

П.В. Сенченко

БАЗЫ ДАННЫХ

МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ВЫПОЛНЕНИЮ КУРСОВОГО ПРОЕКТА

для студентов направления подготовки бакалавров
231000.62 – Программная инженерия
080500.62 – Бизнес-информатика
**081100.62 – Государственное и муниципальное
управление»**

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
**ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)**

П.В. Сенченко

БАЗЫ ДАННЫХ

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ
К ВЫПОЛНЕНИЮ КУРСОВОГО ПРОЕКТА**

для студентов направления подготовки бакалавров
231000.62 – Программная инженерия
080500.62 – Бизнес-информатика
**081100.62 – Государственное и муниципальное
управление»**

Томск 2015

П.В. Сенченко.

Базы данных: Методические указания к выполнению курсового проекта. — Томск: Томск. гос. ун-т систем управления и радиоэлектроники, 2014. — 27 с.

Цель учебно-методического пособия к курсу Базы данных – методическое сопровождение выполнения курсового проекта по итогам изучения дисциплины.

Учебно-методическое пособие предназначено для студентов направления подготовки бакалавров 231000.62 – Программная инженерия, 080500.62 – Бизнес-информатика, 081100.62 – Государственное и муниципальное управление» а также студентов родственных специальностей и направлений, сотрудников и специалистов, занимающихся разработкой баз данных.

СОДЕРЖАНИЕ

Введение	4
Указания к выполнению курсового проекта	5
Рекомендуемая литература	25
Приложение. Форма титульного листа пояснительной записки	26

Введение

Выполнение курсового проекта направлено на аккумулирование приобретенных навыков разработки баз данных.

Процесс изучения дисциплины направлен на формирование следующих компетенций:

- владение основными способами и средствами информационного взаимодействия, получения, хранения, переработки, интерпретации информации, наличием навыков работы с информационно-коммуникационными технологиями; способностью к восприятию и методическому обобщению информации, постановке цели и выбору путей ее достижения (ОК-8);
- умение обобщать и систематизировать информацию для создания баз данных, владением средствами программного обеспечения анализа и моделирования систем управления (ПК-17);
- владение технологиями защиты информации (ПК-27);
- способность осуществлять технологическое обеспечение служебной деятельности специалистов (по категориям и группам должностей государственной гражданской службы и муниципальной службы) (ПК-46).

По итогам создания курсового проекта студенты с учетом полученных теоретических знаний должны:

Уметь:

- построить концептуальную информационную модель предметной области в концепции баз данных;
- проектировать реляционную модель данных для выбранной предметной области с использованием нормализации;
- производить моделирование предметной области, уметь строить для нее ER-диаграмму и отображать ER-диаграмму в схему реляционной базы данных;
- реализовать простые информационные технологии с использованием функциональных возможностей современных СУБД;
- разрабатывать программные объекты для работы с базами данных: экранные формы, отчеты, разрабатывать все виды запросов при помощи построителей запросов.
- разрабатывать запросы на языке SQL;
- строить индексы;
- обеспечивать защиту данных средствами СУБД;

Владеть:

- методикой проектирования баз данных на основе нормализации отношений.
- средствами разработки баз данных и простых элементов пользовательского интерфейса в современных СУБД.
- методикой проектирования баз данных на основе разработки ER-модели предметной области.
- как минимум одним средством автоматизированного проектирования ER-диаграмм (Power Designer, Erwin и др.).

Указания к выполнению курсового проекта

Цель выполнения проекта:

- освоение методики проектирования концептуальной информационной модели предметной области преобразование концептуальной модели в физическую структуру базы данных (БД);
- закрепление теоретических знаний по курсу организация баз данных.

Задачи курсового проекта:

- формализовать исходное описание предметной области;
- построить концептуальную информационную модель, используя методику, изученную в рамках теоретического курса;
- сгенерировать физическую структуру базы данных;
- реализовать автоматизированную информационную систему (далее АИС, пользовательское приложение), демонстрирующую накопленные студентом знания по дисциплине.

Средства выполнения и форма отчетности:

проектирование предметной области выполняется средствами автоматизированного проектирования и графического представления (Power Designer, ERwin, MS Visio и др.). При отсутствии средств моделирования концептуальную и физическую модель необходимо реализовать средствами MS Word. База данных разрабатывается в среде любой современной СУБД (MS Access, Oracle, MS SQL, MySQL и др.), пользовательское приложение может быть создано либо средствами выбранной СУБД, либо с помощью любых языков программирования высокого уровня (Delphi, Visual Basic, C# и др.).

Результаты выполнения работы представляются в пояснительной записке, подготовленной в среде MS Word. Пользовательское приложение и базу данных необходимо представить вместе с пояснительной запиской к курсовому проекту.

Варианты индивидуального задания

Таблица 1

№	Название предметной области АИС
1.	Библиотека
2.	Магазин продовольственных товаров
3.	ВУЗ
4.	Супермаркет
5.	Документооборот предприятия
6.	Агентство недвижимости
7.	Компьютерная фирма
8.	Поликлиника
9.	Турфирма
10.	Гостиница
11.	Автосалон
12.	Банк
13.	Деканат
14.	Отдел кадров
15.	Аэропорт

Порядок выполнения работы:***1. Создание концептуальной информационной модели предметной области***

Каждый студент получает для работы предметную область (Таблица 1).

Концептуальная модель представляется в виде набора ER-диаграмм. Осуществляется формализация исходного описания в виде набора сущностей с последующим их преобразованием и связыванием в концептуальную модель.

Процесс проектирования сопровождается составлением ряда сущностей, необходимыми пояснениями – обоснованиями принимаемых решений

Проектирование концептуальной модели предметной области целесообразно производить с помощью специального средства проектирования: Power Designer.

Основные этапы проектирования концептуальной модели:

1. Первичный анализ информационных потребностей пользователей, выделение объектов предметной области и формирование исходных сущностей:
 - анализ информационных документов;

- анализ конкретных информационных потребностей (запросов) пользователей.
- 2. Проектирование исходных сущностей:
 - определение атрибутов сущностей и их типов данных;
 - нормализация сущностей до 3 НФ.
- 3. Связывание сущностей в концептуальную информационную модель:
 - определение уникальных идентификаторов сущностей (первичных ключей);
 - определение связей между сущностями.

Ограничения концептуальной модели:

- предметная область должна быть описана 8-10 взаимосвязанными сущностями;
- каждая сущность должна содержать не менее 3 атрибутов;
- в каждой сущности должен быть определен уникальный идентификатор сущности.

2. Создание физической модели данных

На основе спроектированной концептуальной модели создается физическая модель данных, свойственная для конкретной СУБД.

При формировании физической модели для определенной СУБД в модели определяются внешние ключи в связываемых сущностях. Добавляются промежуточные таблицы связи, с целью исключения связей многие-ко-многим (М:М), конкретизируются типы данных атрибутов.

Power Designer позволяет произвести автоматическую генерацию физической модели на основе созданной ранее концептуальной. Модели.

3. Указание к практическому использованию Power Designer

Для запуска пакета Power Designer в меню программы (Windows) найдите папку Sybase и запустите файл Power Designer. Для создания концептуальной модели данных необходимо выбрать File/ New или на панели инструментов выбрать значок . Далее появится окно для выбора создаваемой модели (рис. 1), в котором надо выбрать Conceptual Data Model

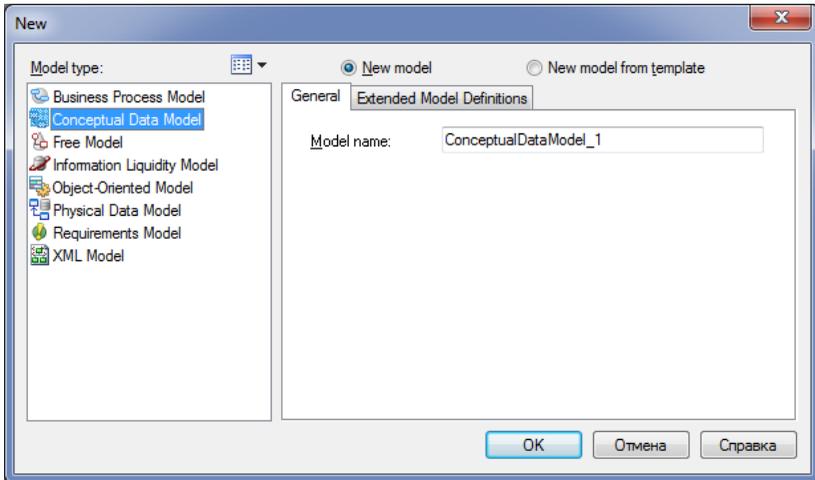


Рис. 1. Окно выбора модели.

После нажатия кнопки ОК появится окно, в котором создается ER-диаграмма.

Создание сущностей

Для создания сущности, в панели Palette (рис. 2) нажмите кнопку с белым прямоугольником (с подсказкой Entity).



Рис. 2. Панель элементов с выбранным элементом сущность

Далее, поместите указатель мыши на рабочее поле в нужном месте и щелкните кнопкой мыши. Прямоугольник, изображающий сущность появится в указанном месте. При этом, курсор мыши на рабочем поле выглядит как выбранный элемент, т.о. можно создавать несколько

ко выбранных элементов одного типа без повторного их выбора на панели элементов.

Для того, чтобы изменить свойства созданной сущности, дважды щелкните на нее левой кнопкой мыши или нажмите правую кнопку и в выпавшем меню, выберите пункт Properties, в результате чего откроется окно свойств сущности (рис. 3).

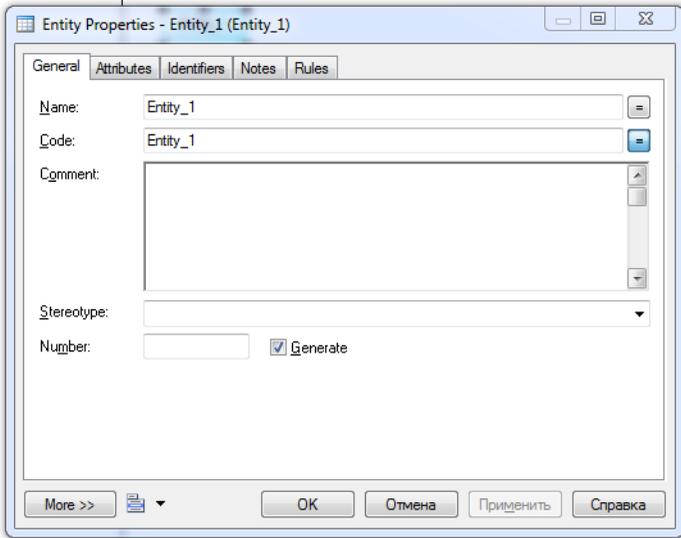


Рис. 3. Окно свойств сущности

В открывшемся окне пять закладок.

Закладка General позволяет ввести следующие основные параметры:

- Name — имя сущности, которое будет видеть пользователь;
- Code — имя кода сущности, которое будет использоваться при генерации физической модели;
- Number — ограничение количества записей в таблице после генерации физической модели;
- Comment — комментарий, предназначенный для улучшения понимания модели.

- P (Primary Identifier) — первичный идентификатор сущности (в физической модели данных атрибут будет являться первичным ключом или его составной частью);
- D (Displayed) — отображаемый, т.е. будет ли атрибут показываться в модели.

Более полную информацию по свойствам атрибута можно получить, дважды щелкнув по полю, расположенному слева от поля с именем атрибута. Здесь можно вводить комментарий в поле Comment, задавать список значений для данного атрибута, определять верхние и нижние границы значений

Закладка Identifiers содержит автоматически заполняемую таблицу первичных идентификаторов сущности, но позволяет делать это вручную, когда необходимо создать суррогатный первичный идентификатор.

Закладка Rules позволяет вводить необходимые правила на ввод значений в таблицу.

Остальные закладки Notes, Version info носят описательный характер для улучшения понимания модели.

Для фиксации всех изменений в необходимо нажать кнопку Apply.

Домен

Домен – это множество допустимых значений атрибута определенного типа данных. Домен определяется заданием стандартного типа данных, к которому относятся элементы домена и заданием произвольного логического выражения, применяемому к этому типу данных.

Для создания домена необходимо в главном меню выбрав пункт Model, выбрать пункт Domains. На рис. 5 показано форма определения домена. Данная форма содержит аналогичные поля как для определения свойств атрибутов (в закладке Attributes в свойствах формы), а также дополнительные поля Length и Precision для описания длины и точности значений атрибутов.

При создании атрибутов сущностей в концептуальной модели не создаются атрибуты, являющиеся внешними ключами сущностей.

После того, как созданы все необходимые сущности и атрибуты, необходимо определить связи между ними.

Установка связей

Для установки связи между двумя сущностями, необходимо нажать кнопку (с двумя белыми прямоугольниками и линией между ними рис. 7).



Рис. 7. Панель элементов с выбранным элементом связь

Необходимо перевести курсор мыши на одну сущность и, нажав левую кнопку мыши и, не отпуская ее, перевести курсор на вторую сущность. Далее можно отпустить кнопку мыши – связь установлена.

Для изменения свойств связи, необходимо дважды щелкнуть левой кнопкой мыши на линию связи (или нажать правую кнопку мыши и выпавшем меню выбрать пункт Properties). Откроется окно (рис. 8), с закладками:

- закладки Notes и Version info используются для подробного описания связи. В закладке Rules можно задавать параметры ограничения связи;
- в закладке General указывается имя и код связи, а две кнопки с именами используемых сущностей позволяют вызвать окно со свойствами соответствующей сущности.

- Закладка Cardinalities позволяет указать вид связи (один–к–одному, один–ко–многим, многие–ко–многим и т.д.) и устанавливает свойства связи от Сущности1 к Сущности2 и наоборот:
- Mandatory определяет обязательность связи, показывая, что экземпляр Сущности1 (запись) может существовать только при наличии соответствующего экземпляра в Сущности2;

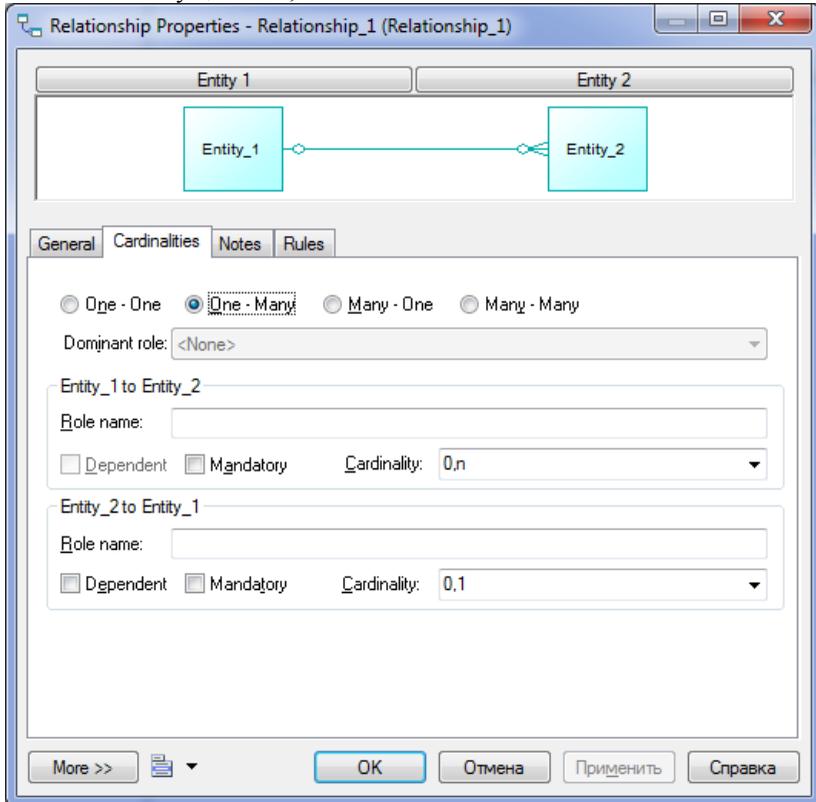


Рис. 8. Окно свойств связи

- Dependent показывает, что каждый экземпляр Сущности1 отождествляется с экземпляром в Сущности2 (первичный ключ на стороне «один» при создании физической модели войдет в состав первичного ключа в таблице на стороне «многие»);

- Role – текст, описывающий связь от Сущности1 к Сущности2.

Связи многие-ко-многим преобразуются в физической модели в промежуточные таблицы.

После того, как созданы все сущности, указаны атрибуты и установлены все связи необходимо проверить концептуальную правильность построения концептуальной модели. Для этого необходимо выбрать в меню Tools/Check Model (или нажать F4). Появится окно (рис. 9), в котором предлагается выбрать объекты для проверки.

Package – система проверит правильность циклических связей.

Domain – система проверит правильность заполнения доменов.

Data items – проверять ли атрибуты.

Entities – система проверит правильность создания сущностей.

Entity attributes – проверка правильности свойств сущности

Entity identifier - проверка правильности идентификаторов сущности.

Relationships – проверка связей.

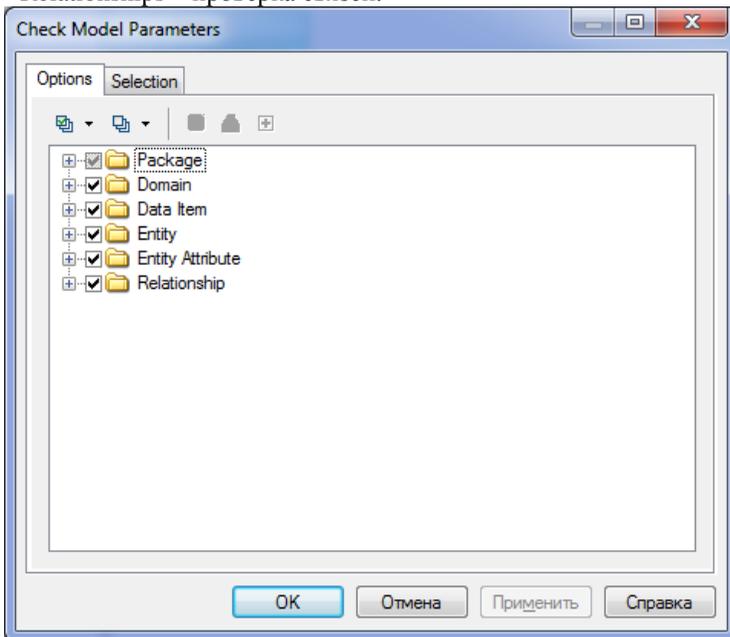


Рис. 9. Окно проверки концептуальной модели

После нажатия кнопки ОК система проверит всю концептуальную модель, выдаст ошибки (или предупреждения), если таковые

имеются. Для просмотра сведений об ошибке необходимо дважды «щелкнуть» по ней кнопкой мыши.

На основе спроектированной концептуальной модели создается **физическая модель данных**, свойственная для конкретной СУБД.

При формировании физической модели данных определяются внешние ключи в связываемых сущностях. Добавляются промежуточные таблицы связи, с целью исключения связей многие-ко-многим (М:М).

После того, как проверка концептуальной модели закончится успешно, можно генерировать физическую модель. Для этого необходимо выбрать в меню Tools/Generate Physical Model. Откроется окно генерации физической модели (рис. 10).

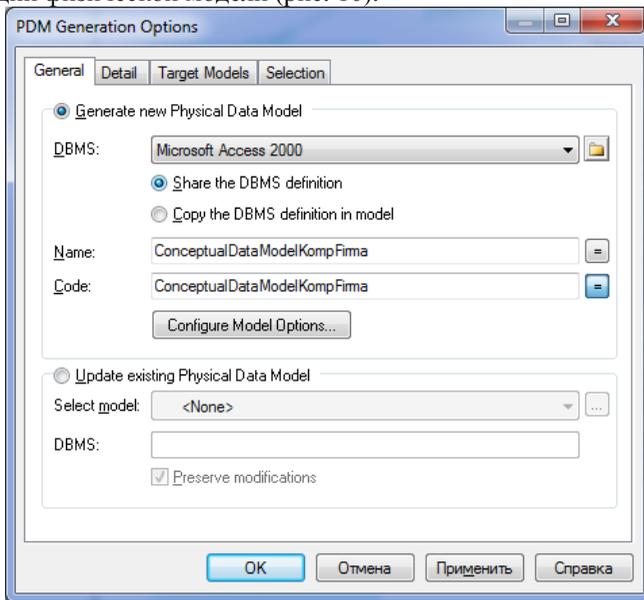


Рис. 10. Окно генерации физической модели

В этом окне необходимо выбрать опцию Generate new Physical Data Model, и выбрать в поле DBMS из выпадающего списка Microsoft Access 2000. Name – поле для ввода имени файла для дальнейшей генерации физической модели. Для задания свойств модели щелкните по кнопке Configure Model Options. Откроется окно свойств создаваемой физической модели (рис. 11).

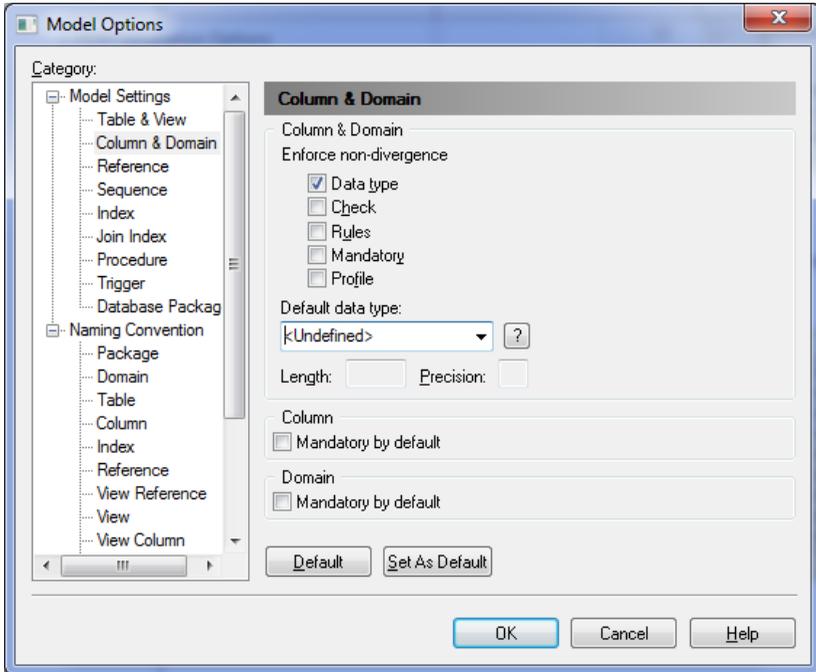


Рис. 11 Окно свойств создаваемой физической модели

Здесь можно выбрать следующие параметры создания физической модели: окно свойств создаваемой физической модели

- Data type – учитывать типы данных концептуальной модели при генерации физической модели;
- Check – проверка доменов и соответствие полей атрибутов с выбранным доменом.
- Rules – проверка правил на ввод значений в таблицу.
- Mandatory – учитывать свойства обязательности заполнения.
- Default data type – позволяет установить тип данных по умолчанию для всех не установленных типов данных в атрибутах.
- Domain – использовать ссылки на домен.

В директории Naming convention можно задавать имена шаблонов таблиц, атрибутов и т.д.

Установив необходимые параметры, нажмите кнопку ОК. Если концептуальная модель спроецирована корректно, то система создаст

физическую модель базы данных для того типа СУБД, который был указан в параметрах.

Сгенерированную физическую модель также необходимо проверить, нажав клавишу F4

Появится окно (рис. 12), в котором предлагается выбрать объекты для проверки.

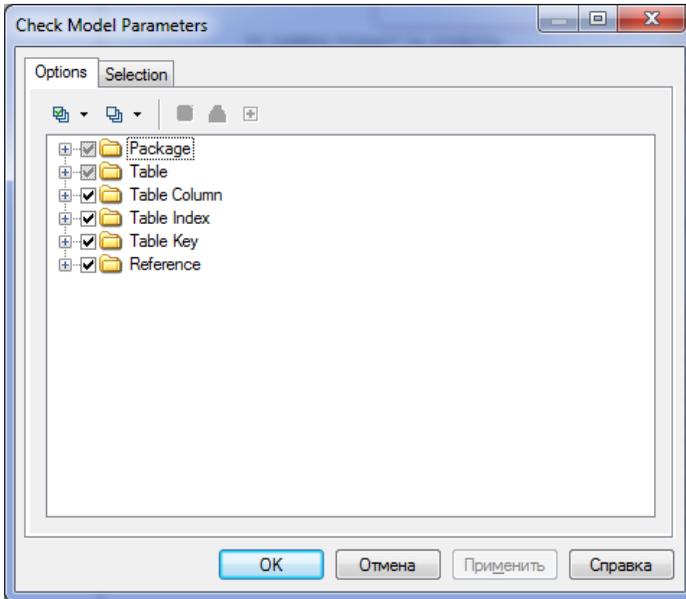


Рис. 12. Окно проверки физической модели

После нажатия кнопки ОК система проверит всю физическую модель, выдаст ошибки (или предупреждения), если таковые имеются.

Если ошибки отсутствуют, на основе данной модели необходимо создать новую базу данных.

Для создания базы данных в меню DataBase выберите пункт Generate DataBase. В появившемся окне (рис. 13) необходимо выбрать опцию Direct generation, путь к файлу, в котором будет сохранен скрипт на создание БД (набор управляющих SQL-запросов), а также дополнительные характеристики БД, перейдя по вкладкам Options, Format, Selection. После чего нажмите кнопку ОК.

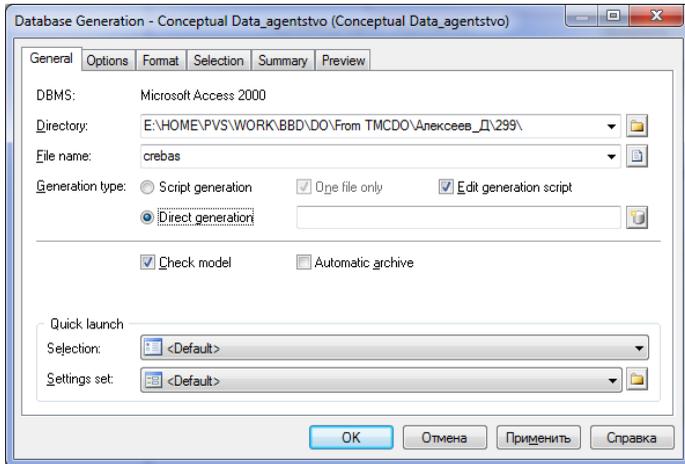


Рис. 13. Окно создания БД

Система попросит указать источник данных (рис. 14).

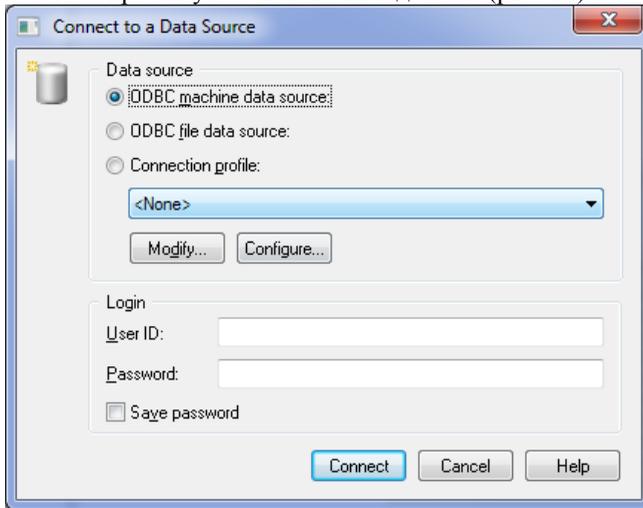


Рис. 14. Окно доступа к БД

Нажмите кнопку **Configure** для создания собственного источника данных. В появившемся окне «Администратор источников данных ODBC» (рис. 15) нажмите кнопку **Добавить**.

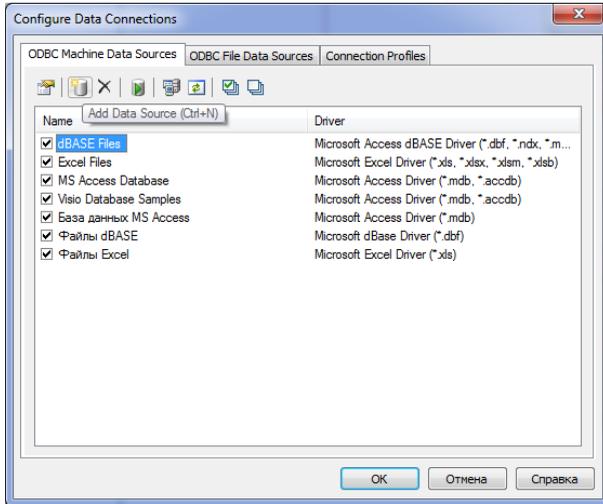


Рис. 15. Окно конфигурации ODBC

В появившемся окне «Создание нового источника» (рис. 16) выберите пользовательский тип источника данных и нажмите кнопку Далее.

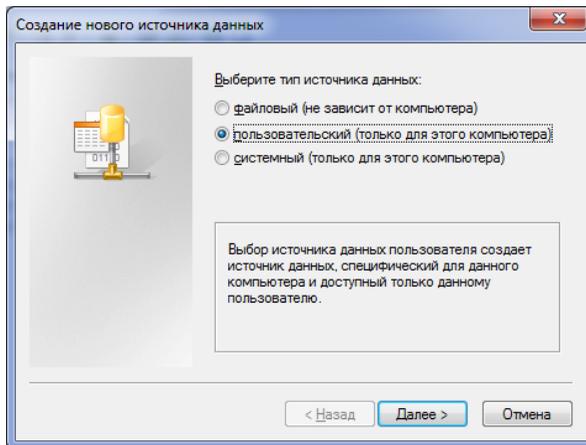


Рис. 16. Окно выбора типа источника данных

В следующем окне (рис. 17) выберите необходимый драйвер ODBC и нажмите кнопку Далее.

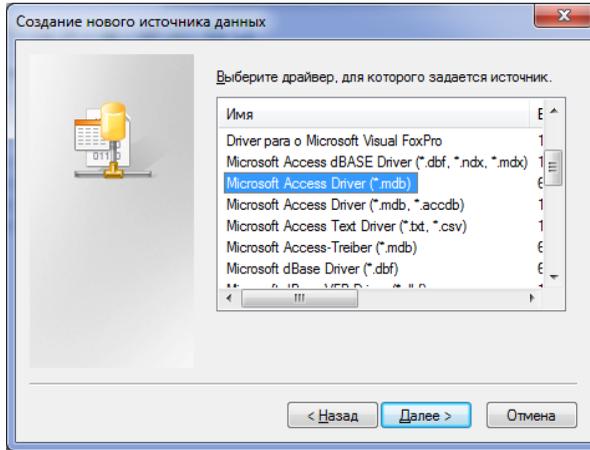


Рис. 17. Окно выбора драйвера ODBC

В следующее окне (рис. 18) нажмите кнопку Готово и задайте имя источника данных и путь в существующему или к новому mdb-файлу (рис. 19).

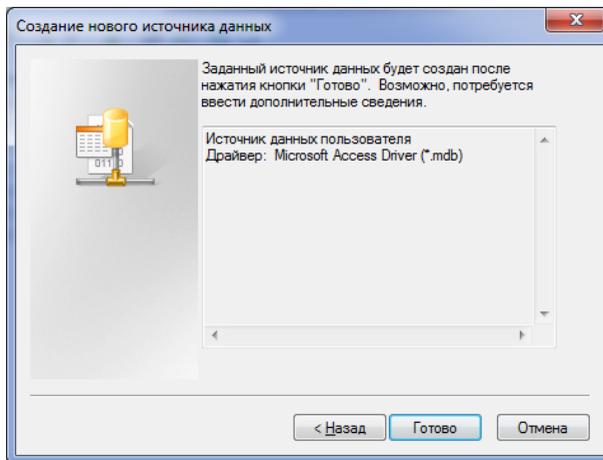


Рис. 18. Подтверждение выбора драйвера ODBC

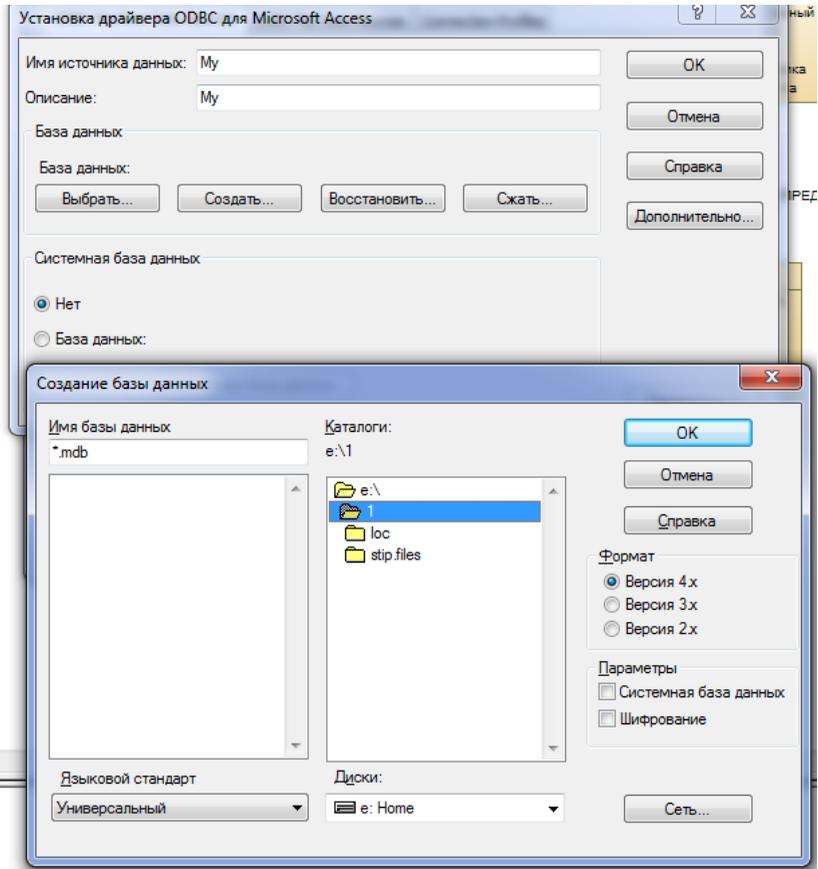


Рис. 19. Подключение к базе данных

Вернитесь в окно доступа к БД (рис. 14) и выберите созданный источник данных (рис. 20).

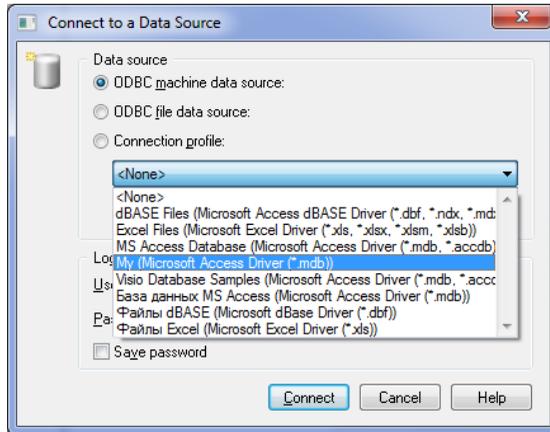


Рис. 20. Выбор своего источника данных

Нажмите кнопку Connect. В появившемся окне (рис 21) будет представлен SQL-скрипт, сгенерированный системой, запуск которого приведет к созданию схемы БД в выбранном mdb-файле.

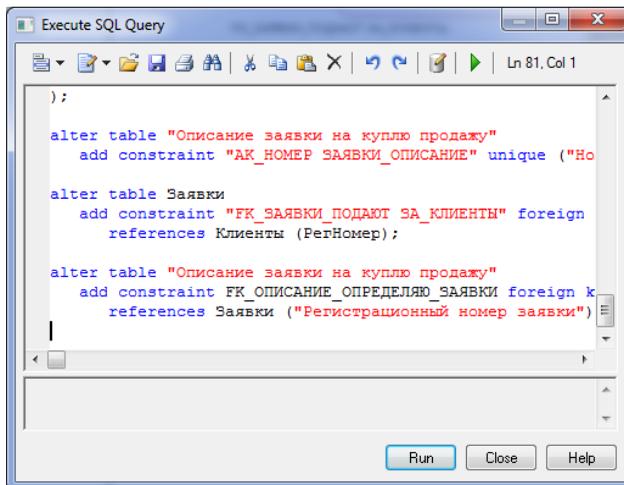


Рис. 21. Окно SQL-скрипта на создание схемы БД

Для запуска SQL-скрипта нажмите кнопку Run. Если система не выдаст сообщений об ошибках, откройте в среде MS Access созданную БД и проверьте ее соответствие физической модели.

4. Создание пользовательского приложения

Приложение, работающее с созданной базой данных должно обеспечивать выполнение следующих функций:

- ввод информации в БД;
- удаление информации из БД;
- редактирование внесенной информации;
- выборка (поиск) данных по таблицам БД с использованием различных критериев;
- формирование отчетов и вывод информации из базы данных на экран и на принтер;

Добавление, замена и удаление информации должны производиться в экранных формах разрабатываемого пользовательского приложения.

5. Оформление пояснительной записки

Пояснительная записка оформляется согласно действующим стандартам вуза ОС ТУСУР 01-2013 (СТО 02069326.1.01-2013) и ОС ТУСУР 02-2013 (СТО 02069326.1.02-2013). Стандарты в электронном виде представлены на образовательном портале ТУСУРа в разделе «Нормативные акты ТУСУРа».

Пояснительная записка к курсовому проекту должна включать:

- титульный лист,
- содержание,
- введение,
- основную часть,
- заключение,
- список использованных литературных источников,
- приложения.

Титульный лист оформляется согласно действующим стандартам вуза, пример оформления титульного листа пояснительной записки к курсовому проекту представлен в Приложении).

Введение должно содержать цель выполняемой курсовой работы, основные принципы, положенные в основу ее проведения, область применения.

В *основной части* должен быть отражен процесс и результат проектирования базы данных и пользовательского приложения. Основная часть должна содержать:

- описание предметной области;
- описание и обоснование выбранного средства реализации (СУБД, средства проектирования, программной среды написания приложения);

- концептуальную информационную модель предметной области с полным описанием выделенных сущностей;
- физическую модель и схему базы данных;
- описание пользовательского приложения.

Заключение должно содержать краткие выводы по результатам выполненной работы.

Список использованных литературных источников оформляется согласно действующим стандартам.

В приложениях приводятся: экранные формы пользовательского приложения, тексты SQL-запросов, создаваемых в приложении и другая сопроводительная информация.

Рекомендуемая литература

1. Сенченко, П.В. Организация баз данных: Учебное пособие/ П. В. Сенченко; Федеральное агентство по образованию, Томский государственный университет систем управления и радиоэлектроники. – Томск: ТУСУР, 2004. – 184 с.: ил. – Библиогр.: с. 183-184. – ISBN 5-86889-224-0: УДК 681.3.016(075.8) (гриф СИБРУМЦ) (Наличие в библиотеке ТУСУР: экземпляры всего: 34, из них: анл: 6 , счз1: 1 , счз5: 1, аул: 26.)

2. Дейт К. Дж. Введение в системы баз данных: Пер. с англ./ К. Дж. Дейт. – 6-е изд. – Киев; М.: Диалектика, 1998. – 784 с.: ил. – (Системное программирование). – (в пер.): Б.ц. (наличие в библиотеке ТУСУР: АНЛ – 1 экз.)

3. Саймон, Алан Р. Стратегические технологии баз данных: менеджмент на 2000 год: Пер. с англ./ Алан Р. Саймон; Ред. М. Р. Когаловский, Пер. М. Р. Когаловский, Пер. Н. И. Вьюкова, Пер. Г. Т. Никитина. – М.: Финансы и статистика, 1999. – 480 с.: ил. (наличие в библиотеке ТУСУР: счз1(1), счз5(1))

4. Кузнецов С.Д. Основы современных баз данных. Информационно-аналитические материалы Центра Информационных технологий. М.– режим доступа к сайту <http://citforum.ru/database/osbd/contents.shtml> свободный (дата обращения: 28.02.2012)

5. Кириллов В.В. Основы проектирования реляционных баз данных. Учебное пособие режим доступа к сайту <http://citforum.ru/database/dbguide/index.shtml> свободный (дата обращения: 28.02.2012)

Приложение. Форма титульного листа пояснительной записки

Наименование министерства (ведомства), в систему которого входит вуз, – строчными буквами кроме первой прописной	
Организационно-правовая форма вуза – строчными буквами после первой прописной	
Наименование вуза прописными буквами, в скобках – сокращенное наименование	
Наименование обеспечивающей кафедры – строчными буквами кроме первой прописной, в скобках – сокращенное наименование	
ТЕМА РАБОТЫ - ПРОПИСНЫМИ БУКВАМИ	
Вид работы и название учебной дисциплины – строчными буквами кроме первой прописной	
Студент гр. (номер) <u>(подпись)</u> И. О. Фамилия <u>(дата)</u>	
Место для оценки <small>оценка</small>	Руководитель (должность, ученая степень, звание) <u>(подпись)</u> И. О. Фамилия <u>(дата)</u>
Место и год выполнения	