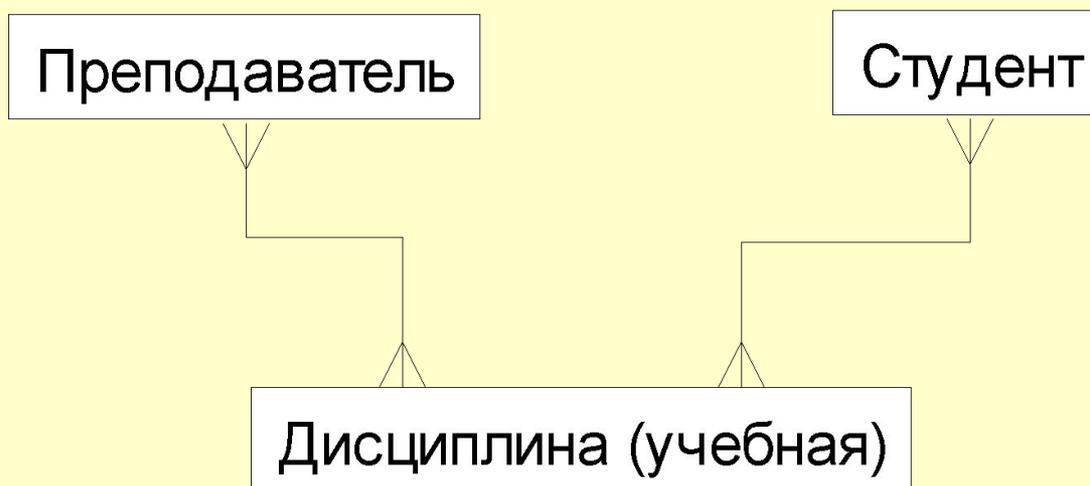


Лекция 14

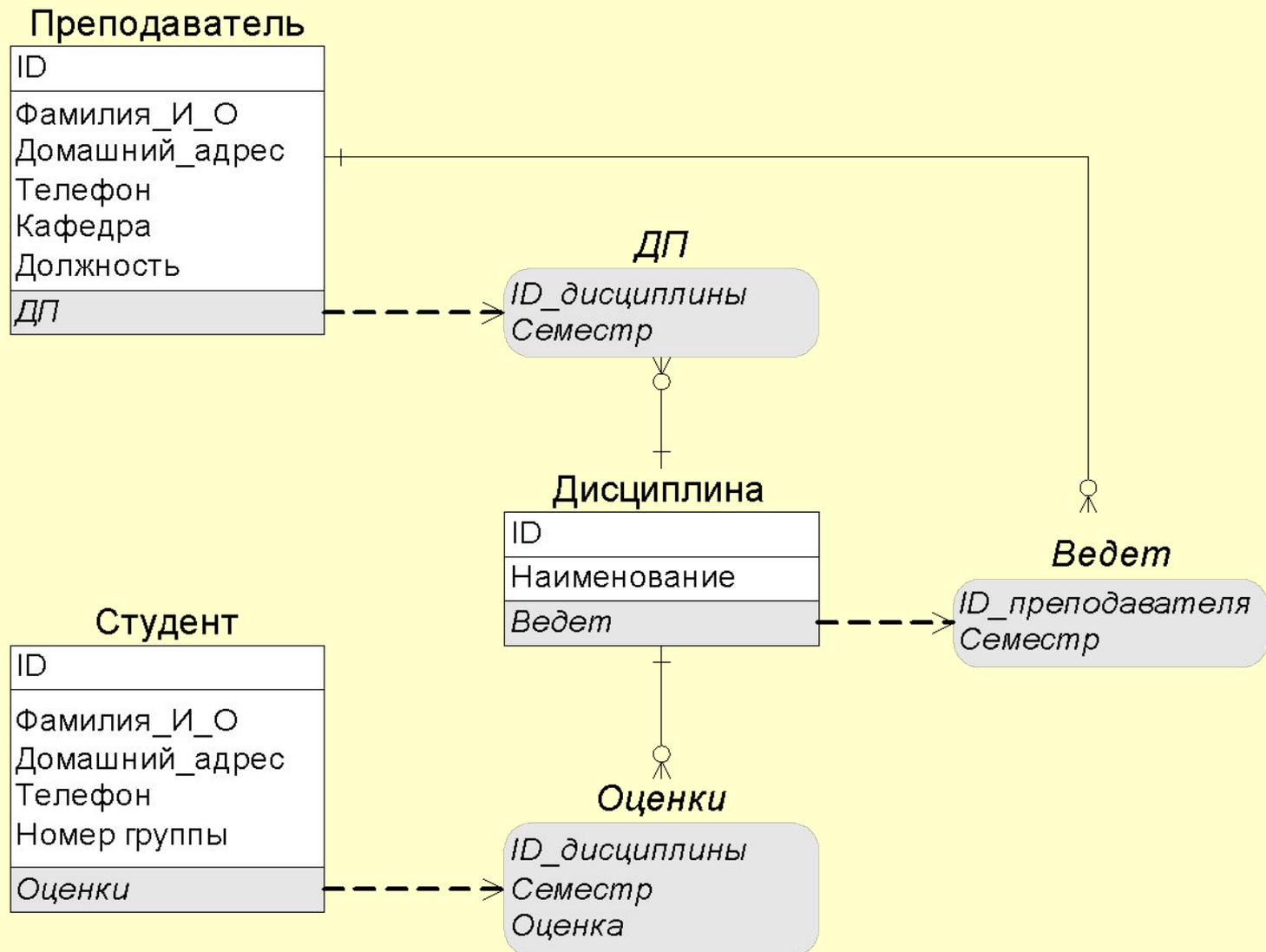
**Постреляционные, объектные модели
данных. Технологии интеграции
распределенных данных на основе XML**

Ограничения нормализации

- Требование атомарности значений атрибута
- Запрет внутренней структуры атрибута



Не первая нормальная форма (НФ2) (Множеств. атрибут как вложенная таблица)



Пример фрагмента постреляционной БД

Преподаватель

ID	Фамилия И.О.	Дом. адрес	Телефон	Кафедра	Должность	Дисциплина	Семестр
2	Зайчиков А.А.	1 7	1 1

Дисциплина

ID	Наименование	Семестр	Преподаватель
1	Алг. и стр. данных	1	2
2	Английский язык	1	10
3	История Отечества	1	5
...			
7	Программирование	1	2

Студент

ID	Фамилия И.О.	Дом. адрес	Телефон	№ группы	Дисциплина	Семестр	Оценка
1	Иванов В.П.	2	1	1
					3	1	5
				
					8	1	5

Пример структуры в СУБД Postgres

```
CREATE TYPE TMarks AS (Дисциплина INTEGER, Семестр INTEGER,
    Оценка INTEGER)
CREATE TYPE TDiscip AS (Дисциплина INTEGER, Семестр INTEGER)

CREATE TABLE Персона (
    ID INTEGER PRIMARY KEY, ФИО CHARACTER VARYING(50),
    Домашний_адрес CHARACTER VARYING(100), Телефон CHARACTER VARYING(20),
)

CREATE TABLE Преподаватель (
    Кафедра INTEGER, Должность CHARACTER VARYING(20),
    ДП TDiscip ARRAY[]
) Персона

CREATE TABLE Студент (
    Номер_группы CHARACTER VARYING(10),
    Оценки TMarks ARRAY[]
) Персона

CREATE TYPE TPrep AS (Преподаватель INTEGER, Семестр INTEGER)

CREATE TABLE Дисциплина (
    ID INTEGER PRIMARY KEY, Наименование CHARACTER VARYING(50),
    Ведет TPrep ARRAY[]
)
```

Достоинства и недостатки постреляционной модели данных

Достоинства:

- более естественное описание предметной области
- меньшие трудозатраты

Недостатки:

- повышенные требования к вычислительным ресурсам

Базовые понятия ОО подхода

- Класс (тип)
- Объект (экземпляр)
- Атрибут (данные)
- Метод (функция)

Базовые механизмы ОО подхода

- Инкапсуляция (атрибутов и методов в объекте)
- Наследование (свойств родительского класса)
- Полиморфизм (одноименные методы класса)
- Абстракция (спецификация интерфейсов взаимодействия объектов)

Уровни ОО моделирования

- Анализ (модель предметной области)
- Проектирование (концептуальная модель системы)
- Реализация (спецификация системы с учетом языка программирования)

Универсальный язык моделирования (UML)

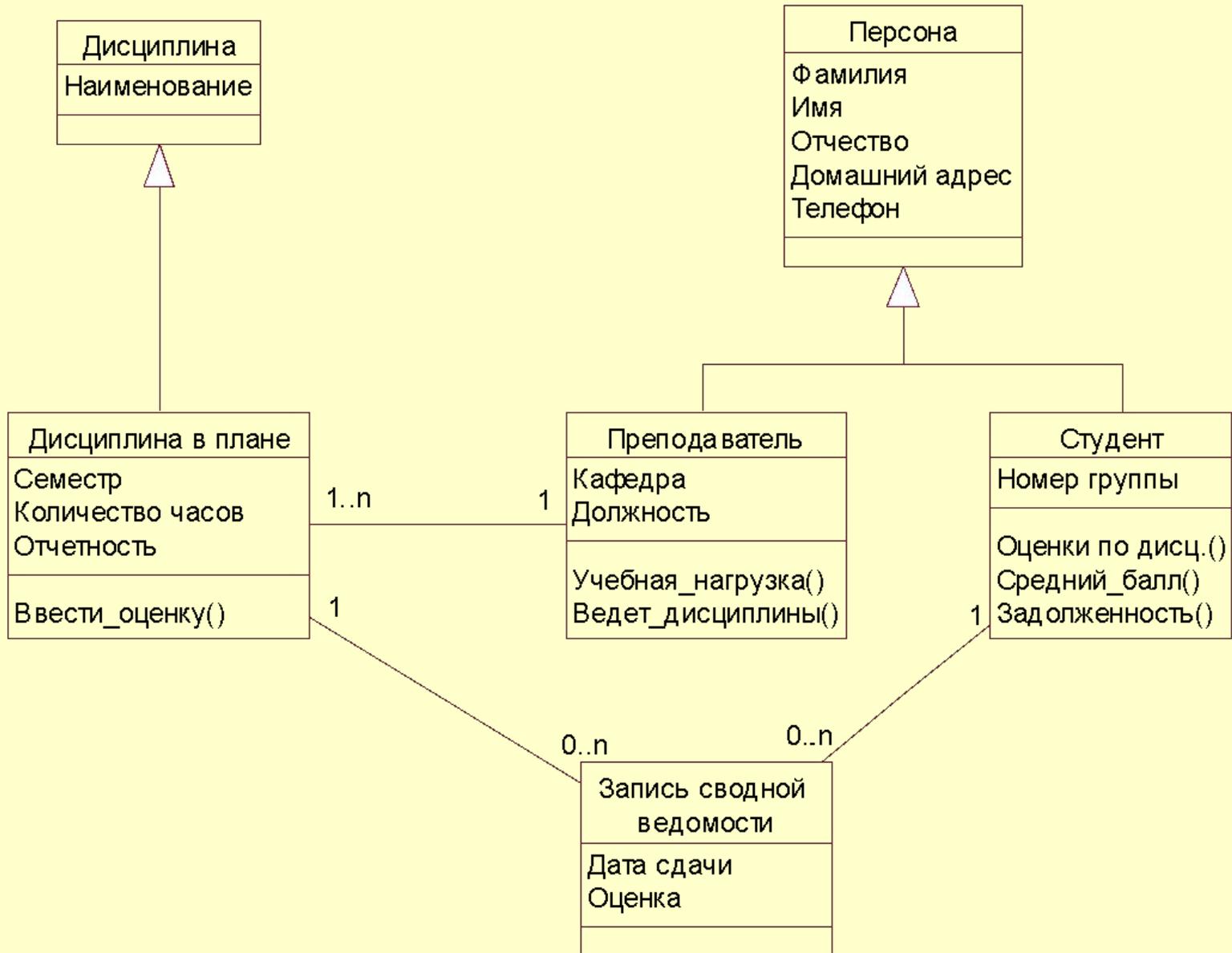
Виды диаграмм:

- вариантов использования (use-case);
- классов (class);
- объектов (object);
- взаимодействия (interaction):
 - последовательности (sequence);
 - кооперативных (collaboration);
- пакетов (package);
- состояний (statechart);
- деятельности (activity);
- размещения (deployment).

ОО модель уровня анализа (задача «Сессия»)



ОО модель уровня проектирования (задача «Сессия»)



Способы реализации ОО подхода в СУБД

- «Интерфейсный» (классы-интерфейсы для доступа к таблицам и записям) – MS SQL Server
- Смешанный (объектно-реляционный) – Oracle Database
- «Чистый» - Cache'

Достоинства и недостатки ОО подхода

Достоинства:

- близость ОО концепции к восприятию мира, свойственному человеку;
- потенциально большее быстроедействие систем;
- отсутствие деления системы на базу данных и программу;
- упрощение описания (предметной области, системы).

Недостатки:

- громоздкость описания ОО модели;
- сложность перехода (смены парадигмы);
- специфические проблемы (идентификация объектов, формализация языков запросов и т.п.).

Технологии XML

- **XML (Extensible Markup Language)** - язык разметки, описывающий класс объектов данных, называемых XML-документами
- **Объектная модель документа (DOM)** - представление структуры и содержания документа в виде совокупности узлов, каждый из которых имеет свои свойства
- **Языки определения XML-документов** - DTD и XML-схемы
- **Спецификация Namespaces** - средства уточняющие обозначения к именам элементов и ссылкам
- **Языки указателей (XPath), ссылок (XLink), запросов (XQuery)**
- **Язык стилей и преобразований (XSL, XSLT)**

Синтаксис XML

- 1) В XML все элементы должны иметь закрывающий тэг
- 2) В тэгах XML учитывается регистр
- 3) Элементы XML должны быть правильно вложены друг в друга
`<i>Этот текст пишется полужирным курсивом</i>`
- 4) XML-документы должны иметь единственный корневой элемент
- 5) Значения атрибутов всегда должны быть заключены в кавычки
- 6) Все пробелы являются значимыми
- 7) В XML есть несколько зарезервированных символов (<, >, &, “, ‘), которые используются только как элементы синтаксиса XML

XML-документ = пролог + тело

- Пролог: *объявление XML* и *объявление типа документа*
<?xml version="1.1" encoding="UTF-8" standalone="yes"?>
<!DOCTYPE sampledoc SYSTEM "sample.dtd">
- Тело XML-документа :
 - ✓ элементы и атрибуты
 - ✓ секции CDATA
 - ✓ директивы анализатора
 - ✓ комментарии
 - ✓ спецсимволы
 - ✓ текстовые данные

Элемент данных - структурная единица XML-документа

```
<flower> rose </flower>
<root>
  <child>
    <subchild>.....</subchild>
  </child>
</root>
```

Атрибуты - определяют собственные характеристики элемента:

Название = "значение"

```
<color RGB="true">#ff08ff</color>
<color RGB="false">white</color>
```

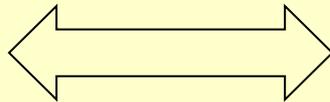
Сущности и специальные символы

< **>** **"**;

$ (десятичная форма записи),

**** (шестнадцатеричная форма записи)

"<"



ссылка на сущность

`<message>if salary < 1000 then</message>`

XML и реляционная модель данных

Таблица Customers

CustID	Name	Phone
1001	Иванов И.И.	5111222
1002	Петров П.П.	5222111

1) в атрибутной форме

```
<Customers>
```

```
<Customer CustID='1001' Name='Иванов И.И.' Phone='5111222'/>
```

```
<Customer CustID='1002' Name='Петров П.П.' Phone='5222111'/>
```

```
</Customers>
```

2) в элементной форме

```
<Customers>
```

```
<Customer>
```

```
<CustID>1001</CustID>
```

```
<Name>Иванов И.И.</Name>
```

```
<Phone>5111222</Phone>
```

```
</Customer>
```

```
<Customer>
```

```
<CustID>1002</CustID>
```

```
<Name>Петров П.
```

```
П.</Name>
```

```
<Phone>5222111</Phone>
```

```
</Customer>
```

```
</Customers>
```

3) в смешанной форме

```
<Customers>
```

```
<Customer CustID='1001'> Иванов
```

```
И.И.
```

```
<Phone>5111222</Phone>
```

```
</Customer>
```

```
<Customer CustID='1002'> Петров
```

```
П.П.
```

```
<Phone>5222111</Phone>
```

```
</Customer>
```

```
</Customers>
```

Представление связей с помощью XML

Таблица Order

OrderNo	Date	Customer
1235	01/01/2001	1001
1236	07/07/2007	1002

Таблица Item

ItemNo	OrderNo	ProductID	Price	Quantity
1	1235	1432	12.99	2
2	1235	1678	11.49	1
3	1236	1432	12.99	3

```
<Orders>
<Order OrderNo='1235' Date='01/01/2001' Customer='1001'>
<Item ProductID='1432' Price='12.99' Quantity='2'/>
<Item ProductID='1678' Price='11.49' Quantity='1'/>
</Order>
<Order OrderNo='1236' Date='07/07/2007' Customer='1002'>
<Item ProductID='1432' Price='12.99' Quantity='3'/>
</Order>
</Orders>
```