

«СОГЛАСОВАНО»
Решением П(Ц)МК
Протокол № 8
от «28» июня 2021г.

«УТВЕРЖДАЮ»
Заместитель директора по УР
_____ С.В. Клюквина
«28» июня 2021г.

***Методические рекомендации для студентов по
проведению лабораторных занятий по дисциплине
«МДК 11.01. Разработка, администрирование и защита баз
данных»
для специальности 09.02.07 «Информационные системы и
программирование»***

Разработал преподаватель СКМ и Э ФГБОУ ВО «СГТУ имени Гагарина Ю.А» Пояркова Т.А.

Данные методические рекомендации предназначены для студентов 4 курса, специальности 09.02.07 «Информационные системы и программирование»

СОДЕРЖАНИЕ

СОДЕРЖАНИЕ.....	3
Введение	4
Цель лабораторных занятий	4
Организация лабораторного занятия	5
Структура лабораторного занятия	8

Введение

Лабоаторные занятия, как виды учебных занятий, направлены на экспериментальное подтверждение теоретических положений и формирование учебных и профессиональных практических умений и составляют важную часть теоретической и профессиональной практической подготовки. Семинар является видом лабораторных занятий.

В процессе практического занятия обучающиеся выполняют одно или несколько практических заданий под руководством преподавателя в соответствии с изучаемым содержанием учебного материала.

Выполнение обучающимися практических занятий проводится с целью:

- формирования практических умений в соответствии с требованиями к уровню подготовки обучающихся, установленными рабочей программой дисциплины по конкретным разделам/ темам дисциплины;
- обобщения, систематизации, углубления, закрепления полученных теоретических знаний;
- совершенствования умений применять полученные знания на практике, реализации единства интеллектуальной и практической деятельности;
- развития интеллектуальных умений у будущих специалистов: аналитических, проектировочных, конструктивных и др.;
- выработки таких профессионально значимых качеств, как самостоятельность, ответственность, точность, творческая инициатива при решении поставленных задач при освоении общих компетенций.

Цель лабораторных занятий

В результате освоения учебной дисциплины «МДК 11.01. Разработка, администрирование и защита баз данных» обучающийся должен **уметь**:

- создавать объекты баз данных в современных системах управления базами данных и управлять доступом к этим объектам;
- работать с современными case-средствами проектирования баз данных;
- формировать и настраивать схему базы данных;
- разрабатывать прикладные программы с использованием языка SQL;
- создавать хранимые процедуры и триггеры на базах данных;
- применять стандартные методы для защиты объектов базы данных.

В результате освоения учебной дисциплины обучающийся должен **знать**:

- основные положения теории баз данных, хранилищ данных, баз знаний;
- основные принципы построения концептуальной, логической и физической модели данных;
- современные инструментальные средства разработки схемы базы данных;
- методы описания схем баз данных в современных системах управления базами данных (СУБД);
- структуры данных СУБД, общий подход к организации представлений, таблиц, индексов и кластеров;
- методы организации целостности данных;
- способы контроля доступа к данным и управления привилегиями;
- основные методы и средства защиты данных в базах данных;
- модели и структуры информационных систем;
- основные типы сетевых топологий, приемы работы в компьютерных сетях;

Методические рекомендации по проведению лабораторных занятий

- информационные ресурсы компьютерных сетей;
- технологии передачи и обмена данными в компьютерных сетях;
- основы разработки приложений баз данных.

Изучение дисциплины направлено на формирование следующих компетенций:

Код	Наименование результата обучения
ПК 11.1	Осуществлять сбор, обработку и анализ информации для проектирования баз данных
ПК 11.2	Проектировать базу данных на основе анализа предметной области
ПК 11.3	Разрабатывать объекты базы данных в соответствии с результатами предметной области
ПК 11.4	Реализовывать базу данных в конкретной системе управления базами данных
ПК 11.5	Администрировать базы данных
ПК 11.6	Защищать информацию в базе данных с использованием технологии защиты информации
ОК 1	Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.
ОК 2	Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности
ОК 3	Планировать и реализовывать собственное профессиональное и личностное развитие
ОК 4	Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами
ОК 5	Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста.
ОК 6	Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей
ОК 7	Содействовать сохранению окружающей среды, ресурсосбережению, эффективно действовать в чрезвычайных ситуациях
ОК 8	Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности.
ОК 9	Использовать информационные технологии в профессиональной деятельности
ОК 10	Пользоваться профессиональной документацией на государственном и иностранном языке.
ОК 11	Планировать предпринимательскую деятельность в профессиональной сфере

Организация лабораторного занятия

Лабораторное занятие должно проводиться в учебной лаборатории «Программирования и баз данных» оснащенная необходимым для реализации программы учебной дисциплины оборудованием.

В соответствии с требованиями ФГОС 09.02.07 «Информационные системы и программирование» реализация ППССЗ должна обеспечивать выполнение обучающимися их лабораторных работ, включая как обязательный компонент *Лабораторные занятия (с использованием персональных компьютеров)*.

Выполнению практических занятий предшествует проверка знаний обучающихся - их теоретической готовности к выполнению задания.

Лабораторные занятия могут носить репродуктивный, частично-поисковый и поисковый характер.

Работы, носящие *репродуктивный характер*, отличаются тем, что при их проведении обучающиеся пользуются подробными инструкциями, в которых указаны: цель работы, пояснения (теория, основные характеристики), оборудование, аппаратура, материалы и их характеристики, порядок выполнения работы, таблицы, выводы (без формулировки), контрольные вопросы, учебная и специальная литература.

Работы, носящие *частично-поисковый характер*, отличаются тем, что при их проведении обучающиеся не пользуются подробными инструкциями, им не дан порядок выполнения необходимых действий, и они требуют от обучающихся самостоятельного подбора оборудования, выбора способов выполнения работы в инструктивной и справочной литературе и др.

Работы, носящие *поисковый характер*, характеризуются тем, что обучающиеся, опираясь на имеющиеся у них теоретические знания, должны решить новую для них проблему.

При планировании лабораторных занятий необходимо находить оптимальное соотношение репродуктивных, частично-поисковых и поисковых работ, чтобы обеспечить высокий уровень интеллектуальной деятельности.

Формы организации обучающихся при проведении лабораторных занятий - фронтальная, групповая и индивидуальная.

При *фронтальной форме* организации занятий все обучающиеся выполняют одновременно одну и ту же работу.

При *групповой форме* организации занятий одна и та же работа выполняется бригадами по 2 - 5 человек.

При *индивидуальной форме* организации занятий каждый обучающийся выполняет индивидуальное задание.

Для повышения эффективности проведения практических занятий рекомендуется:

- разработка сборников задач, заданий и упражнений;
- разработка контрольно-диагностических материалов для контроля за подготовленностью обучающихся к практическим занятиям, в том числе в форме педагогических тестовых материалов для автоматизированного контроля;
- подчинение методики проведения практических занятий ведущим дидактическим целям с соответствующими установками обучающимся;

Методические рекомендации по проведению лабораторных занятий

- применение коллективных и групповых форм работы, максимальное использование индивидуальных форм с целью повышения ответственности каждого обучающегося за самостоятельное выполнение полного объема работ;
- проведение практических занятий на повышенном уровне трудности с включением в них заданий, связанных с выбором обучающимися условий выполнения работы, конкретизацией целей, самостоятельным отбором необходимого оборудования;
- подбор дополнительных задач и заданий для обучающихся, работающих в более быстром темпе, для эффективного использования времени, отводимого на Лабораторные занятия.

Структура лабораторного занятия

Лабораторная работа №1

Тема: Построение модели данных выбранной предметной области

Цели и задачи урока: Закрепить навыки построения сетевой модели

Тип урока – лабораторное занятие.

Методы обучения – частично поисковый

Обеспечение занятия – персональный компьютер.

Ход урока

1. Объяснение нового материала
2. Выполнение лабораторной работы
3. Закрепление пройденного материала (ответы на контрольные вопросы)
4. Подведение итогов

Литература

1. Фуфаев Д.Э., Фуфаева Э.В. Разработка и эксплуатация автоматизированных информационных систем. ОИЦ «Академия». 2019 (4-е изд.ст.)
2. Э.В. Фуфаев, Д.Э. Фуфаев, Базы данных, Москва, издательский центр «Академия», 2020
3. Э.В. Фуфаев, Д.Э. Фуфаев, Разработка удаленных баз данных, Москва, издательский центр «Академия», 2020

Теоретические сведения

Ядром любой БД является модель данных.

Модель данных – это совокупность структур данных и операций их обработки.

Т.к. СУБД имеет 3-х уровневую архитектуру, то понятие модели данных связано с каждым уровнем.

Физическая модель данных связана с организацией внешней памяти и структур хранения, используемых в данной операционной среде.

На **концептуальном** уровне модели данных наиболее важны для разработчиков БД, т.к. именно ими определяется тип СУБД.

Для **внешнего** уровня отдельных моделей данных нет, они лишь являются подсхемами концептуальных моделей данных.

Кроме моделей данных, соответствующих трем уровням архитектуры СУБД, существуют *предшествующие* им, не связанные с компьютерной реализацией. Они служат переходным звеном от реального мира к БД. Это класс инфологических (семантических) моделей.

Инфологические (семантические) модели данных используются на ранних стадиях проектирования БД.

Даталогические модели данных уже поддерживаются конкретной СУБД.

Физические модели данных связаны с организацией данных на носителях.

Документальные модели данных соответствуют слабоструктурированной информации, ориентированной на свободные форматы документов на естественном языке.

Модели данных, **ориентированные на формат документа**, связаны со стандартным общим языком разметки SGML (Standart Generaliset Markup Language), а также HTML, предназначенным для управления процессом вывода содержимого документа на экран.

Дескрипторные модели данных – самые простые, широко использовались раньше. В них каждому документу соответствует дескриптор – описатель, который имеет жёсткую структуру и описывает документ в соответствии с заранее определенными характеристиками.

Методические рекомендации по проведению лабораторных занятий

Тезаурусные модели данных основаны на принципе организации словарей. Содержат языковые конструкции и принципы их взаимодействия в заданной грамматике. Эти модели используются, например, в системах-переводчиках.

Объектно-ориентированная модель перекликается с семантическими моделями данных. Принципы похожи на принципы объектно-ориентированных языков программирования. Структура таких моделей графически представима в виде дерева, узлами которого являются объекты. Свойства объектов описываются типом.

Детали приборов

Код	Расположение поверхностей	Дополнительная характеристика
1	Тела вращения	Валы
2	Тела вращения	Втулки
3	Не тела вращения	Плоские
4	Не тела вращения	Объемные

Объекты **иерархической** модели данных связаны иерархическими отношениями и образуют ориентированный граф. Основные понятия иерархических структур: уровень, узел (совокупность свойств данных, описывающих объект), связь.

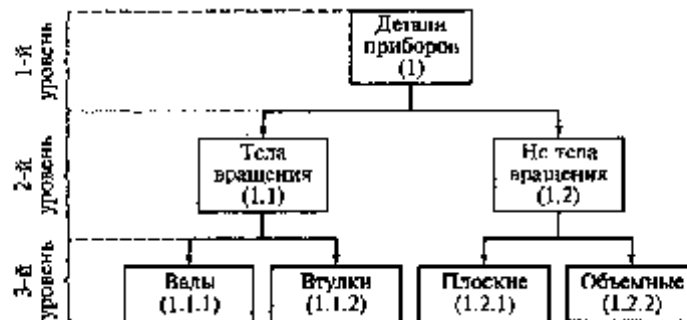


Рисунок 1

В **сетевой** модели данных при тех же основных понятиях (уровень, узел, связь) каждый элемент может быть связан с любым другим элементом.

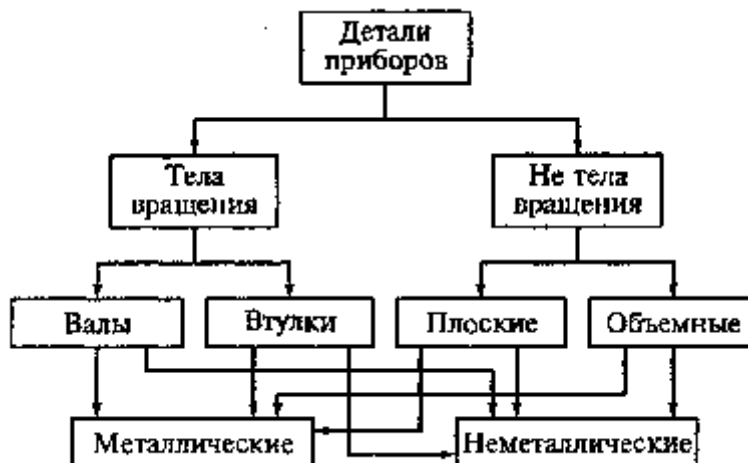


Рисунок 2

В **реляционной** модели данных данные представлены только в виде таблиц.



Рисунок 3.

Задания:

Задание 1. Ознакомьтесь с предложенным материалом

Задание 2. Разработать сетевое, реляционное и иерархическое представление для полей представленными данными о фирмах-поставщиках наименование, телефон адрес, город. Для реляционного представления указать данные.

Задание 3. Ответить на вопросы

Задание 4. Составить отчет по проделанной работе

Контрольные вопросы:

1. Что такое база данных?
2. Что такое система баз данных?
3. Что такое система управления базами данных?
4. Основное назначение?
5. Основные компоненты СУБД?
6. Что подразумевает понятие абстрагирование в СУБД?
7. Понятие модели данных
8. Цель этапа концептуального проектирования

Лабораторная работа №2

Тема: Проектирование ER диаграммы концептуальной модели для выбранной предметной области

Цели и задачи урока: Закрепить навыки построения ER диаграмм

Тип урока – лабораторное занятие.

Методы обучения – частично поисковый

Обеспечение занятия – персональный компьютер.

Ход урока

1. Объяснение нового материала
2. Выполнение лабораторной работы
3. Закрепление пройденного материала (ответы на контрольные вопросы)
4. Подведение итогов

Литература

Методические рекомендации по проведению лабораторных занятий

1. Фуфаев Д.Э., Фуфаева Э.В. Разработка и эксплуатация автоматизированных информационных систем. ОИЦ «Академия». 2019 (4-ее изд.ст.)
2. Э.В. Фуфаев, Д.Э. Фуфаев, Базы данных, Москва, издательский центр «Академия», 2020
3. Э.В. Фуфаев, Д.Э. Фуфаев, Разработка удаленных баз данных, Москва, издательский центр «Академия», 2020

Теоретические сведения

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.

Выделим 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.

Методические рекомендации по проведению лабораторных занятий

Контракты будем рассматривать 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.

ER–диаграмма издательской компании приведена на рис. 3 (базовые сущности на рисунках выделены полужирным шрифтом).

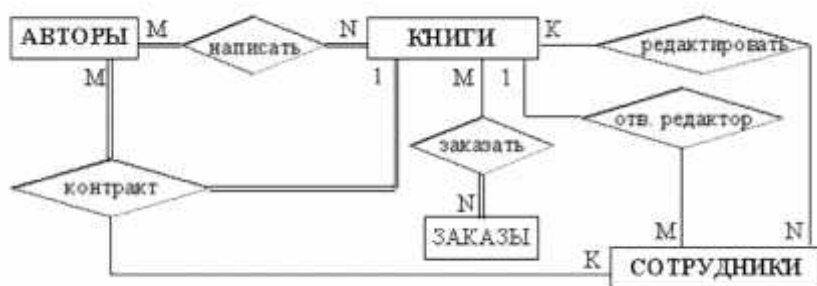


Рисунок 4. ER–диаграмма издательской компании

Задания:

Задание 1. Ознакомьтесь с предложенным материалом

Задание 2. Выбрать предметную область (например: склад, больница, аптека, аэропорт и т.д.) и составить для нее:

- описание предметной области (от имени конечного пользователя);
- ER-диаграмму.

Ограничение: от 5ти до 10ти сущностей для описания области. Каждый объект должен иметь хотя бы один атрибут.

Задание 3. Ответить на вопросы

Задание 4. Составить отчет по проделанной работе

Контрольные вопросы:

- Концепция ER модели
- Структурные ограничения ER модели
- Обозначения концепций и структурных ограничений на ER-диаграмме
- Основные этапы формирования концептуальной модели
- Сравнение ER-модели и объектно-ориентированной модели.

Лабораторная работа №3

Тема: Построение логической модели при помощи утилит автоматизированного проектирования.

Цели и задачи урока: Закрепить навыки построения логической модели данных

Тип урока – лабораторное занятие.

Методы обучения – частично поисковый

Обеспечение занятия – персональный компьютер.

Ход урока

1. Объяснение нового материала
2. Выполнение лабораторной работы
3. Закрепление пройденного материала (ответы на контрольные вопросы)
4. Подведение итогов

Литература

1. Фуфаев Д.Э., Фуфаева Э.В. Разработка и эксплуатация автоматизированных информационных систем. ОИЦ «Академия». 2019 (4-ее изд.ст.)
2. Э.В. Фуфаев, Д.Э. Фуфаев, Базы данных, Москва, издательский центр «Академия», 2020
3. Э.В. Фуфаев, Д.Э. Фуфаев, Разработка удаленных баз данных, Москва, издательский центр «Академия», 2020

Теоретические сведения

Инфологическая модель позволяет представить предметную область в формализованном виде. При этом используются наиболее естественные для человека формы представления той информации, которую предполагается хранить в создаваемой базе данных. Поэтому для построения инфологической модели данных используют различные виды семантических моделей: семантические сети, модель «сущность-отношение» и др. В лабораторной работе используется модель «сущность-отношение» (ER-модель). Основными элементами этой модели являются сущности, их свойства (атрибуты) и связи между сущностями.

Сущность определяет некоторый объект предметной области, информацию о котором необходимо хранить в БД. При этом объекты могут быть как реальными (например, студент), так и абстрактными, (например, зачет). Объекты предметной области образуют классы объектов, имеющих одинаковые свойства. Такие классы определяют *тип сущности* (в дальнейшем, просто сущность). Понятие тип сущности относится к набору однородных личностей, предметов (напр. сущность Студент, характеризующаяся фамилией, номером группы, номером зачетной книжки). Конкретный объект из этого класса задает *экземпляр сущности*. Экземпляр сущности определяет конкретный объект в наборе. Напр., конкретного студента, для которого указанные характеристики имеют конкретные значения: Семенов Гр1 1232009. Характеристики сущности называются атрибутами. Так, фамилия, номер группы, номер зачетной книжки являются атрибутами сущности Студент. Имена атрибутов должны быть уникальным для конкретного типа сущности. Для разных типов сущностей имена атрибутов могут повторяться (напр., атрибут Фамилия может быть определен для многих сущностей: преподаватель, студент, автор книги и т.п.). Атрибуты определяют структуру информации, которая должна быть собрана о сущности.

Среди всех атрибутов сущности особое положение занимают атрибуты, образующие ключ. *Ключ* – это минимальный набор атрибутов, по значениям которых можно однозначно найти требуемый экземпляр сущности. Различают сильные и слабые типы сущностей. Сильные сущности существуют сами по себе, а существование слабых сущностей зависит от существования сильных.

Методические рекомендации по проведению лабораторных занятий

Другим элементом ER-модели является связь. Связь определяет зависимость двух или более сущностей. С помощью связей отражаются семантические соотношения между сущностями. Наличие множества связей и определяет сложность инфологических моделей. При построении ER-модели используются бинарные связи, т.е. связи между двумя сущностями. Различают следующие типы связей:

Связь ОДИН-К-ОДНОМУ (1:1). Такая связь определяет то, что в каждый момент времени каждому экземпляру сущности А соответствует 1 или 0 экземпляров сущности В (рис.5): студент занимается в одной аудитории



Рис. 5. Связь 1:1

Связь ОДИН-КО-МНОГИМ (1:M). Она означает, что одному экземпляру сущности А соответствуют несколько экземпляров сущности В (рис.6.): Преподаватель обучает многих студентов

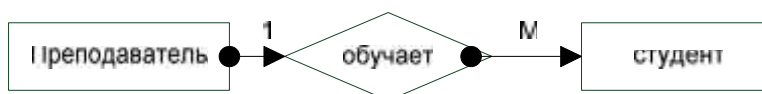


Рис. 6. Связь 1:M

Связь МНОГИЕ-КО-МНОГИМ (M:N). Она означает, что многим экземплярам сущности А соответствует много экземпляров сущности (рис. 1.3): Преподаватели обучают студентов.

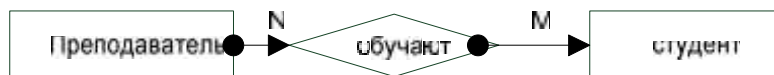


Рис. 7. Связь N:M

Все связи требуют описания, которое включает:

- идентификатор связи;
- формулировку имен связи с точки зрения связываемых сущностей;
- тип связи.

Особое внимание при построении модели БД необходимо уделять обеспечению целостности данных. *Целостность* – свойство БД, которое понимается, как способность БД поддерживать правильность данных в любой момент времени. Поддержание целостности может рассматриваться как защита данных от неверных изменений или разрушений. Значительная часть правил, определяющих целостность БД, зависит от смысловых правил работы с данными в конкретной предметной области. Поэтому при проектировании ER-модели необходимо подробно определить все смысловые ограничения и правила, связанные с обработкой данных.

Задания:

Задание 1. Ознакомьтесь с предложенным материалом

Задание 2. Спроектируйте связи к базе данных **Кадры**

Задание 3. Установите связь один ко многим между таблицами **Подразделения** и **Сотрудники**. В окне **Изменение связей** установите флажок **Обеспечение целостности данных**. После этого станут доступны флажки **Каскадное обновление связанных полей** и **Каскадное удаление связанных полей**.

Задание 4. Заполните таблицы

Задание 5. Ответьте на вопросы

Методические рекомендации по проведению лабораторных занятий

Задание 6. Составьте отчет о проделанной работе

Вопросы:

1. Что такое внешний ключ?
2. Как формализуется связь 1:1?
3. Как формализуется связь 1:M?
4. Как формализуется связь M:N?

Лабораторная работа №4

Тема: Установка и нормализация отношений в базе данных. Определение функциональной зависимости

Цели и задачи урока: Закрепить навыки выполнения нормализации данных

Тип урока – лабораторное занятие.

Методы обучения – частично поисковый

Обеспечение занятия – персональный компьютер.

Ход урока

1. Объяснение нового материала
2. Выполнение лабораторной работы
3. Закрепление пройденного материала (ответы на контрольные вопросы)
4. Подведение итогов

Литература

1. Фуфаев Д.Э., Фуфаева Э.В. Разработка и эксплуатация автоматизированных информационных систем. ОИЦ «Академия». 2019 (4-ее изд.ст.)
2. Э.В. Фуфаев, Д.Э. Фуфаев, Базы данных, Москва, издательский центр «Академия», 2020
3. Э.В. Фуфаев, Д.Э. Фуфаев, Разработка удаленных баз данных, Москва, издательский центр «Академия», 2020

Теоретические сведения

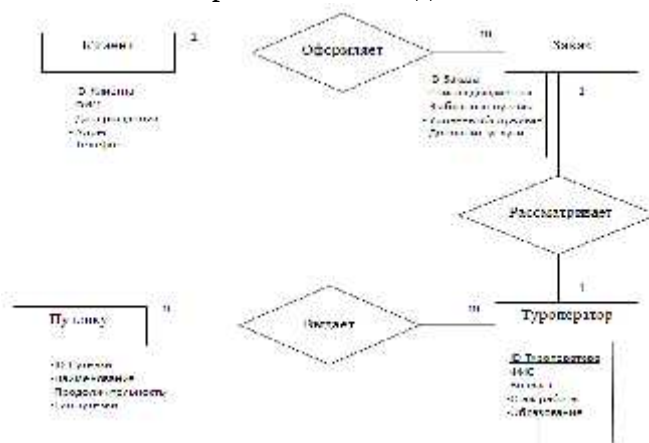


Рисунок 1 – ER диаграмма туристической фирмы

Первая нормальная форма ER-диаграммы

В первой нормальной форме ER-диаграммы устраняются атрибуты, содержащие множественные значения, т. е. производится выявление неявных сущностей, «замаскированных» под атрибуты. Мы должны просмотреть схему отношения и разделить составные атрибуты на различные строки/столбцы. Возможно, эту операцию придется повторить несколько раз до тех пор, пока каждый из атрибутов не станет атомарным (с учетом сказанного в предыдущем абзаце).

Методические рекомендации по проведению лабораторных занятий

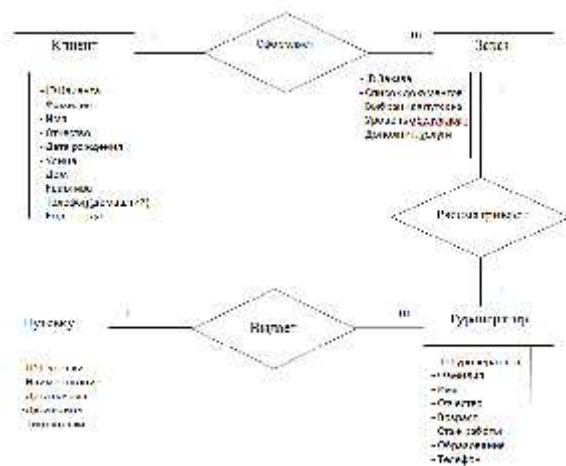


Рисунок 2 – ER диаграмма туристической фирмы, первой нормальной формы

Вторая нормальная форма ER-диаграммы

Во второй нормальной форме устраняются атрибуты, зависящие только от части уникального идентификатора. Эта часть уникального идентификатора определяет отдельную сущность.

Другими словами, отношение находится во 2НФ, если оно находится в 1НФ, и при этом все неключевые атрибуты зависят только от ключа целиком, а не от какой-то его части.

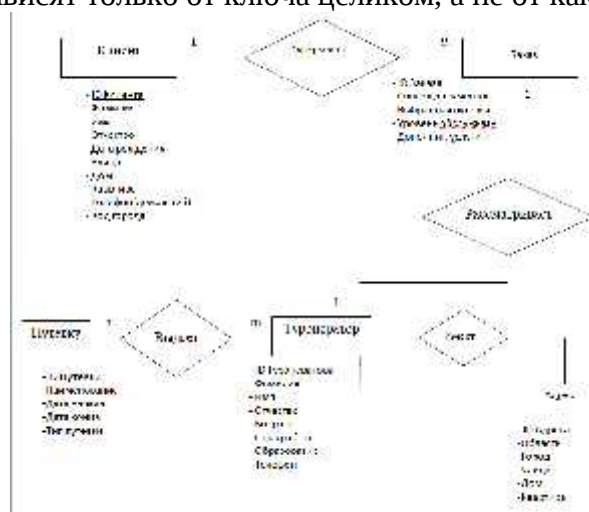


Рисунок 3 – ER диаграмма туристической фирмы, второй нормальной формы

Третья нормальная форма ER-диаграммы

В третьей нормальной форме устраняются атрибуты, зависящие от атрибутов, не входящих в уникальный идентификатор. Эти атрибуты являются основой отдельной сущности. Например рассмотреть заказ может так же и юрист, если возникнет какой либо юридический вопрос; рассмотреть заказ может и сам директор, если возникнет какой ни будь частный случай или вопрос стоит с финансовой точки зрения; нужно разбить таблицы на 2 из которых первая будет относиться к обычному заказу, где его рассмотрит туроператор и частному где его будет рассматривать другой работник фирмы в зависимости от частности случая.

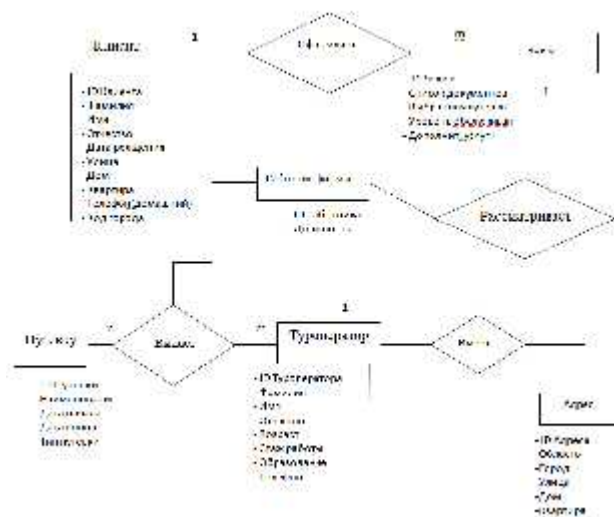


Рисунок 4 – ER диаграмма туристической фирмы, третьей нормальной формы
Задания:

Задание 1. Ознакомьтесь с предложенным материалом

Задание 2. Спроектируйте связи к базе данной

Задание 3. Установите связь один ко многим между таблицами **Подразделения** и **Сотрудники**. В окне **Изменение связей** установите флажок **Обеспечение целостности данных**. После этого станут доступны флажки **Каскадное обновление связанных полей** и **Каскадное удаление связанных полей**.

Задание 4. Составить отчет

Лабораторная работа №5

Тема: Обеспечение доступа к данным при помощи технологии ADO

Цели и задачи урока: Изучить возможности работы с технологией ADO

Тип урока – лабораторное занятие.

Методы обучения – репродуктивный

Обеспечение занятия – персональный компьютер.

Ход урока

1. Объяснение нового материала
2. Выполнение лабораторной работы
3. Закрепление пройденного материала (ответы на контрольные вопросы)
4. Подведение итогов

Литература

1. Фуфаев Д.Э., Фуфаева Э.В. Разработка и эксплуатация автоматизированных информационных систем. ОИЦ «Академия». 2019 (4-ее изд.ст.)
2. Э.В. Фуфаев, Д.Э. Фуфаев, Базы данных, Москва, издательский центр «Академия», 2020
3. Э.В. Фуфаев, Д.Э. Фуфаев, Разработка удаленных баз данных, Москва, издательский центр «Академия», 2020

Теоретические сведения

Платформа .NET определяет ряд пространств имен, позволяющих непосредственно взаимодействовать с локальными и удаленными базами данных (БД). Вместе они известны как ADO.NET.

Методические рекомендации по проведению лабораторных занятий

Мы рассмотрим два уровня работы с ADO.NET:

подключенный уровень (connected layer) — позволяет взаимодействовать с базой данных с помощью объектов подключения, чтения данных и команд конкретного поставщика данных; **отключенный уровень** (disconnected layer) — позволяет смоделировать в оперативной памяти данные из базы данных с помощью многочисленных классов из пространства имен System.Data (DataSet, DataTable и т.д.)

Существует и **третья возможность** взаимодействия с базами данных через объектную модель C#, с помощью ORM-фреймворка Entity Framework 4.0, которую мы пока рассматривать не будем.

Пространство имен System.Data

ADO.NET поставляется с тремя пространствами имен клиента базы данных: одно для **SQL Server**; второе — для любой базы данных, доступной через **OLE DB**; третье — для источников данных **Open Database Connectivity (ODBC)**. Если выбрана база данных, отличная от SQL Server, то отдадут предпочтение OLE DB, если только не окажется, что нет другого выбора кроме ODBC. Если же в качестве базы данных используется Oracle, то на сайте Oracle .NET Developer можно получить их поставщика .NET — ODP.NET.

С точки зрения программиста, тело ADO.NET составляет базовая сборка с именем **System.Data.dll**. В этом двоичном файле находятся пространства имен, которые представляют типы конкретного поставщика данных ADO.NET, это: Odbc { }, OleDb { }, SqlClient { }, и кроме того: Common { }, Sql { }, SqlTypes { } и другие. Большинство шаблонов проектов Visual Studio автоматически ссылаются на эту ключевую библиотеку System.Data { } доступа к данным. Однако для импорта нужных пространств имен необходимо изменить кодовые файлы, например:

```
using System;
using System.Data;
using System.Data.SqlClient;
...
```

Учтите также, что кроме System.Data.dll, существуют и другие ориентированные на ADO.NET сборки (например, System.Data.OracleClient.dll или System.Data.Entity.dll), которые необходимо вручную указывать в текущем проекте с помощью диалогового окна Add Reference (Добавление ссылки).

Компоненты ADO.NET

Двумя основными компонентами ADO.NET для доступа к данным и их обработки являются поставщики данных .NET Framework и DataSet.

Поставщики данных .NET Framework

— это компоненты, которые специально сконструированы для обработки данных и быстрого, однопроходного доступа к данным только для чтения:

1. Объект Connection обеспечивает подключение к источнику данных.
2. Объект Command позволяет обращаться к командам базы данных для возврата данных, изменения данных, выполнения хранимых процедур и передачи или получения сведений о параметрах.
3. DataReader обеспечивает высокопроизводительный поток данных из источника данных.
4. DataAdapter предоставляет «мост» между объектом DataSet и источником данных. DataAdapter использует объекты Command для выполнения команд SQL на источнике данных для загрузки DataSet с данными и согласования изменений данных, выполненных в DataSet, вновь с источником данных.

5. Parameter представляет именованный параметр в параметризованном запросе.

6. Transaction инкапсулирует транзакцию в базе данных.

Конкретные имена этих основных классов различаются у различных поставщиков (например, SqlConnection, OleDbConnection, OdbcConnection и MySqlConnection), но все эти объекты порождены от одного и того же базового класса (в случае объектов подключения это DbConnection), который реализует идентичные интерфейсы (вроде IDbConnection). Поэтому если вы научитесь работать с одним поставщиком данных, то легко справитесь и с остальными.

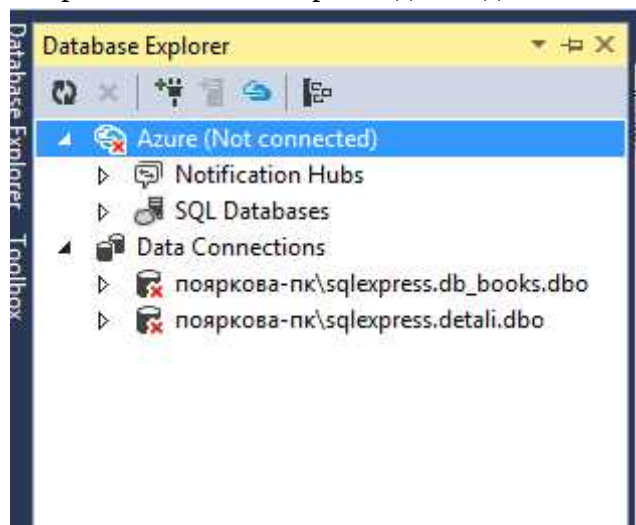
Класс DataSet в ADO.NET

— специально сконструирован для доступа к данным независимо от источника данных. Поэтому он может быть использован со многими и разными источниками данных, с XML-данными или для управления данными, локальными для приложения. DataSet содержит коллекцию одного или нескольких объектов **DataTable**, состоящих из строк и столбцов данных, а также первичный ключ, внешний ключ, ограничение и связанные сведения о данных в объектах DataTable.

Пример «Продажа билетов в кинотеатре» демонстрирует способ работы с источником данных в автономном режиме. При получении DataSet с помощью объекта — адаптера данных кино1TableAdapter подключение **открывается и закрывается автоматически** (метод кино1TableAdapter.Fill(this.kinoDataSet.кино1);). Визуальный компонент dataGridView1 отображает содержимое киноDataSet через связующий компонент кино1BindingSource. Получив объект DataSet, программа может просматривать и обрабатывать данные без затрат на сетевой трафик. А если нужно занести изменения в хранилище данных, то адаптер данных (метод кино1TableAdapter.Update(this.kinoDataSet.кино1);) выполняет действия по обновлению данных — при этом подключение **открывается заново для проведения обновлений в базе, после чего сразу же закрывается.** Легко представить себе регламент работы нескольких кассиров с одной базой данных — после ввода информации о проданных билетах достаточно лишь нажать кнопку «Обновить». **Промоделируйте работу нескольких кассиров с одной базой данных в своей локальной сети.**

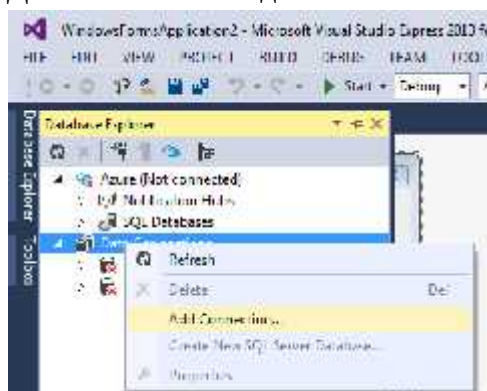
Задание

1. Запустите VisualStudio
2. Создайте новый проект
3. Открыть окно Data Explorer для подключения базы данных



Методические рекомендации по проведению лабораторных занятий

4. Добавьте новое подключение

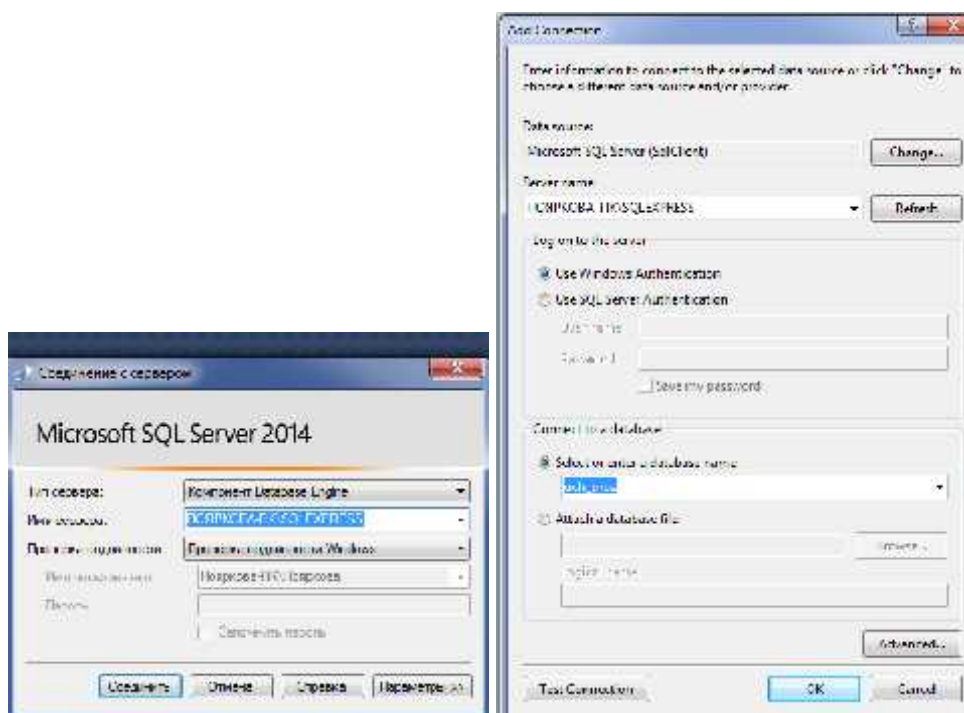


5. Подключить сервер

6. На вкладке соединение укажите тип сервера которое совпадает с типом сервера MS SQL

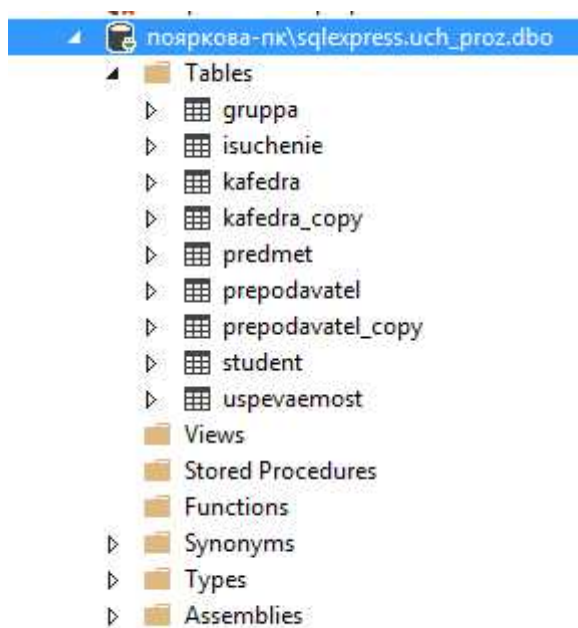
7. Установите переключатель использовать встроенную учетную запись

8. Выберите вашу базу данных из списка, проверьте соединение и подключитесь



9. Проверить соединение. Подключить сервер

10. Проверьте что все необходимые объекты появились в списке объектов баз данных

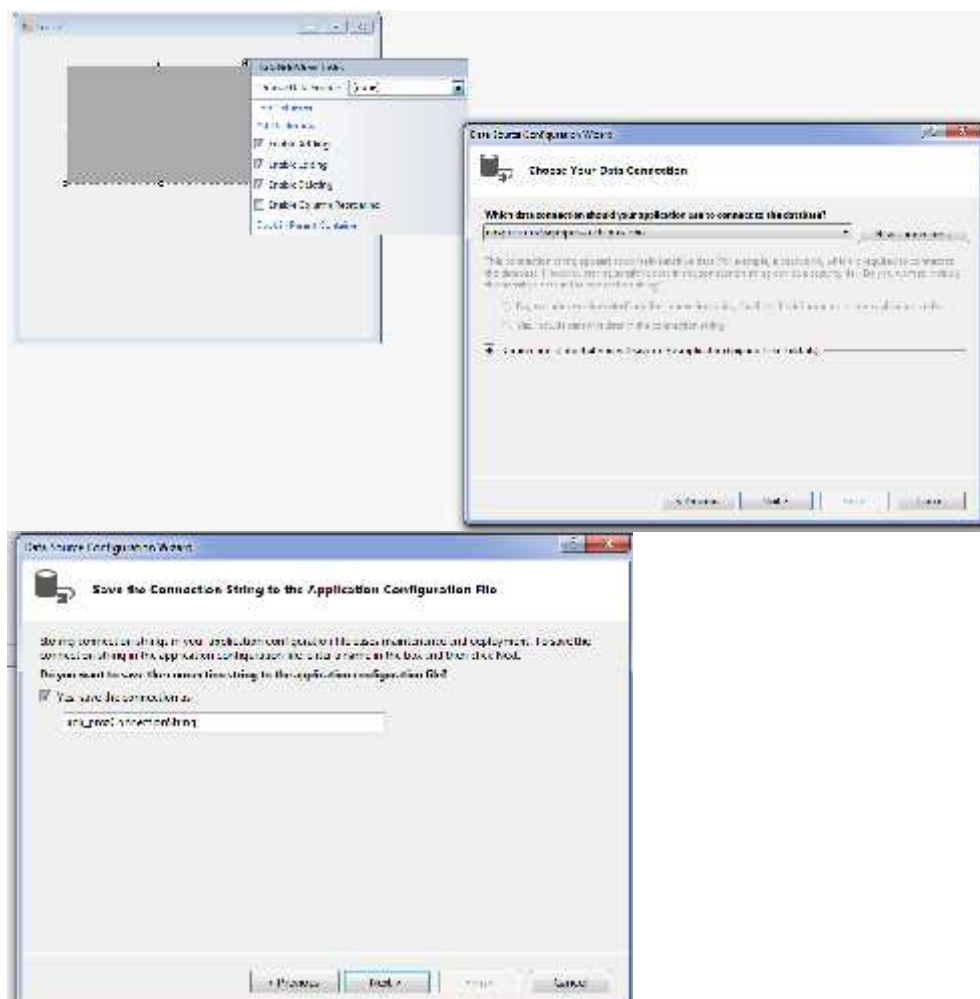


11. Установите значение True у объекта ADOConnection в свойстве Connected

12. Создайте меню на главной форме



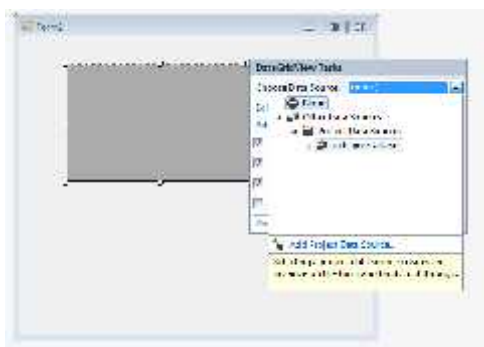
13. Создайте формы для заполнения сведений Например, сведения о группе (Форма «Учебная группа» запускается при выборе Студенты → Группы)
14. Выводим сведения из таблицы группа на форму. Добавим на форму бъект Data Source и настроим его. Выбираем соответствующий Connect.



15. Выбираем соответствующие объекты

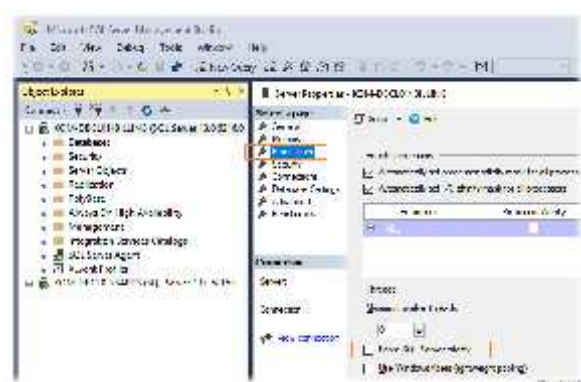


16. Подключаем DataGridView.
17. Выбираем соответствующий Data Source и Data Set



18. Добавить и настроить навигатор. Прикрепим его к тем же Data Source и Data Set что и основную таблицу.

19. Просмотреть результат. Аналогично добавьте форму Кафедра.



Задание слишком высокого приоритета может лишить ресурсов операционную систему и сетевые функции, что может вызвать проблемы при завершении работы SQL Server и выполнении на сервере других административных задач.

безопасность

Permissions

Разрешения на выполнение хранимой процедуры **sp_configure** без параметров или только с первым параметром по умолчанию предоставляются всем пользователям. Для выполнения процедуры **sp_configure** с обоими параметрами для изменения параметра конфигурации или запуска инструкции RECONFIGURE необходимо иметь разрешение ALTER SETTINGS на уровне сервера. Разрешение ALTER SETTINGS неявным образом предоставлено предопределенным ролям сервера **sysadmin** и **serveradmin**.

Настройка параметра повышения приоритета

1. В обозревателе объектов щелкните правой кнопкой мыши сервер и выберите пункт **Свойства**.
2. Щелкните узел **Процессоры**.
3. В разделе **Потоки** установите флажок **Повысить приоритет SQL Server**.
4. Остановите и снова запустите SQL Server.

Настройка параметра повышения приоритета

1. Установите соединение с компонентом Компонент Database Engine.
2. На панели «Стандартная» нажмите **Создать запрос**.
3. Скопируйте следующий пример в окно запроса и нажмите кнопку **Выполнить**. В этом примере описывается использование процедуры [sp_configure](#) для задания значения параметра priority boost равным 1.

SQLКопировать

```
USE AdventureWorks2012 ;
```

```
GO
```

```
EXEC sp_configure 'show advanced options', 1;
```

```
GO
```

```
RECONFIGURE ;
```

```
GO
```

```
EXEC sp_configure 'priority boost', 1 ;
```

```
GO
```

```
RECONFIGURE;
```

```
GO
```

Чтобы изменения вступили в силу, необходимо перезапустить сервер.

Задание

1. Установите приоритет в режиме менеджера
2. Измените приоритет в командном режиме