

---

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ  
РОССИЙСКОЙ ФЕДЕРАЦИИ**

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

**«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ  
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ» (ТУСУР)**

Кафедра ЭМИС

Вагнер Дмитрий Петрович

Методические указания по проведению практических и  
самостоятельных работ студентов по курсу  
**«Технология разработки программных средств»**

Томск 2018

Целью практических работ и индивидуальных заданий для самостоятельной работы по курсу «**Технология разработки программных средств**» является закрепление теоретических знаний по курсу и овладение навыками концептуального, логического и физического проектирования программных средств и систем, а также применение CASE-средств для автоматизации разработки готовых систем.

Вагнер Д.П., ст.преподаватель кафедры ЭМИС ТУСУР

## СОДЕРЖАНИЕ

Введение .....	4
ПРАКТИЧЕСКАЯ РАБОТА №1. Инструменты повышения надежности программных средств и систем .....	5
ПРАКТИЧЕСКАЯ РАБОТА №2. Проектирование программных средств и систем на основе источников данных .....	8
ПРАКТИЧЕСКАЯ РАБОТА №3. Технологии концептуального проектирования.....	12
ПРАКТИЧЕСКАЯ РАБОТА №4. Автоматизированное проектирование систем с использованием CASE-средств .....	17
Методические указания по самостоятельной работе.....	19
Список рекомендованной литературы .....	22
Приложение А.....	23

## **Введение**

**Цель практических работ** — овладение навыками концептуального, логического и физического проектирования информационных систем, а также применение CASE-средств для автоматизации разработки готовых систем.

Методические указания к практическим работам по курсу «Методы и средства проектирования информационных систем и технологий» содержат только те понятия и определения, которые необходимы для их успешного выполнения.

## **ПРАКТИЧЕСКАЯ РАБОТА №1.** Инструменты повышения надежности программных средств и систем

**Цель работы:** получение навыков организации резервного копирования и восстановления программных средств и систем в условиях функционирования операционной системы, с использованием стандартных операций, а также при использовании возможностей самих систем.

**Темы для предварительного изучения.** Функции и возможности ОС Windows, возможности СУБД mysql, язык запросов SQL .

### **Постановка задачи**

**Задание 1.1.** Средствами ОС (использование пакетных файлов, скриптов, сервисов, Планировщика заданий) организуйте периодическое **резервное копирование** произвольной программной системы несколькими способами:

- а) Каждая следующая резервная копия заменяет предыдущую;
- б) Каждая следующая резервная копия создается в отдельном каталоге(например, с номерами 1, 2, 3 и т.д. или с текущей датой резервирования 21022012, 22022012 и т.д.), таким образом, чтобы пользователь имел возможность обратиться к любой резервной копии в случае необходимости в дальнейшем;

**Задание 1.2 .** Средствами ОС, а также используя любой из доступных архиваторов, организуйте периодическое **архивирование** каталога информационной системы несколькими способами:

- а) Каждая следующая архивная копия заменяет предыдущую;
- б) Каждая следующая архивная копия создается отдельно(например с названиями 1, 2, 3 и т.д. или 21022012, 22022012 и т.д.), таким образом, пользователь имеет возможность обратиться к любой архивной копии в случае необходимости в дальнейшем;

### **Сведения из теории.**

Для того, чтобы считать в бэкап данные из информационной базы предприятия можно выбрать данные с помощью запросов и сохранить их в файлы текстовых форматов. Для этого используется SQL-команда `SELECT INTO OUTFILE` и парная ей `LOAD DATA INFILE`. Выгрузка производится построчно (можно отобрать для сохранения только нужные строки, как в обычном `SELECT`). Структура таблиц нигде не указывается — об этом должен заботиться программист. Он также должен позаботиться о включении команд `SELECT INTO OUTFILE` в транзакцию, если это необходимо для обеспечения целостности данных. На практике `SELECT INTO OUTFILE` используется для частичного бэкапа очень больших таблиц, которые нельзя скопировать никаким другим образом.

В большинстве случаев намного более удобна утилита `mysqldump`. Утилита `mysqldump` формирует файл, содержащий все SQL-команды, необходимые для полного восстановления БД на другом сервере. Отдельными опциями можно добиться совместимости этого файла с практически любой СУБД (не только MySQL), кроме того, существует возможность выгрузки данных в форматах CSV и XML. Для восстановления данных из таких форматов существует утилита `mysqlimport (source)`.

Утилита `mysqldump` консольная. Существуют её надстройки и аналоги, позволяющие управлять бэкапом через веб-интерфейс, например, `Syrex Dumper`.

Недостатки универсальных утилит бэкапа в текстовые файлы — это относительно невысокая скорость работы и отсутствие возможности делать инкрементные бэкапы.

**Задание 1.3.** Создайте собственную БД `mysql`, в которой будут функционировать не менее 3-х таблиц. Таблицы необходимо заполнить информацией, в каждой таблице должно быть не менее 3 полей и 3 строк.

**Задание 1.4.** С помощью команд `SELECT INTO OUTFILE` и `LOAD DATA INFILE` осуществить резервное копирование данных из любой таблицы БД в файл, затем удалить все данные и восстановить их из бэкапа.

**Задание 1.5.** С помощью команд `mysqldump` и `mysqlimport ( source )` или аналогичных им осуществить резервное копирование БД в файл, затем удалить все данные и восстановить их из бэкапа.

## **ПРАКТИЧЕСКАЯ РАБОТА №2. Проектирование программных средств и систем на основе источников данных**

**Цель работы:** получение навыков работы с распределенными источниками данных для информационной системы, знакомство с функциями диспетчера связанных таблиц, освоение инструментов экспорта и импорта данных из ИС.

**Темы для предварительного изучения.** Распределенные БД. Использование ODBC технологии при взаимодействии с БД.

**Задание 2.1. Создать в MS Access новую БД, в которой необходимо осуществить подключение к базе данных, предварительно созданной в MySQL с помощью ODBC-драйвера.** В БД MySQL создайте всего 1 таблицу произвольной структуры, заполненную 1-2 строками информации. Для подключенной таблицы создайте форму с удобным интерфейсом работы с данными.

Для взаимодействия MS Access с другими источниками данных необходимо воспользоваться вкладкой «Внешние данные» - Раздел «Импорт». Взаимодействие с другими источниками данных можно реализовать с помощью универсального интерфейса доступа к данным ODBC. Для этого предварительно необходимо создать и настроить источник данных (Панель управления – Администрирование – Источники данных(ODBC)) для соответствующей внешней СУБД. Интерфейс настройки соединения с СУБД MySQL приведен на рисунке 1.

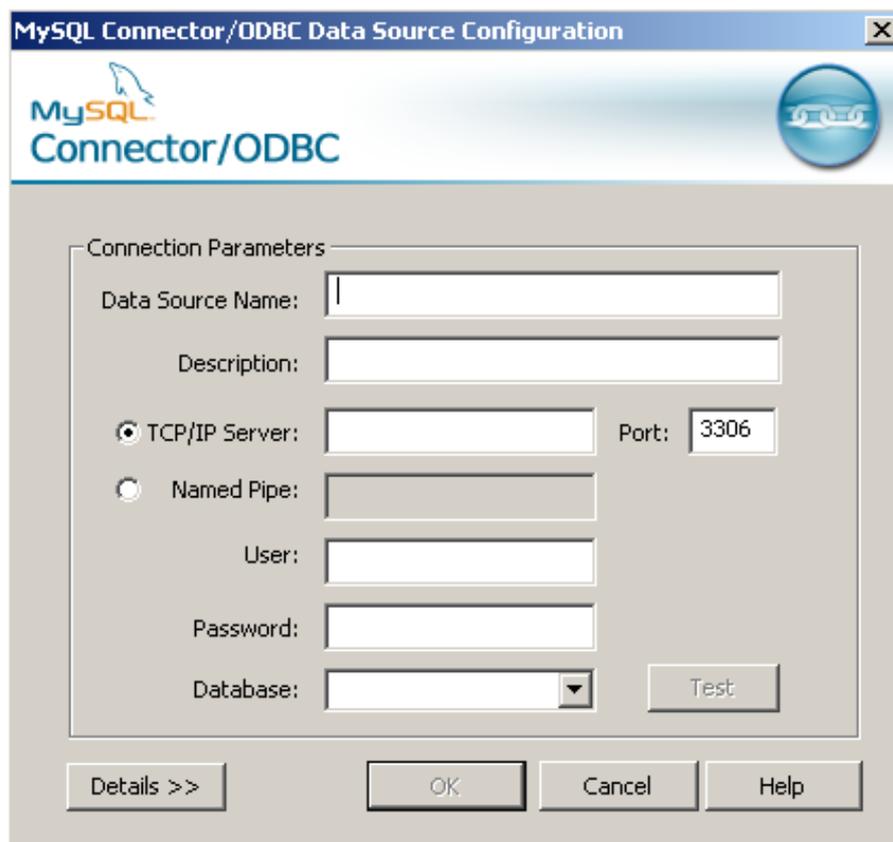


Рис. 1 Настройка соединения с MySQL

**Задание 2.2. Создать Распределенную БД "Успеваемость" в СУБД MS Access по приведенной на рисунке 2 схеме.**

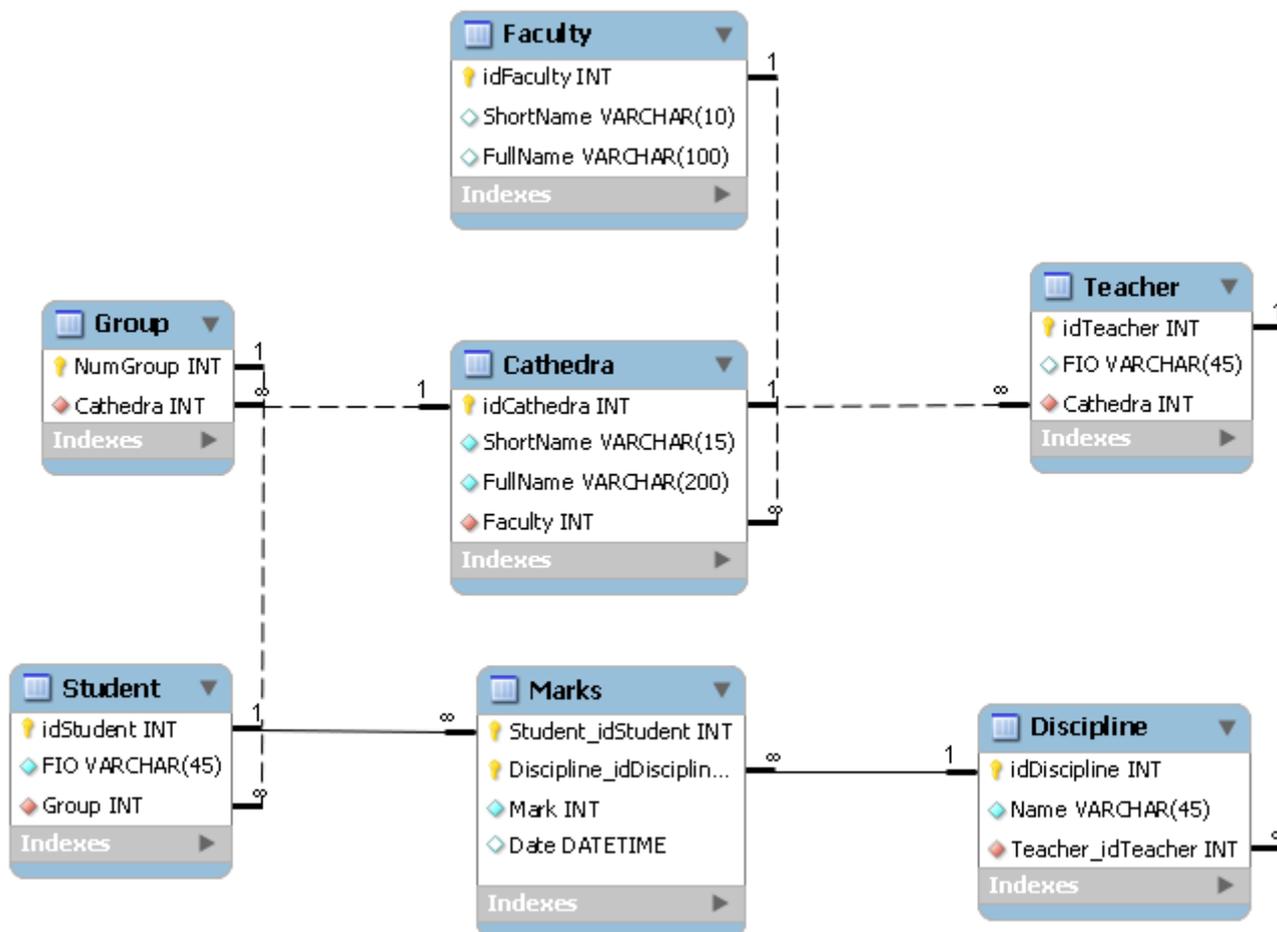


Рис. 2 Схема данных БД «Успеваемость»

При этом создать таблицы в базе данных необходимо следующим образом:

1. Таблицы Student и Group создать в СУБД MySQL с помощью программного комплекса MySql Workbench. Полученные таблицы БД «Students» подключить к исходной БД.

2. Таблицы Teacher и Discipline создать в отдельной БД MS Access «Преподаватели» и подключить через диспетчер связанных таблиц

3. Таблицу Faculty реализовать в файле Excel или текстовом файле Факультеты и подключить через инструмент экспорта с созданием связи.

Оставшиеся таблицы создать в исходной БД.

Через инструмент «Схема данных» настроить связи между таблицами согласно рисунку 2

Таблицы необходимо заполнить информацией, не менее 5 строк в каждую.

**Задание 2.3. Освоить инструменты экспорта данных из БД в различные форматы.**

## **ПРАКТИЧЕСКАЯ РАБОТА №3. Технологии концептуального проектирования**

**Цель работы:** получение навыков работы по концептуальному проектированию информационных систем, практическое освоение основных приемов и правил методологии информационного моделирования;

**Темы для предварительного изучения.** Основные этапы проектирования. Современные средства проектирования и программирования баз данных, освоение способов разработки концептуальной модели и реализация модели средствами СУБД. Нормализация БД.

**Задание 3.1.** Произвести анализ предметной области книжного издательства. Составить концептуальную модель (сущности, атрибуты, связи) и итоговую полноатрибутную ER-диаграмму.

**Введение.** Основная цель системы обработки данных заключается в повышении эффективности работы компании, учреждения или организации. Система обработки данных должна:

- обеспечивать получение общих или детализированных данных по итогам работы;
- позволять легко определять тенденции изменения важнейших показателей;
- обеспечивать получение информации, критической по времени, без существенной задержки;
- выполнять точный и полный анализ данных.

Одной из популярных среди настольных СУБД является Microsoft Access. Основными преимуществами являются: популярность среди многих конечных пользователей и осуществление высокой устойчивости данных, простота в освоении, использовании непрофессиональными программистами, возможность подготавливать отчеты из баз данных различных форматов

произвольной формы на основании различных данных; возможность разработки некоммерческих приложений.

Описание предметной области. База данных создаётся для информационного обслуживания редакторов, менеджеров и других сотрудников компании. БД должна содержать данные о сотрудниках компании, книгах, авторах, финансовом состоянии компании и предоставлять возможность получать разнообразные отчёты.

В соответствии с предметной областью система строится с учётом следующих особенностей:

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

В результате анализа должны быть получены базовые сущности этой предметной области:

- **Сотрудники** компании. Атрибуты сотрудников – ФИО, табельный номер, пол, дата рождения, паспортные данные, ИНН, должность, оклад, домашний адрес и телефоны. Для редакторов необходимо хранить сведения о редактируемых книгах; для менеджеров – сведения о подписанных контрактах.

- **Авторы.** Атрибуты авторов – ФИО, ИНН (индивидуальный номер налогоплательщика), паспортные данные, домашний адрес, телефоны. Для авторов необходимо хранить сведения о написанных книгах.
- **Книги.** Атрибуты книги – авторы, название, тираж, дата выхода, цена одного экземпляра, общие затраты на издание, авторский гонорар.

**Контракты** рассматриваются как связь между авторами, книгами и менеджерами. Атрибуты контракта – номер, дата подписания и участники.

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

Проверочная ER–диаграмма издательской компании приведена в приложении А.

**Задание 3.2.** Полученную модель реализовать в виде схемы БД MS Access путем сопоставления каждой сущности и каждой связи, имеющей атрибуты, отношения (таблицы БД).

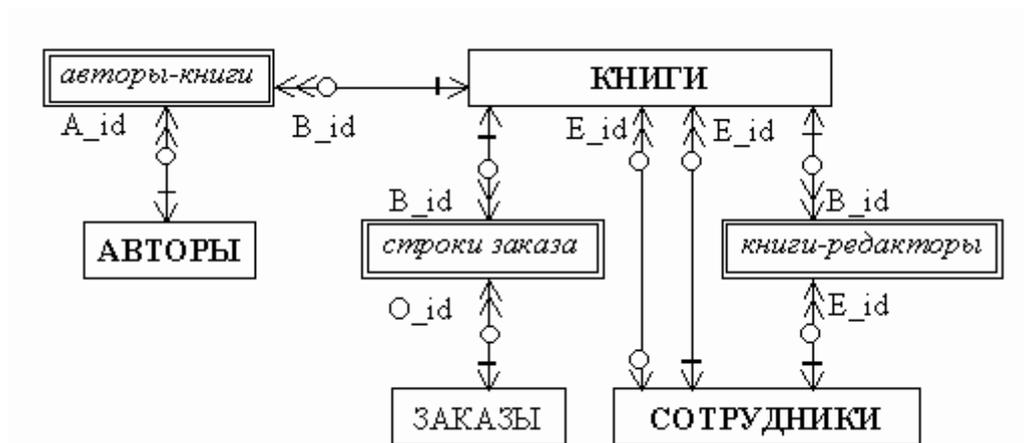


Рис. 3 Диаграмма «Издательская компания»

При создании схемы БД обратить внимание на следующие вопросы:

1. Связь типа 1:1 – обязательная связь, например 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. Исключение для связи типа 1:1 составляют ситуации, когда для увеличения производительности системы в отдельную таблицу выделяются редко используемые данные большого объёма.

2. Связь типа 1:n (один-ко-многим) между отношениями реализуется через внешний ключ. Ключ вводится для того отношения, к которому осуществляется множественная связь. Например, связь «редактировать» между отношениями КНИГИ и СОТРУДНИКИ принадлежит к типу n:m (многие-ко-многим). Этот тип связи реализуется через вспомогательное отношение, которое является соединением первичных ключей соответствующих отношений.

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

Особое внимание необходимо уделить вопросам **нормализации**:

1. **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.

2. **2НФ**. Неключевые атрибуты отношений должны функционально полно зависеть от первичных ключей.

3. **ЗНФ.** В отношении *ЗАКАЗЫ* атрибут *Адрес заказчика* не должен зависеть от атрибута *Заказчик*, 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 ПК. Так как каждый заказчик может сделать несколько заказов, связь между отношениями *ЗАКАЗЧИКИ* и *ЗАКАЗЫ* будет 1:n и суррогатный ПК станет внешним ключом для отношения *ЗАКАЗЫ*. Аналогично необходимо отследить зависимости и между другими отношениями в БД

**Задание 3.3.** Аналогично заданиям 3.1-3.2 произвести проектирование ИС (выделить базовые сущности, связи между ними, составить ER-дигramму), разработать схему БД в соответствии с вариантом выданным преподавателем.

Варианты для задания 3.3:

1. Информационная система библиотеки
2. Информационная система ВУЗа
3. Информационная система швейного производства
4. Информационная система ресторана
5. Информационная система больницы
6. Информационная система склада
7. Информационная система зоопарка
8. Информационная система аэропорта
9. Информационная система аптеки
10. Информационная система автомастерской
11. Информационная система школы
12. Информационная система фотоцентра

## **ПРАКТИЧЕСКАЯ РАБОТА №4.** Автоматизированное проектирование систем с использованием CASE-средств

**Цель работы:** получение навыков работы с современными методами и средствами автоматизированного проектирования информационных систем на примере программного средства MySQL Workbench.

**Темы для предварительного изучения.** Программные средства поддержки жизненного цикла ПО. Оценка и выбор CASE-средств. Применение CASE-средств для автоматизации процессов проектирования систем.

**Задание 4.1.** Знакомство с назначением CASE-технологии на примере MySQL Workbench, предназначенного для построения логических и физических моделей предметных областей, проведения анализа и генерации готовых БД.

Для создания моделей данных в MySQL Workbench можно использовать несколько нотаций: например, IDEF1X и IE (Information Engineering). В данной работе будет использоваться нотация IDEF1X. Для внесения сущности в модель необходимо кликнуть по кнопке сущности на панели инструментов, затем кликнуть по тому месту на диаграмме, где Вы хотите расположить новую сущность. Кликнув правой кнопкой мыши по сущности и выбрав из всплывающего меню пункт Entity Editor... можно вызвать диалог Entity Editor, в котором определяются имя, описание и комментарии сущности.

Каждый атрибут хранит информацию об определенном свойстве сущности. Каждый экземпляр сущности должен быть уникальным. Атрибут или группа атрибутов, которые идентифицируют сущность, называется первичным ключом. Для описания атрибутов следует, кликнув правой кнопкой по сущности, выбрать в появившемся меню пункт Attribute Editor.

Для установки связи между сущностями нужно воспользоваться кнопками в палитре инструментов. На логическом уровне можно установить идентифицирующую связь один ко многим, связь многие ко многим и неидентифицирующую связь один ко многим (соответственно кнопки - слева направо в палитре инструментов). Идентифицирующая связь устанавливается между независимой (родительский конец связи) и зависимой (дочерний конец связи) сущностями. Зависимая сущность изображается прямоугольником со скругленными углами. Экземпляр зависимой сущности определяется только через отношение к родительской сущности. При установлении идентифицирующей связи атрибуты первичного ключа родительской сущности переносятся в состав первичного ключа дочерней сущности (миграция атрибутов). В дочерней сущности они помечаются как внешний ключ - (FK). При установлении неидентифицирующей связи дочерняя сущность остается независимой, а атрибуты первичного ключа родительской сущности мигрируют в состав неключевых компонентов родительской сущности.

**Задание 4.2.** Разработать концептуальную модель издательства из задания 3.2. и отобразить эту модель в среде MySQL Workbench.

**Задание 4.3.** Сгенерировать полученную модель в реальную СУБД mysql. Изучить особенности генерации SQL-кода.

**Задание 4.4.** С помощью CASE-средства MySQL Workbench осуществить проектирование предметной области из задания 3.4 в соответствии с вариантом задания, выданным преподавателем.

## **Методические указания по самостоятельной работе**

1. Математические и методологические аспекты проектирования информационных систем. (14 часов)

2. Унифицированный язык визуального моделирования(UML). Этапы проектирования ИС с использованием UML. (25 часов)

3.Проектирование пользовательского интерфейса систем. Проектирование оконного и web-интерфейсов (12 часов)

4. Проектирование распределенных информационных систем (14 часов)

5. Технологии проектирования экономических информационных систем. (18 часов)

6. Использование методологии RAD при проектировании систем (14 часов)

## **ТЕОРЕТИЧЕСКАЯ ЧАСТЬ**

**Тема 1.** Математические и методологические аспекты проектирования информационных систем.(14 часов)

### **Вопросы для рассмотрения**

1. Модели выбора проектных решений
2. Разработка модели системы на основе сетей Петри

**Тема 2.** Унифицированный язык визуального моделирования(UML). (25 часов)

### **Вопросы для рассмотрения**

1. Особенности разработки ИС с использованием UML.
2. Этапы проектирования ИС с использованием UML.

**Тема 3.** Проектирование пользовательского интерфейса систем. (12 часов)

**Вопросы для рассмотрения**

1. Принципы проектирования интерфейса, ориентированного на пользователя.
2. Проектирование web-интерфейса.

**Тема 4.** Планирование и управление проектами с использованием MS Project. (14 часов)

**Вопросы для рассмотрения**

1. Этапы планирования проекта в MS Project
2. Основные бизнес-процессы.
3. Планирование ресурсной базы проекта.

**Тема 5.** Технологии проектирования экономических информационных систем. (18 час)

**Вопросы для рассмотрения**

1. Стадии и этапы проектирования ЭИС.
2. Особенности проектирования корпоративных ЭИС.
3. Особенности разработки ЭИС в среде 1С.

**Тема 6.** Использование методологии RAD при проектировании систем (14 часов)

**Вопросы для рассмотрения**

1. Особенности методологии RAD
2. Отличия ЖЦ ИС при использовании RAD
3. Примеры использования методологии.

## **ФОРМА КОНТРОЛЯ**

1. Сообщения в форме докладов, индивидуальный опрос.
2. Индивидуальные отчеты, опрос по принципу коллоквиумов.

## Список рекомендованной литературы

1. Гвоздева Т.В., Баллод Б.А. Проектирование информационных систем, Издательство: Феникс, 2009 г, 512 стр.
2. Вендров А.М. Практикум по проектированию программного обеспечения экономических информационных систем. – М.: Финансы и статистика, 2002.
3. Грекул В.И. Проектирование информационных систем. Курс лекций : Учебное пособие для вузов / В. И. Грекул, Г. Н. Денищенко, Н. Л. Коровкина. - М. : Интернет-Университет Информационных Технологий, 2005. - 298с. : ил.
4. Карпова Т.С. Базы данных: модели, разработка, реализация : учебное пособие / Т. С. Карпова. - СПб. : Питер, 2002. - 303[1] с. : ил.
5. Кузнецов М.В. MySQL 5 / М.В, Кузнецов, И.В. Симдянов. – СПб.: БХВ-Петербург, 2010. – 1024 с.:ил.
6. Ковалев С.П. Формальный подход к разработке программных систем: Уч.пособие / НГУ. Новосибирск, 2004. 180с.
7. Ларман К. Применение UML и шаблонов проектирования. Пер с англ. - М.: Вильнюс, 2001.
8. Боровской И.Г., Колесникова С.И., Матолыгин А.А. Специализированная подготовка разработчиков бизнес-приложений. Уч.пособие. Томск: ТУСУР, 2007. – 267с.
9. Пирогов В.Ю. Информационные системы и базы данных: организация и проектирование: учеб. пособие. – СПб.: БХВ-Петербург, 2009. – 528 с.:ил.

**Пример ER-диаграммы издательства**

