МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«Национальный исследовательский ядерный университет «МИФИ»
Обнинский институт атомной энергетики –
филиал федерального государственного автономного образовательного учреждения высшего образования «Национальный исследовательский ядерный университет «МИФИ»
(ИАТЭ НИЯУ МИФИ)

ОТДЕЛЕНИЕ ИНТЕЛЛЕКТУАЛЬНЫХ КИБЕРНЕТИЧЕСКИХ СИСТЕМ

Одобрено на заседании Ученого
совета ИАТЭ НИЯУ МИФИ,
протокол №25.8 от 25.08.2025

МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ
для выполнения лабораторных работ по дисциплине

Базы данных

название дисциплины

для студентов направления подготовки

38.03.05 Бизнес-информатика

образовательная программа:

Цифровая трансформация компании

Форма обучения: очная

Обнинск 2025

СОДЕРЖАНИЕ

Введение …….	4
ЛАБОРАТОРНАЯ РАБОТА № 1. Проектирование базы данных для выбранной предметной области …….	6
Приложение 1. Темы предметных областей для лабораторных работ …….	7
Приложение 2. Пример оформления отчета ……	11
ЛАБОРАТОРНАЯ РАБОТА № 2.Создание базы данных для выбранной предметной области на основе СУБД Access …….	18
Приложение 3. Пример создания БД на основе предметной области «Конференции»…….	22
ЛАБОРАТОРНАЯ РАБОТА № 3. Доступ к базе данных через web……..	27
Приложение 4. Создание БД «Конференции» с использованием СУБД MySQL и написание приложения к ней на скриптовом языке PHP …….	29
Приложение 5. Создание БД «Конференции» с использованием СУБД Oracle и написание приложения к ней на скриптовом языке JSP ……	35
Приложение 6. Экспорт данных из СУБД ACCESS в СУБД ORACLE …….	44
ЛАБОРАТОРНАЯ РАБОТА № 4. Доступ к БД через WEB (облачная реализация) ……..	46
Приложение 7. Создание БД «Конференции» с использованием СУБД SQL AZURE и написание приложения к ней на языке C# …….	48
ЛАБОРАТОРНАЯ РАБОТА № 5. Знакомство с инструментальными средствами для создания новых информационных технологий на основе баз данных ….	60
Приложение 8. Проектирование БД в ERWin (на примере проектирования БД «Конференции») …	61
ЛАБОРАТОРНАЯ РАБОТА № 6. Подготовка документации на базу данных …….	75

	Приложение 9. Лист изменений	99
	Приложение 10. Структура описания [наименование базы данных]	99
	Приложение 11. Примеры заполнения титульных листов для различных вариантов описаний технологий	103

ВВЕДЕНИЕ

По предмету «Базы данных», изучаемому на кафедре «Компьютерные системы, сети, технологии» (КССТ) факультета кибернетики ИАТЭ, имеются теоретические материалы в виде учебного пособия «Базы данных» (Обнинск: ИАТЭ, 2004. – 107 с.), полные электронные версии лекций в виде документов в редакторе Word, требования к выполнению лабораторных работ и списки литературы для каждой лабораторной работы. Однако у многих студентов возникают проблемы с поиском материалов для лабораторных работ и вопросы по их выполнению, поэтому необходимо было создать лабораторный практикум по выполнению лабораторных работ для всех вариантов их реализации.

Данное методическое пособие было подготовлено с целью

- предоставления студентам необходимой информации по выполнению лабораторных работ;
- поиска источника информации с программным обеспечением для скачивания, установки и настройки всего необходимого для выполнения лабораторных работ;
- стимулирования у студентов интереса к изучению курса «Базы данных»;
- предоставления необходимой информации по условиям сдачи лабораторных работ, т.е. сроки сдачи, примерные объемы работ и отчетов;
- предоставления вариантов отчетов;
- разработки примерных алгоритмов выполнения лабораторных работ.

В практикуме отражены основные моменты выполнения лабораторных работ. При описании каждой лабораторной работы обращается внимание на различные проблемы, возникшие при изучении тех или иных инструментов и методик. Все примеры реализации представлены на

одной предметной области «Сведения о научных конференциях». Лабораторный практикум построен по следующей схеме: каждая лабораторная работа рассматривается в 4-х разделах (исходные данные, инструменты, методы, форма отчета).

Лабораторный практикум создан в помощь студентам, содержит основные сведения, необходимые для выполнения лабораторных работ по курсу, однако, естественно, в таком объеме нельзя поместить всю необходимую информацию, поэтому студентам рекомендуется самостоятельно изучить предложенные источники из списка литературы и другие источники этой тематики. В предлагаемом пособии проведено поэтапное описание используемых инструментов, методов и алгоритмов, владение которыми необходимо для выполнения лабораторных работ.

Лабораторный практикум с необходимыми инструментами (СУБД) и примерами реализации имеется в электронном виде.

ЛАБОРАТОРНАЯ РАБОТА № 1

ПРОЕКТИРОВАНИЕ БД ДЛЯ ВЫБРАННОЙ ПРЕДМЕТНОЙ ОБЛАСТИ

Цель работы: обучение студентов основным принципам проектирования баз данных (БД), а также методологии, технологии, средствам формирования БД в среде системы управления базами данными (СУБД) Access.

В результате выполнения лабораторной работы студент должен знать методологию и технологию проектирования БД, а также ее формирования в среде реляционной СУБД, уметь спроектировать конкретную БД.

Этапы выполнения работы

Основа грамотного построения базы данных – это правильная организация проектирования и выполнение всех его этапов.

1. Выбор предметной области (учебное заведение, склад, метаданные, Web - ресурсы и др.). Для выполнения этого этапа необходимо

- проанализировать информационные потребности пользователей;
- сформировать состав документов, подлежащих включению в БД;
- разработать состав и форму представления информации по каждому документу;
- создать кодификаторы для упорядочения данных в БД.

2. Описание трудностей в существующей схеме обработки данных.

3. Выявление целей создания базы данных.

4. Выявление факторов, способствующих созданию БД.

5. Выявление факторов, затрудняющих создание БД.

6. Составление плана организационно-технических мероприятий по созданию системы.

7. Определение задач и функций системы.

8. Описание логического представления базы данных.

9. Обоснование выбора СУБД для реализации БД.

10. Составление технического задания на создание базы данных.

Форма отчетности. Выслать преподавателю отчет о выполненной лабораторной работе по электронной почте. Сдать зачет по проделанной лабораторной работе.

Объем. Текст с описанием должен содержать 4−5 страниц, набранных на компьютере (шрифт Times New Roman,12'), с таблицами.

Сроки представления: в течение первого месяца занятий.

Литература

1. Модин А.А., Яковенко Е.Г., Погребной Е.П. Справочник разработчика АСУ. / Под ред. Н.П.Федоренко, В.В.Карибского (2-е издание, доп., перераб.). – М.: Экономика. 1978. – 583 с.

2. Бойко В.В., Савинков В.М. Проектирование информационной базы на основе СУБД. – М.: Финансы и статистика. 1982. – 174 с.

Приложение 1

ТЕМЫ ПРЕДМЕТНЫХ ОБЛАСТЕЙ ДЛЯ СОЗДАНИЯ БД

1. Web - ресурсы (описание сайта – темы – ссылки – пользователи – подборки ссылок).
2. Автопарк - учет автомобилей: за кем закреплен, километраж, путевки (гараж – подразделение - водители - путевки).
3. Анкетирование (деканат – кафедра – группа тестируемых – состав групп – люди – бланки анкет – вопросы – результаты анкетирования).

4. Аттестационные ведомости (деканат – кафедра – группа – студенты – ведомости – дисциплины – оценки).
5. Буфет – учет продуктов: когда, кем, что и сколько поставлено, кто получал, сколько продано, сколько осталось (буфеты – продукты – типы продуктов – приход/расход – перечень продуктов – работники).
6. Гарантийные талоны (оборудование – типы оборудования – фирма поставщик – гарантийные талоны).
7. Деканат: учет оплаты учебных задолженностей, контроль успеваемости, выдача справок (деканат – группы – студенты – оценки – оплата за обучение – справки).
8. Журнал загрузки аудитории (кафедра – группа – преподаватель – дисциплина – программный продукт – записи журнала).
9. Журнал посещаемости лекций и лабораторных работ (кафедра – группа – студент – дисциплина – преподаватель – занятие – посещаемость занятий).
10. Журнал учета компьютерного времени: с предоставлением информации загрузки компьютерных аудиторий по кварталам, по месяцам по преподавателям и по программным продуктам.
11. Записная книжка-планировщик (пользователь – планировщик – мероприятия).
12. Зачетная книжка студента (деканат – кафедра – группа – студенты – зачетные книжки).
13. Картотека библиотеки (библиотека – карточка книги – издательство – город – тема книги – дисциплина – список выборки книги по дисциплине – позиции выборки).
14. Картотека больных: кто, когда поступил в больницу, с каким диагнозом, как проходил курс лечения.
15. Каталог дисциплин: какие дисциплины и на каком курсе читаются (деканат – кафедра – поток – дисциплины – дисциплины потока).

16. Кафедра (кафедра – группа – студенты – преподаватели – дисциплины – ведомости успеваемости).
17. Книги на кафедре: какие книги, по какой теме и где находятся (кафедра – книги – темы книг – авторы – аудитории – люди – кто и когда брал и возвращал книги).
18. Контрольные вопросы по дисциплинам, темам и разделам (дисциплина – преподаватели – набор билетов – билеты – вопросы к билетам – вопросы – темы вопросов).
19. Музыкальная картотека (диск – группа – тип произведения).
20. Планирование рабочего дня (список пользователей – планировщик – мероприятия).
21. Радиодетали (изготовитель – продавец – покупатель).
22. Расписание занятий в университете (деканат – кафедра – группа – дисциплины – аудитории – преподаватели – расписание занятий).
23. Расписание маршрутов движения транспорта (станции – маршруты – станции маршрутов – время прибытия/убытия).
24. Распоряжения и контроль их выполнения.
25. Рецепты (раздел – подраздел – тип рецепта – рецепт).
26. Склад: учет прихода и расхода товара со склада (люди – склад – товар – тип товара– приход/расход).
27. Студенческое общежитие (корпус – комнаты – проживающие – оплата за проживание).
28. Табельный учет загрузки преподавателей по месяцам (деканат – кафедра – преподаватели – табель – состав табеля – загрузка преподавателей).
29. Телефонный справочник университета. Информация о телефонных аппаратах: тип, принадлежность кафедре, аудитория, расположение и т.д. (раздел – подразделе-

ние – тип телефонного аппарата – расположение и номер телефона аудитории – абоненты).

30. Темы курсовых проектов (кафедра – группа – студент – дисциплина – семестр – тема курсового проекта).
31. Тестирование знаний: в БД хранятся билеты по дисциплинам и варианты ответов. По результатам тестирования подводится итог и выводится оценка (деканат – кафедра – группа – студенты – преподаватели – билеты – вопросы к билетам – результаты тестирования).
32. Университет (деканат – кафедра – группа – студенты – экзаменационные ведомости – дисциплины – преподаватели – оценки – договора поступления – оплаты).
33. Учебная загрузка аудиторий университета. В какой аудитории, у какой группы, по какой дисциплине, когда и какие занятия проводятся (деканат – кафедра – группа – здание – учебная аудитория – дисциплина – преподаватель – расписания – пункты расписаний).
34. Учет мебели (деканат – кафедра – ответственный на кафедре – акты – списки по актам – мебель – тип мебели).
35. Учет оборудования организации (кафедра – помещение – материально ответственный – оборудование).
36. Учет помещений университета: информация о корпусе, типе аудитории, принадлежности кафедре (корпус – аудитория – деканат – кафедра – ответственный за помещение).
37. Учет программного обеспечения (кафедра – аудитория – носитель программ – программа).
38. Учет сетевого и компьютерного оборудования (деканат – кафедра – аудитория – оборудование – тип оборудования – материально ответственный).
39. Учет электрооборудования: в какой аудитории, какие источники освещения, какое электрооборудование установлено, общая потребляемая мощность и т.д.

(здание – помещение – название и размещение электрооборудования – тип оборудования).

Приложение 2

ПРИМЕР ОФОРМЛЕНИЯ ОТЧЕТА

ОТЧЕТ
по лабораторной работе №1
Предметная область: **База данных «Конференции»**
Выполнил: студент группы ВТ-0-00 Иванов Иван
E-mail: name@domain.ru

Введение

В настоящее время нет общего ресурса, посвященного конференциям, существуют лишь отдельные попытки создания таких ресурсов, посвященных конференциям по узко-определенной тематике и, в основном, проводимых под руководством одной какой-либо организации.

Необходимо предоставить пользователю возможность удобного поиска необходимой ему информации о конференции, а также возможность организациям, проводящим конференции, оперативного ввода и редактирования информации о конференции.

В данной работе будет сделана попытка объединить возможности как ввода и редактирования информации, так и удобного поиска и просмотра.

Цель создания БД - быстро получить необходимую информацию о проходящих в ближайшее время конференциях по интересующей тематике.

Доводы в пользу создания информационной системы:

-нет единого источника информации по конференциям, не только общего, но и по определенной тематике;

-существуют некоторые источники, но вследствие статического представления на них данных они не обновляются очень долгое время, поэтому информация, предо-

ставленная на них, зачастую, не имеет никакой практической ценности;

-в связи со статическим представлением информации невозможно осуществить удобный поиск и навигацию по конференциям.

Создание информационной системы позволяет

-собрать всю необходимую информацию по конференциям заданной тематики;

-быстро найти и в удобной форме ознакомиться с информацией;

-обновлять информацию достаточно быстро, чтобы она имела практическую ценность.

К недостаткам использования информационной системы можно отнести необходимость

-создания структуры данных, многие сущности и атрибуты которой будут избыточны в силу того, что по многим конференциям невозможно предоставить информацию в полном объеме;

-структурирования информации перед вводом ее в систему.

Задачи и функции системы:

- ввод информации по конференции для заинтересованных участников;
- поиск и отображение необходимой информации;
- структурирование информации;
- получение статистической информации по состоянию БД.

В каждой СУБД могут быть свои специфические функции, кроме того, заказчик может потребовать реализации дополнительных функций, поэтому в отчете должен быть представлен расширенный список функций, чем дан выше, например, контроль данных превышение критических значений, конвертирование и загрузка данных в БД, др.

Критерии выбора СУБД:

- объем данных;
- простота структуры данных;
- удобный доступ к базе через Интернет.

Этим критериям удовлетворяет множество БД, поэтому можно рассмотреть дополнительные критерии, например, бесплатность БД (малая стоимость), надежность, кроссплатформенность.

Далее студенты должны самостоятельно найти характеристики СУБД, сравнить их с этими критериями и после этого обосновать выбор СУБД.

В пособии показаны простейшие реализации этой системы на следующих инструментах: MS Access, MySQL + PHP, Oracle + JSP.

Требования к программному обеспечению

Необходимо наличие программных продуктов, перечисленных выше, и ОС, в которой эти продукты достаточно надежно работают. Система должна работать в ОС - Windows 2000 и выше.

Требования к аппаратному обеспечению

Требования для различный реализаций резко отличаются, например, для устойчивого функционирования связки mysql+php вполне достаточно будет и Pentium 100 с 16Мб RAM (однако для Windows 2000 естественно требования более высокие), а для связки Oracle + JSP необходимо Pentium 500 и выше.

Проектирование базы данных

База данных состоит из шести связанных таблиц (5 основных и одной вспомогательной).

Вспомогательная таблица – **Страна**, в которой для удобства и точного ввода названий представлены все страны, эта таблица связана с полем Страна таблицы **Конференция**.

Таблица **Конференция** – основная таблица, в которой представлена информация о конференции: место и дата

проведения конференции, тема конференции и прочее; также с этой таблицей связаны все остальные.

Таблица **Член оргкомитета** – в этой таблице отображена информация о членах оргкомитета организации; она связана с полем Член оргкомитета таблицы **Конференция**.

Таблица **Исполнительный секретарь** – по структуре и предназначению подобна таблице **Член оргкомитета**.

Таблица **Секция** – в этой таблице отражена информация о всех секциях, проходящих в рамках данной конференции, связана с полем Секция таблицы **Конференция**.

Таблица **Материал** – в этой таблице содержится информация о каждом материале, представленном на конференции: т.к. материал обычно представляется в какую-либо секцию, то таблица связана с полем Материал таблицы **Секция**.

Таблица 1

Функции БД «Конференции»

Название функции	**Форма представления**
Загрузка данных	Справка по каждой таблице
Конвертирование данных	База данных
Контроль данных	Проконтролированная БД
Редактирование и ввод новых данных	Пополненная БД
Выполнение запросов на выборку данных	Справки
Получение статистики по БД	Справки
Подготовка отчетов	Справки

Этапы создания БД

- проведение обследования;
- разработка концептуальной схемы и структуры базы данных;

- разработка технологической схемы формирования БД;
- ввод информации в БД или конвертирование информации других систем;
- разработка приложений для вывода информации (запросов, отчетов, форм вывода).

Таблица 2

Перечень запросов

Формулировка запроса	**Форма представления**
Поиск по одной таблице - сведений о конференциях - материалов конференции	Справка
Поиск по двум таблицам - материалов конференции по номеру секции - конференции по автору доклада	Справка
Вычислительные функции −получить количество организованных конференций по странам, городам, организациям, годам	Таблица агрегированных данных

Структура таблиц

Структура таблиц создаваемой БД «Конференция» дана в табл.3- 8.

Таблица 3

Страна

Имя поля	**Формат**	**Полное название поля**
Id	integer	Идентификатор
Name	varchar	Название

Таблица 4

Конференция

Имя поля	Формат	Полное название поля
Id	integer	Идентификатор
Country	integer	Страна
City	varchar	Город
Place	varchar	Место проведения
Name	varchar	Название
short_name	varchar	Краткое название
date_begin	date	Дата начала
date_end	date	Дата окончания
date_register	date	Дата регистрации
date_art	date	Дата предоставления статей
Description	varchar	Краткое описание
Address	varchar	Почтовый адрес
Tel	varchar	Телефон
Fax	varchar	Факс
Email	varchar	E-mail
www	varchar	WWW-адрес

Таблица 5

Член оргкомитета

Имя поля	Формат	Полное название поля
Id	integer	Идентификатор
Name	varchar	Название
Address	varchar	Адрес
Tel	varchar	Телефон
Fax	varchar	Факс
Email	varchar	E-mail
www	varchar	WWW-адрес

Таблица 6

Исполнительный секретарь

Имя поля	Формат	Полное название поля
Id	integer	Идентификатор
Name	varchar	Название
Post	varchar	Должность
Address	varchar	Адрес
Tel	varchar	Телефон
Email	varchar	E-mail

Таблица 7

Секция

Имя поля	Формат	Полное название поля
Id	integer	Идентификатор
Number	integer	Количество работ
Name	varchar	Название
Description	varchar	Описание

Таблица 8

Материал

Имя поля	Формат	Полное название поля
Id	integer	Идентификатор
Name	varchar	Название
short_text	varchar	Аннотация
long_text	varchar	Текст
Author	varchar	Автор
Post	varchar	Должность
Address	varchar	Адрес
Tel	varchar	Телефон
Email	varchar	E-mail

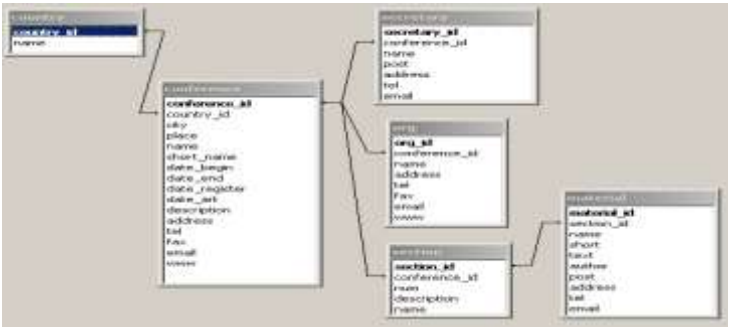


Рис.1. Схема БД

Формы ввода и вывода. Необходимо создать формы для

- ввода информации в различные таблицы БД;
- ввода параметров для осуществления поиска информации;
- вывода информации в достаточном объеме и в удобном виде.

Внешний вид форм ввода-вывода во многом зависит от инструмента, на котором реализована данная ИС.

Источниками информации являются заинтересованные организации, которые хотят поместить информацию о проводящихся под их руководством конференциях для доступа к ней заинтересованных лиц. Для подготовки информации использованы сайты, где размещена информация о конференциях в статическом виде (http://conf.msu.ru:7778/conference/static_html.show_start_page?lang_id_prm=2, http://www.allconferences.com/, http://www.rsci.ru/confs/). Кроме того, информация о конференциях имеется в Каталоге конференций.

ЛАБОРАТОРНАЯ РАБОТА № 2

СОЗДАНИЕ БД ДЛЯ ВЫБРАННОЙ ПРЕДМЕТНОЙ ОБЛАСТИ НА ОСНОВЕ СУБД ACCESS

Цель работы – изучение возможностей СУБД Access и функций ее отдельных компонент, приобретение навыков работы с таблицами, формами и запросами.

Этапы выполнения работы

Выполняется на основе материалов лабораторной работы №1 по проектированию БД для выбранной предметной области. Основные темы изучения архитектуры и функциональных возможностей СУБД Access представлены в табл.9.

Таблица 9

Изучение архитектуры и функциональных возможностей СУБД Access

Тема	Цель работы	Результат выполнения лаб. работы
Концепции и возможности	Изучение концепций и возможностей СУБД; анализ назначения и возможностей каждой компоненты системы	Студент должен знать методы проектирования БД, назначение, особенности и общую характеристику СУБД Access, уметь загружать систему, конфигурировать систему в соответствии с потребностями пользователя
Создание БД	Изучение функций: создание таблиц БД,	Студент должен знать принципы,

Тема	Цель работы	Результат выполнения лаб. работы
	ввод данных, разработка экранных форм для ввода данных	особенности и режимы реализации функции пополнения БД и обработки документов, уметь создать новый документ с использованием СУБД Access
Поиск данных	Изучение возможностей поиска данных	Студент должен знать принципы, особенности и режимы реализации функции поиска, уметь реализовать все поисковые технологии (проводить поиск, как по реквизитам, так и по тексту), сохранять и обрабатывать результаты поиска
Вывод	Разработка экранных форм для вывода данных	Результаты поиска данных представляются в виде созданных форм вывода
Вычислительные функции	Использование в формах вывода вычислительных функций (сумма, среднее, макс, мин)	Результаты вычислений оформляются в виде отдельной формы. Вычислительные функции представляются как

Тема	Цель работы	Результат выполнения лаб. работы
		количественные характеристики БД
Оформление БД	Создание главной формы. Подготовка файла помощи	На главной форме организуется запуск всех форм ввода данных, поиска, вывода, отчетов, статистики

При создании БД разрабатывается физическая структура данных, определяются связи таблиц, загружаются таблицы данных и выполняются операции с данными, файлами; для этого необходимо

-запустить СУБД Access, выбрать пункт меню База данных/Новая и создать базу данных;

-с помощью Конструктора разработать формы, вызов которых должен быть из главной формы;

-с помощью Мастера таблиц сформировать таблицу, определить форматы полей;

-ввести в БД информацию о конкретных документах и занести их в соответствующие таблицы.

При этом студент должен приобрести следующие знания и навыки:

- иметь общее представление о СУБД Access и основных операциях с базой данных (открыть, ввод, редактирование, поиск, сортировка, добавление новой записи, удаление, просмотр, отображение и др.);
- создания и изменения структуры таблиц, баз данных, связей таблиц, в том числе с помощью Мастера таблиц;
- ввод данных, в том числе из различных форматов;
- выполнение операций с файлами (восстановление, преобразование, сжатие, установка папки, защита дан-

ных; импорт данных из электронных таблиц, таблиц Word, текста, HTML; работа с объектами базы данных, репликация базы данных, работа с внешними данными, экспорт данных и др.);

- работа с запросами и фильтрами, вычисление статистических характеристик;
- умение создавать экранные формы и отчеты;
- вывод результатов в отчет, отправка по электронной почте, в печать, для Web.

Требования к приложению: необходимо создание главной формы с помощью Конструктора; на это форме должны быть название БД, кнопки входа для ввода данных, поиска, получения агрегированной информации, помощи и выхода из БД; форма ввода должна позволять вводить данные сразу во все таблицы; форма поиска создается по запросам, определенным в 1 л.р. Обратите внимание на поиск по двум таблицам и двум синхронизированным атрибутам; форма получения агрегированной информации должна выдавать таблицу; в форме помощи должна быть краткая текстовая информация о разработчике БД и сведения о БД (рис.2).

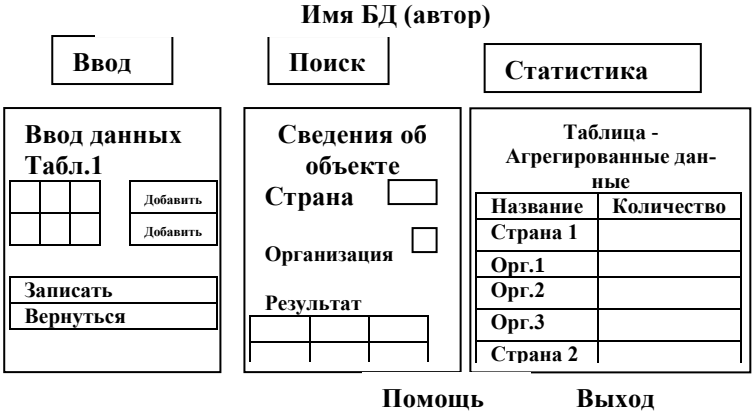


Рис.2 - Главная форма приложения

Ррекомендации по созданию приложения следующие.

Главная форма должна включать кнопки для перехода на функции:

- ввод и редактирование данных;
- поиск данных (по одной таблице и двум таблицам);
- получение агрегированных данных;
- сведения о БД;
- выход из приложения.

Форма ввода и редактирования должна обеспечивать целостность данных, поэтому должен быть синхронизирован ввод одних и тех же атрибутов находящихся в разных таблицах. При этом на одной форме должна быть возможность перехода на любую таблицу для ввода или редактирования.

Форма поиска должна представлять собой всплывающие списки синхронизированных атрибутов. Список должен состоять из неповторяющихся значений атрибутов. При выборе значения первого атрибута. появляется список для второго атрибута, соответствующий результатам поиска единственных значений второго атрибута по значению первого поискового атрибута.

Получение агрегированных характеристик представляется в виде краткой таблицы количественных значений содержания БД.

Форма отчетности предусматривает демонстрацию созданной БД на экране компьютера. Показываются заполненные информацией таблицы СУБД Access для выбранной предметной области. По полученным таблицам создаются экранные формы (для каждой таблицы отдельно и объединяющие данные нескольких таблиц) для выдачи результатов поиска, создается приложение для вычисления 3–5 расчетных величин.

При защите лабораторной работы студент должен продемонстрировать работу с таблицами базы данных, основными операциями с файлами, запросами и фильтрами, выводом данных.

Поиск должен быть организован по нескольким полям, при этом в список для поиска должны быть занесены все значения поискового поля, чтобы не вводить каждый раз значение поля для поиска, а просто выбрать – быстро и без ошибок. Значения поисковых атрибутов выбираются из таблицы БД, где они используются (как правило, в главной таблице). Необходимо реализовать три варианта поиска: поиск по одной таблице, поиск по двум таблицам и получение справки о БД – количество объектов по нескольким наиболее интересным (важным) атрибутам, характеризующим предметную область (например, количество объектов по странам, организациям).

На кнопочной форме создать небольшой файл с текстом помощи для БД (кто сделал, что это за БД и краткие ее характеристики – объем, состав, структура).

Нужна одна главная форма, на которой создать кнопки с формами ввода данных, поиска информации, файла с текстом помощи, выход и БД). В каждой вспомогательной форме должен быть возврат в главную форму.

Общие требования к оформлению л.р. - Если 1 лаб. раб. сдана в марте, то объем 2 л.р. увеличивается в два раза (число строк в главной таблице увеличивается до 100, число таблиц до 10, число атрибутов до 50).

- Если 1 лаб. раб. сдана в апреле, то объем 2 л.р. увеличивается в три раза (число строк в главной таблице увеличивается до 200, число таблиц до 15, число атрибутов до 75).

Кто сдаст все лаб. раб. до 15 апреля может рассчитывать на получение четырех балов за экзамен "автоматом". На 5 надо сдавать экзамен.

Пересылка лаб. работ осуществляется по электронной почте vjaz@meteo.ru. При пересылке лаб. работ надо соблюдать следующие правила:
Текст сообщения должен включать: обращение, текст сообщения (что направляется) и подпись (ФИО, группа).
Приложения к сообщению оформляется с.о.: имя файла должно отражать обязательно фамилию студента - в начале имени файла.
3. Приступать к выполнению 2-ой лаб. раб. можно только после утверждения темы и согласования состава структуры БД.

Типовые ошибки для 2 и 3 л.р. Поиск по двум таблицам и двум синхронизированным значениям атрибутов.

Необходимо из таблицы, где находится поисковый атрибут, выбрать его единичные значения и представить пользователю в виде всплывающего списка. Пользователь выбирает необходимое значение по выбранному пользователем первого значения атрибута. Запускается второй селект на выборку единичных значений для второго поискового атрибута, который представляется во втором списке. Пользователь выбирает значение второго поискового атрибута и отправляется третий селект на поиск по двум выбранным пользователем значениям атрибутов. Здесь всегда должен быть результат.

Статистика должна представлять собой одну таблицу, в которой будет перечень объектов и количество свойств по ним.

Ввод данных. Для обеспечения целостности БД необходимо ввод данных организовать с главной таблицы. Если необходимого значения атрибута во всплывающем списке нет. То рядом с этим полем ставится кнопка Добавить, по которой вызывается соответствующая форма для ввода. Например, если нет Издателя, вызывается таблица Издатель и заполняются все атрибуты.

Все названия полей для ввода пишутся на русском языке, в скобочка указать единицы измерения для значений атрибутов или формат поля, др.

Идентификатор должен присваиваться автоматически, оператор не обязан это отслеживать.

Результаты проверки на предельные значения числовых атрибутов и дат должны выводиться сразу после ввода (поле ввода подсвечивается красным цветом), а не при сохранении всей записи. иначе непонятно к какому атрибуту выдана ошибка.

Небольшие перечни (классификации объекта) лучше вводить через всплывающий список, иначе его написание невозможно стандартизовать.

После ввода нового объекта (любого значения атрибута - жанр, язык) значения атрибутов должны появляться во всплывающих списках для ввода и поиска.

При вводе нужно обеспечить синхронизацию атрибутов. Если оператор выбрал Microsoft, то не должна для ввода появляться торговая марка для другого издателя.

Всплывающие списки при выводе необходимо сортировать, чтобы оператору и пользователю было легче выбрать необходимое значение.

Объем. База данных должна включать в себя не менее пяти таблиц; общее число атрибутов не менее 25. БД должна включать в себя 50 строк в главной таблице по выбранной предметной области.

Срок представления: в течение второго месяца занятий.

Литература

1. Microsoft Access 97: Справочник. Наиболее полное справочное руководство. – Санкт - Петербург, Москва, Харьков, Минск. 1998. – 410 с.
2. http://www.hiprog.com/access/default.asp
3. http://msa.polarcom.ru/acclinks.htm

Приложение 3

ПРИМЕР СОЗДАНИЯ БД НА ОСНОВЕ ПРЕДМЕТНОЙ ОБЛАСТИ «КОНФЕРЕНЦИИ»

Для создания БД в СУБД Access необходимо выполнить следующие основные шаги.

Запустить СУБД Access.

Выбрать название проектируемой БД – conference.

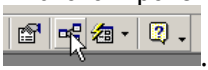
Выбрать режим «Создание таблиц в режиме конструктора» и создать структуру БД (рис.3) в соответствии с разработанной в лабораторной работе №1.

conference : таблица

Имя поля	Тип данных
conference_id	Счетчик
country_id	Числовой
city	Текстовый
place	Текстовый
name	Текстовый
short_name	Текстовый
date_begin	Дата/время
date_end	Дата/время
date_register	Дата/время
date_art	Дата/время
description	Поле MEMO
address	Текстовый
tel	Текстовый
fax	Текстовый
email	Текстовый
www	Текстовый

Рис.3. Таблица Conference в процессе создания

В итоге должно быть создано шесть таблиц.

Теперь необходимо задать нужные связи между таблицами. Для этого производим клик по кнопке «Схема данных» - . По правой кнопке добавляем все наши таблицы; теперь создаем связи следующим образом: нажимаем мышкой на атрибуте одной сущности и тянем до соответствующего атрибута другой, получаем картину, рис.4.

Заполнить таблицы данными. Для заполнения необходимо просто кликнуть на имени таблицы.

	country_Id	name
+	1	Россия
+	2	США
+	3	Великобритани
+	4	Испания
+	5	Дания
+	6	Швеция
+	7	Германия
+	8	Япония
+	9	Украина
+	10	Канада
+	11	Австралия
	(Счетчик)	

Рис.4. Таблица Country

Аналогично заполняем остальные таблицы.

Создать формы для ввода данных и представления результатов поиска. Сначала создадим простейшую форму ввода-вывода данных, для этого выбираем раздел Формы – Создание с помощью мастера (рис.5).

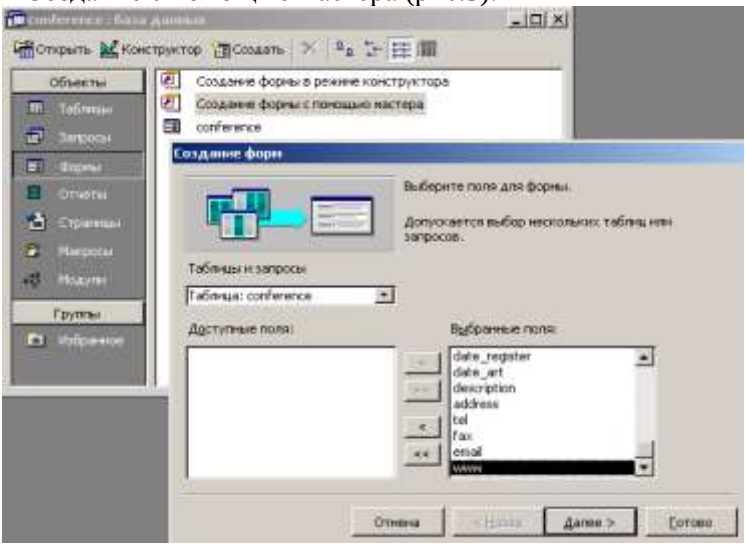


Рис.5. Этапы создания формы

Переносим все поля, например, из таблицы conference, нажимаем Готово, получаем следующую картинку - рис.6.

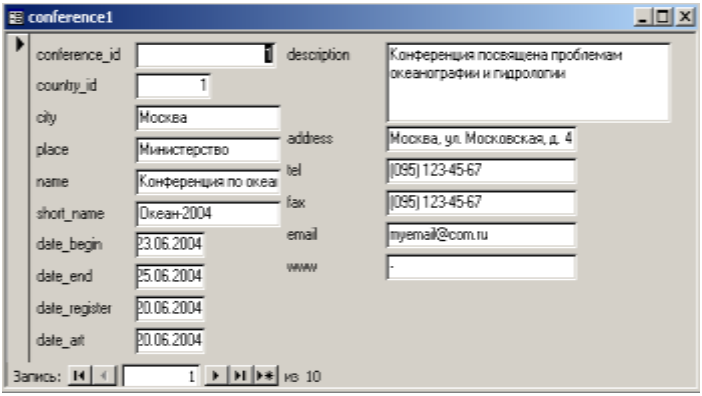


Рис.6. Этапы создания формы

Это автоматически сформированный вид формы. Необходимо более удобно расположить элементы; для этого просто перетаскиваем их в нужные места формы, выбирая форму в панели Формы, и производя клик на пункт Конструктор, – форма открывается в режиме конструктора и после некоторых манипуляций принимает вид, приведенный на рис.7.

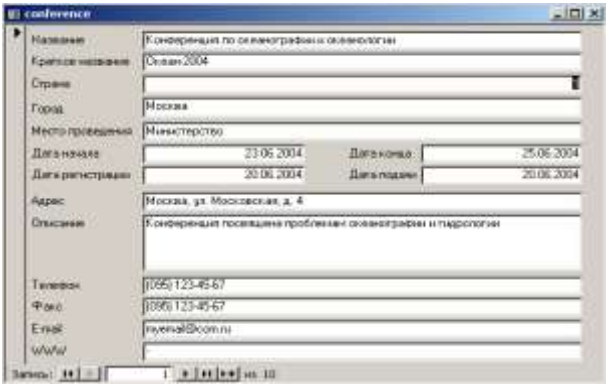


Рис.7. Этапы создания формы

У данной формы есть некоторые недостатки: во-первых, вместо названия страны отображается идентификатор из таблицы country, во-вторых, нигде не отражена информация об организаторах, секретарях, секциях и материалах, относящихся к данной конференции.

Чтобы осуществить выбор страны, разумно использовать элемент формы типа список, выбираем его и вставляем в форму, при этом таблицу выбираем country и оба столбца из нее, причем оставляем отмеченной галочку «прятать ключевой столбец». После этого правой кнопкой мыши или двойным щелчком на этом элементе (в режиме Конструктора) можно вызвать его свойства, заходим на вкладку Данные и в верхней строке выбираем country_id – все готово, осталось только заменить элемент формы и форма примет вид, рис.8.

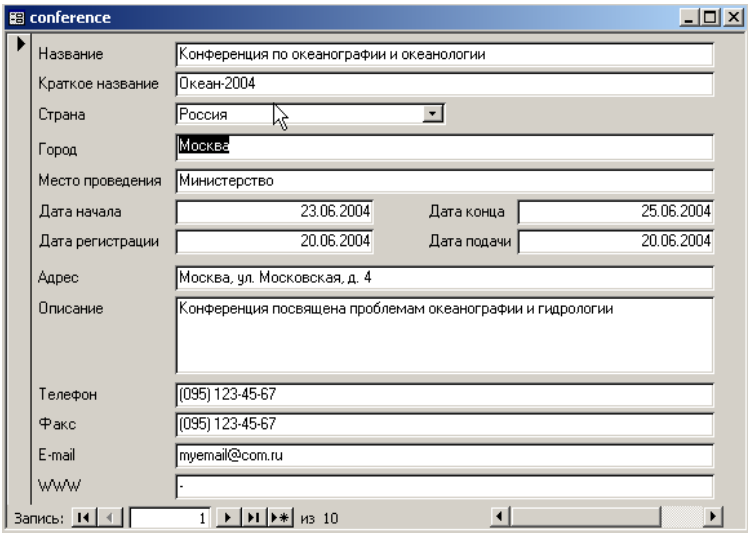


Рис.8. Этапы создания формы

Создадим еще две формы (для таблиц Организатор и Секретарь) подобным же образом, они представлены на рис.9–109.

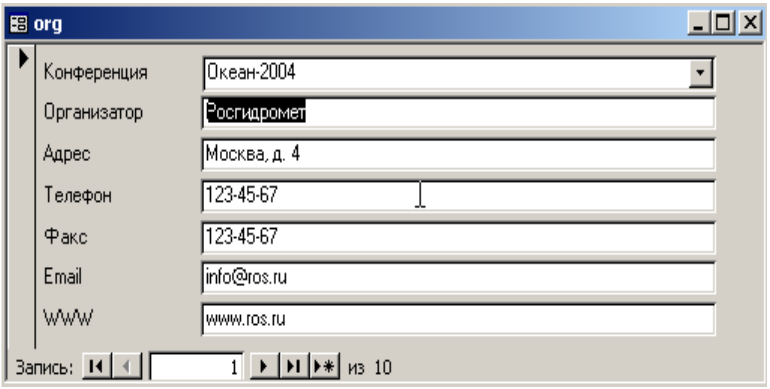


Рис.9. Форма «Организатор»

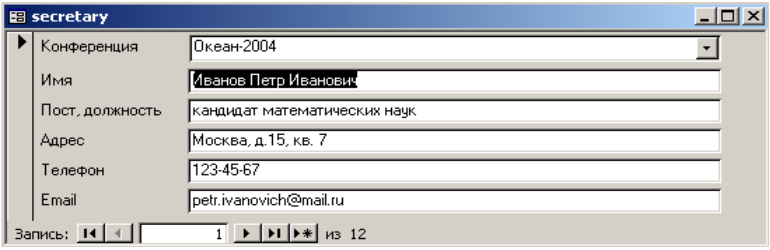


Рис.10. Форма «Секретарь»

Теперь, чтобы объединить все данные, воспользуемся инструментом Подчиненная форма. Чтобы сделать вывод информации более удобным, выберем инструмент Набор вкладок и сделаем две вкладки – Организатор и Секретарь, в каждую из которых вставим соответствующую подчиненную форму.

Нижняя часть формы Конференция выглядит теперь так – рис.11.

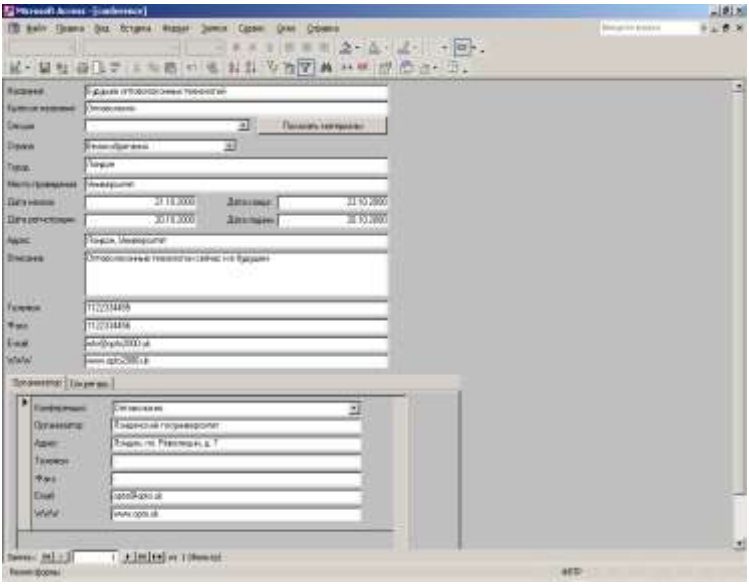


Рис.11. Подчиненная форма

ЛАБОРАТОРНАЯ РАБОТА № 3
ДОСТУП К БД ЧЕРЕЗ WEB

Цель работы: изучение возможностей различных средств доступа к данным через Web для создания приложения для доступа к БД на основе выбранной предметной области.

Этапы выполнения работы

Выбирается одно из средств доступа к данным через web: Php + MySql (сервер Apache); JSP + Oracle (сервер JBOSS). Возможно использование и другой линейки инструментов.

Устанавливается все необходимое программное обеспечение. Конвертируется БД Access с помощью ESF Database Migration Toolkit - Professional.

Разрабатывается приложение для доступа к данным через web - страницу. При этом студент должен

приобрести знания и навыки по созданию, разметке Web-страниц; динамическому представлению данных на Web - странице.

Студент должен приобрести следующие знания и навыки:

- умение установить и настроить все необходимое программное обеспечение;
- разработки БД в среде выбранной СУБД;
- создания динамических web-страниц.

Форма отчетности. Созданное приложение помещается на бесплатный хостинг, например, по адресам http://runetovec.ru/showrazdel.php?razdelid=2803, http://register.h15.ru/about.shtml, https://www.reg.ru/user/?rtm_source=e-mail&rtm_medium=e-mail&rtm_campaign=BooksRegKorolyuk.

Адрес приложения на бесплатном хостинге высылается преподавателю по электронной почте. При защите лабораторной работы студент должен продемонстрировать возможности динамической работы с базой данных.

Объем данных соответствует объему созданной БД в лабораторной работе № 2. Созданное приложение функционально должно повторять приложение, реализованное в СУБД Access (ввод данных, поиск данных, статистическая справка по БД).

Срок представления: в течение третьего месяца занятий.

Литература

1. Матросов А., Сергеев А., Чаунин М:. HTML 4.0. Новый уровень создания HTML -документов. – СПб.: БХВ-Петербург. 2000. – 672 с.
2. Фролов А., Фролов Г. Практика применения Perl, PHP, Apache, MySQL для активных Web-сайтов. Издательство «Русская Редакция». – 2002. – 576 с. В комплекте с CD.

3. Использование технологий WWW для доступа к базам данных. (Материал подготовлен *ЦНИТ НГУ*). http://www.citforum.ru/database/cnit/index.shtml
4. Дейв Энсор, Йен Стивенсон. Oracle – 8: Рекомендации разработчикам. – Киев: Издательская группа BHV. – 1998. – 125 с.

Приложение 4

СОЗДАНИЕ БД «КОНФЕРЕНЦИИ» С ИСПОЛЬЗОВАНИЕМ СУБД MYSQL И НАПИСАНИЕ ПРИЛОЖЕНИЯ К НЕЙ НА СКРИПТОВОМ ЯЗЫКЕ PHP

Установка необходимого программного обеспечения

Поскольку будет создано приложение для доступа к БД через Интернет, то необходимо следующее программное обеспечение:

-сервер (был выбран сервер Apache как наиболее распространенный в Интернет);

-СУБД (была выбрана СУБД MySQL);

-язык, позволяющий создавать документы, в которых будет представлена необходимая информация из БД (был выбран язык PHP).

Можно скачать все эти компоненты по отдельности, установить их и настроить; если кто-то собирается серьезно заниматься программированием в Интернет, можно посоветовать проделать это хотя бы один раз.

Однако для выполнения данной лабораторной работы можно пойти более простым путем – использовать пакет «Денвер» (джентльменский набор web-разработчика), где все необходимые компоненты интегрированы в один модуль, причем вся настройка будет осуществлена автоматически. Дистрибутивы можно взять на сайте http://www.dklab.ru или сайте http://vt.obninsk.ru в разделе

Интернет-технологии. Настройка этой программы настолько проста, что нет необходимости описывать ее.

Создание виртуального сервера

После установки Денвера у Вас в системе появится диск Z. Заходим в папку home – это корень Вашего web-сервера, создаем в ней папку conference, в которой создаем папку www – в этой папке будут лежать документы вашего сервера.

Создание БД «Конференции»

Поскольку была выбрана БД MySQL, удовлетворяющая стандартам SQL, то перед созданием БД необходимо ознакомиться с самим языком SQL либо с его конкретной реализацией в MySQL.

Существует несколько способов для создания БД: можно создавать БД из консоли самой СУБД MySQL, можно использовать специальные утилиты.

Однако довольно часто и достаточно удобно использовать phpMyAdmin – специальный модуль, написанный на языке PHP и осуществляющий взаимодействие с БД MySQL. В Денвер входит этот модуль.

Запускаем Apache вместе с MySQL (запускаем файл start.bat, созданный после установки Денвера). Чтобы загрузить phpMyAdmin, набираем в строке браузера http://localhost/phpmyadmin/. У Вас появится следующая картинка – рис.12.

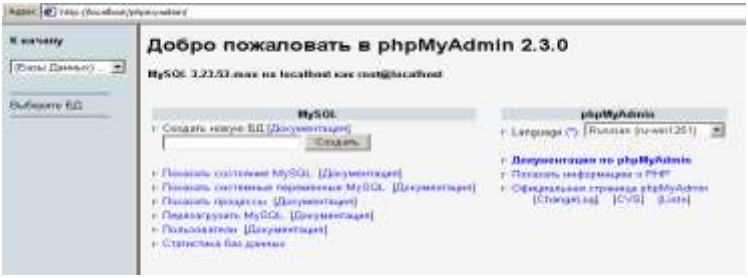


Рис.12. Вход в phpMyAdmin

Создаем новую БД «Конференции» – рис.13.



Рис.13. Создание новой базы данных с помощью phpMyAdmin

Создание таблиц БД

Создавать таблицы БД можно как с помощью запросов SQL, так и с помощью самого phpMyAdmin (ниже будут продемонстрированы оба метода).

Создание с помощью запроса SQL – нажимаем ссылку SQL, появляется следующая картинка, рис.14.

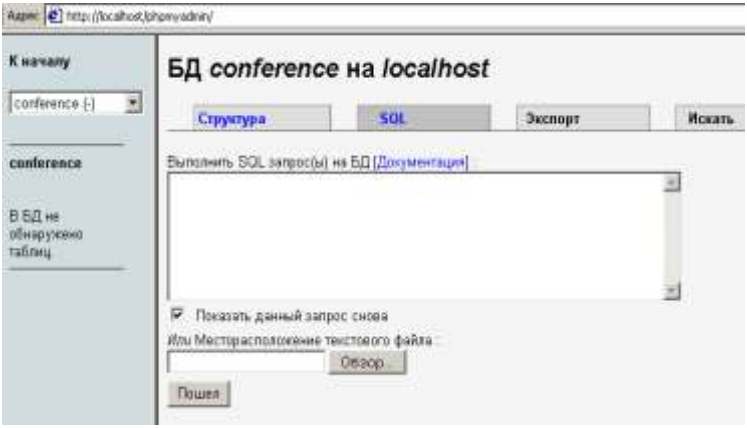


Рис.14. Создание таблиц в БД с помощью запроса SQL

В поле набираем SQL-запрос

```
CREATE TABLE conference (
        id int(20) NOT NULL auto_increment,
        country_id int,
        name varchar(255),
        date date,
        place varchar(255),
        description varchar(255),
        address varchar(255),
        tel varchar(150),
        fax varchar(150),
        email varchar(100),
        www varchar(100),
        PRIMARY KEY  (id)
) TYPE=MyISAM;
```

Нажимаем «Пошел», по запросу создается таблица – рис.15.

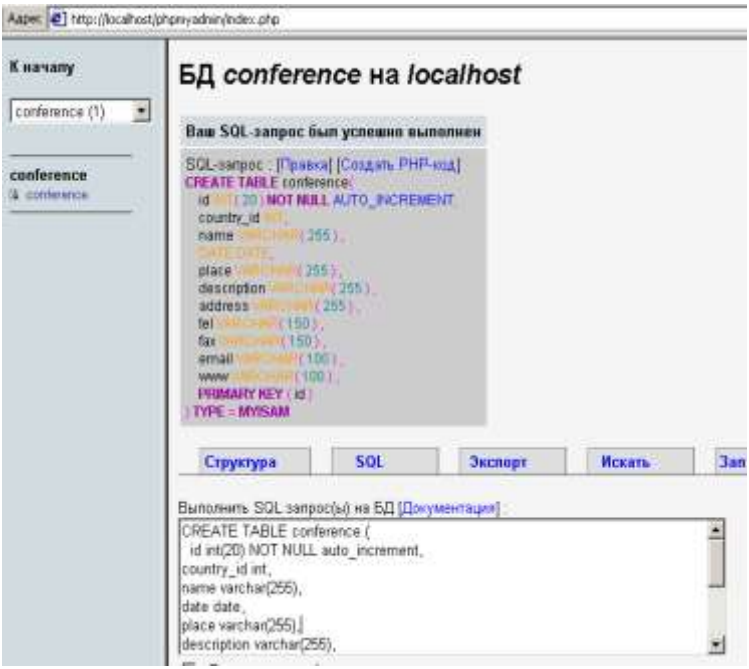


Рис.15. Создание таблиц в БД с помощью phpMyAdmin

Создание таблицы с помощью phpMyAdmin – рис.16.

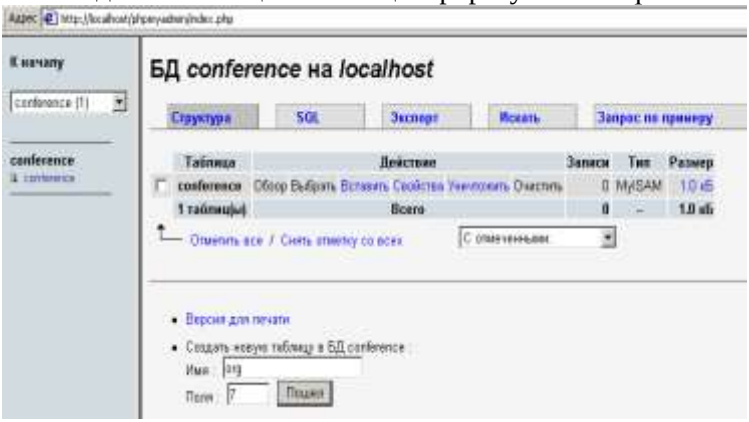


Рис.16. Автоматическое создание таблиц в БД с помощью phpMyAdmin

Создаем таблицу с именем org и количеством полей – 7 (рис.17).

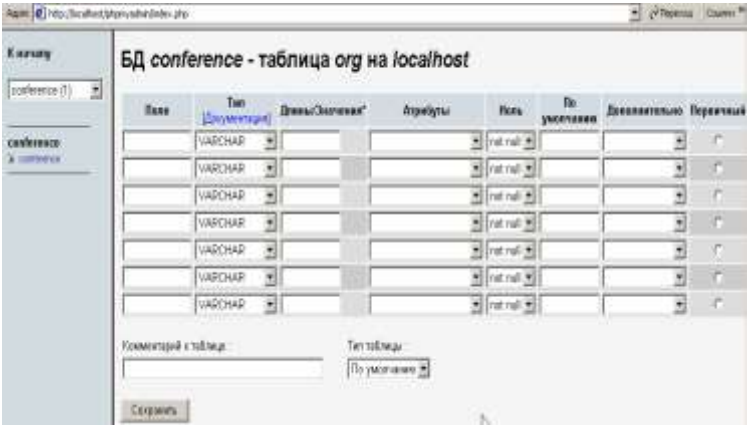


Рис.17. Автоматическое создание таблиц в БД с помощью phpMyAdmin

Для каждого столбца задаем тип поля, его длину и прочие параметры (рис.18).

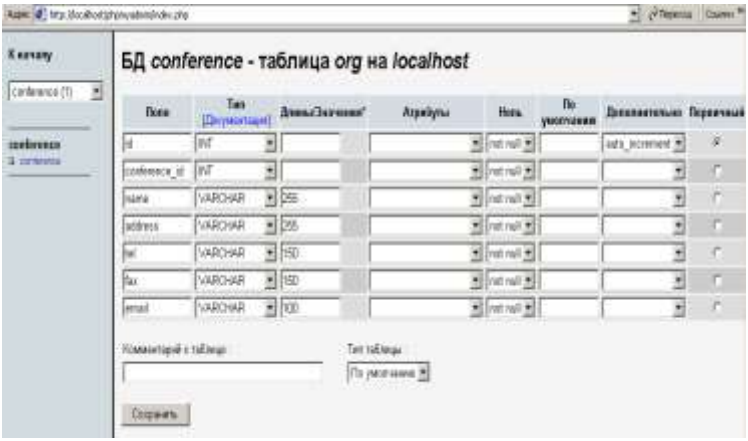


Рис.18. Автоматическое создание таблиц в БД с помощью phpMyAdmin (задание типов и размерностей)

Аналогично создаем остальные таблицы (см. приложение 1).

Создание приложения для работы с БД на языке PHP

Для создания приложения необходимо знать основы языка PHP. Минимальный набор команд:

echo – вывод информации в поток вывода (информация передается Apache и через него браузеру клиента);

while – команда цикла;

if – команда ветвления;

mysql_connect, mysql_select_db, mysql_query, mysql_num_rows, mysql_fetch_assoc – команды работы с языком SQL.

Две основные части приложения – это ввод новых данных и отображение (поиск) данных.

Приложение 5

СОЗДАНИЕ БД «КОНФЕРЕНЦИИ» С ИСПОЛЬЗОВАНИЕМ СУБД ORACLE И НАПИСАНИЕ ПРИЛОЖЕНИЯ К НЕЙ НА СКРИПТОВОМ ЯЗЫКЕ JSP

Требования к серверу

- Windows 2000/Oracle/JSDK/JVM/JBOSS.
- Возможна замена сервера JBOSS на другой.

Установка и настройка необходимого программного обеспечения. Как уже говорилось ранее, все программное обеспечение устанавливается на windows2000, в других ОС этапы установки могут отличаться.

Установка и настройка СУБД Oracle. Для примера была использована версия СУБД Oracle 9i. В установке этой программы не должно возникнуть никаких сложностей – все параметры устанавливаются по умолчанию, необходимо только задать имя базы; в нашем случае имя базы было выбрано «test».

Также после установки необходимо зайти в настройки Служб (Пуск – Настройка – Панель управления – Администрирование – Службы), остановить все службы Oracle (чтобы пока они не мешали) и выставить им ручной старт вместо автоматического. СУБД Oracle в нашем случае был установлен в папку d:\oracle.

Установка и настройка Java. В нашем случае была установлена версия «j2sdk1.4.1_02»; с ее установкой также не было никаких проблем. После установки необходимо прописать в переменных окружения JAVA_HOME = c:\j2sdk1.4.1_02 (путь, куда установлен пакет), а также добавить в path путь к исполняемым файлам (в нашем случае – c:\j2sdk1.4.1_02\bin).

Установка и настройка сервера. Для организации работы клиент-сервера на клиентской стороне необходим браузер, а на серверной – сервер, умеющий обрабатывать java-сервлеты. Был выбран сервер JBOSS; его можно не

устанавливать, а просто скачать уже установленные дистрибутивы (в нашем случае он был установлен в папку C:\JBOSS).

Установка драйверов для связи с СУБД Oracle. Для того, чтобы java могла работать с Oracle, необходимо установить драйверы (классы, обеспечивающие взаимодействие). Ищем в папке D:\oracle\ora90\jdbc\lib файл classes12.jar, копируем его в папку C:\j2sdk1.4.1_02\jre\lib\ext.

Установка редактора. Не лишним будет установка какого-нибудь специализированного редактора, типа Idea, Kawa (хотя можно писать код и в блокноте).

Запуск необходимых приложений

Запуск служб СУБД Oracle. Заходим в панель управления службами и запускаем службы «OracleOraHome90TNSListener» и «OracleServiceTEST» (название второй службы зависит от того, как Вы назовете базу).

Запуск сервера. Запуск JBOSS осуществляется с помощью файла C:\jboss\bin\run.bat.

Создание таблиц в БД Oracle. Создавать таблицы можно несколькими способами.

Создадим одну из них с помощью утилиты SQL Plus (Oracle–OraHome90 – Application Development – SQL Plus).

Запускаем утилиту; параметры соединения: имя пользователя – system, пароль – manager.

После выдачи приглашения «SQL >» можно набирать команды языка SQL.

Итак, создаем таблицы: SQL> create table country (id integer, name varchar(255));

Проще и намного удобнее создавать таблицы не с помощью консоли, а с помощью специальных программ, например, PL/SQL Developer (рис.19). При входе пишем, как и ранее, system – manager, БД – test.

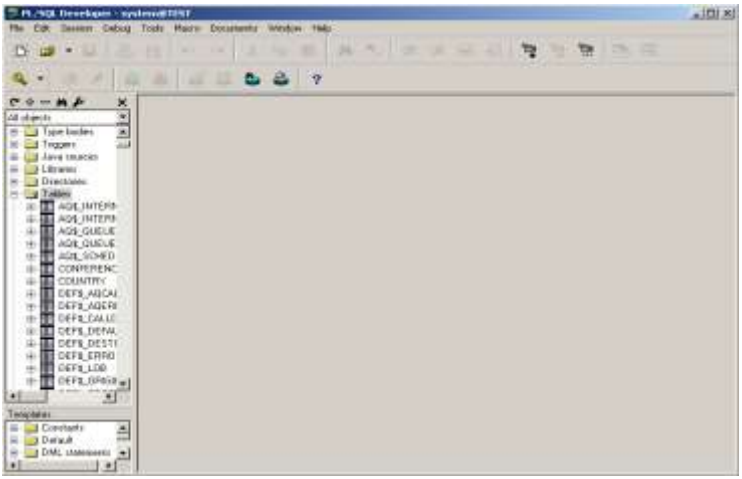


Рис.19. Внешний вид PL/SQL Developer

На вкладке Tables видим множество служебных таблиц и наши созданные таблицы. Чтобы создать новую таблицу, нажимаем правой кнопкой на Tables и выбираем New (рис.20).

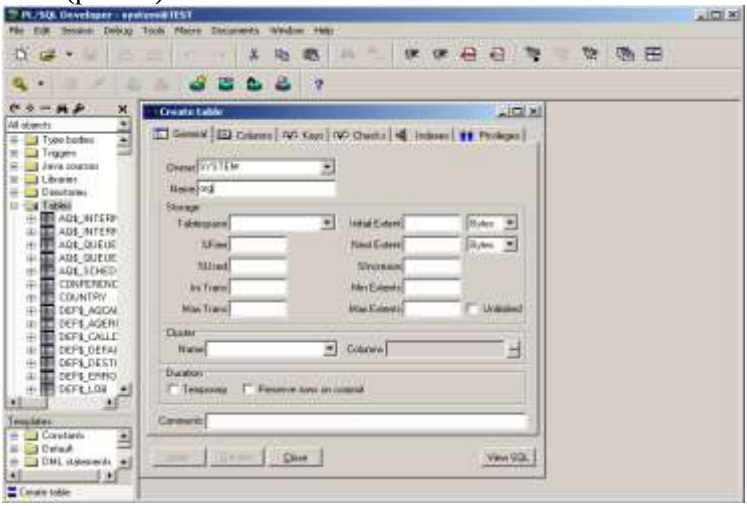


Рис.20. Создание таблицы в PL/SQL Developer

Owner – SYSTEM, Name (название таблицы) – org. После этого переходим на вкладку Columns и создаем структуру таблицы (рис.21).

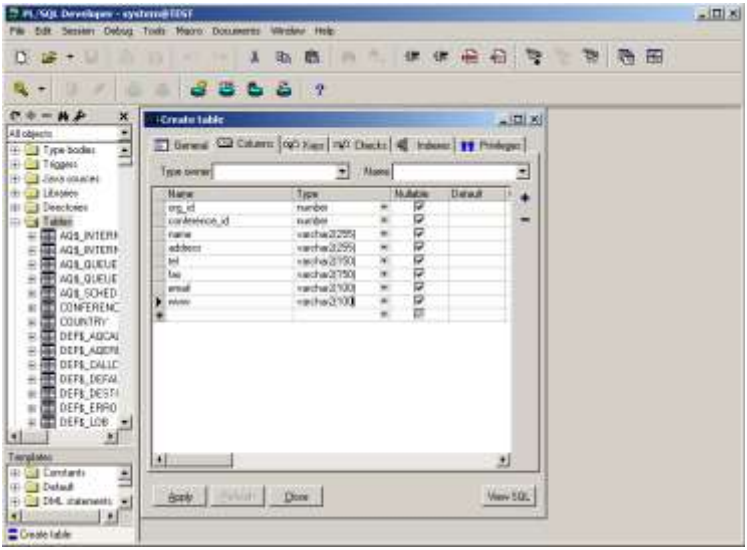


Рис.21. Создание таблицы в PL/SQL Developer

На вкладке Keys задаем первичный ключ с именем поля org_id для таблицы, состоящий из одной колонки, и нажимаем Apply – таблица создана.

Аналогично создаем остальные таблицы. Данный пример показывает лишь создание простейших таблиц без связей и внешних ключей.

Также необходимо в каждой таблице для ключевого поля реализовать систему автоматической раздачи ключей. Это можно сделать с помощью так называемых последовательностей (рис.22).



Рис.22. Создание последовательности в PL/SQL Developer

Выбираем пункт Sequences, owner, как и ранее, system; т.к. счетчик будет использоваться для таблицы conference, то и назовем его соответственно conference_id. Необходимо задать также Min value = 0, Start with = 0, Increment by = 1, остальные значения будут подставлены автоматически.

Создание простого приложения на Java-сервлетах

Место размещения сервлетов будет папка C:\jboss\server\default\deploy\conference.war (папка должна иметь расширение war и лежать именно по такому пути относительно сервера JBOSS). Запускаем JBOSS; для тестирования в данной папке создадим файл index.html, например, следующего вида:

```
<html>
<head>
<title>Лабораторная работа №3. Oracle + JSP</title>
</head>
<body>
```

Работает!
</body>
</html>

Через браузер получить доступ к этому файлу можно по следующему пути: http://localhost:8085/conference/. Основной файл add.jsp используется для добавления информации в БД. Код java в сервлетах обрамляется тэгами <% %>. Код реализует подключение классов пакета java.sql (необходимого для взаимодействия с БД), кроме того, он задает необходимые кодировки страницы и принимаемых данных (класс request отвечает за прием данных).

```
<%@ page contentType = "text/html; charset=WINDOWS-1251" %>
<%@ page import  = "java.sql.*" %>
<% request.setCharacterEncoding("Cp1251"); %>
```

Следующий фрагмент реализует соединение с БД, причем для нашего случая используются следующие параметры соединения:

```
url = "jdbc:oracle:thin:@localhost:1521:test",
login = "system", password = "manager";
try {
DriverManager.registerDriver(new ora-
cle.jdbc.driver.OracleDriver());
conn = DriverManager.getConnection(url, login, password);
} catch(Exception e) {
out.println("Невозможно установить соединение: " + e);
}
```

Метод getParameter класса request используется для доступа к cgi-переменным, которые передаются методами get или post. Сам запрос к БД использует метод executeQuery класса Statement. Поскольку скрипт add.jsp используется только для ввода данных в таблицу, то обработка выборки в нем не реализована (она будет показана

ниже). Покажем простейшую реализацию запроса – ввод данных в таблицу country:

```
result = stmt.executeQuery("insert into country (country_id,name) values (country_id.NEXTVAL,'"+country+"')")
```

В запросе используется последовательность country_id и ее свойство NEXTVAL.

Ввод в другие таблицы реализован аналогично, за исключением следующей детали. Некоторые поля реализованы как select на странице для удобства ввода, например, при вводе информации по конференции ввод страны осуществляется не в поле типа текст, а в поле select, причем значения этого поля формируются автоматически (здесь и будет показана работа с запросом из БД). В данном случае выборка и обработка данных осуществляется так:

```
result = stmt.executeQuery("select * from country");
while (result.next())
{
out.println("<option value='"+result.getString(1)+"'>"+result.getString(2)+"</option>\n");
}
```

Как видно из примера выше, result.next() возвращает всю строку данных, а потом методом getString получаем необходимый столбец (индекс столбца начинается с «1»). Файл view.jsp используется для просмотра информации из БД. Работа скрипта состоит в следующем: при выборе соответствующей таблицы, например, таблицы конференций, в компактном виде загружается информация обо всех конференциях (их названия) и для каждой конференции существует возможность просмотра всей информации или ее редактирования (аналогично и для остальных таблиц).

Для того, чтобы была возможность редактирования информации, необходимо модифицировать скрипт add.jsp: если на входе скрипта установлена переменная id, то счи-

тается, что пользователь не создает новую запись, а редактирует ее с заданным id.

Приложение 6

ЭКСПОРТ ДАННЫХ ИЗ СУБД ACCESS В СУБД ORACLE

В том случае, если необходимо разовое конвертирование Access -> Oracle (,а не регулярное), то можно поступить так, как указано в прилагаемом файле. Если же необходима автоматизация этого процесса, то необходимо создавать VBA процедуры в Access.

Разовое конвертирование Access -> Oracle включает экспорт таблиц и их загрузка в Oracle.

Экспорт таблиц Access в ASCII файлы

1) В окне БД Access, данные которой необходимо конвертировать, в разделе *Таблицы* выделить нужную таблицу;
2) Меню *Файл -> Экспорт*;
3) В появившемся окне задать каталог, куда будет экспортирована таблица, а в поле *Тип файла* выбрать *Текстовые файлы (*.txt;*.csv;*.tab;*.asc)*, задать имя фала (можно такое же, как имя экспортируемой таблицы, но обязательно латинскими буквами). Нажать кнопку *Экспорт*;
4) В появившемся окне мастера выбрать *с разделителями – поля разделяются запятыми или табуляцией*, нажать *Далее*;
5) В разделе *Разделитель полей:* выбрать *точка с запятой*, убрать галочку в поле *Включить имена полей в первой строке*, если она установлена, в поле *Ограничитель текста:* выбрать *{нет}*, нажать *Далее*;
6) В поле *Экспорт в файл* задать строку в виде <Путь к файлу><Имя файла><**.**txt>
7) Зайти в каталог, куда была экспортирована таблица и переименовать файл в <Имя файла><**.**dat>.

Загрузка ASCII файлов в Oracle с помощью утилиты SQL*Loader

1) Создать в БД Oracle таблицу по составу и типу полей аналогичную полям таблицы Access. Например:

```
create table T1
(field1 number,
field2 varchar2(50))
```

2) Создать управляющий файл с расширением .ctl (имя файла может быть идентичным имени таблицы), который ставит в соответствие поля ASCII файла атрибутам таблицы в БД Oracle. Например:

```
LOAD DATA
INTO TABLE "T1"
APPEND
FIELDS TERMINATED BY ';'
(
 FIELD1,
 FIELD2
)
```

Здесь:

T1 – название таблицы в БД Oracle;

FIELD1, FIELD2 – название атрибутов (полей) таблицы в порядке следования в ASCII файле.

Сохранить .ctl файл в той же директории, что и ASCII файл;

3) Меню *Пуск -> Выполнить….* Ввести cmd. В появившемся окне консоли перейти в каталог с ASCII и ctl файлом и выполнить команду вида:

```
sqlldr userid=username/password@DB control=T1.ctl data=T1.dat log=T1.log
```

Здесь:

username – имя пользователя в БД Oracle;

password – пароль пользователя в БД Oracle;

DB – имя вашей БД Oracle (SID или ServiceName);

T1.ctl – имя управляющего файла;

T1.dat – имя ASCII файла данных;

T1.log – имя журнального файла. Он будет сформирован по окончании загрузки.

4) Если все прошло успешно в созданном журнальном файле T1.log быть запись:

   Таблица "T1":

   n Строки успешно загружено.

где n – количество строк в Вашем ASCII файле.

В том случае, если по каким-то причинам не удалось загрузить данные, то об этом будет отмечено в журнальном файле T1.log. При этом «забракованные» записи данных будут содержаться в файле T1.bad.

ЛАБОРАТОРНАЯ РАБОТА № 4
ДОСТУП К БД ЧЕРЕЗ WEB (ОБЛАЧНАЯ РЕАЛИЗАЦЯ)

Цель работы: изучение возможностей облачного сервиса Microsoft Azure для создания БД на основе выбранной предметной области, а также приложения, используемого для доступа к созданной БД через Web.

Этапы выполнения работы

Для выполнения лабораторной работы необходимо сначала завести учётную запись Microsoft, а затем аккаунт Microsoft Azure (на данный момент для студентов доступна бесплатная версия). Зарегистрировавшись, студент должен создать пустую БД на Azure.

Устанавливается все необходимое программное обеспечение. Необходимо установить среду разработки ПО (рекомендуется Microsoft Visual Studio 2015). При помощи VS 2015 подключиться к созданной на Azure пустой БД.

Разрабатывается приложение для доступа к данным через web - страницу. При этом студент должен приобрести знания и навыки по созданию, разметке Web-страниц; динамическому представлению данных на Web - странице.

Студент должен приобрести следующие знания и навыки:

- умение установить и настроить все необходимое программное обеспечение;
- разработки БД в среде выбранной СУБД;
- создания динамических web-страниц.

Форма отчетности. Созданное приложение помещается на облачный хостинг Microsoft Azure.

Адрес приложения на облачном хостинге высылается преподавателю по электронной почте. При защите лабораторной работы студент должен продемонстрировать возможности динамической работы с базой данных.

Объем данных соответствует объему созданной БД в лабораторной работе № 2. Созданное приложение функционально должно повторять приложение, реализованное в СУБД Access (ввод данных, поиск данных, статистическая справка по БД).

Срок представления: в течение третьего месяца занятий.

Литература

1. Документация по работе с Microsoft Visual Studio 2015. https://msdn.microsoft.com/ru-ru/library/dd831853.aspx
2. Документация по работе с Microsoft Azure. https://docs.microsoft.com/ru-ru/azure/
3. Справочник по работе с HTML.
4. http://htmlbook.ru/HTML

Приложение 7

СОЗДАНИЕ БД С ИСПОЛЬЗОВАНИЕМ СУБД SQL AZURE И НАПИСАНИЕ ПРИЛОЖЕНИЯ К НЕЙ НА ОБЪЕКТНО-ОРИЕНТИРОВАННОМ ЯЗЫКЕ C#

Установка необходимого программного обеспечения

Поскольку будет создано приложение для доступа к БД через Интернет, то необходимо следующее программное обеспечение:

-среда разработки (выбрана MS Visual Studio 2015);

-СУБД (выбрана СУБД SQL Azure);

-язык, позволяющий создавать документы, в которых будет представлена необходимая информация из БД (выбран язык C#).

Нужно скачать среду разработки VS 2015. Затем при выборе необходимых языков программирования в ходе настройки среды нужно обязательно указать C# (если планируется работа с этим языком), однако возможен выбор другого языка.

Регистрация на MS Azure

После регистрации на Microsoft (https://support.microsoft.com/ru-ru/help/4026324/microsoft-account-sign-up-for-a-microsoft-account) нужно создать бесплатную учётную запись Azure (https://azure.microsoft.com/ru-ru/free/).

Создание БД

В случае успешной регистрации у Вас появляется доступ к панели мониторинга Azure. Выглядит она следующим образом (рис.23).

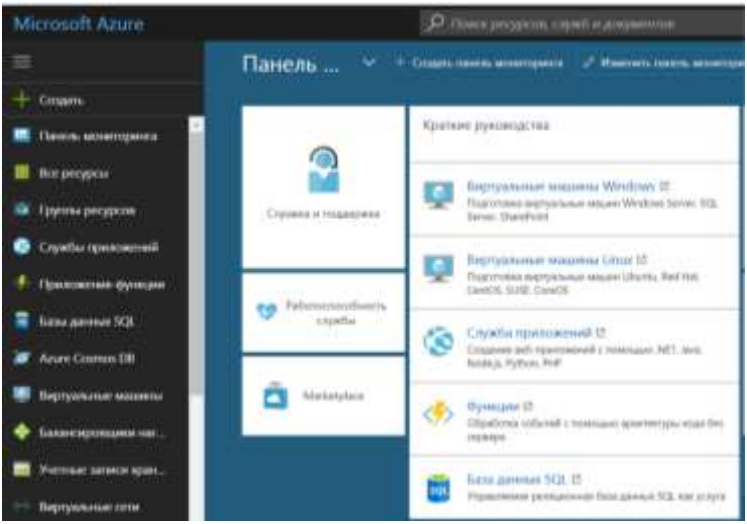


Рис23. Панель мониторинга MS Azure

Далее создается БД. Для ее создания необходимо выбрать на панели мониторинга раздел «Базы данных SQL» (рис. 24)

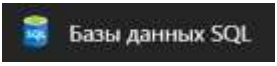


Рис. 24. Кнопка для добавления БД

Далее необходимо нажать на кнопку «Добавить», чтобы добавить новую БД. (рис. 25)

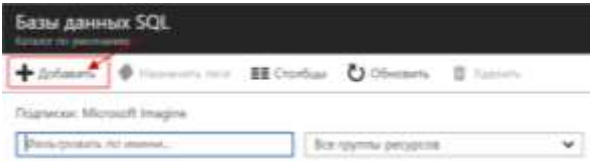


Рис.25. Кнопка для добавления новой БД

Далее необходимо заполнить появившиеся поля (рис.26). Необходимо также запомнить содержимое, вве-

денное в поля «Имя для входа администратора сервера» и «Пароль».

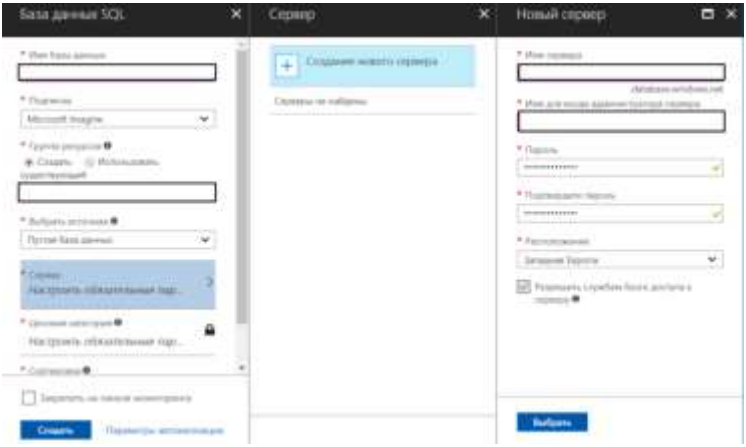


Рис. 26. Форма для создания новой БД

После заполнения появляется картинка (рис. 27).



Рис. 27. Оповещение о развертывании новой БД

Переходим к просмотру информации о созданной БД. Для дальнейшего подключения к БД из среды разработки будет необходима строка подключения к БД (чтобы увидеть её, необходимо нажать на обведённую на изображении ссылку) (рис. 28).

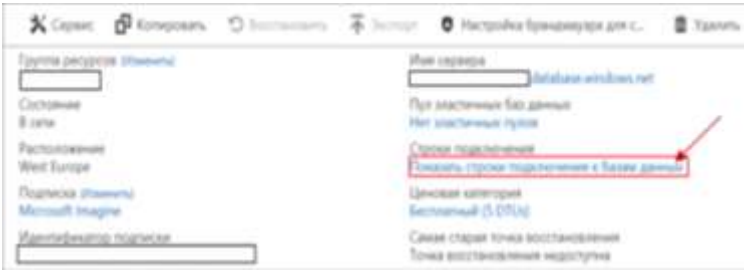


Рис. 28. Копирование строки подключения к БД

Переходим к созданию приложения для работы с БД. Запускаем среду программирования. Во вкладке «Файл» выбираем «Создать», затем «Проект…» (рис. 29).

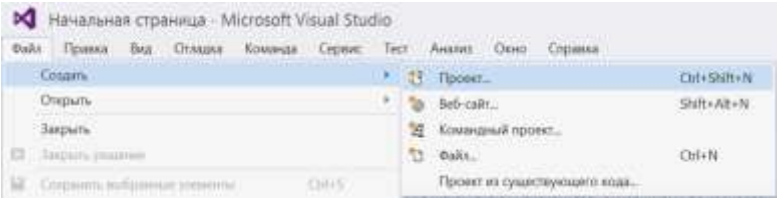


Рис. 29. Создание проекта для работы с БД

В появившемся окне выбираем параметры будущего приложения. Так как для работы был выбран язык объектно-ориентированный язык C#, создаётся приложение ASP.NET. Для его создания в левой части появившегося окна нужно выбрать Visual C#, а затем «Веб-приложение ASP.NET». Заполнив все поля в правой части появившегося окна, нужно нажать кнопку «Ок» (рис. 30).



Рис. 30. Создание приложения для работы с БД

В созданном приложении необходимо подключиться к БД, созданной ранее на MS Azure. Для этого в верхней панели во вкладке «Сервис» необходимо выбрать пункт «Подключиться к базе данных…» (рис. 31).

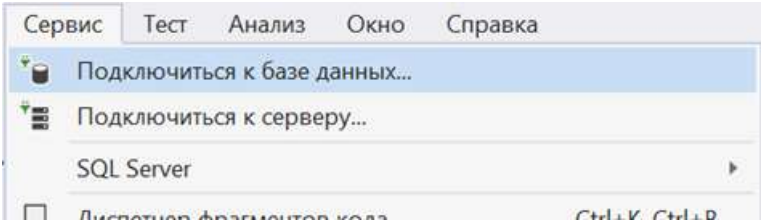


Рис. 31. Подключение приложения к БД

После этого откроется окно «Добавить подключение». Необходимо заполнить его поля. В поле «Источник данных» необходимо оставить Microsoft SQL Server (SqlClient). В поле «Имя сервера» необходимо ввести скопированную ранее строку подключения к БД на MS Azure. В поле «Имя пользователя» необходимо ввести содержимое поля «Имя для входа администратора сервера», кото-

рое было заполнено при создании БД на MS Azure, в поле «Пароль» ввести пароль, соответствующий тому паролю, который был введен при создании БД. Далее остаётся лишь выбрать имя созданной БД из предложенного списка, который содержит только одно имя, и нажать кнопку «Ок» (рис. 32).

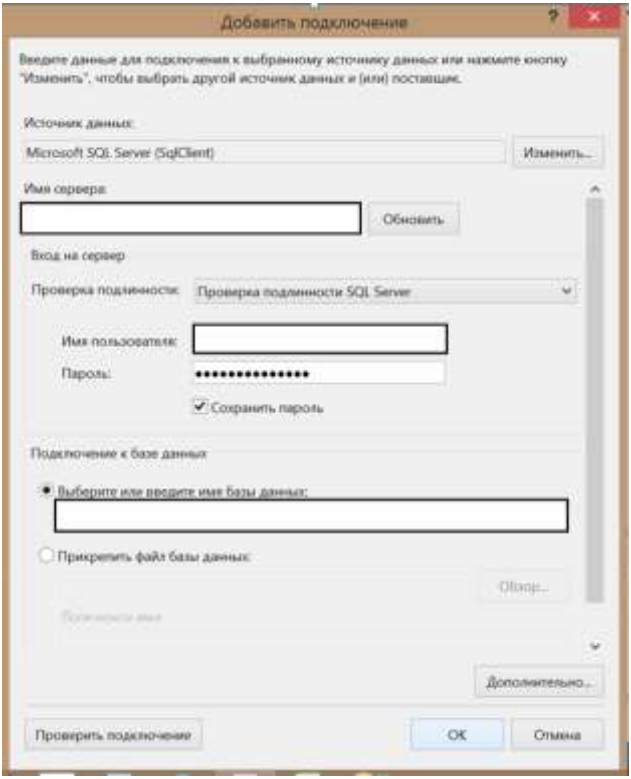


Рис. 32. Форма подключения приложения к БД

После этого будет создано подключение. Чтобы убедиться в том, что БД подключена, и начать работать с ней, необходимо во вкладке «Обозреватель серверов» в левой

части экрана посмотреть «Подключения данных» (рис. 33).

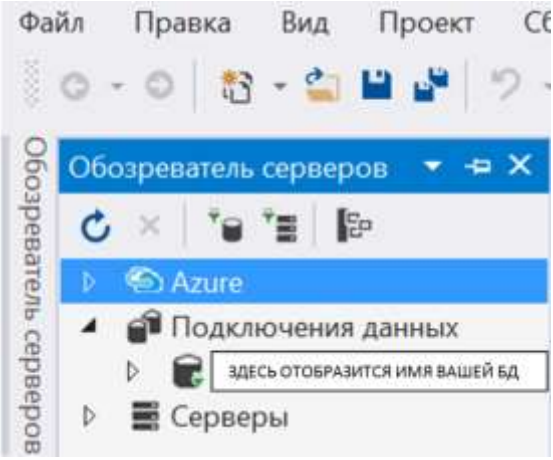


Рис. 33. Проверка подключения приложения к БД

Далее создаются таблицы БД, аналогичные таблицам БД, созданной с использованием СУБД Access в рамках выбранной предметной области в ходе лабораторной работы №2. (рис. 34).

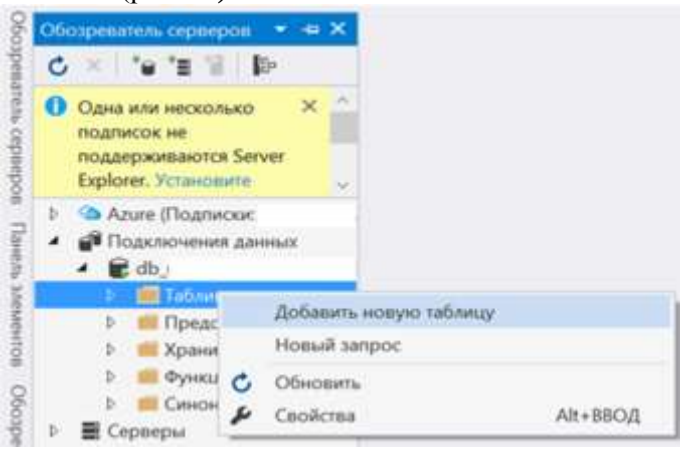


Рис. 34. Создание таблиц БД

Создавать структуру таблицы можно двумя способами. Первый способ подразумевает заполнение параметров атрибутов таблиц примерно таким же способом, как во второй лабораторной работе, то есть от пользователя требуется лишь ввести названия атрибутов, выбрать из списка их форматы, назначить первичный ключ, разрешить/запретить атрибуту быть пустым. Второй способ подразумевает создание таблиц при помощи написания SQL – запросов (рис. 35).

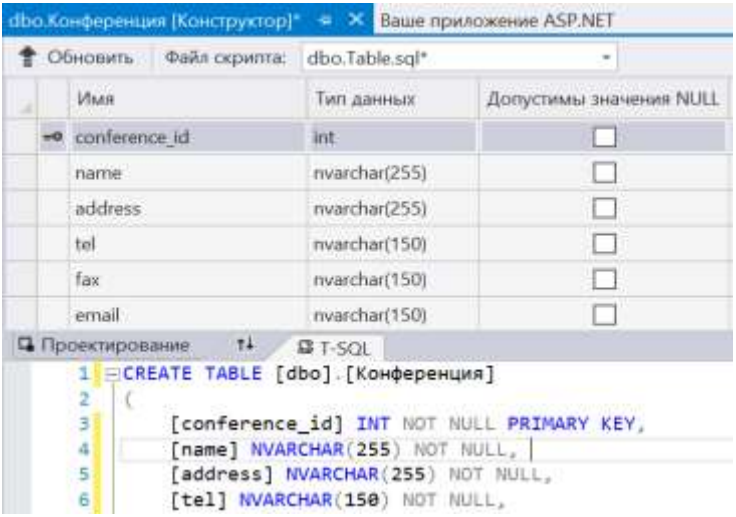


Рис. 35. Создание таблиц БД двумя способами

Для создания таблицы Конференция код, приведенный ниже, либо будет сгенерирован автоматически при заполнении верхних полей, либо может быть введен в нижнее поле, тогда верхние заполнятся автоматически.

```
CREATE TABLE [dbo].[Конференция]
(
        [id] INT NOT NULL PRIMARY KEY,
```

```
        [conference_id] INT NOT NULL,
    [name] NVARCHAR(255) NOT NULL,
    [address] NVARCHAR(255) NOT NULL,
    [tel] NVARCHAR(150) NOT NULL,
    [fax] NVARCHAR(150) NOT NULL,
    [email] NVARCHAR(150) NOT NULL,
    [date] DATE NOT NULL,
    [place] NVARCHAR(255) NOT NULL,
    [description] NVARCHAR(255) NOT NULL,
    [www] NVARCHAR(100) NOT NULL
)
```

По аналогии создаются все таблицы БД. Затем начинается работа над приложением для доступа к БД через Web.

Создание приложения для работы с БД

Особенностью приложений ASP.NET является наличие трёх необходимых компонентов: моделей, контроллеров и представлений. Для каждой таблицы БД необходимо создать модели. Рассмотрим создание модели на примере таблицы Конференция (WebApplication2 – название проекта). Models → Добавить → Класс. (рис. 36).

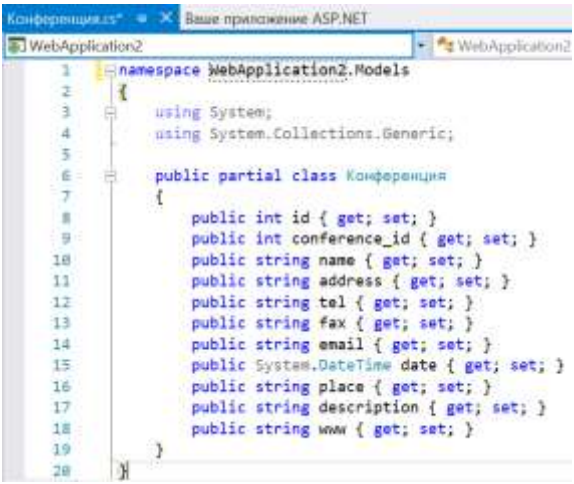


Рис. 36. Создание модели данных

Далее необходимо на стороне контроллера (в нашем случае контроллер Home) прописать следующий код:

```
using System.Data;
using System.Data.Entity;
using System.Web;
using System.Web.Mvc;
using WebApplication2.Models;

namespace WebApplication2.Controllers
{
    public class HomeController : Controller
    {
        private ConferenceEntities conference = new ConferenceEntities();
        public ActionResult Index()
        {
            return View(conference.Конференция.ToList());
        }
    }
}
```

В представлении Index.cstml прописываем код, при помощи которого будут выведены дынные из таблицы Конференция. В данном случае будет выведено содержимое полей name, tel, place.

```
@model
IEnumerable<WebApplication2.Models.Конференция>
@{
   ViewBag.Title = "Каталог конференций";
}
<table class="table">
   <tr>
      <th>
         Название
      </th>
      <th>
         Телефон для справок
      </th>
      <th>
         Город
      </th>
   </tr>
   @foreach (var item in Model)
   {
     <tr>
      <td>
       @Html.DisplayFor(modelItem => item.name)
      </td>
      <td>
       @Html.DisplayFor(modelItem => item.tel)
      </td>
      <td>
       @Html.DisplayFor(modelItem => item.place)
      </td>
     </tr>
   }
</table>
```

Занесем 2 записи в таблицу Конференция (рис. 37).



Рис. 37. Заполнение таблицы

Запустив приложение, получаем следующий результат (рис. 38).



Рис. 38. Результат работы приложения

Далее повторяем аналогичные шаги для всех таблиц и для всех необходимых запросов к БД. Также необходимо создать дизайн своего приложения по своему вкусу (всё, что связано с дизайном, необходимо прописывать в представлениях).

Публикация приложения на MS Azure

Для публикации готового приложения на MS Azure сначала необходимо создать пустое приложение на облаке. Для этого в панели мониторинга выбрать Службы приложений → Добавить. Далее необходимо скачать профиль публикации для созданного пустого приложения. Этот файл будет использован для публикации приложения, созданного в среде разработки (рис. 39).

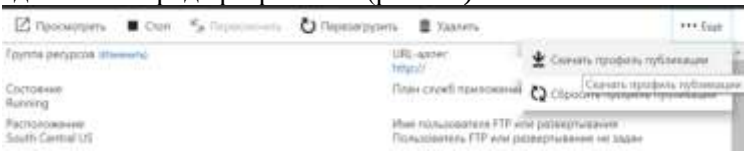


Рис. 39. Скачивание профиля публикации

Далее необходимо в среде разработки выбрать действие Опубликовать (рис. 40).



Рис. 40. Начало публикации

В появившемся окне нужно нажать кнопку Import и выбрать свежескачанный файл с профилем публикации. Далее нажать кнопку Опубликовать (рис. 41).

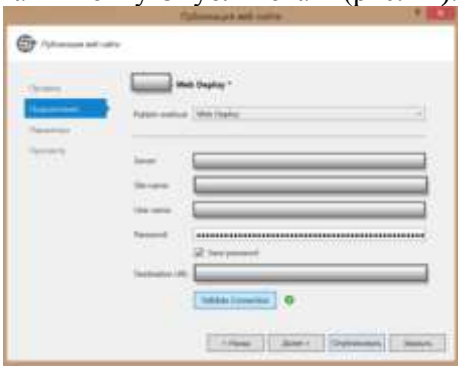


Рис. 41. Публикация

Приложение будет развёрнуто на Azure.

ЛАБОРАТОРНАЯ РАБОТА № 5

ЗНАКОМСТВО С ИНСТРУМЕНТАЛЬНЫМИ СРЕДСТВАМИ ДЛЯ СОЗДАНИЯ НОВЫХ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ НА ОСНОВЕ БАЗ ДАННЫХ

Цель работы – изучение возможностей case-средств для проектирования БД на основе выбранной предметной области.

Этапы выполнения работы

Выбирается одно из инструментальных средств (case-средств): Rational Rose, ERWIN.

Изучаются концепции и возможности выбранного средства. Студент должен приобрести навыки по инсталляции систем, разработке пилотного проекта на основе этих систем. При этом используются материалы лабораторных работ № 1–3 по проектированию и созданию БД. Для каждого средства выбирается свое направление пилотного проекта.

Производится разработка пилотного проекта. Для выбранного средства создается свой вариант проекта – разрабатываются диаграммы по базе данных с учетом опыта их проектирования и создания, накопленного при выполнении в лабораторных работах № 1–3 и 5.

Оптимизируется структура БД:

- проводится анализ существующей структуры БД, полноты заполнения полей и делается вывод о правильности использования форматов полей;
- вырабатывается критерий оптимизации структуры БД (полнота описания предметной области, качество данных, скорость доступа к данным, стандартизация структур данных) и на основе этих критериев проводиться модернизация схемы и таблиц БД;
- описываются проведенные изменения в схеме и таблицах БД.

Критериями оптимизации БД являются:

- повышение эффективности работы с БД (например, уменьшение числа связанных таблиц с четырех до трех, использование нескольких ключевых атрибутов, формализация представления атрибутов для повышения эффективности поиска данных);
- стандартизация структур данных (применение структур данных, которые можно использовать в любой стране, организации);
- повышение полноты и качества данных (расширение / уточнение состава атрибутов, характеризующих свойства объектов предметной области, их форматов и точности представления, организовать дополнительный контроль данных по взаимосвязям атрибутов).

Полнота описания предметной области достигается вводом новых сущностей и атрибутов. После проверки полноты заполнения существующих таблиц делается вывод о необходимости ввода того или иного атрибута. Скорость доступа оценивается по максимальному количеству связанных таблиц. Если число связанных таблиц больше 3, то перерабатывается схема БД. Если таблицы имеют огромное число строк, то предлагается создание индексов в таблицах. Для наиболее распространенных сущностей (страны, организации, персоны, проекты, библиографии и т. д.) предлагается использование международных стандартов на структуры (vCard, Mark, Dublin Core и др.) и классификаторы (стран, дат. Координат места, др.).

Студент должен приобрести следующие знания и навыки:

- иметь общее представление о case-средствах для проектирования БД;
- проектирования БД в выбранном case-средстве;
- автоматического создания (экспорта) БД из case-средства в различные БД.

Форма отчетности – текст отчета и демонстрация реализации. При защите лабораторной работы студент дол-

жен продемонстрировать возможности инструментальных средств, представить пилотный проект, созданный на основе одного из инструментальных средств, предварительно загрузить средство в компьютер в аудитории.

Объем. Отчет о пилотном проекте должен быть не менее пяти страниц с копиями экранов, описанием демонстрационных задач, проблем и трудностей, возникших при освоении инструментального средства.

Срок представления: в течение четвертого месяца занятий.

Литература

1. Вендров А.М. CASE - технологии. Современные методы и средства проектирования информационных систем. – М.: СИТ МГУ. 1996. - http://www.interface.ru/fset.asp?Url=/LOGWORKS/caset/glava1/case.htm
2. Сайт проектирования баз данных и хранилищ данных – http://www.erwin.ru.

Приложение 6

ПРОЕКТИРОВАНИЕ БД В ERWIN (НА ПРИМЕРЕ ПРОЕКТИРОВАНИЯ БД «КОНФЕРЕНЦИИ»)

Установка ERWin

- Использовался Computer Associates ERwin 4.0.
- После установки в корне диска создается папка CA_LIC.
- Копируем файл KEYGEN.exe в эту папку.
- Запускаем в ней файл RegIT.exe.
- Выбираем Register Later.
- Registration ID берем из кряка KEYGEN.exe.

Запуск, настройка, проектирование

- Запуск – Create a New Model – Logical/Physical.
- Target Database для начала выберем Access 2000.
-  - вид панели инструментов, выбираем второй инструмент; он используется для создания сущностей (таблиц БД) (рис.22).
- Задаем имя сущности – Конференция.
- Если возникли проблемы с отображением русских букв – заходим в меню Format – Default Fonts & Colors и везде, где необходимо, меняем Arial на Arial CYR, не забываем поставить Apply To на All Object (внизу).
- Кликаем на прямоугольник Сущности, получаем следующее (рис. 42).

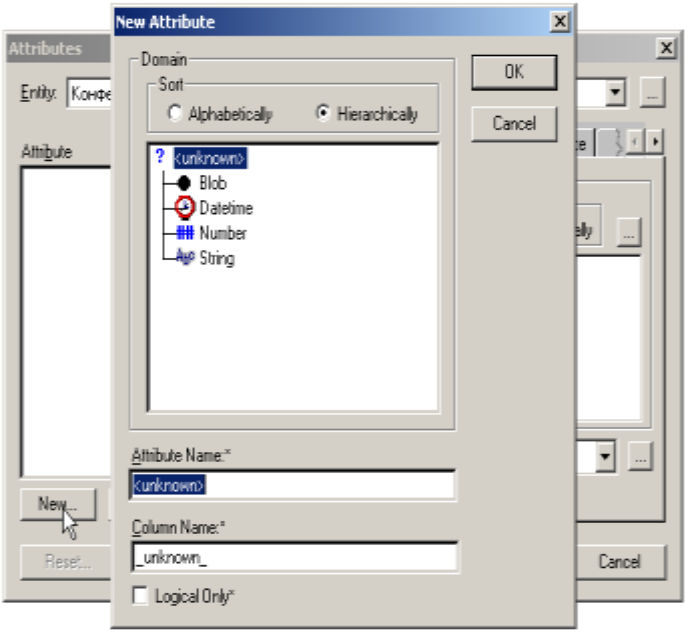


Рис. 42. Создание сущности «Конференция»

- Создаем новый атрибут – id конференции (Attribute Name – id конференции, Column Name – conference_id, тип – Number).
- Отмечаем его как первичный ключ – Primary Key.
- Во вкладке Datatype выбираем INTEGER.
- Один атрибут создан; так же создаем остальные.

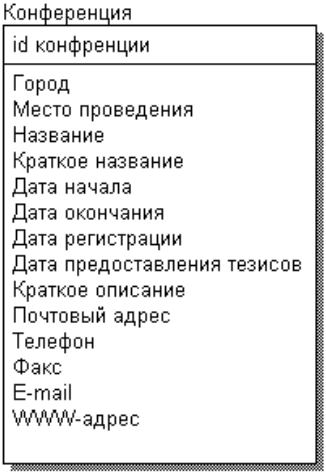
Таблица 10

Описание полей таблицы «Конференция»

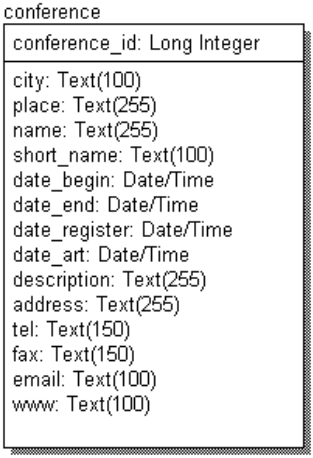
Attribute Name	**Column Name**	**Тип (логический)**	**Datatype (физический)**
Город	city	String	VARCHAR(100)
Место проведения	place	String	VARCHAR(255)

Attribute Name	**Column Name**	**Тип (логический)**	**Datatype (физический)**
Название	name	String	VARCHAR(255)
Краткое название	short_name	String	VARCHAR(100)
Дата начала	date_begin	Datetime	DATE
Дата окончания	date_end	Datetime	DATE
Дата регистрации	date_register	Datetime	DATE
Дата представления тезисов	date_art	Datetime	DATE
Краткое описание	description	String	VARCHAR(255)
Почтовый адрес	address	String	VARCHAR(255)
Телефон	tel	String	VARCHAR(150)
Факс	fax	String	VARCHAR(150)
E-mail	email	String	VARCHAR(100)
WWW-адрес	www	String	VARCHAR(100)

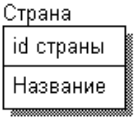
В результате должна получиться следующая картина.



При переводе в вид физического представления (поменять на панели управления Logical на Physical – при этом необходимо поменять имя сущности) получим следующую таблицу:



Аналогично создаем сущность Страна (далее речь идет о логическом представлении).



Объединяем сущности, выбираем инструмент , кликаем на сущность Страна, а затем на сущность Конференция, получаем следующую картину (рис.43).

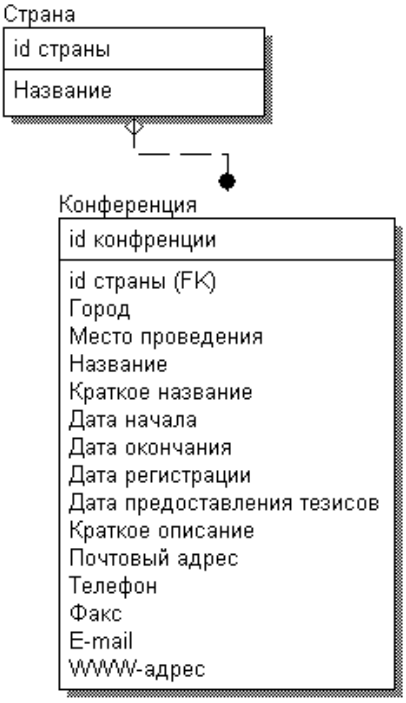


Рис. 43. Связь таблиц

Аналогично создаем все остальные сущности (таблицы) и объединяем их, получаем логическую и физическую модели. Логическая модель представлена на рис.44.

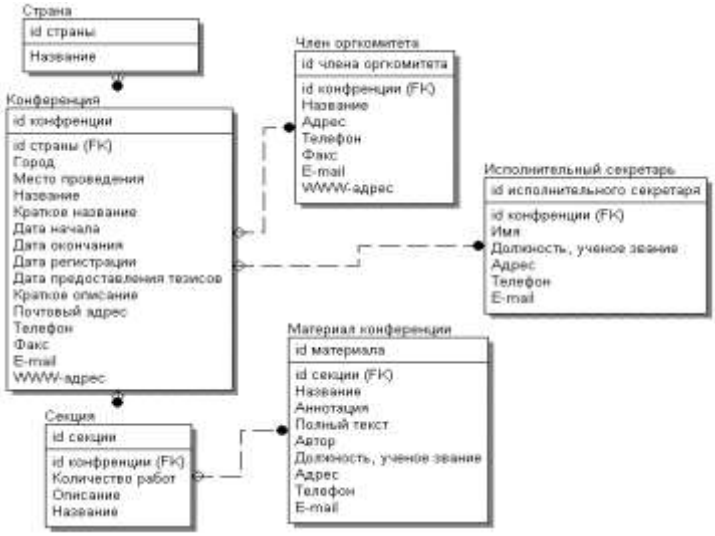


Рис.44. Логическая модель

Физическая модель дана на рис. 45.

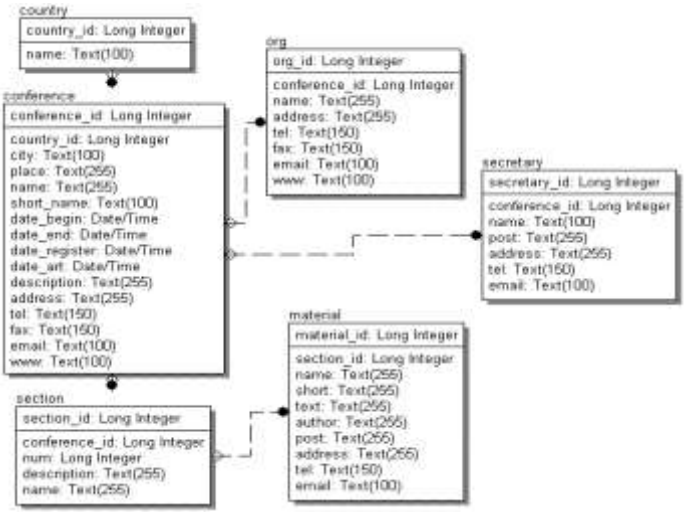


Рис.45. Физическая модель

Автоматическое создание (экспорт) таблиц в Access

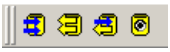
Сначала необходимо в Access создать пустую БД conference.mdb. В ERWin переходим к физическому представлению данных. Нам понадобится первый инструмент из этой панели  «Forward Engineer» – 'Прямое проектирование'. При нажатии на эту кнопку получаем картинку, изображенную на рис.46.



Рис.46. Процесс экспорта таблиц

Нажимаем Generate, получаем рис. 47.

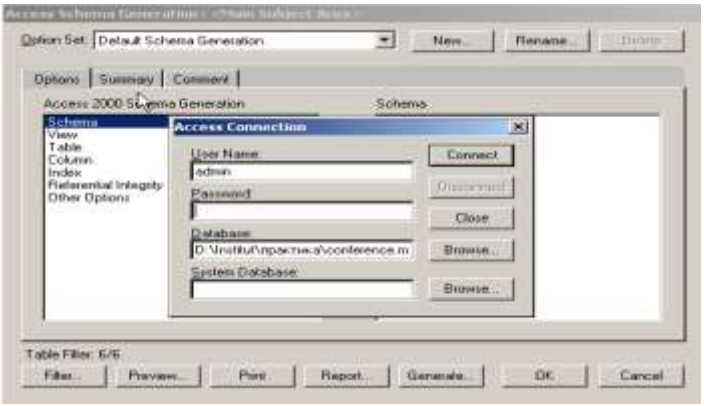


Рис.47. Процесс экспорта таблиц: ввод параметров БД

Выбираем параметры – User Name – admin; по нажатию кнопки Browse выбираем базу. Далее происходит автоматическая генерация (сама БД conference.mdb не должна быть при этом открыта).

Оптимизация БД

Добавляем новые сущности, уточняем состав атрибутов.

Приводим к схеме БД типа «Звезда» или используем многомерный подход при создании схемы БД.

Применяем стандарты структур данных и международные и национальные классификаторы.

ЛАБОРАТОРНАЯ РАБОТА 6
Подготовка документации на базу данных

Цель работы: обучение студентов основным принципам документирования на примере созданной в рамках лабораторных работ базы данных (БД) и приложений для нее.

В результате выполнения курсовой работы студент должен знать методологию документирования БД и программных приложений для нее, а также подготовить документацию на них в соответствии с ГОСТами серии 19, 24 и 34.

Форма отчетности. Выслать преподавателю отчет о выполненной курсовой работе по электронной почте. Сдать зачет по проделанной работе.

Объем. Семь документов (Общее описание технологии, Описание программы, Текст программ, Руководство администратора, Описание организации информационной базы, Инструкция оператору, Руководство пользователя).

Сроки представления: последний месяц семестра.

1 Состав и структура документации

1.1 Основные положения

1.1.1 Существующие стандарты (ГОСТы серии 19 и 34, руководящие документы) предусматривают разработку большого количества документов на программные ком-

плексы и другие компоненты информационных технологий (автоматизированных систем).

В настоящем документе приведено описание профиля документирования для стадий – рабочее проектирование и ввод в действие (ГОСТ серии 34). Предусматривается подготовка типового набора документов по каждой информационной технологии (рис. 48).

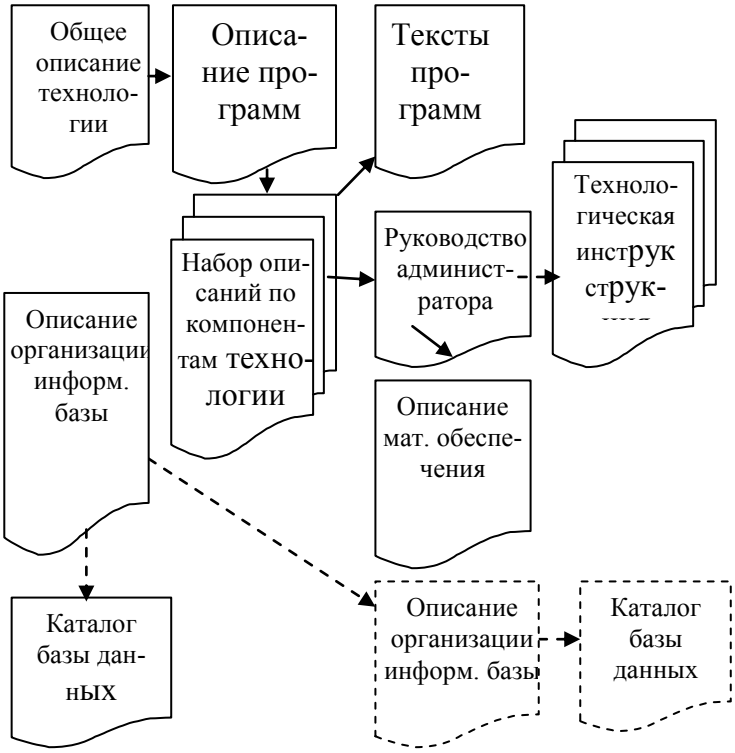


Рис. 48 – Состав документации технологий

Документ «Общее описание технологии» содержит общие сведения о построении технологии, ее компонен-

тах, методах и средствах функционирования, состав документации технологии.

Документ «Описание организации информационной базы» содержит сведения о принципах информационного обеспечения технологии, описание схем и связей информационных потоков, структур и содержания баз данных (наборов данных). Документ составляется для всей технологии или каждого компонента технологии в зависимости от ее специфики или обоснованного решения разработчиков.

Документ «Описание математического обеспечения» включает описание методов и алгоритмов решения прикладных задач программными комплексами компонентов технологии. Документ является обязательным.

Документ «Описание программ» содержит сведения о назначении, структуре и логике функционирования программного комплекса компонент технологии.

Документ «Тексты программ» содержит исходные тексты (коды) программ, реализующих компоненты технологии. Тексты программ должны быть подготовлены как часть документации (предоставляется на магнитном носителе ПЭВМ), если в технологии применяется оригинальное (созданное при разработке технологии) программное обеспечение.

Документ «Руководство администратора» включает описание действий пользователя по установке, настройке и использованию программного комплекса компоненты технологии.

Документ «Технологическая инструкция» специфицирует порядок и последовательность действий персонала по эксплуатации программного комплекса и является расширением (каждая инструкция может быть также подготовлена как отдельный документ) документа “Руководство администратора”. Технологическая инструкция составляется как элемент организационного обеспечения мно-

гофункциональных программных комплексов, используемых несколькими группами персонала.

Деление технологии на компоненты определяется спецификой технологии и определяется разработчиком. Это деление должно быть описано в документе «Общее описание технологии». Если разработчики не производят деление технологии на отдельные компоненты в силу различных обстоятельств и причин, то раздел 3 в документе «Общее описание технологии» не приводится и все описания особенностей технологии даются в разделе 2 документа «Общее описание технологии» (рис. 49).

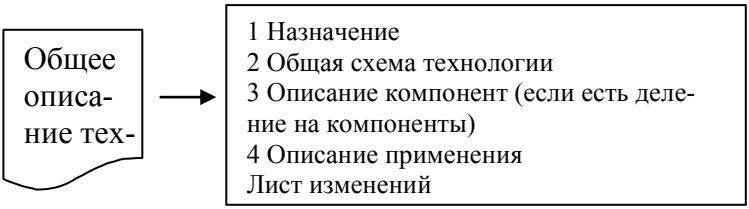


Рис. 49 – Содержание основного текста документа «Общее описание технологии»

1.2 Общее описание технологии

Документ «Общее описание технологии» основан на положениях ГОСТ 19.502-78.

Структура документа показана, см. рис.49.

1.2.1 В разделе “Назначение” должно быть описано:

1) вид деятельности, для автоматизации которой предназначена технология;

2) перечень объектов автоматизации, задействованных в технологии;

3) перечень функций, реализуемых технологией.

1.2.2 Раздел “Общая схема технологии” должен специфицировать отдельные элементы:

1) структуру технологии и назначение ее компонент;

2) сведения о технологии в целом и ее компонентах в части методического (математического), программного, информационного и организационного обеспечения технологии;

3) общую схему функционирования технологии.

1.2.3 В разделе “Описание компонент” (если технология разделяется на ряд компонент) должно быть указано:

1) назначение каждой компоненты

2) структуру каждой компоненты

3) описание функционирования компоненты

1.2.4 Раздел «Описание применения» должен включать:

1) описание технологического процесса обработки данных;

2) порядок и регламент выполнения работ

3) условия применения (состав и характеристики операционных систем, СУБД, ГИС и другого общего программного обеспечения, минимально необходимый набор вычислительных средств и иного оборудования для бесперебойной работы технологии, условия организационного характера)

1.2.5. В разделе “Документация технологии” необходимо поместить перечень разработанной и предъявляемой к испытаниям документации на технологию.

1.2.6 Лист изменений служит для внесения несущественных корректировок и изменений, которые не меняют сути документа или раздела документа. Общий вид листа изменений приведен в Приложении 9.

1.3 Описание компонент технологии

1.3.1 Описание программ

В основу документа «Описание программ» положен ГОСТ 19.402-78 «Описание программы». Структура документа приведена на рис. 50.

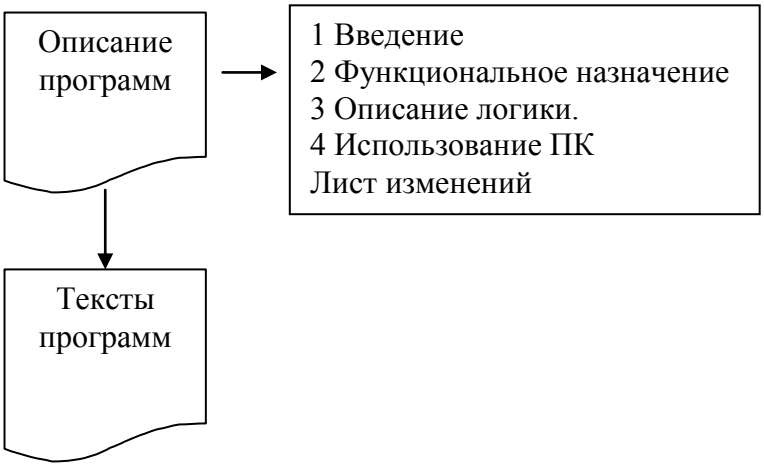


Рис. 50 – Содержание основного текста документа «Описание программ»

1.3.1.1 В разделе “Введение” должны быть приведены общие сведения о программных средствах, их текущем статусе и разработчиках.

1.3.1.2 В разделе “Функциональное назначение” должны быть приведены следующие сведения:

1) назначение программы или программного комплекса (программных средств - ПС)

2) общее описание функционирования программных средств

3) сведения об условиях применения с указанием требований относительно общего программного обеспечения (операционные системы, СУБД, ГИС) и комплекса технических средств - типа и мощности ПЭВМ и других устройств, которые необходимы для использования ПК

1.3.1.3 Раздел “Описание логики программы” должен давать:

1) описание структуры ПС и основных частей;

2) описание функций составных частей и связей между ними;

3) сведения о языке программирования;

4) описание входных и выходных данных (для каждой из составных частей);

5) описание логики составных частей (при необходимости следует составлять описание схем ПС).

В подразделе «Описание структуры» должны быть указаны:

- используемые методы;

- алгоритм работы ПС (дается привязка к документу “Описание методов(моделей)”;

- структура ПС с описанием функций составных частей и связи между ними;

- внешние и внутренние программные связи, включая описание системных данных (информации), используемых в служебных целях – конфигурационные файлы, таблицы управления технологическим процессом и др..

В подразделе «Входные данные» должны быть указаны:

- ссылка на раздел описания данных (документ «Описание организации информационной базы»)

- организация и предварительная подготовка входных данных

- описание и способ преобразования входных данных.

В подразделе «Выходные данные» должны быть указаны:

- ссылка на раздел описания данных (документ «Описание организации информационной базы»)

В документе допускается содержание разделов и подразделов иллюстрировать пояснительными примерами, таблицами, схемами, графиками.

1.3.1.4 В разделе «Использование программ» дается конкретизация действий при работе с ПК. В основу этого раздела положен ГОСТ 19.503-79.

В подразделе "Установка программы" должно быть приведено описание действий установке программного комплекса пользователем.

В подразделе "Настройка программы" должно быть приведено описание действий по настройке программы на условия конкретного применения (настройка на состав технических средств, выбор функций и др.). При необходимости приводят поясняющие примеры.

В подразделе "Проверка работоспособности" должно быть приведено описание способов проверки, позволяющих дать общее заключение о работоспособности программы (контрольные примеры, методы прогона, результаты).

В подразделе "Сообщения об ошибках" должны быть указаны тексты сообщений, выдаваемых в ходе выполнения настройки, проверки программы, а также в ходе выполнения программы, описание их содержания и действий, которые необходимо предпринять по этим сообщениям.

2.3.1.5 Тексты программ должны быть подготовлены как часть документации, если в технологии применяется оригинальное (созданное при разработке технологии) программное обеспечение. Тексты программ даются как приложение к основному документу “Описание программ” (возможно представление отдельной книгой). Вид документа определяется ГОСТ 19.401-78 и состоит из титула и сведений о программных средствах (информационная часть программного документа). Тексты программ должны быть представлены на магнитных носителях ПЭВМ (дискета, CD).

В информационной части документа должны быть приведены общие сведения о программных средствах по структуре: наименование программы, имя программы и ее назначение, а также описание программного файла с текстом программы на магнитных носителях:

- наименование и имя программы,
- ФИО автора,
- дата создания программы,
- номер версии,
- дата последней модификации; .
- имя программного файла.

Каждый программный файл должен сдержать заголовок (комментируемая часть программы), содержащий вышеуказанную информацию.

На техническом носителе все программные файлы должны быть собраны в одну директорию. В эту же директорию помещается файл READ.ME, в который необходимо занести наименование технологии, ведомства, организации, ФИО ответственного, контактный телефон и e-mail, именование директории, описание программных файлов.

При однокомпонентной технологии все тексты программ собираются внутри одной (главной) директории по структуре:

READ.ME

программный файл 1 (программа 1);

программный файл 2 (программа 2);

программный файл N (программа N).

Правила наименования главной директории:

Латинская аббревиатура организации

Подчеркивание

Фамилия ответственного, написанная латиницей

Подчеркивание

Удобочитаемая аббревиатура технологии.

Пример именования главной директории: RIHMI_IVANOV_CLIMAT

В самих текстах обязательными являются комментарии и пояснения по тексту.

1.3.2 Руководство администратора

Руководство администратора основано на ГОСТ 34.201-89 (рис. 51).



Рис. 51. Содержание основного текста документа «Руководство администратора»

1.3.2.1 Во введении указывают:

- область применения;
- краткое описание возможностей;
- уровень подготовки пользователя

1.3.2.2 В разделе "Назначение" указывают виды деятельности и функции, для которых предназначен данное ПС

1.3.2.3 В разделе "Подготовка к работе" указывают:

1) состав ПС

2) порядок загрузки ПС и последовательность работы

3) порядок проверки работоспособности.

1.3.2.4 В разделе "Описание операций" указывают:

1) описание выполняемых функций, задач, комплексов задач, процедур;

2) описание технологического процесса выполнения функций, комплексов задач (задач), процедур.

3) используемые технологические инструкции

1.3.2.5 Для каждой операции должно быть указано:

1) наименование;

2) условия, при соблюдении которых возможно выполнение операции;

3) подготовительные действия;

4) основные действия в требуемой последовательности;

В описании действий допускаются ссылки на файлы подсказок, размещенные на магнитных носителях.

1.3.2.6 В разделе "Аварийные ситуации" указывают:

1) действия в случае отказа программно-технических средств;

2) действия по восстановлению программ и/или перезапуску

3) действия в случаях обнаружении других аварийных ситуаций.

1.3.2.7 В разделе "Рекомендации по освоению" указывают рекомендации по освоению и эксплуатации, включая описание контрольного примера, правила его запуска и выполнения.

1.3.3 Технологические инструкции - Инструкция оператора

Технологические инструкции основаны на ГОСТ 24.209-80. Общая схема документа показана на рис. 52.

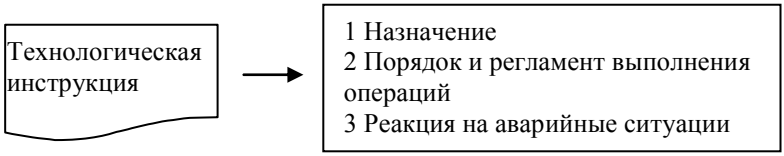


Рис. 52. Содержание основного текста документа «Технологическая инструкция»

1.3.3.1 В разделе "Назначение" указывают:

1) виды деятельности, выполняемые в рамках инструкции;

2) наименование используемого ПС и спецификации его применения в рамках инструкции со ссылками на документы “Описание программного комплекса” и “Руководство администратора”.

3) условия использования инструкции с указанием должностей и уровня квалификации персонала, на который распространяется данная инструкция;

1.3.3.2 В разделе "Порядок и регламент выполнения операций" указывают:

1) сведения о порядке и последовательности выполнения операций технологического процесса обработки данных, временной регламент их выполнения;

2) уточнения по применению ПС для выполнения операций посредством ссылок на разделы документа “Руководство администратора” или посредством описания применения ПС в рамках действий по инструкции;

3) порядок регистрации выполненной операции (операций);

4) состав и содержание документов (отчетов, справок) в твердой копии или с применением ПК или других программных средств, используемых для контроля выполнения инструкции.

1.3.3.3 В разделе "Реакция на аварийные ситуации" указывают действия исполнителя операций в случае отказа программно-технических средств или обнаружении других аварийных ситуаций, включая реквизиты администратора (разработчика) технологии для обращения за помощью.

1.3.4 Описание математического обеспечения

Описание математического обеспечения основано на документе РД 19.101-77 (рис. 53).

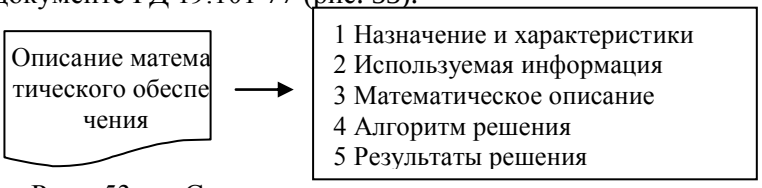


Рис. 53 – Содержание основного текста документа «Описание математического обеспечения»

В случае, если в технологии используется нескольких методов (моделей) в разделе “Введение” дается привязка методов (моделей) к соответствующим компонентам технологии и основной текст документа дается для каждого метода (модели)

1.3.4.1 В разделе "Назначение и характеристика" должно быть приведено:

1) назначение метода (модели), наименование компоненты технологии (программного комплекса);

2) краткие сведения о процессе (объекте) морской среды или морской деятельности, в связи с которым используют метод (модель);

3) ограничения на возможность и условия применения метода (модели) и характеристики качества решения (точность, время решения и т.д.);

4) общие требования к входным и выходным данным (форматам, кодам и т. д.), обеспечивающие информационную совместимость решаемых задач в системе;

5) сведения о сертификации метода (модели) и ссылки на опубликованные нормативные документы (издания) с описанием метода (модели).

1.3.4.2 В разделе "Используемая информация" должен быть приведен перечень массивов информации, используемых при реализации метода (модели), в том числе:

1) массивы входной информации;

2) массивы информации, полученные в результате работы других программных комплексов технологии и сохраняемые для реализации данного метода (модели).

По каждому массиву приводят наименование и краткую характеристику со ссылкой на раздел (подразделы) “Описание массивов информации” документа “Описание информационной базы”.

1.3.4.3 В разделе "Результаты решения" должен быть приведен перечень массивов информации, формируемых в

результате реализации метода (модели). По каждому массиву должно быть дано наименование и краткая характеристика со ссылкой на раздел (подразделы) “Описание массивов информации” документа “Описание информационной базы”.

1.3.4.4 В разделе "Математическое описание" приводят:

1) математическую модель или экономико-математическое описание процесса (объекта);

2) перечень принятых допущений и оценки соответствия принятой модели реальному процессу (объекту);

3) сведения о результатах научно-исследовательских работ, если они использованы для разработки алгоритма.

1.3.4.5 В разделе "Алгоритм решения" следует приводить:

1) описание логики алгоритма реализации метода (модели) и способа формирования результатов решения с указанием последовательности этапов счета, расчетных и (или) логических формул, используемых в алгоритме;

2) указания о точности вычисления (при необходимости);

3) соотношения, необходимые для контроля достоверности вычислений;

4) описание связей между частями и операциями алгоритма;

Алгоритмом должны быть предусмотрены все ситуации, которые могут возникнуть в процессе решения задачи. При изложении алгоритма следует использовать условные обозначения реквизитов со ссылкой на соответствующие массивы информации. В расчетных соотношениях (формулах) должны быть использованы обозначения реквизитов, приведенные при описании их состава в других разделах документа.

Алгоритм представляют одним из следующих способов:

1) графический (в виде схемы);

2) табличный;

3) текстовой;

4) смешанный (графический или табличный с текстовой частью).

Способ представления алгоритма выбирает разработчик, исходя из сущности описываемого алгоритма и возможности формализации его описания.

Алгоритм в виде схемы выполняют по правилам, установленным ГОСТ 19.005. Алгоритм в виде таблиц выполняют по правилам, установленным ГОСТ 2.105. Алгоритм в виде текстового описания выполняют по правилам, установленным ГОСТ 24.301.

1.3.5 Описание организации информационной базы

Документ «Описание организации информационной базы» основан на требованиях ГОСТ 34.201-89 и должен содержать разделы (рис. 54).

1.3.5.1 В разделе «Общее описание информационного обеспечения» должны быть указаны:

1) принципы организации информационного обеспечения, принятые методы организации информационной базы, классификации и кодирования;

2) наименование и характеристики массивов входной информации с указанием источников и носителей входных данных, оценки интенсивности и объема потоков данных;

3) наименование и назначение баз данных (наборов данных) технологии по компонентам.

1.3.5.2 В разделе «Описание системы классификации и кодирования” должно быть приведено:

1) описание средств классификации и кодирования в технологии со ссылками на первоисточники или указанием о вновь разработанных средствах;

2) описание таблиц классификаций, кодификаций и кодировок с необходимыми ссылками на первоисточники.

1.3.5.3 В разделе «Описание организации информационной базы» должно быть приведено:

1) описание модели (методов построения) информационной базы на уровне баз данных (наборов данных) и функций компонент технологии с указанием характера взаимосвязей: между базами (наборами) данных; между массивами баз данных (наборов данных) и массивами входной информации.

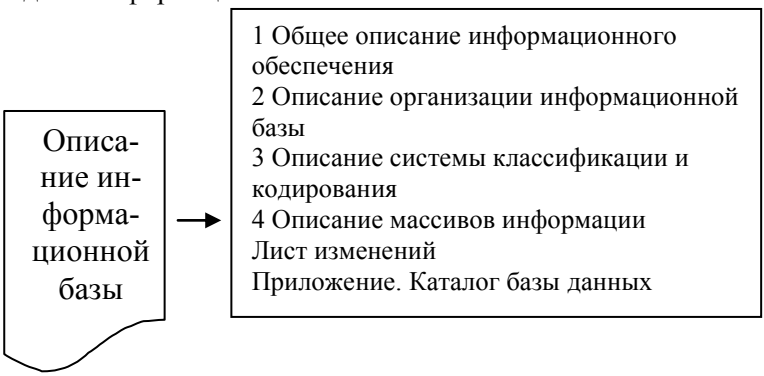


Рис. 54 – Содержание основного текста документа «Описание информационной базы»

2) описание структуры информационной базы – для каждой базы (набора) системных и тематических данных приводится описание логической структуры массивов информации - структуры таблиц БД (формат файлов данных) с указанием обозначения, типов данных и диапазона изменения элементов таблицы (записи).

1.3.5.4. В разделе "Организация ведения информационной базы" должны быть приведены описания последовательности процедур при создании и обслуживании баз данных (наборов данных) с указанием, при необходимо-

сти, процедур и средств защиты базы от разрушения и несанкционированного доступа.

1.3.5.5 В разделе «Описание массивов информации» должно быть приведено описание баз (наборов) данных информационной базы, имеющих непосредственное отношение к описываемому компоненту. В приложении к документу должен быть приведен каталог данных на момент предъявления технологии к испытаниям. Структуры описания массивов информации и каталога базы данных приведены в приложении 10.

1.3.5.6 Лист изменений служит для внесения несущественных корректировок и изменений, которые не меняют сути документа или раздела документа. Общий вид листа изменений дан в Приложении 9.

1.3.6 Руководство пользователя

Руководство пользователя включает следующие разделы:

1. Введение

1.1. Область применения

1.2. Краткое описание возможностей

1.3. Уровень подготовки

2. Назначение и условия применения

3. Подготовка к работе

3.1. Состав и содержание дистрибутивного носителя данных

3.2. Порядок загрузки данных и программ

3.3. Порядок проверки работоспособности

4. Описание операций

4.1. Описание операций пользователя

4.1.1 Выполняемые функции и задачи

4.1.2. Описание операций технологического процесса обработки данных, необходимых для выполнения задач

4.2. Описание выполняемых функций, задач, комплексов задач, процедур

4.3. Завершение сеанса работы
5. Аварийные ситуации

Руководство должно ответить на следующие вопросы:

- а что это;
- а что им можно делать;
- а что им нельзя делать (у особо одаренных);
- а что надо, чтобы оно работало;
- а что там у него внутри (у детей, склонных к глубокому анализу);
- а как его настроить, чтобы работало так, как я хочу;
- а как его проверить, работает оно или не работает;
- а что и где надо нажимать;
- а что оно еще может делать;
- а что оно говорит, если что-то не так нажимаешь?

2 ТРЕБОВАНИЯ ПО ОФОРМЛЕНИЮ ДОКУМЕНТАЦИИ

За основу оформления всех документов взят действующий ГОСТ 7.32-91 (ИСО 5966-82) по оформлению НТО по НИР в редакции настоящего документа.

Создаваемая документация выпускается на листах белой бумаги формата А4 (210х297).

Поля документов: верх, низ, справа – 10 мм, слева – 20 мм.

Используемый шрифт - Times New Roman, 12 п.

Межсимвольное расстояние – нормальное, межстрочное расстояние – 1,5 интервала.

Красная строка (абзац) устанавливается на 12,5 мм от левой границы.

Нумерация листов – сквозная по всему документу внизу листа по центру. На первом (титульном) листе номер не проставляется.

Иллюстративный материал следует располагать в документе непосредственно после текста, в котором она упоминается впервые, или на следующей странице Иллюстрации должны иметь название, которое помещают под иллюстрацией по центру. Иллюстрация обозначается словами «Рисунок NN – (NN – порядковый номер рисунка)», которые помещают перед наименованием. Иллюстрации следует нумеровать арабскими цифрами порядковой нумерацией в пределах всего документа. На все рисунки должны быть ссылки в документах.

Цифровой материал должен оформляться в виде таблиц. Таблицу следует располагать в документе непосредственно после текста, в котором она упоминается впервые, или на следующей странице. На все таблицы должны быть ссылки в документе. Таблицы следует нумеровать арабскими цифрами порядковой нумерацией в пределах всего документа. Номер следует размещать в левом верхнем углу над таблицей после слова «Таблица». Наименование таблицы помещается непосредственно после ее номера. Оформление таблиц - по ГОСТ 2.105.

Перечисления, при необходимости, могут быть приведены внутри пунктов или подпунктов. Перечисления следует нумеровать порядковой нумерацией арабскими цифрами со скобкой, например, 1), 2), 3) и т.д., и печатать строчными буквами с абзацного отступа. В пределах одного пункта или подпункта не допускается более одной группы перечислений.

Примечания следует помещать в документе при необходимости пояснения содержания текста, таблицы или иллюстрации. Примечания размещают непосредственно после пункта, подпункта, таблицы, иллюстрации, к которым они относятся, и печатают с прописной буквы с абзацного отступа. Слово «Примечание» следует печатать с прописной буквы с абзацного отступа вразрядку и не подчеркивать. Одно примечание не нумеруют. Несколько

примечаний следует нумеровать порядковой нумерацией арабскими цифрами.

Сноски, при необходимости, могут быть приведены внутри документа. Сноски следует нумеровать порядковой нумерацией арабскими цифрами со скобкой, например, 1), 2), 3) и т.д.

Уравнения и формулы следует выделять из текста в отдельную строку. Выше и ниже каждой формулы или уравнения должно быть оставлено не менее одной свободной строки. Если уравнение не умещается в одну строку, оно должно быть перенесено после знака равенства (=) или после знаков плюс (+), минус (-), умножения (х), деления (:), или других математических знаков.

Формулы в документе следует нумеровать порядковой нумерацией в пределах всего документа арабскими цифрами в круглых скобках в крайнем правом положении на строке.

Если дана только одна формула или уравнение, их не нумеруют.

Пояснение значений символов и числовых коэффициентов следует приводить непосредственно под формулой в той же последовательности, в которой они даны в формуле. Значение каждого символа и числового коэффициента следует давать с новой строки. Первую строку пояснения начинают со слова «где» без двоеточия.

Ссылки на источники следует указывать порядковым номером по списку источников, выделенным прямыми квадратными скобками. Наряду с общим списком допускается приводить ссылки на источники в подстрочном примечании. Оформление ссылок - по ГОСТ 7.1.

Ссылки на разделы, подразделы, пункты, подпункты, иллюстрации, таблицы, формулы, уравнения, перечисления, приложения следует указывать их порядковым номером, например, «... в разд. 4», «... по п.1.3.4», «... в под-

пункте 1.3.4.1, перечисление 3», «... по формуле (3)», «... в уравнении (2)», «... на рис.8», «... в приложении 9».

Документация структурно должна содержать:

- титульный лист;
- содержание;
- перечень сокращений, условных обозначений, символов, единиц и терминов;
- введение;
- основная часть;
- заключение;
- список использованных источников (если есть);
- приложения (если необходимы).

2.1 Титульный лист

Титульный лист является первой страницей документа и служит источником информации, необходимой для обработки и поиска документа.

На титульном листе (образец показан на рис. 55) приводят следующие сведения:

- наименование министерства и организации - исполнителя;
- номер протокола проведенных испытаний и дата утверждения результатов испытаний;
- наименование документа;
- место и дата составления документа.

Наименование министерства или ведомства
Наименование организации
Наименование технологии
Наименование документа
Версия n1.n2.n3
Место издания год издания

Рис. 55. Образец оформления титульного листа

Примеры заполнения титульных листов для различных вариантов описаний технологий приведены в Приложении 10 к настоящему документу.

2.2 Содержание

Содержание включает введение, наименование всех разделов, подразделов, пунктов (если они имеют наименование) и заключение с указанием номеров страниц, с которых начинаются эти элементы документа.

2.3 Перечень сокращений, условных обозначений, символов, единиц и терминов

Принятые в документе малораспространенные сокращения, условные обозначения, символы, единицы и специфические термины должны быть представлены в виде отдельного списка.

Если сокращения, условные обозначения, символы, единицы и термины повторяются в документе менее трех раз, отдельный список не составляют, а расшифровку дают непосредственно в тексте документа при первом упоминании.

2.4 Введение

Введение должно содержать основание и обоснование необходимости создания документа, сведения о связи с другими документами, определение целей и задач документа, его место в общем комплекте документации на проект.

При необходимости во введении требуется помещать перечень использованных нормативов и законов: использованные в документе нормативы (ГОСТ, ОСТ, ВСН, СНИП и др.) и законы должны быть представлены в виде отдельного списка. Причем, их именование должно проводиться в полном объеме. Не допускается указать только ГОСТ 34, а требуется дать точно его номер и название и т.п.

2.5 Основная часть

Основная часть документа должна содержать понятное и однозначно трактуемое описание сути документа.

В зависимости от особенностей документа основная часть излагаются в виде текста, таблиц, сочетания иллюстраций и таблиц или сочетания текста, иллюстраций и таблиц.

Основную часть документа следует делить на разделы и пункты. Разделы основной части могут делиться на пункты или на подразделы и пункты. Пункты, при необходимости, могут делиться на подпункты. Каждый пункт должен содержать законченную информацию.

Заголовки структурных элементов документа и разделов основной части следует располагать в середине строки без точки в конце и печатать прописными буквами, не подчеркивая.

Заголовки подразделов и пунктов следует начинать с абзацного отступа и печатать с прописной буквы вразрядку, не подчеркивая, без точки в конце.

Если заголовок включает несколько предложений, их разделяют точками. Переносы слов в заголовках не допускаются.

Расстояние между заголовками структурных элементов документа и разделов основной части и текстом должно быть не менее 3 - 4 интервалов.

Пункты и подпункты основной части следует начинать печатать с абзацного отступа.

2.6 Заключение

Заключение должно содержать краткие выводы и оценку полноты решений, рекомендации по конкретному использованию.

2.7 Список использованных источников

Список должен содержать сведения об источниках, использованных при составлении документа. Сведения об источниках приводятся в соответствии с требованиями ГОСТ 7.1.

2.8 Приложения

В приложения рекомендуется включать объемные и вспомогательные материалы, например,

- математические формулы;
- таблицы вспомогательных цифровых данных;
- протоколы испытаний;
- заключение экспертиз;
- иллюстрации вспомогательного характера;
- распечатки ЭВМ с результатами обработки;
- протоколы рассмотрения работ;
- результаты экспертиз
- акты внедрения и др.

Приложения следует оформлять как продолжение основной части документа на его последующих страницах или в виде отдельной книги, располагая приложения в порядке появления на них ссылок в тексте документа.

Каждое приложение должно начинаться с новой страницы и иметь содержательный заголовок, напечатанный прописными буквами. В правом верхнем углу над заголовком прописными буквами должно быть напечатано слово «Приложение».

Если приложений в документе более одного, их следует нумеровать арабскими цифрами порядковой нумерацией.

При оформлении приложений отдельной книгой на титульном листе под номером книги следует писать слово «Приложения».

При необходимости, текст приложений может быть разделен на разделы, подразделы, пункты, которые следует нумеровать в пределах каждого приложения.

Имеющиеся в тексте приложения иллюстрации, таблицы, формулы и уравнения следует нумеровать так же, как и в основной части документа.

Приложение 9

ЛИСТ ИЗМЕНЕНИЙ
(общий вид)

№ п	Дата внесения изменений	Номер раздела, пункта	Содержание внесенного изменения	Версия документа после внесения изменений	Должность, Ф.И.О.
Пример					
1	30.09.2006	Раздел 2, п.2.2	Замена номера ГОСТ на 19.502-78	1.0.1	Зав. лабораторией организации, И.И. Петров

Приложение 10

СТРУКТУРА ОПИСАНИЯ [НАИМЕНОВАНИЕ БАЗЫ ДАННЫХ]

Название атрибута	**Комментарий**
1. Общие сведения	
Полное наименование	Полное наименование базы данных (набора данных)
Краткое название	Краткое название массива данные
Наименование и аббревиатура организации: создатель массива с указанием ведомства	Наименование и аббревиатура организации, создавшей массив информации с указанием ведомства
Организационно-	Государственный, частный

Название атрибута	**Комментарий**
правовая форма	
2. Технические характеристики	
Система хранения	ОС, тип СУБД, ГИС, ППП системы файлов данных
Формат данных	Краткая характеристика формата (структуры таблиц) со ссылкой на соответствующий подраздел “Описание организации информационной базы”
Форма представления	Алфавитно-цифровой массив, текстово-графические документы, пространственные (векторные) данные
Режим обновления массива	Ежедневно, ежемесячно, ежегодно, по мере поступления новой информации, не пополняется
Объем массива (в логических единицах)	Тип логической единицы (пунктов, штук и др.) и количество логических единиц
Объем массива (Мбайт):	Объем данных в физических единицах (Мбайт)
2. Характеристики содержания	
Параметры	Перечень параметров, входящих в описываемый массив информации (температура воды, воздуха, соленость, количество населения, объем перевозок через порт)
Временная характеристика	Характеристика временного покрытия - дата начала и окончания наблюдений и др.
Частота получения новой информации	Свободный текст
Географическая характеристика	Названия географических районов, для которых создан массив

Название атрибута	Комментарий
	данных (Балтийское море, Баренцево Море, Финский залив, или названий городов, портов, др.)
Пространственный масштаб	Пространственный масштаб представления данных (фиксированная точка, 1, 2, 5 градусный квадрат, сетка 1*1 минуту, др.)
Временной масштаб	Временной масштаб представления данных (срочные, суточные, многолетние месячные данные и др.)
Контроль данных	Краткая характеристика контроля данных
Уровень обработанности данных	Данные наблюдений, обобщенные данные методами статистики, анализы или аналитическая информация, прогностическая информация
3. Пользовательские характеристики	
Удаленный доступ	Если – да, то указать URL адрес
Услуги	Разовые обращения, фрагмент массива, поставка массива, другое
Ограничения на доступ	Ограничения на доступ (свободный доступ, доступ по паролю)
4. Контактные данные	
Адрес организации	Полный адрес, телефон, факс, электронная почта, сайт
Администратор массива	ФИО, телефон, факс, электронная почта
Дата заполнения:	Дата заполнения описания в виде GGGG-MM-DD
Дата последней редакции:	Дата последней редакции в виде GGGG-MM-DD

СТРУКТУРА КАТАЛОГА ДАННЫХ
[НАИМЕНОВАНИЕ БАЗЫ ДАННЫХ]

Название атрибута	Комментарий
Идентификатор логической единицы массива информации	Указывается принятый в технологии идентификатор логической единицы в соответствии с определением в описании массива информации
Краткое название	Краткое название логической единицы информации
Параметры	Перечень параметров, входящих в описываемый массив информации (температура воды, воздуха, соленость, количество населения, объем перевозок через порт)
Временная характеристика	Временная характеристика логической единицы данных - дата начала и окончания наблюдений, др.
Географическая характеристика	Названия географических районов (Балтийское море, Баренцево Море, Финский залив или названия городов, портов, др.)