

Проектирование баз данных

Проектирование БД представляет собой длительный и трудоемкий процесс и является скорее искусством, чем наукой. Основными ресурсами проектировщика БД служат его собственные интуиция и опыт.

Уровни проектирования:

- 1) Концептуальный.** На естественном языке с помощью диаграмм и других средств описываются объекты предметной области и их взаимосвязи.
- 2) Логический.** Выбирается модель данных (сетевая, иерархическая, реляционная), производится отображение данных концептуальной модели в логическую модель в рамках выбранной модели данных.
- 3) Физический.** Производится выбор СУБД, типов данных и методов доступа к ним, которые обеспечивает выбранная СУБД.

КОНЦЕПТУАЛЬНАЯ МОДЕЛЬ БД

Описание информационных объектов, или понятий предметной области и связей между ними и описание ограничений целостности, т.е. требований к допустимым значениям данных и к связям между ними, такая модель является инфологической (или модель «сущность-связь»).

Представление данных с помощью модели «сущность-связь»

Основными конструктивными элементами инфологических моделей являются *сущности, связи между ними и их свойства (атрибуты)*.

Сущность – любой различимый объект (объект, который мы можем отличить от другого), информацию о котором необходимо хранить в базе данных. Сущностями могут быть *люди, предметы, события*.

Экземпляр сущности – конкретный представитель сущности. Например, сущность ГОРОД, экземпляры – Москва, Киев и т.д.

Атрибут – поименованная характеристика сущности. Примерами атрибутов для сущности АВТОМОБИЛЬ являются ТИП, МАРКА, НОМЕРНОЙ ЗНАК, ЦВЕТ и т.д. для сущности СОТРУДНИК атрибутами являются ТАБЕЛЬНЫЙ_НОМЕР, ФАМИЛИЯ, ВОЗРАСТ и др.

Ключ – минимальный набор атрибутов, по значениям которых можно однозначно найти требуемый экземпляр сущности.

Например, ключом сущности АВТОМОБИЛЬ является НОМЕРНОЙ ЗНАК.

Связь – это ассоциация, установленная между несколькими сущностями.

Например,
поскольку каждый сотрудник работает в каком-либо отделе, между сущностями СОТРУДНИК и ОТДЕЛ существует связь, которую назовем «работает в»;

могут существовать и связи между сущностями одного типа, например связь «родитель – потомок» между двумя сущностями ЧЕЛОВЕК;

Связь также может иметь атрибуты.

Например, для связи «работает в» можно задать атрибут СТАЖ_РАБОТЫ_В_ОТДЕЛЕ

Язык моделирования

При построении инфологических моделей можно использовать язык **диаграммы «сущность-связь» (ER-диаграммы)**, которые **представляют собой графические схемы**.

Основные обозначения, используемые в ER-диаграммах:

Обозначение	Значение
 имя сущности	сущность
 имя атрибута	атрибут
 <u>имя атрибута</u>	ключевой атрибут
 имя связи	связь
	прямыми линиями соединяются атрибуты с сущностями и сущности со связями

Пример ER-диаграммы:



Характеристика связей

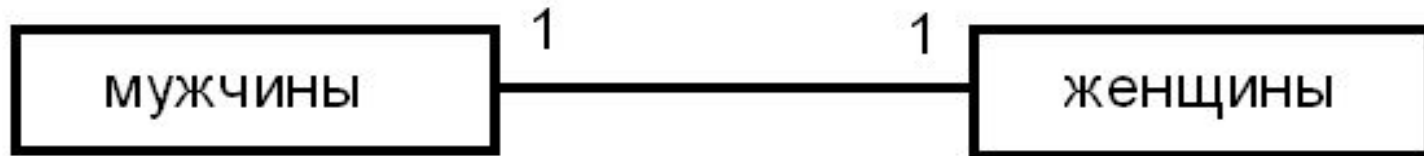
Между двумя сущностями, например, А и В возможны следующие виды связей.

1. **Связь ОДИН-К-ОДНОМУ (1:1):** в каждый момент времени каждому представителю (экземпляру) сущности А соответствует 1 или 0 представителей сущности В.

ПРИМЕР изображения связи на ER-диаграмме:



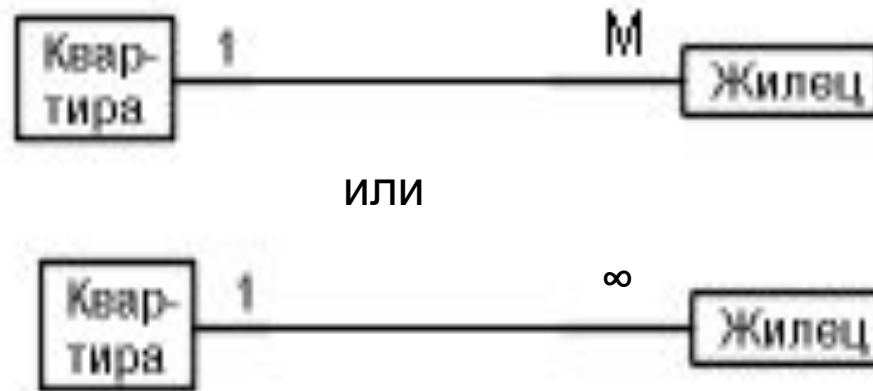
или



Мужчина женат на одной женщине (традиционный брак).

2. **Связь ОДИН-КО-МНОГИМ (1:M)**: одному представителю сущности А соответствуют 0, 1 или несколько представителей сущности В.

ПРИМЕР:

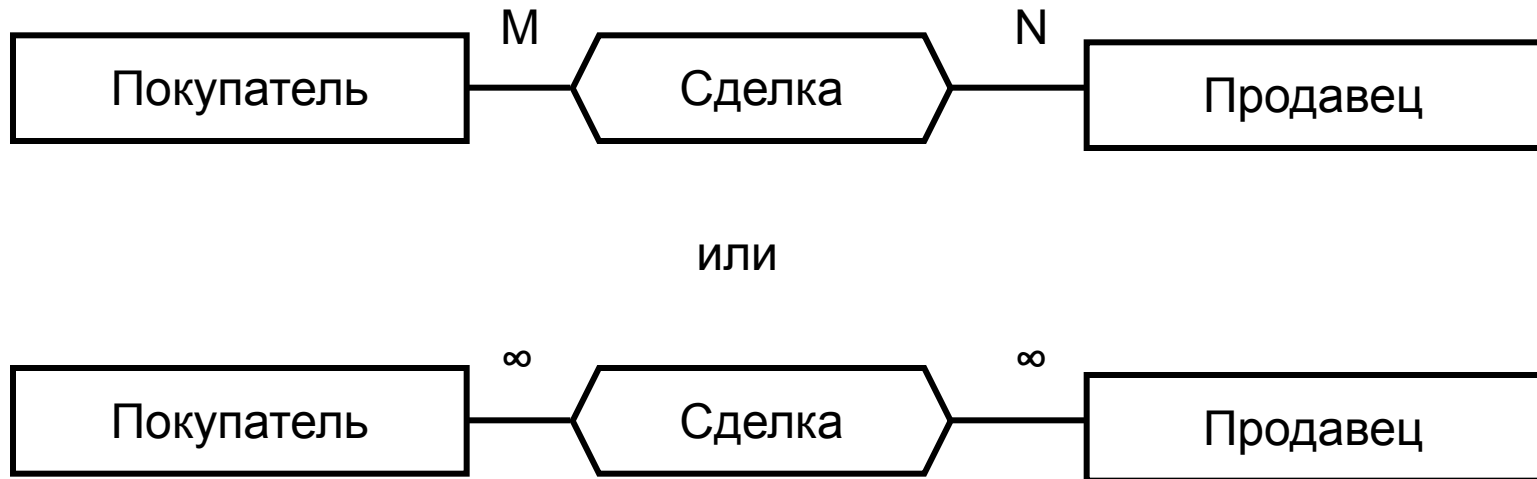


Квартира может пустовать, в ней может жить один или несколько жильцов.

Между двумя сущностями возможны связи в обоих направлениях, поэтому существует связь **МНОГИЕ-К-ОДНОМУ (M:1)**

3. Связь МНОГИЕ-КО-МНОГИМ (M:N): одному представителю сущности А соответствуют 0, 1 или несколько представителей сущности В, а одному представителю сущности В соответствуют 0, 1 или несколько представителей сущности А.

ПРИМЕР:



Покупатель может приобретать товар у нескольких продавцов, продавец может продавать товар нескольким покупателям

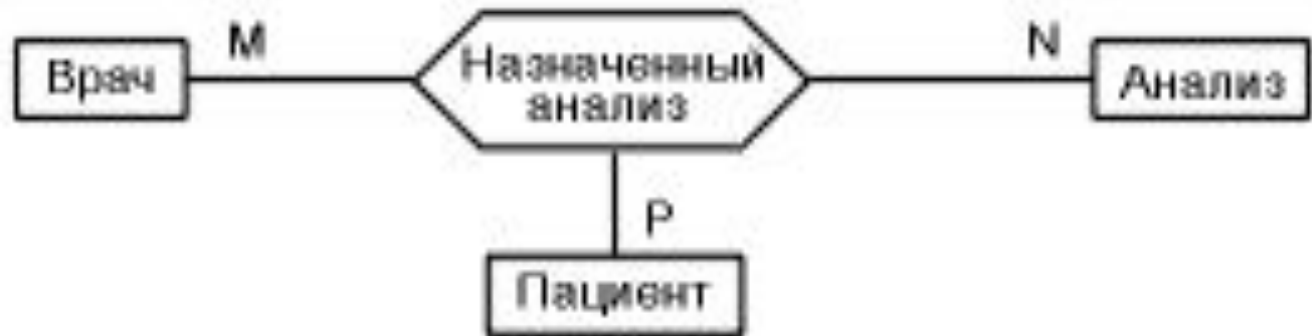
Более сложные связи:

- множество связей между одними и теми же сущностями



Пациент, имея одного лечащего врача, может иметь также несколько врачей-консультантов; врач может быть лечащим врачом нескольких пациентов и может одновременно консультировать несколько других пациентов;

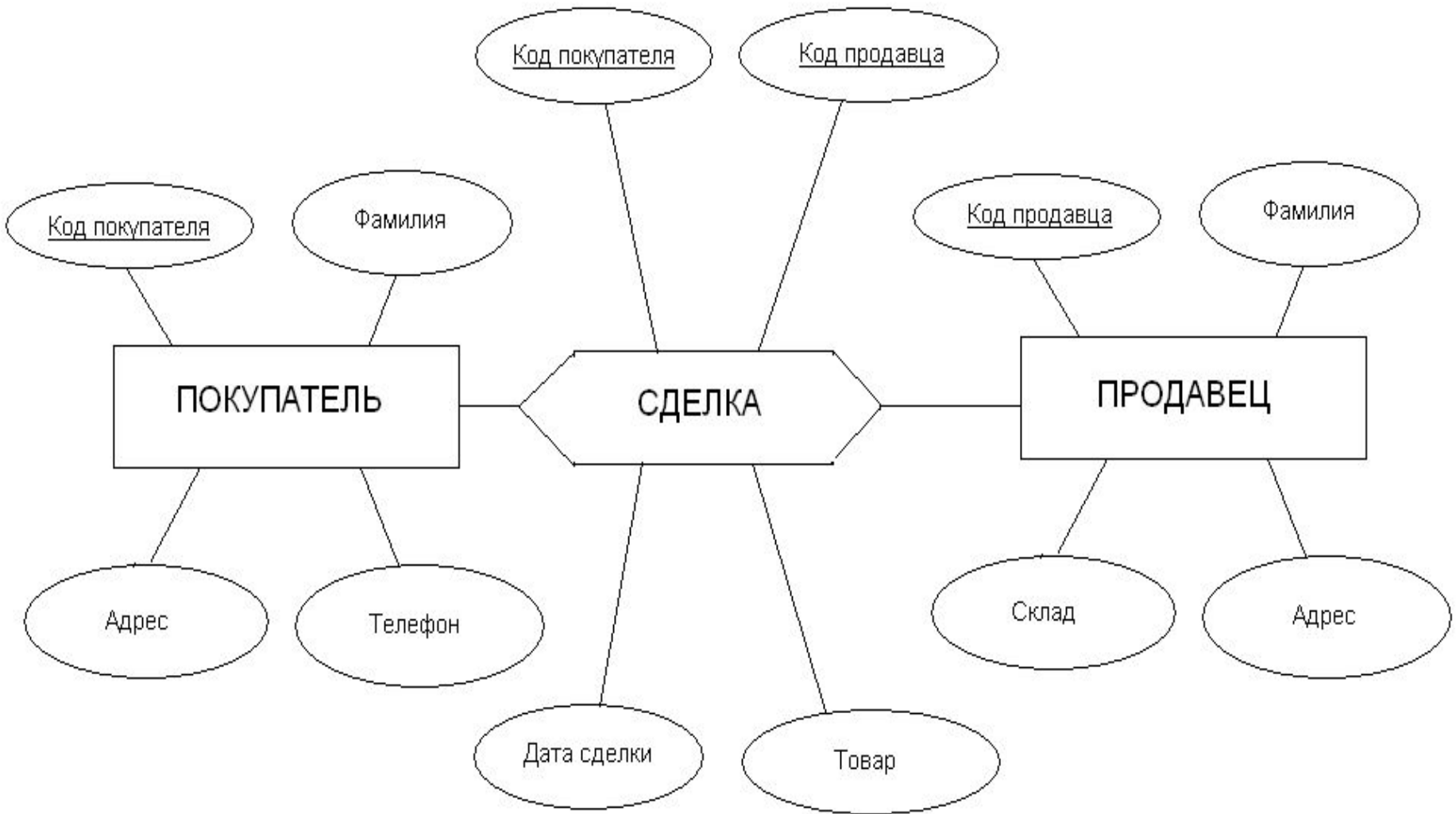
- тренирные связи



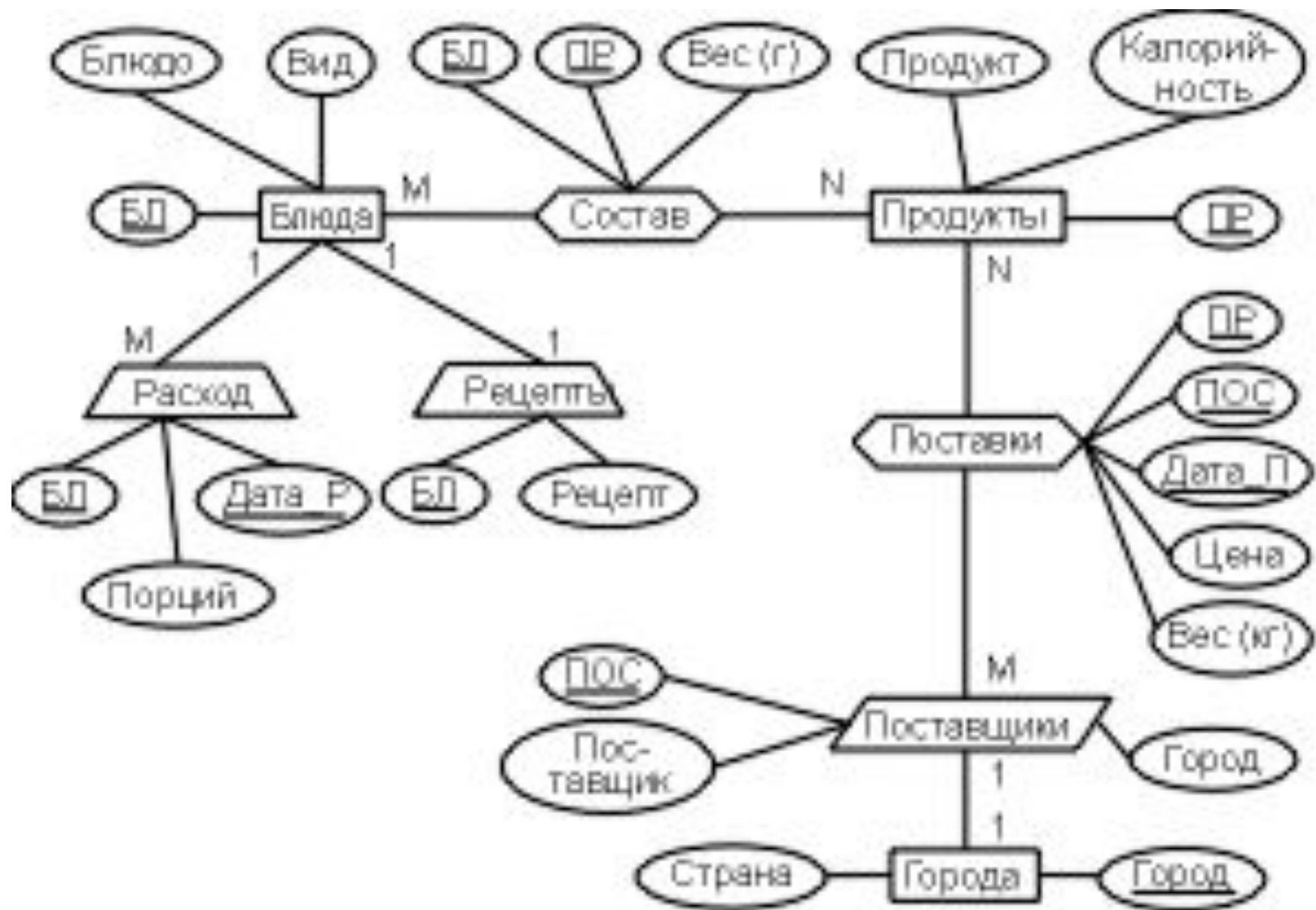
Врач может назначить несколько пациентов на несколько анализов, анализ может быть назначен несколькими врачами нескольким пациентам и пациент может быть назначен на несколько анализов несколькими врачами.

связи более высоких порядков (смысл которых иногда очень сложен)

Концептуальная модель данных на языке ER-диаграмм



Ввод лишь нескольких основных атрибутов может значительно усложнить ER-диаграмму



ЛОГИЧЕСКАЯ МОДЕЛЬ БД

Логическая модель БД строится на основе её концептуальной модели

Для построения логической модели требуется выполнить следующие действия:

1. Создать по одной таблице для каждой сущности и каждой связи.
2. Для каждой таблицы задать первичный и внешний ключи

Логическая модель не зависит от конкретной СУБД

ФИЗИЧЕСКАЯ МОДЕЛЬ БД

В физической модели содержится информация обо всех объектах БД (таблицах, индексах, процедурах и др.) и используемых типах данных.

Физическая модель зависит от конкретной СУБД.

Специфика конкретной СУБД может включать в себя ограничения на наименование объектов БД, ограничения на поддерживаемые типы и т.д.

Физическое проектирование является начальным этапом реализации БД

Пример проектирования базы данных

Разработать базу данных, в которой будут храниться личные данные студентов (фамилия, год рождения, адрес), изучаемые дисциплины, оценки, данные о преподавателях (фамилия, наименование кафедры, должность, читаемая дисциплина).

1 этап Разработка концептуальной модели БД

Первый этап проектирования заключается в описании объектов базы данных – сущностей, определении их характеристик (атрибутов) и в установлении связей между сущностями.

ФИО студента	Дата рождения	...	Дисциплина	Оценка	ФИО преподавателя	должность	...
Иванов Н.	13.01.1992	...	Информатика	4	Леонов Н.А.	доцент	...
Иванов Н.	13.01.1992	...	Математика	5	Петров О.Д.	доцент	...
Иванов Н.	13.01.1992	...	Физика	5	Николаев А.Н.	ассистент	...
Рыкова С.	24.10.1991	...	Информатика	4	Леонов Н.А.	доцент	...
Рыкова С.	24.10.1991	...	Математика	4	Петров О.Д.	доцент	...
Рыкова С.	24.10.1991	...	Физика	3	Николаев А.Н.	ассистент	...
...							

Если хранить данные в одной таблице, то в строке с фамилией студента, изучающего конкретную дисциплину, будут храниться все атрибуты преподавателя, читающего эту дисциплину. Если несколько студентов изучают данную дисциплину, то многократно будут повторяться данные о преподавателе. А если студент изучает не одну дисциплину, то многократно повторяются данные об одном и том же студенте. Это огромная избыточность данных.

А если хранить данные о студенте в одной таблице, о преподавателе – в другой, о дисциплинах - в третьей и установить связи между таблицами, то избыточность хранимых данных многократно уменьшится без ущерба для логической организации информации.

Для проектируемой базы данных можно выделить три объекта (сущности), которые не будут обладать избыточностью:

- *Студент,*
- *Преподаватель,*
- *Дисциплина.*

Зададим следующие атрибуты сущностей:

Студент	
Код студента	Число
Фамилия	Текст
Имя	Текст
Отчество	Текст
Дата рождения	Дата
Номер группы	Текст

Преподаватель	
Код преподавателя	Число
Фамилия	Текст
Имя	Текст
Отчество	Текст
Кафедра	Текст
Должность	Текст

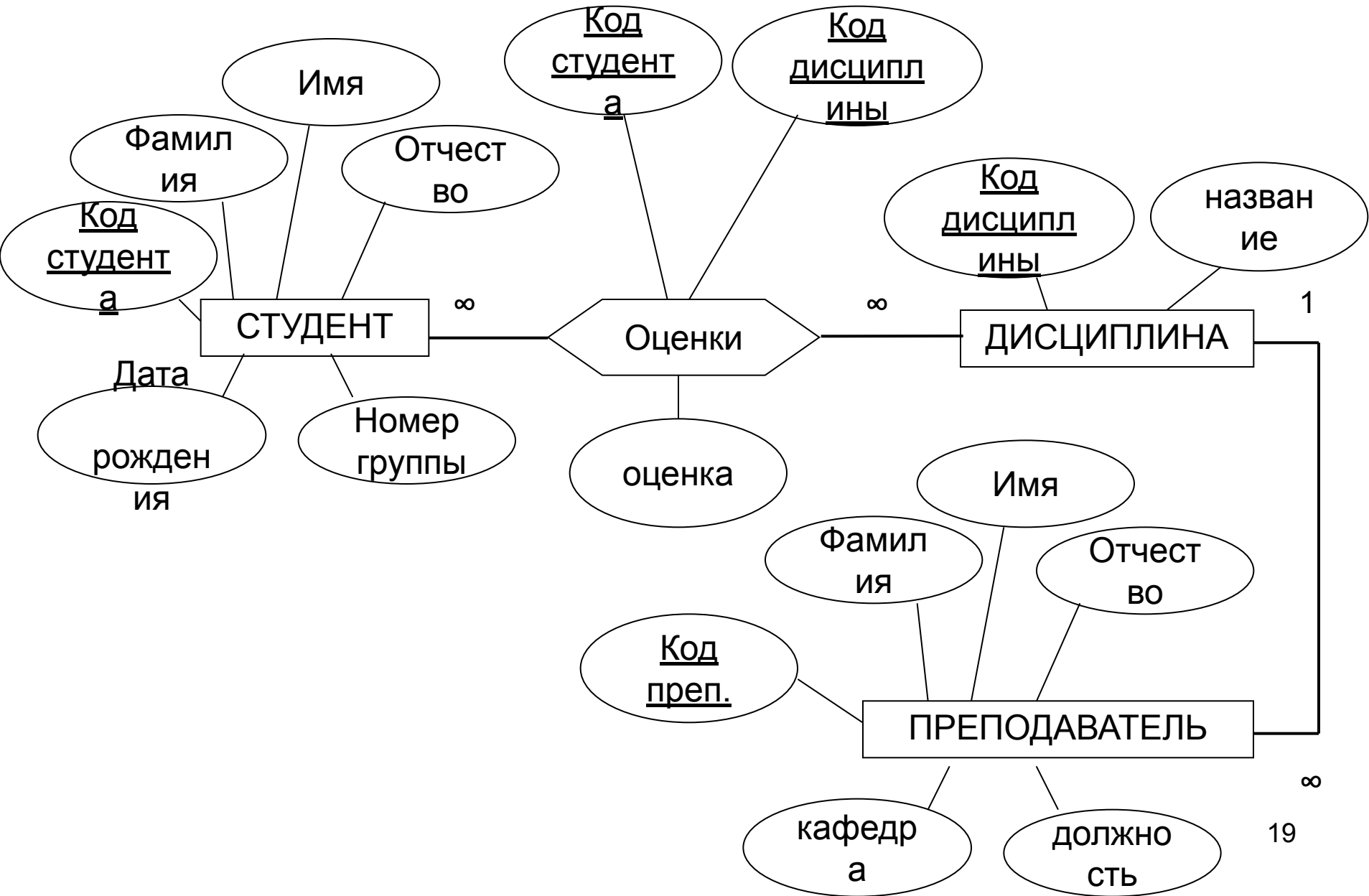
Дисциплина	
Код дисциплины	Число
Название дисциплины	Текст

Связи между сущностями

Рассмотрим связь между сущностями *Студент* и *Дисциплина*. Каждый студент изучает несколько дисциплин, и каждая дисциплина изучается множеством студентов, следовательно, связь между сущностями *Студент* и *Дисциплина* – «многие-ко-многим». Для данной связи можно задать следующие атрибуты - код студента, код дисциплины, оценка.

Рассмотрим связь между сущностями *Дисциплина* и *Преподаватель*. Одну дисциплину могут читать несколько преподавателей, но один преподаватель читает одну дисциплину, поэтому связь между сущностями *Дисциплина* и *Преподаватель* будет «один-ко-многим».

Концептуальная модель БД (ER-диаграмма)



Концептуальная модель БД (на языке инфологического моделирования)

Студент (Код студента, Фамилия, Имя, Отчество, Дата рождения, номер группы)

Дисциплина (Код дисциплины, название дисциплины)

Преподаватель (Код преподавателя, Фамилия, Имя, Отчество, кафедра, должность)

Оценки [Студент M, Дисциплина N]
(Код студента, Код преподавателя, Оценка)

2 этап Разработка логической модели БД

1. Создать по одной таблице для каждой сущности и связи, имеющей атрибуты.

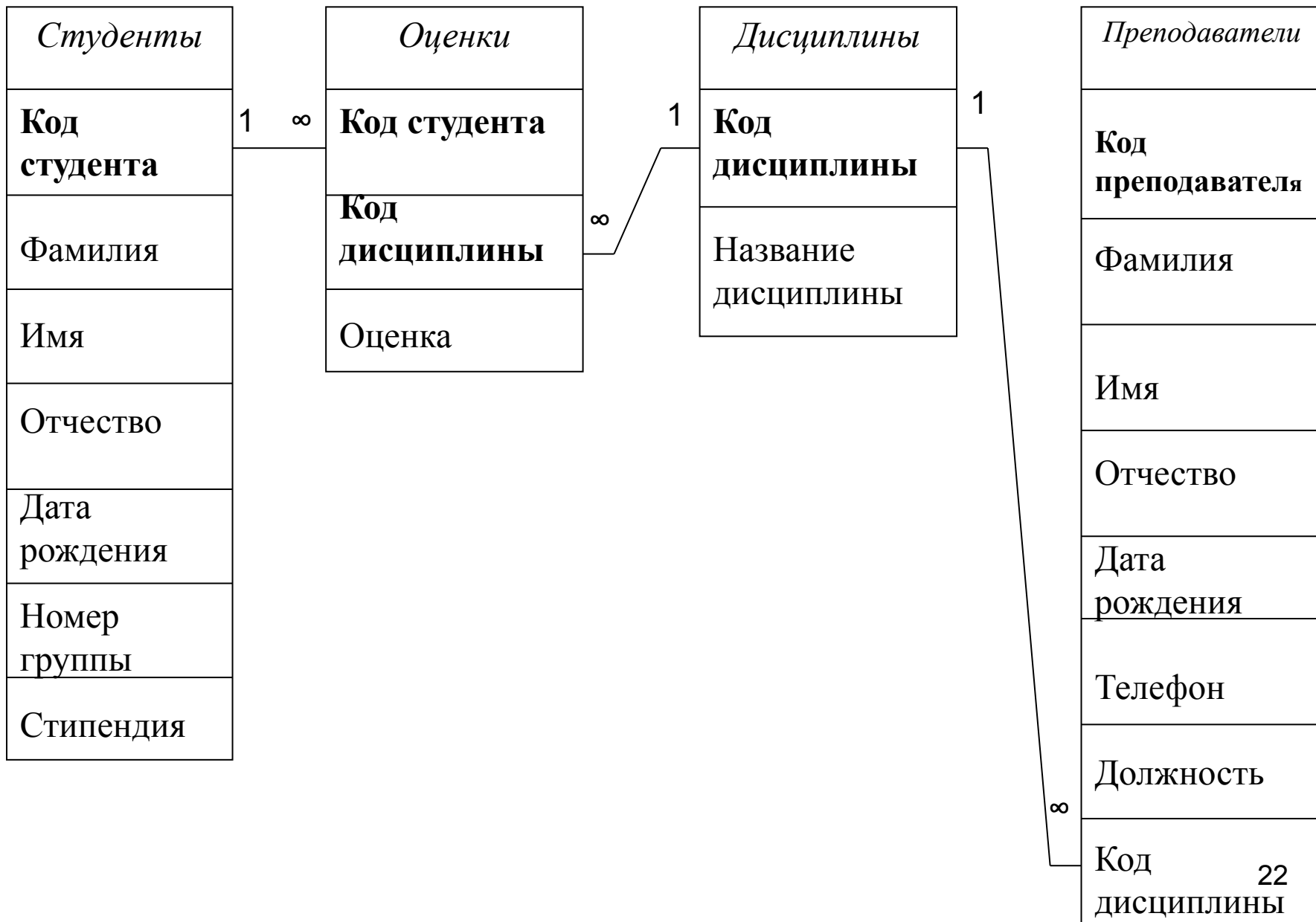
2. Для каждой сущности задать первичный и внешний ключи.

1. На основе концептуальной модели можно создать четыре таблицы: *Студенты*, *Оценки*, *Дисциплины*, *Преподаватели*.

2. Ключевые поля: в таблице *Студенты* - **Код студента**, *Дисциплины* - **Код дисциплины**, *Преподаватели* - **Код преподавателя**, *Оценки* - **Код студента**, **Код дисциплины**.

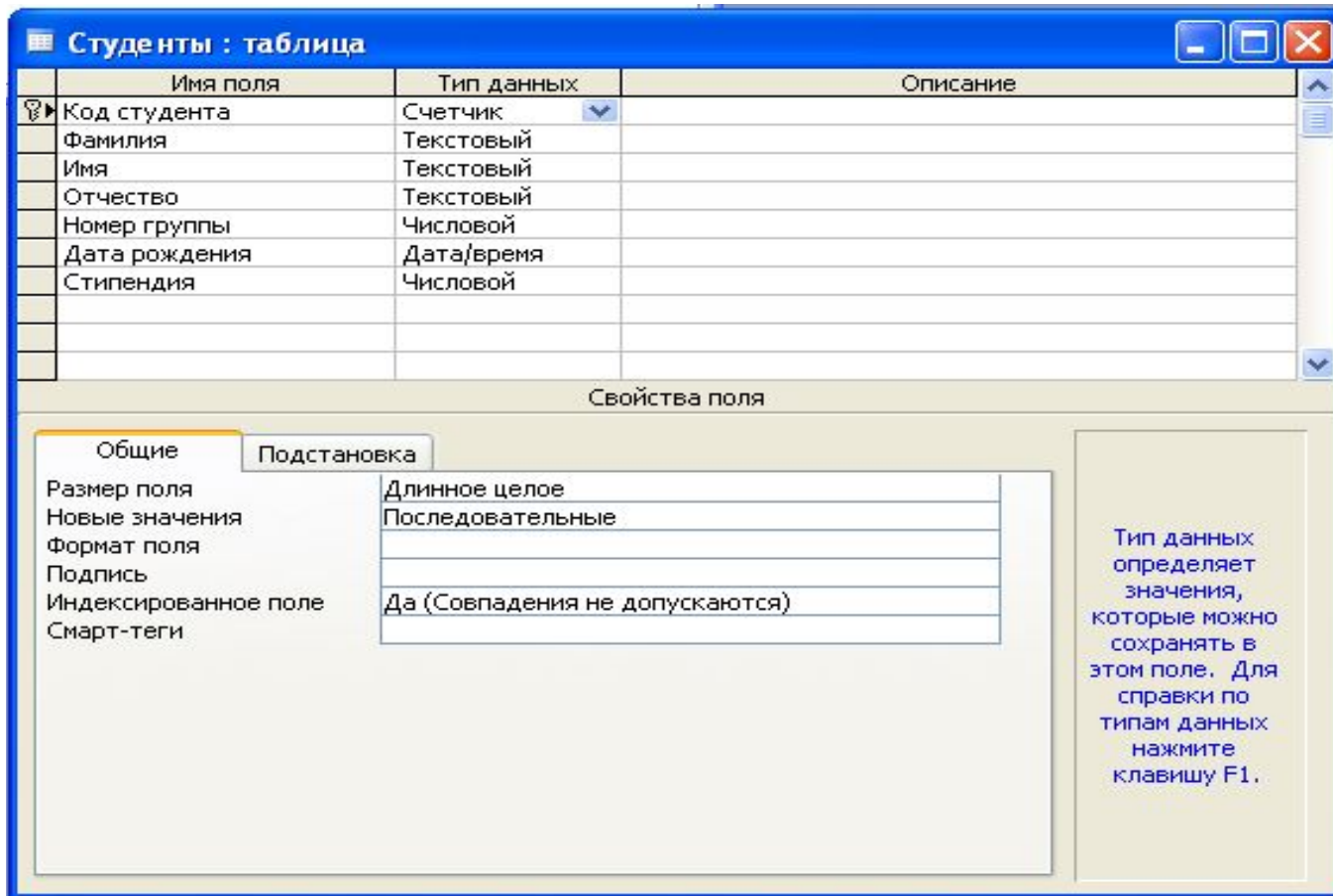
3. В таблицу *Преподаватели* введем поле **Код дисциплины**, которое будет полем **внешнего ключа** для связи с таблицей *Дисциплины*.

Логическая модель БД



3 этап Разработка физической модели БД

Этот этап представляет реализацию логической модели с помощью конкретной СУБД, например Access.



Имя поля	Тип данных	Описание
Код студента	Счетчик	
Фамилия	Текстовый	
Имя	Текстовый	
Отчество	Текстовый	
Номер группы	Числовой	
Дата рождения	Дата/время	
Стипендия	Числовой	

Свойства поля

Общие Подстановка

Размер поля: Длинное целое

Новые значения: Последовательные

Формат поля:

Подпись:

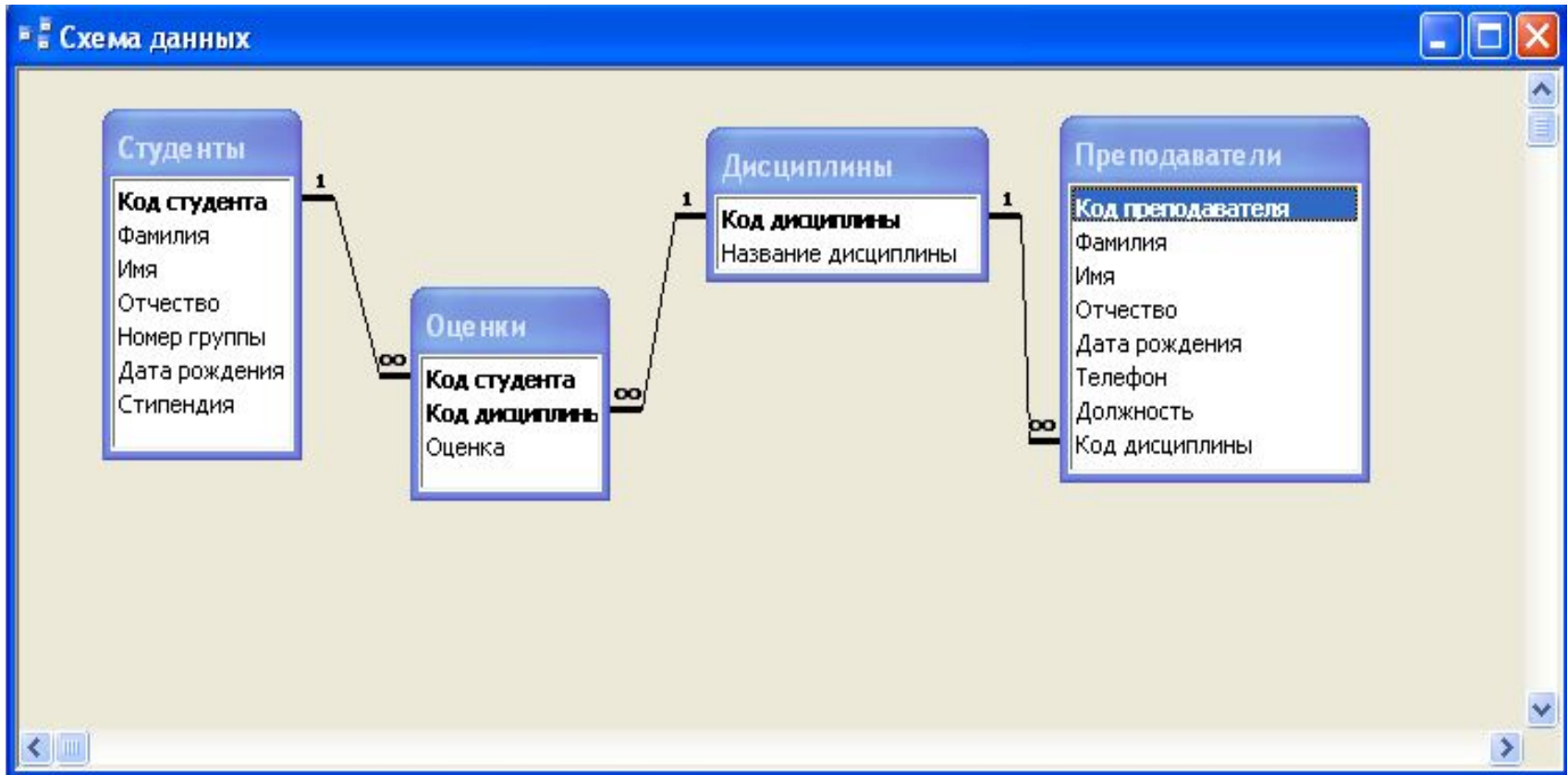
Индексированное поле: Да (Совпадения не допускаются)

Смарт-теги:

Тип данных определяет значения, которые можно сохранять в этом поле. Для справки по типам данных нажмите клавишу F1.

Создание таблиц с помощью **Конструктора**

Связывание таблиц



Ввод данных в таблицы



The screenshot shows a Windows application window titled "Студенты : таблица". The window contains a table with the following data:

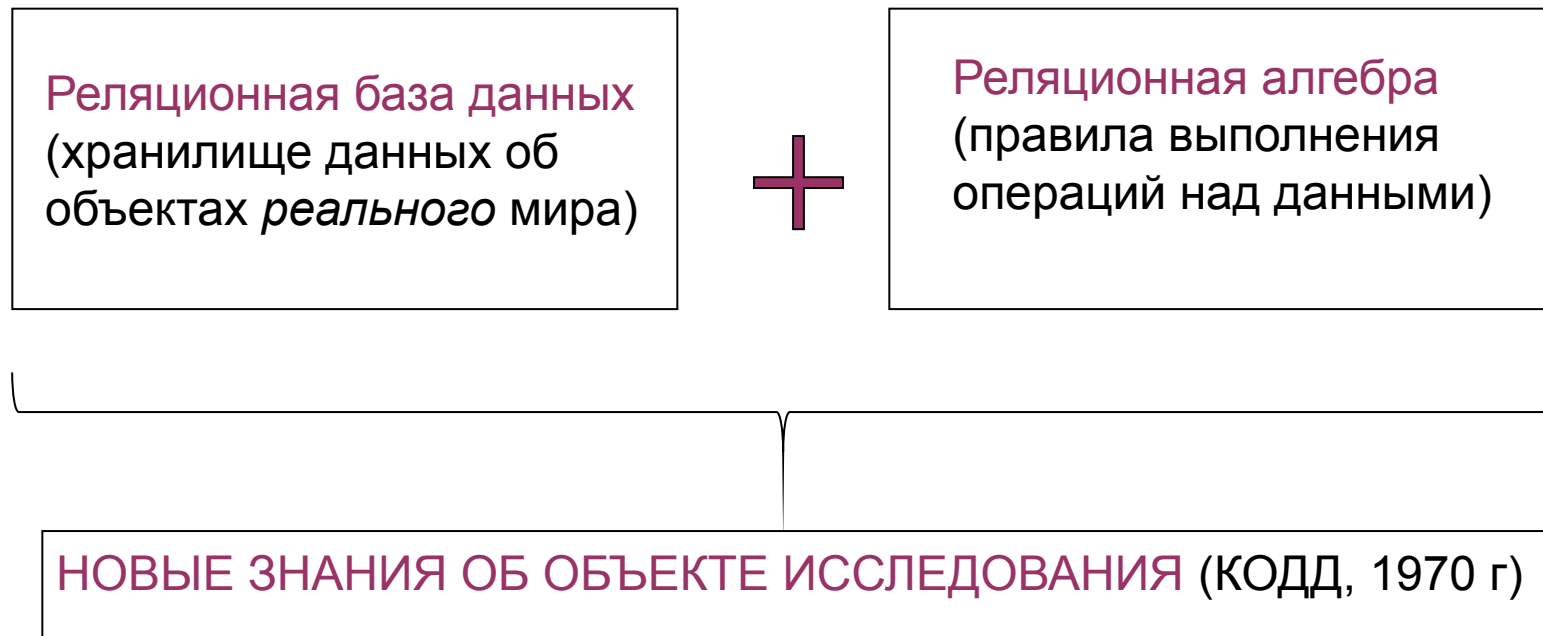
	Код студента	Фамилия	Имя	Отчество	Номер группы	Дата рождения	Стипендия
	1	Иванов	Николай	Петрович	1211	12.09.1990	600
	2	Кротов	Сергей	Борисович	1211	06.08.1991	600
	3	Сергеева	Светлана	Николаевна	1212	25.04.1990	900
	(Счетчик)				0		0

At the bottom of the window, there is a status bar with the text "Запись:" followed by navigation icons (back, forward, first, last, refresh) and a text box containing the number "3", followed by "из 3".

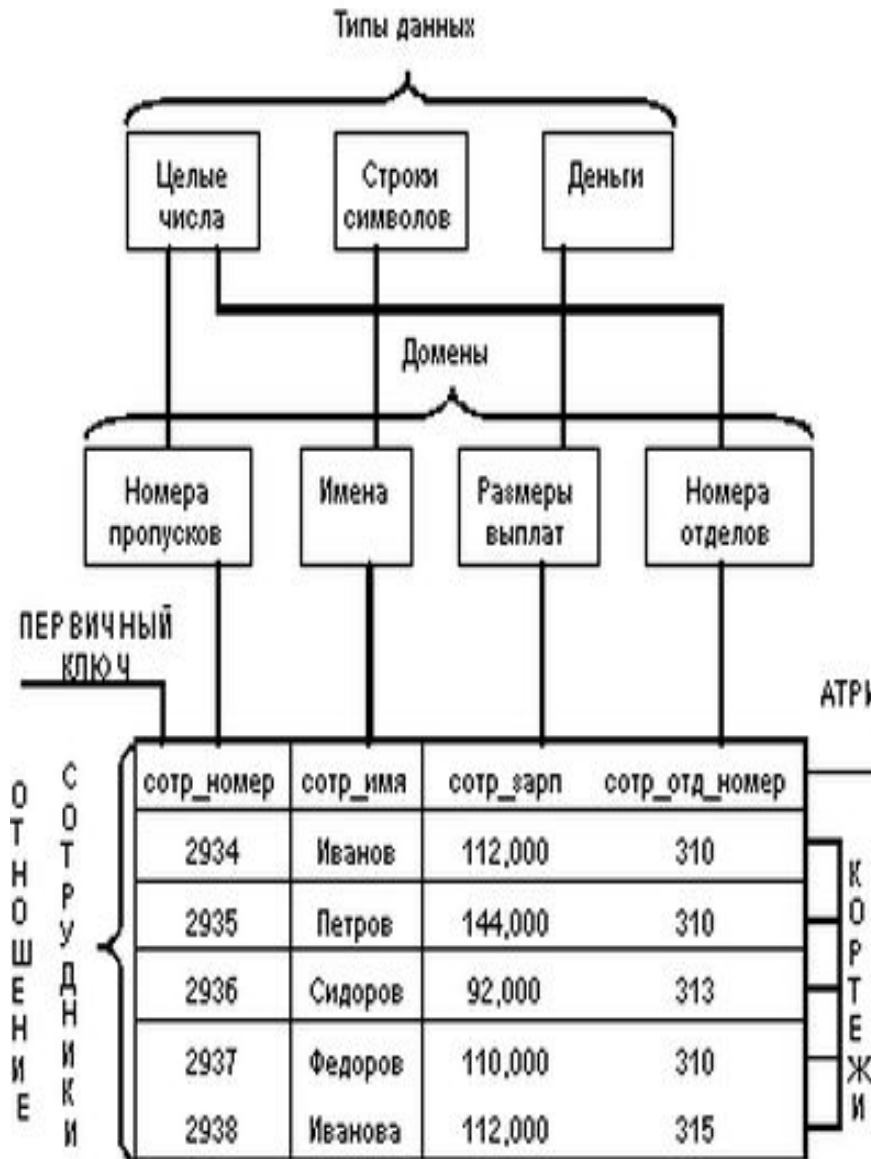
Теоретические основы реляционных баз данных

РЕЛЯЦИОННАЯ БАЗА ДАННЫХ – это набор взаимосвязанных отношений (таблиц), каждое из которых содержит конечное множество однотипных записей.

Реляционным БД свойственна наглядность и привычность представления данных (табличный вид)



ТЕРМИНОЛОГИЯ



Домен – допустимое потенциальное ограниченное подмножество значений заданного типа.

Кортеж – это элемент отношения, представляет собой набор из n -значений, по одному значению для каждого атрибута отношения (n – количество атрибутов).

Схема отношения – это именованное множество пар (имя атрибута, имя домена).

Степень, или “арность” схемы отношения,

-это мощность этого множества (число его атрибутов).

Кардинальное число – количество записей в таблице (отношении).

Ключ отношения, первичный ключ или ключевое поле – это один из атрибутов в данном отношении, который однозначно определяет (идентифицирует) каждую запись в таблице.

То есть, *значения ключевого атрибута для всех экземпляров данной таблицы различны*. Ключ отношения для удобства размещают в первом столбце.

Примеры ключа: № зачетной книжки, ИНН налогоплательщика.

Помимо простого ключа (№ лицевого счета в банке) может встречаться составной ключ (состоять из нескольких атрибутов, например, ПАСПОРТНЫЕ ДАННЫЕ: СЕРИЯ + № ПАСПОРТА) и частичный ключ (быть частью значения атрибута: № ПАСПОРТА – частичный ключ, извлекаемый из составного ПАСПОРТНЫЕ ДАННЫЕ).

Признаки отношения (необходимые):

1. Шапка таблицы состоит из конечное множества атомарных атрибутов (нет подзаголовков).
2. Каждая таблица имеет имя и первичный ключ.
3. Все атрибуты различны, а их значения однотипны.
4. Для каждого значения ключа существует запись причем только одна (ссылочная целостность).
5. Изменение значения в любом неключевом поле не должно изменять значение другого поля (функциональная независимость неключевых атрибутов)

Резюме: выполнение 1)- 5) позволяет считать таблицу отношением, то есть объектом реляционных операций.

Является ли данная таблица отношением?

	Код сотруд	ФИО	Должность	Дата найма	Дата рождения	Домашний телефон	Оклад
▶ +	2	Иванов	турагент	22.09.2007	24.05.1976	(8-3822)67-65-98	6900
+	3	Никитин	турагент	29.03.2006	25.03.1977	(8-3822)87-64-90	6900
+	4	Уваров	турагент	13.04.2007	13.04.1971	(8-3823)98-54-83	6900
+	5	Павлов	турагент	23.01.1994	12.01.1978	(8-3833)66-77-44	6900
+	6	Свиридов	бухгалтер	03.08.1993	01.09.1975	(8-3832)88-66-00	9293
+	7	Симонов	экономист	27.01.2007	07.09.1979	(8-3822)77-24-99	9171
+	9	Пиановский	турагент	22.12.1996	24.05.1976	(8-3822)67-65-98	6900
+	10	Горов	турагент	22.12.1996	24.05.1976	(8-3822)67-65-98	6900
+	11	Говоров	турагент	13.04.2007	13.01.1975	(8-3822)87-65-43	6900
+	12	Поморовская	турагент	23.01.1990	12.01.1978	(8-3833)66-77-44	6900
+	13	Свиридовский	бухгалтер	03.08.2007	01.09.1975	(8-3832)88-36-00	9293
+	14	Самойлов	экономист	14.10.2007	07.09.1979	(8-3822)77-28-99	9171
+	15	Логинов	турагент	22.12.1996	24.05.1976	(8-3822)67-65-98	6900
+	16	Баранов	турагент	22.12.1996	24.05.1976	(8-3822)67-65-98	6900
*	(Счетчик)						6000

Запись: 1 из 14

СВОЙСТВА ОТНОШЕНИЙ

1. однородность столбцов (элементы столбца принимают значения на одном домене)
2. отсутствие кортежей-дубликатов
3. отсутствие упорядоченности кортежей
4. отсутствие упорядоченности атрибутов
5. атомарность значений атрибутов

ВИДЫ ОТНОШЕНИЙ

По содержимому:

Объектное: хранит данные об объекте моделирования.

Пример объектного отношения – таблица СТУДЕНТ.

Связное: хранит первичные ключи двух или более объектных отношений, связывая их между собой в пределах одной базы данных. Ключи в связных отношениях называются *внешними*, поскольку они являются первичными ключами других отношений.

По физическому сохранению в памяти компьютера:

Хранимое: отношение, которое физически поддерживается в памяти (объектное, связное, запрос, ...).

Нехранимое (вычисляемое): отношение, полученное в ходе промежуточных вычислений и ненужное для последующей работы. Таким путём экономится машинная память и удешевляется программный продукт.

Пример связного отношения

студент

фио	курс	специальность
Иванов	2	химия
Петров	4	бух. учёт
Титов	3	физика

предмет

название	число семестров
алгебра	4
история	3
информатика	4
иностр. язык	2

изучает

студент	предмет
Иванов	алгебра
Иванов	информатика
Петров	иностр. язык
Титов	история

Для двух объектных отношений
СТУДЕНТ(ФИО, КУРС, СПЕЦИАЛЬНОСТЬ)
ПРЕДМЕТ(НАЗВАНИЕ, ЧИСЛО СЕМЕСТРОВ)
связным отношением может быть
ИЗУЧАЕТ(СТУДЕНТ, ПРЕДМЕТ), то есть оно
содержит ключи обоих объектных отношений.