КОНЦЕПТУАЛЬНОЕ ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ

Жизненный цикл БД и концептуальное проектирование

Этап начальной разработки	• Анализ деятельности компании • Постановка задачи и определение ограничений • Определение целей • Определение сферы действия и возможностей
Проектирование базы данных	• Концептуальное проектирование • Выбор программного обеспечения СУБД • Логическое проектирование • Физическое проектирование
Реализация и загрузка	• Установка СУБД • Создание базы данных • Загрузка или конвертирование данных
Тестирование и оценка	• Тестирование базы данных • Настройка базы данных • Оценка базы данных и ее прикладных программ
Функционирование	• Организация необходимых информационных потоков
Сопровождение	• Внесение изменений • Развитие
Вывод из эксплуатации	• Осознание необходимости вывода из эксплуатации • Подготовка к выводу из эксплуатации • Утилизация

Процесс проектирования базы данных – последовательность переходов от неформального словесного описания информационной структуры предметной области к формализованному описанию объектов предметной области в терминах некоторой модели.

Предметная область – часть реального мира, подлежащая изучению с целью организации управления и, в конечном счете, автоматизации.

Проектирование базы данных

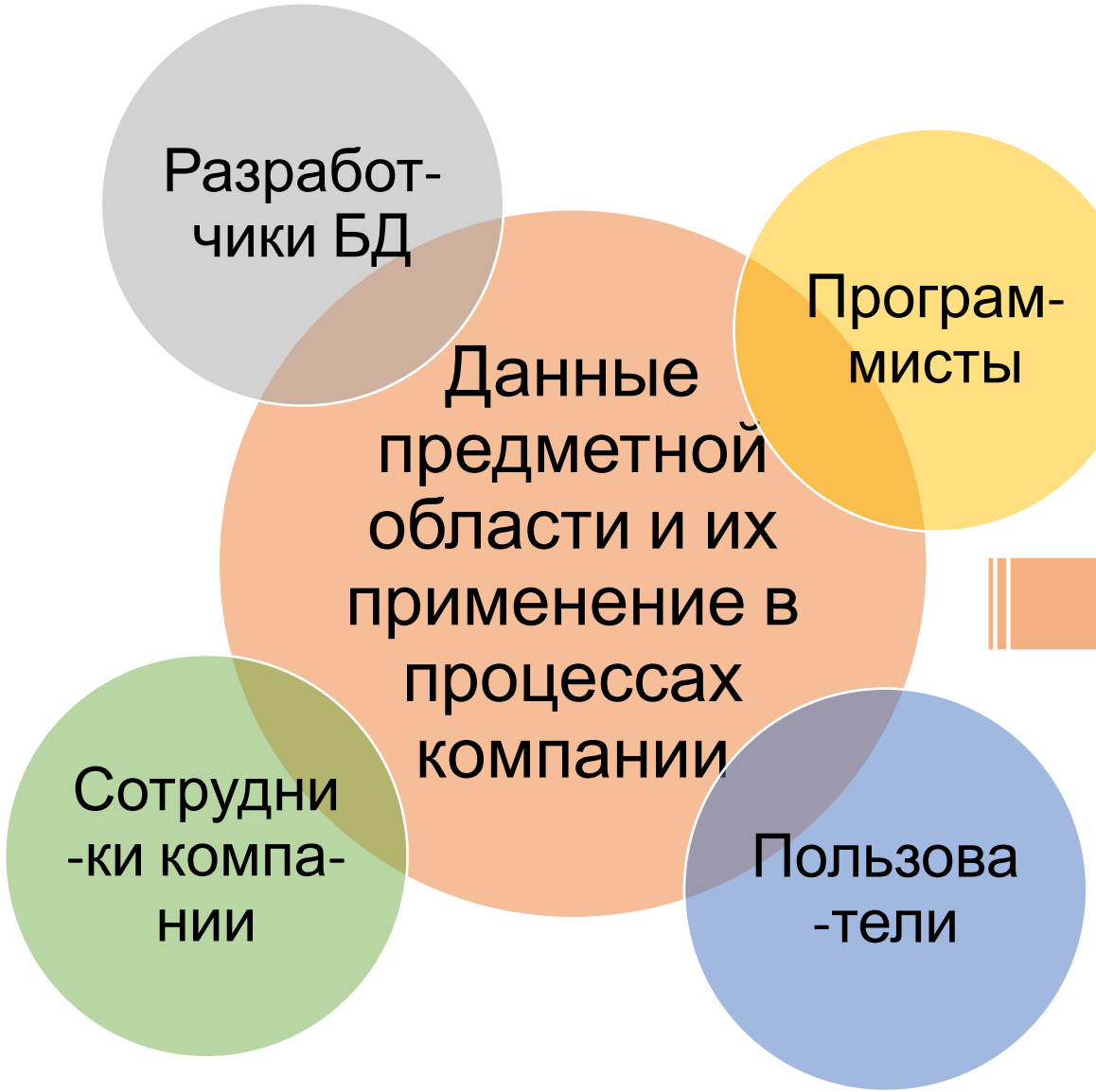
1. **Концептуальное проектирование**
2. Выбор программного обеспечения СУБД
3. Логическое проектирование
4. Физическое проектирование

Концептуальное проектирование базы данных

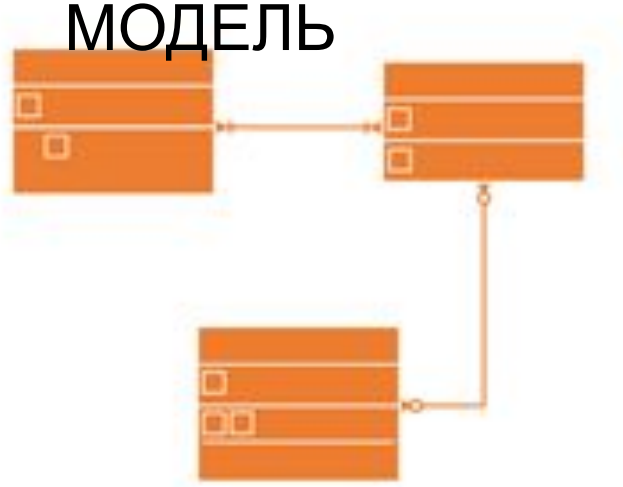
- На этапе концептуального проектирования для создания абстрактной структуры базы данных применяются **методы семантического моделирования** (другие, не вполне корректные названия: моделирование данных, ER-моделирование, моделирование сущностей, объектное моделирование).

Семантическое моделирование

2



ОБЩАЯ МОДЕЛЬ



- ❑ не усложнена техническими подробностями
- ❑ не допускает различных толкований

- Семантическое моделирование = концептуальное моделирование, используется для проектирования БД
- Результат концептуального проектирования – концептуальная модель.
- Концептуальная модель должна выражать ясное понимание сферы деятельности предприятия и выполняемых им функций.
- На данном уровне абстракции тип оборудования или используемая модель базы данных (реляционная, сетевая и др.) не определяются, поэтому концептуальный проект базы данных не должен зависеть от оборудования и программного обеспечения.

Основные семантические концепции

ПОНЯТИЕ	ОПРЕДЕЛЕНИЕ	ПРИМЕР
ТИП СУЩНОСТИ (ENTITY TYPE) **СУЩНОСТЬ** (ENTITY)	Группа объектов реального мира, обладающих одинаковыми свойствами Объект реального мира	Студент, дисциплина, оценка Работник, отдел, человек Поставщик, деталь, поставка и др.
ЭКЗЕМПЛЯР СУЩНОСТИ	Конкретный представитель данной сущности	Петров И.И., История, хорошо Сидоров В.В., бухгалтерия MOBIS, масляный фильтр
СВОЙСТВО (PROPERTY) = АТРИБУТ	Элемент информации, описывающий сущность	ФИО, название дисциплины, виды оценки Отдел работника, рост человека Номер поставщика, поставляемое количество деталей и др.
СВЯЗЬ (RELATIONSHIP)	Сущность, служащая для реализации взаимодействия между двумя или более сущностями	Успеваемость (дисциплина – оценка) Должность (работник – отдел) Поставка (поставщик – деталь)
ПОДТИП (SUBTYPE)	Сущность типа Y является подтипом сущности типа X тогда и только тогда, когда каждый экземпляр сущности типа Y обязательно является экземпляром сущности типа X	«Работник» – подтип сущности «Человек»

Определения в предыдущей таблице носят неформальный характер, так как являются концепциями «реального мира», а не формальными терминами

⇒

гибкость интерпретации семантического моделирования

⇒

один и тот же объект реального мира может быть представлен одними пользователями в качестве сущности, другими – в качестве свойства, а третьими – в качестве связи

ER-модель

Один из наиболее известных и получивших широкое распространение методов семантического моделирования является **метод построения модели «сущность—связь»**

Модель «сущность-связь»

Entity-Relationship model

ER-модель

E – **E**ntity – сущность

R – **R**elationship – связь

Технология построения диаграмм под названием «ER-диаграммы»

Предложена Ченом в 1976 году

Нотации

- Нотация Чена
- Нотация UML
- Нотация Мартина
- Нотация IDEF1X
- Нотация Баркера
- нотация Rain85;
- Нотация «птичья лапка» (Crow’s Foot, «воронья лапка») и т. д.


Компоненты ER-модели
Сущность
Атрибут
Связь

Сущность

1. Элемент реального мира, который может существовать независимо [2].
2. Класс однотипных объектов, информация о которых должна быть учтена в модели [3]

Примеры: СТУДЕНТ, ДИСЦИПЛИНА, ОЦЕНКА.

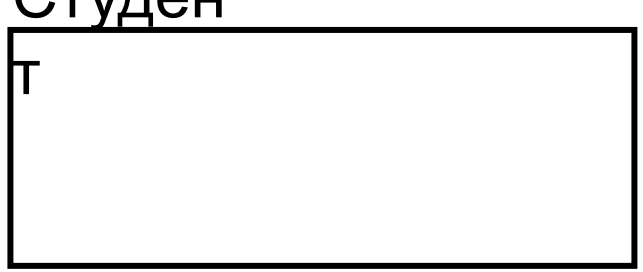
Нотация Чена

Изображается в виде прямоугольника, в котором прописными буквами указывается имя сущности в единственном числе.

СТУДЕНТ

Нотация IDEF1X

Изображается в виде прямоугольника, над которым помещается название сущности (уникальное)



Сущности:

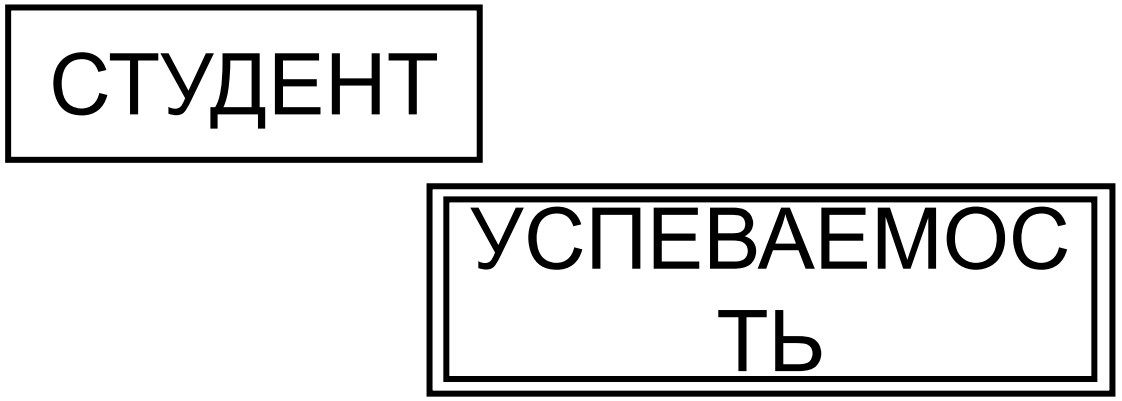
- обычные (сильные, независимые);
- слабые (зависимые).

Слабая сущность – сущность, существование которой зависит от другой (обычной) сущности, т.е. слабая сущность не может существовать, если обычной сущности не существует.

Примеры: СТУДЕНТ, ДИСЦИПЛИНА – обычные (сильные) сущности, УСПЕВАЕМОСТЬ – слабая сущность.

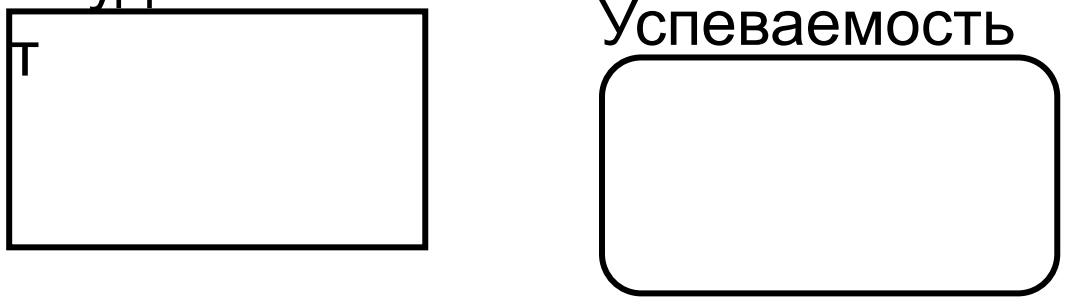
Нотация Чена

Слабая сущность изображается в виде прямоугольника с двойным контуром



Нотация IDEF1X

Слабая сущность изображается в виде прямоугольника со скругленными краями



Экземпляр сущности (entity occurrence)

1. Однозначно идентифицируемый объект, который относится к сущности определенного типа [4]
2. Конкретный представитель данной сущности [3]

Примеры: «Иванов Иван Иванович 12.06.2002 Хабаровск Ленина 16-44».

- Экземпляры сущностей должны быть *различимы*, т.е. сущности должны иметь некоторые **свойства, уникальные для каждого экземпляра этой сущности**.
- На ER-диаграмме экземпляры сущностей не отображаются.

Атрибут (свойство) сущности

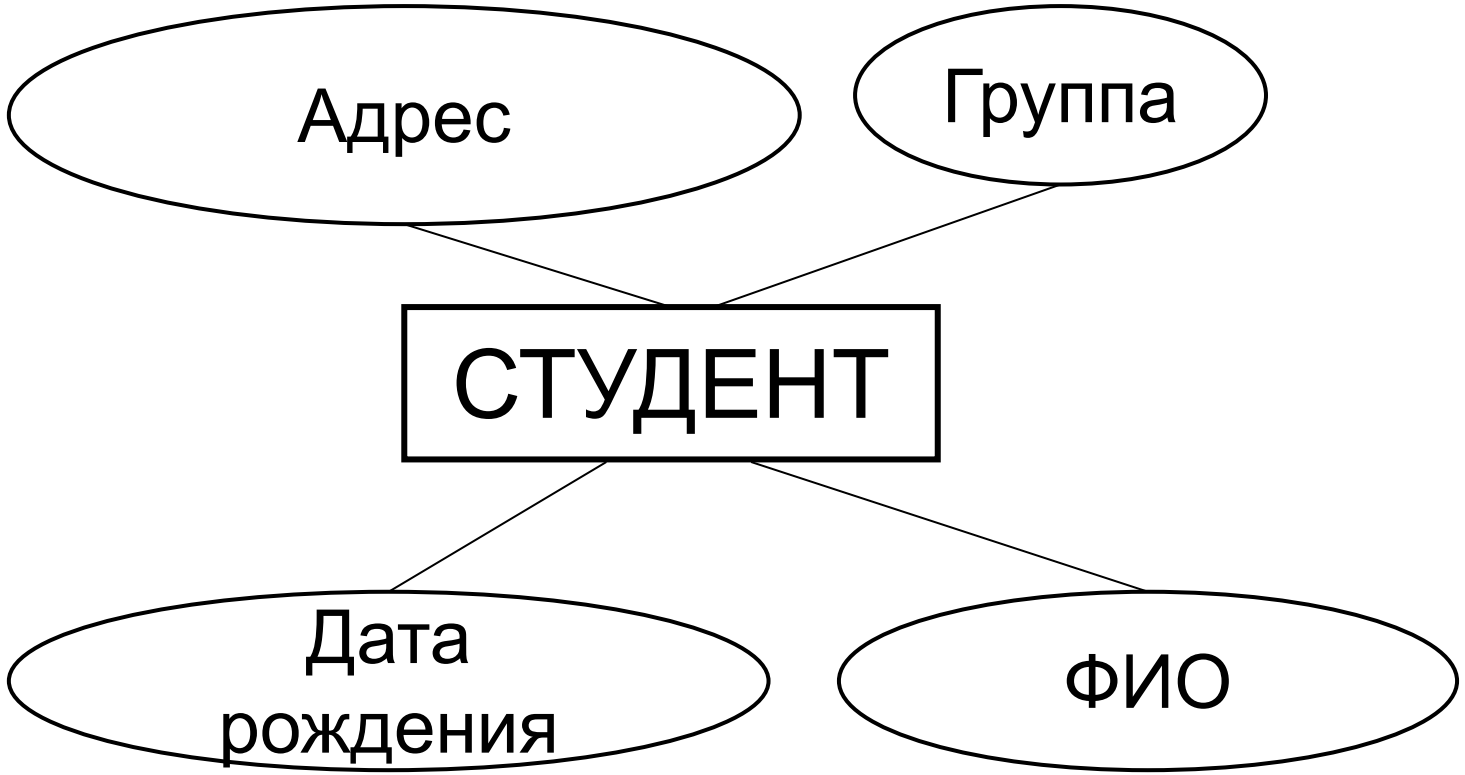
1. Именованная характеристика, являющаяся некоторым свойством сущности [3]

- Атрибуты содержат значения, которые описывают каждый экземпляр сущности и составляют основную часть информации, сохраняемой в базе данных
- Наименование атрибута должно быть выражено существительным в единственном числе (возможно, с характеризующими прилагательными).

Примеры: «Год рождения», «Фамилия», «Имя», «Отчество» – атрибуты сущности «СТУДЕНТ»

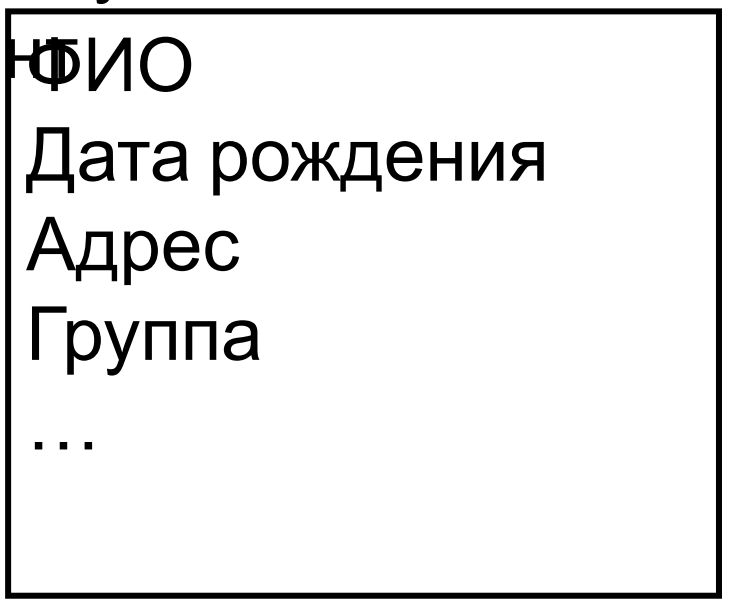
Нотация Чена

Изображается в виде овала с указанием наименования атрибута, соединяется с элементом «сущность» прямой линией



Нотация IDEF1X

Имена атрибутов указываются внутри прямоугольника, изображающего сущность



Атрибуты

Простые/ составные

Ключевые

Однозначные/ многозначные

Базовые/ производные

Простые / составные атрибуты

Простой атрибут – это

1 атрибут, состоящий из одного компонента с независимым существованием [4]

2 атрибут, который не может быть разделен на атрибуты без потери смысла значения.

Пример: атрибут «Название города» (значения атрибута: «Хабаровск», «Владивосток», «Москва» и т.д.).

Составной атрибут – это

1 атрибут, состоящий из нескольких компонентов, каждый из которых характеризуется независимым существованием [4]

2 атрибут, значение которого составлено из значений простых атрибутов.

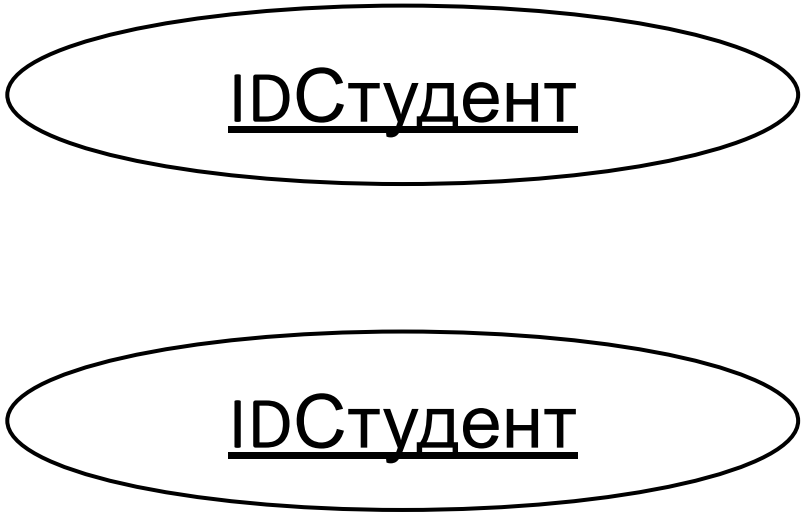
Пример: атрибут «Адрес» (значения атрибута: «Хабаровск, Ленина, 14, 44»).

Ключевые атрибуты

Ключевой атрибут – часть сущности, которая однозначно ее идентифицирует.

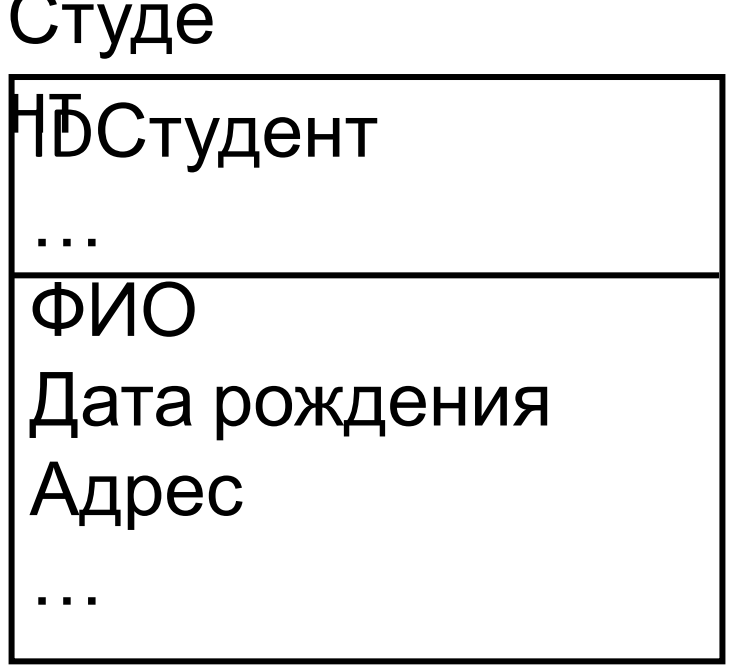
Нотация Чена

Имя ключевого атрибута подчеркивается одной чертой. Имя атрибута, являющегося внешним ключом – двойной чертой.



Нотация IDEF1X

Прямоугольник, представляющий сущность, делится на две части. В верхней части записывается имя ключевого атрибута (или нескольких), в нижней – список остальных имен атрибутов.



Однозначные / многозначные атрибуты

Однозначный атрибут – атрибут, который может содержать единственное значение для каждого экземпляра сущности.

Пример: атрибут «Год рождения».

Многозначный атрибут – атрибут, который может содержать множество значений для каждого экземпляра сущености.

Пример: атрибут «Телефон» (если студент указал несколько номеров телефона, то данный атрибут будет многозначным).

Базовые / производные атрибуты

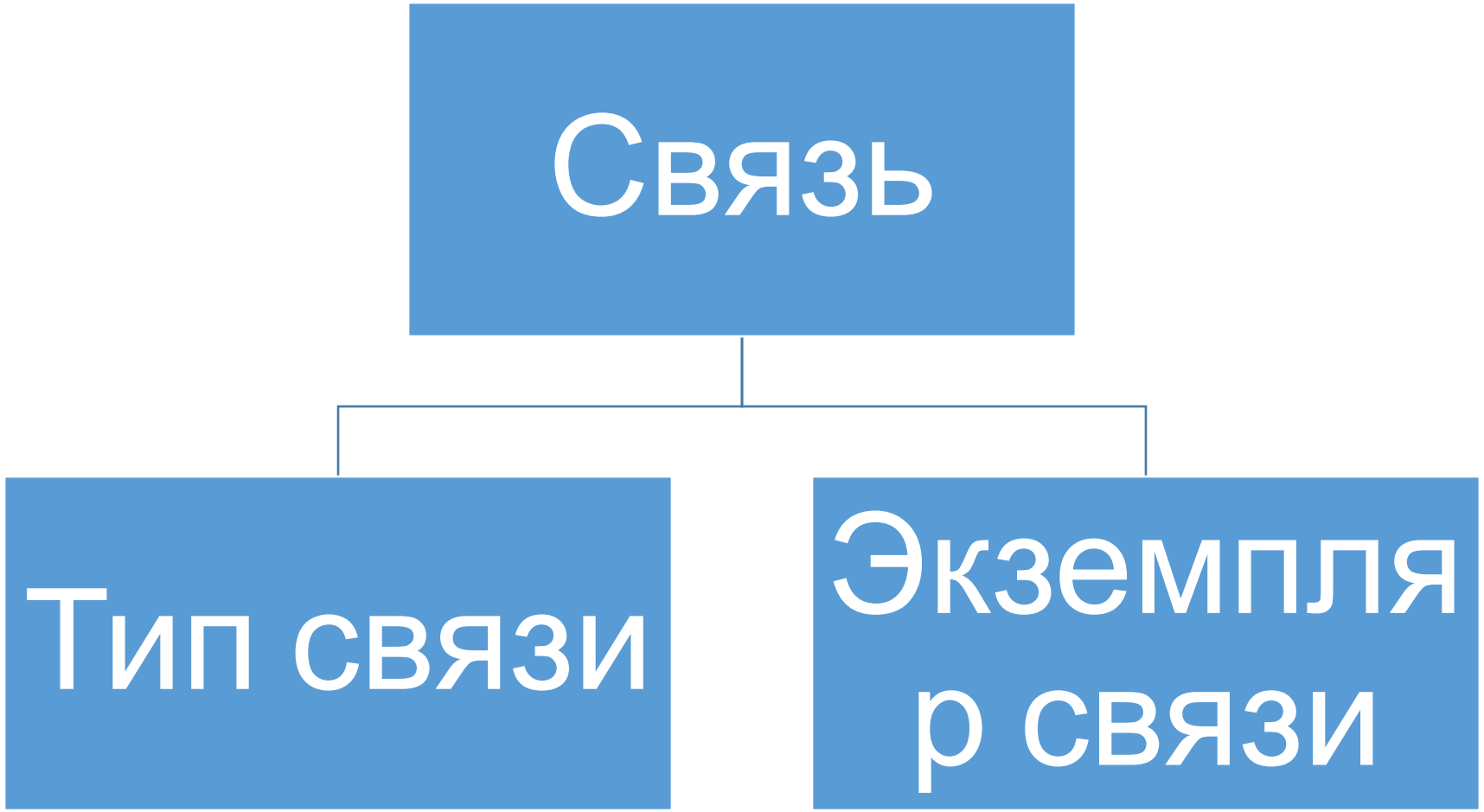
Базовый атрибут – атрибут, который хранится в базе данных.

Пример: атрибут «Дата рождения».

Производный атрибут – атрибут, значение которого можно получить с помощью некоторого алгоритма.

Пример: атрибут «Возраст» (вычисляется с использованием значения атрибута «Дата рождения» и текущей даты).

Связь


Связь
Тип связи
Экземпляр связи

Тип связи

- **Тип связи** (relationship type) – набор ассоциаций между одним (или несколькими) типами сущностей, участвующими в этой связи [5].

Каждому типу связи присваивается имя, которое должно описывать его назначение.

Примеры:

Сущность 1	**Сущность 2**	**Имя типа связи**	**Реальный мир**
СТУДЕНТ	ДИСЦИПЛИНЫ	Изучает	Студент изучает дисциплины
ПРЕПАРАТ	АПТЕКА	Продажи	Препарат продается в аптеке
СОБСТВЕНН	КВАРТИРА	Владение	Собственник владеет квартирой

Экземпляр связи

- **Экземпляр связи** (relationship occurrence) – однозначно идентифицируемая ассоциация, которая включает по одному экземпляру сущности из каждого участвующего в связи типа сущности [5].

Экземпляр связи обозначает все конкретные экземпляры сущности, участвующие в этой связи.

Пример (тип связи «Изучает»):

Экземпляр сущности 1	**Экземпляр сущности 2**	**Имя типа связи**	**Реальный мир**
Иванов Иван Иванович, 12.05.1999, БО921ПИА, …	История	Изучает	Иванов И.И. изучает историю
Иванов Иван Иванович, 12.05.1999, БО921ПИА, …	Физика	Изучает	Иванов И.И. изучает физику
Петров Петр Петрович, 23.02.2002, БО931САП, …	Англ. Яз.	Изучает	Петров П.П. изучает английский язык

Обычно вместо терминов «тип связи» и «экземпляр связи» применяется более общий термин «связь» (если это не приводит к неоднозначности).

Связь:

1. Набор ассоциаций между одной или несколькими сущностями, участвующими в этой связи.
2. Некоторая ассоциация между *двумя* сущностями. Одна сущность может быть связана с другой сущностью или сама с собой [3].
3. Связь представляет собой взаимодействие между сущностями [2].

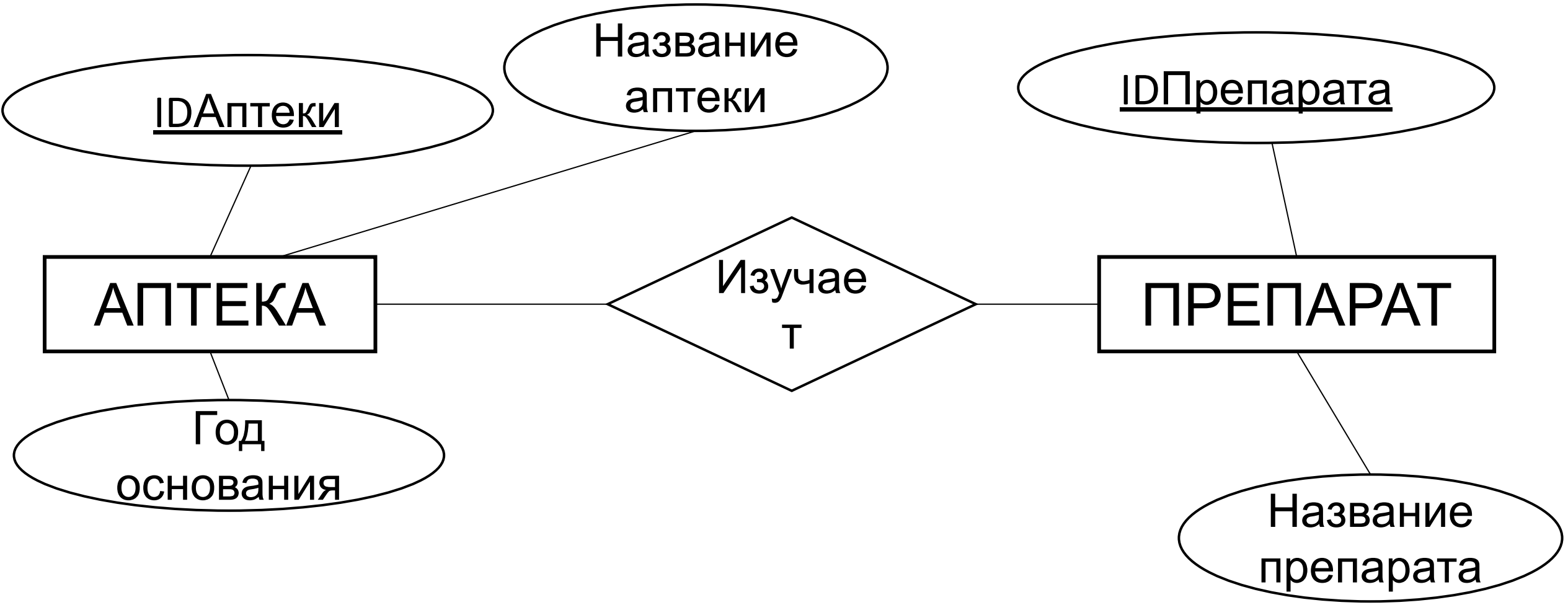
Связи позволяют по одной сущности находить другие сущности, связанные с нею.

Схематическое изображение типов связей

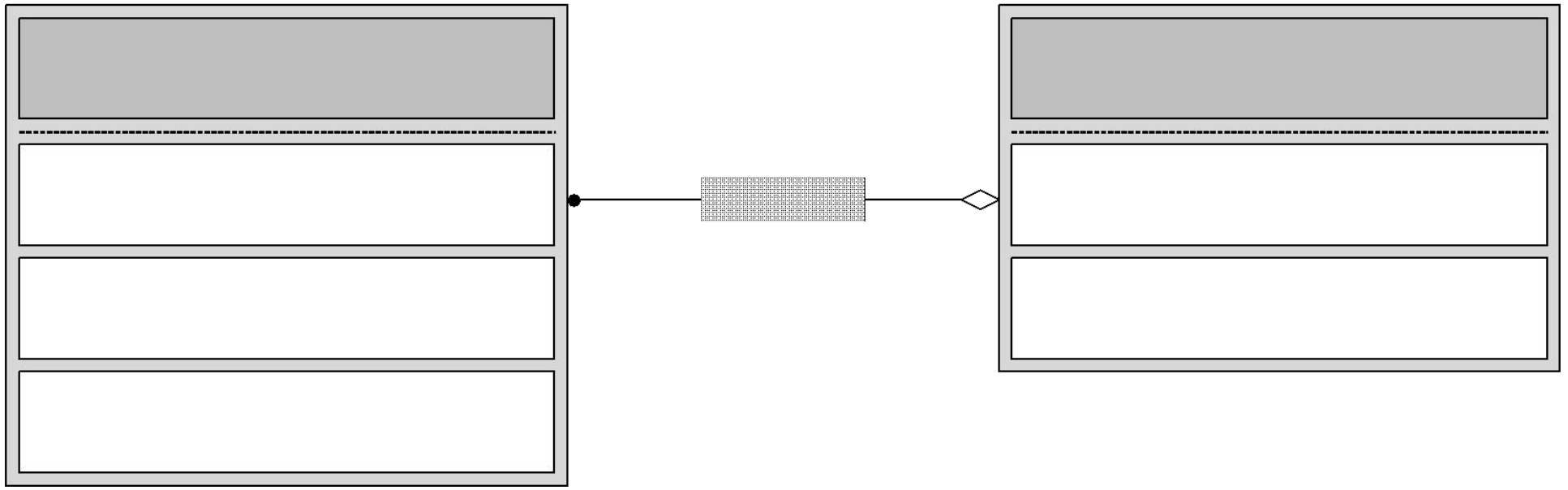
- Каждая связь изображается в виде линии, соединяющей соответствующие сущности
- В качестве имени связи принято использовать глагол (Арендует) или короткую фразу с глаголом (Взят В Аренду), первая буква каждого слова в имени связи является прописной
- По возможности в ER-модели все имена связей должны быть уникальными

Нотация Чена

Изображается линией, которая соединяет сущности, участвующие в связи.
Название связи – в ромбе.

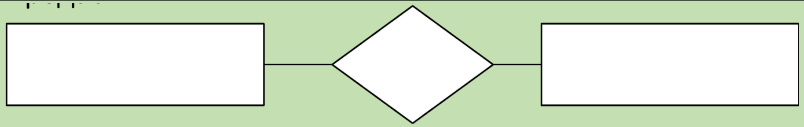
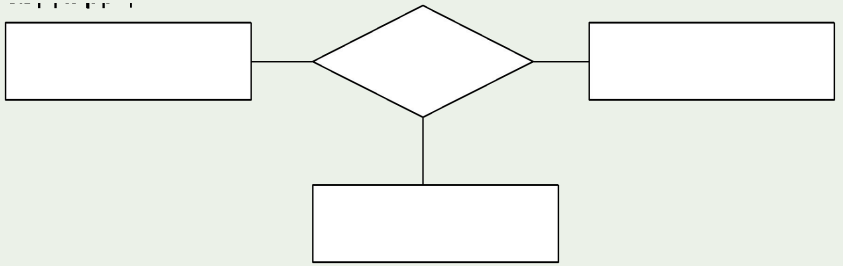
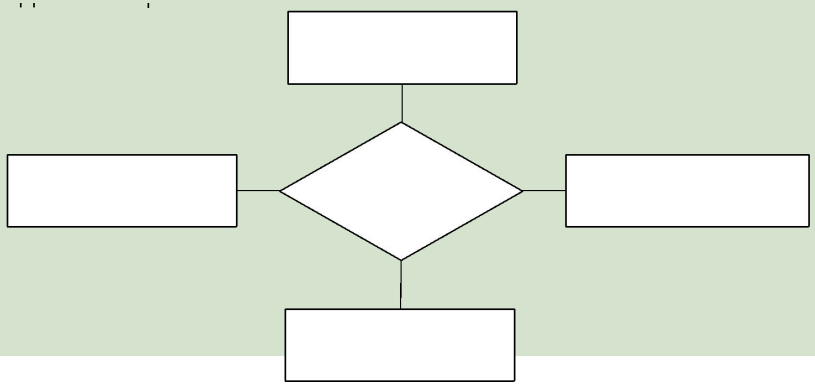


Классическая нотация IDEF1X



Степень связи

- **Степень связи** – количество сущностей, которые охвачены данной связью.

Кол-во сущ-ностей	Степень связи	Название связи	Пример (в нотации UML)	Реальный мир
2	2	Двусторонняя		В аптеке продается препарат
3	3	Трехсторонняя		Сотрудник компании регистрирует клиента в отделении
4	4	Четырехсторон-няя		Покупатель, консультируемый юристом и поддерживаемый финансовой организацией, заключает сделку

Структурные ограничения

- Ограничения являются отражением определенных требований реального мира:
  - каждый студент должен иметь паспортные данные и адрес;
  - для каждого препарата должен быть указан тип упаковки;
  - в каждой аптеке должен продаваться хотя бы один препарат;
  - в каждой группе должен учиться хотя бы один студент и др.
- Накладываются на элементы концептуальной модели (сущности, связи).


Ограничения на связи
Ограничения кратности
Ограничения кардинальности
Ограничения степени участия
Основной тип ограничений

Кратность

- **Кратность** – количество (заданное как одно значение или как диапазон значений) возможных экземпляров сущности некоторого типа, которые могут быть связаны с одним экземпляром сущности другого типа с помощью определенной связи [5].

Кратность является одним из основных ограничений, накладываемых на связь с целью отражения в ER-модели определенных требований реального мира.

Ограничения кратности отражают требования (или бизнес-правила), установленные пользователем или предприятием.

Примеры ограничений предметной области «Аптека»

- Отдельная аптека может иметь только один юридический адрес (1 : 1)
- Отдельный препарат может поставляться в различных упаковках (1 : * или 1 : M или 1 : ∞)
- В различных аптеках продаются различные препараты (* : * или M : M или ∞ : ∞)

Типы двусторонних связей

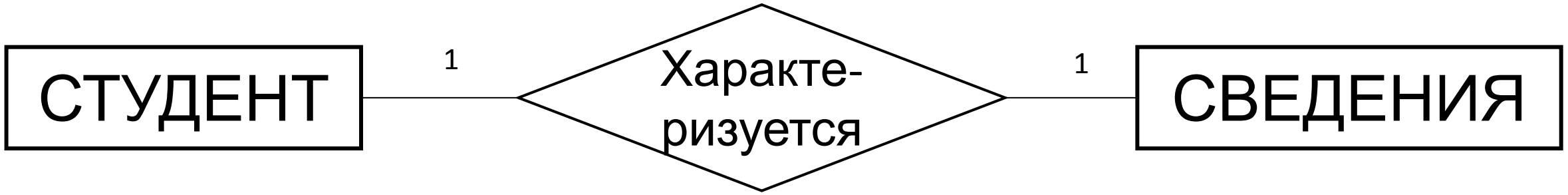
Один-к-одному (1 : 1)

Один-ко-многим (1 : *)

Многие-ко-многим (* : *)

Связь «один-к-одному»

- Связь типа ***один-к-одному*** означает, что один экземпляр первой сущности связан с одним экземпляром второй сущности. Связь один-к-одному чаще всего свидетельствует о том, что на самом деле имеется всего одну сущность, неправильно разделенная на две.

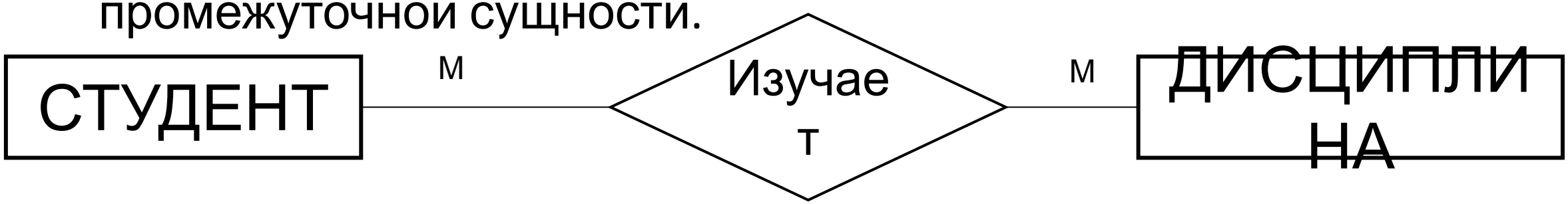


Связь «один-ко-многим»

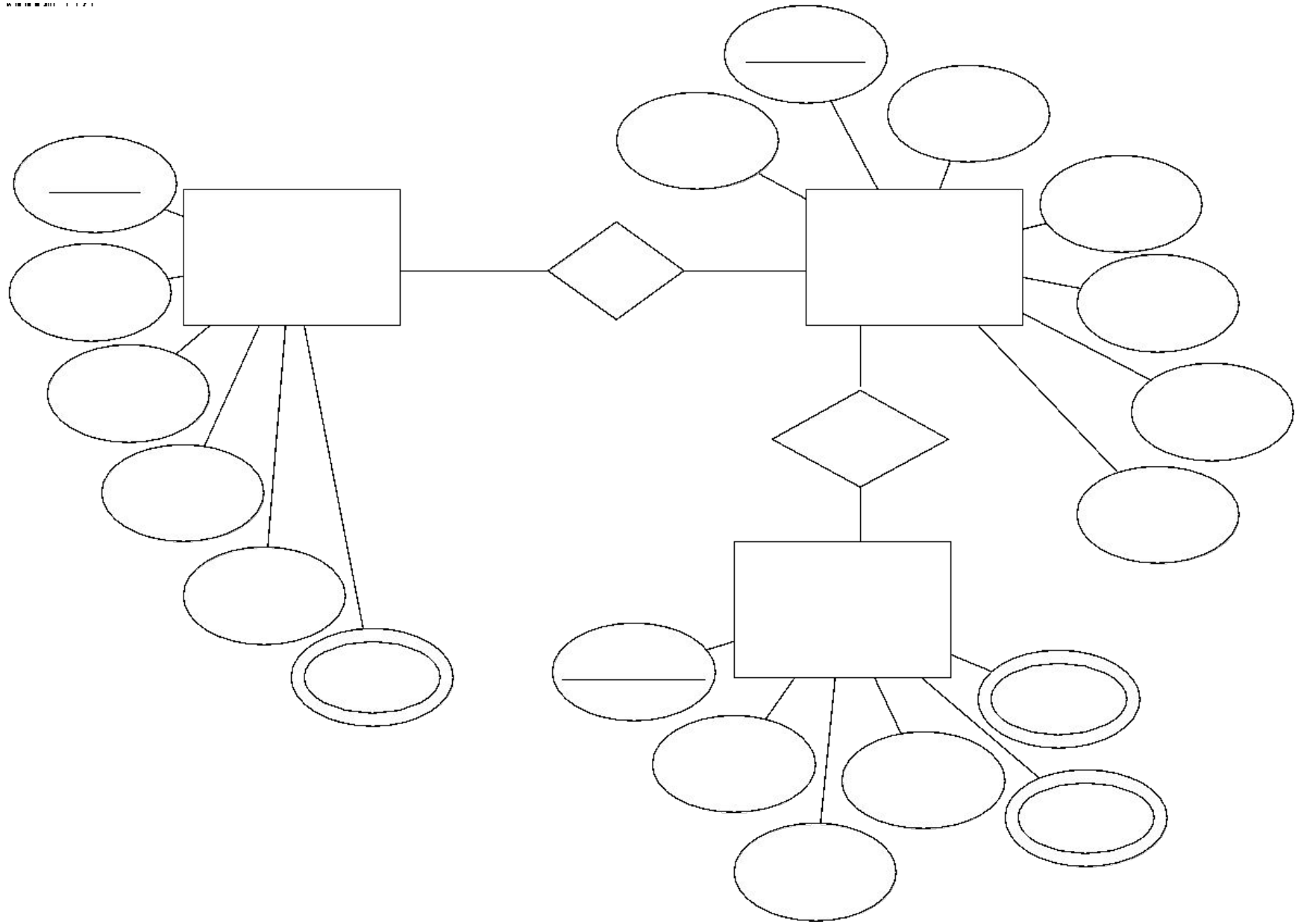
- Связь типа ***один-ко-многим*** означает, что один экземпляр первой сущности связан с несколькими экземплярами второй сущности. Это наиболее часто используемый тип связи. Левая сущность (со стороны "один") называется ***родительской***, правая (со стороны "много") – ***дочерней***.

Связь «многие-ко-многим»

- Связь типа ***много-ко-многим*** означает, что каждый экземпляр первой сущности может быть связан с несколькими экземплярами второй сущности, и каждый экземпляр второй сущности может быть связан с несколькими экземплярами первой сущности.
- Тип связи много-ко-многим является *временным* типом связи, допустимым на ранних этапах разработки модели. В дальнейшем этот тип связи должен быть заменен двумя связями типа один-ко-многим путем создания промежуточной сущности.



Пример ER-диаграммы



Этапы концептуального проектирования

- Анализ требований к базе данных
- **ER-моделирование и нормализация**
- Проверка модели данных
- Проектирование распределенной базы данных

Шаги этапа «ER-моделирование и нормализация»

1. **Определение, анализ и уточнение бизнес-правил**
2. **Выявление основных сущностей**
3. **Определение связей сущностей**
4. **Определение атрибутов, первичных и внешних ключей для каждой сущности**
5. Нормализация сущностей
6. Завершение первоначальной ER-диаграммы
7. Проверка конечным пользователем полученной модели на основе имеющихся данных и технических требований
8. Модификация ER-диаграммы (при необходимости)

ЛИТЕРАТУРА

1. Роб, П. Системы баз данных: проектирование, реализация, управление / П. Роб, К. Коронел. – 5-е изд., перераб. и доп.: Пер. с англ. – СПб.: БХВ-Петербург, 2004. – 1040с.: ил.
2. Ролланд, Ф.Д. Основные концепции баз данных / Фред. Д. Ролланд.: Пер. с англ. – М. Издательский дом «Вильямс», 2002. – 256с.
3. Сергей Кузнецов. «Базы данных. Вводный курс». URL: http://citforum.ru/database/advanced_intro
4. Дейт, К. Дж. Введение в системы баз данных, 8-е издание.: Пер. с англ. — М.: Издательский дом "Вильямс", 2018. — 1328 с.: ил.
5. Конноллн, Томас, Бегг, Карелии. Базы данных. Проектирование, реализация и сопровождение. Теория и практика. 3-е издание. : Пер. с англ. — М. : Издательский дом "Вильямс", 2003. — 1440 с.