

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«МЭИ»

Н.В. Усманова

ПРОЕКТИРОВАНИЕ БАЗ ДАННЫХ

Методические указания по выполнению курсовой работы

по курсу
«**Базы данных**»

Москва
2020

СОДЕРЖАНИЕ

1. ОБЩИЕ ПОЛОЖЕНИЯ	3
2. ВЫБОР ТЕМЫ КУРСОВОЙ РАБОТЫ.....	4
3. ПОСЛЕДОВАТЕЛЬНОСТЬ ПРОЕКТИРОВАНИЯ БАЗЫ ДАННЫХ. СОДЕРЖАНИЕ КУРСОВОЙ РАБОТЫ.....	5
4.ТРЕБОВАНИЯ К ОФОРМЛЕНИЮ КУРСОВОЙ РАБОТЫ	12
5. ПОРЯДОК АТТЕСТАЦИИ И ЗАЩИТЫ КУРСОВОЙ РАБОТЫ	16
ПРИЛОЖЕНИЯ	17

1. ОБЩИЕ ПОЛОЖЕНИЯ

Цель курсового проектирования – применение на практике знаний, полученных в процессе изучения курса «Базы данных», и получение практических навыков создания автоматизированных информационных систем (АИС), основанных на базах данных.

Проектирование базы данных (БД) является одной из наиболее сложных и ответственных задач, связанных с созданием АИС.

Проектирование базы данных – это процесс, который подразумевает использование определённой технологии. Никто не сомневается в том, что в случае нарушения технологии изготовления печатной платы, например, эта плата либо вообще не будет работать, либо не будет соответствовать заявленным характеристикам. Но почему-то считается, что соблюдать технологию проектирования БД (и вообще программного обеспечения) совершенно необязательно. И начинают работу по реализации реляционной БД с создания таблиц. Получившаяся в ходе такого «проектирования» база данных будет ненадёжной, неэффективной и сложной в сопровождении. (Исключением могут быть случаи простых предметных областей, которые можно отразить в базе данных, состоящей из 3-4 таблиц). Поэтому при создании базы данных необходимо придерживаться определённой технологии проектирования БД.

Опишем вкратце процесс проектирования реляционной базы данных. База данных – это, фактически, модель предметной области (ПрО). Значит, для создания БД надо сначала проанализировать ПрО и создать её модель (это называется **инфологическим проектированием**).

Основой для *анализа предметной области* служат документы, которые отражают ПрО, и информация, которую можно получить от специалистов этой предметной области в процессе общения с ними.

Для анализа берутся те документы, которые имеют отношение к решаемой задаче. Изучение документов позволяет выявить объекты (сущности ПрО) и атрибуты сущностей – данные, которые должны храниться в БД.

Из общения со специалистами необходимо извлечь сведения об особенностях ПрО, которые позволяют установить ограничения целостности, зависимости и связи между объектами (субъектами) предметной области. Также специалисты обладают знаниями о том, каковы алгоритмы обработки данных и какие задачи ставятся перед информационной системой.

Модель ПрО может быть описана любым удобным для разработчика способом (словесное описание, набор формул, диаграмма потоков данных и т.п.). Но, если при проектировании баз данных используется метод

сущность–связь, то схема ПрО выполняется в виде ER–диаграммы (entity-relation diagram, диаграмма «сущность-связь»).

После создания модели ПрО определяются **требования к операционной обстановке**: какое аппаратное и программное обеспечение необходимо для реализации БД и АИС в целом. Основные технические параметры (объём оперативной и дисковой памяти, наличие сетевой платы и др.) определяются исходя из планируемого объёма БД, режима работы (локальный или удалённый доступ) и требований к эффективности работы системы (например, ко времени реакции на запрос пользователя или к общей производительности БД). В зависимости от планируемой нагрузки (интенсивности запросов) и требований к надёжности выбирается операционная система. Затем осуществляется выбор СУБД, под управлением которой будет работать создаваемая база данных.

На следующем этапе – этапе **логического проектирования** – ER-диаграмма формальным способом преобразуется в схему реляционной базы данных (РБД). На основании схемы РБД и описания сущностей ПрО составляются отношения (таблицы) базы данных. Потом выполняется нормализация отношений. Это необходимо сделать для того, чтобы исключить нарушения логической целостности данных и повысить таким образом надёжность и достоверность данных. В отдельных случаях после нормализации может выполняться денормализация, но причина для этого может быть только одна: повышение эффективности выполнения критических запросов.

В результате всех этих операций создаётся концептуальная схема БД – основной документ для базы данных.

Далее, на этапе **физического проектирования** полученные отношения описываются на языке DDL (Data definition language) – языке определения данных, который поддерживается выбранной СУБД. Также необходимо определить способы хранения данных (кластеризация, хеширование) и способы доступа к данным (индексирование) и создать соответствующие индексы и кластеры (если нужно). Если пользователей АИС можно разделить на группы по характеру решаемых задач, то для каждой группы создаётся свой набор прав доступа к объектам БД.

2. ВЫБОР ТЕМЫ КУРСОВОЙ РАБОТЫ

Необходимо выбрать предметную область, для которой будет разрабатываться база данных.

Список вариантов предметных областей приведен ниже, а также Вы можете предложить свой вариант. Варианты предметных областей для БД:

1. Разработать БД для кафе.
2. Разработать БД для поликлиники.
3. Разработать БД для учебного заведения.
4. Разработать БД для аптеки.
5. Разработать БД для магазина.
6. Разработать БД для агентства недвижимости.
7. Разработать БД для автосервиса.
8. Разработать БД для автовокзала.
9. Разработать БД для салона красоты.
10. Разработать БД для библиотеки.
11. Разработать БД для гостиницы.
12. Разработать БД для биржи труда.
13. Разработать БД для транспортной компании.
14. Разработать БД для детского сада.
15. Разработать БД для кинотеатра.

Тема курсовой работы будет формулироваться следующим образом: «Проектирование базы данных кафе (поликлиники, учебного заведения и т.п.)».

3. СОДЕРЖАНИЕ КУРСОВОЙ РАБОТЫ ПОСЛЕДОВАТЕЛЬНОСТЬ ПРОЕКТИРОВАНИЯ БАЗЫ ДАННЫХ.

1. Введение

Структура введения должна содержать ряд обязательных элементов: актуальность выбранной темы, цель написания курсовой работы, решаемые задачи, объект и предмет курсовой работы, информационная база.

При обосновании темы студент должен объяснить, в чем **актуальность** курсовой работы, как она связана с происходящим процессами на различных уровнях агрегации.

Цель работы непосредственно должна быть связана с темой курсовой работы и быть ясной, лаконичной (не более двух-трех предложений) и корреспондироваться с темой курсовой работы. Как правило, студент должен указать, что он собирается сделать: *проанализировать, обосновать, раскрыть, выявить, охарактеризовать, разработать*. Цель работы является наиболее масштабным конкретно-практическим результатом, достижению которой будет посвящена курсовая работа и должна быть связана с названием работы, а также комплексом рекомендаций, сформированных для объекта исследования.

Задачи являются направлениями в достижении цели. Необходимо помнить, что достижение цели в работе возможно через решение

различного рода комбинаций задач. Следовательно, приводятся 2-3 задачи курсовой работы, вытекающие из цели и конкретизирующие ее.

Цель и задачи курсовой работы должны представлять основные пути решения проблемы (или реализации возможности), заявленной в названии курсовой работы.

Во введении обязательно указывается **объект и предмет** исследования, которые непосредственно связаны с сущностью темы и экономическими отношениями, возникающими в процессе различного рода деятельности.

Введение не должно превышать 1-2 страницы машинописного текста.

2. Расчётно-практическая часть.

Последовательность проектирования базы данных

Процесс проектирования включает в себя следующие шаги:

1. Определение задач, стоящих перед базой данных.
2. Сбор и анализ документов, относящихся к исследуемой предметной области.
3. Описание особенностей ПрО, которые позволяют установить зависимости и связи между объектами (субъектами) предметной области.
4. Создание модели предметной области.
5. Определение групп пользователей и перечня задач, стоящих перед каждой группой.
6. Выбор аппаратной и программной платформы для реализации БД.
7. Создание логической схемы БД.
8. Создание схем отношений, определение типов данных атрибутов и ограничений целостности.
9. Нормализация отношений (до третьей или четвёртой нормальной формы).
10. Определение прав доступа пользователей к объектам БД.
11. Написание текста создания основных объектов базы данных на языке SQL в синтаксисе выбранной СУБД (пользователи, таблицы и др.).
12. Написание текста создания вспомогательных объектов базы данных (представления, индексы, триггеры, роли и т.д.).

Эти шаги можно объединить с 3 этапа: 1. Инфологическое проектирование (1-6). 2. Логическое проектирование БД (7-10). 3. Физическое проектирование БД (11-12). На сегодняшний день не существует формальных способов моделирования реальности, но инфологический подход закладывает основы методологии проектирования базы данных как модели предметной области.

1. Инфологическое проектирование

Основными задачами этапа инфологического проектирования являются определение предметной области системы и формирование взгляда на неё с позиций сообщества будущих пользователей БД, т.е. информационно-логической модели ПрО.

Инфологическая модель ПрО представляет собой описание структуры и динамики ПрО, характера информационных потребностей пользователей в терминах, понятных пользователю и не зависящих от реализации БД. Это описание выражается в терминах не отдельных объектов ПрО и связей между ними, а их типов, связанных с ними ограничений целостности и тех процессов, которые приводят к переходу ПрО из одного состояния в другое.

Основными подходами к созданию инфологической модели предметной области являются:

1. Функциональный подход к проектированию БД («от задач»).
2. Предметный подход к проектированию БД («от предметной области»).
3. Метод «сущность-связь» (entity–relation, ER–method).

Мы будем использовать метод «сущность–связь» как наиболее распространённый (в нотации П. Чена). Приведём основные термины, которыми мы будем пользоваться:

Сущность – это объект, о котором в системе будут накапливаться данные. Для сущности указывается название и тип (сильная или слабая). Сильные сущности существуют сами по себе, а существование слабых сущностей зависит от существования сильных.

Атрибут – свойство сущности. Различают:

1) *Идентифицирующие и описательные атрибуты*. Идентифицирующие позволяют отличить один экземпляр сущности от другого. Описательные атрибуты включают в себя интересующие нас свойства сущности.

2) *Составные и простые атрибуты*. Простой атрибут имеет неделимое значение. Составной атрибут является комбинацией нескольких элементов, возможно, принадлежащих разным типам данных (ФИО, адрес и др.).

3) *Однозначные и многозначные атрибуты* (могут иметь соответственно одно или много значений для каждого экземпляра сущности). Например, дата рождения – это однозначный атрибут, а номер телефона – многозначный.

4) *Основные и производные атрибуты*. Значение основного атрибута не зависит от других атрибутов; значение производного атрибута вычисляется на основе значений других атрибутов. Например, возраст вычисляется на основе даты рождения и текущей даты.

5) *Обязательные и необязательные* (первые должны быть указаны при размещении данных в БД, вторые могут не указываться).

Для каждого атрибута необходимо определить название, указать тип данных и описать ограничения целостности – множество значений, которые может принимать данный атрибут.

Связь – это осмысленная ассоциация между сущностями. Для связи указывается название, тип (факультативная или обязательная), кардинальность (1:1, 1:n или m:n) и степень (унарная, бинарная, тернарная или n-арная).

На рис. 1 приведены обозначения, которые мы будем использовать в ER-диаграммах.

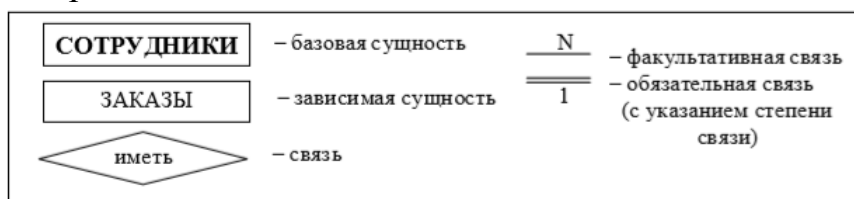


Рис.1. Обозначения, используемые в ER-диаграммах

Определение требований к операционной обстановке

После создания модели ПрО определяются требования к **операционной обстановке**. На этом этапе производится оценка требований к вычислительным ресурсам, необходимым для функционирования системы, определение типа и конфигурации конкретной ЭВМ, выбор типа и версии операционной системы. Объём вычислительных ресурсов зависит от предполагаемого объёма проектируемой базы данных и от интенсивности их использования. Если БД будет работать в многопользовательском режиме, то требуется подключение её к сети и наличие соответствующей многозадачной операционной системы.

2. Логическое проектирование реляционной БД

На этапе логического проектирования разрабатывается логическая (концептуальная) структура БД. Для реляционной модели существуют формальные правила, которые позволяют преобразовать инфологическую модель ПрО в виде ER-диаграммы в логическую схему базы данных. Кроме получения схемы БД в целом на этом этапе выполняют создание схем отношений и их нормализацию. Этот этап более подробно рассмотрен в Приложении 4 "Пример проектирования реляционной базы данных".

Проектирование схемы БД должно решать задачи минимизации дублирования данных и упрощения процедур их обработки и обновления. При неправильно спроектированной схеме БД могут возникнуть аномалии модификации данных. Они обусловлены отсутствием средств

явного представления типов множественных связей между объектами ПрО и неразвитостью средств описания ограничений целостности на уровне модели данных.

Для решения подобных проблем проводится **нормализация отношений**. Механизм нормализации реляционных отношений разработал Э.Ф. Кодд (E.F. Codd). Этот механизм позволяет по формальным признакам любое отношение преобразовать к третьей нормальной форме.

Проектирование реляционной базы данных проходит в том же порядке, что и проектирование БД других моделей данных, но имеет свои особенности. Проектирование схемы БД должно решать задачи минимизации дублирования данных и упрощения процедур их обработки и обновления. При неправильно спроектированной схеме БД могут возникнуть аномалии модификации данных. Они обусловлены отсутствием средств явного представления типов множественных связей между объектами ПрО и неразвитостью средств описания ограничений целостности на уровне модели данных. Для решения подобных проблем проводится нормализация отношений. Механизм нормализации реляционных отношений разработал Э.Ф. Кодд (E.F. Codd). Этот механизм позволяет по формальным признакам любое отношение преобразовать к третьей нормальной форме.

Нормализация схемы отношения выполняется путём декомпозиции схемы. **Декомпозицией** схемы отношения R называется замена её совокупностью схем отношений A_i таких, что

$$R = \bigcup_i A_i$$

и не требуется, чтобы отношения A_i были непересекающимися.

Первая нормальная форма относится к понятию простого и сложного (составного или многозначного) атрибута.

Первая нормальная форма (1НФ).

Отношение приведено к 1НФ, если все его атрибуты простые.

Для того чтобы привести к 1НФ отношение, содержащее сложные атрибуты, нужно:

- 1) разбить составные атрибуты на простые,
- 2) построить декартово произведение всех многозначных атрибутов с кортежами, к которым они относятся.

Для идентификации кортежа в этом случае понадобится составной ключ, включающий первичный ключ исходного отношения и все многозначные атрибуты.

Вторая нормальная форма основана на понятии *функциональной зависимости*. Пусть X и Y – атрибуты некоторого отношения. Если в любой момент времени каждому значению X соответствует единственное

значение Y , то говорят, что Y функционально зависит от X ($X \rightarrow Y$). Атрибут X в функциональной зависимости $X \rightarrow Y$ называется *детерминантом* отношения.

В нормализованном отношении все неключевые атрибуты функционально зависят от ключа отношения. Неключевой атрибут функционально полно зависит от составного ключа, если он функционально зависит от ключа, но не находится в функциональной зависимости ни от какой части составного ключа.

Вторая нормальная форма (2НФ).

Отношение находится во 2НФ, если оно приведено к 1НФ и каждый неключевой атрибут функционально полно зависит от составного первичного ключа. (Таким образом, если отношение в 1НФ имеет простой первичный ключ, оно сразу находится во второй нормальной форме).

Для того чтобы привести отношение ко 2НФ, нужно:

- построить его проекцию, исключив атрибуты, которые не находятся в функционально полной зависимости от составного первичного ключа;
- построить дополнительно одну или несколько проекций на часть составного ключа и атрибуты, функционально зависящие от этой части ключа.

Третья нормальная форма основана на понятии *транзитивной зависимости*. Пусть X, Y, Z – атрибуты некоторого отношения. При этом $X \rightarrow Y$ и $Y \rightarrow Z$, но обратное соответствие отсутствует, т.е. Z не зависит от Y или Y не зависит от X . Тогда говорят, что Z транзитивно зависит от X ($X \rightarrow \rightarrow Z$).

Третья нормальная форма (3НФ).

Отношение находится в 3НФ, если оно находится во 2НФ и каждый неключевой атрибут нетранзитивно зависит от первичного ключа.

Для того чтобы привести отношение к 3НФ, нужно:

- построить проекцию, исключив транзитивно зависящие от ключа атрибуты;
- построить дополнительно одну или несколько проекций на детерминанты исходного отношения и атрибуты, функционально зависящие от них.

Исключение составляют случаи, когда для транзитивной зависимости $X \rightarrow \rightarrow Z$ ($X \rightarrow Y$ и $Y \rightarrow Z$) либо Z зависит от Y , либо Y зависит от X , т.е. между атрибутами X и Y , например, существует связь 1:1. В такой ситуации декомпозиция отношения не производится.

Четвертая нормальная форма основана на понятии *многозначной зависимости*. Многозначная зависимость существует, если заданным значениям атрибута X соответствует множество, состоящее из нуля (или более) значений атрибута Y ($X \twoheadrightarrow Y$).

Различают тривиальные и нетривиальные многозначные

зависимости. *Тривиальной* называется такая многозначная зависимость $X \twoheadrightarrow Y$, для которой $Y \subset X$ или $X \cup Y = R$, где R – рассматриваемое отношение. Тривиальная многозначная зависимость не нарушает 4НФ. Если хотя бы одно из двух этих условий не выполняется, то такая зависимость называется *нетривиальной*.

Четвертая нормальная форма (4НФ).

Отношение находится в 4НФ, если оно находится в 3НФ и в нём отсутствуют нетривиальные многозначные зависимости.

Для того чтобы привести отношение к 4НФ, нужно построить две или более проекции исходного отношения, каждая из которых содержит ключ и одну из многозначных зависимостей.

3. Физическое проектирование БД

Этап физического проектирования заключается в определении схемы хранения, т.е. физической структуры БД. Схема хранения зависит от той физической структуры, которую поддерживает выбранная СУБД. Физическая структура БД, с одной стороны, должна адекватно отражать логическую структуру БД, а с другой стороны, должна обеспечивать эффективное размещение данных и быстрый доступ к ним. Результаты этого этапа документируются в форме схемы хранения на языке определения данных (DDL, Data Definition Language) выбранной СУБД. Принятые на этом этапе решения оказывают огромное влияние на производительность системы.

Одной из важнейших составляющих проекта базы данных является разработка средств защиты БД. Защита данных имеет два аспекта: защита от сбоев и защита от несанкционированного доступа. Для защиты от сбоев на этапе физического проектирования разрабатывается стратегия резервного копирования. Для защиты от несанкционированного доступа каждому пользователю доступ к данным предоставляется только в соответствии с его правами доступа, набор которых также является составной частью проекта БД.

Объем расчетно-практической главы должен составлять примерно 20-25 страниц машинописного текста, включая таблицы и рисунки.

3. Заключение

В Заключении подводятся итоги проведенного исследования, кратко формулируются основные результаты курсовой работы, в обобщенном виде излагаются выводы о реализации целей и задач курсовой работы.

В Заключении студент должен обоснованно изложить свои взгляды на основные вопросы, рассмотренные в работе. Заключение не должно

содержать новых сведений, фактов, аргументов и т. п., а выводы должны логически вытекать из основного текста работы.

Заключение является структурно-логической основой доклада (презентации результатов курсовой работы) в процессе защиты курсовой работы.

Заключение не должно превышать 1-2 страницы машинописного текста.

4. Список используемых источников и Интернет-ресурсов

Список использованных источников и Интернет-ресурсов содержит источники и литературу, использованные студентом в ходе подготовки и написания курсовой работы и состоит из **не менее 10-15 названий различных источников** (монографий, журнальных и газетных статей, справочников, учебников, учебных пособий и официальных документов, Интернет-ресурсы).

Список имеет следующую структуру:

- I. Нормативно-правовые акты
- II. Монографии, учебники и учебные пособия
- III. Статьи в периодических изданиях
- IV. Справочные материалы и Интернет-ресурсы
- V. Источники на иностранном языке

Источники следует располагать в алфавитном порядке.

5. Приложения

Раздел «Приложения» должен содержать текстовые документы, графики, диаграммы, схемы, карты, таблицы, а также расчеты, выполненные с применением вычислительной техники. Они служат для иллюстрации отдельных положений исследуемой проблемы или являются результатом предлагаемых рекомендаций автора. Приложения помещают после списка литературы в порядке их упоминания в тексте. С помощью сносок следует указать, на основании каких источников составлено приложение (либо составлено автором).

Общий объем курсовой работы без приложений составляет примерно 25-30 страниц.

4.ТРЕБОВАНИЯ К ОФОРМЛЕНИЮ КУРСОВОЙ РАБОТЫ

Курсовая работа должна быть оформлена на одной стороне листа бумаги формата А4 (ориентация-книжная), содержит, примерно, 1800 знаков на странице (включая пробелы и знаки препинания). Допускается представлять таблицы и иллюстрации на листах бумаги формата не более

А3.

Страницы, на которых излагается текст, должны иметь следующие поля страницы: верхнее – 20 мм; нижнее – 20 мм; левое – 30 мм; правое – 10 мм.

Шрифт курсовой работы Times New Roman, размер шрифта- 14 (текст в таблицах-12), межстрочный интервал – 1,5 строки, абзацный отступ-1,25. Подчеркивание слов и выделение их курсивом не допускается.

Названия структурных элементов «ВВЕДЕНИЕ», «ЗАКЛЮЧЕНИЕ», «СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ И ИНТЕРНЕТ-РЕСУРСОВ», «ПРИЛОЖЕНИЕ» и заголовки глав, печатаются прописными (заглавными) буквами, полужирного начертания.

Заголовки параграфов (подзаголовки) набирают строчными (кроме первой прописной) полужирными буквами с отступом от предыдущего текста в 1 пустую строку.

Заголовки, подзаголовки и подстрочные сноски (состоящие из нескольких строк) печатаются через одинарный интервал. Абзацный отступ должен соответствовать 0,7 см и быть одинаковым по всей работе.

Нумерация разделов производится арабскими цифрами без точки в конце. Главы делятся на параграфы и нумеруются арабскими цифрами без точки в конце, а именно:

Пример:

ГЛАВА 1. ИНФОЛОГИЧЕСКОЕ ПРОЕКТИРОВАНИЕ

1.1. Анализ предметной области

На титульном листе курсовой работы (проекта) указывается наименование университета, института, кафедры, группы, название темы курсовой работы, фамилия и инициалы автора работы и руководителя, год написания работы (**см. Приложение 1**).

На втором листе приводится содержание курсовой работы (**см. Приложение 2**).

Отклонение от нормативного объема работы является серьезным нарушением и повлечет за собой снижение оценки!

Параграфы (разделы) должны иметь нумерацию в пределах каждой главы (раздела), а главы (разделы) – в пределах всего текста работы. Если глава содержит только один параграф (что нежелательно), то нумеровать его не нужно.

Нумерация страниц

Страницы курсовой работы должны нумероваться арабскими цифрами, нумерация должна быть сквозная, по всему тексту работы. Номер страницы проставляют в центре нижней части листа без точки.

Титульный лист включается в общую нумерацию страниц работы, однако номер страницы на нем не ставится.

Если в работе имеются иллюстрации и таблицы на отдельном листе, то они включаются в общую нумерацию страниц работы. Каждую главу работы следует начинать с нового листа. Параграф начинать с нового листа не нужно.

Иллюстрации и таблицы. Если в работе имеются схемы, таблицы, графики, диаграммы, фотоснимки, то их следует располагать непосредственно после текста, в котором они упоминаются впервые, или на следующей странице. Иллюстрации следует нумеровать арабскими цифрами сквозной нумерацией (то есть по всему тексту) – 1,2,3, и т.д., либо внутри каждой главы – 1.1,1.2, и т.д.

При наличии в работе таблицы ее наименование (краткое и точное) должно располагаться над таблицей слева, без абзацного отступа в одну строку с номером через тир. Ее название (заголовок) пишется с прописной буквы, и точка в конце не ставится.

Таблицу, как и рисунок, располагать непосредственно после текста, в котором она упоминается впервые, или на следующей странице. Таблицы в тексте следует нумеровать сквозной нумерацией арабскими цифрами по всему тексту или в рамках главы (2.1 и т.д.). Если таблица вынесена в приложение, то она нумеруется отдельно арабскими цифрами с добавлением перед номером слова «Приложение» - Приложение 1.

Разрывать таблицу и переносить часть ее на другую страницу можно только в том случае, если целиком не уместается на одной странице. При этом таблица, перетекающая с одной страницы на другую, должна быть **ОДНОЙ** таблицей с одним заголовком.

Пример оформления таблиц и рисунков представлен в Приложении 3.

Список использованных источников и Интернет-ресурсов

После заключения, начиная с новой страницы, необходимо поместить список использованных источников и Интернет-ресурсов. Список использованных источников должен содержать подробную информацию о каждом использованном источнике. Такая информация различна в зависимости от вида источника. В любом случае, основой оформления списка использованных источников является библиографическое описание источников.

Образцы библиографических описаний произведений печати в списках литературы

1. Описание книги одного автора

Маркин, А. В. Программирование на SQL : учебное пособие для среднего профессионального образования / А. В. Маркин. - Москва : Издательство Юрайт, 2020. - 435 с.

Тягунов, С. И. Логика как искусство мышления: Учеб. пособие / С. И. Тягунов. - СПб.: Изд-во СПбГУЭФ, 2015. - 107 с.

2. Описание книги 2, 3-х авторов

Стружкин, Н. П. Базы данных: проектирование : учебник для вузов / Н. П. Стружкин, В. В. Годин. - Москва : Издательство Юрайт, 2020. - 477 с.

Хомоненко А.Д. Базы данных: Учебник для высших учебных заведений / А.Д. Хомоненко, В.М. Цыганков, М.Г. Мальцев; Под ред. проф. А.Д. Хомоненко. - СПб.: КОРОНА принт, 2000. - 416с.

Eckhouse, R.H. Minicomputer systems. Organization, programming and application / R.H. Eckhouse, H.R. Morris. - New York, 1999. - 491 p.

3. Описание книги 4-х и более авторов

Производственный менеджмент / С.Д. Ильенкова, А.В. Бандурин, Г.А. Горбовцов; Под ред. С.Д. Ильенкова. - М.: ЮНИТИ, 2000. - 583с.

4. Описание статей из газет, журналов и сборников

Федоров, В.Н. Управление электроприводами кузнечно-прессового оборудования/В.Н. Федоров // Сб. науч. тр. института /ВоГТУ. Т. 1. - Вологда, 1997. - С. 65-72.

Зиненко, В.И. Охрана природы в городе / В.И.Зиненко // Знание-сила. - 2002 .- № 3. - С. 6-14.

Сенаторов, А. Япония: коалиционный выбор либерал-демократов /А.Сенаторов, И.Цветов // Проблемы Дальнего Востока. - 2000. - № 1. - С.30-41.

5. Описание нормативно-правовых актов

О государственной судебно-экспертной деятельности в Российской Федерации: Федер.закон от 31 мая 2001 г. № 73-ФЗ // Ведомости Федер.Собр.Рос.Федерации. - 2001. - N 17. - С. 11-28.

О некоторых вопросах Федеральной налоговой полиции: Указ Президента РФ от 25.02.2000 № 433 // Собрание законодательства РФ. - 2000. - № 9. -Ст.1024.

ГОСТ 34.602–89. Информационная технология. Технические задания на создание автоматизированной системы.

ГОСТ 19.701–90 (ИСО 5807–85). Единая система программной документации. Схемы алгоритмов, программ данных и систем. Условные обозначения и правила выполнения.

6. Описание диссертаций, авторефератов диссертаций, депонированных рукописей

Новосельский, В.Б. Методы автоматизации проектирования распределенных баз данных: дис. канд. техн. наук : 05.13.12 / В.Б. Новосельский. С.-Петерб. гос. ун-т информац. технологий, механики и оптики.- Санкт-Петербург, 2008.- 114 с.

Новосельский, В.Б. Методы автоматизации проектирования распределенных баз данных: автореф. дис. канд. техн. наук : 05.13.12 / В.Б. Новосельский. С.-Петербург. гос. ун-т информац. технологий, механики и оптики.- Санкт-Петербург, 2008.- 16 с.

7. Электронные ресурсы

Илюшечкин, В. М. Основы использования и проектирования баз данных : учебник для среднего профессионального образования / В. М. Илюшечкин. - испр. и доп. - Москва : Издательство Юрайт, 2020. - 213 с. [Электронный ресурс]. - URL: <https://urait.ru/bcode/452874> (дата обращения: 10.09.2019).

Общие требования к приложениям

Приложения – дополнительные к основному тексту материалы справочного, документального, иллюстративного или другого характера. Приложения размещаются в конце работы, после списка использованной литературы в порядке их упоминания в тексте. Каждое приложение должно начинаться с нового листа, и иметь тематический заголовок и общий заголовок «Приложение № ____». Если приложение представляет собой отдельный рисунок или таблицу, то оно оформляется в соответствии с требованиями, предъявляемыми к иллюстрациям, таблицам.

Иллюстрации и таблицы нумеруются в пределах каждого приложения в отдельности. Например: рис. 3.1 (первый рисунок третьего приложения), таблица 1.1 (первая таблица первого приложения).

5. ПОРЯДОК АТТЕСТАЦИИ И ЗАЩИТЫ КУРСОВОЙ РАБОТЫ

Руководитель проверяет работу, при условии законченного оформления и положительной оценки содержания, **допускает работу** к защите. Работа, не отвечающая установленным требованиям, возвращается для доработки с учетом сделанных замечаний и повторно представляется на кафедру в срок, указанный руководителем, но до начала зачетной сессии.

Защита курсовой работы проводится в форме компьютерного тестирования.

Формой аттестации студента по курсовой работе является дифференцированный зачет («отлично», «хорошо», «удовлетворительно», «неудовлетворительно»).

Студент, не представивший в установленный срок курсовую работу или не защитивший ее по неуважительной причине, считается имеющим академическую задолженность.

ПРИЛОЖЕНИЕ 1

Оформление титульного листа курсовой работы
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО
ОБРАЗОВАНИЯ
МОСКОВСКИЙ ЭНЕРГЕТИЧЕСКИЙ ИНСТИТУТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)**

Институт дистанционного и дополнительного образования

**КУРСОВАЯ РАБОТА ПО ДИСЦИПЛИНЕ
«БАЗЫ ДАННЫХ»**
на тему: «Проектирование базы данных проектной организации»

Выполнил студент группы ИД-0601-17 Попов И.И.

Научный руководитель Усманова Н.В.

К защите допущен _____
(дата, подпись руководителя)

Работа защищена с оценкой _____
(дата, подпись руководителя)

Москва-2020

ПРИМЕР СОДЕРЖАНИЯ (СТРУКТУРЫ) КУРСОВОЙ РАБОТЫ ПО ТЕМЕ:

«Проектирование базы данных проектной организации»

ВВЕДЕНИЕ	3
ГЛАВА 1. ИНФОЛОГИЧЕСКОЕ ПРОЕКТИРОВАНИЕ	4
1.1. Анализ предметной области... ..	4
1.2. Анализ информационных задач и круга пользователей системы.....	8
1.3. Определение требований к операционной обстановке	14
ГЛАВА 2. ЛОГИЧЕСКОЕ ПРОЕКТИРОВАНИЕ РЕЛЯЦИОННОЙ БАЗЫ ДАННЫХ	17
2.1. Преобразование ER–диаграммы в схему базы данных	17
2.2. Составление реляционных отношений	19
2.3. Нормализация полученных отношений	20
2.4. Определение дополнительных ограничений целостности	17
2.5. Описание групп пользователей и прав доступа.....	19
ГЛАВА 3. ФИЗИЧЕСКОЕ ПРОЕКТИРОВАНИЕ РЕЛЯЦИОННОЙ БАЗЫ ДАННЫХ	20
3.1 Создание таблиц	20
3.2 Создание представлений	21
3.3 Назначение прав доступа	24
3.4 Создание индексов.....	24
3.5 Разработка стратегии резервного копирования	24
ЗАКЛЮЧЕНИЕ	25
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ И ИНТЕРНЕТ – РЕСУРСОВ	26
ПРИЛОЖЕНИЕ	28

Пример оформления таблиц и рисунков

Таблица 3.1 - Схема отношения ОТДЕЛЫ (Departs)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечания</i>
Аббревиатура отдела	D_ID	C(10)	первичный ключ
Название отдела	D_NAME	V(100)	обязательное поле
Комнаты	D_ROOMS	V(20)	обязательное многозначное поле
Телефоны	D_PHONE	V(40)	обязательное многозначное поле

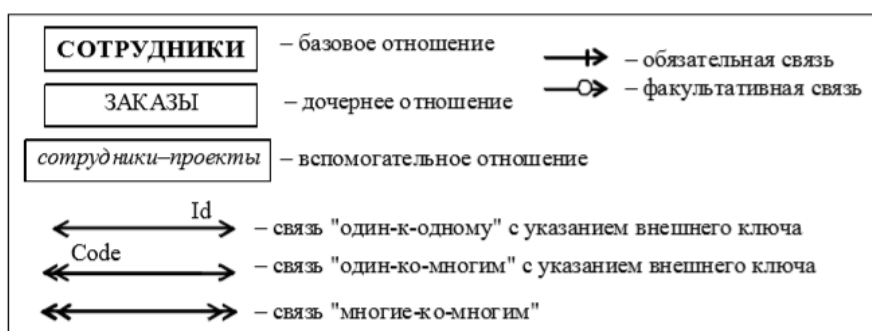


Рис. 3.1. Обозначения, используемые на схеме базы данных

Пример проектирования базы данных

В качестве примера возьмем базу данных проектной организации. Основной вид деятельности такой организации – выполнение проектов по договорам с заказчиками.

ГЛАВА 1. ИНФОЛОГИЧЕСКОЕ ПРОЕКТИРОВАНИЕ

1.1. Анализ предметной области

База данных создаётся для информационного обслуживания руководства организации, руководителей проектов и участников проектов. БД должна содержать данные об отделах организации, сотрудниках и проектах. В соответствии с предметной областью система строится с учётом следующих особенностей:

- Каждый сотрудник работает в определённом отделе, в каждом отделе могут работать несколько сотрудников.

- Каждый проект относится к определённому отделу, каждый отдел может отвечать за выполнение нескольких проектов.

- Каждый сотрудник может принимать участие в выполнении нескольких проектов, над каждым проектом может трудиться несколько сотрудников.

- Для каждого проекта назначается руководитель из числа сотрудников того отдела, к которому относится проект.

- Каждый проект должен быть выполнен в заданные сроки, каждый проект может состоять из нескольких этапов. Если проект состоит из одного этапа, то сроки его выполнения должны совпадать со сроками выполнения проекта в целом.

- Оклад сотрудника зависит от занимаемой должности, за участие в проектах сотрудник получает дополнительное вознаграждение.

- Виды участия сотрудников в проектах: руководитель, консультант, исполнитель.

- Каждый отдел занимает одно или несколько помещений (комнат), в каждом помещении может быть один или несколько стационарных телефонов.

Примечание: Описания особенностей ПрО должно быть достаточно для того, чтобы создать ER–диаграмму.

Сущности ПрО

Для создания ER-модели необходимо выделить сущности предметной области:

1) **Отделы.** Атрибуты: название, аббревиатура, комнаты, телефоны.

2) **Сотрудники.** Атрибуты: ФИО, паспортные данные, дата рождения, пол, ИНН (индивидуальный номер налогоплательщика), номер пенсионного страхового свидетельства, адреса, телефоны (рабочий, домашний, мобильный), данные об образовании (вид образования (высшее, среднеспециальное и т.д.), специальность, номер диплома, дата окончания учебного заведения), должность, оклад, логин (имя пользователя).

Примечания: 1. Логин потребуется нам для назначения дифференцированных прав доступа. 2. В нашем задании не предусмотрена полная информационная поддержка сотрудников отдела кадров, поэтому мы не будем отражать в БД такие сведения как дату поступления сотрудника на работу, его переводы с одной должности на другую, уходы в отпуска и т.п.

3) **Проекты.** Атрибуты: номер договора; полное название проекта; сокращённое название проекта; дата подписания договора; заказчик; контактные данные заказчика; дата начала проекта; дата завершения проекта; сумма по проекту; дата реальной сдачи проекта; сумма, полученная по проекту на текущую дату.

4) **Этапы проекта.** Атрибуты: номер по порядку, название, дата начала этапа, дата завершения этапа, форма отчетности, сумма по этапу, дата реальной сдачи этапа; сумма, полученная по этапу на текущую дату.

Исходя из выявленных сущностей, построим ER–диаграмму (рис. 1). Напомним, что пометки у линий означают степень связи: 1:1, 1:N и N:M.

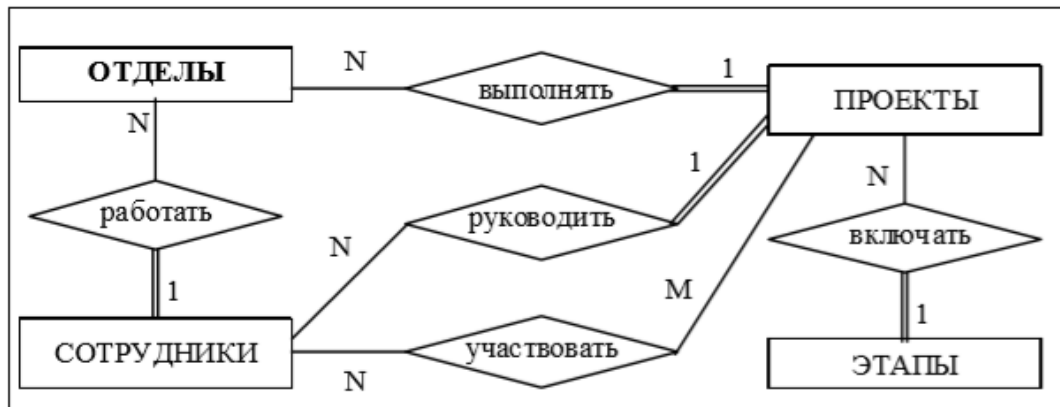


Рис. 1. ER–диаграмма ПрО «Проектная организация»

1.2. Анализ информационных задач и круга пользователей системы

Определим группы пользователей, их основные задачи и запросы к БД:

1. Руководители организации:

- заключение новых договоров;
- назначение руководителей проектов;
- получение списка всех участников проектов;
- изменение должностных окладов и штатного расписания;
- получение полной информации о проектах;
- внесение изменений в данные о проектах;
- архивирование данных по завершённым проектам.

2. Руководитель проекта:

- назначение участников проекта;
- получение списка сотрудников, работающих над конкретным проектом;
- получение полной информации о проекте, руководителем которого он является;
- получение сведений о сотрудниках, которые могут стать участниками проекта;
- определение размера дополнительного вознаграждения сотрудников по конкретному проекту;
- внесение изменений в данные об этапах проекта.

3. Сотрудники отдела кадров:

- приём/увольнение сотрудников;
- внесение изменений в данные о сотрудниках.

4. Бухгалтеры:

- получение ведомости на выплату зарплаты.

5. Сотрудники – участники проектов:

- просмотр данных о других участниках проекта;
- просмотр данных о сроках сдачи проекта и форме отчётности.

1.3. Определение требований к операционной обстановке

Для выполнения этого этапа необходимо знать (хотя бы ориентировочно) объём работы организации (т.е. количество проектов и сотрудников), а также иметь представление о характере и интенсивности запросов.

Объём внешней памяти, необходимый для функционирования системы, складывается из двух составляющих: память, занимаемая модулями СУБД (ядро, утилиты, вспомогательные программы), и память, отводимая под данные (МД). Для реальных баз данных обычно наиболее существенным является МД.

На основе результатов анализа ПрО можно приблизительно оценить объём памяти, требуемой для хранения данных. Примем ориентировочно, что:

- одновременно осуществляется около десяти проектов, работа над проектом продолжается в среднем год (по 1К на каждый проект);
- каждый проект состоит в среднем из четырёх этапов (по 0,5К на этап);
- в компании работают 100 сотрудников (по 0,5К на каждого сотрудника);
- в выполнении каждого проекта в среднем участвуют 10 сотрудников (по 0,2К);
- устаревшие данные переводятся в архив (накапливаются в архиве БД).

Тогда объём памяти для хранения данных за первый год примерно составит:

$$M_d = 2(10 \cdot 1 + 10 \cdot 4 \cdot 0,5 + 100 \cdot 0,5 + (10 \cdot 10 \cdot 0,2)) = 200 \text{ К},$$

Коэффициент 2 необходим для того, чтобы учесть необходимость выделения памяти под дополнительные структуры (например, индексы). Объём памяти будет увеличиваться ежегодно на столько же при сохранении объёма работы.

Требуемый объём оперативной памяти определяется на основании анализа интенсивности запросов и объёма результирующих данных. Для нашей БД требуемый объём памяти мал, поэтому никаких специальных требований к объёму внешней и оперативной памяти компьютера не предъявляется.

ГЛАВА 2. ЛОГИЧЕСКОЕ ПРОЕКТИРОВАНИЕ РЕЛЯЦИОННОЙ БАЗЫ ДАННЫХ

2.1. Преобразование ER–диаграммы в схему базы данных

База данных создаётся на основании схемы базы данных. Для преобразования ER–диаграммы в схему БД приведём уточнённую ER–диаграмму, содержащую атрибуты сущностей (рис. 2).

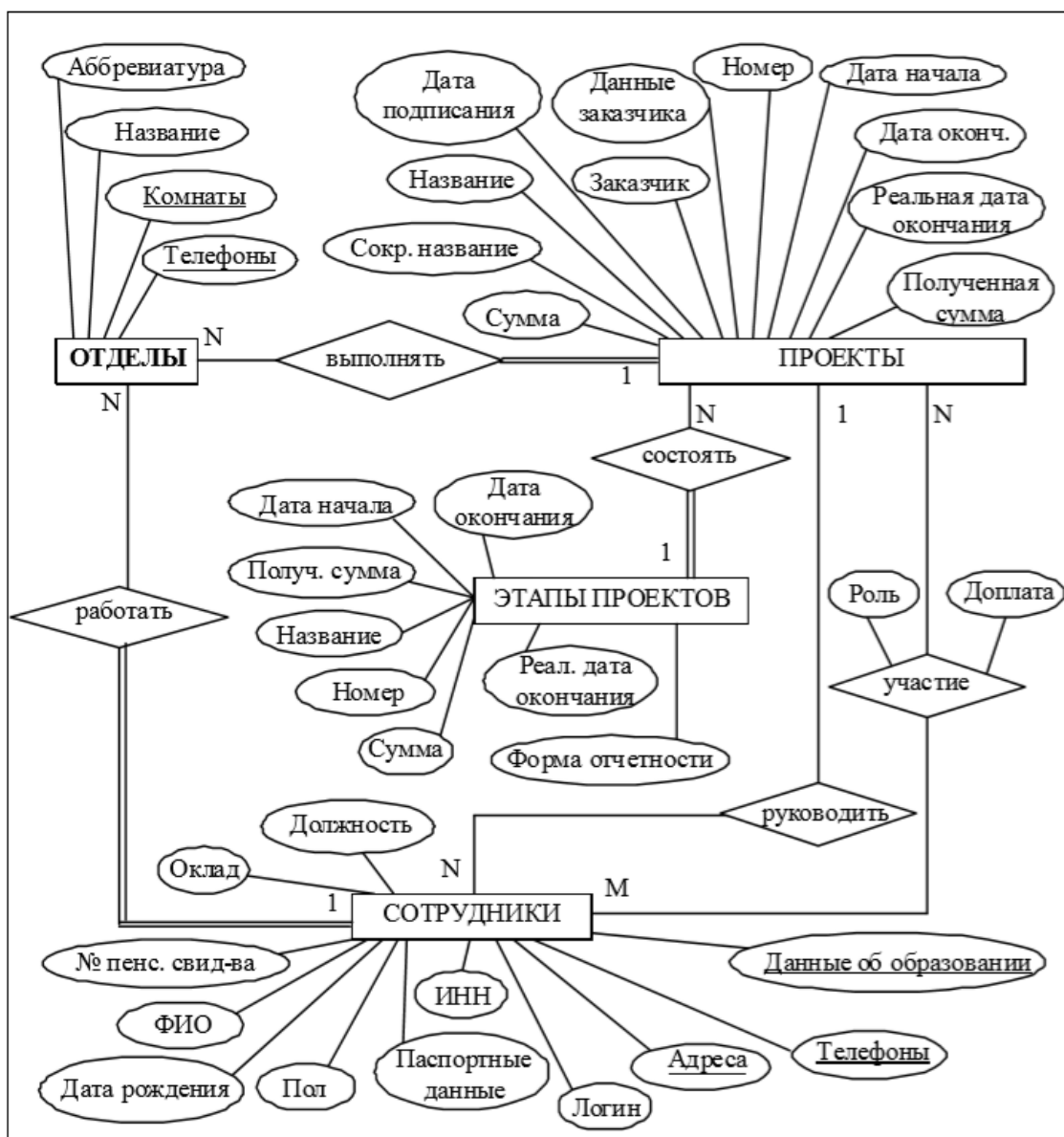


Рис. 2. Уточнённая ER–диаграмма проектной организации

Примечание. Многочисленные атрибуты на рисунке выделены подчеркиванием.

Преобразование ER–диаграммы в схему БД выполняется путем сопоставления каждой сущности и каждой связи, имеющей атрибуты, отношения (таблицы) БД. Связь типа 1:n (один-ко-многим) между отношениями реализуется через внешний ключ. Ключ вводится для того отношения, к которому осуществляется множественная связь. Внешнему ключу должен соответствовать первичный или уникальный ключ основного (родительского) отношения.

Связь *участвовать* между ПРОЕКТАМИ и СОТРУДНИКАМИ принадлежит к типу n:m (многие-ко-многим). Этот тип связи реализуется через вспомогательное отношение *Участие*, которое содержит комбинации первичных ключей соответствующих исходных отношений.

Для схемы БД будем использовать обозначения, представленные на рис. 3.

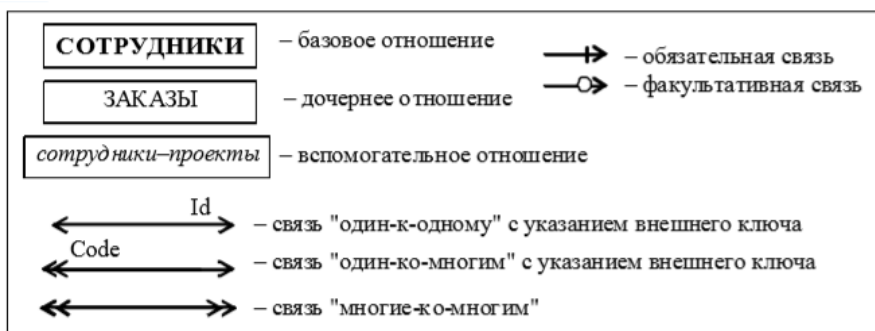


Рис. 3. Обозначения, используемые на схеме базы данных

Полученная схема реляционной базы данных (РБД) приведена на рис. 4.

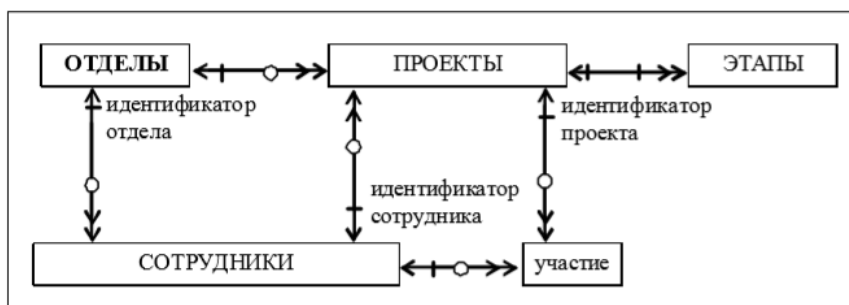


Рис. 4. Схема РБД, полученная из ER-диаграммы проектной организации

Бинарная связь между отношениями не может быть обязательной для обоих отношений. Такой тип связи означает, что, например, прежде чем добавить новый проект в отношение *ПРОЕКТЫ*, нужно добавить новую строку в отношение *ЭТАПЫ*, и наоборот. Поэтому для такой связи необходимо снять с одной стороны условие обязательности. Так как все эти связи будут реализованы с помощью внешнего ключа, снимем условие обязательности связей для отношений, содержащих первичные ключи.

Схема на рисунке 4 содержит три цикла:

- «сотрудники–проекты–участие–сотрудники»;
- «отделы–сотрудники–проекты–отделы»;
- «отделы–сотрудники–участие–проекты–отделы».

Цикл допустим только в том случае, если связи, входящие в него, независимы друг от друга. Например, для нашей ПрО справедливо такое правило: сотрудник любого отдела может быть участником (исполнителем или консультантом) проекта любого отдела. Эти связи независимы, поэтому цикл «отделы–сотрудники–участие–проекты–отделы» не будет приводить к нарушению логической целостности данных.

С другой стороны, только сотрудник отдела, отвечающего за выполнение проекта, может быть руководителем проекта. Но система не помешает нам назначить руководителем проекта сотрудника любого отдела. При добавлении проекта с внешним ключом *Руководитель* система проверит только, что такой человек есть в таблице *СОТРУДНИКИ*. А значение внешних ключей *Отдел* в таблицах *СОТРУДНИКИ* и *ПРОЕКТЫ* сравнивать не будет.

Таким образом, остальные циклы могут приводить к возможности нарушения логической целостности данных. Существует несколько подходов для разрешения ситуаций, в которых связи, входящие в цикл, зависят друг от друга.

Рассмотрим эту ситуацию в общем случае. Сначала слегка упростим схему: реализуем связь «руководить» через таблицу УЧАСТИЕ – это позволит не отвлекаться на малозначительные детали.

Будем считать, что в выполнении проекта могут участвовать только сотрудники, работающие в том же отделе, к которому относится проект (рис. 5,а). При циклической схеме СУБД не сможет гарантировать логическую целостность данных без использования дополнительных средств.

Один из способов разрешения таких ситуаций – разорвать цикл, исключив одну из связей (рис. 5,б) или введя промежуточное отношение (рис. 5,в). В нашем случае можно было бы разорвать связь «сотрудники–проекты», если бы каждый сотрудник участвовал во всех проектах своего отдела. Промежуточное отношение можно было бы использовать, если бы существовала общая связь между сущностями, входящими в цикл. Например, если бы каждый сотрудник заключал договор с отделом на выполнение работ в рамках проекта, то отношение ДОГОВОРЫ отражало бы связь между отделом, сотрудником и проектом.

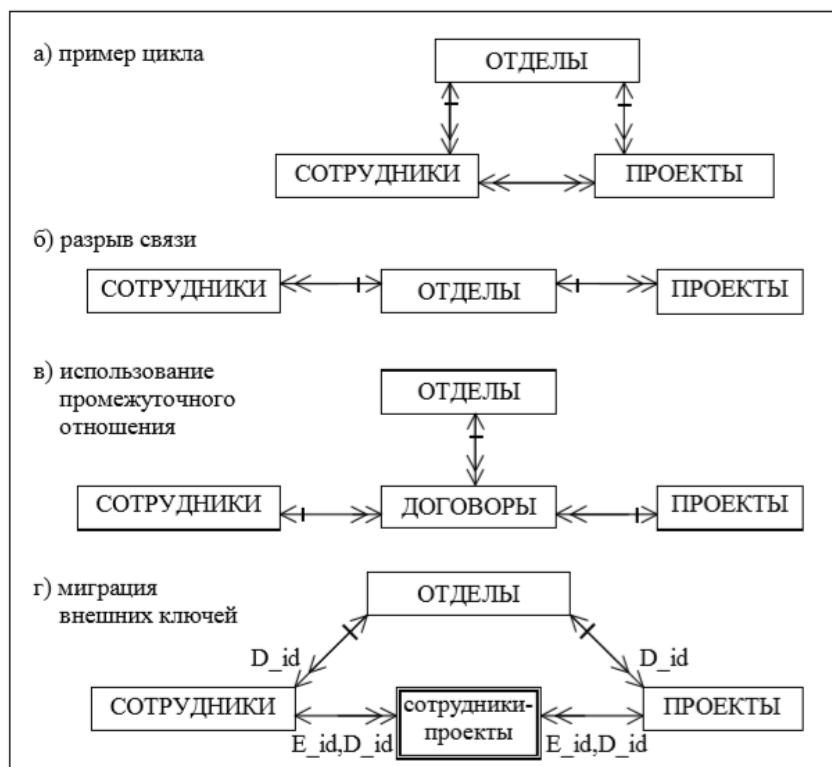


Рис.5. Некоторые способы разрешения циклов в схеме базы данных

Другой способ разрешения цикла заключается в том, что в промежуточное отношение СОТРУДНИКИ – ПРОЕКТЫ, которое реализует связь многие-многим, добавляются (мигрируют) внешние ключи Код отдела (D_id) из отношений СОТРУДНИКИ и ПРОЕКТЫ (рис. 5,г). Эти ключи проверяются на равенство друг другу с помощью соответствующего ограничения целостности (check). Использование этого способа возможно в том случае, когда соответствующие связи (отдел–проект и отдел–сотрудник) имеют тип один-ко-многим и являются обязательными.

В тех ситуациях, когда все эти способы непригодны, логическая целостность контролируется программно или вручную. Если принято решение переложить обязанности по контролю за логической целостностью данных на пользователя, то эти обязанности должны быть отражены в документации (в руководстве пользователя).

Примем для нашей ПрО, что руководитель проекта может одновременно выполнять и другие обязанности в этом проекте, чтобы цикл «сотрудники– проекты– участие–сотрудники» не приводил к возможности нарушения логической целостности данных. Зато цикл «отделы–сотрудники (руководители)–проекты–отделы» включает зависимые связи: руководитель проекта назначается из того отдела, который отвечает за выполнение проекта в целом. Здесь можно было бы применить разрыв связи «отделы–проекты» и определять, к какому отделу относится проект через руководителя (по отделу руководителя проекта). Но такой подход в данном случае имеет существенный недостаток. Заменяв руководителя проекта сотрудником другого отдела, можно одновременно изменить отдел, отвечающий за выполнение проекта, т.е. объединить в одно действие два независимых изменения, а это недопустимо.

Исходя из вышесказанного, мы не будем разрывать связь, а примем решение реализовать эту проверку программно. Приложение должно будет при назначении руководителя проекта выдавать список сотрудников того отдела, который отвечает за выполнение данного проекта. Руководителя можно будет выбрать только из этого списка, а не вводить вручную.

2.2. Составление реляционных отношений

Каждое реляционное отношение соответствует одной сущности (объекту ПрО) и в него вносятся все атрибуты этой сущности. Для каждого отношения определяются первичный ключ и внешние ключи (в соответствии со схемой БД). В том случае, если базовое отношение не имеет потенциальных ключей, вводится *суррогатный первичный ключ*, который не несёт смысловой нагрузки и служит только для идентификации записей.

Отношения приведены в табл. 1-5. Для каждого отношения указаны атрибуты с их внутренним названием, типом и длиной. Типы данных обозначаются так: N – числовой, С – символьный тип фиксированной длины, V – символьный тип переменной длины, D – дата (этот тип имеет стандартную длину, зависящую от СУБД, поэтому она не указывается).

Потенциальными ключами отношения ОТДЕЛЫ являются атрибуты Аббревиатура и Название отдела. Первый занимает меньше места, поэтому мы выбираем его в качестве первичного ключа.

Таблица 1. Схема отношения ОТДЕЛЫ (Departs)

Содержание поля	Имя поля	Тип, длина	Примечания
Аббревиатура отдела	D_ID	C(10)	первичный ключ
Название отдела	D_NAME	V(100)	обязательное поле
Комнаты	D_ROOMS	V(20)	обязательное многозначное поле
Телефоны	D_PHONE	V(40)	обязательное многозначное поле

Потенциальными ключами отношения СОТРУДНИКИ являются поля Паспортные данные, ИНН и Номер страхового пенсионного свидетельства. Все они занимают достаточно много места, а паспортные данные кроме того могут меняться. Введём суррогатный первичный ключ Номер сотрудника.

Таблица 2. Схема отношения СОТРУДНИКИ (Employees)

Содержание поля	Имя поля	Тип, длина	Примечания
Номер	E_ID	N(4)	суррогатный первичный ключ

Фамилия, имя, отчество	E_NAME	V(50)	обязательное поле
Дата рождения	E_BORN	D	обязательное поле
Пол	E_SEX	C(1)	обязательное поле, 'м' или 'ж'
Паспортные данные	E_PASP	V(50)	обязательное поле
ИНН	E_INN	C(12)	обязательное поле
Номер пенсионного страхового свидетельства	E_PENS	C(14)	обязательное уникальное поле
Отдел	E_DEPART	C(10)	обязательное уникальное поле
Должность	E_POST	V(30)	обязательное поле
Оклад	E_SAL	N(8,2)	обязательное поле, > 12130 руб.
Данные об образовании	E_EDU	V(200)	обязательное многозначное поле
Адреса	E_ADDR	V(100)	многозначное поле
Телефоны	E_PHONE	V(30)	многозначное поле
Логин	E_LOGIN	V(30)	

Примечание. Суррогатный первичный ключ также может вводиться в тех случаях, когда потенциальный ключ имеет большой размер (например, длинная символьная строка) или является составным (не менее трёх атрибутов).

В отношении ПРОЕКТЫ три потенциальных ключа: *Номер проекта*, *Название проекта* и *Сокращённое название*. Меньше места занимает первый из них, но он малоинформативен. Зато сокращённое название, используемое в качестве внешнего ключа в других таблицах, позволит специалисту идентифицировать проект без необходимости соединения с отношением ПРОЕКТЫ.

Таблица 3. Схема отношения ПРОЕКТЫ (Projects)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечания</i>
Номер проекта	P_ID	N(6)	обязательное уникальное поле
Название проекта	P_TITLE	V(100)	обязательное поле
Сокращённое название	P_ABBR	C(10)	первичный ключ
Отдел	P_DEPART	C(10)	внешний ключ (к Departs)
Заказчик	P_COMPANY	V(40)	обязательное поле
Данные заказчика	P_LINKS	V(200)	обязательное поле
Руководитель	P_CHIEF	N(4)	внешний ключ (к Employees)
Дата начала проекта	P_BEGIN	D	обязательное поле
Дата окончания проекта	P_END	D	обязательное поле, больше даты начала проекта
Реальная дата окончания	P_FINISH	D	
Стоимость проекта	P_COST	N(10)	обязательное поле
Полученная сумма	P_SUM	N(10)	обязательное поле, значение по умолчанию – 0

Потенциальным ключом отношения ЭТАПЫ является комбинация внешнего ключа и номера этапа, а потенциальным ключом вспомогательного отношения УЧАСТИЕ является комбинация первых трёх полей этого отношения. Можно вообще не вводить первичный ключ для данных отношений, т.к. на них никто не ссылается. Но уникальность этих комбинации является в данном случае ограничением целостности данных, поэтому мы возьмём эти комбинации в качестве первичных ключей соответствующих отношений.

Таблица 4. Схема отношения ЭТАПЫ ПРОЕКТА (Stages)

Содержание поля	Имя поля	Тип, длина	Примечания	
Проект	S_PRO	C(10)	внешний ключ (к Projects)	составной первичный ключ
Номер этапа	S_NUM	N(2)		
Название этапа	S_TITLE	V(200)	обязательное поле	
Дата начала этапа	S_BEGIN	D	обязательное поле	
Дата окончания этапа	S_END	D	обязательное поле, > даты начала	
Реальная дата окончания	S_FINISH	D	больше даты начала этапа	
Стоимость этапа	S_COST	N(10)	обязательное поле	
Полученная сумма по этапу	S_SUM	N(10)	обязательное поле, значение по умолчанию – 0	
Форма отчётности	S_FORM	V(100)	обязательное поле	

Таблица 5. Схема отношения УЧАСТИЕ (Job)

Содержание поля	Имя поля	Тип, длина	Примечания*
Проект	J_PRO	C(10)	внешний ключ (к Projects)
Сотрудник	J_EMP	N(4)	внешний ключ (к Employees)
Роль	J_ROLE	V(20)	обязательное поле
Доплата	J_BONUS	N(2)	

* – в отношении УЧАСТИЕ первичный ключ состоит из первых 3-х полей этого отношения.

2.3. Нормализация полученных отношений (до 4НФ)

Механизм нормализации подразумевает определённую последовательность преобразования отношений к третьей нормальной форме. Мы не будем чётко придерживаться этой последовательности, т.к. она избыточна, и многозначные атрибуты сразу вынесем в отдельные отношения на первом же этапе. **1НФ**. Для приведения таблиц к 1НФ требуется составить прямоугольные таблицы (одно значение атрибута – одна ячейка таблицы) и разбить сложные атрибуты на простые.

Примечание. В реальных БД сложные атрибуты разбиваются на простые, если: а) этого требует внешнее представление данных; б) в запросах поиск может осуществляться по отдельной части атрибута.

Разделим атрибут Фамилия, имя, отчество на два атрибута Фамилия и Имя, отчество, Паспортные данные на Номер паспорта (уникальный), Дата выдачи и Кем выдан, а Данные об образовании – на Вид образования, Специальность, Номер диплома и Год окончания учебного заведения.

Многозначные атрибуты Комнаты и Телефоны из отношения ОТДЕЛЫ вынесем в отдельное отношение КОМНАТЫ, а домашние и мобильные телефоны и адреса сотрудников – в отношение АДРЕСА-ТЕЛЕФОНЫ. Так как в комнате может не быть телефона, первичный ключ отношения КОМНАТЫ не определен (ПК не может содержать null-значения), но на этих атрибутах можно определить составной уникальный ключ. В отношении АДРЕСА-ТЕЛЕФОНЫ также нет потенциальных ключей: оставим это отношение без первичного ключа, т.к. на это отношение никто не ссылается. Данные об образовании сотрудников также вынесем в отдельное отношение.

Что касается рабочих телефонов сотрудников, то один из этих номеров – основной – определяется рабочим местом сотрудника (рассматриваются только стационарные телефоны). Будем хранить этот номер в атрибуте Рабочий телефон. Наличие других номеров зависит от того, есть ли в том же помещении (комнате) другие сотрудники, имеющие стационарные телефоны. Добавим в отношение СОТРУДНИКИ атрибут Номер комнаты, чтобы дополнительные номера телефонов сотрудника можно было вычислить из других кортежей с таким же номером комнаты.

Связь между отношениями СОТРУДНИКИ и КОМНАТЫ реализуем через составной внешний ключ (Номер комнаты, Рабочий телефон).

Мы также удалим вычисляемый атрибут Полученная сумма из отношения ПРОЕКТЫ, т.к. он является суммой значений аналогичного атрибута из отношения ЭТАПЫ ПРОЕКТОВ. Но атрибут Стоимость проекта оставим, т.к. она фигурирует в документации по проекту. А для обеспечения логической целостности данных предусмотрим в приложении проверку того, что сумма по всем этапам совпадает со стоимостью проекта.

2НФ. В нашем случае составные первичные ключи имеют отношения ЭТАПЫ ПРОЕКТА и УЧАСТИЕ. Неключевые атрибуты этих отношений функционально полно зависят от составных первичных ключей.

3НФ. В отношении ПРОЕКТЫ атрибут Данные заказчика зависит от атрибута Заказчик, а не от первичного ключа, поэтому его следует вынести в отдельное отношение ЗАКАЗЧИКИ. Но при этом первичным ключом нового отношения станет атрибут Заказчик, т.е. длинная символьная строка. Целесообразнее перенести в новое отношение атрибуты Заказчик и Данные заказчика и ввести для него суррогатный ПК. Так как с каждым заказчиком может быть связано несколько проектов, связь между отношениями ПРОЕКТЫ и ЗАКАЗЧИКИ будет 1:n и суррогатный ПК станет внешним ключом для отношения ПРОЕКТЫ.

В отношении СОТРУДНИКИ атрибут Оклад зависит от атрибута Должность. Поступим с этой транзитивной зависимостью так же, как в предыдущем случае: создадим отношение ДОЛЖНОСТИ, перенесём в него атрибуты Должность и Оклад, а первичным ключом сделаем название должности. В отношениях СОТРУДНИКИ и ОБРАЗОВАНИЕ атрибуты (Дата выдачи и Кем выдан) и (Номер диплома и Год окончания учебного заведения) зависят не от первичного ключа, а от атрибутов соответственно Номер паспорта и Специальность. Но если мы выделим их в отдельное отношение, то получим связи типа 1:1. Следовательно, здесь декомпозиция нецелесообразна.

4НФ. Отношение АДРЕСА-ТЕЛЕФОНЫ нарушают 4НФ, т.к. не всякий телефон привязан к конкретному адресу (т.е. мы имеем две многозначных зависимости в одном отношении). Но выделять Телефоны в отдельное отношение не стоит, т.к. эти сведения носят справочный характер и не требуется их автоматическая обработка.

Отношения, полученные после нормализации, приведены в табл. 5-14.

Таблица 5. Схема отношения ОТДЕЛЫ (Departs)

Содержание поля	Имя поля	Тип, длина	Примечания
Аббревиатура отдела	D_ID	C(10)	первичный ключ
Название отдела	D_NAME	V(100)	обязательное поле

Таблица 6. Схема отношения КОМНАТЫ (Rooms)

Содержание поля	Имя поля	Тип, длина	Примечания
Отдел	R_DEPART	V(12)	внешний ключ (к Departs)
Номер комнаты	R_ROOM	N(4)	составной уникальный ключ
Телефоны	R_PHONE	V(20)	

Таблица 7. Схема отношения ДОЛЖНОСТИ (Posts)

Содержание поля	Имя поля	Тип, длина	Примечания
Название должности	P_POST	V(30)	первичный ключ
Оклад	P_SAL	N(8,2)	обязательное поле, > 12130 руб.

Таблица 8. Схема отношения СОТРУДНИКИ (Employees)

Содержание поля	Имя поля	Тип, длина	Примечания
Идентификатор сотрудника	E_ID	N(4)	суррогатный первичный ключ
Фамилия	E_NAME	V(25)	обязательное поле
Имя, отчество	E_LNAME	V(30)	обязательное поле
Дата рождения	E_BORN	D	обязательное поле
Пол	E_SEX	C(1)	обязательное поле
Серия и номер паспорта	E_PASP	C(10)	обязательное уникальное поле
Когда выдан паспорт	E_DATE	D	обязательное поле
Кем выдан паспорт	E_PASP	V(50)	обязательное поле
ИНН	E_INN	C(12)	обязательное поле
Номер пенсионного страхового свидетельства	E_PENS	C(14)	обязательное уникальное поле
Отдел	E_DEPART	V(12)	внешний ключ (к Departs)
Должность	E_POST	V(30)	внешний ключ (к Posts)
Номер комнаты	E_ROOM	N(4)	составной внешний ключ (к Rooms)
Рабочий телефоны	E_PHONE	V(20)	
Логин	E_LOGIN	V(30)	

Таблица 9. Схема отношения ОБРАЗОВАНИЕ (Edu)

Содержание поля	Имя поля	Тип, длина	Примечания
Идентификатор сотрудника	U_ID	N(4)	внешний ключ (к Employees)
Вид образования	U_TYPE	V(20)	обязательное поле
Специальность	U_SPEC	V(40)	
Номер диплома	U_DIPLOM	V(15)	
Год окончания учебного заведения	U_YEAR	N(4)	обязательное поле

Таблица 10. Схема отношения АДРЕСА-ТЕЛЕФОНЫ (AdrTel)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечания</i>
Идентификатор сотрудника	A_ID	N(4)	внешний ключ (к Employees)
Адрес	A_ADDR	V(50)	обязательное поле
Телефон	A_PHONE	V(30)	

Таблицы ОБРАЗОВАНИЕ и АДРЕСА-ТЕЛЕФОНЫ не имеют потенциальных ключей, но мы не будем вводить суррогатные первичные ключи, т.к. на эти таблицы никто не ссылается.

Таблица 11. Схема отношения ЗАКАЗЧИКИ (Clients)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечания</i>
Номер заказчика	C_ID	N(4)	суррогатный первичный ключ
Заказчик	C_COMPANY	V(40)	обязательное поле
Адрес заказчика	C_ADR	V(50)	обязательное поле
Контактное лицо	C_PERSON	V(50)	обязательное поле
Телефон	C_PHONE	V(30)	

Таблица 12. Схема отношения ПРОЕКТЫ (Projects)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечания</i>
Номер проекта	P_ID	N(6)	обязательное уникальное поле
Название проекта	P_TITLE	V(100)	обязательное поле
Сокращённое название	P_ABBR	C(10)	первичный ключ
Отдел	P_DEPART	V(12)	внешний ключ (к Departs)
Заказчик	P_COMPANY	N(4)	внешний ключ (к Clients)
Руководитель	P_CHIEF	N(4)	внешний ключ (к Employees)
Дата начала проекта	P_BEGIN	D	обязательное поле
Дата окончания проекта	P_END	D	обязательное поле, больше даты начала проекта
Реальная дата окончания	P_FINISH	D	больше даты начала проекта
Стоимость проекта	P_COST	N(10)	обязательное поле, > 0

Таблица 13. Схема отношения ЭТАПЫ ПРОЕКТА (Stages)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечания</i>
Проект	S_PRO	C(10)	внешний ключ (к Projects) составной первичный ключ
Номер этапа	S_NUM	N(2)	
Название этапа	S_TITLE	V(200)	обязательное поле
Дата начала этапа	S_BEGIN	D	обязательное поле

Дата окончания этапа	S_END	D	обязательное поле, больше даты начала
Реальная дата окончания	S_FINISH	D	больше даты начала этапа
Стоимость этапа	S_COST	N(10)	обязательное поле
Полученная сумма по этапу	S_SUM	N(10)	обязательное поле, значение по умолчанию – 0
Форма отчётности	S_FORM	V(100)	обязательное поле

Таблица 14. Схема отношения УЧАСТИЕ (Job)

Содержание поля	Имя поля	Тип, длина	Примечания*
Проект	J_PRO	C(10)	внешний ключ (к Projects) составной первичный ключ
Сотрудник	J_EMP	N(4)	
Роль	J_ROLE	V(20)	обязательное поле
Доплата	J_BONUS	N(2)	

Схема базы данных после нормализации приведена на рис. 6.

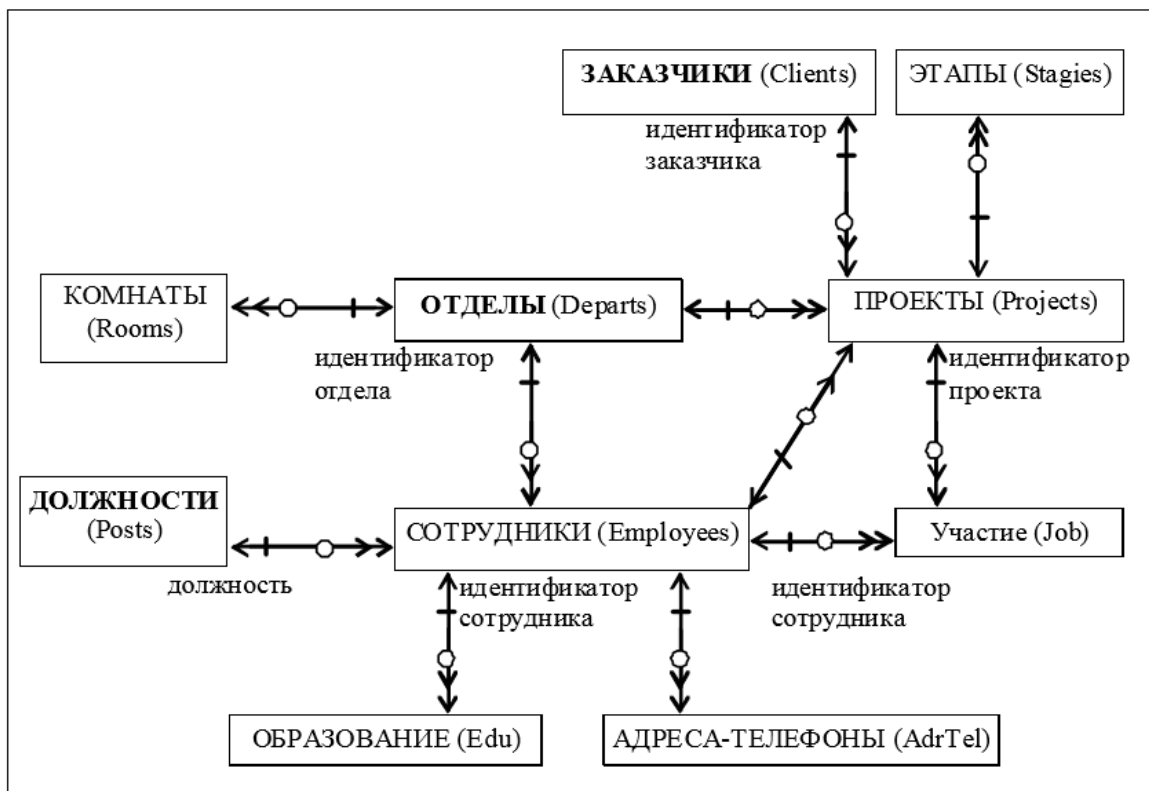


Рис. 6. Окончательная схема БД проектной организации

2.4. Определение дополнительных ограничений целостности

Перечислим ограничения целостности, которые не указаны в табл. 1–10.

1. Атрибут Вид образования может принимать одно из следующих значений: 'начальное', 'среднее', 'средне-специальное', 'высшее'.
2. Атрибут Роль может принимать одно из двух значений: 'исполнитель' или 'консультант'.

3. В поле Доплата хранится величина доплаты сотруднику за участие в проекте (в процентах к его окладу). Значение поля больше либо равно 0.

4. Нумерация в поле Номер этапа начинается с 1 и является непрерывной для каждого проекта.

5. Дата начала первого этапа проекта должна соответствовать началу проекта в целом, дата завершения последнего этапа должна соответствовать завершению проекта в целом. Этапы не должны пересекаться по времени и между ними не должно быть разрывов.

6. Стоимость проекта должна быть равна сумме стоимостей всех этапов этого проекта. Ограничения 4-6 нельзя реализовать в схеме отношения. В реальных БД подобные ограничения целостности реализуются вручную или программно (через внешнее приложение или специальную процедуру контроля данных – триггер).

Примечание. Вопросы архивирования данных в этом пособии подробно не рассматриваются. Но следует отметить, что обычно архив является частью БД и представляет собой набор отдельных таблиц, которые не связаны с оперативной частью БД внешними ключами. Структура архивных таблиц либо соответствует структуре тех оперативных таблиц, данные которых подлежат архивированию, либо представляет собой денормализованную таблицу, соответствующую декартову произведению оперативных таблиц. Данные в архивные таблицы переносятся специальной программой (или набором запросов) автоматически или по команде пользователя. По истечении периода хранения данные могут удаляться из архива.

2.5. Описание групп пользователей и прав доступа

Опишем для каждой группы пользователей права доступа к каждой таблице. Права доступа должны быть распределены так, чтобы для каждого объекта БД был хотя бы один пользователь, который имеет право добавлять и удалять данные из объекта. Права приведены в табл. 11. Используются следующие сокращения:

- s – чтение данных (select);
- i – добавление данных (insert);
- u – модификация данных (update);
- d – удаление данных(delete).

Таблица 15. Права доступа к таблицам для групп пользователей

Таблицы	Группы пользователей (роли)				
	Руководител и организации	Сотрудник и отд. кадров	Руководител и проектов	Бухгалтер ы	Участник и проектов
Отделы	S	SIUD	S	S	
Комнаты	S	SUID	S	S	S
Должности	SIUD			S	
Сотрудники	S	SUID	S	S	
Адреса- телефоны	S	SUID	S	S	
Образовани е	S	SUID	S	S	
Заказчики	SIUD		S		
Проекты	SIUD		S		
Этапы проектов	SIUD		SUI		
Участие	S		S	S	

Права на изменение данных в таблице УЧАСТИЕ будут назначены через представление, т.к. изменять данные этой таблицы может только руководитель проекта.

Права назначает администратор БД (или администратор безопасности, если система сложная и администраторов несколько).

Приведём описание схемы БД на DDL.

ГЛАВА 3. ФИЗИЧЕСКОЕ ПРОЕКТИРОВАНИЕ РЕЛЯЦИОННОЙ БАЗЫ ДАННЫХ

3.1. Создание таблиц

1. Отношение **Departs** (отделы):

```
create table departs (  
  d_id varchar(12) primary key,  
  d_name varchar(100) not null);
```

2. Отношение **Rooms** (комнаты):

```
create table rooms (  
  d_depart varchar(12)  
  references departs(d_id),  
  r_room numeric(4) not null,  
  r_phone varchar(20),  
  unique(r_room, r_phone));
```

3. Отношение **Posts** (должности):

```
create table posts (  
  p_post varchar(30) primary key,  
  p_salary numeric(8,2) not null check(p_salary>=12130));
```

4. Отношение **Employees** (сотрудники):

```
create table employees (  
  e_id numeric(4) primary key,  
  e_fname varchar(25) not null,  
  e_lname varchar(30) not null,  
  e_born date not null,  
  e_sex char(1) check(e_sex in ('ж', 'м')),  
  e_pasp char(10) not null unique,  
  e_date date not null,  
  e_given varchar(50) not null,  
  e_inn char(12) not null unique,  
  e_pens char(14) not null unique,  
  e_depart varchar(12) references departs,  
  e_post varchar(30) references posts,  
  e_room numeric(4) not null,  
  e_phone varchar(20) not null,  
  e_login varchar(30),  
  foreign key(e_room, e_phone)  
  references rooms(r_room, r_phone));
```

(Если внешний ключ ссылается на первичный ключ отношения, его можно не указывать, как в случае ссылок на **Departs** и **Posts**).

5. Отношение Edu (образование):

```
create table edu (  
  u_id numeric(4) references employees,  
  u_type varchar(20) not null,  
  u_spec varchar(40),  
  u_diplom varchar(15),  
  u_year float(4) not null,  
  check(u_spec in ('начальное', 'среднее', 'высшее', 'средне-  
специальное')));
```

6. Отношение AdrTel (адреса-телефоны):

```
create table adrtel (  
  a_id numeric(4) references employees,  
  a_adr varchar(50),  
  a_phone varchar(30));
```

7. Отношение Clients (заказчики):

```
create table clients (  
  c_id numeric(4) primary key,  
  c_company varchar(40) not null,  
  c_adr varchar(50) not null,  
  c_person varchar(50) not null,  
  c_phone varchar(30));
```

8. Отношение Projects (проекты):

```
create table projects (  
  p_id numeric(6) not null unique,  
  p_title varchar(100) not null,  
  p_abbr char(10) primary key,  
  p_depart varchar(12) references departs,  
  p_company numeric(4) references clients,  
  p_chief numeric(4) references employees,  
  p_begin date not null,  
  p_end date not null,  
  p_finish date,  
  p_cost numeric(10) not null check(p_cost>0),  
  check (p_end>p_begin),  
  check (p_finish is null or p_finish>p_begin));
```

9. Отношение Stages (этапы проектов):

```
create table stages (  
  s_pro char(10) references projects,  
  s_num numeric(2) not null,  
  s_title varchar(200) not null,  
  s_begin date not null,  
  s_end date not null,  
  s_finish date,  
  s_cost numeric(10) not null,  
  s_sum numeric(10) not null,  
  s_form varchar(100) not null,
```

```

    check (s_cost>0),
    check (s_end>s_begin),
    check (s_finish is null or s_finish>s_begin));

```

10. Отношение Job (участие):

```

create table job (
    j_pro char(10) references projects,
    j_emp numeric(4) references employees,
    j_role varchar(20) not null,
    j_bonus numeric(2) not null,
    check(j_bonus>0),
    check (j_role in ('исполнитель', 'консультант')));

```

3.2. Создание представлений (готовых запросов)

Приведём примеры нескольких готовых запросов (представлений):

1. Список всех текущих проектов (getdate – функция, возвращающая текущую дату, определена в СУБД MySQL; в других системах аналогичная функция может называться по-другому, например, now() в MS Access, sysdate в Oracle и т.д.):

```

create view curr_projects as
select *
    from projects
    where p_begin<=sysdate and sysdate<=p_end;

```

2. Определение суммы по текущим проектам, полученной на текущую дату:

```

create or alter view summ (title, cost, total) as
select p_title, p_cost, sum(s_sum)
    from curr_projects, stages
    where p_abbr=s_pro
    group by p_title, p_cost;

```

3. Данные о проектах для руководителя проектов:

```

create or alter view my_projects as
select *
    from projects p
    where exists (select * from employees e
        where e.e_id=p.p_chief and e.e_login=user);

```

Функция user возвращает имя пользователя, выполняющего текущий запрос. Таким образом, каждый пользователь получит данные только о тех проектах, руководителем которых является. Используя аналогичный способ, можно ограничить участника проекта данными только о сотрудниках тех проектов, в которых он сам участвует.

4. Данные об участниках проектов для руководителя проектов:

```

Create or alter view my_staff as
select j.*
    from job j
    where exists (select *
        from employees e, projects p
        where e.e_id=p.p_chief and e.e_login=user
        and j.j_pro=p.p_abbr);

```

5. Данные о других участниках проекта:

```
create or alter view my_emps as
select je.j_pro, e.e_fname, e.e_lname e_name,
e_depart, e_post, e_phone, e_room
from employees e, job je
where e.e_id=je.j_emp and exists (select *
from job jm, employees m
where m.e_id=jm.j_emp and
m.e_login=user and je.j_pro=jm.j_pro);
```

Для того чтобы можно было работать с этими представлениями, соответствующим пользователям нужно назначить права доступа к представлениям. Эти права перечислены в табл. 16.

Таблица 16. Права доступа к представлениям

Представления	Группы пользователей (роли)		
	Руководители организации	Руководители проектов	Участники проектов
Текущие проекты (curr_projects)	S	S	
Сумма по текущим проектам (summ)	S	S	
Проекты для руководителя (my_projects)		SIUD	
Участники проектов для руководителей (my_staff)		SIUD	
Участники проектов (my_emps)			S

3.3. Назначение прав доступа

Права доступа пользователей предоставляются с помощью команды GRANT. Рассмотрим для примера права сотрудника компании *ok_user*, который является сотрудником отдела кадров. Права доступа к отношениям *Departs* и *Rooms* могут быть описаны следующим образом:

```
grant select, insert, update, delete on departs to ok_user;
grant select, insert, update, delete on rooms to ok_user;
```

Права доступа руководителей проектов (сотрудников, *staff*) к представлению *my_projects* могут быть описаны следующим образом:

```
grant select, insert, update, delete on my_projects to staff;
```

Если сотрудник не является руководителем проекта, он не получит данных через этот запрос и не сможет воспользоваться правами доступа к нему.

Права доступа участников проекта (сотрудников, *staff*) к представлению *my_emps* могут быть описаны следующим образом:

```
grant select on my_emps to staff;
```

Если сотрудник не является участником проекта, он не получит данных через этот запрос и не сможет воспользоваться правами доступа к нему.

3.4. Создание индексов

Анализ готовых запросов показывает, что для повышения эффективности работы с данными необходимо создать индексы для всех внешних ключей. Приведём примеры создания индексов:

```
create index e_posts on employees(e_post);  
create index p_chief on projects(p_chief);  
create index e_tel on employees(e_room, e_phone);
```

3.5. Разработка стратегии резервного копирования

Интенсивность обновления разработанной базы данных низкая, поэтому для обеспечения сохранности вполне достаточно проводить полное резервное копирование БД раз в день (перед окончанием рабочего дня). Для разработанной БД нет необходимости держать сервер включенным круглосуточно, поэтому можно создать соответствующее задание операционной системы, которое будет автоматически запускаться перед выключением сервера.