ЗАНЯТИЕ №

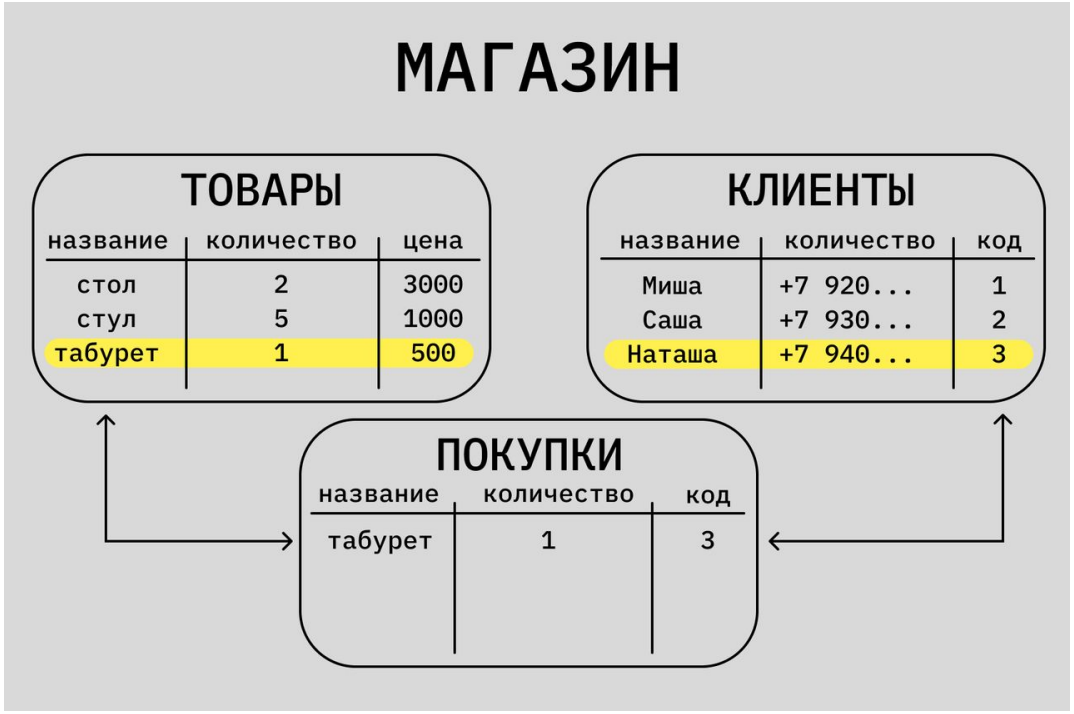
Логическое (даталогическое) проектирование БД

Проектирование БД:	
Стадии проектирования БД:	**Этапы проектирования БД:**
1. Концептуальное (инфологическое) проектирование БД.	1.1 Анализ предметной области (ПрО): - описание объектов и понятий ПрО, - связей между ними 1.2 Определение требований к БД и функций БД. (какие данные необходимо хранить в БД, запросы пользователей к БД и т.п.) 1.3 Проектирование концептуальной (инфологической) модели данных (ER – диаграммы, модели предметной области)
2. Логическое (даталогическое) проектирование БД.	2.1 Выбор СУБД 2.2 Проектирование логической (даталогической) модели (схемы данных БД).
3. Физическое проектирование БД.	

Реляционные БД

Реляционная база данных — это составленная по реляционной модели база данных, в которой данные, занесенные в таблицы, имеют изначально заданные отношения. Сами таблицы в такой базе данных также соотносятся друг с другом строго определенным образом.

В таблицах есть поле для внешнего ключа со ссылками на другие таблицы



2.2 Проектирование схемы данных БД с учетом выбранной СУБД.

Получение реляционной модели данных из ER-диаграммы:

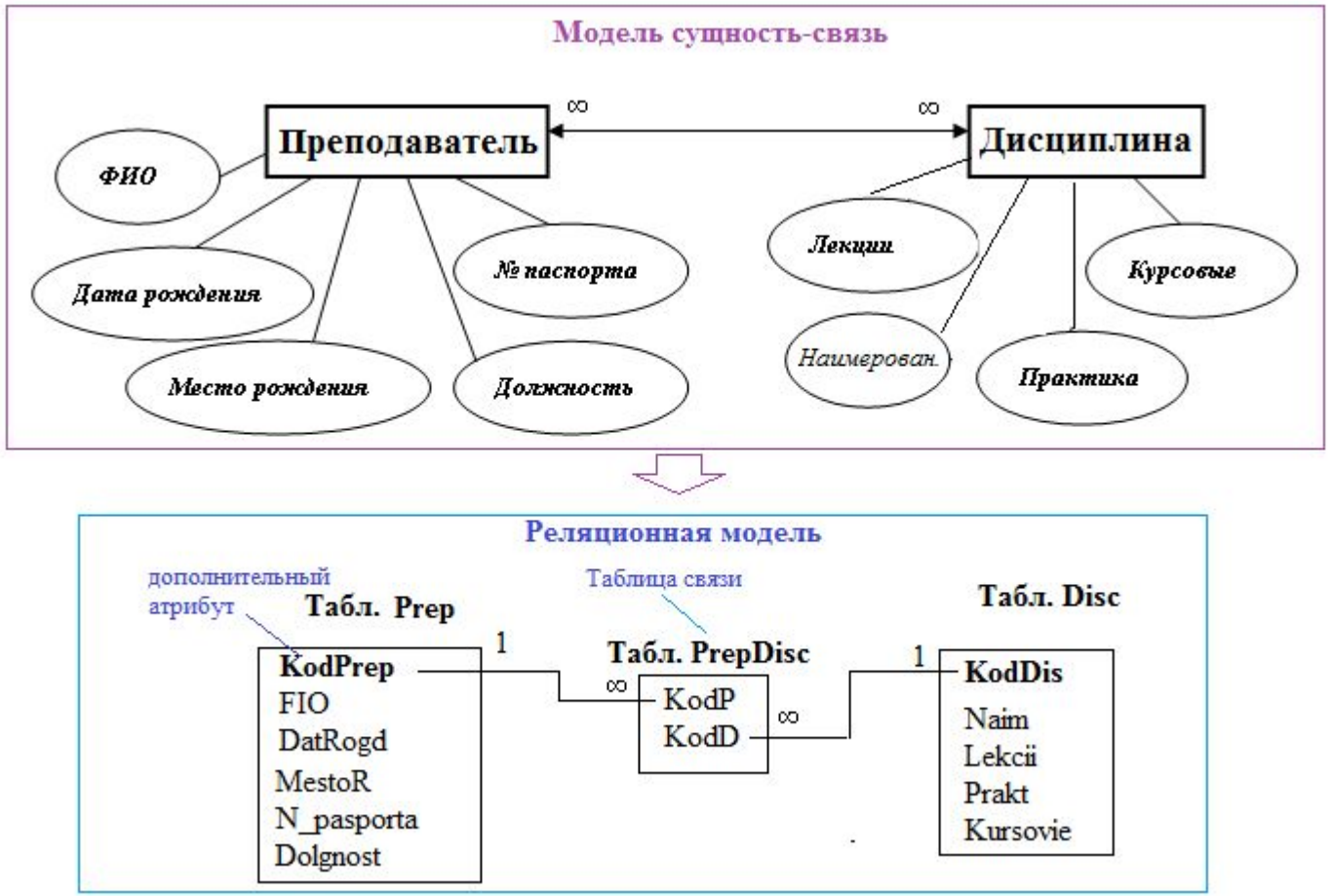


Рис.1. Преобразование модели сущность-связь со связью M:N в реляционную модель

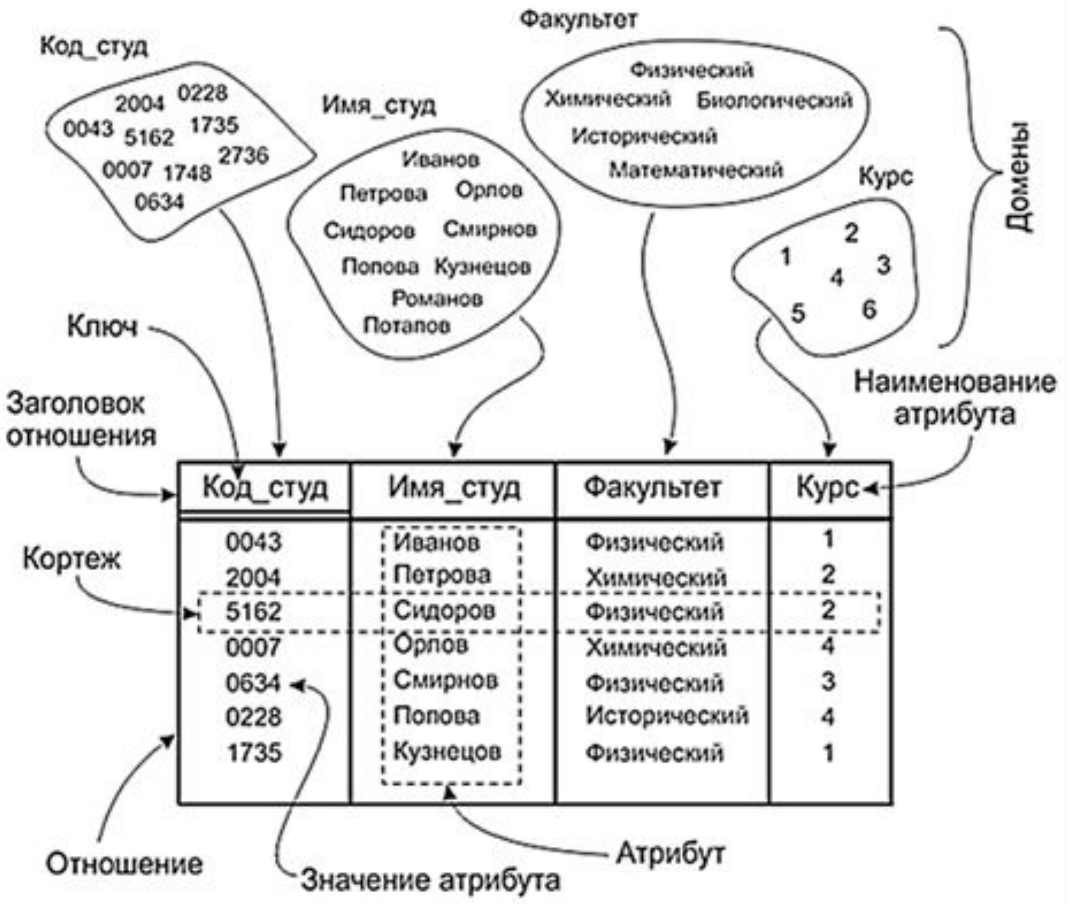
Основными понятиями, с помощью которых определяется реляционная модель, являются следующие: домен, отношение, кортеж, кардинальность, атрибуты, степень, первичный ключ.

Отношение – таблица.
Кортеж – строка.
Кардинальность – количество строк в таблице.
Атрибут – поле, столбец таблицы.
Степень отношения – количество полей, столбцов.
Первичный ключ – это столбец или некоторое подмножество столбцов, которые уникально, т.е. единственным образом определяют строки.
Внешний ключ – это столбец или подмножество одной таблицы, который может служить в качестве первичного ключа для другой таблицы.

Домен – это совокупность значений, из которой берутся значения соответствующих атрибутов определенного отношения. С точки зрения программирования, домен – это тип данных, определяемый системой (стандартный) или пользователем.

Реляционная модель данных (РМД) предъявляет к таблицам следующие требования:

1.данные в ячейках таблицы должны быть структурно неделимыми;
2.данные в одном столбце должны быть одного типа;
3.каждый столбец должен быть уникальным (недопустимо дублирование столбцов);
4.столбцы размещаются в произвольном порядке;
5.строки размещаются в таблице также в произвольном порядке;
6.столбцы имеют уникальные наименования.



Получение реляционной схемы из ER-диаграммы.

1. Каждая простая сущность превращается в таблицу (отношение). Имя сущности становится именем таблицы.

2. Каждый атрибут становится возможным столбцом с тем же именем. Столбцы, соответствующие необязательным атрибутам, могут содержать неопределенные значения; столбцы, соответствующие обязательным атрибутам – не могут. Если атрибут является множественным, то для него строится отдельное отношение.

3. Компоненты уникального идентификатора сущности превращаются в первичный ключ. Если имеется несколько возможных уникальных идентификаторов, выбирается наиболее используемый. Если в состав уникального идентификатора входят связи, то к числу столбцов первичного ключа добавляется копия уникального идентификатора сущности, находящейся на дальнем конце связи (этот процесс может продолжаться рекурсивно). Для именования этих столбцов используются имена концов связей и/или имена сущностей.

4. Связи «многие к одному» и «один к одному» становятся внешними ключами. Т.е. создается копия уникального идентификатора с конца связи «один», и соответствующие столбцы составляют внешний ключ.

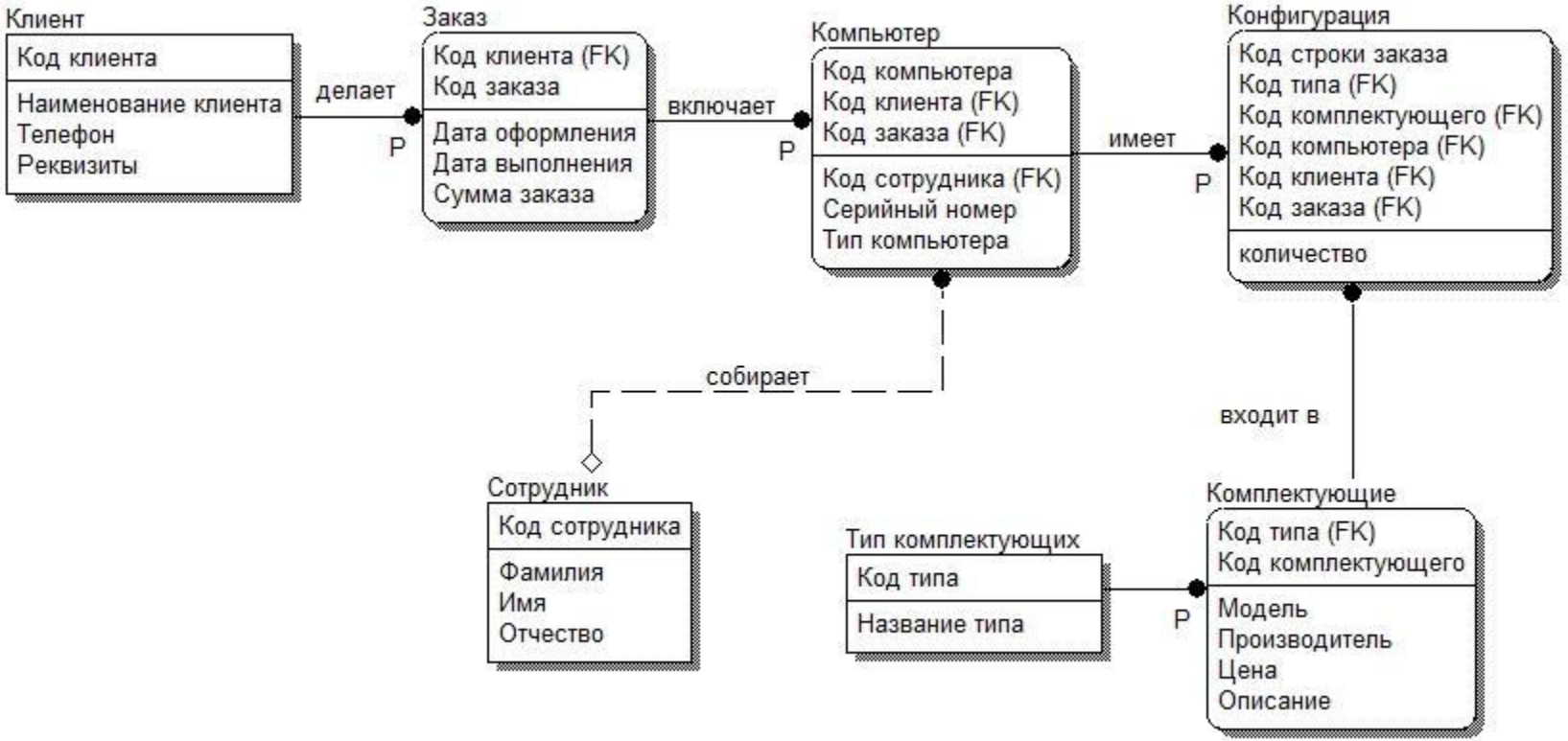
5. Индексы создаются для первичного ключа (уникальный индекс), а также внешних ключей и тех атрибутов, которые будут часто использоваться в запросах.

6. Если в концептуальной схеме присутствуют подтипы, то возможны два варианта. Все подтипы хранятся в одной таблице, которая создается для самого внешнего супертипа, а для подтипов создаются представления. В таблицу добавляется по крайней мере один столбец, содержащий код типа, и он становится частью первичного ключа.
Во втором случае для каждого подтипа создается отдельная таблица (для более нижних – представления) и для каждого подтипа первого уровня супертип воссоздается с помощью представления UNION (из всех таблиц подтипов выбираются общие столбцы – столбцы супертипа).

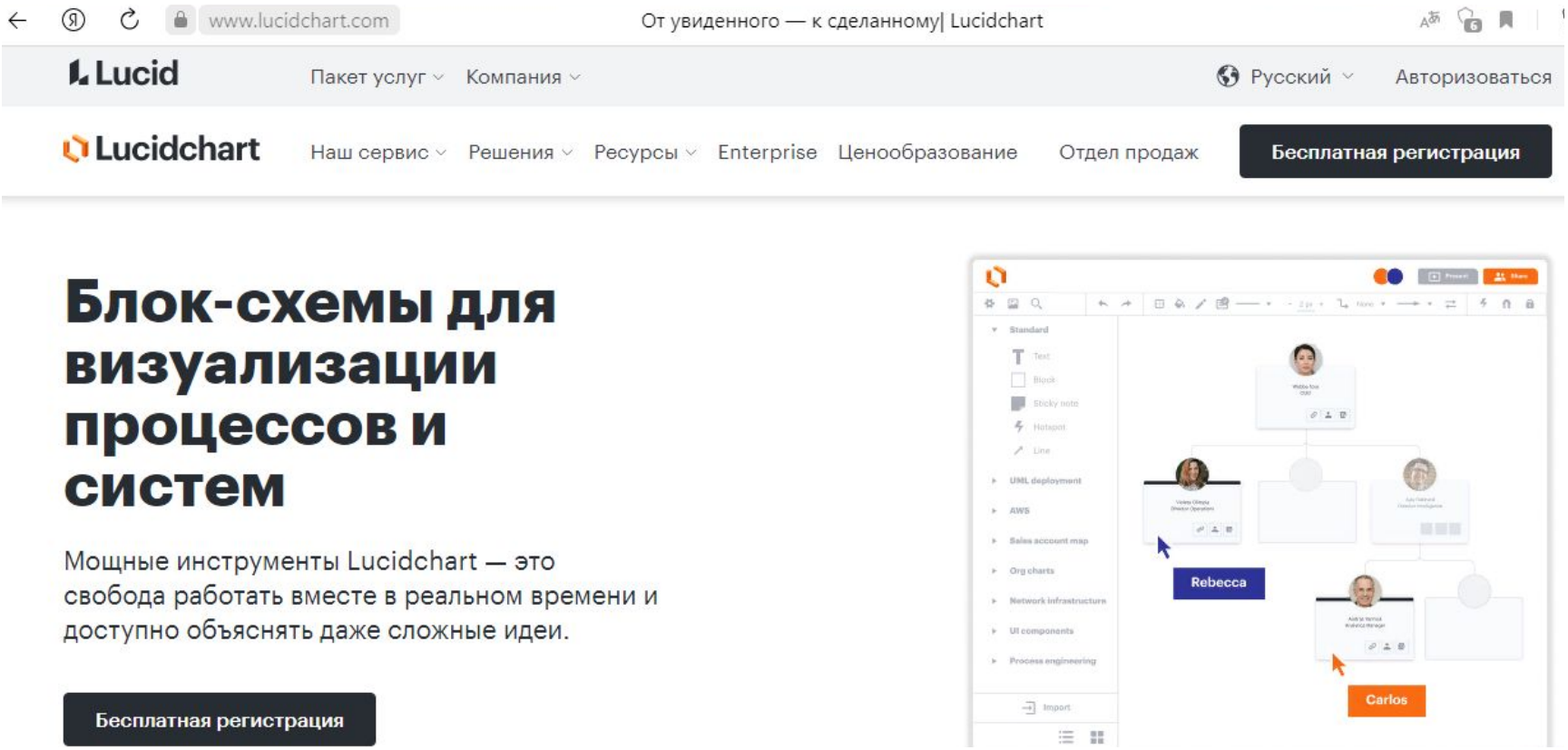
7. Если остающиеся внешние ключи все принадлежат одному домену, т.е. имеют общий формат, то создаются два столбца: идентификатор связи и идентификатор сущности. Столбец идентификатора связи используется для различных связей. Столбец идентификатора сущности используется для хранения значений уникального идентификатора сущности на дальнем конце соответствующей связи.

Если результирующие внешние ключи не относятся к одному домену, то для каждой связи, покрываемой дугой исключения, создаются явные столбцы внешних ключей.

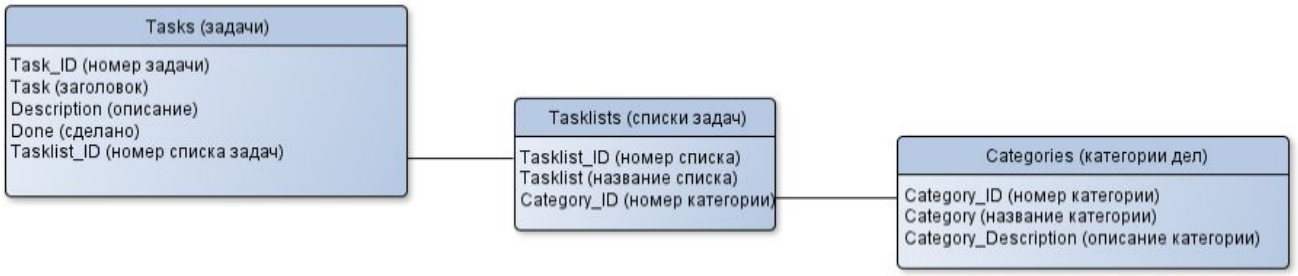
Логическая модель базы данных



Lucidchart – это программа для построения диаграмм.



https://www.lucidchart.com/pages/ru



Словарь данных:

Таблица **Tasks** (задачи) включает поля

Поле	Описание	Тип поля
Task_ID	Порядковый номер задачи (каждой новой задаче автоматически присваивается уникальный номер)	Целое число
Task	Краткое описание (заголовок) задачи.	Текст
Description	Подробное описание задачи. Может быть пустым (для описания большинства простых задач достаточно заголовка).	Текст
Done	Признак «Сделано» – Нет, если задача ещё не выполнена, Да, если выполнена.	Целое число, представляющее логическое значение (1 - да / 0 - нет)
Tasklist_ID	Порядковый номер списка, в который входит задача (см. следующую таблицу)	Целое число

Таблица **Tasklists** (списки задач) включает поля ……

Таблица **Categories** (категории) включает поля .....

Нормальная форма — свойство отношения в реляционной модели данных, характеризующее его с точки зрения избыточности, потенциально приводящей к логически ошибочным результатам выборки или изменения данных.
Нормальная форма определяется как совокупность требований, которым должно удовлетворять отношение.
Процесс преобразования отношений базы данных к виду, отвечающему нормальным формам, называется **нормализацией**.

По правилам нормализации есть семь **нормальных форм баз данных**: ● первая, ● вторая, ● третья, ● нормальная форма Бойса-Кодда, ● четвёртая, ● пятая, ● шестая.
Приводить данные к нормальным формам можно только последовательно.
То есть в базе данных второй нормальной формы данные по умолчанию уже должны быть нормализованы по правилам первой нормальной формы и так далее.

В некоторых случаях попытка нормализовать данные до «идеального» состояния может привести к созданию множества таблиц, ключей и связей. Это усложнит работу с базой и снизит производительность СУБД. Поэтому обычно данные нормализуют.

Первая нормальная форма
В базе данных не должно быть дубликатов и составных данных.

Покупатель
Александр Александрович Сушков
Егор Сергеевич Кузнецов
Иван Иванович Иванов
Сергей Александрович Петров

Имя	Отчество	Фамилия
Александр	Александрович	Сушков
Егор	Сергеевич	Кузнецов
Иван	Иванович	Иванов
Сергей	Александрович	Петров

Слева данные о фамилии, имени и отчестве покупателей записаны в одно поле. Справа эти данные приведены к первой нормальной форме — каждый элемент записан в отдельное поле

Элементы составных данных лучше разнести по разным полям, иначе в процессе работы с данными могут появиться ошибки и аномалии.

Вторая нормальная форма

Если упростить: у каждой записи в базе данных должен быть первичный ключ. Первичный ключ — это элемент записи, который не повторяется в других записях.

Допустим, 10 декабря покупатель Егор Кузнецов купил цельнозерновой хлеб за 75 рублей в сетевом магазине продуктов города Москвы. Запись о его покупке появилась в базе данных. Нельзя исключать, что другой Егор Кузнецов в этот день купит такой же товар в другом магазине сети. Запись о покупке тоже появится в базе.

Номер чека	Имя	Фамилия	Товар	Стоимость
1283	Егор	Кузнецов	Цельнозерновой хлеб	75,00
4569	Егор	Кузнецов	Цельнозерновой хлеб	75,00

Чтобы записи не перепутались, можно добавить к ним идентификатор покупки, например номер чека. Идентификатор покупки — это первичный ключ

Третья нормальная форма
В записи не должно быть столбцов с неключевыми значениями, которые зависят от других неключевых значений.

Личный № сотрудника	Должность	Отдел	Руководитель отдела
120	Программист	Отдел разработки	Иванов И.И.
121	Инженер данных	Отдел аналитики	Кузнецов Е.А.

Личный номер сотрудника — это первичный ключ. Данные во втором и третьем столбце напрямую зависят от первичного ключа. Но между личным номером сотрудника и руководителем отдела только косвенная или транзитивная связь. Её в базе данных третьей нормальной формы быть не должно

Данные о руководителях отделов нужно вынести в другую таблицу. Тогда в основной таблице не будет транзитивных связей, и она будет соответствовать третьей нормальной форме.

Задание:

Изучить материал занятия и выполнить:

1. Изучить содержание этапа проектирования БД - логическое (даталогическое) проектирование БД.
2. Разработать логическую модель базы данных используя CASE-средство Lucidchart.
3. Разработать словарь данных в формате документа Word.
4. Провести нормализацию БД до третьей нормальной формы
5. Представить результаты работы на этапе проектирования БД - логическое (даталогическое) проектирование БД.