НЕКОММЕРЧЕСКОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
УЧЕБНО-НАУЧНО-ПРОИЗВОДСТВЕННЫЙ КОМПЛЕКС
«МЕЖДУНАРОДНЫЙ УНИВЕРСИТЕТ КЫРГЫЗСТАНА»

«СОГЛАСОВАНО»
Проректор по учебно-административной работе НОУ УНПК «МУК»,
к.ю.н., Карабалаева С.Б.
_____
«18» 11 2019 г.

«УТВЕРЖДЕНО»
Ректор НОУ УНПК «МУК»,
к.т.н., доцент Савченко Е.Ю.
_____
«19» 11 2019 г.

УЧЕБНО-МЕТОДИЧЕСКИЙ КОМПЛЕКС

Название дисциплины: Базы данных

Направление: 710100 «Информатика и вычислительная техника»

Профиль: Компьютерные информационные системы для бизнеса

Квалификация выпускника: Бакалавр

Форма обучения: Очная

Составитель (и): преп. Маджинов А.Р._____

График проведения модулей
5-семестр

неделя	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
лекц. зан.	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
лаб. зан.	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4

«РАССМОТРЕНО»
На заседании кафедры «КИСиУ»
НОУ УНПК «МУК»
Протокол № 2
от «14» 10 2019 г.
Зав. кафедрой
д.т.н., проф. Миркин Е.Л.
_____

«ОДОБРЕНО»
На заседании Учебно-методического объединения НОУ УНПК «МУК»
Протокол № 2
от «15» 11 2019 г.
Председатель Учебно-методического объединения
Матвеева Т.В. _____

Директор Научной библиотеки
НОУ УНПК «МУК»
Асанова Ж.Ш._____

Бишкек 2019 г.

Оглавление

АННОТАЦИЯ....4

УЧЕБНО-МЕТОДИЧЕСКИЙ КОМПЛЕКС ДИСЦИПЛИНЫ (МОДУЛЕЙ)....5

1. Пояснительная записка....5

1.1. Миссия и стратегия....5

1.2. Цель и задачи дисциплины (модулей)....5

1.3. Формируемые компетенции, а также перечень планируемых результатов обучения по дисциплине (модулю) (знания, умения владения), сформулированные в компетентностном формате....5

1.4. Место дисциплины (модулей) в структуре ООП ВПО....5

2. Структура дисциплины (модулей)....6

4. Конспект лекций....9

5. Информационные и образовательные технологии....10

6. Фонд оценочных средств для текущего, рубежного и итогового контролей по итогам освоению дисциплины (модулей)....11

6.1. Перечень компетенций с указанием этапов их формирования в процессе освоения дисциплины....12

6.2. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности....12

6.3. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания....13

6.4. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности....14

7. Учебно-методическое и информационное обеспечение дисциплины....15

7.1.Список источников и литературы....15

7.2. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимый для освоения дисциплины (модулей).....16

8. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся.....17

8.2. Методические указания для обучающихся по освоению дисциплины (модулей) 20

8.3. Методические рекомендации по подготовке письменных работ 25

8.4. Иные материалы 25

9. Материально-техническое обеспечение дисциплины 25

10. Глоссарий 25

11. Приложения. 32

АННОТАЦИЯ

Одной из областей компьютерных технологий, бурно развивающихся в настоящее время, является автоматизация обработки информации на основе программно-вычислительных комплексов, называемых информационными системами. Информационные системы оперируют большими объемами информации, хранящимися во внешней памяти. В настоящее время, подавляющее большинство информационных систем строится на основе реляционных баз данных.

Одним из наиболее популярных инструментов для создания и управления базами данных является Microsoft Access.

Изучая данный продукт, студенты смогут не только ознакомиться с возможностями Access, но и выработать свой подход к решению различных задач. Кроме того, они научаться разрабатывать собственные приложения баз данных.

УЧЕБНО-МЕТОДИЧЕСКИЙ КОМПЛЕКС ДИСЦИПЛИНЫ (МОДУЛЕЙ)

1. Пояснительная записка

1.1. Миссия и стратегия

Миссия НОУ УНПК "МУК" – подготовка международно - признанных, свободно мыслящих специалистов, открытых для перемен и способных трансформировать знания в ценности на благо развития общества. Видение НОУ УНПК «МУК»- создание динамичного и креативного университета с инновационными научно-образовательными программами и с современной инфраструктурой, способствующие достижению академических и профессиональных целей.

Стратегии развития - модернизация образовательной деятельности университета – совершенствование образовательного процесса в соответствии с требованиями Болонского процесса.

1.2. Цель и задачи дисциплины (модулей)

Цель дисциплины «Базы данных» позволяет освоить будущим специалистам инженерам теоретические знания и сформировать у них практические навыки в применении баз данных для создания, обработки и хранения больших объемов информации при решении различных прикладных задач.

Задачи дисциплины:

- создание у студентов упорядоченной системы знаний по проектированию баз данных, управлению и администрированию базами данных, основам структурированного языка запросов SQL, о методах сжатия больших информационных массивов, о реальных возможностях СУБД;
- ознакомление студентов с практикой создания информационной модели данных для конкретной предметной области и применения СУБД для создания приложений баз данных.

Дисциплина «Базы данных» относится к циклу ОПД.Ф.02 Цикл общепрофессиональных дисциплин. Федеральный компонент

1.3. Формируемые компетенции, а также перечень планируемых результатов обучения по дисциплине (модулю) (знания, умения владения), сформулированные в компетентностном формате.

Дисциплина (модуль) направлена на формирование следующих компетенций:

- профессиональными (ПК):
  - способен разрабатывать модели компонентов информационных систем, включая модели баз данных (ПК-4);
  - способность разрабатывать компоненты программных комплексов и баз данных, использовать современные инструментальные средства и технологии программирования (ПК-5);
  - способность инсталлировать программное и аппаратное обеспечение для информационных и автоматизированных систем (ПК-11).

В результате освоения дисциплины обучающийся должен демонстрировать следующие результаты образования:

1. Знать: ПК-4, ПК-5, ПК-11.
2. Уметь: ПК-5, ПК-11.
3. Владеть: ПК-4, ОК-5.

1.4. Место дисциплины (модулей) в структуре ООП ВПО

Дисциплина (модуль) «Базы Данных» является частью цикла (блока) дисциплин учебного плана по направлению подготовки 710100 «Информатика и вычислительная техника», специальности Компьютерные информационные системы в бизнесе. Для

освоения дисциплины (модулей) необходимы компетенции, сформированные в ходе изучения следующих дисциплин и прохождения практик: Основы программирования, Математическая логика и теория алгоритмов.

2. Структура дисциплины (модулей)

Структура дисциплины (модулей) для очной формы обучения

Общая трудоемкость дисциплины составляет 6 кредита, 180 ч., в том числе аудиторная работа обучающихся с преподавателем 96 ч., самостоятельная работа обучающихся 84 ч.

№ п/п	***Раздел, Темы Дисциплины***	***Семестр***	***Неделя семестра***	***Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)***				***Формы текущего контроля успеваемости (по неделям семестра) Форма промежуточной аттестации (по семестрам)***
				лекции	***Сем. Заня/лаб. заня***	***СРС***	***СРСиП***	
	Раздел 1.							
1	Назначение и компоненты (объекты) базы данных. Проектирование реляционной базы данных. Язык манипулирования данными для реляционной модели. Создание базы данных. Создание таблицы. Первичный и внешний ключ. Функциональны зависимости. Связывание таблиц. Виды связей. Индексирование.Каскадирование	5	1	2	4	1	1	опрос, проверка задания, посещаемость
2	Microsoft Access. Интерфейс приложения. Создание базы данных «Учебный процесс». Создание таблиц БД «Учебный процесс». Типы данных. Свойства полей. Определение свойств полей. Выбор первичного ключа. Схема данных. Обеспечение целостности данных	5	2	2	4	1	1	опрос, проверка задания, посещаемость
3	Модели данных (иерархическая, сетевая, реляционная). Обзор современных СУБД. Уровни представления баз данных. Декомпозиция отношений, транзитивные зависимости.	5	3	2	4	1	1	опрос, проверка задания, посещаемость

	Метод: сущность – связь.							
4	Различные представления о данных в базах данных. Основные этапы проектирования баз данных. 1. Различные представления о данных в базах данных	5	4	2	4	1	1	опрос, проверка задания, посещаемость
5	Основные этапы проектирования баз данных. 1. Концептуальное (инфологическое) проектирование. 2. Логическое (даталогическое) проектирование. 3. Физическое проектирование.	5	5	2	4	1	1	Модуль 1
	Раздел 2.							
6	Модели данных. 1. Классификация моделей данных. 2. Сетевая и иерархическая модели..	5	6	2	4	2	1	опрос, проверка задания, посещаемость
7	Реляционная модель данных. 1. Основные определения 2. Операции над отношениями. 3. Реляционная алгебра 4. Проектирование реляционной базы данных. Понятие нормализации.	5	7	2	4	2	1	опрос, проверка задания, посещаемость
8	Целостность баз данных. 1. Общие понятия и определения целостности 2. Операторы DDL в языке SQL с заданием ограничений целостности 3. Средства определения схемы базы данных 4. Средства изменения описания таблиц и средства удаления таблиц 5. Понятие представления. Операции создания представлений	5	8	2	4	2	1	опрос, проверка задания, посещаемость
9	Табличные языки запросов. 1. Понятие табличного языка запросов. 2. Общая характеристика языка QBE	5	9	2	4	2	1	Модуль 2
10	Программное обеспечение работы с современными базами данных. 1. Основные задачи программного обеспечения баз данных 2. Проблемы создания и ведения реляционных баз данных 3. Понятие языка SQL и его основные части	5	10	2	4	2	1	опрос, проверка задания, посещаемость
	Раздел 3.							
11	Язык SQL. 1. Общее представление об основных операторах языка SQL 2. Интерактивный режим работы с SQL (интерактивный SQL) 3. Использование языка SQL для выбора информации из таблицы. Использование SQL для выбора информации из нескольких таблиц 4. Использование SQL для вставки, редактирования и удаления данных в таблицах 5. Язык SQL и операции реляционной алгебры	5	11	2	4	2	1	опрос, проверка задания, посещаемость

12	Разработка приложений. 1. Механизм доступа к данным.Сравнение BDE и ADO 2. Создание базы данных MS Access 3. Практика работы с БД MS Access из Delphi	5	12	2	4	2	1	Модуль 3
13	Распределенные БД. 1. Модели "клиент-сервер" в технологии баз данных 2. Двухуровневые модели 3. Модель сервера приложений 4. Модели серверов баз данных	5	13	2	4	2	1	опрос, проверка задания, посещаемость
14	Банки данных. 1. Понятие банка данных. 2. Преимущества и недостатки банков данных.	5	14	2	4	2	1	опрос, проверка задания, посещаемость
15	Хранилища данных 1. Понятия о хранилищах 2. Типы хранилищ данных	5	15	2	4	3	2	Модуль 4
16	Консультация	5	16	2	4	0	0	

3. Содержание дисциплины (модулей)

№	Наименование раздела, темы дисциплины	Краткое содержание
1	Основные понятия	Развитие основных понятий представления данных. . Основные определения. Основные свойства баз данных. Классификация баз данных Системы управления базами данных (СУБД). Понятие системы управления базами данных. Классификация СУБД. Основные функции систем управления базами данных Различные архитектурные решения, используемые при реализации многопользовательских СУБД. Краткий обзор СУБД. Централизованная архитектура. Технология с сетью и файловым сервером (архитектура "файл-сервер"). Технология "клиент – сервер". Трехзвенная (многозвенная) архитектура "клиент – сервер". Краткий обзор СУБД. Настольные СУБД. Серверные СУБД
2	Модели данных	Классификация моделей данных. Сетевая и иерархическая модели. Реляционная модель данных. Основные определения. Операции над отношениями. Реляционная алгебра. Проектирование реляционной базы данных. Понятие нормализации Целостность баз данных. Общие понятия и определения целостности. Операторы DDL в языке SQL с заданием ограничений целостности. Средства определения схемы базы данных. Средства изменения описания таблиц и средства удаления таблиц. Понятие представления. Операции создания представлений. Табличные языки запросов. Понятие табличного языка запросов. Общая характеристика языка QBE
3	Язык SQL	Общее представление об основных операторах языка SQL Интерактивный режим работы с SQL (интерактивный SQL). Использование языка SQL для выбора информации из таблицы. Использование SQL для выбора информации из нескольких таблиц. Использование SQL для вставки, редактирования и удаления данных в таблицах. Язык SQL и операции реляционной алгебры Разработка приложений. Механизм доступа к данным. Сравнение

		BDE и ADO.

4. Конспект лекций

Тема 1. Назначение и компоненты (объекты) базы данных.

1. Назначение и компоненты (объекты) базы данных. Проектирование реляционной базы данных. Язык манипулирования данными для реляционной модели.
2. Создание базы данных. Создание таблицы. Первичный и внешний ключ.
3. Функциональны зависимости. Связывание таблиц. Виды связей. Индексирование.
4. Каскадирование

Тема 2. Запросы. Виды запросов.

1. Запросы. Назначение запросов. Виды запросов. Запросы на выборку.
2. Группировка и сортировка записей в запросах. Статистические функции.
3. Параметрические и перекрестные запросы.
4. Запросы действия. Обновление записей. Удаление записей. Добавление записей.
5. Запросы SQL. Инструкции языка SQL.

Тема 3. Различные архитектурные решения, используемые при реализации многопользовательских СУБД. Краткий обзор СУБД.

1. Централизованная архитектура.
2. Технология с сетью и файловым сервером (архитектура "файл-сервер").
3. Технология "клиент – сервер".
4. Трехзвенная (многозвенная) архитектура "клиент – сервер".
5. Краткий обзор СУБД. Настольные СУБД. Серверные СУБД.

Тема 4. Различные представления о данных в базах данных. Основные этапы проектирования баз данных.

1. Различные представления о данных в базах данных.
2. Основные этапы проектирования базы данных.

Тема 5. Основные этапы проектирования баз данных.

1. Концептуальное (инфологическое) проектирование.
2. Логическое (даталогическое) проектирование.
3. Физическое проектирование.

Тема 6. Модели данных.

1. Классификация моделей данных.
2. Сетевая и иерархическая модели..

Тема 7. Реляционная модель данных.

1. Основные определения
2. Операции над отношениями.
3. Реляционная алгебра
4. Проектирование реляционной базы данных. Понятие нормализации.

Тема 8. Целостность баз данных.

1. Общие понятия и определения целостности
2. Операторы DDL в языке SQL с заданием ограничений целостности
3. Средства определения схемы базы данных
4. Средства изменения описания таблиц и средства удаления таблиц
5. Понятие представления. Операции создания представлений

Тема 9. Табличные языки запросов.

1. Понятие табличного языка запросов.
2. Общая характеристика языка QBE

Тема 10. Программное обеспечение работы с современными базами данных.

1. Основные задачи программного обеспечения баз данных
2. Проблемы создания и ведения реляционных баз данных
3. Понятие языка SQL и его основные части

Тема 11. Язык SQL.
1. Общее представление об основных операторах языка SQL
2. Интерактивный режим работы с SQL (интерактивный SQL)
3. Использование языка SQL для выбора информации из таблицы. Использование SQL для выбора информации из нескольких таблиц
4. Использование SQL для вставки, редактирования и удаления данных в таблицах
5. Язык SQL и операции реляционной алгебры
Тема 12. Разработка приложений.
1. Механизм доступа к данным.Сравнение BDE и ADO
2. Создание базы данных MS Access
3. Практика работы с БД MS Access из Delphi
Тема 13. Распределенные БД.
1. Модели "клиент-сервер" в технологии баз данных
2. Двухуровневые модели
3. Модель сервера приложений
4. Модели серверов баз данных
Тема 14. Банки данных.
1. Понятие банка данных.
2. Преимущества и недостатки банков данных.
Тема 15. Хранилища данных
1. Понятия о хранилищах
2. Типы хранилищ данных
Тема 16. Безопасность данных.
1. Защита информации в базах данных.
2. Основные концепции обеспечения безопасности баз данных. 3. Реализация системы защиты в MS SQL Server 4. Проверка полномочий.

5. Информационные и образовательные технологии

Информационные и образовательные технологии

№ п/п	Наименование раздела	Виды учебной работы	Формируемые компетенции (указывается код компетенции)	Информационные и образовательные технологии
1	Раздел №1.	Лекция	ПК-4, ПК-5, ПК-11	Лекция-визуализация с применением слайд-проектора, Дискуссия, Лекция с разбором конкретных ситуаций
		Лабораторная работа	ПК-4, ПК-5, ПК-11	Дискуссия, Консультирование с разбором абстрактных ситуаций
		Самостоятельная	ПК-4, ПК-5, ПК-11	Использование

		работа		электронного курса лекций, Консультирование и проверка заданий посредством электронной почты
2	Раздел №2.	Лекция	ПК-4, ПК-5, ПК-11	Лекция-визуализация с применением слайд-проектора, Дискуссия, Лекция с разбором конкретных ситуаций
		Лабораторная работа	ПК-4, ПК-5, ПК-11	Дискуссия, Консультирование с разбором абстрактных ситуаций
		Самостоятельная работа	ПК-4, ПК-5, ПК-11	Использование электронного курса лекций, Консультирование и проверка заданий посредством электронной почты
3	Раздел №3.	Лекция	ПК-4, ПК-5, ПК-11	Лекция-визуализация с применением слайд-проектора, Дискуссия, Лекция с разбором конкретных ситуаций
		Лабораторная работа	ПК-4, ПК-5, ПК-11	Дискуссия, Консультирование с разбором абстрактных ситуаций
		Самостоятельная работа	ПК-4, ПК-5, ПК-11	Использование электронного курса лекций, Консультирование и проверка заданий посредством электронной почты

6. Фонд оценочных средств для текущего, рубежного и итогового контролей по итогам освоению дисциплины (модулей)

6.1. Перечень компетенций с указанием этапов их формирования в процессе освоения дисциплины

№ п/п	Контролируемые разделы дисциплины (модулей)	Код контролируемой компетенции (компетенций)	Наименование оценочного средства
1	Разделы №1, №2, №3	ПК-4, ПК-5, ПК-11	опрос, выполнение лабораторных работ
2	Разделы №1, №2, №3	ПК-4, ПК-5, ПК-11	опрос, выполнение лабораторных работ
3	Разделы №1, №2, №3	ПК-4, ПК-5, ПК-11	опрос, выполнение лабораторных работ
4	Разделы №1, №2, №3	ПК-4, ПК-5, ПК-11	опрос, выполнение лабораторных работ
5	Разделы №1, №2, №3	ПК-4, ПК-5, ПК-11	опрос, выполнение лабораторных работ
6	Разделы №1, №2, №3	ПК-4, ПК-5, ПК-11	опрос, выполнение лабораторных работ
7	Разделы №1, №2, №3	ПК-4, ПК-5, ПК-11	опрос, выполнение лабораторных работ
8	Разделы №1, №2, №3	ПК-4, ПК-5, ПК-11	опрос, выполнение лабораторных работ
9	Разделы №1, №2, №3	ПК-4, ПК-5, ПК-11	опрос, выполнение лабораторных работ

6.2. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности

Форма контроля	Срок отчетности	Макс. количество баллов	
		За одну работу	Всего
Текущий контроль: - опрос - выполнение лабораторных работ - посещаемость	 1, 2, 3, 4, 5 недели 1, 2, 3, 4, 5 недели 1, 2, 3, 4, 5 недели	 8 баллов 6 баллов 0,2	 До 40 баллов До 30 баллов 10 баллов
Рубежный контроль: (сдача модуля)	5 неделя	100%×0,2=20 баллов	
Итого за I модуль			До 100 баллов

Форма контроля	Срок отчетности	Макс. количество баллов	
		За одну работу	Всего
Текущий контроль: - опрос	6, 7, 8, 9, 10 недели	8 баллов	До 40 баллов

- выполнение лабораторных работ - посещаемость	6, 7, 8, 9, 10 недели 6, 7, 8, 9, 10 недели	6 баллов 0,2	До 30 баллов 10 баллов
Рубежный контроль: (сдача модуля)	10 неделя	100%×0,2=20 баллов	
Итого за II модуль			До 100 баллов

Форма контроля	Срок отчетности	Макс. количество баллов	
		За одну работу	Всего
Текущий контроль: - опрос - выполнение лабораторных работ - посещаемость	 11, 12, 13, 14, 15 недели 11, 12, 13, 14, 15 недели 11, 12, 13, 14, 15 недели	 8 баллов 6 баллов 0,2	 До 40 баллов До 30 баллов 10 баллов
Рубежный контроль: (сдача модуля)	15 неделя	100%×0,2=20 баллов	
Итого за III модуль			До 100 баллов
Итоговый контроль (экзамен)	Сессия	ИК = Бср × 0,8 + Бэкз × 0,2	

Полученный совокупный результат (максимум 100 баллов) конвертируется в традиционную шкалу:

Рейтинговая оценка (баллов)	**Оценка экзамена**
От 0 до 54	неудовлетворительно
от 55 до 69 включительно	удовлетворительно
от 70 до 84 включительно	хорошо
от 85 до 100	отлично

6.3. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания

Текущий контроль (0 - 80 баллов)

При оценивании посещаемости, опроса и выполнении лабораторных работ учитываются:

- посещаемость (10 баллов)
- степень раскрытия содержания материала (25 баллов);
- изложение материала (грамотность речи, точность использования терминологии и символики, логическая последовательность изложения материала (20 баллов);
- знание теории изученных вопросов, сформированность и устойчивость используемых при ответе умений и навыков (25 баллов).

Рубежный контроль (0 – 20 баллов)

При оценивании контрольной работы учитывается:

- полнота выполненной работы (задание выполнено не полностью и/или допущены две и более ошибки или три и более неточности) – 10 баллов;

- обоснованность содержания и выводов работы (задание выполнено полностью, но обоснование содержания и выводов недостаточны, но рассуждения верны) – 5 баллов;
- работа выполнена полностью, в рассуждениях и обосновании нет пробелов или ошибок, возможна одна неточность – 5 баллов.

Итоговый контроль (экзаменационная сессия) – ИК = Бср × 0,8 + Бэкз × 0,2

При проведении итогового контроля обучающийся должен ответить на 3 вопроса (два вопроса теоретического характера и один вопрос практического характера).

При оценивании ответа на вопрос теоретического характера учитывается:

- теоретическое содержание не освоено, знание материала носит фрагментарный характер, наличие грубых ошибок в ответе (0 баллов);
- теоретическое содержание освоено частично, допущено не более двух-трех недочетов (10 баллов);
- теоретическое содержание освоено почти полностью, допущено не более одного-двух недочетов, но обучающийся смог бы их исправить самостоятельно (20 баллов);
- теоретическое содержание освоено полностью, ответ построен по собственному плану (30 баллов).

При оценивании ответа на вопрос практического характера учитывается:

- ответ содержит менее 20% правильного решения (0-9 баллов);
- ответ содержит 21-89 % правильного решения (10-39 баллов);
- ответ содержит 90% и более правильного решения (40 баллов).

6.4. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности.

Перечень вопросов:

1. Понятие информации и данных.
2. Понятие ЭИС Классификация ЭИС.
3. Понятия банка данных.
4. Предметная область.
5. Классификация и основные свойства единиц информации.
6. Экономические показатели. Определение и структура показателя.
7. Понятия банка данных. Преимущества и недостатки.
8. Компоненты банков данных и их краткая характеристика.
9. Языковые средства СУБД.
10. Классификация банков данных.
11. Тенденции развития СУБД.
12. Характеристика современных реляционных СУБД.
13. Языки 4-го поколения и их реализация в современных СУБД.
14. Этапы проектирования баз данных.
15. Пользователи банков данных.
16. Функции администраторов банков данных.
17. Технические средства банков данных.
18. Инфологическое моделирование.
19. Сравнение методик ER-моделирования.
20. ERWin. Виды сущностей и их создание.
21. ERWin . Виды связей.
22. ERWin. Особенности построения ER-модели.
23. Базовая ER-модель. Виды сущностей.
24. Базовая ER-модель. Виды связей.
25. Особенности проектирования реляционных БД.

26. Даталогическое моделирование.
27. Факторы, влияющие на проектирование баз данных.
28. Виды связей между объектами и их отражение в даталогической модели.
29. Критерии оценки БД.
30. Основные понятия: отношение, ключ, вероятный ключ, внешний ключ.
31. Особенности реляционных моделей
32. Нормальные формы
33. Нормализация отношений.
34. Реляционная алгебра
35. Алгоритм перехода от ER-модели к структуре реляционной базы данных
36. CASE -средства проектирования БД
37. Возможности совместной обработки таблиц в реляционных СУБД
38. Возможности задания ограничений целостности в современных СУБД.
39. Ограничения целостности. Понятие и классификация.
40. Ограничения целостности связи. Понятие. Возможности реализации в современных СУБД.
41. Задание ограничений целостности в ERWin
42. Последовательные и списковые структуры.
43. Двоичные деревья сортировки.
44. Индексирование.
45. Хеширование.
46. Способы организации ввода данных в базу данных
47. Классификация экранных форм
48. Задание ограничений целостности в экранных формах
49. Характеристика генератора экранных форм конкретной СУБД.
50. Экранные формы для ввода и корректировки данных.
51. Табличные языки запросов. Общая характеристика.
52. Классификация запросов.
53. Особенности обработки полей разных типов. Работа с вычисляемыми полями.
54. Использование агрегирующих функций.
55. Совместная обработка таблиц
56. Реализация запросов со сложными условиями в табличных языках.
57. Корректировка данных с помощью табличного языка.
58. Язык SQL. Общая характеристика.
59. Стандарты SQL.
60. Общая структура команды Select языка SQL.
61. Вложенные запросы в SQL.
62. Корректировка данных в SQL.
63. SQL. Создание объектов.
64. SQL. Встроенный JOIN.
65. SQL. Понятие курсора.
66. SQL. Группировка данных. Использование обобщающих функций.
67. SQL. Возможности совместной обработки таблиц.
68. SQL. Упорядочение данных. Операция объединения.
69. SQL. Возможности задания состава колонок, выводимых в ответ.
70. SQL. Возможности задания условий отбора.

7. Учебно-методическое и информационное обеспечение дисциплины

7.1.Список источников и литературы

- Литература:

◦ Основная:

- Балдин К. В., Уткин В. Б. Информационные системы в экономике. Учебник / К.В. Балдин. - М.: Дашков и Ко, 2012. – 395с.
- Илюшечкин, В.М. Основы использования и проектирования баз данных: Учебное пособие / В.М. Илюшечкин. - М.: Юрайт, 2011 - 213с.
- Советов Б. Я. , Цехановский В. В. , Чертовской В. Д. Базы данных: теория и практика: учебник для бакалавров /Б.Я. Советов. - М.: ЮРАЙТ, 2011. - 459 с.
- Маркин А. В. Построение запросов и программирование на SQL. Учебное пособие / А.В. Маркин.- М.: Диалог-МИФИ, 2008. – 318 с.
- Мишенин А. И. Теория экономических информационных систем: Учебник 4-е изд., доп. и перераб. / А.И. Мишенин. - М.: Финансы и статистика, 2008. - 240 с.

◦ Дополнительная:

- Баженова, И.Ю. Основы проектирования приложений баз данных: Учебное пособие / И.Ю. Баженова. - М.: Интернет-Университет Информационных Технологий, 2009 - 325с.
- Бекаревич, Ю.Б. Microsoft Access за 21 занятие студента / Ю.Б. Бекаревич, Н.В. Пушкина. - СПб.: БХВ - Петербург, 2005 - 544с.
- Бумфрей Ф., Диренцо О., Даккетт Й. XML. Новые перспективы WWW. – Издательство «ДМКПресс», 2006 – 688 с.
- Гайдамакин, Н.А. Автоматизированные информационные системы, базы и банки данных: Вводный курс / Н.А. Гайдамакин. – М.: Гелиос АРВ, 2002 – 368с.
- Голицына, О.Л. Системы управления базами данных: Учебное пособие / О.Л. Голицына. - М.: ФОРУМ-ИНФРА-М, 2006 - 432с.
- Гринвальд, Р. ORACLE: Справочник / Р. Гринвальд, Д. Крейнс. - СПб: Символ-Плюс, 2005 - 976с.
- Дейт, К.Дж. Введение в системы баз данных. : Пер.с англ. / К.Дж. Дейт. - Киев; М.; СПб.: Вильямс, 1999 - 848с.
- Керман, М.К. Программирование и откладка в Delphi: Учебный курс / М.К. Керман. – Киев; М.; СПб.: Вильямс, 2002 – 672с.
- Кириллов, В.В. Введение в реляционные базы данных / В.В. Кириллов, Г.Ю. Громов. - СПб: БХВ - Петербург, 2009 - 464с.
- Мезенцев, К.Н. Автоматизированные информационные системы: учебник / К.Н. Мезенцев. – М.: Академия, 2010 – 176 с.
- Мусина, Т.В. Visual FoxPro 9.0:Учебный курс / Т.В. Мусина. - Киев: Век+;СПб:Корона-Век, 2011 - 736с.
- Смирнов, С.Н. Обработка документов средствами Oracle: Практикум по XML и JDBC /С.Н. Смирнов. – М.: Гелиос АРВ, 2004 – 192 с.

◦

7.2. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимый для освоения дисциплины (модулей)

1. Википедия - http://ru.wikipedia.org
2. Интернет-портал образовательных ресурсов по ИТ - http://www.intuit.ru
3. Портал математических интернет-ресурсов - http://www.math.ru/
4. Портал математических интернет-ресурсов - http://www.allmath.com/
5. Портал ресурсов по математике, алгоритмике и ИТ - http://algolist.manual.ru/
6. http://www.iprbookshop.ru/
7. http://kyrlibnet.kg/ru/

8. http://biblioteka.kg/

8. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся.

8.1. Планы практических (семинарских) и лабораторных занятий. Методические указания по организации и проведению

- Тема 1 (2 ч.) Основные понятия.
  - Цель занятия: Основные понятия и определения информационных систем, основанных на базах данных, предметная область, информационное обеспечение
  - Форма проведения – дискуссия
  - Содержание занятия:
- 1. Развитие основных понятий представления данных.
- 2. Основные определения.
- 3. Основные свойства баз данных.
- 4. Классификация баз данных.
  - ▪
  - Литература: []
  - Материально-техническое обеспечение занятия: ЭВМ

- Тема 2 (2 ч.) Системы управления базами данных (СУБД).
  - Цель занятия: Понятие СУБД. Назначение и основные компоненты системы баз данных. Основные функции СУБД
  - Форма проведения – разработка программного обеспечения
  - Содержание занятия:
- 1. Понятие системы управления базами данных.
- 2. Классификация СУБД.
- 3.Основные функции систем управления базами данных.
  - Литература: []
  - Материально-техническое обеспечение занятия: ЭВМ
  -
- Тема 3 (2 ч.) Различные архитектурные решения, используемые при реализации многопользовательских СУБД. Краткий обзор СУБД

  Цель занятия: Обзор современных систем управления базами данных (СУБД): dbase, Visual dBase, Clipper, FoxPro и Visual FoxPro, Paradox, Access, ORACLE, Microsoft SQL Server, Sybase, Informix, Линтер.
  - Форма проведения – разработка программного обеспечения
  - Содержание занятия:
- Причины появления СУБД. Этапы развития СУБД. Типовая организация современной СУБД. Требования к СУБД при выборе. Достоинства и недостатки СУБД.
  - Литература: []
  - Материально-техническое обеспечение занятия: ЭВМ

- Тема 4 (2 ч.) Различные представления о данных в базах данных. Основные этапы проектирования баз данных.

Цель занятия: Трехуровневая архитектура БД. Уровни представления баз данных: внешний, концептуальный, внутренний. Понятия схемы и подсхемы.

  - Форма проведения – разработка программного обеспечения
  - Содержание занятия:
- 1. Различные представления о данных в базах данных.

- 2. Основные этапы проектирования базы данных.
  - Литература: []
  - Материально-техническое обеспечение занятия: ЭВМ

- Тема 5 (2 ч.) Основные этапы проектирования баз данных
  - Цель занятия: Подходы к упрощению концептуальной модели данных. Исключение связей. Исключение атрибутов. Методика преобразования концептуальных структур данных в реляционные структуры
  - Форма проведения – разработка программного обеспечения
  - Содержание занятия:

    Концептуальное (инфологическое) проектирование.

    Логическое (даталогическое) проектирование.
    - 3. Физическое проектирование
  - Литература: []
  - Материально-техническое обеспечение занятия: ЭВМ

- Тема 6 (2 ч.) Модели данных.
  - Цель занятия: Трехуровневая архитектура БД. Уровни представления баз данных: внешний, концептуальный, внутренний. Понятия схемы и подсхемы
  - Форма проведения – разработка программного обеспечения
  - Содержание занятия:
- 1. Классификация моделей данных.
- 2. Сетевая и иерархическая модели.
  - Литература: []
  - Материально-техническое обеспечение занятия: ЭВМ

- Тема 7 (2 ч.) Реляционная модель данных

  Цель занятия: Жизненный цикл базы данных. Основные этапы. Достоинства и недостатки моделей. Примеры моделей.
  - Форма проведения – разработка программного обеспечения
  - Содержание занятия:
- 1. Основные определения
- 2. Операции над отношениями.
- 3. Реляционная алгебра
  - Литература: []
  - Материально-техническое обеспечение занятия: ЭВМ

- Тема 8 (2 ч.) Целостность баз данных
  - Цель занятия: Подходы к проектированию. Основные принципы проектирования Форма проведения – разработка программного обеспечения
  - Содержание занятия:
- 1. Общие понятия и определения целостности
- 2. Операторы DDL в языке SQL с заданием ограничений целостности
- 3. Средства определения схемы базы данных
- 4. Средства изменения описания таблиц и средства удаления таблиц
- 5. Понятие представления. Операции создания представлений
  - Литература: []
  - Материально-техническое обеспечение занятия: ЭВМ
  -

Тема 9 (2 ч.) Табличные языки запросов.

Цель занятия: Структурированный язык запросов SQL. Язык определения данных (DDL). Язык манипулирования данными для реляционной модели (DML). Основные понятия и компоненты языка SQL

- Форма проведения – разработка программного обеспечения
- Содержание занятия:

- 1. Понятие табличного языка запросов.
- 2. Общая характеристика языка QBE
  - Литература: []
  - Материально-техническое обеспечение занятия: ЭВМ

- Тема 10 (2 ч.) Программное обеспечение работы с современными базами данных
  - Цель занятия: Администрирование баз данных. Управление учетными записями и правами доступа
  - Форма проведения – разработка программного обеспечения
  - Содержание занятия:
    - 1. Основные задачи программного обеспечения баз данных
    - 2. Проблемы создания и ведения реляционных баз данных
    - 3. Понятие языка SQL и его основные части
  - Литература: []
  - Материально-техническое обеспечение занятия: ЭВМ

- Тема 11 (2 ч.) Язык SQL.
  - Цель занятия: Структурированный язык запросов SQL. Язык определения данных (DDL). Язык манипулирования данными для реляционной модели (DML). Основные понятия и компоненты языка SQL
  - Форма проведения – разработка программного обеспечения
  - Содержание занятия:
    - 1. Общее представление об основных операторах языка SQL
    - 2. Интерактивный режим работы с SQL (интерактивный SQL)
    - 3. Использование языка SQL для выбора информации из таблицы. Использование SQL для выбора информации из нескольких таблиц
    - 4. Использование SQL для вставки, редактирования и удаления данных в таблицах
    - 5. Язык SQL и операции реляционной алгебры
  - Литература: []
  - Материально-техническое обеспечение занятия: ЭВМ

- Тема 12 (2 ч.) Разработка приложений
  - Цель занятия: изучение разработки приложений на основе накопленных знаний в области SQL
  - Форма проведения – разработка программного обеспечения
  - Содержание занятия:
    - 1. Механизм доступа к данным.Сравнение BDE и ADO
    - 2. Создание базы данных MS Access
    - 3. Практика работы с БД MS Access из Delphi
  - Литература: []
  - Материально-техническое обеспечение занятия: ЭВМ

- Тема 13 (2 ч.) Распределенные БД
  - Цель занятия: изучение распределения в условия работающей БД

- Форма проведения – разработка программного обеспечения
- Содержание занятия:
  - 1. Модели "клиент-сервер" в технологии баз данных
  - 2. Двухуровневые модели
  - 3. Модель сервера приложений
  - 4. Модели серверов баз данных
- Литература: []
- Материально-техническое обеспечение занятия: ЭВМ

- Тема 14 (2 ч.) Банки данных.
  - Цель занятия: изучение понятий работы БД
  - Форма проведения – разработка программного обеспечения
  - Содержание занятия:
    - Понятие банка данных.
    - Преимущества и недостатки банков данных.
  - Литература: []
  - Материально-техническое обеспечение занятия: ЭВМ

- Тема 15 (2 ч.) Хранилища данных
  - Цель занятия: изучение взаимодействий хранилищ с сервером БД
  - Форма проведения – разработка программного обеспечения
  - Содержание занятия:
    - 1. Понятия о хранилищах
    - 2. Типы хранилищ данных
  - Литература: []
  - Материально-техническое обеспечение занятия: ЭВМ

8.2. Методические указания для обучающихся по освоению дисциплины (модулей)

Вид работы	Содержание (перечень вопросов)	Трудоемкость самостоятельной работы (в часах)	Рекомендации
Модуль №1.			
Конспектирование материала на лекционных занятиях Выполнение заданий лабораторных работ Выполнение тестовых и контрольных работ. Работа с учебной литературой. Написание программы	Стратегии разработки программных средств и систем и реализующие их модели жизненного цикла	5	[1, 2]
Конспектирование материала на лекционных занятиях Выполнение заданий лабораторных работ Выполнение тестовых и контрольных работ. Работа с учебной	Выбор модели жизненного цикла для конкретного проекта	5	[1, 2]

литературой. Написание программы			
Итого		10	
Модуль №2.			
Конспектирование материала на лекционных занятиях Выполнение заданий лабораторных работ Выполнение тестовых и контрольных работ. Работа с учебной литературой. Написание программы	Классические методологии разработки программных средств	8	[1, 2, 3]
Конспектирование материала на лекционных занятиях Выполнение заданий лабораторных работ Выполнение тестовых и контрольных работ. Работа с учебной литературой. Написание программы	Case-технологии технологии структурного анализа и проектирования программных средств	8	[1, 2, 4]
Итого		16	
Модуль №3.			
Конспектирование материала на лекционных занятиях Выполнение заданий лабораторных работ Выполнение тестовых и контрольных работ. Работа с учебной литературой. Написание программы	Методология объектно-технологии ориентированного анализа и проектирования сложных систем	8	[1, 2, 5]
Конспектирование материала на лекционных занятиях Выполнение заданий лабораторных работ Выполнение тестовых и контрольных работ. Работа с учебной литературой. Написание программы	Инструментальные средства разработки программного обеспечения	8	[1, 2, 5]
Итого		16	
Итого по дисциплине		42	

Вид работы	Содержание (перечень вопросов)	Трудоемкость самостоятельной работы (в часах)	Рекомендации
Раздел №1.			
Конспектирование материала на лекционных занятиях Выполнение заданий лабораторных работ Выполнение тестовых и контрольных работ. Работа с учебной литературой. Написание программы		2	
Конспектирование материала на лекционных занятиях Выполнение заданий лабораторных работ Выполнение тестовых и контрольных работ. Работа с учебной литературой. Написание программы		2	
Конспектирование материала на лекционных занятиях Выполнение заданий лабораторных работ Выполнение тестовых и контрольных работ. Работа с учебной литературой. Написание программы		2	
Конспектирование материала на лекционных занятиях Выполнение заданий лабораторных работ Выполнение тестовых и контрольных работ. Работа с учебной литературой. Написание программы		2	
Конспектирование материала на лекционных занятиях Выполнение заданий лабораторных работ		2	

Выполнение тестовых и контрольных работ. Работа с учебной литературой. Написание программы			
Итого		10	
Раздел №2.			
Конспектирование материала на лекционных занятиях Выполнение заданий лабораторных работ Выполнение тестовых и контрольных работ. Работа с учебной литературой. Написание программы		3	
Конспектирование материала на лекционных занятиях Выполнение заданий лабораторных работ Выполнение тестовых и контрольных работ. Работа с учебной литературой. Написание программы		3	
Конспектирование материала на лекционных занятиях Выполнение заданий лабораторных работ Выполнение тестовых и контрольных работ. Работа с учебной литературой. Написание программы		3	
Конспектирование материала на лекционных занятиях Выполнение заданий лабораторных работ Выполнение тестовых и контрольных работ. Работа с учебной литературой. Написание программы		3	
Конспектирование материала на		3	

лекционных занятиях Выполнение заданий лабораторных работ Выполнение тестовых и контрольных работ. Работа с учебной литературой. Написание программы			
Итого		15	
Раздел №3.			
Конспектирование материала на лекционных занятиях Выполнение заданий лабораторных работ Выполнение тестовых и контрольных работ. Работа с учебной литературой. Написание программы		3	
Конспектирование материала на лекционных занятиях Выполнение заданий лабораторных работ Выполнение тестовых и контрольных работ. Работа с учебной литературой. Написание программы		3	
Конспектирование материала на лекционных занятиях Выполнение заданий лабораторных работ Выполнение тестовых и контрольных работ. Работа с учебной литературой. Написание программы		3	
Конспектирование материала на лекционных занятиях Выполнение заданий лабораторных работ Выполнение тестовых и контрольных работ. Работа с учебной литературой.		3	

Написание программы			
Закрепление пройденного курса		5	
Итого		17	
Итого по дисциплине		42	

8.3. Методические рекомендации по подготовке письменных работ

Разработанное программное обеспечение должно быть предоставлено в скомпилированом виде, а так же в виде текстового файла, содержащего исходный код программы.

8.4. Иные материалы

Не предусмотрено.

9. Материально-техническое обеспечение дисциплины

Для изучения дисциплины, необходимо следующее оборудование: ЭВМ, проектор.

Требования к аудитории: компьютерный класс, имеющий ЭВМ в количестве идентичном количеству обучающихся, ЭВМ для преподавателя с подключенным проектором, наличие доски и средств для отображения/удаления информации на доске (мел/ветошь, маркер/губка).

9. Глоссарий

2. **АДМИНИСТРИРОВАНИЕ БД.** Управление базой данных (БД) как единым информационным ресурсом организации с целью эффективного обслуживания коллектива пользователей. Администрирование БД предполагает выполнение комплекса мероприятий, обеспечивающих точность, непротиворечивость, полноту, защиту и доступность данных в нужной форме, в нужном месте и в нужное время. Это касается вопросов обеспечения информацией как пользователей, так и приложений, работающих с БД. В рамках администрирования БД обеспечивается безопасность БД, осуществляется восстановление БД, производится ее обслуживание, осуществляется обучение пользователей, обеспечивается резервное копирование данных, организуется сопровождение БД, производится тестирование системы, организуется управление доступом пользователей к данным, обеспечивается целостность данных. Ответственность за поддержанием ресурсов БД может быть поделена между администратором данных и администратором базы данных. В некоторых организациях не проводится различия между этими двумя ролями администрирования БД.
3. **АДМИНИСТРАТОР БД (АБД)**. Специалист, отвечающий за управление техническими аспектами системы управления базы данных. Он отвечает за определение данных, разработку программ, ввод данных в БД, удаление данных, поддержание целостности и защиты данных, управление операциями БД. Кроме того, АБД участвует в планировании и разработке БД, в обучении пользователей всех категорий. Для АБД необходимо знание конкретной СУБД и системного окружения.
4. **АДМИНИСТРАТОР ДАННЫХ (АД).** Специалист, отвечающий за определение и реализацию общей политики по планированию и управлению данными в соответствии со стратегическим планом развития информационной системы корпорации. К обязанностям АД относится планирование развития базы данных,

разработка и сопровождение нормативов и стандартов по представлению данных, разработка бизнес-правил и "деловых" процедур, определение процедур сбора и обработки данных, непосредственно не зависящих от систем управления базой данных, концептуальное и логическое проектирование. АД консультирует и дает свои рекомендации руководству корпорации, контролирует соответствие общего направления развития базы данных целям корпорации.

5. **АРХИТЕКТУРА БАЗЫ ДАННЫХ (АРХИТЕКТУРА ANSI-SPARC).** Архитектура базы данных, состоящая из концептуального, внешнего и внутреннего уровней. Три уровня абстракции представления БД позволяют реализовать идею отделения логической структуры и манипуляции данными, как они понимаются пользователями, от физического представления, требуемого компьютерным оборудованием. На концептуальном уровне осуществляется концептуальное проектирование БД. Оно включает анализ информационных потребностей пользователей и определение необходимых им элементов данных. Внешний уровень составляет пользовательское представление данных БД. Каждая пользовательская группа получает свое представление данных в БД. Каждое представление данных дает ориентированное на пользователя описание элементов данных и отношений между ними. Внутренний уровень отражает представление БД на физическом уровне. За этот уровень отвечают разработчики БД. На этом уровне определяются индексы, указатели, методы доступа. Выгода трехуровневой архитектуры БД является независимость физического и логического представления данных. Трехуровневый подход к построению архитектуры БД был предложен в 1975 году Комитетом планирования стандартов и норм SPARC (Standards Planning and Requirements Committee) Национального Института Стандартизации США (American National Standard Institute -ANSI).
6. **АТОМАРНОСТЬ ДАННЫХ**. Свойство данных, означающее их неделимость для операций используемой модели представления и обработки данных. Т.е. данные рассматриваются, адресуются и обрабатываются как единое целое (как одно значение).
7. **АТРИБУТ**. Поименованная характеристика сущности. Например, сущность ГРАЖДАНИН может включать в себя следующие характеристики (атрибуты): НОМЕР ПАСПОРТА, ФАМИЛИЯ, ИМЯ, ОТЧЕСТВО, ГОД РОЖДЕНИЯ и др. С помощью атрибутов может быть идентифицирован экземпляр сущности. Атрибут может быть использован для представления связей между сущностями. Например, сущности ГРАЖДАНИН и ЗАКАЗ могут быть связаны между собой посредством атрибута НОМЕР ПАСПОРТА. При этом атрибут НОМЕР ПАСПОРТА должен быть включен в сущность ЗАКАЗ. Атрибуты могут быть ключевыми и описательными. Ключевой атрибут идентифицирует сущность. Описательный атрибут описывает одну из его характеристик. В рассматриваемом примере оправданно выбрать в качестве ключевого атрибута атрибут НОМЕР ПАСПОРТА, а в качестве описательных атрибутов все остальные. Каждый атрибут имеет набор свойств. Так например: размер, формат, маска ввода, значение по умолчанию, условие на значение, обязательность для ввода, список допустимых значений.
8. **БАЗА ДАННЫХ (БД)**. Совокупность специальным образом организованных, логически связанных данных, предназначенных для обеспечения информационных

потребностей организации. Другими словами БД это множество взаимосвязанных единиц данных, которые могут обрабатываться одной или несколькими прикладными системами. БД наряду с системой управления базой данных (СУБД), оборудованием и пользователями БД входит в состав системы баз данных. По сфере применения БД подразделяются на специальные и универсальные. По месту расположения данных БД подразделяются на локальные (персональные), общие (интегрированные) и распределенные. При проектировании и реализации БД используются модели данных - концептуальные способы структурирования данных. Наиболее известными моделями данных являются иерархическая, сетевая и реляционная. Процесс проектирования БД может быть представлен следующими этапами: формулировка и анализ требований, концептуальное (инфологическое) проектирование, проектирование реализации (датологическое проектирование), физическое проектирование.

9. **БАЗОВАЯ ТАБЛИЦА.** Таблица базы данных, содержащая реально хранимые на носителе данные. Структура и взаимосвязь базовых таблиц разрабатывается на этапах проектирования БД. На основе одной или нескольких базовых таблиц СУБД может формировать виртуальные таблицы (представления, хранимые запросы). В отличии от базовой таблицы, виртуальная таблица данных реально на носителе не хранится. Хранится лишь сам запрос, который создает виртуальную таблицу по требованию пользователя. Базовая таблица данных хранится на носителе.
10. **БАЗОВЫЙ ЯЗЫК.** Язык программы, в которую могут быть включены команды структурированного языка запросов SQL. Для каждой СУБД в качестве базового языка могут выступать различные языки : си, паскаль, бейсик и др.
11. **БАЗЫ ДАННЫХ КЛИЕНТ/СЕРВЕР**. База данных, функционирующая в локальной сети, состоящей из клиентских компьютеров, которые обслуживает компьютер-сервер. На серверном компьютере запускается программа - сервер базы данных. Эта программа обслуживает доступ клиентов к базе данных. На клиентских компьютерах запускаются прикладные программы, которые выполняют запросы к БД, установленной на сервере
12. **БЕЗОПАСНОСТЬ ДАННЫХ.** Свойство БД, которое характеризуется сведением к минимуму неправильного использования и повреждения данных пользователями. Администратор базы данных разрабатывает процедуры контроля, предотвращающие неправильное использование данных. Он назначает права исходной группе пользователей. Затем владельцы данных могут предоставлять доступ к данным другим пользователям. Доступы различаются: доступ только к некоторой части данных, доступ только для чтения, доступ с правом обновления. Доступ к БД контролируется специальным механизмом, например с использованием паролей. Таким образом снижается риск того, что данные одной группы пользователей будут повреждены другой группой. С обеспечением безопасности данных связана проблема целостности данных - точности и непротиворечивости данных, хранящихся в БД. Целостность данных обеспечивается написанием и использованием программ, контролирующих ввод данных в БД. В случае отказа оборудования для обеспечения безопасности данных предусматривают использование процедур резервного копирования и восстановления БД. В процессе параллельной обработки данных при

многопользовательском режиме данные могут быть повреждены. В связи с этим необходимо реализовать мероприятия, не допускающие повреждения данных в процессе их параллельной обработки. Для неуязвимости данных со стороны несанкционированного доступа к ним используются средства шифровки данных.

13. **БЛОКИРОВКА ЗАПИСЕЙ**. Предотвращает доступ к записи другой транзакции, пока первая транзакция не выполнит своих действий. Средства блокировки обеспечивают безопасность данных в случае параллельной обработки данных.
14. **БРОУЗЕР (WEB-броузер)**. Специальное клиентское программное обеспечение для WEB-среды, которое обеспечивает доступ к набору Интернет-сервисов.
15. **ВОЗМОЖНЫЙ КЛЮЧ.** Это атрибут или совокупность атрибутов, которые являются идентифицирующими атрибутами сущности и могут выступать в качестве первичного ключа для сущности после соответствующего их описания.
16. **ВОССТАНОВЛЕНИЕ БД.** Приведение БД в состояние, существовавшее незадолго до ее отказа. База данных приводится в состояние, которое она имела на момент начала незавершенных транзакций, а затем выполняются эти транзакции. В восстановлении после отказов ключевую роль играет протокол, в котором фиксируются все изменения вносимые в БД, а также состояние каждой транзакции.
17. **ВТОРИЧНЫЙ КЛЮЧ.** Это атрибут или совокупность атрибутов, которые используются в операции поиска требуемых данных в таблице.
18. **ДАННЫЕ**. Информация, фиксированная в определенной форме, пригодной для хранения, обработки и передачи. Примером компонентов первичных данных являются знаки естественного языка. Для использования в автоматизированных информационных системах эти знаки кодируются. Элементом данных является наименьшая семантически значимая единица информации. Данные после их интерпретации становятся информацией. Офисная информационная система хранит данные, которые обычно относятся к управлению. Это различные управленческие документы, счета, электронная почта и т.п. Однако современные системы могут работать и с такими данными, как электронные таблицы, фотографии, голосовые комментарии к тексту, диаграммы, аудио- и видеозаписи, географические данные и др.
19. **ДЕКОМПОЗИЦИЯ СХЕМ ОТНОШЕНИЙ**. Операция замены схемы отношения совокупностью ее подсхем, причем таких, что их объединение вновь приводит к исходной схеме.
20. **ЖИЗНЕННЫЙ ЦИКЛ БАЗЫ ДАННЫХ.** Процесс проектирования, реализации и поддержание системы баз данных.
21. **ЗАПИСЬ** Блок данных. Различают логические и физические записи. Логическая запись представляет собой поле или набор полей. Физическая запись - физический блок данных. Понятие записи используется на всех уровнях проектирования баз данных с различной степенью детализации их описания. Запись является минимальной доступной единицей данных. Обычно физические записи хранятся в одном из следующих четырех форматов: фиксированной длины, деблокированные; фиксированной длины, блокированные; переменной длины, деблокированные; переменной длины , блокированные. Если записи блокированы, то каждая физическая запись состоит более чем из одной логических записей. Для записей

переменной длины и блокированных предусмотрены соответственно области длина записи и длина блока.

22. **ЗАПРОС**. Средство просмотра, анализа и изменения данных из одной или нескольких таблиц. Сам запрос не содержит данных, но позволяет выбирать данные из таблиц и выполнять над ними ряд операций. Результатом выполнения запроса могут быть вновь созданные таблицы, измененные таблицы, выборка из одной или более таблиц. Запросы, связанные с изменениями таблиц, позволяют удалять записи по какому-либо критерию, добавлять записи в таблицы из других таблиц, модифицировать значения полей. Запросы на выборку позволяют: выводить отсортированную информацию по значению любых полей таблицы; фильтровать выводимую информацию с использованием в качестве фильтра математических выражений, в которых могут быть задействованы значения полей и параметров; группировать выводимые данные по различным полям; преобразовывать представление выводимой информации к виду удобному для ее восприятия. В современных СУБД для создания типовых запросов используются средства создания по образцу. Пользователю предоставляется специальные бланки, в которых он указывает таблицы и запросы на основе которых формируется запрос, выводимые поля, способ сортировки, условие отбора, тип запроса и др. В реляционных базах данных и некоторых других приложениях для создания запросов предусмотрено использование специального языка запросов SQL - структурированного языка запросов.
23. **ИЗБЫТОЧНОСТЬ ДАННЫХ**. Дублирование данных в базе данных. Избыточность данных может возникнуть при непродуманном проектировании реляционных таблиц. Например, в таблице КНИГИ хранится информация об издательствах, в которых изданы книги. Так как каждое издательство выпустило не одну книгу, то информация об одном и том же издательстве в таблице КНИГИ будет дублироваться столько раз сколько книг это издательство выпустило. В результате этого неоправданно расходуется память компьютера. Для исключения избыточности данных в приведенном примере необходимо спроектировать вторую таблицу ИЗДАТЕЛЬСТВА, а затем связать две таблицы между собой. Для исключения избыточности данных выполняется процесс нормализации отношений. В некоторых случаях избыточность может вводиться искусственно при проектировании БД с целью повышения скорости обработки запросов пользователей, или с целью повышения надежности системы в условиях работы со сбоями.
24. **ИЕРАРХИЧЕСКАЯ МОДЕЛЬ ДАННЫХ**. Модель данных, в которых связи между данными имеют вид древовидной иерархической структуры. Например, база данных КЛИЕНТЫ может строиться на основе следующей модели. На верхнем уровне иерархии данные о КЛИЕНТАХ. КЛИЕНТУ подчинены его СЧЕТА, СЧЕТАМ в свою очередь подчинены СТРОКИ СЧЕТОВ. В иерархической базе данных эти файлы связываются между собой физическими указателями, или полями данных, добавленных к отдельным записям. Каждая запись о клиенте содержит указатель первой записи счета этого клиента. В свою очередь, записи счетов содержат указатели на другие записи счетов и на записи строк счетов.

25. **ИНФОРМАЦИЯ.** Сведения о каком-либо событии, явлении, процессе или объекте. Во всех предметных областях основополагающую роль играют операции сбора, передачи, хранения и переработки информации. Количественными аспектами информации занимается теория передачи информации Качественными аспектами информации занимается теория информационных систем. В последнем случае важен смысл информации, ее содержание. Информация представляет собой содержание данных, которому придается смысл в определенном контексте. Информация кодируется с помощью данных.
26. **ИНФОРМАЦИОННАЯ СИСТЕМА**. Автоматизированная система, организующая данные в целях получения информациии. Под информационной системой можно обозначать любые совокупности данных и методов их обработки. Примерами информационных систем являются прикладные программы, текстовые процессоры, электронные таблицы, системы доказательства теорем, информационно-поисковые системы, системы баз данных и др.
27. **КОНЦЕПТУАЛЬНОЕ ПРОЕКТИРОВАНИЕ БД**. Создание модели предметной области в форме, доступной для понимания пользователем, Эта модель служит для достижения взаимопонимания между различными группами пользователями, а также между пользователями и разработчиками БД. Такая модель строится без учета физического представления данных в конкретной СУБД и называется инфологической моделью предметной области. Для описания инфологической модели могут быть использованы различные средства, в том числе и естественный язык. Но из-за громоздкости описания модели на естественном языке и неоднозначности его трактовки обычно используются формализованные языки. Так как модель должна обеспечивать ввод новых данных без изменения существующих, то она должна обладать свойством расширяемости. Инфологическая модель обычно строится совместно потенциальными пользователями и разработчиками БД. На этапе ее построения осуществляется структуризация данных и описание связей между ними. Представление информационной структуры обычно реализуется в графической форме. Инфологическая модель БД строится на основе этапа формулировки и анализа требований к БД. В свою очередь она используется для этапа даталогического проектирования БД.
28. **КЛИЕНТ БД**. Это так называемый "клиентский" процесс, требующий для своей работы определенных ресурсов, в том числе, как правило, и сетевых (однако, в некоторых случаях клиентский процесс может выполняться на том же компьютере, что и серверный процесс). В контексте базы данных клиент БД управляет пользовательским интерфейсом и логикой приложения. Клиент БД принимает от пользователя запрос, проверяет синтаксис и генерирует запрос к базе данных в соответствии с логикой приложения. Затем передает запрос серверному процессу и ожидает ответа. Сервер принимает запрос от клиента, обрабатывает поступивший к базе данных запрос и затем посылает ответ на запрос обратно клиенту. Клиент БД, получив ответ, форматирует поступившие от сервера данные и представляет их пользователю.
29. **КЛЮЧ ПЕРВИЧНЫЙ**. Первичный ключ - это набор атрибутов, однозначно определяющих каждую строку реляционной таблицы. Ключ состоящий более чем

из одного атрибута называется составным ключем. Первичный ключ должен удовлетворять требованиям уникальности и минимальности. Уникальность - для любой пары кортежей рассматриваемого отношения найдется хотя бы один атрибут, значение которого в этой паре кортежей различны. Минимальность - ни один из атрибутов не может быть удален из ключа без нарушения требования уникальности. В реляционной таблице может оказаться более одного набора атрибутов, который можно выбрать в качестве первичного ключа. Такой набор называется потенциальным или возможным ключом. Один из возможных ключей объявляется первичным, остальные альтернативными. Первичный ключ это потенциальный ключ, выбранный для преимущественного использования в целях однозначного определения строк таблицы. Внешний ключ это набор атрибутов одной таблицы, являющийся первичным ключом другой таблицы. Внешний ключ используется для организации логических связей между таблицами. Они используются для того, чтобы связать данные одной таблицы с данными в другой таблице. В некоторых СУБД допускается, что атрибуты внешнего ключа не обязательно должны иметь те же имена, что атрибуты первичного ключа, которым они соответствуют, но они должны базироваться на одних и тех же доменах.

30. **КОНСТРУКТОРЫ ОБЪЕКТОВ БД**. Инструментальные средства, позволяющие описывать и редактировать объекты БД. В инструментальных средствах СУБД предусматриваются визуальные средства, которые позволяют проектировать таблицы, связи между таблицами, формы, запросы, отчеты, макросы без использования языков описания данных и манипулирования данными. При этом могут быть задействованы соответствующие панели инструментов и панели элементов, использование которых обеспечивает оперативное и удобное описание элементов объектов БД, описание их свойств.
31. **КОНТРОЛЬ ДОСТУПА**. Комплекс средств и мероприятий, позволяющих избежать несанкционированный доступ к данным. Средства контроля особенно важны при совместном использовании данных.
32. **МАКРОС.** Небольшая программа на языке макрокоманд системы. С помощью макросов можно определять реакцию приложения на различные события в формах и отчетах. Например, на такие события, как нажатие кнопок, изменение данных, открытие и закрытие форм, получение фокуса элементом управления и др. Макрос в СУБД представляет собой структуру из одной или более макрокоманд, которые выполняются последовательно, либо в порядке, заданном определенными условиями. Использование макросов во многих случаях позволяет избежать программирования на языках управления данными. Каждая макрокоманда имеет имя и, возможно, один или несколько аргументов, которые задаются разработчиком БД. Использование макросов оправдано тем, что их легко создавать и во многих случаях отпадает необходимость в использовании языков программирования.
33. **МАСТЕРА СОЗДАНИЯ ОБЪЕКТОВ БД**. Инструментальные средства, которые упрощают выполнение типовых операций при разработке баз данных. Используются для проектирования таблиц, форм, запросов, отчетов. Разработчику объекта БД предлагается выполнить последовательность шагов, при выполнении которых описывается состав и свойства объекта. На каждом шаге необходимо

выбрать или ввести варианты ответов. Мастера позволяют существенно сократить время проектирования типовых объектов, свести к минимуму возможность внесения ошибок.

34. **МОДЕЛЬ ПРЕДМЕТНОЙ ОБЛАСТИ.** Формализованное описание предметной области. Модель используется для исследования и проектирования объектов предметной области и связи между ними. При построении модели необходимо абстрагироваться от свойств объектов, не существенных с точки зрения задач, решаемых на основе использования модели. Основным требованием к модели является адекватное отображение предметной области. Модель в полной мере должна отображать предметную область без учета несущественных для решаемой проблемы свойств предметной области. В связи с этим средства представления модели должны обладать достаточными выразительными возможностями для отображения предметной области и явлений и процессов, имеющих в ней место. Модель должна быть непротиворечивой. Она должна быть единым интегрированным описанием предметной области и однозначно отражать взгляды и потребности всех пользователей системы, для построения которой разрабатывается модель.

11. Приложения.

Краткий конспект лекций.

Модуль 1.

Проектирование базы данных. Создание таблиц.

Лекций -3 ч.

Темы лекционных занятий:

- Назначение и компоненты (объекты) базы данных. Проектирование реляционной базы данных. Язык манипулирования данными для реляционной модели.
- Создание базы данных. Создание таблицы. Первичный и внешний ключ.
- Функциональны зависимости. Связывание таблиц. Виды связей. Индексирование. Каскадирование.

Назначение и компоненты (объекты) базы данных.

БД – некоторое множество данных, которое структурировано и непротиворечиво.

БД обладает удобством поиска информации (полнотекстовый поиск, когда ищется все вхождения некоторого слова в содержащихся в базе данных или поиск по любым полям[применяется для реляционных БД])

СУБД имеют 2 собственных языка программирования: язык описания данных, язык манипулирования данными

Реляционная БД содержит множество прямоугольных таблиц с данными. Индекс таблицы – вспомогательные. файлы, которые создаются для того, чтобы ускорить выполнение операции с участием этой таблицы. Индекс строится или для определенного столбца или для совокупности столбцов. Причем нельзя путать индекс совокупности и несколько различных индексов для отдельных столбцов.

Индексы бывают уникальными и неуникальными. Если создается уникальный индекс по некоторому столбцу, то система гарантирует, что в данном столбце не будет повторяющихся значений.

БД – совместный используемый набор логически связанных данных, хранимый на ЭВМ, предназ-наченный для удовлетворения потребностей информационных или пользователей

Свойства БД

- интегрированность (БД-набор логически связ.фак-торов, собранных вместе с мин избыточностью.)
- разделяемость (одни и те же данные могут имп-ся разными пользователями одновременно)
- персистентность (объемы и персистентный хар-р;

Данные после окончания процесса сохр-ся) Все удаления объектов произв-ся явным образом.

Пользователи СУБД

- Администратор БД отвечает за управление БД, включая ее планирование, разработку, хранение, сопровождение, физич. реализацию
- Разработчики БД – разработчики физической и логической структур БД
- Прикладные программисты – разработчики приложений, реализующих взаимодействие пользователей с БД
- Конечные пользователи – потребители информации

Реляционная модель данных, содержащая набор четких предписаний к базовой организации любой реляционной системы управления базами данных (СУБД), позволяет пользователям работать в ненавигационной манере, т.е. для выборки информации из БД человек должен всего лишь указать список интересующих его таблиц и те условия, которым должны удовлетворять выбираемые данные. СУБД скрывает от пользователя выполняемые ей последовательные просмотры таблиц, выполняя их наиболее эффективным образом. Очень важная особенность реляционных систем состоит в том, что результатом выполнения любого запроса к таблицам БД является также таблица, которую можно сохранить в БД и/или по отношению к которой можно выполнять новые запросы.

Базовым требованием к реляционным СУБД является наличие мощного и в тоже время простого языка, позволяющего выполнять все необходимые пользователям операции. В последние годы таким повсеместно принятым языком стал язык реляционных БД SQL - Structured Query Language (теперь все чаще название языка понимается как Standard Query Language).

Создание базы данных. Создание таблицы. Первичный и внешний ключ.

При разработке структуры таблицы необходимо указать, какую информацию она должна содержать. После того как определена информация, которая должна быть включена в таблицу, ее следует разбить по полям. Этот процесс включает в себя выбор имени поля, которое в таблице должно быть уникальным. Необходимо включать в имя поля как можно больше информации о его содержимом, но в то же время стараться избегать длинных имен. Обычно при разработке баз данных в различных таблицах могли содержаться поля с одинаковыми именами (чаще всего таблицы связывались по этим полям). Access также позволяет использовать одинаковые имена полей в различных таблицах, но по причине того, что Access использует имена полей при определении условий целостности данных, лучше не допускать повторения имен полей.

Перед началом заполнения таблицы нужно определить связи этой таблицы с другими таблицами базы данных, определить ключевое поле и создать индексы.

Функциональны зависимости.

Связь между таблицами устанавливает отношения между совпадающими значениями в ключевых полях — обычно между полями разных таблиц. В большинстве случаев связывают **ключевое поле** одной таблицы с соответствующим ему полем (часто имеющим то же имя), которое называют полем **внешнего ключа** во второй таблице. Таблица, содержащая ключевое поле, называется **главной**, а таблица, содержащая внешний ключ — **связанной.** **Существует четыре вида отношений:**

- **Один к одному**. При использовании отношения "один к одному" запись в таблице "А" (главной таблице) может иметь не более одной связанной записи в таблице "В" (связанной таблице) и наоборот. Этот тип связи используют не очень часто, поскольку такие данные могут быть помещены в одну таблицу. Связь с отношением "один к одному" используют для разделения очень широких таблиц или для отделения части таблицы по соображениям защиты. В таблицах связанных отношением "один к одному" ключевые поля должны быть уникальными.
- **Один ко многим**. Связь с отношением "один ко многим" является наиболее часто используемым типом связи между таблицами. В такой связи каждой записи в таблице "А" могут соответствовать несколько записей в таблице "В", а запись в таблице "В" не может иметь более одной соответствующей ей записи в таблице "А". В таблице "А" ключевое поле должно быть уникальным. Связь с отношением "один ко многим" таблиц "А" и "В" можно рассматривать как связь с отношением "многие к одному" таблиц "В" и "А".
- **Многие к одному**. При использовании отношения "многие к одному" одной записи в таблице "А" может соответствовать одна запись в таблице “В”, а запись в таблице "В" может иметь более одной соответствующей ей записи в таблице "А". В таблице "А" ключевое поле должно быть уникальным. Связь с отношением "один ко многим" таблиц "А" и "В" можно рассматривать как связь с отношением "многие к одному" таблиц "В" и "А".
- **Многие ко многим**. При использовании отношения "многие ко многим" одной записи в таблице "А" могут соответствовать несколько записей в таблице "В", а одной записи в таблице "В" несколько записей в таблице "А". В этом случае не требуется уникальности ключевых полей таблиц "А" и "В". Этот вид отношения используется для связывания таблиц, имеющих только внешние ключи.

Связанные поля не обязательно должны иметь одинаковые имена, однако, они должны иметь одинаковые типы данных. Кроме того, связываемые поля типа "Числовой" должны иметь одинаковые значения свойства "Размер поля". Исключениями из этого правила являются поля счетчика с последовательной нумерацией, которые могут связываться с числовыми полями размера "Длинное целое". Существует возможность создать связь между текстовыми полями разной длины, но это может вызвать сложности при создании запросов.

Автоматическое обеспечение ссылочной целостности данных является важной особенностью Access. Если на связь между таблицами наложены условия ссылочной целостности, то Access не позволяет добавлять в связанную таблицу записи, для которых нет соответствующих записей в главной таблице, и изменять записи в главной таблице таким образом, что после этого в связанной таблице появятся записи, не имеющие главных записей; а также удалять записи в главной таблице, для которых имеются подчиненные записи в связанной таблице. Условия целостности данных определяют систему правил, используемых в Access для поддержания связей между записями в связанных таблицах. Эти правила делают невозможным случайное удаление или изменение связанных данных. После наложения условий целостности данных на операции со связанными таблицами накладываются ограничения. Невозможно ввести в поле внешнего ключа связанной таблицы значение, не содержащееся в ключевом поле главной таблицы.

Необходимо так же определить первичный ключ. Если таблица никогда не будет использоваться в качестве главной, то ключ для нее определять не нужно. В главных таблицах обычно содержится информация о реальных объектах, причем с каждым объектом ассоциируется только одна запись. Определение ключа таблицы является простейшим способом предотвращения появления в таблице одинаковых записей. В главной таблице связи должен быть определен первичный ключ. Access считает таблицы, у которых такой ключ не определен, подозрительными. При открытии таких таблиц в режиме конструктора появляется диалоговое окно, сообщающее о том, что ключ таблицы не определен. Ключ можно определить и в связанных таблицах, что поможет избежать появления повторяющихся данных. Ключ таблицы можно задать по значению нескольких полей. Access автоматически индексирует таблицу по значению ключа, но может потребоваться создать дополнительные индексы по значениям других полей. Индексы позволяют ускорить поиск данных в тех полях таблицы, по которым она проиндексирована. Каждая таблица Access может иметь до 32 индексов, 5 из которых могут быть составными (в составной индекс может входить не более 10 полей). Тем не менее, не стоит увлекаться созданием индексов для каждого поля таблицы и всех их комбинаций, поскольку это может существенно замедлить процесс заполнения таблицы (при изменении или добавлении записи автоматически обновляются все индексы).

Одним из основных принципов разработки реляционных баз данных является то, что все данные, содержащиеся в поле таблицы, должны иметь один и тот же тип. Для каждого поля таблицы необходимо задать тип данных. По умолчанию используется тип данных «Текстовый». При задании типа данных поля можно также указать размер, формат и другие параметры, влияющие на отображение значения поля и точность числовых данных.

Модуль 2.

Запросы. Виды запросов.

Лекций - 5 ч.

Темы лекционных занятий:

- Запросы. Назначение запросов. Виды запросов. Запросы на выборку.
- Группировка и сортировка записей в запросах. Статистические функции.
- Параметрические и перекрестные запросы.
- Запросы действия. Обновление записей. Удаление записей. Добавление записей.
- Запросы SQL. Инструкции языка SQL.

Запросы

Запросы являются важным инструментом в любых системах управления базами данных. Они используются для выделения, обновления и добавления новых записей в таблицы. Чаще всего запросы используются для выделения специфических групп записей, чтобы удовлетворить определенному критерию. Кроме того, их можно использовать для получения данных из различных таблиц, обеспечивая единое представление связанных элементов данных.

В Access существует несколько типов запросов для различных целей

- **Запросы на выборку** отображают данные из одной или нескольких таблиц в виде таблицы.
- **Перекрестные запросы** собирают данные из одной или нескольких таблиц в формате, похожем на формат электронной таблицы. Эти запросы используются для анализа

данных и создания диаграмм, основанных на суммарных значениях числовых величин из некоторого множества записей.

- **Запросы на изменение** используются для создания новых таблиц из результатов запроса и для внесения изменений в данные существующих таблиц. С их помощью можно добавлять или удалять записи из таблицы и изменять записи согласно выражениям, задаваемым в режиме конструктора запроса.
- **Запросы с параметрами** — это такие запросы, свойства которых изменяются пользователем при каждом запуске. При запуске запроса с параметром появляется диалоговое окно, в котором нужно ввести условие отбора. Этот тип запроса не является обособленным, т. е. параметр можно добавить к запросу любого типа.

Запрос SQL — это запрос, создаваемый при помощи инструкций SQL. Язык

SQL (Structured Query Language) используется при создании запросов, а также для обновления и управления реляционными базами данных, такими как базы данных Microsoft Access.Когда пользователь создает запрос в режиме конструктора запроса, Microsoft Access автоматически создает эквивалентную инструкцию SQL. Фактически, для большинства свойств запроса, доступных в окне свойств в режиме конструктора, имеются эквивалентные предложения или параметры языка SQL, доступные в режиме SQL. При необходимости, пользователь имеет возможность просматривать и редактировать инструкции SQL в режиме SQL. После внесения изменений в запрос в режиме SQL его вид в режиме конструктора может измениться.

Некоторые запросы, которые называют запросами SQL, невозможно создать в бланке запроса. Для запросов к серверу, управляющих запросов и запросов на объединение необходимо создавать инструкции SQL непосредственно в окно запроса в режиме SQL. Для подчиненного запроса пользователь должен ввести инструкцию SQL в строку Поле или Условие отбора в бланке запроса.

Запрос на объединение.

Запрос на объединение объединяет поля из двух или нескольких таблиц или запросов в одно поле в результатах запроса. Такие запросы обычно используют для объединения данных из двух таблиц. Например, может потребоваться объединение сведений из таблиц «Поставщики» и «Клиенты» о названиях и городах для всех компаний из Бразилии. После этого можно на основе запроса на объединение создать запрос на создание таблицы, генерирующий новую таблицу.

Модуль 3.

Создание интерфейса пользователя. Разработка форм и отчетов.

Лекций - 4 ч.

Темы лекционных занятий:

- Разработка форм. Разделы формы. Свойства формы и ее разделов.
- Многотабличные формы. Рекомендации по оформлению.
- Составление отчетов. Разделы отчета. Использование вычисляемых значений.
- Кнопочная форма. Автоматизация работы приложения.

Формы

Формы Access позволяют создавать пользовательский интерфейс для таблиц базы данных. Хотя для выполнения тех же самых функций можно использовать режим таблицы, формы предоставляют преимущества для представления данных в упорядоченном и привлекательном виде. Формы позволяют также создавать списки

значений для полей, в которых для представления множества допустимых значений используются коды. Правильно разработанная форма ускоряет процесс ввода данных и минимизирует ошибки.

Формы создаются из набора отдельных элементов управления: текстовые поля для ввода и редактирования данных, кнопки, флажки, переключатели, списки, метки полей, а также рамки объектов для отображения графики и объектов OLE. Форма состоит из окна, в котором размещаются два типа элементов управления: динамические (отображающие данные из таблиц), и статические (отображающие статические данные, такие, как метки и логотипы).

Формы Access являются многофункциональными; они позволяют выполнять задания, которые нельзя выполнить в режиме таблицы. Формы позволяют производить проверку корректности данных, содержащихся в таблице. Access позволяет создавать формы, включающие другие формы (форма внутри формы называется подчиненной). Формы позволяют вычислять значения и выводить на экран результат.

Приложения для работы с базами данных можно разделить на три основных категории:

- **Управление транзакциями**. Эти формы, работают непосредственно с таблицами, при условии, что активна только одна таблица. Если для добавления и редактирования информации в более чем одной таблице используется единственная форма, то можно создать запрос на обновление, включающий все нужные поля, а затем создать форму на основе этого запроса. Главная форма также может использовать единственную таблицу в качестве источника данных, но, кроме того, включать подчиненные формы, в качестве источника данных которых используются другие таблицы. Подчиненная форма, которая появляется ниже текстовых полей главной формы, используется для просмотра и добавления данных в таблицу, связанную с подчиненной формой.
- **Доступ к данным**. Такие формы предназначены только для просмотра информации. Формы, предназначенные только для доступа к данным, можно разрабатывать как с целью перспективного планирования действий, так и для принятия экстренного решения. Экстренное решение приводит к единственному действию, такому, например, как предоставление большого кредита клиенту, или посылка представителя по продажам для определения причины отказа клиента от товара. Приложения Access позволяют использовать три уровня вложенности форм: главная, подчиненная и подчиненная форма, включенная в предыдущую подчиненную форму. Формы, предназначенные только для доступа к данным, в качестве источника данных часто используют запросы, которые суммируют данные по временным промежуткам.
- **Администрирование баз данных**. Приложения этой категории выполняют административные функции такие, как создание баз данных или таблиц, разграничение прав доступа пользователей к объектам баз данных, обеспечение безопасности с помощью шифрования, периодическое уплотнение базы данных, а также операции резервного копирования.

  Приложения для администрирования баз данных требуют наличия полного доступа ко всем объектам, содержащимся в базе данных.

Простейший путь создания основной и подчиненной форм - использование Мастера форм, который позволяет создавать формы (с подчиненными или без), содержащие поля из одной или более таблиц или запросов. Мастер форм создает базовый внешний вид формы и добавляет текстовые поля для отображения и редактирования значений полей таблиц. Независимо от того, являетесь ли вы профессиональным разработчиком приложений или начинающим пользователем, использование Мастера форм заметно упрощает и ускоряет процесс создания простых форм, которые затем можно усовершенствовать в режиме конструктора.

Отчёты

Конечным продуктом большинства приложений баз данных является отчет. В Access отчет представляет собой специальный тип непрерывных форм, предназначенных для печати. Для создания отчета, который можно распечатать и распределить между потребителями, Access комбинирует данные в таблицах, запросах и даже формах. Распечатанная версия формы может служить отчетом.

1. **Отчеты в одну колонку** представляют собой один длинный столбец текста, содержащий значения всех полей каждой записи таблицы или запроса. Надпись указывает имя, а справа от нее указывается значение поля. Новое средство Access Автоотчет позволяет создать отчет в одну колонку щелчком по кнопке панели инструментов Автоотчет. Отчеты в одну колонку используются редко, поскольку такой формат представления данных приводит к лишней трате бумаги.
2. **В ленточных отчетах** для каждого поля таблицы или запроса выделяется столбец, а значения всех полей каждой записи выводятся по строчкам, каждое в своем столбце. Если в записи больше полей, чем может поместиться на странице, то дополнительные страницы будут печататься до тех пор, пока не будут выведены все данные; затем начинается печать следующей группы записей.
3. **Многоколоночные отчеты** создаются из отчетов в одну колонку при использовании колонок "газетного" типа или колонок "змейкой", как это делается в настольных издательских системах и текстовых процессорах. Информация, которая не помещается в первом столбце, переносится в начало второго столбца, и так далее. Формат многоколоночных таблиц позволяет сэкономить часть бумаги, но применим не во всех случаях, поскольку выравнивание столбцов едва ли соответствует ожиданиям пользователя.

В основном отчёты проще всего построить при помощи Мастера отчётов. Мастер отчетов старается создать оптимальный вариант окончательного отчета с первой попытки. Обычно мастер в достаточной степени приближается к законченному варианту, так что тратиться намного меньше времени на редактирование базового отчета мастера, чем ушло бы на создание отчета из незаполненного шаблона.

Кнопочные формы.

Главная кнопочная форма создается с целью навигации по базе данных, т.е. она может использоваться в качестве главного меню БД. Элементами главной кнопочной формы являются объекты форм и отчётов.

Запросы и таблицы не являются элементами главной кнопочной формы. Поэтому для создания кнопок Запросы или Таблицы на кнопочной форме можно использовать макросы. Сначала в окне базы данных создают макросы «Открыть Запрос» или «Открыть Таблицу» с уникальными именами, а затем в кнопочной форме создают кнопки для вызова этих макросов.

Для одной базы данных можно создать несколько кнопочных форм. Кнопки следует группировать на страницах кнопочной формы таким образом, чтобы пользователю было понятно, в каких кнопочных формах можно выполнять определенные команды (запросы, отчеты, ввода и редактирования данных). Необходимо отметить, что на подчиненных кнопочных формах должны быть помещены кнопки возврата в главную кнопочную форму.Построение всех кнопочных форм, необходимых для управления сложным приложением, может оказаться довольно трудоёмким процессом. Специальная надстройка MS "Access" – диспетчер кнопочных форм – помогает выполнить эту работу. Эта надстройка применяет довольно сложную технику для управления всеми кнопочными формами с помощью одной формы и использует специальную таблицу – драйвер с именем Элементы кнопочной формы, что позволяет определить любое число кнопочных форм и создать до 8 кнопок в каждой из них.

Модуль 4.

Программирование на языке макросов. Программирование на языке VBA.

Лекций - 5 ч.

Темы лекционных занятий:

- Макросы. Использование макросов в Access. Условия выполнения макросов. Группы макросов.
- Модули. Программы на языке Visual Basic for Applications (VBA). Модули класса. Стандартные модули.
- События. Виды событий. Пошаговое выполнение.
- Импорт и экспорт данных.
- Физическая организация базы данных Общие понятия репликации. Защита баз данных.

Макросы

При работе с базой данных часто приходится снова и снова выполнять одни и те же задачи. Чтобы не повторять одни и те же действия, можно использовать макросы. База данных неуклонно разрастается по мере того, как туда вносятся новые записи в формы, создаются запросы и распечатываются новые отчеты. При этом разрабатываются и сохраняются для последующего использования новые объекты. Приходится регулярно выполнять некоторые задачи. Для выполнения таких регулярных задач удобно использовать макросы Access. После создания этих небольших программ можно запускать их автоматически при определенных действиях пользователя (например, при щелчке на кнопке или при открытии определенной формы). Для автоматического запуска макросов в Access используются так называемые события. Макросы Access позволяют автоматически повторять различные действия и тем самым избавляют от написания сложных программ и подпрограмм. Выполнение прикладной задачи можно представить как результат последовательного выполнения действий, каждое из которых может быть записано соответствующей командой на языках Visual Basic и VBA. Кроме того, с помощью макроса можно имитировать выбор команды меню или перемещение мыши. Макросы Access обычно не дублируют отдельных нажатий клавиш или перемещений мыши, выполняя особые, заданные пользователем задачи, например открытие формы или запуск отчета. Используя макросы для выполнения рутинных операций, в существенно экономит время и силы. Кроме того, поскольку всякий раз при запуске макроса будет осуществляться одна и та же последовательность действий, макрос сделает работу с базой данных более эффективной и аккуратной.

Назначать макросы, зависящие от событий, можно с помощью окна свойств форм и отчетов. Что касается таблиц и запросов, они свойствами событий не обладают. Кроме того, при работе с формами можно запускать макросы с помощью имеющихся в Access панелей инструментов, командных кнопок и меню. Использование этих особенностей позволяет очень легко и эффективно использовать макросы, связанные с событиями.

Макросы можно запускать и на уровне отдельных элементов управления, используя для этого соответствующие события. Например, можно проверить правильность ввода сложных данных непосредственно на уровне поля (а не после выхода этой записи), используя для этого свойство данного поля, а не свойства на уровне всей формы.

Любой макрос может состоять из одной или нескольких макрокоманд. Можно ввести последовательность выполняемых макрокоманд в столбец.

Модули.

Visual Basic for Applications (VBA, Visual Basic для приложений) — немного упрощённая реализация языка программирования Visual Basic, встроенная в линейку продуктов Microsoft Office (включая версии для Mac OS), а также во многие другие программные пакеты, такие как AutoCAD, WordPerfect и ESRI ArcGIS. VBA покрывает и

расширяет функциональность ранее использовавшихся специализированных макро-языков, таких как WordBasic. Может использоваться для управления приложением (меню, панели инструментов, формы пользователя, диалоговые окна и тд.). VBA может также использоваться для создания импорта и экспорта различных форматов файлов.

VBA является интерпретируемым языком. Как и следует из его названия, VBA близок к Visual Basic, но может выполняться лишь в рамках приложения, в которое он встроен. Кроме того, он может использоваться для управления одним приложением из другого, с помощью OLE Automation (например, таким образом можно создать документ Word на основе данных из Excel).

VBA функционально богат и чрезвычайно гибок, но имеет ряд ограничений, включая ограниченную поддержку указателей на функции (используются в Windows API). Имеется возможность использовать (но не создавать) библиотеки DLL.

Классы, как и модули, представляют собой ссылочные типы, инкапсулирующие элементы, определенные в них. Однако они различаются по способу доступа к элементам из других процедур.

Различия между классами и модулями. Основное различие между классами и модулями состоит в том, что классы могут быть инициализированы как объекты, а модули — нет. Существует только одна копия данных стандартного модуля, поэтому при изменении одной частью программы общей переменной в стандартном модуле при последующем чтении этой переменной любая другая часть программы получает измененное значение. Напротив, объект данных существует отдельно для каждого экземпляра объекта. В отличие от стандартных модулей, классы могут реализовывать интерфейсы.

Существует несколько способов обмена данными между приложениями Microsoft Office Access и Microsoft Office Excel.

Импорт и экспорт данных.

Чтобы содержащиеся в Access данные переместить в Excel, можно скопировать данные таблицы Access и вставить их в лист Excel, подключить базу данных Access к листу Excel или экспортировать данные Access в лист Excel.

Чтобы содержащиеся в Excel данные переместить в Access, можно скопировать данные листа Excel и вставить их в таблицу Access, импортировать лист Excel в таблицу Access или связать лист Excel с таблицей Access.

С помощью мастера экспорта Access можно экспортировать в лист Excel объект Access, например, таблицу, запрос, форму или выделенные записи. При выполнении операции экспорта можно сохранить ее параметры для последующего использования, а также можно установить расписание автоматического выполнения экспорта через заданные промежутки времени.

Ниже приведены часто встречающиеся ситуации, в которых нужно экспортировать в Excel данные Access:

Отдел или рабочая группа применяют для обработки данных и Access, и Excel. Данные хранятся в базах данных Access, а анализ данных и рассылка результатов анализа проводятся с помощью Excel. Обычно сотрудники экспортируют данные в Excel по мере необходимости, но нужно сделать этот процесс более эффективным.

Сотрудники привыкли работать с Access, но руководитель предпочитает просматривать отчеты в Excel. Сотрудникам приходится периодически копировать данные в Excel, и нужно автоматизировать этот процесс, чтобы экономить время.

Для получения дополнительных сведений об экспорте данных из Access в Excel см. справочную систему приложения Access.

Защита баз данных.

Защита с использованием пароля БД

Данный способ защиты позволяет установить пароль на открытие БД, для всех пользователей. Для его создания необходимо открыть файл БД в "монопольном" режиме и выбрать пункт меню Сервис / Защита / Задать пароль базы данных. Для работы с такой базой данных в MS Access потребуется вводить пароль. Вот пример работы с файлом БД, используя DAO или ADO.

Шифрование базы данных — это простейший способ защиты. При шифровании базы данных ее файл сжимается и становится недоступным для чтения с помощью служебных программ или текстовых редакторов.

Защита при помощи терминального доступа к серверу

Раздел 5.
Методические указания для лабораторных работ.

Для успешного выполнения лабораторных работ студентам рекомендуется использовать:

1. Лекционный материал по данному курсу.
2. Встроенную справочную систему приложения MS ACCESS.

Модуль 1.
Проектирование базы данных. Создание таблиц.

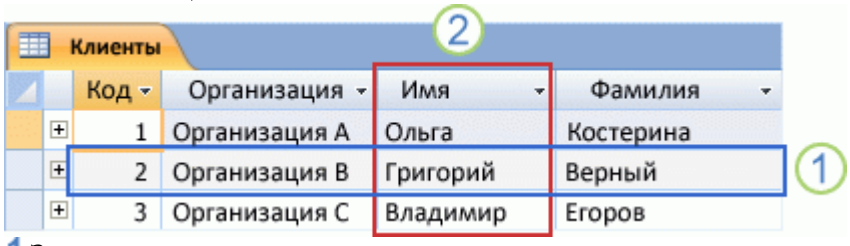
Лабораторных занятий - 9 ч.

Темы практических занятий:

- Microsoft Access. Интерфейс приложения. Создание базы данных «Учебный процесс».
- Создание таблиц БД «Учебный процесс». Типы данных. Свойства полей. Определение свойств полей. Выбор первичного ключа.
- Схема данных. Обеспечение целостности данных.

Что такое таблица?

Таблица содержит данные по определенной теме, например, сведения о сотрудниках или товарах. Каждая запись в таблице включает данные об одном элементе, например о конкретном сотруднике. Запись состоит из полей и включает такие сведения, как имя, адрес и телефонный номер. Кроме того, запись обычно называется строкой, а поле – столбцом.



1 Запись или строка
2 Поле или столбец

База данных может включать множество таблиц, в которых хранятся данные по различным темам. Каждая таблица может состоять из множества полей различного типа, включая текст, числа, даты и рисунки.

В следующем списке приведено несколько типичных примеров таблиц, которые можно создать.

Таблица «Клиенты», в которой перечислены клиенты организации и их адреса.

Каталог продаваемых товаров, включающий цены и изображения каждого из них.

Таблица «Задачи», в которой отслеживаются задачи и сроки их выполнения.

Перечень оборудования или складских запасов.

Создание новой таблицы

В простой базе данных, такой как список контактов, можно использовать всего одну таблицу. Однако во многих базах данных используется несколько таблиц. При создании новой базы данных на компьютере создается новый файл, который используется как контейнер для всех объектов в базе данных, включая таблицы.

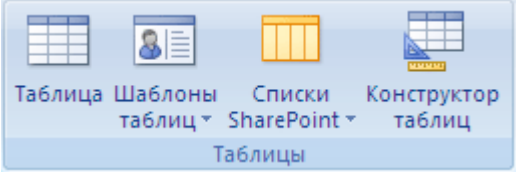
Таблицу можно создать с помощью создания новой базы данных, вставки таблицы в существующую базу данных, а также импорта или создания ссылки на таблицу из другого источника данных, такого как книга Microsoft Office Excel 2007, документ Microsoft Office Word 2007, текстовый файл или другая база данных. При создании новой базы данных в нее автоматически вставляется новая пустая таблица. Затем можно ввести данные, чтобы начать определение полей.

Создание новой таблицы в новой базе данных

Щелкните значок Кнопка Microsoft Office , а затем выберите команду Создать.
В поле Имя файла введите имя файла. Чтобы изменить его расположение, щелкните значок папки для выбора нужной папки.
Нажмите кнопку Создать
Откроется новая база данных, в которой будет создана и открыта в режиме таблицы новая таблица с именем «Таблица1».

Создание новой таблицы в существующей базе данных

Щелкните значок Кнопка Microsoft Office , а затем выберите команду Открыть.
В диалоговом окне Открытие файла базы данных выберите и откройте базу данных.
На вкладке Создание в группе Таблицы щелкните Таблица.

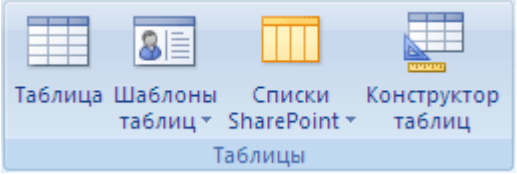


В базу данных вставляется новая таблица, которая открывается в режиме таблицы.

Создание таблицы на основе шаблона таблицы

Для создания таблиц «Контакты», «Задачи», «Вопросы», «События» и «Основные фонды» можно использовать шаблоны таблиц, которые включены в Office Access 2007. Шаблоны таблиц совместимы со списками Службы Microsoft Windows SharePoint Services 3.0 с теми же именами.

Щелкните значок Кнопка Microsoft Office , а затем выберите команду Открыть.
В диалоговом окне Открытие файла базы данных выберите и откройте базу данных.
На вкладке Создание в группе Таблицы щелкните Шаблоны таблицы и затем выберите из списка один из доступных шаблонов.



Будет вставлена новая таблица на основе выбранного шаблона таблицы.

Добавление полей в таблицу в режиме таблицы

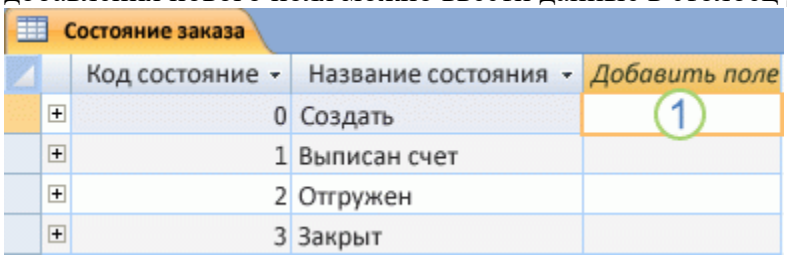
В полях (также называемых столбцами) сохраняются элементы отслеживаемых данных. Например, в таблице «Контакты» можно создать поля «Имя», «Фамилия»,

«Телефон», «Адрес» и т. д. В таблице «Товары» можно создать поле «Наименование товара», «Код товара» и «Цена».

Важно внимательно подойти к выбору полей. Например, чаще всего не не имеет смысла создавать поле для хранения вычисляемого значения. Это значение может рассчитать приложение Office Access 2007, когда это потребуется. При определении полей старайтесь разбивать информацию на наименьшие элементы, которые можно использовать отдельно. Например, полные имена удобно разбить на два поля, «Имя» и «Фамилия». Вообще, если планируется создавать отчеты, выполнять сортировку, поиск или расчет для определенного элемента информации, рекомендуется создать для него собственное поле. Дополнительные сведения о разработке базы данных и выборе полей см. в статье Основные сведения о создании баз данных.

Поля определяются рядом характеристик. Например, у каждого поля есть имя, однозначно определяющее это поле в таблице. Кроме того, у поля есть тип данных, выбираемый в соответствии с сохраняемыми в этом поле данными. Этот тип данных определяет значения, которые можно сохранить, и операции, которые можно выполнить с данными, а также объем памяти, выделяемой для каждого значения. С каждым полем также связана группа параметров, называемых свойствами, которые определяют внешний вид и функциональные характеристики этого поля. Например, свойство «Формат (Format)» определяет структуру отображения данных в поле, то есть вид, в котором они должны отображаться.

Создаваемая новая таблица открывается в режиме таблицы. Для быстрого добавления нового поля можно ввести данные в столбец Добавить поле.



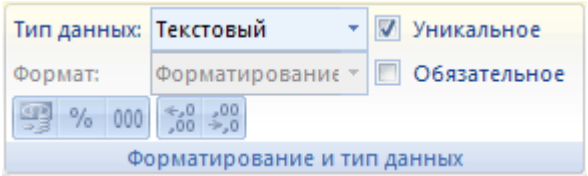
Задание типа данных поля в режиме таблицы

Office Access 2007 может автоматически определять тип данных для поля, созданного в режиме таблицы. Однако иногда требуется переопределить тип данных, назначенный автоматически. Например, продаваемым товарам могут быть присвоены коды для их идентификации. Если ввести числовой код, например 1003, в таблицу в режиме таблицы, средство автоматического определения типа данных задаст числовой тип данных для этого поля. Однако числовой тип данных следует использовать только для полей, участвующих в математических расчетах. Обычно идентификаторы, к которым относится код товара, хранятся в полях с текстовым типом данных.

Чтобы явным образом задать тип данных и формат для поля, переопределив тип, назначенный Office Access 2007, используйте команды в группе Форматирование и тип данных на вкладке Данные.

Явное задание типа данных

На вкладке Режим таблицы в группе Форматирование и тип данных щелкните Тип данных.



Выберите нужный тип данных

Используйте приведенную ниже таблицу для правильного определения типа данных.

Явное задание формата

На вкладке Режим таблицы в группе Форматирование и тип данных щелкните Формат.

Выберите нужный формат.

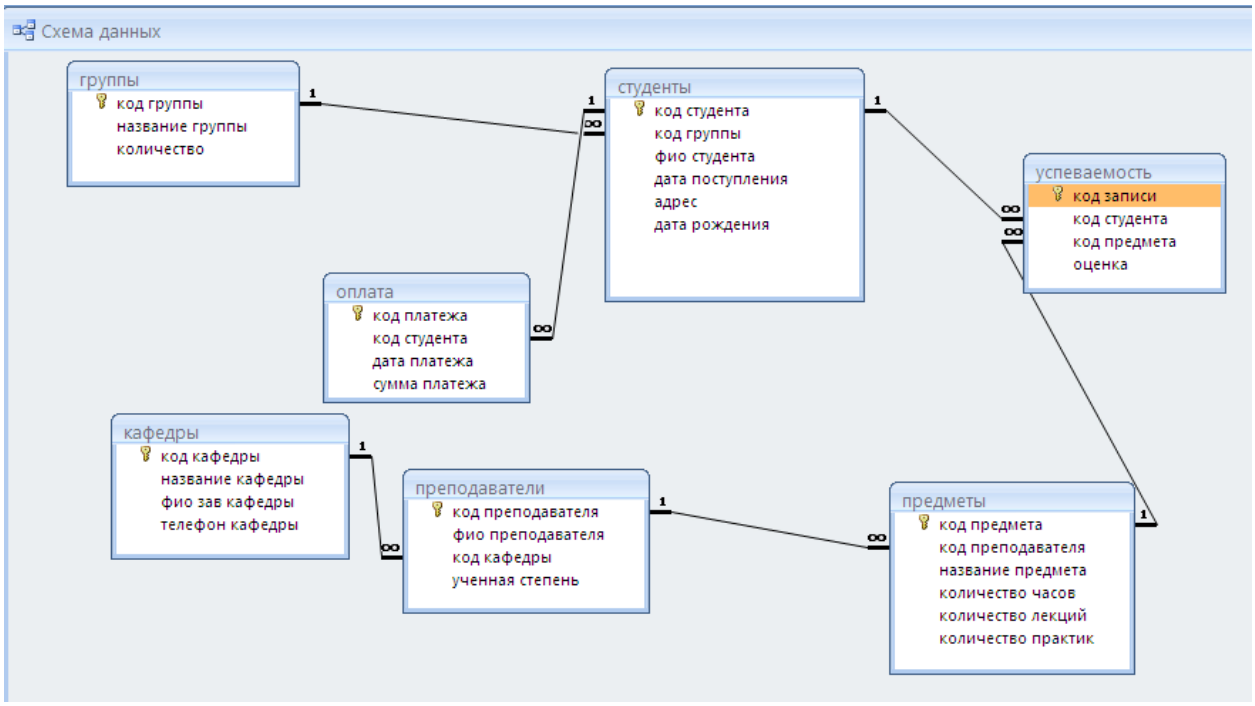
В следующей таблице описаны типы данных, используемые для полей в Office Access 2007.

Тип данных	Хранит	Размер
Текстовый	Алфавитно-цифровые знаки Используется для текста или текста и чисел, не применяемых в расчетах (например, кода товара).	До 255 знаков.
Поле MEMO	Алфавитно-цифровые знаки (более 255 знаков) или форматированный текст. Используется для текста длиннее 255 знаков или форматированного текста. Типичные примеры использования поля типа «Поле MEMO» — примечания, длинные описания и абзацы с полужирным шрифтом или курсивом.	До 1 гигабайта знаков или 2 гигабайт памяти (2 байта на знак), из которых в элементе управления можно отобразить 65 535 знаков.
Числовой	Числовые значения (целые или дробные). Используется для хранения числовых данных, используемых в вычислениях, за исключением денежных значений (для денежных значений используется тип данных «Денежный»).	1, 2, 4 и 8 байт или 16 байт (если используется для кода репликации).
Дата/время	Даты и время. Используется для хранения значений даты и времени. Обратите внимание, что каждое сохраняемое значение содержит одновременно и компонент даты, и компонент времени.	8 байт.
Денежный	Денежные значения. Используется для хранения денежных значений (валюты).	8 байт.
Счетчик	Уникальное числовое значение, которое автоматически вводит Office Access 2007 при добавлении записи. Используется для создания уникальных значений, например для первичного ключа. Обратите внимание, что в поле с типом данных «Счетчик» числа могут последовательно увеличиваться на указанное приращение или выбираться случайно.	4 байта или 16 байт, если используется для кода репликации.
Логический	Логические значения.	1 бит (8 бит = 1 байт).

	Используется для полей, которые могут содержать одно из двух значений, например «Да» и «Нет» или True и False.	
Поле объекта OLE	OLE-объекты или другие двоичные данные. Используется для хранения OLE-объектов других приложений Microsoft Windows.	До 1 Гбайт.
Вложение	Рисунки, изображения, двоичные файлы, файлы Microsoft Office. Стандартный тип данных для сохранения цифровых изображений и любого типа двоичных файлов.	Для сжатых вложений — 2 гигабайта. Для несжатых вложений примерно 700 Кбайт в зависимости от степени возможного сжатия вложения.
Гиперссылка	Гиперссылки. Используется для хранения гиперссылок вызова веб-страниц одним щелчком с помощью URL-адреса или файлов с помощью формата универсального имени UNC. Кроме того, можно использовать ссылку на объекты Access, хранящиеся в базе данных.	До 1 гигабайта знаков или 2 гигабайт памяти (2 байта на знак), из которых в элементе управления можно отобразить 65 535 знаков.
Мастер подстановок	Фактически типом данных не является, а вызывает мастер подстановок. Используется для запуска мастера подстановок, с помощью которого можно создать поле, позволяющее выбрать значение из другой таблицы, запроса или списка значений, используя поле со списком.	На основе таблицы или запроса — размер привязанного столбца. На основе значения — размер текстового поля, содержащего значение.

<u>Задания.</u>

Создать базу данных «учебный процесс» приведенную на схеме.



Модуль 2.

Запросы. Виды запросов.

Лабораторных занятий - 15 ч.

Темы практических занятий:

- Разработка запросов в БД «Учебный процесс». Конструктор запросов. Условие отбора.
- Запросы с группировкой. Использование статистических функций.
- Разработка параметрических и перекрестных запросов.
- Реализация запросов действия.
- Создание запросов SQL.

Запрос на выборку — это объект базы данных, который служит для отображения данных в режиме таблицы. Запрос может получать данные из одной или нескольких таблиц, из существующих запросов или из комбинаций таблиц и запросов. Таблицы или запросы, используемые для получения данных, называются источниками записей.

Создание простых запросов на выборку при использовании мастера и при работе в режиме конструктора происходит одинаково. Для этого следует выбрать источник записей и поля, которые требуется включить в запрос. При необходимости можно задать условия для уточнения результатов запроса.

Чтобы ограничить результаты запроса определенными записями, можно задать одно или несколько условий.

Критерий запроса можно использовать в качестве условия для поля. В условии указываются значения поля, которые требуется включить в запрос, например «отображать только записи, в которых поле «Город» имеет значение «Каракол»».

Откройте запрос в режиме конструктора. В поля запроса добавьте условие, чтобы в результаты включались только записи о контактах, проживающих в Караколе. Для уточнения результатов поиска также можно добавить условия для полей «Адрес» и «Телефон».

В строке Условия строки «Город» введите Каракол.

Затем добавьте еще два условия, чтобы сузить результаты поиска. Предположим, что требуется отобразить только те записи, в которых присутствуют адрес и номер телефона.

В строку Условия поля «Адрес» введите значение Is Not Null AND <>"". Введите то же значение в строке Условия поля «Телефон».

Примечание. Условие Is Not Null AND <>"" является истинным, если поле содержит любое известное не пустое значение. Условие является ложным, если поле не содержит известного значения (Null) или значение заведомо пустое (""). Этим выражением можно проверять поле на наличие известного не пустого значения.

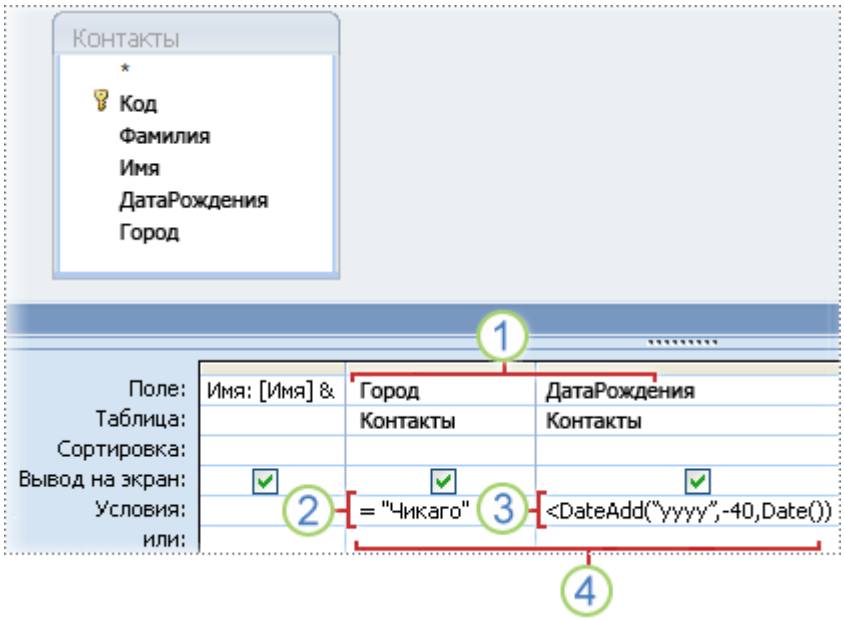
Переключитесь в режим таблицы, чтобы просмотреть результаты.

Здесь приводится список некоторых наиболее часто применяемых условий для типа данных. Если приведенные примеры не соответствуют требованиям, возможно придется составить собственные условия. Предварительно необходимо изучить полный список функций, операторов, специальных знаков и синтаксис выражений, включающих ссылки на поля и текстовые константы.

Чтобы добавить в запрос условие, необходимо открыть запрос в режиме конструктора. Затем следует определить поля, на которые распространяется данное условие. Если в бланке запроса отсутствует нужное поле, его можно добавить самостоятельно, перетащив его из окна конструктора запросов или дважды щелкнув его. (При двойном щелчке поле автоматически добавляется в следующий пустой столбец сетки полей.) Наконец, следует ввести тип условия в строке Условия.

Условия, заданные для различных полей в строке Условия, объединяются с помощью оператора AND. Например, условия, заданные в полях «Город» и «ДатаРождения», интерпретируются следующим образом:

Город = "Чикаго" AND ДатаРождения < DateAdd("yyyy", -40, Date())



1 Поля «Город» и «ДатаРождения» включают условия.

2 Этому условию будут удовлетворять только те записи, в которых поле «Город» принимает значение «Чикаго».

3 Этому условию будут удовлетворять только те записи, в которых возраст человека составляет не менее 40 лет.

4 В результат будут включены только те записи, которые удовлетворяют обоим условиям.

Просуммировать числовой столбец в запросе можно с помощью статистической функции. Статистические функции выполняют вычисления со столбцами данных и возвращают единственное значение. В Microsoft Access существует множество статистических функций, включая Sum, Count, Avg (для вычисления среднего значения), Min и Max. Суммирование данных производится путем добавления в запрос функции Sum, подсчет данных — путем использования функции Count и т.д.

Кроме того, в Office Access 2007 предусмотрено несколько способов добавления функции Sum и других статистических функций в запрос. Существует возможность:

- открыть запрос в режиме таблицы и добавить строку итогов. Строка итогов — новое средство в Office Access 2007, позволяющее использовать статистические функции в одном или нескольких столбцах в результатах запроса без необходимости изменять структуру запроса;
- создать итоговый запрос. Итоговый запрос вычисляет промежуточные итоги для групп записей, строка итогов — общие итоги для одного или нескольких столбцов (полей) данных. Например, при необходимости вычисления промежуточных сумм по всем продажам по городам или по кварталам, следует использовать итоговый запрос для группировки записей по нужной категории, а затем просуммировать все объемы продаж;
- создать перекрестный запрос. Перекрестный запрос — это особый тип запросов, отображающий результаты в сетке, по виду схожей с листом Microsoft Office Excel 2007. Перекрестные запросы суммируют значения и затем группируют их по двум наборам фактических данных — один набор вдоль боковой стороны (заголовки строк) и второй набор вдоль верхней части (заголовки столбцов).

Задания.

1. Отобрать всех студентов, фамилии которых начинаются на букву К.
2. Отобрать всех студентов отличников.
3. Отобрать всех студентов, родившихся в зимние месяцы 1990г.
4. Вывести фамилии студентов, внесших оплату в течении октября 2010 года.
5. Отобрать информацию по конкретному преподавателю (ФИО преподавателя, название кафедры, название предмета, количество часов). ФИО преподавателя вводится пользователем в окне параметров.
6. Вывести список преподавателей, имеющих степень кандидата или доктора наук.
7. Отобрать информацию: ФИО, название группы, дата рождения, адрес, телефон, успеваемость; по конкретному студенту. ФИО вводится пользователем в окне параметров.
8. Создать зачетную книжку (предмет, количество часов, ФИО преподавателя, оценка).
9. Вывести студентов и их среднюю успеваемость.
10. Создать перекрестный запрос - строки – ФИО студента, столбцы- предметы, на пересечении –оценки.
11. Создать перекрестный запрос - строки –ФИО преподавателя, столбцы- название предмета, на пересечении –количество часов.
12. Создать запрос с условием в вычисляемом поле. Необходимо вывести всех студентов, их успеваемость (средняя оценка по всем предметам). А также их статус:
    - двоечник (средняя оценка <2.7);
    - троечник (средняя оценка >=2.7 И <3.5);
    - ударник (средняя оценка >=3.5 И <4.5);
    - отличник (средняя оценка >=4.5).

13. Создать запрос с вычисляемым полем. Необходимо перевести все суммы платежа в сомы.
14. Отобрать всех студентов (фио, группа), которые внесли оплату в диапазоне от 200 до 300 $.
15. Вывести информацию о сумме оплаты, внесенной студентами каждой группы и наименование группы. Например, студенты группы МКС-1 внесли 45620 сом, студенты группы МКС-2 внесли 4566 сом…..
16. Создать запрос на обновление. Преподавателю присвоили новую ученую степень. Необходимо обновить данные. (Например, Сидоров был старшим преподавателем, стал доцентом).
17. Создать запрос на добавление. Из колледжа на 3-ий курс перевелись 5 студентов в определенную группу, информация по этим студентам хранится в новой таблице Студенты1. Необходимо добавить их в таблицу студенты.
18. Создать запрос на удаление. Какая-то группа была расформирована по некоторым причинам. Необходимо удалить ее.
19. Создать запрос, который вычисляет стоимость за обучение для отличников и ударников со скидкой (для отличников – скидка 40 %, для ударников – скидка 20 %) .
20. Создать запрос с вычисляемым полем, позволяющий посчитать возраст студента.

Модуль 3.

Создание интерфейса пользователя. Разработка форм и отчетов.

Лабораторных занятий - 12 ч.

Темы практических занятий:

- Разработка форм и отчетов в БД (Тема выбирается в зависимости от варианта). Простые и сложные формы.
- Создание многотабличных форм. Создание форм помощи.
- Отчеты. Отчеты с группировкой и сортировкой.
- Создание кнопочной формы. Разделы кнопочной формы.

Форма — это объект базы данных, который можно использовать для ввода, изменения или отображения данных из таблицы или запроса. Формы могут применяться для управления доступом к данным: с их помощью можно определять, какие поля или строки данных будут отображаться. Например, некоторым пользователям достаточно видеть лишь несколько полей большой таблицы. Если предоставить им форму, содержащую только нужные им поля, это облегчит для них использование базы данных. Для автоматизации часто выполняемых действий в форму можно добавить кнопки и другие функциональные элементы.

Доработать форму можно также в режиме конструктора. Можно добавить в форму новые элементы управления и поля, разместив их на сетке макета. Окно свойств позволяет посредством изменения множества свойств формы настроить ее внешний вид.

Чтобы переключиться в режим конструктора, щелкните правой кнопкой мыши имя формы в области переходов и выберите команду Конструктор .

Форма будет отображена в режиме конструктора.

Изменить свойства формы, ее разделов и входящих в нее элементов управления можно с помощью окна свойств. Для отображения окна свойств нажмите клавишу F4.

Из области Список полей можно добавить в макет формы поля из базовой таблицы или базового запроса. Для отображения области Список полей выполните одно из следующих действий.

На вкладке Форматирование в группе Элементы управления выберите команду Добавить существующие поля.

Нажмите сочетание клавиш ALT+F8.

Можно перетащить поля непосредственно из области Список полей в форму.

Чтобы добавить одно поле, дважды щелкните его или перетащите его из области Список полей в тот раздел формы, где оно должно отображаться.

Чтобы добавить сразу несколько полей, щелкните их последовательно, удерживая нажатой клавишу CTRL. Затем перетащите выбранные поля в форму.

Чтобы добавить одно поле, дважды щелкните его или перетащите его из области Список полей в тот раздел формы, где оно должно отображаться.

Чтобы добавить сразу несколько полей, щелкните их, удерживая нажатой клавишу CTRL. Затем перетащите выбранные поля в форму.

Программа Microsoft Office Access 2007 помогает создавать множество различных отчетов любой степени сложности. Прежде чем приступать к работе, следует выбрать источник, из которого будут извлекаться записи отчета. Отчет может представлять собой как простой список, так и подробную сводку данных о продажах, сгруппированных по районам. Однако в любом случае необходимо сначала определить, в каких полях содержатся данные, которые должны войти в отчет, и в каких таблицах или запросах находятся эти поля.

В программе Access макет отчета разбит на разделы. Разделы можно просматривать в режиме конструктора. Чтобы правильно создавать отчеты, необходимо понимать назначение каждого раздела. Например, от выбора раздела, в который будет помещен вычисляемый элемент управления, зависит способ вычисления результата. Ниже перечислены типы разделов и указано назначение каждого из них.

Заголовок отчета Печатается только один раз в начале отчета. В заголовок включается информация, обычно помещаемая на обложке, например эмблема компании, название отчета или дата. Если в заголовке отчета помещен вычисляемый элемент управления, использующий статистическую функцию Sum, сумма рассчитывается для всего отчета. Заголовок отчета печатается перед верхним колонтитулом.

Верхний колонтитул Печатается вверху каждой страницы. Верхний колонтитул используется, например, в тех случаях, когда нужно, чтобы название отчета повторялось на каждой странице.

Заголовок группы Печатается перед каждой новой группой записей. Используется для печати названия группы. Например, если отчет сгруппирован по изделиям, в заголовках групп можно указать их названия. Если поместить в заголовок группы вычисляемый элемент управления, использующий статистическую функцию Sum, сумма будет рассчитываться для текущей группы.

Область данных Этот раздел печатается один раз для каждой строки данных из источника записей. В нем размещаются элементы управления, составляющие основное содержание отчета.

Примечание группы Печатается в конце каждой группы записей. Примечание группы можно использовать для печати сводной информации по группе.

Нижний колонтитул Печатается внизу каждой страницы. Используется для нумерации страниц и для печати постраничной информации.

Примечание отчета Печатается один раз в конце отчета. Примечание отчета можно использовать для печати итогов и другой сводной информации по всему отчету.

Доработать отчет можно также в режиме конструктора. Новые элементы управления и поля добавляются в отчет путем их размещения их в сетку конструктора макета. В окне свойств доступны многочисленные параметры, с помощью которых можно настроить отчет.

Чтобы переключиться в режим конструктора, щелкните правой кнопкой мыши имя отчета в области переходов и выберите команду Конструктор .

Отчет будет открыт в режиме конструктора.

Изменить свойства отчета, его разделов и входящих в него элементов управления можно в окне свойств. Чтобы отобразить окно свойств, нажмите клавишу F4.

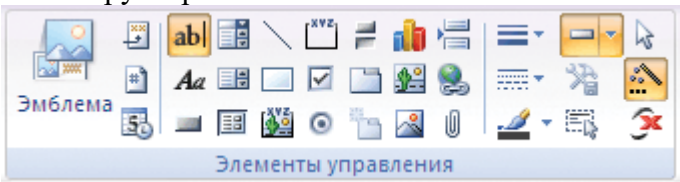
В макет отчета можно добавить поля базовой таблицы или базового запроса из области Список полей. Чтобы отобразить область Список полей, выполните одно из следующих действий.

На вкладке Форматирование в группе Элементы управления выберите команду Добавить существующие поля.

Нажмите клавиши ALT+F8.

После этого можно будет добавлять поля в отчет, перетаскивая их из области Список полей.

Некоторые элементы управления создаются автоматически. Например, присоединенный элемент управления «текстовое поле» появляется при добавлении в отчет поля из области Список полей. В режиме конструктора можно создать множество других элементов управления при помощи инструментов группы Элементы управления на вкладке Конструктор.



Создание кнопочной формы

1. На вкладке Работа с базами данных в группе Работа с базами данных нажмите кнопку Диспетчер кнопочных форм.
   Примечание. Если выводится приглашение создать кнопочную форму, нажмите кнопку Да.
2. Нажмите кнопку Создать.
   На экране появится диалоговое окно Создание.
3. Введите имя новой кнопочной формы и нажмите кнопку ОК.
   Имя новой кнопочной формы добавляется в список поля Страницы кнопочной формы.
4. Выберите имя новой кнопочной формы и нажмите кнопку Изменить.
   Будет открыто диалоговое окно Изменение страницы кнопочной формы.
5. Нажмите кнопку Создать.
   Откройте диалоговое окно Изменение элемента кнопочной формы.
6. В поле Текст введите имя для первой кнопки кнопочной формы, а затем выберите для нее команду из списка Команда.
   Например введите текст Сведения о покупателях в поле Текст, а затем выберите Открыть форму для добавления из списка Команда.
   Примечание. Чтобы создать кнопочную форму, открывающую другие кнопочные формы, выберите в поле Команда команду Перейти к кнопочной форме, а затем выберите кнопочную форму из списка или введите имя другой кнопочной формы.
7. В зависимости от выбранной команды в приложении Access может быть выведен еще один список под списком Команда. Например, если выбрана команда Открыть отчет, в открытой базе данных появляется список отчетов. Если требуется, выберите элемент из списка.

8. Повторяйте шаги 5 и 7 до тех пор, пока не будут добавлены все элементы кнопочной формы. Чтобы завершить создание кнопочной формы, нажмите кнопку Закрыть.

Задания.

1. Создать форму для просмотра и ввода информации в таблицу «кафедры».
2. Создать форму для просмотра и ввода информации в таблицу «группы».
3. Создать форму для ввода информации в таблицу «преподаватели» (для поля код кафедры использовать элемент поле со списком).
4. Создать форму для ввода информации в таблицу «студенты» (для поля код группы использовать элемент поле со списком, для поля код вида обучения - группу переключателей).
5. Создать многотабличную форму для просмотра студентов по группам.
6. Создать многотабличную форму для просмотра преподавателей по кафедрам.
7. Создать форму для ввода информации в таблицу «предметы» (для поля код преподавателя использовать элемент поле со списком).
8. Создать форму для заполнения таблицы "Успеваемость».
9. Создать форму для заполнения таблицы "Оплата".
10. Создать форму для отображения зачетной книжки. Форма содержит: ФИО студента, предметы, кол-во часов, ФИО преподавателя, оценки, оценки прописью, кнопки навигации
11. Создать форму позволяющую вывести студентов по группам, их успеваемость и статус, а также средний бал успеваемости по группам.
12. Создать отчет о преподавателях с группировкой по кафедрам.
13. Создать отчет о студентах по группам на базе параметрического запроса. Т.е. пользователь вводит название группы в окне параметров. Отчет содержит: Название группы (в заголовке отчета), ФИО студента и его данные.
14. Создать отчет об успеваемости студентов. Вывести ФИО студента, название предмета, оценку. Произвести группировку по ФИО студента и итоговое поле по оценке (посчитать среднюю успеваемость).
15. Создать кнопочную форму
16. Создать пять макросов.
17. Реализовать все способы запуска макроса.

Модуль 4.

Программирование на языке макросов. Программирование на языке VBA.

Лабораторных занятий - 15 ч.

Темы практических занятий:

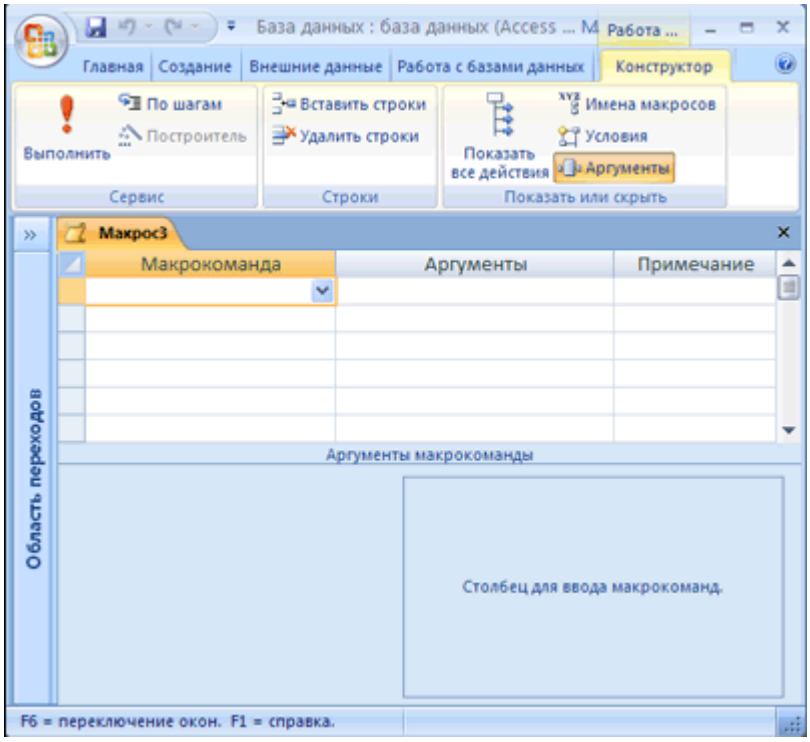
- Конструктор макросов. Создание макрокоманд. Запуск макросов. Способы запуска макросов. Создание макросов в БД «Учебный процесс».
- Преобразование макросов в модули. Редактор Microsoft Visual Basic. Окно модуля.
- Вызов подпрограммы VBA из макроса. Запуск макроса из подпрограммы VBA.
- Импорт данных. Импорт таблиц. Связывание данных. Экспорт.
- Защита данных. Пароль. Рабочая группа. Ограничение доступа

Макрос — это средство для автоматизации задач и добавления функциональных возможностей в формы, отчеты и элементы управления. Например, при добавлении

командной кнопки в форму событие кнопки OnClick связывается с макросом, который содержит команды, выполняемые при каждом нажатии кнопки.

В приложении Access макросы можно рассматривать как упрощенный язык программирования, на котором программа записывается в виде списка макрокоманд для выполнения. При создании макроса каждая макрокоманда выбирается из раскрывающегося списка, после чего к ней добавляется необходимая информация. Макросы позволяют добавлять функциональные возможности в формы, отчеты и элементы управления без необходимости написания кода в модуле Visual Basic для приложений (VBA). Макросы обеспечивают выполнение части команд, доступных в VBA, и для большинства пользователей создание макроса оказывается проще, нежели написание кода VBA.

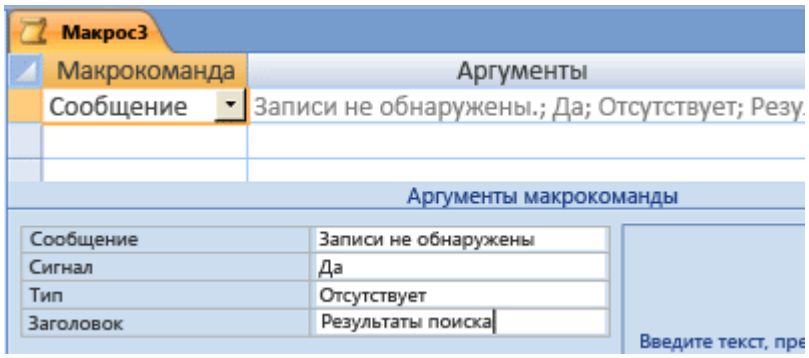
Макрос создается с помощью построителя макросов, показанного на следующем рисунке.



Чтобы отобразить построитель макросов

На вкладке Создание в группе Другие щелкните Макрос. Если эта команда недоступна, щелкните стрелку либо под кнопкой Модуль, либо под кнопкой Модуль класса, а затем щелкните Макрос.

Аргумент — это значение, которое обеспечивает необходимую для макрокоманды информацию, например, какая строка должна отображаться в окне сообщения, с каким элементом управления следует выполнять действия и т. п. Некоторые аргументы являются обязательными, другие — нет. Аргументы отображаются в области Аргументы макрокоманды в нижней части окна построителя макросов.



В построителе макросов в Office Access 2007 имеется столбец Аргументы, позволяющий просматривать (но не изменять) аргументы макрокоманды в строке макрокоманды. Так легче читать макрос, так как уже не нужно выбирать каждую макрокоманду, чтобы увидеть ее аргументы. Чтобы отобразить столбец Аргументы, нажмите кнопку Аргументы в группе Отображение на вкладке Конструктор.

Для непосредственного запуска макроса выберите один из следующих вариантов.

Найдите макрос в области переходов и дважды щелкните его имя.

На вкладке Инструменты базы данных в группе Макрос нажмите кнопку Выполнить макрос, выберите макрос в списке Имя макроса и нажмите кнопку ОК.

Если макрос открыт в режиме конструктора, нажмите кнопку Выполнить на вкладке Конструктор в группе Инструменты. Чтобы открыть макрос в режиме конструктора, щелкните его правой кнопкой мыши в области переходов и нажмите кнопку Конструктор.

<u>Задания.</u>

1. Реализовать окно аутентификации пользователя, с использованием макросов (для одного имени и пароля). (База данных «Учебный процесс»)
2. Создать форму со списком студентов, при нажатии на кнопку, возле фамилии студента, открывается форма с зачетной книжкой данного студента.
3. Создать форму со списком преподавателей, при нажатии на кнопку возле фамилии преподавателя открывается форма с информацией о предметах, которые данный преподаватель преподает.
4. Создать форму с наименованием групп, при нажатии на кнопку возле названия группы открывается форма с информацией о студентах, обучающихся в выбранной группе.
5. Создать форму. В форме разместить два текстовых поля и одну кнопку. При нажатии на кнопку определить (и вывести во второе текстовое поле) является ли число, введенное в первое текстовое поле, совершенным. Число называется совершенным, если оно равно сумме всех своих делителей за исключением его самого.
6. Реализовать окно аутентификации пользователя с использованием языка VBA. Пароли и имена хранятся в таблице базы данных. (База данных «Учебный процесс»)

Раздел 6.
Методические рекомендации по СРС.

Студентам необходимо самостоятельно выполнить один из проектов, описанных ниже.

Требования к выполнению проекта и представлению результатов

Основными частями проектной работы являются:

1. База данных. Это основная часть проекта. Исходя из описания предметной области и требований к базе данных, должны быть последовательно разработаны концептуальная, логическая и физическая модель базы данных. Затем проект базы данных реализуется в виде приложения *MS Access*.

2. Электронные таблицы. Для решения ряда задач необходимо провести дополнительный анализ данных с помощью сводных таблиц, расчетов трендов и других возможностей *MS Excel*.

3. Компьютерная презентация. Необходимо представить обзор проекта в виде презентации *MS Power Point*.

4. Отчет по проекту. Проект должен быть описан в отчете в виде документа *MS Word*.

Требования к базе данных

База данных должна быть создана в *MS Access* в соответствии с проектом структуры базы данных. Должны быть заданы первичные ключи, поля подстановки, установлена ссылочная целостность базы данных.

Все данные должны вводиться через экранные формы, объединенные главной кнопочной формой.

Должны быть построены следующие виды запросов:

- запросы на выборку, включая параметрические;
- запросы на выборку с группировкой;
- созданы итоговые и текущие отчеты по таблицам и запросам.

Требования к анализу данных в Excel

Некоторые таблицы и результаты запросов следует экспортировать в файл *Excel* для дополнительного анализа. В качестве примера можно привести таблицу-результат запроса о всех заказах за прошлый год вместе с информацией по поставщикам, покупателям и сотрудникам.

По экспортированным в *Excel* таблицам можно построить сводные таблицы, таблицы с промежуточными итогами и различные диаграммы.

Могут быть проведены дополнительные статистические расчеты, например, частота заказов в различные города, прогнозы продаж, линейные тренды продаж и т. п.

Требования к презентации

В структуре презентации должны содержаться следующие слайды:

1. Титульный слайд презентации.
2. Обзор базы данных.
3. Проект базы данных.
4. Концептуальная модель.
5. Логическая модель.
6. Структура таблиц в *MS Access*.
7. Первичные ключи и индексы.
8. Схема данных.
9. Получение информации.
10. Запросы на выборку.
11. Запросы с группировкой.
12. Параметрические запросы.

13. Стандартные отчеты.
14. Анализ данных.

Требования к отчету

1. Отчет следует представить как в виде *Word* файла.
2. Основная часть отчета:

   Титульный лист
   Содержание
   Введение
   Проект базы данных
   - Концептуальная модель;
   - Логическая модель;
   - Физическая модель.

   Реализация в MS Access
   - Структура таблиц, ключи и индексы;
   - Связи между таблицами;
   - Основные запросы;
   - Стандартные отчеты;
   - Описание пользовательского интерфейса (формы).

   Анализ данных
   - Примеры анализа в Excel;
   - Диаграммы.

   Выводы
   Руководство пользователя
   - Как запустить приложение;
   - Назначение основных форм;
   - Как закончить.
3. Содержание титульного листа:

   Учебное заведение, факультет
   Курс " "
   Название проекта («Разработка базы данных …»)
   Автор, группа:
   Руководитель:
   Бишкек 201..

Все заголовки разделов и подразделов должны быть выполнены стилями заголовков. Для основного текста должны быть созданы собственные стили. На каждой странице, кроме первой, должны присутствовать колонтитулы с названием работы и фамилией автора.

Варианты заданий:

1. База данных магазина компьютерных комплектующих.
2. База данных для туристической фирмы.
3. База данных для частной нотариальной конторы.
4. База данных для страховой компании.
5. База данных «Ломбард».
6. База данных «Магазин игрушек»
7. База данных «Магазин канцелярских товаров»
8. База данных «Евровидение»
9. База данных «Поликлиника»
10. База данных «Аптека»
11. База данных «Магазина мобильных телефонов»
12. База данных «Магазина ювелирных изделий»
13. База данных «Университет»
14. База данных «Спортивный клуб»
15. База данных «Фонотека»
16. База данных «Книжный магазин»
17. База данных «Библиотека»
18. База данных аренда помещения
19. База данных обслуживания склада
20. База данных склада автозапчастей
21. База данных сети музыкальных магазинов
22. База данных по кулинарным рецептам.
23. База данных «Агентство недвижимости».
24. База данных «Брачное агентство».
25. База данных «Биржа труда»
26. База данных «Выпускники МУК».

Раздел 7.
Самостоятельная работа студента под руководством преподавателя.

Рекомендуемая литература:

1. Кузнецов А. «Microsoft Access 2003. Русская версия. Учебный курс» - СПб.: БХВ-Питер. 2006
2. Стив Ламберт: Microsoft Office Access 2007. Эком, 2007 г.
3. Элисон Балтер Microsoft Office Access 2007: профессиональное программирование..; Вильямс. 2007
4. Справочная система Microsoft Access.

Модуль 1.

Проектирование базы данных. Создание таблиц.

Самостоятельная работа с преподавателем – 5 ч.

Самостоятельная работа с преподавателем.

- Модели данных (иерархическая, сетевая, реляционная).
- Обзор современных СУБД.
- Уровни представления баз данных.
- Декомпозиция отношений, транзитивные зависимости.
- Метод: сущность – связь.

Задания.

Разработать и реализовать базу данных для магазина, как минимум на основе пяти таблиц.

Модуль 2.

Запросы. Виды запросов.

Самостоятельная работа с преподавателем – 7 ч.

Самостоятельная работа с преподавателем.

- Реляционная алгебра и язык SQL.
- Мастер запросов. Окно запроса в режиме конструктора.
- Запросы с вычисляемыми полями.
- Условие iif в вычисляемых полях.

Задания.

1. Отобрать всех продавцов, фамилия которых начинается на букву «Н».
2. Создать запрос позволяющий вывести продавца, название проданного товара, цену товара и дату продажи.
3. Создать запрос позволяющий вывести товары одной категории (заданной в окне параметров).
4. Вывести список товаров, поставляемых только одним из поставщиков (указанным в окне параметров).
5. Создать запрос на обновление. Продавец поменял место жительство, необходимо обновить адрес продавца в БД.
6. Создать запрос на удаление. Один из продавцов был уволен, необходимо удалить его из БД.
7. Создать запрос, позволяющий посчитать зарплату каждого продавца, Зарплата составляет 10% от общей выручки продавца.
8. Создать запрос, позволяющий посчитать премиальные для каждого продавца (50% от зарплаты), если его общая выручка больше 5000 сом.
9. Создать перекрестный запрос - строки –товары, столбцы- ФИО продавца, строки – цена, на пересечении –количество товара.

10. Создать запрос, определяющий, сколько месяцев лежит товар в магазине.
11. Определить статистику прибыли магазина в процентном соотношении, по дням недели.

Модуль 3.

Создание интерфейса пользователя. Разработка форм и отчетов.

Самостоятельная работа с преподавателем – 5 ч.

Самостоятельная работа с преподавателем.

- Элементы управления. Панель элементов.
- Главная и подчиненная форма.
- Использование построителя выражений.
- Способы изменения свойств форм и свойств элементов форм.
- Отчеты. Панель элементов. Окно конструктора отчетов.

Задания.

1. Создать форму для просмотра и ввода информации в таблицу «продавцы».
2. Создать форму для просмотра и ввода информации в таблицу «категории».
3. Создать форму для просмотра и ввода информации в таблицу «фирма-поставщик».
4. Создать форму для просмотра и ввода информации в таблицу «товары».
5. Создать форму для просмотра и ввода информации в таблицу «движение товаров».
6. Создать многотабличную форму для просмотра товаров по категориям.
7. Создать многотабличную форму для просмотра товаров по поставщикам.
8. Создать отчет к с группировкой по названию поставщика. Подвести итоги (сумма, количество)
9. Создать отчет с группировкой по категории товара. Подвести итоги (сумма, количество).
10. Создать годовой отчет (о доходах). Произвести группировку по ФИО продавца.
11. Создать отчет с двумя уровнями группировки с подведением итогов

Модуль 4.

Программирование на языке макросов. Программирование на языке VBA.

Самостоятельная работа с преподавателем – 7 ч.

Самостоятельная работа с преподавателем.

- Использование макросов для автоматизации приложения.
- Безопасность макросов.
- Элементы программы.
- Переменные, константы, массивы.
- Процедуры и функции.
- Назначение и удаление прав доступа.
- Последовательность событий. Отладка программ. Окна отладки.

Задания.

1. Создать кнопочную форму. (База данных «Магазин»)
2. Реализовать окно аутентификации пользователя, с использованием макросов (для одного имени и пароля). (База данных «Магазин»)
3. Создать форму. В форме разместить два текстовых поля и одну кнопку. При нажатии на кнопку определить (и вывести во второе текстовое поле) является ли число, введенное в первое текстовое поле, совершенным. Число называется совершенным, если оно равно сумме всех своих делителей за исключением его самого.
4. Реализовать окно аутентификации пользователя с использованием языка VBA.
5. Реализовать окно аутентификации пользователя с использованием языка VBA. Пароли и имена хранятся в таблице базы данных. (База данных «Магазин»)

Раздел 8.
Контрольно-измерительные средства.

Контрольные вопросы по дисциплине «Банки данных».

1. СУБД Microsoft Access.
2. Базы данных. Модели баз данных. Реляционные базы данных.
3. Проектирование базы данных. Этапы разработки базы данных.
4. Таблицы. Ключи. Связи между таблицами.
5. Ограничения накладываемые на связанные таблицы.
6. Таблицы. Типы данных.
7. Таблицы. Свойства полей.
8. Запросы. Виды запросов.
9. Запросы. Параметрический и перекрестный запросы.
10. Запросы. Запросы действий.
11. Запросы. Запросы с группировкой, с вычисляемыми полями, с условием в вычисляемых полях.
12. Запросы SQL. Основные инструкции языка SQL.
13. Запросы. Ошибки при выполнении запросов действий.
14. Формы. Разделы формы.
15. Формы. Свойства формы.
16. Формы. Элементы управления.
17. Формы. Сложные формы.
18. Формы. Вычисления в формах.
19. Формы. Рекомендации по оформлению.
20. Отчеты. Разделы отчета.
21. Отчеты. Составные отчеты.
22. Создание страниц доступа к данным.
23. Страницы доступа к данным. Элементы управления.
24. Макросы. Использование макросов в Access.
25. Макросы. Основные макрокоманды.
26. Макросы. Конструктор макросов.
27. Макросы. Условия выполнения макросов.
28. Макросы. Группы макросов.
29. Макросы. Способы запуска макроса.
30. Использование макросов для автоматизации приложения.
31. Ссылки на объекты и элементы управления.
32. Макросы. Отладка макросов.
33. Макросы. Безопасность макросов.
34. Модули. Типы модулей.
35. Модули. Элементы программы.
36. Переменные, константы, массивы.
37. Типы данных в VBA. Специальные символы типов данных.
38. Процедуры и функции.
39. Управление выполнением программы.
40. Защита данных.

Тестовые вопросы для самоконтроля

1. Как называются столбцы, значения в которых изначально должны быть различными для разных строк этой таблицы?
    a) внешний ключ

b) индексный ключ
c) первичный ключ

2. Как называются столбцы, которые не являются первичным ключом таблицы, но представляют первичный ключ другой таблицы?
   a) вторичный ключ
   b) внешний ключ
   c) индексный ключ
3. Какая модель базы данных определяется, как набор таблиц с уникальными именами?
   a) объектная модель
   b) реляционная модель
   c) иерархическая модель
4. Укажите тип данных в Access, который должен содержать уникальное значение.
   a) поле OLE
   b) счетчик
   c) поле MEMO
5. Укажите тип данных в Access, который может содержать тест более чем 255 символов.
   a) текстовый
   b) поле MEMO
   c) поле OLE
6. Какой символ шаблона, при использовании маски ввода, означает любое число произвольных символов?
   a) ?
   b) #
   c) *
7. Что такое «связь между таблицами»?
   a) определение типа отношения между таблицами
   b) определение индексов в нескольких таблицах
   c) передача данных из одной таблицы в другую
8. Как называется форма, которая встраивается в другую форму?
   a) вложенная форма
   b) встроенная форма
   c) подчиненная форма
9. Вы хотите, чтобы при открытии базы данных на экране автоматически отображалась заданная форма. Можно ли это сделать и как, если можно?
   a) да, если дать команду Сервис | Параметры запуска (Tools | Startup)
   b) да, если дать команду Файл | Свойства базы данных (File | Properties of Database)
   c) да, если дать команду Сервис | Параметры (Tools | Options)
10. Вы поместили в форму командную кнопку для запуска макроса. Как нужно модифицировать свойства кнопки, чтобы она запускала макрос?
    a) для свойства Нажатие кнопки (OnClick) указать имя макроса
    b) для свойства До обновления (BeforeUpdate) указать имя макроса
    c) имя кнопки должно быть таким же, как имя макроса
11. Какое свойство поля таблицы должно использоваться, чтобы при добавлении новой записи в таблицу в ячейке автоматически устанавливалось заданное значение?
    a) обязательное поле (Required)
    b) значение по умолчанию (Default Value)
    c) условие на значение (Validation Rule)
12. Что можно использовать в качестве источника данных для отчета?

a) запросы
b) таблицы
c) комбинацию таблиц и запросов

13. Что такое однопользовательская система (single-user system)?
    a) система, с которой одновременно может работать только один человек
    b) система с высоким уровнем защиты, с которой может работать только один авторизованный человек
    c) сетевая система, которая не различает своих пользователей по именам
14. Что понимают под термином "общие данные"?
    a) данные, содержащиеся в нескольких таблицах одновременно
    b) данные, не содержащие конкретных значений параметров
    c) возможность использования отдельных областей данных, несколькими разными пользователями
15. Что такое SQL?
    a) язык разметки базы данных
    b) язык программирования высокого уровня
    c) структурированный язык запросов
16. Какая база данных строится на основе таблиц и только таблиц?
    a) реляционная
    b) иерархическая
    c) сетевая
17. Какой из ниже перечисленных элементов не является объектом MS Access?
    a) запрос
    b) книга
    c) макрос
18. В какой модели баз данных существуют горизонтальные и вертикальные связи между элементами?
    a) иерархической
    b) реляционной
    c) сетевой
19. Из чего состоит макрос?
    a) из набора макрокоманд
    b) из набора гиперссылок
    c) из совокупности операторов Visual Basic
20. Какого раздела не существует в конструкторе форм?
    a) раздел заголовка
    b) итоговый раздел
    c) верхний колонтитул
21. Какие типы полей записей в СУБД Access допустимы?
    a) числовой, текстовый, гипертекстовый, логический
    b) числовой, символьный, графический музыкальный
    c) логический , дата, числовой, денежный, OLE
22. Как называется структура данных, для которой характерна подчиненность объектов нижнего уровня объектам верхнего уровня?
    a) реляционной
    b) табличной
    c) иерархической
23. Как именуются поля реляционной базы данных?
    a) именуются пользователем произвольно с определенными ограничениями
    b) именуются по правилам, специфичным для каждой конкретной системы управления базами данных
    c) нумеруются по правилам, верным для любой конкретной СУБД

24. Что может быть записано в поле файла реляционной базы данных (БД)?
    a) только номера записей
    b) данные только одного типа
    c) как числовые, так и текстовые данные одновременно
25. Укажите наиболее точный аналог реляционной базы данных.
    a) двумерная таблица
    b) генеалогическое дерево
    c) неупорядоченное множество данных
26. Какое свойство отсутствует у полей с текстовым типом данных?
    a) формат поля
    b) значение по умолчанию
    c) индексированное поле
27. Что позволяет свойство «Значение по умолчанию»?
    a) заполнить каждую запись указанного поля, каким либо значением без последующего изменения информации по умолчанию
    b) заполнить каждую запись конкретного поля указанным значением с возможностью откорректировать это значение
    c) заполнить только некоторые записи указанным значением
28. Для чего служит база данных?
    a) хранения и упорядочения информации
    b) ведения расчетно-вычислительных операций
    c) обработки текстовой документации
29. В чем измеряется длина поля?
    a) в битах
    b) в байтах
    c) в символах
30. Что считается записью?
    a) столбец
    b) таблица
    c) строка
31. Сколько символов вмещает поле с текстовым типом данных?
    a) 20 символов
    b) 256 символов
    c) 65536 символов
32. Какое из условий отбора в запросе для поля, имеющего тип данных «Дата/Время», является верным?
    a) <= *20.05.2005*
    b) <= #20.05.2005#
    c) <= "20.05.2005"
33. Укажите модель данных, в которой каждый элемент может быть связан с любым другим элементом, а так же может быть и главным, и подчиненным по отношению к другим элементам.
    a) реляционная
    b) иерархическая
    c) сетевая
34. Что обновляется в результате каскадного обновления?
    a) содержимое связанных полей
    b) имя столбца
    c) дата
35. Если ключевое поле создается автоматически, какой тип данных ему соответствует?
    a) текстовый

b) логический
c) счетчик

36. Какое предложение используется для указания источника данных в запросе?
    a) where
    b) from
    c) load
37. Какая команда используется в запросе для группировки данных?
    a) sort by
    b) group by
    c) order by
38. Для чего предназначены запросы?
    a) для хранения данных базы
    b) для отбора и обработки данных базы
    c) для автоматического выполнения группы команд
39. Для чего предназначены формы?
    a) для ввода данных базы и их просмотра
    b) для отбора и обработки данных базы
    c) для выполнения сложных программных действий
40. Для чего предназначены модули?
    a) для автоматического выполнения группы команд
    b) для ввода данных базы и их просмотра
    c) для выполнения сложных программных действий
41. В каком режиме работает с базой данных пользователь?
    a) в заданном
    b) в эксплутационном
    c) в проектировочном
42. В каких элементах таблицы хранятся данные базы?
    a) в записях
    b) в полях
    c) в ячейках
43. Содержит ли какую-либо информацию таблица, в которой нет ни одной записи?
    a) пустая таблица содержит информацию о структуре базы данных
    b) пустая таблица ни содержит, ни какой информации
    c) таблица без записей существовать не может
44. Выберите объект базы данных Microsoft Access, в котором невозможно редактировать записи?
    a) в запросах
    b) в формах
    c) в отчетах
45. Какое расширение имеет база данных, созданная с помощью Microsoft Access?
    a) .mdb
    b) .dba
    c) .xls
46. Какая панель используется для создания кнопки в форме базы данных?
    a) инструментов;
    b) компонентов
    c) элементов
47. Как называются модули в Access, которые связаны с определенной формой или отчетом?
    a) связанные модули
    b) модули класса
    c) стандартные модули

48. Можно ли преобразовать данные из типа MEMO в текстовой тип?
    a) нельзя
    b) можно, но преобразуемое поле должно содержать менее 255 символов
    c) можно, но Access обрезает содержимое поля после 255 символа
49. Что такое QBE?
    a) управляющий запрос
    b) язык программирования запросов
    c) запрос по образцу
50. К какой группе свойств формы относится свойство «блокировка записей»?
    a) макет
    b) другие
    c) данные