

ЧОУ ВО Курский институт менеджмента, экономики и бизнеса

Кафедра прикладной информатики и математики

**Методические указания по курсовому проектированию
по дисциплине "Базы данных"**

учебно-методическое пособие для студентов направления подготовки бакалавров 09.03.03 Прикладная информатика, профиль подготовки: Прикладная информатика в экономике

Курск - 2016 г.

УДК 004:36(075.8)

ББК 32.973.26–018.2я73+ 60.9я73

Т92

Автор-составитель - Н.А. Туякбасарова, кандидат социологических наук, доцент кафедры прикладной информатики и математики Курского института менеджмента, экономики и бизнеса.

Рецензент - зав. кафедрой прикладной информатики и математики, канд. физ.-мат. наук, доц. Федоров А.В.

Т92 Туякбасарова Н.А.

Методические указания к курсовому проектированию по курсу "Базы данных": учебно-методическое пособие / Н.А. Туякбасарова; Курск. ин-т менеджмента, экономики и бизнеса. – Курск: ООО «Мечта», 2013. — 29 с.

Настоящее учебно-методическое пособие разработано на основе государственного образовательного стандарта ВПО РФ по направлению подготовки бакалавров 230700.62 Прикладная информатика, профиль подготовки: Прикладная информатика в экономике. Содержит пример проектирования реляционной базы данных.

Предназначено для студентов направления подготовки бакалавров 230700.62 Прикладная информатика, профиль подготовки: Прикладная информатика в экономике в качестве рекомендаций к курсовому проектированию.

УДК 004:36(075.8)

ББК 32.973.26–018.2я73+60.9я73

© Туякбасарова Н.А., 2013.

© Курский институт менеджмента, экономики и бизнеса, 2013.

Оглавление

ЦЕЛИ И ЗАДАЧИ КУРСОВОГО ПРОЕКТИРОВАНИЯ	4
1. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ	4
1.1. Общие положения	4
1.2. Этапы проектирования базы данных	5
1.2.1. Инфологическое проектирование	5
1.2.2. Определение требований к операционной обстановке	9
1.2.3. Выбор СУБД и других программных средств	9
1.2.4. Логическое проектирование БД	9
1.2.5. Физическое проектирование БД	10
1.3. Особенности проектирования реляционной базы данных	10
2. ПРИМЕР ПРОЕКТИРОВАНИЯ РЕЛЯЦИОННОЙ БАЗЫ ДАННЫХ	11
2.1. Инфологическое проектирование	11
2.1.1. Анализ предметной области	11
2.1.2. Анализ информационных задач и круга пользователей системы	13
2.2. Определение требований к операционной обстановке	13
2.3. Выбор СУБД и других программных средств	14
2.4. Логическое проектирование реляционной БД	15
2.4.1. Преобразование ER–диаграммы в схему базы данных	15
2.4.2. Составление реляционных отношений	17
2.4.3. Нормализация полученных отношений (до 4НФ)	20
2.4.4. Определение дополнительных ограничений целостности	26
2.4.5. Описание групп пользователей и прав доступа	27
2.5. Физическое проектирование БД	27
ВАРИАНТЫ ЗАДАНИЙ НА КУРСОВОЕ ПРОЕКТИРОВАНИЕ	31
Литература	36

ЦЕЛИ И ЗАДАЧИ КУРСОВОГО ПРОЕКТИРОВАНИЯ

Целью курсового проектирования является применение на практике знаний, полученных в процессе изучения курса "Базы данных", и формирование практических навыков создания баз данных, составляющих основу автоматизированных информационных систем.

По своей структуре курсовой проект представляет собой следующую совокупность разделов:

1. Титульный лист (см. Приложение)
2. Введение
3. Инфологическое проектирование
4. Определение требований к операционной обстановке, в которой будет функционировать база данных
5. Выбор системы управления базой данных (СУБД) и других инструментальных программных средств
6. Логическое проектирование БД
7. Физическое проектирование БД
8. Заключение
9. Список литературы
10. Приложения

Во введении необходимо обосновать актуальность темы курсового проектирования, указать объект и предмет исследования, поставить цель и определить задачи, описать методологическую базу проектирования.

В заключении необходимо сделать выводы, подвести итоги выполненной работы по проектированию базы данных.

В Главе 2 данного учебно-методического пособия в качестве образца приведен пример, в котором рассматривается проектирование базы данных и вариант описания этого процесса от инфологического до физического проектирования БД.

Прежде чем приступать к курсовому проектированию, ознакомьтесь с теоретическими сведениями Главы 1 настоящего учебно-методического пособия.

1. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

1.1. Общие положения

Проектирование базы данных (БД) – одна из наиболее сложных и ответственных задач, связанных с созданием информационной системы (ИС). В результате её решения должен быть определен эффективный способ организации данных и инструментальные средства управления данными.

Проект базы данных должен удовлетворять следующим требованиям:

1. Схема базы данных должна быть корректной, т.е. она должна быть гомоморфным образом моделируемой предметной области (ПО), где каждому информационному объекту соответствуют данные в памяти ЭВМ, а каждому процессу – адекватные процедуры обработки данных.
2. Должно обеспечиваться ограничение на объёмы внешней и оперативной памяти и другие ресурсы вычислительной системы.
3. Эффективность функционирования, т.е. соблюдение ограничений на время реакции системы на запрос и обновление данных.
4. Защита данных от аппаратных и программных сбоев и несанкционированного доступа.
5. Простота и удобство эксплуатации.
6. Гибкость, т.е. возможность развития и адаптации к изменениям предметной области и/или требований пользователей.

Удовлетворение требований 1–4 обязательно для принятия проекта.

1.2. Этапы проектирования базы данных

Процесс проектирования включает в себя следующие этапы:

1. Инфологическое проектирование.
2. Определение требований к операционной обстановке, в которой будет функционировать информационная система.
3. Выбор системы управления базой данных (СУБД) и других инструментальных программных средств.
4. Логическое проектирование БД.
5. Физическое проектирование БД.

Инфологический подход не предоставляет формальных способов моделирования реальности, но он закладывает основы методологии проектирования баз данных.

1.2.1. Инфологическое проектирование

Основными задачами инфологического проектирования являются определение предметной области (ПО) системы и формирование взгляда на ПО с позиций будущих пользователей, т.е. инфологической модели ПО.

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

Рассмотрим основные подходы к созданию инфологической модели предметной области.

Функциональный подход к проектированию БД

Этот метод реализует принцип "от задач" и применяется тогда, когда известны функции некоторой группы лиц и/или комплекса задач, для обслуживания информационных потребностей которых создаётся рассматриваемая БД.

Предметный подход к проектированию БД

Предметный подход к проектированию БД применяется в тех случаях, когда у разработчиков есть чёткое представление о самой ПО и о том, какую именно информацию они хотели бы хранить в БД, а структура запросов не определена или определена не полностью. Тогда основное внимание уделяется исследованию ПО и наиболее адекватному её отображению в БД с учётом самого широкого спектра информационных запросов к ней.

Проектирование с использованием метода "сущность-связь"

Метод "сущность–связь" (entity–relation, ER–method) является комбинацией двух предыдущих и обладает достоинствами обоих. Этап инфологического проектирования начинается с моделирования ПО. Проектировщик разбивает её на ряд локальных областей, каждая из которых (в идеале) включает в себя информацию, достаточную для обеспечения запросов отдельной группы будущих пользователей или решения отдельной задачи (подзадачи). Каждое локальное представление моделируется отдельно, затем они объединяются.

Выбор локального представления зависит от масштабов ПО. Обычно она разбивается на локальные области таким образом, чтобы каждая из них соответствовала отдельному внешнему приложению и содержала 6-7 сущностей.

Сущность – это объект, о котором в системе будет накапливаться информация. Сущности бывают как физически существующие (например, СОТРУДНИК или АВТОМОБИЛЬ), так и абстрактные (например, ЭКЗАМЕН или ДИАГНОЗ).

Для сущностей различают тип сущности и экземпляр. Тип характеризуется именем и списком свойств, а экземпляр – конкретными значениями свойств.

Типы сущностей можно классифицировать как сильные и слабые. Сильные сущности существуют сами по себе, а существование слабых сущностей зависит от существования сильных. Например, читатель библиотеки – сильная сущность, а абонемент этого читателя – слабая, которая зависит от наличия соответствующего читателя. Слабые сущности называют подчинёнными (дочерними), а сильные – базовыми (основными, родительскими).

Каждая сущность характеризуется свойствами (атрибутами). Различают:

1. *Идентифицирующие и описательные атрибуты.* Идентифицирующие атрибуты имеют уникальное значение для сущностей данного типа и являются *потенциальными ключами*. Они позволяют однозначно распознавать экземпляры сущности. Из потенциальных ключей выбирается один первичный ключ (ПК). В качестве ПК обычно выбирается потенциальный ключ, по ко-

тому чаще происходит обращение к экземплярам записи. Кроме того, ПК должен включать в свой состав минимально необходимое для идентификации количество атрибутов. Остальные атрибуты называются описательными и включают в себе интересующие свойства сущности.

2. *Составные и простые атрибуты.* Простой атрибут состоит из одного компонента, его значение неделимо. Составной атрибут является комбинацией нескольких атрибутов, возможно, принадлежащих разным типам данных (например, ФИО или адрес). Решение о том, использовать составной атрибут или разбивать его на компоненты, зависит от характера его обработки и формата пользовательского представления этого атрибута.
3. *Однозначные и многозначные атрибуты* (могут иметь соответственно одно или много значений для каждого экземпляра сущности).
4. *Основные и производные атрибуты.* Значение основного атрибута не зависит от других атрибутов. Значение производного атрибута вычисляется на основе значений других атрибутов (например, возраст студента вычисляется на основе даты его рождения и текущей даты).

Спецификация атрибута состоит из его названия, указания типа данных и описания ограничений целостности – множества значений (или домена), которые может принимать данный атрибут.

Далее осуществляется спецификация связей внутри локального представления. Связи могут иметь различный содержательный смысл (семантику). Различают связи типа "сущность-сущность", "сущность-атрибут" и "атрибут-атрибут" для отношений между атрибутами, которые характеризуют одну и ту же сущность или одну и ту же связь типа "сущность-сущность".

Каждая связь характеризуется именем, обязательностью, типом и степенью. Различают **факультативные** (необязательные) и **обязательные** связи. Если вновь порождённый объект одного типа оказывается по необходимости связанным с объектом другого типа, то между этими типами объектов существует обязательная связь (обозначается двойной линией). Иначе связь является факультативной.

По типу различают множественные связи "один к одному" (1:1), "один ко многим" (1:n) и "многие ко многим" (m:n). ER-диаграмма, содержащая различные типы связей, приведена на рис. 1. Обратите внимание, что обязательные связи на рис. 1 выделены двойной линией.

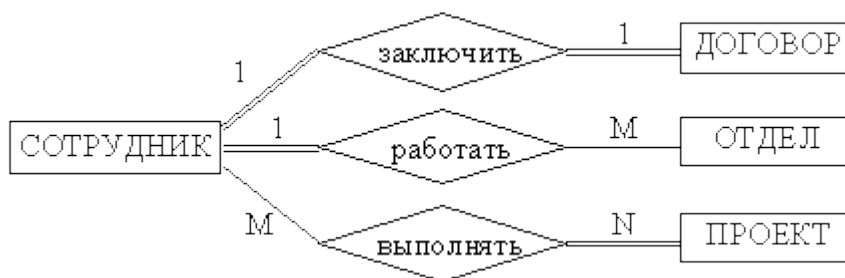


Рис.1. ER-диаграмма с примерами типов множественных связей

Степень связи определяется количеством сущностей, которые охвачены данной связью. Пример бинарной связи – связь между отделом и сотрудниками, которые в нём работают. Примером тернарной связи является связь типа *экзамен* между сущностями *ДИСЦИПЛИНА*, *СТУДЕНТ*, *ПРЕПОДАВАТЕЛЬ*. Из последнего примера видно, что связь также может иметь атрибуты (в данном случае это *Дата проведения* и *Оценка*). Пример ER–диаграммы с указанием сущностей, их атрибутов и связей приведен на рис. 2.

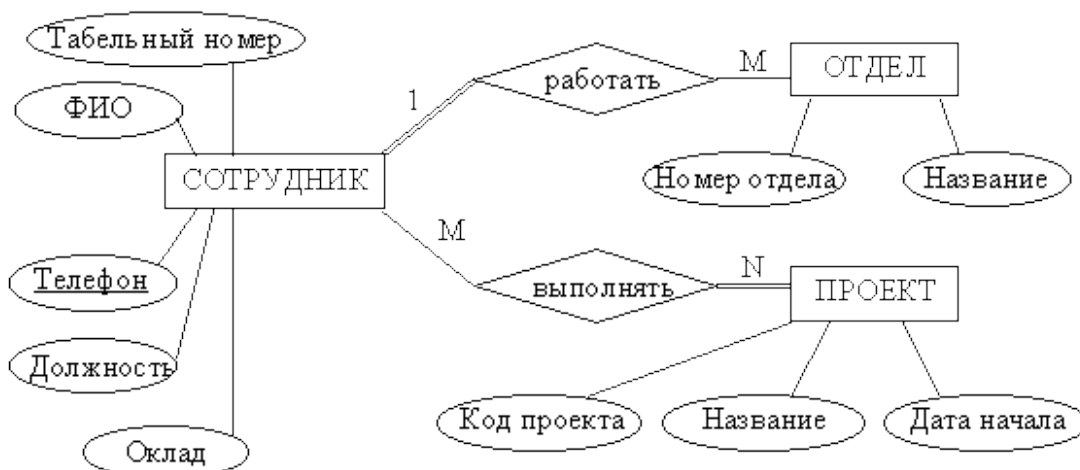


Рис.2. Пример ER–диаграммы с однозначными и многозначными атрибутами

После того, как созданы локальные представления, выполняется их объединение. При небольшом количестве локальных областей (не более пяти) они объединяются за один шаг. В противном случае обычно выполняют бинарное объединение в несколько этапов.

При объединении проектировщик может формировать конструкции, производные по отношению к тем, которые были использованы в локальных представлениях. Такой подход может преследовать следующие цели:

- объединение в единое целое фрагментарных представлений о различных свойствах одного и того же объекта;
- введение абстрактных понятий, удобных для решения задач системы, установление их связи с конкретными понятиями, использованными в модели;
- образование классов и подклассов подобных объектов (например, класс "изделие" и подклассы типов изделий, производимых на предприятии).

На этапе объединения необходимо выявить и устранить все противоречия. Например, одинаковые названия семантически различных объектов или связей или несогласованные ограничения целостности на одни и те же атрибуты в разных приложениях. Устранение противоречий вызывает необходимость возврата к этапу моделирования локальных представлений с целью внесения в них соответствующих изменений.

По завершении объединения результаты проектирования являются собой концептуальную инфологическую модель предметной области. Модели локальных представлений – это внешние инфологические модели.

1.2.2. Определение требований 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.

1.2.3. Выбор СУБД и других программных средств

Выбор СУБД является одним из важнейших моментов в разработке проекта БД, так как он принципиальным образом влияет на весь процесс проектирования БД и реализацию информационной системы. Теоретически при выборе СУБД нужно принимать во внимание десятки факторов. Но практически разработчики руководствуются лишь собственной интуицией и несколькими наиболее важными критериями, к которым, в частности, относятся:

- тип модели данных, которую поддерживает данная СУБД, её адекватность потребностям рассматриваемой предметной области;
- характеристики производительности системы;
- запас функциональных возможностей для дальнейшего развития ИС;
- степень оснащённости системы инструментарием для персонала администрирования данными;
- удобство и надёжность СУБД в эксплуатации;
- стоимость СУБД и дополнительного программного обеспечения.

1.2.4. Логическое проектирование БД

На этапе логического проектирования разрабатывается логическая структура БД, соответствующая логической модели ПО. Решение этой задачи существенно зависит от модели данных, поддерживаемой выбранной СУБД.

Результатом выполнения этого этапа являются схемы БД концептуального и внешнего уровней архитектуры, составленные на языках определения данных (DDL, Data Definition Language), поддерживаемых данной СУБД.

В настоящем курсовом проекте логическая схема БД должна быть реализована в нотации IDEF1X с помощью инструментального средства All Fusion Data Modeler (ER Win).

1.2.5. Физическое проектирование БД

Этап физического проектирования заключается в увязке логической структуры БД и физической среды хранения с целью наиболее эффективного размещения данных, т.е. отображении логической структуры БД в структуру хранения. Решается вопрос размещения хранимых данных в пространстве памяти, выбора эффективных методов доступа к различным компонентам "физической" БД. Результаты этого этапа документируются в форме схемы хранения на языке определения данных (DDL). Принятые на этом этапе решения оказывают определяющее 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 разрабатывается стратегия резервного копирования. Для защиты от несанкционированного доступа каждому пользователю доступ к данным предоставляется только в соответствии с его правами доступа.

1.3. Особенности проектирования реляционной базы данных

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 ПО и неразвитостью средств описания ограничений целостности на уровне модели данных.

Для решения подобных проблем проводится нормализация отношений.

2. ПРИМЕР ПРОЕКТИРОВАНИЯ РЕЛЯЦИОННОЙ БАЗЫ ДАННЫХ

В качестве примера рассмотрим проектирование базы данных книжного издательства.

2.1. Инфологическое проектирование

2.1.1. Анализ предметной области

База данных создаётся для информационного обслуживания редакторов, менеджеров и других сотрудников издательства. БД должна содержать данные о сотрудниках, книгах, авторах, финансовом состоянии издательства и предоставлять возможность получения отчётов.

В соответствии с описанием предметной области база данных проектируется с учётом следующих особенностей:

1. каждая книга издаётся в рамках контракта;
2. книга может быть написана несколькими авторами;
3. контракт подписывается одним менеджером и всеми авторами книги;
4. каждый автор может написать несколько книг (по разным контрактам);
5. порядок, в котором авторы указаны на обложке, влияет на размер гонорара;
6. если сотрудник является редактором, то он может работать одновременно над несколькими книгами;
7. у каждой книги может быть несколько редакторов, один из них – ответственный редактор;
8. каждый заказ оформляется на одного заказчика;
9. в заказе на покупку может быть перечислено несколько книг.

На **первом этапе** проектирования базы данных выделим базовые сущности предметной области и определим их атрибуты:

Сотрудники издательства (ФИО, табельный номер, пол, дата рождения, паспортные данные, ИНН, домашний адрес, телефоны, должность, оклад). Для редакторов необходимо хранить сведения о редактируемых книгах; для менеджеров – сведения о подписанных контрактах.

Авторы (ФИО, ИНН, паспортные данные, домашний адрес, телефоны). Для авторов необходимо хранить сведения о написанных книгах.

Книги (авторы, название, тираж, дата выхода, цена одного экземпляра, общие затраты на издание, авторский гонорар).

Контракты (номер, дата подписания, участники) будем рассматривать как связующее звено между сущностями авторы, книги и менеджерами.

Для отражения финансового положения компании в системе нужно учитывать заказы на книги. Для сущности **Заказ** необходимо хранить номер заказа, дату поступления заказа, дату его выполнения, список заказанных книг с указанием количества экземпляров, наименование заказчика, адрес заказчика, телефон заказчика.

Определим связи между базовыми сущностями:

1. Книги **издаются** по Контракту
2. Книги **пишут** Авторы
3. Контракт **подписывает** Сотрудник-менеджер
4. Контракт **подписывают** Авторы
5. Сотрудник-редактор **редактирует** Книги
6. Заказ **содержит** Книги

На **втором этапе** проектирования построим диаграммы ER-типа с учетом всех сущностей и связей между ними. Диаграмма ER-типа предметной области «Издательство» приведена на рис.1.

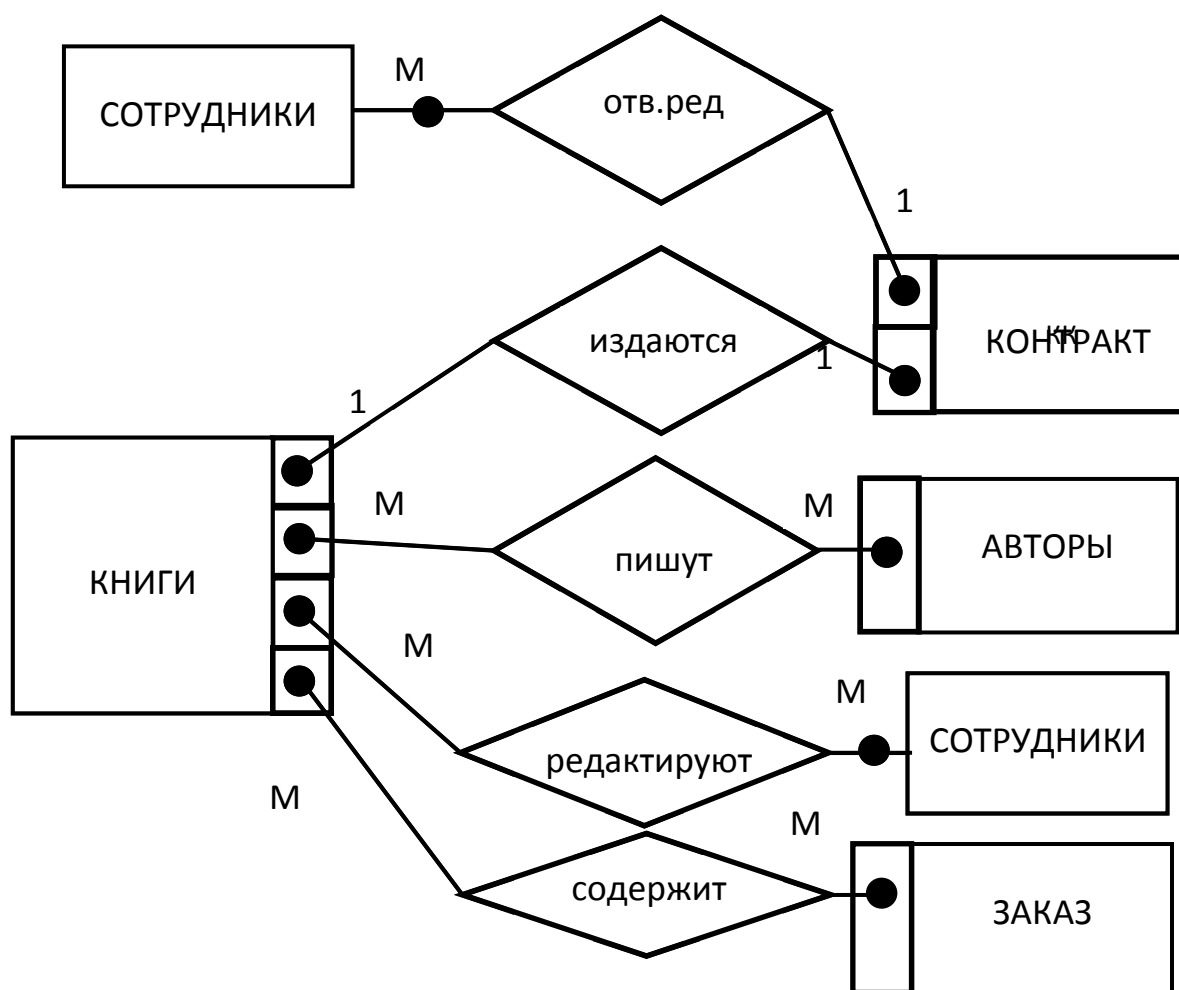


Рис.1. ER-диаграмма предметной области «Издательство»

2.1.2. Анализ информационных задач и круга пользователей системы

Система создаётся для обслуживания следующих групп пользователей:

1. администрация (дирекция);
2. менеджеры;
3. редакторы;
4. сотрудники компании, обслуживающие заказы.

Определим границы информационной поддержки пользователей:

1) Функциональные возможности:

- ведение БД (запись, чтение, модификация, удаление в архив);
- обеспечение логической непротиворечивости БД;
- обеспечение защиты данных от несанкционированного или случайного доступа (разграничение прав доступа);
- реализация наиболее часто встречающихся запросов в готовом виде;
- предоставление возможности сформировать произвольный запрос на языке манипулирования данными.

2) Готовые запросы:

- получение списка всех текущих проектов (книг, находящихся в печати и в продаже);
- получение списка редакторов, работающих над книгами;
- получение полной информации о книге (проекте);
- получение сведений о конкретном авторе (с перечнем всех книг);
- получение информации о продажах (по одному или по всем проектам);
- определение общей прибыли от продаж по текущим проектам;
- определение размера гонорара автора по конкретному проекту.

2.2. Определение требований к операционной обстановке

Для выполнения этого этапа необходимо знать (хотя бы ориентировочно) объём работы издательства (т.е. количество книг, авторов и заказчиков), 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). Наиболее существенным обычно является МД. Объём памяти МД, требуемый для хранения данных, можно приблизительно оценить по формуле

$$M_c = 2 \sum_{i=1}^n l_i * (N_i + N_{ai})$$

где l_i – длина записи в i -й таблице (в байтах), N_i – примерное (максимально возможное) количество записей в i -й таблице, N_{ai} – количество записей в архиве i -й таблицы. Коэффициент 2 перед суммой нужен для того, чтобы выделить память для хранения индексов, промежуточных данных, для выполнения объёмных операций (например, сортировки) и т.п.

Подсчитаем приблизительно, какой объём внешней памяти потребуется для хранения данных. Примем ориентировочно, что:

- одновременно осуществляется около пятидесяти проектов, работа над проектом продолжается в среднем два месяца (по 0,3К);
- в компании работает 100 сотрудников (по 0,2К на каждого сотрудника);
- издательство сотрудничает с тридцатью авторами (по 0,2К);
- в день обслуживается порядка двадцати заявок (по 0,1К);
- устаревшие данные переводятся в архив.

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

$$M_c = 2(100*0,2+6(50*0,3)+30*0,2+250(20*0,1)) = 1232 \text{ К} \approx 1,2 \text{ М},$$

где 250 – количество рабочих дней в году, а 12 мес./2 мес. = 6. Объём памяти будет увеличиваться ежегодно на ту же величину при сохранении объёма работы.

Объём памяти, занимаемый программными модулями пользователя, обычно невелик по сравнению с объёмом самих данных, поэтому может не учитываться. Требуемый объём оперативной памяти определяется на основании анализа интенсивности запросов и объёма результирующих данных.

2.3. Выбор СУБД и других программных средств

Анализ информационных задач показывает, что для реализации требуемых функций подходят почти все СУБД для ПЭВМ (FoxPro, Clipper, MS Access, MySQL и др.). Все они поддерживают реляционную модель данных и предоставляют разнообразные возможности для работы с данными.

Объём внешней и оперативной памяти, требующийся для функционирования СУБД, обычно указывается в сопроводительной документации.

Для того чтобы в учебном примере не привязываться к конкретной СУБД, выполним описание логической схемы БД на SQL-92.

2.4. Логическое проектирование реляционной БД

2.4.1. Преобразование ER-диаграммы в схему базы данных

База данных создаётся на основании схемы базы данных. Преобразование ER-диаграммы в схему БД выполняется путем сопоставления каждой сущности и каждой связи, имеющей атрибуты, отношения (таблицы БД).

На схеме (рис. 1) есть связь типа 1:1 – обязательная связь между КНИГАМИ и КОНТРАКТАМИ. Такие отношения следует объединять в одно. Дополнительный эффект от объединения этих отношений – слияние связей авторы–контракты и авторы–книги: ведь в нашем случае контракт заключается именно для написания книги.

Примечание: исключение для связи типа 1:1 составляют ситуации, когда для увеличения производительности системы в отдельную таблицу выделяются редко используемые данные большого объёма.

Связь типа 1:М (один-ко-многим) между отношениями реализуется через внешний ключ, создаваемый для отношения, к которому осуществляется множественная связь (КНИГИ).

Между сущностями КНИГИ и ЗАКАЗЫ существует связь М:М. Формируем 3 отношения согласно правилу с номером 6: КНИГИ, ЗАКАЗЫ, СТРОКИ_ЗАКАЗА. Отношение СТРОКИ_ЗАКАЗА наследует ключевые атрибуты отношений КНИГИ и ЗАКАЗЫ. Добавляем в отношение СТРОКИ_ЗАКАЗА дополнительный атрибут Количество.

Между сущностями КНИГИ и АВТОРЫ существует связь типа М:М. Формируем 3 отношения согласно правилу с номером 6: АВТОРЫ, КНИГИ, АВТОРЫ_КНИГИ. Отношение АВТОРЫ_КНИГИ наследует ключевые поля отношений АВТОРЫ и КНИГИ. Добавляем в отношение АВТОРЫ_КНИГИ дополнительные атрибуты N_на обложке, процент гонорара.

Связь между отношениями КНИГИ и СОТРУДНИКИ принадлежит к типу М:М. Этот тип связи реализуется через вспомогательное отношение, которое является соединением первичных ключей соответствующих отношений. Таким образом, получаем три отношения КНИГИ, СОТРУДНИКИ, КНИГИ_СОТРУДНИКИ.

Бинарная связь между отношениями не может быть обязательной для обоих отношений. После объединения сущностей КНИГИ и КОНТРАКТЫ остаются следующие связи, обязательные для всех участников: между авторами и книгами, заказами и заказчиками, между заказами и строками заказов. Такой тип связи означает, что, например, прежде чем добавить новый заказ в отношение ЗАКАЗЫ,

нужно добавить новую строку в отношении СТРОКИ ЗАКАЗА, 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.

Учитывая вышесказанное, получаем уточненную схему БД предметной области «Издательство» (рис.2).

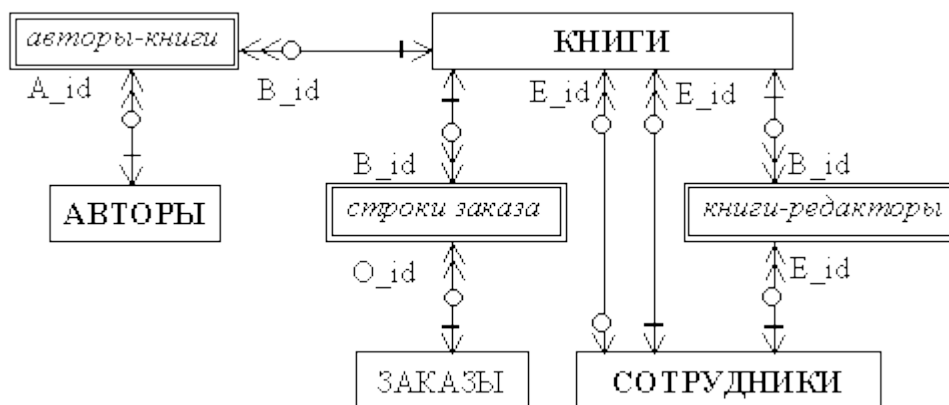


Рис.2. Уточнённая схема БД предметной области «Издательство»

На схеме используются следующие обозначения.

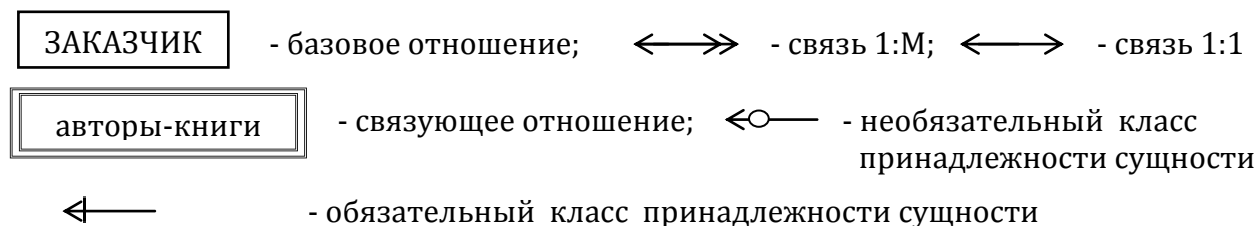


Схема на рис.2 содержит цикл "сотрудники–книги–сотрудники". Цикл допустим только в том случае, если связи, входящие в него, независимы друг от друга. Примем для нашей предметной области, что ответственный редактор книги может являться также просто редактором этой же книги или не входить в число редакторов. При этом цикл не приводит к нарушению логической целостности данных.

Примечание. Существует несколько подходов для разрешения ситуаций, в которых связи, входящие в цикл, зависят друг от друга. Рассмотрим пример цикла "отделы–проекты–сотрудники–отделы" (рис. 3,а). Будем считать, что в выполнении проекта могут участвовать только сотрудники, работающие в том же отделе, к которому относится проект. При циклической схеме СУБД не сможет гарантировать логическую целостность данных без использования дополнительных средств.

Один из способов – разорвать цикл, исключив одну из связей (рис. 3,б) или введя промежуточное отношение (рис. 3,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, ес-

ли бы существовала общая связь между сущностями, входящими в цикл. Например, если бы каждый сотрудник заключал договор с отделом на выполнение работ в рамках проекта. Тогда сущность ДОГОВОРЫ отражала бы связь между отделами, сотрудниками и проектами.

Другой способ разрешения цикла заключается в том, что в промежуточное отношение СОТРУДНИКИ_ПРОЕКТЫ, которое реализует связь многие-ко-многим, добавляются (мигрируют) внешние ключи Код отдела (D_id) из отношений СОТРУДНИКИ и ПРОЕКТЫ (рис. 3,г). Эти ключи проверяются на равенство друг другу с помощью соответствующего ограничения целостности. 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 не пригодны, логическая целостность контролируется программно или вручную.

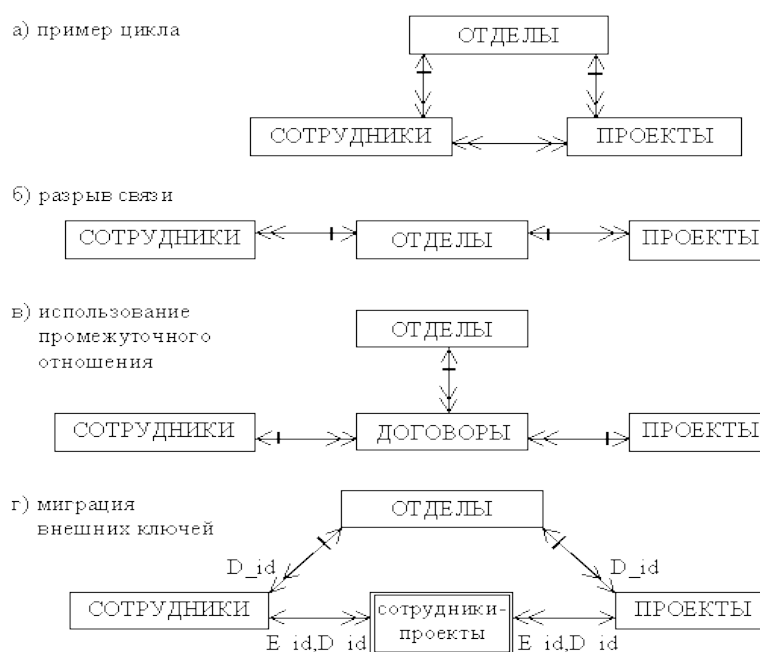


Рис.3. Некоторые способы разрешения циклов в схеме базы данных

2.4.2. Составление реляционных отношений

Каждое реляционное отношение соответствует одной сущности (объекту предметной области) и в него вносятся все атрибуты сущности. Для каждого отношения необходимо определить первичный ключ и внешние ключи (если они есть). 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.

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).

Потенциальными ключами отношения АВТОРЫ являются атрибуты:

Паспортные данные и ИНН. Первый хранится как длинная строка, а последний по условиям предметной области не является обязательным. Поэтому для авторов необходимо ввести суррогатный ключ – A_id. Книги можно идентифицировать по атрибуту Контракт: его номер обязателен и уникален. Потенциальные ключи отношения СОТРУДНИКИ – атрибуты ИНН, Паспортные данные, Табельный номер, причём все они обязательные. Табельный номер занимает меньше памяти, чем ИНН, поэтому он и будет первичным ключом. Кортежи отношения ЗАКАЗЫ можно идентифицировать ключом Номер заказа.

Потенциальными ключами вспомогательных отношений являются комбинации первичных ключей соответствующих базовых отношений.

Отношения приведены в табл. 1-7. Для каждого отношения указаны атрибуты с их внутренним названием, типом и длиной. Типы данных обозначаются так: N – числовой, C – символьный, D – дата (последний имеет стандартную длину, зависящую от СУБД, поэтому она не указывается).

Таблица 1. Схема отношения СОТРУДНИКИ (Employees)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечания</i>
Табельный номер	E_ID	N(4)	первичный ключ
Фамилия, имя, отчество	E_NAME	C(50)	обязательное поле
Дата рождения	E_BORN	D	
Пол	E_SEX	C(1)	обязательное поле
Паспортные данные	E_PASSP	C(50)	обязательное поле
ИНН	E_INN	N(12)	обязательное уникальное поле
Должность	E_POST	C(30)	обязательное поле
Оклад	E_SALARY	N(8,2)	обязательное поле
Адрес	E_ADDR	C(50)	
Телефоны	E_TEL	C(30)	многозначное поле

Таблица 2. Схема отношения КНИГИ (Books)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечания</i>
Номер контракта	B_CONTRACT	N(6)	первичный ключ
Дата подписания контракта	B_DATE	D	обязательное поле
Менеджер	B_MAN	N(4)	внешний ключ (к Employees)
Название книги	B_TITLE	N(40)	обязательное поле
Цена	B_PRICE	N(6,2)	цена экземпляра книги
Затраты	B_ADVANCE	N(10,2)	общая сумма затрат на книгу
Авторский гонорар	B_FEE	N(8,2)	общая сумма гонорара
Дата выхода	B_PUBL	D	
Тираж	B_CIRCUL	N(5)	
Ответственный редактор	B_EDIT	N(4)	внешний ключ (к Employees)

Таблица 3. Схема отношения АВТОРЫ (Authors)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечания</i>
Код автора	A_ID	N(4)	суррогатный первичный ключ
Фамилия, имя, отчество	A_NAME	C(50)	обязательное поле
Паспортные данные	A_PASSP	C(50)	обязательное поле
ИНН	A_INN	N(12)	уникальное поле
Адрес	A_ADDR	C(50)	обязательное поле
Телефоны	A_TEL	C(30)	многозначное поле

Таблица 4. Схема отношения ЗАКАЗЫ (Orders)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечания</i>
Номер заказа	O_ID	N(6)	первичный ключ
Заказчик	O_COMPANY	C(40)	обязательное поле
Дата поступления заказа	O_DATE	D	обязательное поле
Адрес заказчика	O_ADDR	C(50)	обязательное поле
Дата выполнения заказа	O_READY	D	

Таблица 5. Схема отношения КНИГИ–АВТОРЫ (Titles)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечания</i>
Код книги (№ контракта)	B_ID	N(6)	внешний ключ (к Books)
Код автора	A_ID	N(4)	внешний ключ (к Authors)
Номер в списке	A_NO	N(1)	обязательное поле
Гонорар	A_FEE	N(3)	процент от общего гонорара

Таблица 6. Схема отношения КНИГИ–РЕДАКТОРЫ (Editors)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечания</i>
Код книги (№ контракта)	B_ID	N(6)	внешний ключ (к Books)
Код редактора	E_ID	N(4)	внешний ключ (к Employees)

Таблица 7. Схема отношения СТРОКИ ЗАКАЗА (Items)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечания</i>
Номер заказа	O_ID	N(6)	внешний ключ (к Orders)
Код книги (№ контракта)	B_ID	N(6)	внешний ключ (к Books)
Количество	B_COUNT	N(4)	обязательное поле

2.4.3. Нормализация полученных отношений (до 4НФ)

1НФ. Для приведения таблиц к 1НФ требуется составить прямоугольные таблицы (один атрибут – один столбец) и разбить сложные атрибуты на простые, а многозначные атрибуты вынести в отдельные отношения.

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 атрибуты Фамилия, имя, отчество на два атрибута Фамилия и Имя, отчество и Паспортные данные на атрибуты Номер паспорта (уникальный), Дата выдачи и Кем выдан.

Многозначный атрибут Телефоны отношения СОТРУДНИКИ имеет неоднородные значения. Один из номеров телефонов – основной – определяется рабочим ме-

стом сотрудника (рассматриваются только стационарные телефоны). Наличие других номеров зависит от того, есть ли в том же помещении (комнате) другие сотрудники, имеющие стационарные телефоны. Можно добавить в отношении СОТРУДНИКИ атрибут Номер комнаты, а в атрибуте Рабочие телефоны хранить номер того телефона, который стоит на рабочем месте сотрудника. Дополнительные номера телефонов можно будет вычислить из других кортежей с таким же номером комнаты. Но в случае увольнения сотрудника мы потеряем сведения о номере рабочего телефона.

Поэтому создадим новое отношение КОМНАТЫ и включим в него атрибуты Номер комнаты и Телефон. Так как в комнате может не быть телефона, первичный ключ нового отношения не определен (ПК не может содержать null-значения), но на этих атрибутах можно определить составной уникальный ключ. Связь между отношениями СОТРУДНИКИ и КОМНАТЫ реализуем через составной внешний ключ (Номер комнаты, Телефон). Значение внешнего ключа для каждого сотрудника будем брать из того кортежа, в котором хранится основной рабочий телефон этого сотрудника.

2НФ. В нашем случае составные первичные ключи имеют отношения СТРОКИ ЗАКАЗА, КНИГИ-АВТОРЫ и КНИГИ-РЕДАКТОРЫ. Неключевые атрибуты этих отношений функционально полно зависят от первичных ключей.

3НФ. В отношении ЗАКАЗЫ атрибут Адрес заказчика зависит от атрибута Заказчик, а не от первичного ключа, поэтому адрес 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 Адрес заказчика и ввести для него суррогатный первичный ключ (ПК). Так как каждый заказчик может сделать несколько заказов, связь между отношениями ЗАКАЗЧИКИ и ЗАКАЗЫ будет 1:М и суррогатный ПК станет внешним ключом для отношения ЗАКАЗЫ.

В отношении СОТРУДНИКИ атрибут Оклад зависит от атрибута Должность. Поступим с этой транзитивной зависимостью так же, как в предыдущем случае: создадим новое отношение ДОЛЖНОСТИ, перенесём в него атрибуты Должность и Оклад и введём суррогатный первичный ключ.

В отношениях СОТРУДНИКИ и АВТОРЫ атрибуты Дата выдачи и Кем выдан зависят от атрибута Номер паспорта, а не от первичного ключа. Но если мы выделим их в отдельное отношение, то получившиеся связи будут иметь тип 1:1. Следовательно, декомпозиция нецелесообразна.

4НФ. Отношения данного примера не нарушают 4НФ, т.к. не содержат нетривиальных многозначных зависимостей.

В реальных базах данных после нормализации может проводиться денормализация. Она проводится с одной целью – повышение производительности БД. Рассмотрим некоторые запросы к созданной базе данных.

Например, запрос на получение списка телефонов авторов или домашних телефонов сотрудников потребует в нормализованной БД соединения отношений. Пользователю безразлична форма представления этого списка: номера телефонов через запятую или в столбец. Поэтому мы откажемся от создания отдельных отношений с номерами телефонов, и вернёмся к варианту с многозначными полями. (Это не касается рабочих телефонов сотрудников).

Другой запрос: как определяется, можно ли выполнить очередной заказ? Для каждой позиции заказа нужно просуммировать количество книг по выполненным заказам, получить остаток (тираж минус полученная сумма) и сравнить остаток с объёмом заказа. Такой расчёт может потребовать много времени, поэтому предлагается добавить в отношение КНИГИ производный атрибут Остаток тиража. Значение этого атрибута должно автоматически пересчитываться при установлении даты выполнения заказа.

После проведённых преобразований схема БД выглядит так (рис. 4):

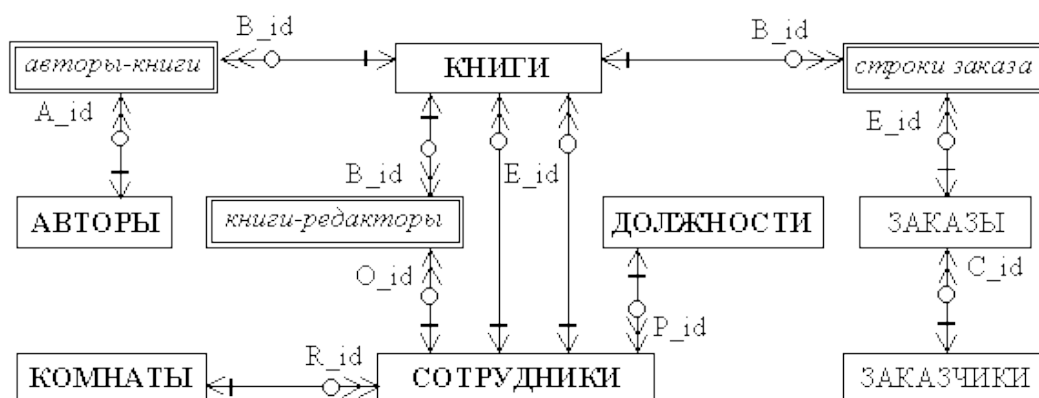


Рис.4. Окончательная схема БД предметной области «Издательство»

Окончательные схемы отношений базы данных с указанием ключей и других ограничений целостности приведены в табл. 8–17.

Таблица 8. Схема отношения ДОЛЖНОСТИ (Posts)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечания</i>
Код должности	P_ID	N(3)	суррогатный первичный ключ
Название должности	P_POST	C(30)	обязательное поле
Оклад	P_SAL	N(8,2)	обязательное поле

Таблица 9. Схема отношения КОМНАТЫ (Rooms)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечания</i>
Номер комнаты	R_NO	N(3)	обязательное поле
Номер телефона	R_TEL	C(10)	

Таблица 10. Схема отношения СОТРУДНИКИ (Employees)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечания</i>
Табельный номер	E_ID	N(4)	первичный ключ
Фамилия	E_FNAME	C(20)	обязательное поле
Имя, отчество	E_LNAME	C(30)	обязательное поле
Дата рождения	E_BORN	D	
Пол	E_SEX	C(1)	обязательное поле
Код должности	E_POST	N(3)	внешний ключ (к Posts)
Номер комнаты	E_ROOM	N(3)	составной внешний ключ (к Rooms)
Номер телефона	E_TEL	C(10)	
ИНН	E_INN	C(12)	обязательное поле
Номер паспорта	E_PASSP	C(12)	обязательное поле
Кем выдан паспорт	E_ORG	C(30)	обязательное поле
Дата выдачи паспорта	E_PDATE	D	обязательное поле
Адрес	E_ADDR	C(50)	

Таблица 11. Схема отношения ЗАКАЗЧИКИ (Customers)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечания</i>
Код заказчика	C_ID	N(4)	суррогатный первичный ключ
Заказчик	C_NAME	C(30)	обязательное поле
Адрес заказчика	C_ADDR	C(50)	обязательное поле

Таблица 12. Схема отношения АВТОРЫ (Authors)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечания</i>
Код автора	A_ID	N(4)	суррогатный ключ
Фамилия	A_FNAME	C(20)	обязательное поле
Имя, отчество	A_LNAME	C(30)	обязательное поле
ИНН	A_INN	C(12)	
Номер паспорта	A_PASSP	C(12)	обязательное поле
Кем выдан паспорт	A_ORG	C(30)	обязательное поле
Дата выдачи паспорта	A_PDATE	D	обязательное поле
Адрес	A_ADDR	C(50)	обязательное поле
Телефоны	A_TEL	C(30)	многозначное поле

Таблица 13. Схема отношения КНИГИ (Books)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечания</i>
Номер контракта	B_CONTRACT	N(6)	первичный ключ
Дата подписания контракта	B_DATE	D	обязательное поле
Менеджер	B_MAN	N(4)	внешний ключ (к Employees)
Название книги	B_TITLE	N(40)	обязательное поле
Цена	B_PRICE	N(6,2)	цена экземпляра книги
Затраты	B_ADVANCE	N(10,2)	общая сумма затрат на книгу
Авторский гонорар	B_FEE	N(8,2)	общая сумма гонорара
Дата выхода	B_PUBL	D	
Тираж	B_CIRCUL	N(5)	
Ответственный редактор	B_EDIT	N(4)	внешний ключ (к Employees)
Остаток тиража	B_REST	N(5)	производное поле

Таблица 14. Схема отношения ЗАКАЗЫ (Orders)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечания</i>
Номер заказа	O_ID	N(6)	первичный ключ
Код заказчика	O_COMPANY	N(4)	внешний ключ (к Customers)
Дата поступления заказа	O_DATE	D	обязательное поле
Дата выполнения заказа	O_READY	D	

Таблица 15. Схема отношения КНИГИ–АВТОРЫ (Titles)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечания</i>
Код книги (№ контракта)	B_ID	N(6)	внешний ключ (к Books)
Код автора	A_ID	N(4)	внешний ключ (к Authors)
Номер в списке	A_NO	N(1)	обязательное поле
Гонорар	A_FEE	N(3)	процент от общего гонорара

Таблица 16. Схема отношения СТРОКИ ЗАКАЗА (Items)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечания</i>
Номер заказа	O_ID	N(6)	внешний ключ (к Orders)
Код книги (№ контракта)	B_ID	N(6)	внешний ключ (к Books)
Количество	B_COUNT	N(4)	обязательное поле

Таблица 17. Схема отношения КНИГИ–РЕДАКТОРЫ (Editors)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечания</i>
Код книги (№ контракта)	B_ID	N(6)	внешний ключ (к Books)
Код редактора	E_ID	N(4)	внешний ключ (к Employees)

2.4.4. Определение дополнительных ограничений целостности

Перечислим ограничения целостности, которые не указаны в табл. 8–17.

1. Значения всех числовых атрибутов – больше 0 (или null, если атрибут необязателен).
2. Область значений атрибута Sex отношения EMPLOYEES – символы 'м' и 'ж'.
3. Отношение ROOMS не имеет первичного ключа, но комбинация значений (R_NO, TEL) уникальна.
4. В отношении TITLES порядковые номера авторов на обложке одной книги должны идти подряд, начиная с 1.
5. В отношении TITLES сумма процентов гонорара по одной книге равна 100.

Ограничения (4,5) нельзя реализовать в схеме отношения. В реальных БД подобные ограничения целостности реализуются программно (через внешнее приложение или специальную процедуру контроля данных).

2.4.5. Описание групп пользователей и прав доступа

Опишем для каждой группы пользователей права доступа к каждой таблице и к каждому полю (атрибуту).

1. Администратор БД: имеет доступ ко всем данным (по записи), может изменять структуру базы данных и связи между отношениями. Устанавливает права доступа для всех остальных групп.
2. Представители администрации компании: имеют доступ по чтению ко всем данным и доступ по записи к отношениям POSTS, ROOMS и EMPLOYEES.
3. Менеджер: имеет доступ по чтению ко всем данным, кроме отношения POSTS. Имеют доступ по записи к отношениям AUTHORS, CUSTOMERS, BOOKS, EDITORS, TITLES, ORDERS, ITEMS.
4. Редакторы: имеют доступ по чтению к следующим отношениям:
 - AUTHORS, кроме полей A_passp, A_org, A_pdate, A_INN(паспортные данные и ИНН).
 - BOOKS, кроме полей B_advance, B_fee (затраты и гонорар).
 - EDITORS.
 - TITLES.
5. Сотрудники, принимающие и выполняющие заказы: имеют доступ по записи к отношениям CUSTOMERS, ORDERS, ITEMS и по чтению к полям B_title, B_circul, B_priceи B_rest отношения BOOKS (название, тираж, цена, непроданный остаток тиража).

2.5. Физическое проектирование БД

Приведём фрагмент описания схемы БД на DDL (Data Definition Language, язык описания данных — это семейство компьютерных языков, используемых в компьютерных программах для описания структуры баз данных). На текущий момент наиболее популярным языком DDL является SQL, используемый для получения и манипулирования данными в РСУБД.

В настоящем курсовом проекте для физического проектирования БД в качестве СУБД рекомендуется использовать MySQL. Работать с MySQL можно из командной строки, а так же посредством Denwer, который предоставляет проектировщику web-интерфейс для работы с данными.

1. Создать базу данных IZDAT

```
create database IZDAT;
```

2. Выбрать базу данных

```
use IZDAT;
```

3. Создать отношение POSTS (должности):

```
create table posts (  
p_id numeric(3) primary key,  
p_post varchar(30) not null,  
p_salary numeric(8,2) not null check(p_salary > 0));
```

4. Создать отношение ROOMS (комнаты):

```
create table rooms (  
r_no numeric(3) primary key,  
r_tel varchar(10),  
unique(r_no, r_tel));
```

5. Создать отношение EMPLOYEES (сотрудники):

```
create table employees (  
e_id numeric(4) primary key,  
e_fname varchar(20) not null,  
e_lname varchar(30) not null,  
e_born date,  
e_sex char(1) not null check(e_sex in ('ж','м')),  
e_post numeric(3) references posts,  
e_room numeric(3),  
e_tel varchar(10),  
e_inn char(12) not null,  
e_passp char(12) not null,  
e_org varchar(30) not null,  
e_pdate date not null,  
e_addr varchar(50),  
foreign key(e_room,e_tel) references rooms(r_no,r_tel));
```

Другие отношения создаются аналогично.

Права доступа пользователей, описанные в п. 2.4.4, предоставляются с помощью команды GRANT. Рассмотрим для примера права сотрудника компании user1, который принимает и обслуживает заказы. Права доступа к отношениям CUSTOMERS, ORDERS, ITEMS могут быть описаны следующим образом:

```
grant insert, update on customers to user1;
```

```
grant insert, update, delete on orders to user1;
```

```
grant insert, update, delete on items to user1;
```

Для реализации частичного доступа к отношению BOOKS следует создать соответствующее представление и предоставить доступ к этому представлению:

```
create view goods (id, title, circul, price, rest)
```

```
as select b_contract, b_title, b_circul, b_price, b_rest
```

```
from books;
```

```
grant select on goods to user1;
```

Приведём примеры нескольких готовых запросов:

1. Список всех текущих проектов (книг, находящихся в печати и в продаже):

```
create view edits as
```

```
select b_contract, b_title, b_date
```

```
from books
```

```
where b_rest is null or b_rest > 0;
```

2. Список редакторов, работающих над книгами:

```
create view edits (title, fname, lname) as
```

```
select b_title, e_fname, e_lname /*ответственный редактор*/
```

```
from books, employees e
```

```
where b_edit=e_id and
```

```
(b_publ is null or b_publ > sysdate);
```

```
union /*sysdate – текущая дата*/
```

```
select b_title, a_fname, a_lname
```

```
from books b, employees e, editors d
```

where b.b_contract=d.b_id and d.e_id=a.e_id and
(b_publ is null or b_publ > sysdate)
order by 1;

3. Определение общей прибыли от продаж по текущим проектам:

```
create view edits (title, total) as  
select b_title, (circul-rest)*price-advance  
from books  
where b_rest is null or b_rest > 0;
```

Анализ готовых запросов показывает, что для повышения эффективности работы с данными необходимо создать индексы для всех внешних ключей (и всех первичных ключей, если выбранная СУБД не создаёт их автоматически). Приведём примеры создания индексов:

```
create index e_posts on employees(e_post);  
create index b_editors on books(b_edit);  
create unique index r_tel on rooms(r_no,r_tel);
```

ВАРИАНТЫ ЗАДАНИЙ НА КУРСОВОЕ ПРОЕКТИРОВАНИЕ

1. БД "Деканат".

Задача – информационная поддержка деятельности деканата вуза:

- ведение расписания сессии, хранение результатов сессии;
- составление зачётных и экзаменационных ведомостей;
- составление расписаний экзаменов по группам, кафедрам, для отдельных преподавателей;
- проверка корректности расписания экзаменов (уникальность комбинации "время – дата – аудитория"; между экзаменами в одной группе должно пройти не менее трёх дней);
- подсчёт по результатам зачётов и экзаменов итоговых значений (количество оценок '5', '4', '3', '2', количество неявок, средний балл по группе);
- получение списка экзаменов на текущую дату.

2. БД "Отдел кадров" (института).

Задача – информационная поддержка деятельности отдела кадров.

Различают три группы сотрудников: а) администрация; б) преподавательский и инженерно-технический состав (по кафедрам); в) технический персонал. БД должна содержать штатное расписание по отделам (кафедрам) с указанием количества ставок по должностям, включать архив сотрудников и учитывать сотрудников, находящихся в отпуске по уходу за ребенком.

БД должна предоставлять возможность составления должностных (штатных) расписаний по кафедрам и отделам и следующих списков:

- вакансий (с учётом сотрудников, находящихся в отпуске по уходу за ребенком, т.е. с указанием даты, до которой ставка свободна);
- пенсионеров;
- людей предпенсионного возраста (не более 2-х лет до пенсии);
- бездетных сотрудников;
- юбиляров текущего года;
- многодетных сотрудников (трое и более детей);
- ветеранов (работающих в институте не менее тридцати лет);
- сотрудников, работающих более чем на одной ставке.

3. БД "Плановый отдел".

Задача – информационная поддержка деятельности планового отдела (выбрать конкретное производство).

БД должна осуществлять:

- ведение плановой документации по основному и вспомогательному производствам (план и факт);
- составление заказов на поставку сырья и комплектующих (в соответствии с планом выпуска продукции);
- составление планов работы вспомогательных производств для обеспечения потребностей основного производства;
- подсчёт энергозатрат;
- определение соответствия результатов работы плану (в процентах).

4. БД "Кафедра".

Задача – информационная поддержка учебного процесса и организационной деятельности на кафедре вуза. БД должна содержать учебный план, расписание занятий, списки групп, выпускаемых кафедрой, и списки аспирантов (с руководителями и темами исследований). БД должна обеспечивать составление:

- расписания занятий на семестр (по группам);
- учебного плана (по семестрам) для каждого курса;
- расписания занятий для преподавателей;
- списка телефонов сотрудников;
- нагрузки по часам для преподавателей;
- списка научных кадров по научным направлениям;
- списков студентов-дипломников (по группам и по преподавателям).

5. БД "Библиотека".

Задача – информационная поддержка деятельности научно-технической библиотеки.

БД должна включать два раздела: "Научная литература" и "Журнальные публикации". БД должна обеспечивать:

- ведение автоматизированного учёта выдачи/приёма литературы;
- ведение очередей на литературу (по заказам);
- учёт рейтинга изданий (количество читателей и дата последней выдачи);
- поиск литературы по требуемым разделу, теме, автору, ключевому слову (с заданием интересующего периода);
- составление списков должников по годам.

6. БД "Больница".

Задача – информационная поддержка деятельности регистратуры больницы. БД должна осуществлять:

- учёт поступления пациентов (по отделениям);
- учёт проведённого лечения;
- учёт платных услуг с выдачей счетов на оплату;

- ведение архива выписанных пациентов.

Необходимо предусмотреть определение (по отделениям):

- пропускной способности больницы;
- среднего времени пребывания больных в стационаре;
- наличия свободных мест в палатах (отдельно для мужчин и для женщин);
- количества прооперированных пациентов (из них – с осложнениями и умерших);
- смертности.

7. БД "Магазин" (выбрать конкретный профиль).

Задача – информационная поддержка деятельности магазина выбранного профиля. БД должна осуществлять:

- учёт поставщиков и поставок;
- учёт продаж по отделам;
- подсчёт остатков товаров (по отделам);
- оформление заказов на товары, запасы которых подходят к концу;
- подведение финансовых итогов дня (по отделам и в целом по магазину);
- анализ результативности работы продавцов (для премирования);
- анализ объёмов продаж по дням недели и по месяцам.

8. БД "Поликлиника".

Задача – информационная поддержка деятельности поликлиники. БД должна осуществлять:

- ведение медицинских карт пациентов;
- учёт рецептов, направлений на анализы, процедур;
- учёт платных услуг с выдачей счёта на оплату;
- ведение очередей на приём к специалистам по направлениям лечащих врачей.

Необходимо предусмотреть:

- определение посещаемости отдельных кабинетов (нагрузки врачей);
- подсчёт количества заболевших за день для определения наступления эпидемии.

9. БД адвоката.

Задача – информационная поддержка деятельности адвокатской конторы. БД должна осуществлять:

- ведение списка адвокатов;
- ведение списка клиентов;

- ведение архива законченных дел.

Необходимо предусмотреть:

- получение списка текущих клиентов для конкретного адвоката;
- определение эффективности защиты (максимальный срок минус полученный срок) с учётом оправданий, условных сроков и штрафов;
- определение неэффективности защиты (полученный срок минус минимальный срок);
- подсчёт суммы гонораров (по отдельным делам) в текущем году;
- получение для конкретного адвоката списка текущих клиентов, которых он защищал ранее (из архива, с указанием полученных сроков и статей).

10. БД по недвижимости.

Задача – информационная поддержка деятельности фирмы, занимающейся продажей и арендой жилых и нежилых помещений. БД должна:

- осуществлять ведение списков жилых и нежилых помещений, предназначенных для аренды и/или продажи;
- поддерживать архив проданных и сданных в аренду помещений;
- производить поиск вариантов в соответствии с требованиями клиента.

Необходимо предусмотреть получение разнообразной статистики:

- наличие помещений разных типов;
- изменение цен на рынке;
- уровни спроса и предложения;
- средние показатели (среднее время нахождения помещения в БД (по типам помещений), среднюю стоимость аренды/продажи помещений и т.п.

11. БД "Гостиница".

Задача – информационная поддержка деятельности гостиницы.

БД должна осуществлять:

- ведение списка постояльцев;
- учёт забронированных мест;
- ведение архива выбывших постояльцев за последний год.

Необходимо предусмотреть:

- получение списка свободных номеров (по количеству мест и классу);
- получение списка номеров (мест), освобождающихся сегодня и завтра;
- выдачу информации по конкретному номеру;
- автоматизацию выдачи счетов на оплату номера и услуг;
- получение списка забронированных номеров;

- проверку наличия брони по имени клиента и/или названию организации.

12. БД "Продажа билетов".

Задача – информационная поддержка деятельности транспортных касс (выбрать вид транспорта). БД должна осуществлять:

- ведение списка рейсов и билетов на них с указанием класса;
- учёт забронированных мест;
- ведение архива пассажиров за последний месяц.

Необходимо предусмотреть:

- продажу билетов в оба конца;
- поиск места на рейс в соответствии с требованиями заказчика;
- получение списка свободных мест на рейс;
- выдачу информации по конкретному рейсу;
- получение списка проданных мест;
- проверку наличия брони по имени клиента и/или названию организации.

13. БД "Спортивный клуб".

Задача – информационная поддержка деятельности спортивного клуба. БД должна осуществлять:

- ведение списков спортсменов и тренеров;
- учёт проводимых соревнований (с ведением их архива);
- учёт травм, полученных спортсменами.

Необходимо предусмотреть:

- возможность перехода спортсмена от одного тренера к другому;
- составление рейтингов спортсменов;
- составление рейтингов тренеров;
- выдачу информации по соревнованиям;
- выдачу информации по конкретному спортсмену;
- подбор возможных кандидатур на участие в соревнованиях (соответствующего уровня мастерства, возраста и без травм).

Литература

1. Агальцов В. П. Базы данных / В. П. Агальцов. - 2-е изд., перераб.—Москва : форум : Инфра - М, 2009.
2. Бойко В.В., Савинков В.М. Проектирование баз данных информационных систем. – 2-е изд. – М.: Финансы и статистика, 1989.
3. Голицына О. Л. Базы данных: учебное пособие для вузов / О. Л. Голицына, Н. В. Максимов, И. И. Попов.- М. : Форум : Инфра - М, 2005.
4. Дейт К. Дж. Введение в системы баз данных: Пер. с англ.- 6-е изд.– К.; М.; СПб.: Издательский дом «Вильямс», 2005.
5. Диго С.М. Проектирование и использование баз данных: Учебник. – М.: Финансы и статистика, 2005.
6. Илюшечкин, В. М. Основы использования и проектирования баз данных : учебное пособие / В. М. Илюшечкин. — Москва : Издательство Юрайт : ИД Юрайт, 2011.
7. Карпова И.П. Проектирование реляционных баз данных: учебно-методическое пособие по курсовому проектированию. М.: МГИЭМ, 2005.
8. Коннолли Т., Бегг К., Страчан А. Базы данных: проектирование, реализация и сопровождение. Теория и практика: Пер. с англ.: Учеб. пособие. - 2-е изд.– М.: Издательский дом «Вильямс», 2009.
9. Коннолли Т., Бегг К., Страчан А. Базы данных: проектирование, реализация, сопровождение. Теория и практика, 2-е изд. : Пер. с англ. : Уч. пос. – М.: Изд. дом "Вильямс", 2009.
10. Мейер Д. Теория реляционных баз данных: Пер. с англ. – М.: Мир, 1987.
11. Тиори Т., Фрай Дж. Проектирование структур баз данных : В 2-х кн.: Пер. с англ. – М.: Мир, 1985.
12. Хомоненко А.Д., Цыганков В.М., Мальцев М.Г. Базы данных: Учебник для высших учебных заведений / Под ред. проф. А.Д. Хомоненко. – СПб.: КОРОНА принт, 2007.
13. Хомоненко А.Д., Цыганков В.М., Мальцев М.Г. Базы данных: Учебник для высших учебных заведений / Под ред. проф. А.Д. Хомоненко. – Издание 2-е, доп. и перераб. – СПб.: КОРОНА принт, 2009.

ЧОУ ВПО «Курский институт менеджмента, экономики и бизнеса»

Факультет подготовки бакалавров

Кафедра прикладной информатики и математики

КУРСОВОЙ ПРОЕКТ

по дисциплине «Базы данных»

тема: «_____»

Выполнил: студент (ка) ___ курса группы _____

_____ / Ф.И.О. студента
(личная подпись)

Руководитель:

_____ / Ф.И.О. преподавателя
(личная подпись)

“ ___ ” _____ 20__ г.

Курск – 20__г.