Федеральное государственное образовательное бюджетное

учреждение высшего образования

**«Финансовый университет при Правительстве Российской Федерации»**

**(Финансовый университет)**

Колледж информатики и программирования

**ОТЧЕТ**

**Лекция**

**По теме**

«Проектирование баз данных»

**Дисциплина /Профессиональный модуль:** Проектирование и дизайн информационных систем

**Студент: Карабут Борис**

**Группа: 3ИСИП-521**

**Преподаватель:**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Абзалимов Р.Р./

**Дата выполнения:**

20.11.2023

Москва

2023

**4.1 Задачи и основные этапы проектирования баз данных**

Процесс проектирования баз данных длительный, он требует обсуждений с заказчиком, со специалистами в предметной области. При разработке серьезных корпоративных информационных систем проект базы данных является тем фундаментом, на котором строится вся система в целом. Проектирование баз данных состоит в построении комплекса взаимосвязанных моделей данных.

При создании базы данных для информационной системы наиболее важными являются задачи, связанные с разработкой правильной логической структуры данных, обеспечивающей выполнение всего требуемого набора функций информационной системы. Плохо продуманная база данных оказывается, как правило, неэффективной и даже бесполезной. Разработка базы данных — достаточно сложная задача. Зачастую к ней предъявляется много противоречивых требований. Создание правильной логической структуры предусматривает комплексный анализ всех факторов, влияющих на формирование и обработку данных.

Задача проектировщика состоит в учете всех этих факторов с целью разработки наиболее оптимальной базы данных. Основные задачи проектирования баз данных можно сформулировать следующим образом:

• обеспечение хранения в БД всей необходимой информации;

• обеспечение возможности получения данных по всем необходимым запросам;

• сокращение избыточности и дублирования данных;

• обеспечение целостности базы данных.

В том, как решаются эти задачи, мы и будем разбираться.

Процесс проектирования БД представляет собой последовательность переходов от неформального словесного описания информационной структуры предметной области к формализованному описанию объектов предметной области в терминах некоторой модели.

В общем случае можно выделить следующие этапы проектирования:

1) системный анализ и словесное описание информационных объектов предметной области;

2) проектирование концептуальной (инфологической) модели предметной области — частично формализованное описание объектов предметной области в терминах некоторой семантической модели;

3) даталогическое или логическое проектирование БД, т.е. описание БД в терминах принятой логической модели данных;

4) физическое проектирование БД, т.е. выбор эффективного размещения БД на внешних носителях для обеспечения наиболее эффективной работы приложения.

На каждом из этих этапов разрабатывается та или иная модель данных. Если мы учтем, что между вторым и третьим этапами необходимо принять решение, с использованием какой стандартной СУБД будет реализовываться наш проект, то условно процесс проектирования БД можно представить последовательностью выполнения соответствующих этапов.



*Концептуальное (инфологическое) проектирование* — построение семантической модели предметной области, т.е. информационной модели наиболее высокого уровня абстракции. Такая модель создается без ориентации на какую-либо конкретную СУБД и модель данных. Термины «семантическая модель», «концептуальная модель» и «инфологическая модель» являются синонимами. Кроме того, в этом контексте равноправно могут использоваться слова «модель базы данных» и «модель предметной области» (например, «концептуальная модель базы данных» и «концептуальная модель предметной области»), поскольку такая модель является образом как реальности, так и проектируемой базы данных для этой реальности.

Конкретный вид и содержание концептуальной модели базы данных определяются выбранным для этого формальным аппаратом. Обычно используются графические нотации, подобные ER-диаграммам.

Чаще всего концептуальная модель базы данных включает в себя:

• описание информационных объектов, или понятий предметной

области и связей между ними;

• описание ограничений целостности, т.е. требований к допустимым значениям данных и к связям между ними.

*Логическое (даталогическое) проектирование —* создание схемы базы данных на основе конкретной модели данных, например реляционной. Для реляционной модели данных логическая модель — набор схем отношений, обычно с указанием первичных ключей, а также «связей» между отношениями, представляющих собой внешние ключи. Преобразование концептуальной модели в логическую, как правило, осуществляется по формальным правилам. Этот этап может быть в значительной степени автоматизирован.

На этапе логического проектирования учитывается специфика конкретной модели данных, но может не учитываться специфика конкретной СУБД.

*Физическое проектирование* — создание схемы базы данных для конкретной системы управления базами данных. Специфика конкретной СУБД может включать в себя ограничения на именование объектов базы данных, на под держиваемые типы данных и т.п. Кроме того, специфика конкретной СУБД при физическом проектировании включает выбор решений, связанных с физической средой хранения данных (выбор методов управления дисковой памятью, разделение БД по файлам и устройствам, методов доступа к данным), создание индексов и т.д.

В области проектирования и разработки баз данных используются различные средства моделирования предметной области и БД, причем даже в рамках одной конкретной системы необходим целый комплекс моделей разного назначения. Далее рассмотрим каждый из этих этапов более подробно.

## 4.2 Анализ предметной области

Анализ предметной области предшествует созданию любой информационной системы и является частью ее разработки. На этом этапе определяются потребности пользователей в информации, которые, в свою очередь, предопределяют структуру и содержание базы данных для будущей системы. Предметная область — некоторая совокупность реальных объектов. Каждый из этих объектов обладает определенным набором свойств (атрибутов). Между объектами предметной области могут существовать связи, имеющие различный содержательный смысл. Создавая базу данных для информационной системы, пользователь стремится упорядочить информацию по различным признакам. Это делается для того, чтобы по необходимости извлекать нужную совокупность данных — получать выборку с желаемым сочетанием признаков. Осуществить это возможно только тогда, когда данные структурированы, т.е. отвечают соглашениям о способах представления данных.

Очень важно на этапе проектирования достичь взаимопонимания как между разработчиками системы, так и между экспертами предметной области, заказчиками и т.д., так как каждый имеет свое видение проекта. При этом работа сводится к поэтапному выделению объектов, значимых функций системы, информационных потоков и системы их взаимосвязей.

Точки зрения участников разработки по определенным проблемам могут совпадать, при этом формы их представления могут быть различными, что ведет к осложнению совместной работы над одним проектом. Важным инструментом в данном случае является использование единого языка представления проектных решений — *языка моделирования.* Нужно определить систему обозначений, правил описания процессов, объектов, явлений и их взаимосвязи, позволяющую всем участникам проекта понимать друг друга («говорить на одном языке»).

*Язык моделирования* — это набор графических нотаций, которые используются для описания моделей в процессе проектирования. *Нотация* представляет собой совокупность графических объектов, используемых в модели, и является синтаксисом языка моделирования. Язык моделирования, с одной стороны, должен делать решения проектировщиков понятными пользователю, с другой — предоставлять проектировщикам средства достаточно формализованного и однозначного определения проектных решений, подлежащих реализации в виде программных комплексов, образующих целостную систему

Комплексность подхода и использование единой нотации очень важно не только на этапе моделирования предметной области, но и на последующих этапах разработки программной системы.

Изучение предметной области складывается из непосредственного наблюдения протекающих в ней процессов, изучения документов, циркулирующих в системе, а также интервьюирования участников этих процессов.

При разработке модели предметной области определяют некоторые границы, в пределах которых можно развивать логическую модель данных, т.е. моделировать объекты, не выходящие за пределы рассматриваемой предметной области. При этом все второстепенные детали опускаются, чтобы чрезмерно не усложнять процесс анализа и исследования полученной модели.

Результатом проведения исследования предметной области должен стать перечень системных требований, спецификаций, информационных потоков и их описание. Очень часто для этого применяются стандартные способы описания предметной области с использованием моделей DFD (диаграмма потоков данных), UML (унифицированный язык моделирования).

# **4.3 Концептуальное моделирование**

Концептуальная модель — это определенное множество понятий и связей между ними, являющихся смысловой структурой рассматриваемой предметной области. Эта модель является начальным прототипом будущей базы данных, но строится без привязки к конкретной СУБД. Процесс разработки концептуальной модели предметной области является творческим и довольно трудно поддается формализации. Для построения концептуальной модели необходимо хорошее знание предметной области, ее семантики, понимание логических взаимосвязей ее информации.

В 70—80-е годы в литературе концептуальное проектирование баз данных обозначалось терминами «инфологическое проектирование» и «инфологическая модель». На рисунке дается схема построения концептуальной модели от исходной информации до результата.

В 1976 г. Питером Ченом была предложена так называемая модель «сущность-связь» *{«entity-relationship»).* Она включает сущности и взаимосвязи, отражающие основные бизнес-правила предметной области. В настоящее время для модели «сущность-связь» общепринятым стало сокращенное название — ER-модель. В дальнейшем многими авторами были разработаны свои варианты подобных моделей. Но все варианты моделей «сущность-связь» исходят из одной идеи — графическое изображение нагляднее текстового описания. В настоящее время ER-модель стала фактически стандартом для семантической структуризации предметной области — стандартом при концептуальном моделировании баз данных.

В качестве стандартной графической нотации, с помощью которой можно визуализировать ER-модель, была предложена диаграмма сущность-связь (ER-диаграмма). ER-диаграмма изображается с помощью трех конструктивных элементов: сущность, атрибут и связь.

*Сущность —* это класс однотипных объектов, информация о которых имеет существенное значение для рассматриваемой предметной области. Сущность представляет собой множество экземпляров реальных или абстрактных объектов (людей, событий, состояний, предметов и т.п.).

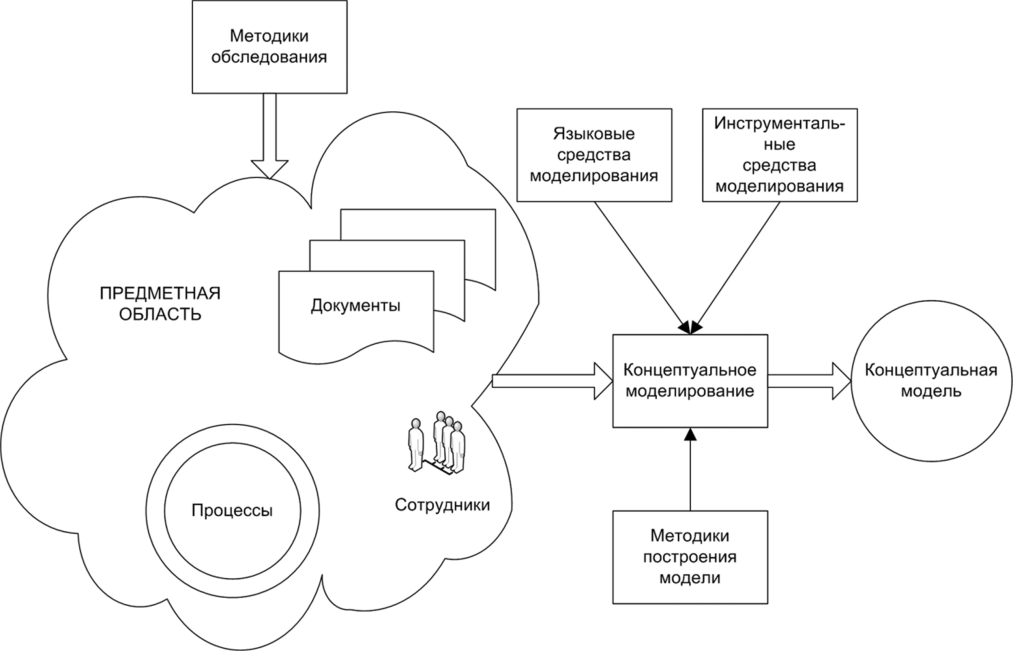
*Экземпляр сущности —* это конкретный представитель данной сущности. Каждый экземпляр сущности должен быть отличим от любого другого экземпляра той же сущности. Например, представителем сущности «Сотрудник» может быть «Сотрудник Иванов И.И.».

Существует несколько систем нотаций (условных обозначений, языков) для описания объектно-связного представления предметной области. Нотация Баркера является наиболее распространенной.

На диаграмме в нотации Баркера сущность изображается прямоугольником, иногда с закругленными углами*.*

Каждая сущность обладает одним или несколькими атрибутами.

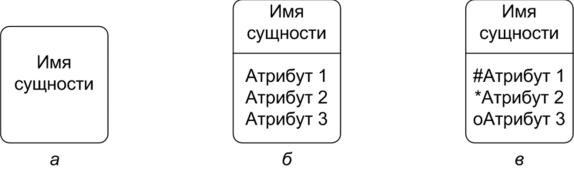
*Атрибут сущности —* это именованная характеристика, являющаяся некоторым свойством сущности. Атрибут предназначен для квалификации, идентификации, классификации, количественной характеристики или выражения состояния сущности*.*



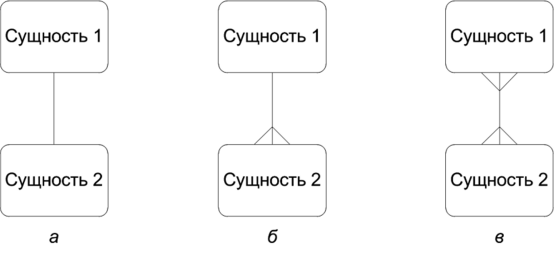
*Экземпляр атрибута —* определенная характеристика конкретного экземпляра сущности, значение атрибута (например, «цвет» — это атрибут, а «зеленый» — экземпляр атрибута).

*Первичный ключ сущности* — это неизбыточный набор атрибутов, значения которых в совокупности являются уникальными для каждого экземпляра сущности. Неизбыточность заключается в том, что удаление любого атрибута из состава ключа приведет к нарушению его уникальности. Первичный ключ предназначен для уникальной идентификации каждого экземпляра сущности, т.е. представляет собой совокупность признаков, позволяющих идентифицировать объект. Ключевые атрибуты помещают в начало списка и помечают символом «#»*.*

*Связь —* это отношение одной сущности к другой или к самой себе. Каждая сущность может обладать любым количеством связей с другими сущностями модели. Различают три типа связей:



Обозначения сущности в нотации Баркера: *а* — без атрибутов;*б —* с указанием атрибутов; *в —* с уточнением атрибутов и их типов (# — ключевой, \* — обязательный, о — необязательный)



Обозначение разных типов связи в нотации Баркера: *а* — «один-к-одному»; *б* — «один-ко-многим»; *в* — «многие-ко-многим»

Связь типа «один-к-одному» означает, что один экземпляр первой сущности связан только с одним экземпляром второй сущности. Такая связь, скорее всего, свидетельствует о том, что была неверно разделена одна сущность на две (хотя иногда к такому типу связи прибегают в случае, если есть необходимость «засекретить» часть данных). Связь типа «один-ко-многим» означает, что каждый экземпляр первой сущности связан с несколькими экземплярами второй сущности. Связь типа «многие-ко-многим» означает, что каждый экземпляр первой сущности может быть связан с несколькими экземплярами второй сущности, и наоборот. Такой тип связи является временным, допустимым на ранних этапах разработки модели. В дальнейшем такую связь необходимо заменить двумя связями типа «один-ко-многим» путем выделения промежуточной сущности.

Некоторые ограничения целостности задаются на ER-диаграмме, другие описываются на естественном языке

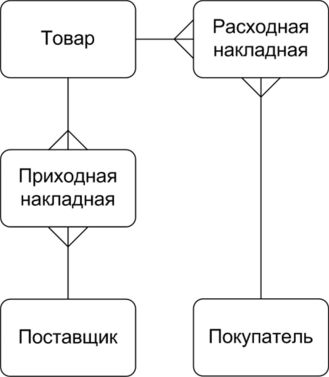
В качестве примера рассмотрим небольшую базу данных, отражающую процесс поставки и продажи некоторого товара постоянным клиентам. База данных проектируется для автоматизированной информационной системы «Склад оптовой торговли».

Исходя из анализа предметной области, можно выделить следующие сущности — Товар, Поставщик и Покупатель. Сразу возникает очевидная связь между сущностями — «покупатели могут приобретать много товаров», «товары могут приобретаться многими покупателями» Отношение между ними относится к типу «многие-ко-многим».



Однако реляционная модель данных требует заменить отношение «многие-ко-многим» на несколько отношений «один-ко-многим». Для разрешения этого отношения добавим еще один тип сущностей, отображающий процесс продажи/покупки товаров — введем ассоциированные сущности «Приходная накладная» и «Расходная накладная».

Установим связи между объектами. Один покупатель может неоднократно покупать товары, поэтому между объектами «Покупатель» и «Расходная накладная» имеется связь «один-ко-многим». Каждое наименование товара может неоднократно участвовать в сделках, в результате между объектами «Товар» и «Расходная накладная» имеется связь «один-ко-многим». Аналогично строятся связи между сущностями «Приходная накладная» и «Товар».



Проанализируем атрибуты сущностей. Каждый поставщик и покупатель являются юридическими лицами и имеют наименование, адрес, банковские реквизиты. Каждый товар имеет наименование, цену, характеризуется единицей измерения. Каждая накладная имеет уникальный номер, дату выписи, список товаров с количествами и ценами, а также общую сумму накладной. Покупатели покупают товары, получая при этом расходные накладные, в которые внесены данные о количестве и цене приобретенного товара. Каждый покупатель может получить несколько накладных. Каждая накладная должна выписываться на одного покупателя и должна содержать не менее одного товара (не может быть «пустой» накладной). Каждый товар, в свою очередь, может быть продан нескольким покупателям по нескольким накладным. Аналогичную цепь рассуждений можно выстроить для определения связей между сущностями «Товар» и «Поставщик». Покупатель может быть одновременно и поставщиком, поэтому эти два объекта объединены в одну сущность «Контрагент».

Важной задачей в создании базы данных является выявление первичных ключей. Для таблицы «Товар» название товара не может служить первичным ключом, так как товары разных типов могут иметь одинаковые названия, поэтому введем первичный ключ «Код», под которым можно понимать, например, артикул товара. Точно так же «Наименование» не может служить первичным ключом в таблице «Контрагенты». Введем первичный ключ «Код» для поставщиков и покупателей, под которым можно понимать номер паспорта, идентификационный номер налогоплательщика или любой другой атрибут, однозначно определяющий каждого контрагента.

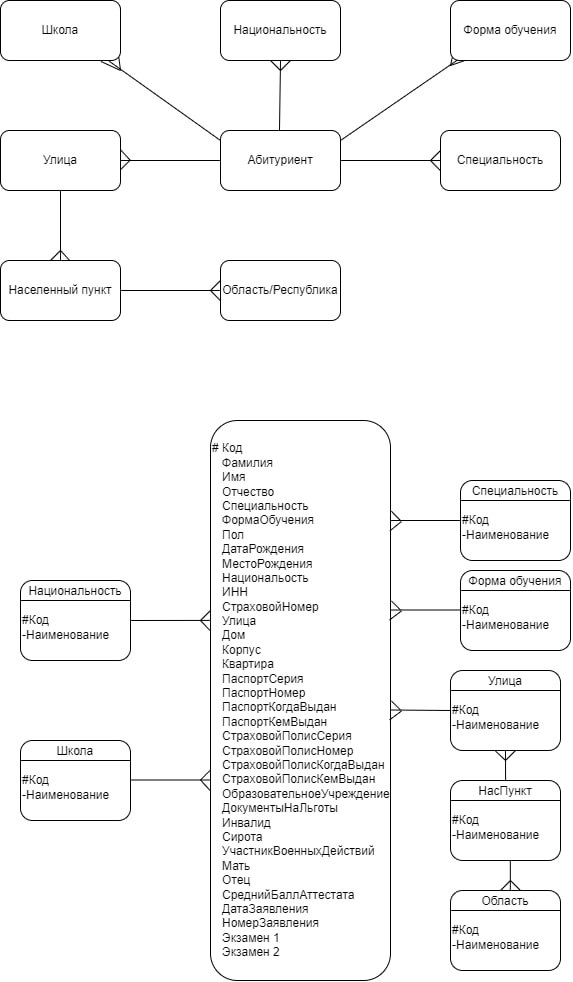
Для таблиц «Расходная накладная»/«Приходная накладная» первичным ключом является поле «Номер», так как номер однозначно определяет каждую накладную. Теперь можно все это внести в диаграмму. После уточнения диаграмма будет выглядеть следующим образом.



Данная диаграмма должна быть проверена с точки зрения возможности получения всех выходных данных, показанных на диаграмме потоков данных разрабатываемой системы.

В качестве самостоятельного объекта в ER-модели следует изображать сущности, для которых фиксируются какие-либо их свойства, либо участвующие более чем в одной связи. При возникновении сомнений лучше принять решение о создании самостоятельного объекта, так как это в дальнейшем потребует меньших изменений в модели.

Следует помнить, что количественные характеристики всегда являются свойствами какого-либо объекта, и никогда — самостоятельными объектами. Например, год рождения не может рассматриваться как самостоятельный объект.



Построение концептуальной модели может выполняться как «вручную», так и с использованием автоматизированных средств проектирования — CASE-средств.