

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Локтионова Оксана Геннадьевна

Должность: проректор по учебной работе

Дата подписания: 18.09.2023 19:08:01

Уникальный программный ключ:

0b817ca911e6668abb1356126a159c551a11e11573c943d4e1485451e51e089

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Юго-Западный государственный университет»
(ЮЗГУ)

Кафедра биомедицинской инженерии

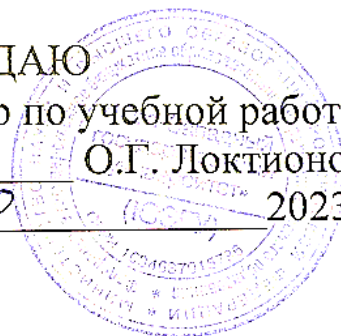
УТВЕРЖДАЮ

Проректор по учебной работе

О.Г. Локтионова

«12» 09

2023 г.



МЕДИЦИНСКИЕ БАЗЫ ДАННЫХ И ЭКСПЕРТНЫЕ СИСТЕМЫ

Методические указания по выполнению лабораторных работ
для студентов направления подготовки
12.03.04 «Биотехнические системы и технологии»

Курск 2023

УДК 615.478

Составители: С.Н. Родионова. В.В. Аксёнов

Рецензент:

Кандидат технических наук, доцент *Т.Н. Конаныхина*

Медицинские базы данных и экспертные системы:
методические указания по выполнению лабораторных работ для студентов направления подготовки 12.03.04 «Биотехнические системы и технологии»/ Юго-Зап. гос. ун-т; сост.: С.Н. Родионова, В.В. Аксёнов. Курск, 2023. - 41 с.

Содержатся теоретические и справочные сведения, необходимые для создания и управления базами данных в среде MS Access и на языке MySQL

Методические указания соответствуют требованиям федеральных государственных образовательных стандартов.

Предназначены для студентов направления подготовки: 12.03.04 «Биотехнические системы и технологии».

Текст печатается в авторской редакции

Подписано в печать . Формат 60x84 1/16.
Усл.печ. л. __. Уч.-изд. л. __. Тираж 30 экз. Заказ 886 .Бесплатно.
Юго-Западный государственный университет.
305040, г. Курск, ул. 50 лет Октября, 94.

СОДЕРЖАНИЕ

ЛАБОРАТОРНАЯ РАБОТА №1. СОЗДАНИЕ ТАБЛИЦ В СРЕДЕ MS ACCESS.	4
ЛАБОРАТОРНАЯ РАБОТА №2. СОЗДАНИЕ ЗАПРОСОВ И ИСПОЛЬЗОВАНИЕ ФОРМ В СРЕДЕ В MS ACCESS.....	9
ЛАБОРАТОРНАЯ РАБОТА №3. СОЗДАНИЕ ОТЧЕТОВ В MS ACCESS.	18
ЛАБОРАТОРНАЯ РАБОТА №4. СОЗДАНИЕ КОНЦЕПТУАЛЬНОЙ МОДЕЛИ ПРЕДМЕТНОЙ ОБЛАСТИ С ПРИМЕНЕНИЕМ CASE — СРЕДСТВ.....	30
ЛАБОРАТОРНАЯ РАБОТА №5. РЕАЛИЗАЦИЯ ЛОГИЧЕСКОЙ МОДЕЛИ СРЕДСТВАМИ СУБД.	36
ЛАБОРАТОРНАЯ РАБОТА №6. РЕАЛИЗАЦИЯ ЗАПРОСОВ К БД.	39

ЛАБОРАТОРНАЯ РАБОТА №1. СОЗДАНИЕ ТАБЛИЦ В СРЕДЕ MS ACCESS.

Цель работы: Научиться создавать базу данных «Записная книжка» в MS Access. Научиться строить таблицы в БД.

Теоретическая часть:

База данных — это систематический сбор данных, хранящихся в электронном виде. Он может содержать любые типы данных, включая слова, цифры, изображения, видео и файлы. Для хранения, извлечения и редактирования данных можно использовать программное обеспечение, называемое системой управления базами данных (СУБД).

Использование Access позволяет:

- добавлять новую информацию в базу данных, например новый артикул складских запасов;
- изменять информацию, уже находящуюся в базе, например перемещать артикул;
- удалять информацию, например если артикул был продан или утилизирован;
- упорядочивать и просматривать данные различными способами;
- обмениваться данными с другими людьми с помощью отчетов, сообщений электронной почты, внутренней сети или Интернета.

Элементами базы данных Access являются:

1. Таблицы
2. Формы
3. Отчеты
4. Запросы
5. Макросы
6. Модули

В рамках данной работы подробнее остановимся на создании таблиц.

Таблица – это некоторая структурированная информация, содержащая характеристики объекта или класса объектов. В настоящее время это самый популярный способ хранения информации в электронном виде. Структура таблицы определяется полями, которые соответствуют столбцам, с указанием типа данных. Содержание таблицы заключено в ее строках, т.е. записях.

Существует несколько типов полей:

Текстовый. Служит для ввода символьных данных: букв, цифр, знаков пунктуации, других символов длиной не более 256 символов. Такой тип должны иметь, например, поля «Фамилия», «адрес».

Числовой. Используется для ввода целых и вещественных чисел. Например, номер дома, вес и т.д.

Дата/время. Позволяет задать дату и время. Например: дата поступления заявки, дата вызова.

Логический тип. Может принимать одно из двух возможных значений: «истина» или «ложь». «Истина» обозначается галочкой, «ложь» - пустой клеткой. Примеры: наличие оплаты, вступление в профсоюз.

Денежный. Разновидность числового типа данных. Отличается более удобным форматом и значком денежной единицы. Для обозначения рублей используется буква «р», дробная часть содержит две цифры для хранения копеек. Например, цена товара, сумма налогов.

OLE-объект позволяет сохранять рисунки, музыкальные клипы, видеозаписи и т.д.

МЕМО-поле – символьное поле длиной до 65535 символов. Данные этого поля не хранятся в таблице, таблица содержит только ссылку на эти данные.

Счетчик – целое значение. Значение счетчика не может повторяться, оно устанавливается автоматически, на единицу больше, чем предыдущее. Его удобно использовать для нумерации записей.

Таблица может быть создана с помощью режимов: *конструктор, мастер таблиц, импорт таблиц*:

Конструктор позволяет изменять имя, тип, формат поля. В этом режиме невозможно вводить данные.

Мастер таблиц. Вы можете выбрать какие-то поля из таблиц-образцов.

Импорт таблиц. Вы можете скопировать готовую таблицу из какой-либо базы данных.

Связь с таблицей позволяет указать адрес готовой таблицы, с которой вы хотите работать. Для однозначного определения каждой записи таблица должна иметь уникальный ключ. Пока не заданы связи между таблицами, надо самостоятельно отслеживать взаимосвязанные записи.

Связи между таблицами дают возможность совместно использовать данные из разных таблиц. Связь каждой пары таблиц обеспечивается одинаковыми полями в них – ключом связи. Связи между таблицами в реляционной базе данных могут быть только типа один-к-одному или один-ко-многим. Связывание таблиц позволяет избежать повторения значений данных в разных таблицах и позволяет упростить процесс их обновления.

Подробнее ознакомиться с принципом работы всех элементов баз данных MS Access можно в руководстве к программе.

Практическая часть.

Порядок выполнения работы:

1. Запустите программу MS Access.
2. Нажмите на кнопку *Новая база данных*
3. Задайте имя новой базы данных - «*Записная книжка Фамилия.accdб*».
4. Создайте таблицу «*Друзья*» с нижеприведенными данными



The screenshot shows the design view of a table named 'Друзья' (Friends) in MS Access. The table has the following fields and data types:

Имя поля	Тип данных
№	Счетчик
Фамилия	Короткий текст
Имя	Короткий текст
Дата рождения	Дата и время
Отчество	Короткий текст
Адрес	Короткий текст
Индекс	Числовой
Телефон	Короткий текст
Хобби	Короткий текст
Эл_почта	Гиперссылка
Семейное положение	Короткий текст

Ключевое поле №п/п.

6. В таблице «*Друзья*» заполните произвольно 10 записей.
7. Добавьте поля «*Отчество*» и «*Дата рождения*»
9. Для поля «*Дата рождения*» установите тип *Дата/время*; в свойствах поля выберите *Краткий формат даты*.

11. Отформатируйте таблицу при помощи инструментов вкладки *Главная группа инструментов* -> *Форматирование текста* следующим образом:

1) цвет сетки - синий;
2) цвет фона - голубой;
3) цвет текста - темно-красный, размер - 12 пт, начертание - курсив.

12. Переименуйте поле «Хобби» в «Увлечения».

13. Удалите запись под номером 8.

14. Измените размер ячеек так, чтобы были видны все данные. Для этого достаточно два раза щелкнуть левой кнопкой мыши на границе полей.

15. Расположите поля в следующем порядке: «№», «Фамилия», «Имя», «Отчество», «Телефон», «Дата рождения», «Увлечения», «Адрес», «Индекс», «Эл. почта».

16. Заполните пустые ячейки таблицы.

17. В режиме *Конструктора* добавьте поле «Семейное положение», в котором будет содержаться фиксированный набор значений - *замужем, не замужем, женат, не женат*. Для создания раскрывающегося списка будем использовать *Мастер подстановок*:

- установите тип данных *Мастер подстановок*;
- в появившемся диалоговом окне выберите строку «*Будет введен фиксированный набор значений*» и нажмите кнопку *Далее*;
- число столбцов - 1;
- введите данные списка - *замужем, не замужем, женат, не женат*;
- нажмите кнопку *Готово*.

18. С помощью раскрывающегося списка заполните новый столбец. Поскольку таблица получилась широкая, то при заполнении данного столбца возникают некоторые неудобства: не видно фамилии человека, для которого заполняется поле «Семейное положение». Чтобы фамилия была постоянно видна при заполнении таблицы, необходимо воспользоваться командой *Закрепить столбцы* из контекстного меню поля «Фамилия».

Контрольные вопросы.

1. Дайте определение базы данных.
2. Для каких целей используется MS Access?
3. Перечислите основные элементы MS Access.

4. Расскажите о функциях элемента «Таблицы».
5. Расскажите о функциях элемента «Формы»
6. Расскажите о функциях элемента «Отчеты»
7. Расскажите о функциях элемента «Запросы»
8. Расскажите о функциях элемента «Макросы»
9. Расскажите о функциях элемента «Модули»

ЛАБОРАТОРНАЯ РАБОТА №2. СОЗДАНИЕ ЗАПРОСОВ И ИСПОЛЬЗОВАНИЕ ФОРМ В СРЕДЕ В MS ACCESS.

Цель работы: Научиться создавать базу данных «Фирма» в MS Access. Освоить создание запросов в БД. Освоить навыки использования форм в базе данных

Теоретическая часть:

Существует четыре типа запросов на изменение: на удаление, на обновление и добавление записей, а также на создание таблицы.

Запрос на удаление удаляет группу записей из одной или нескольких таблиц. Например, запрос на удаление позволяет удалить записи о товарах, поставки которых прекращены или на которые нет заказов. С помощью запроса на удаление можно удалять только всю запись, а не отдельные поля внутри нее.

Запрос на обновление записей вносит общие изменения в группу записей одной или нескольких таблиц. Например, на 10 процентов поднимаются цены на все молочные продукты или на 5 процентов увеличивается зарплата сотрудников определенной категории. Запрос на обновление записей позволяет изменять данные в существующих таблицах.

Запрос на добавление добавляет группу записей из одной или нескольких таблиц в конец одной или нескольких таблиц. Запрос на добавление также полезен при выполнении следующих действий:

- добавление полей на основе условий отбора;
- добавление записей, если некоторые поля из одной таблицы не существуют в другой. Запрос на добавление добавит данные в совпадающие поля и пропустит остальные.

Запрос на создание таблицы создает новую таблицу на основе всех или части данных из одной или нескольких таблиц. Запрос на создание таблицы полезен для выполнения следующих действий:

- создание таблицы для экспорта в другую базу данных Microsoft Access;
- создание отчетов, содержащих данные нескольких таблиц;
- создание резервной копии таблицы.
- создание архивной таблицы, содержащей старые записи;
- повышение быстродействия форм и отчетов, базирующихся на многотабличных запросах или выражениях SQL.

Выбор данных из одной таблицы

В режиме таблицы можно проводить различные операции с данными этой таблицы: просмотр, сортировка, фильтрация и др. Одним из преимуществ запросов является то, что они позволяют достаточно быстро отобрать необходимые данные из нескольких связанных таблиц. При этом все приемы, используемые при работе с одной таблицей, годятся и для сложных многотабличных запросов.

После выполнения запроса на выборку Microsoft Access создает набор записей, содержащий отобранные данные, с которыми можно работать также как и с таблицей. Подробнее ознакомиться с принципом работы всех элементов баз данных MS Access можно в руководстве к программе.

Форма в Access — это объект базы данных, который можно использовать для создания пользовательского интерфейса для приложения базы данных. "Связанная" форма напрямую связана с источником данных, например таблицей или запросом, и может использоваться для ввода, редактирования или отображения данных из этого источника. Кроме того, можно создать "несвязанную" форму. Она не привязывается непосредственно к источнику данных, но также содержит кнопки команд, метки или другие элементы управления, которые необходимы для работы с приложением.

Практическая часть.

Задание 1. Создание базы данных

1. Запустите MS Access.
2. Создадим базу данных «Фирма». Задайте имя новой базы данных - «Фирма Ваша фамилия.accdб».
3. В базе необходимо создать 3 таблицы: *Сотрудники*, *Клиенты* и *Заказы*,
4. Откройте таблицу *Заказы* в режиме Конструктора.

Сотрудники

Имя поля	Тип данных
Код сотрудника	Счетчик
Фамилия	Короткий текст
Имя	Короткий текст
Отчество	Короткий текст
Должность	Короткий текст
Телефон	Короткий текст
Адрес	Короткий текст
Дата рождения	Дата и время
Зарботная плата	Денежный
Фото	Вложение
Эл_почта	Гиперссылка

Клиенты

Имя поля	Тип данных
Код клиента	Счетчик
Название компании	Короткий текст
Адрес	Короткий текст
Номер телефона	Короткий текст
Факс	Числовой
Адрес электронной почты	Гиперссылка
Заметки	Длинный текст

Заказы

Имя поля	Тип данных
Код заказа	Счетчик
Код клиента	Числовой
Код сотрудника	Числовой
Дата размещения	Дата и время
Дата исполнения	Дата и время
Сумма	Денежный
Отметка о выполнении	Логический

5. В таблице *Заказы* для поля *Код сотрудника* выберите тип данных *Мастер подстановок*.

- В появившемся окне выберите команду «*Объект «столбец подстановки» будет использовать значения из таблицы или запроса*» и щелкните на кнопке *Далее*.

- В списке таблиц выберите таблицу *Сотрудники* и щелкните на кнопке *Далее*.

- В списке *Доступные поля* выберите поле *Код сотрудника* и щелкните на кнопке со стрелкой, чтобы ввести поле в список *Выбранные поля*. Таким же образом добавьте поля *Фамилия* и *Имя* и щелкните на кнопке *Далее*.

Создание подстановки

Какие поля объекта "Сотрудники" содержат значения, которые следует включить в поле подстановки? Отобранные поля станут столбцами в объекте "поле подстановки".

Доступные поля:

Отчество
Должность
Телефон
Адрес
Дата рождения
Зарботная плата
Эл_почта

Выбранные поля:

Код сотрудника
Фамилия
Имя

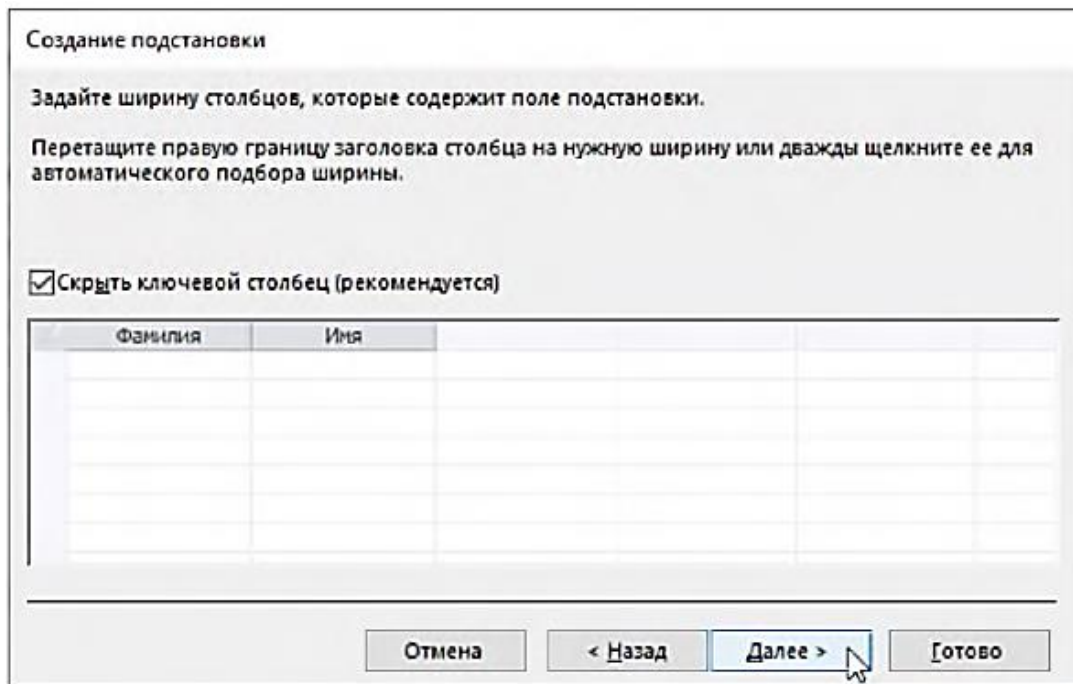
Отмена < Назад Далее > Готово

- Выберите порядок сортировки списка по полю *Фамилия*.

- В следующем диалоговом окне задайте необходимую ширину столбцов раскрывающегося списка.

- Установите флажок *Скрыть ключевой столбе*» и нажмите кнопку *Далее*.

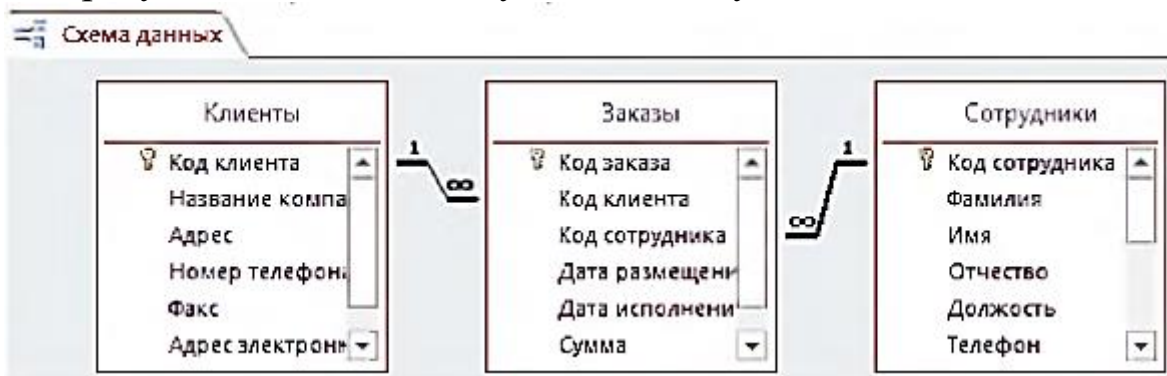
- На последнем шаге *Мастера подстановок* замените при необходимости надпись для поля подстановок и щелкните на кнопке *Готово*.



6. Аналогичным образом в таблице *Заказы* создайте раскрывающийся список для поля *Код клиента*.

7. После создания ключевых полей измените связи. Установите *Обеспечение целостности данных*, *Каскадное обновление связанных полей* и *Каскадное удаление связанных записей*.

В результате должна получиться следующая схема данных:



8. В таблицу *Сотрудники* внесите данные о семи работниках.

9. В таблицу *Клиенты* внесите данные о десяти предприятиях, с которыми работает данная фирма.

10. В таблице *Заказы* оформите несколько заявок, поступивших на фирму.

Задание 2. Отбор данных с помощью запросов.

1 запрос.

Создайте запрос при помощи *Мастер запросов*: укажите таблицу *Сотрудники* и выберите поля *Фамилия, Имя, Телефон*. Введите имя запроса - *Телефоны*.

2-запрос.

Создайте запрос с помощью *Конструктора*: выберите таблицу *Клиенты*. Выберите поля: *Название компании, адрес, телефон*. Отсортируйте записи в поле *Название компании* в алфавитном порядке, для этого в раскрывающемся списке строки *Сортировка* выбрать пункт *по возрастанию*. Сохраните запрос с именем «*Адреса клиентов*».

3 запрос.

Создайте запрос «*Дни рождения*», в котором можно будет просмотреть дни рождения сотрудников. Допустим, мы хотим узнать, у кого из сотрудников день рождения в текущем месяце, например, в апреле. Для этого откройте запрос в режиме *Конструктора*. В строке *Условие отбора* для поля «*Дата рождения*» введите значение **.04.**. В данной записи * означают, что дата и год рождения могут быть любыми, а месяц 4-м (т. е. апрель). После этого окно запроса должно выглядеть так:

Поле:	Фамилия	Имя	Дата рождения
Имя таблицы:	Сотрудники	Сотрудники	Сотрудники
Сортировка:			
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:			Like *.04.*
или:			

4 запрос.

Создайте универсальный запрос для отбора сотрудников «*Дата рождения*», используя *запрос с параметром*. При запуске такого запроса на экран выводится диалоговое окно для ввода значения в качестве условия отбора. Чтобы создать запрос с параметром, пользователю необходимо ввести текст сообщения в строке. *Условие отбора* бланка запроса

Поле:	Фамилия	Имя	Дата рождения
Имя таблицы:	Сотрудники	Сотрудники	Сотрудники
Сортировка:			
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:			Like [Введите дату]
или:			

Запись *Like*[*Введите дату*] означает, что при открытии запроса появится диалоговое окно с текстом «*Введите дату*» и полем для ввода условия отбора. Если ввести условие **.04.**, то в запросе появится список сотрудников, родившихся в апреле. Запустите запрос еще раз и введите значение **.05.**, посмотрите, как изменился запрос.

5 запрос.

Создайте запрос «*Телефоны*» так, чтобы при его запуске выводилось диалоговое окно с сообщением «*Введите фамилию*». Поскольку в запросе нужно вывести конкретную фамилию, в условии отбора слово *Like* писать не надо

6 запрос.

Создайте запрос «*Выполненные заказы*», содержащий следующие сведения: фамилия и имя сотрудника, название компании, с которой он работает, отметка о выполнении и сумма заказа. Данные запроса возьмите из нескольких таблиц. В условии отбора для логического поля *Отметка о выполнении* введите *Да*, чтобы в запросе отображались только выполненные заказы. Сделайте так, чтобы столбец *Отметка о выполнении* не выводился на экран.

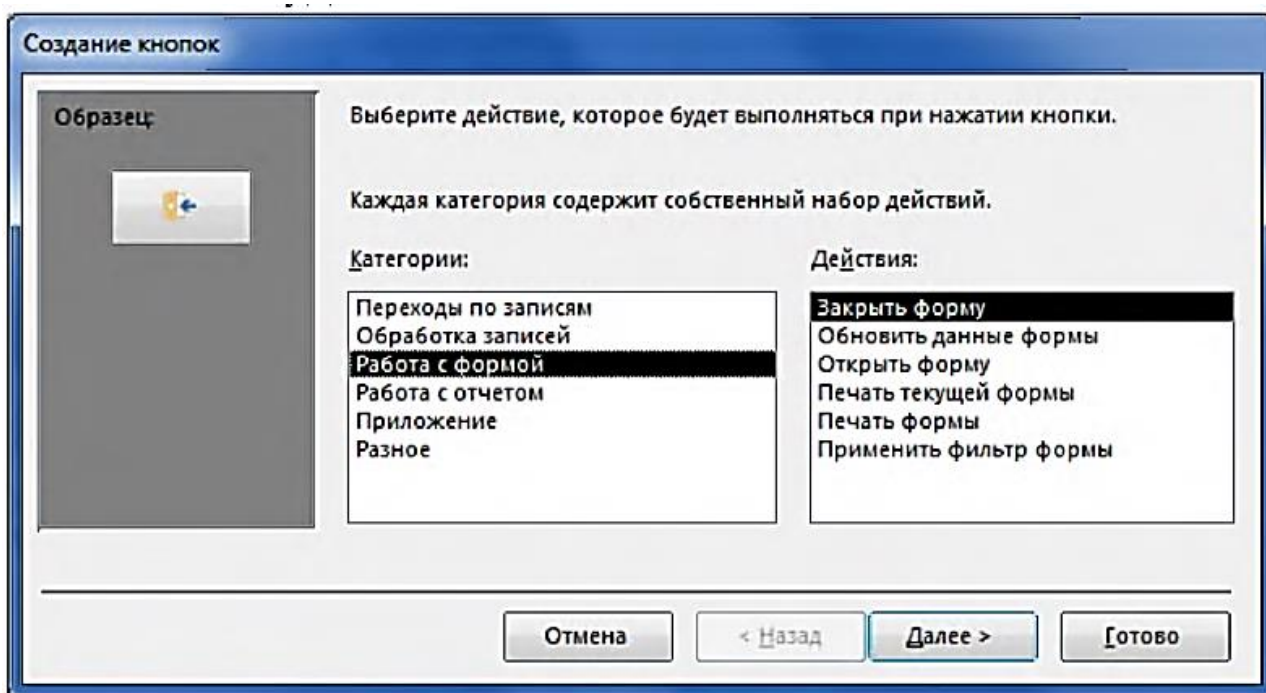
Поле:	Фамилия	Имя	Название компании	Отметка о выполнении	Сумма
Имя таблицы:	Сотрудники	Сотрудники	Клиенты	Заказы	Заказы
Сортировка:					
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:				Да	
или:					

Задание 3. Использование форм в базе данных

1 форма. С помощью Мастера создайте форму *Сотрудники*. В режиме *Конструктора* разместите элементы в удобном для вас порядке, измените размер и цвет текста.

- В заголовок формы добавьте текст *Сотрудники фирмы*.

- В примечание формы добавьте объект *Кнопка* (вкладка ленты Конструктор —> панель инструментов Элементы управления). После того как вы «нарисуете» кнопку указателем, на экране появится диалоговое окно *Создание кнопок*. В категории *Работа с формой* выберите действие *Заккрыть форму* и нажмите кнопку *Далее*.



Выберите рисунок или текст, который будет размещаться на кнопке. В последнем диалоговом окне *Мастера кнопок* задайте имя кнопки и нажмите *Готово*.

- Создайте кнопки *Выход из приложения*, *Поиск записи*, *Удаление записи*.

2-форма. С помощью *Мастера* создайте форму *Клиенты*.

3 форма. С помощью *Мастера* создайте форму *Заказы*.

4 форма. Создайте пустую форму.

Для добавления к форме набора вкладок щелкните по кнопке *Вкладка* на панели инструментов *Элементы управления*. Сначала добавятся только две вкладки с формальными именами *Вкладка 1* и *Вкладка 2*.

Добавьте еще одну вкладку. Для этого щелкните по уже созданной вкладке правой кнопкой мыши и в контекстном меню выберите *Вставить вкладку*.

Переименуйте ярлычки вкладок так, чтобы на них отображались названия данных, которые будут в них располагаться: *Сотрудники*, *Клиенты*, *Помощь*. Это можно сделать выделив нужную вкладку в Окне свойств в категории *Другие* строка *Имя*

Перейдите на вкладку *Сотрудники* и перетащите на нее мышкой из базы данных форму *Сотрудники*.

Аналогичным образом поместите форму *Клиенты* на вкладку *Клиенты*.

На вкладку *Помощь* поместите некоторые советы по работе с базой данных.

Данную форму сохраните с именем *Сотрудники фирмы*.

Сотрудники фирмы

Сотрудники | Клиенты | Помощь

Сотрудники

Код сотруду

Фамилия

Имя

Отчество


Должность

Телефон

Адрес

Дата рождения

Зарботная плата

Фото 

Эл_почта

Запись: 1 из 1

Контрольные вопросы.

1. Как формируются отношения между таблицами?
2. Как происходит изменение отношения между таблицами?
3. Как происходит удаление отношения между таблицами?
4. Какие запросы можно делать в Access?
5. Что такое Форма в Access?
6. Как происходит создание форм?
7. Для каких целей нужен режим конструктора?.
8. Для каких целей нужен режим макета?

ЛАБОРАТОРНАЯ РАБОТА №3. СОЗДАНИЕ ОТЧЕТОВ В MS ACCESS.

Цель работы: Научиться создавать базу данных «Предприятие» в MS Access. Освоить создание отчетов в БД.

Теоретическая часть:

Отчет (report) — это объект базы данных, который используется для вывода на экран, в печать или файл структурированной информации. Reports позволяют извлечь из таблиц или запросов базы данных необходимую информацию и представить ее в виде удобном для восприятия. Report содержит заголовок, область данных, верхний и нижний колонтитулы, примечание и разбит на страницы.

С помощью отчетов можно просматривать, форматировать и группировать информацию в базе данных Microsoft Access. Например, можно создать простой отчет в виде списка номеров телефонов всех контактов или сводный отчет по итогам продаж компании в разных регионах за разные периоды. Можно создавать «несвязанные» отчеты, в которых данные не отображаются, но в этой статье мы рассмотрим отчет, связанный с источником данных (таблицей или запросом). Макет отчета разбит на разделы, которые можно просмотреть в режиме конструктора.

Практическая часть.

Задание 1. Создание базы данных

1. Создайте новую базу данных. Назовите эту базу данных «Кадры + ваша фамилия».

2. Создайте три таблицы,

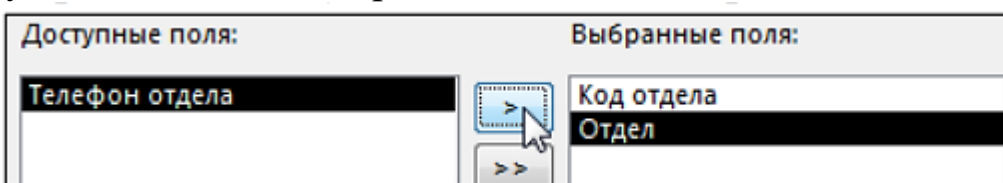
Сохраните таблицы под именами в соответствии с информацией, которая будет отражена в этих таблицах: «Сотрудники», «Должности», «Отделы». Сохраните таблицы. Перейдите в режим конструктора. Для каждой таблицы определите поля, тип данных и размер в соответствии с информацией, которая будет содержаться в этих полях. Пример таблицы сотрудники приведен ниже:

Сотрудники	
Имя поля	Тип данных
Код сотрудника	Счетчик
Фамилия	Короткий текст
Имя	Короткий текст
Отчество	Короткий текст
Адрес	Короткий текст
Телефон	Короткий текст
Дети	Числовой
Дата рождения	Дата и время
Семейное положение	Логический
Надбавка	Денежный
Код отдела	Числовой
Код должности	Числовой

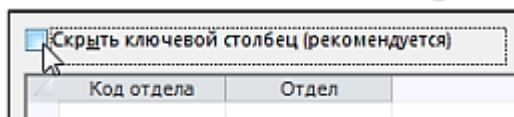
3. Создайте *связи*. Для того чтобы, создать или изменить связи между таблицами, нужно сначала закрыть все таблицы. Мы будем создавать связь с помощью *Мастера подстановок*, поэтому откройте таблицу *Сотрудники* в режиме конструктор и для полей *Код отдела* и *Код должности* измените тип данных на *Мастер подстановок*. В открывшемся окне для *Код отдела* выбираем следующие шаги:

Выбираем «Объект "поле подстановки" получит значения из другой таблицы или другого запроса» нажимаем *Далее*

Выбираем *Таблица Отдел* нажимаем *Далее*. Переносим следующие значения в правый столбец



Нажимаем *Далее* 2 раза. Убираем галочку возле скрыть ключевой столбец



Нажимаем *Далее* выбираем *Код отдела* и нажимаем *Готово*.

Аналогично делаем мастер подстановок для поля *Код должности*, но выбираем уже таблицу *Должности* и поля *Код должности* и *Должность*. Сохраняем таблицу *Сотрудники* и закрываем ее.

Затем на панели инструментов «Работа с данными» нажмите кнопку «Схема данных». Для изменения созданных связей нажимаем на связь, а затем на кнопку «Изменить связи» на панели «Конструктор». В появившемся окне устанавливаем галочки возле следующих полей: «обеспечение целостности данных», «каскадное обновление связанных полей», «каскадное удаление связанных записей». Нажмите кнопку «ОК».

В открывшемся окне оставьте галочку «Обеспечение целостности данных» и создайте связь. Связь отобразится в рабочей области. Так делаем для каждой связи. У вас должна получиться следующая схема



Задание 2. Создание форм.

Чтобы ввести данные для рассматриваемой БД «Кадры», необходимо создать три формы: для таблиц *Должности* и *Отдел*, и самой последней для таблицы *Сотрудники*

Формы можно создавать с помощью конструктора и с помощью мастера. Воспользуемся мастером создания форм.

Выбираем таблицу *Должности*. Далее нужно выбрать поля для таблицы. Выберем все поля таблицы должности.

На следующем этапе «Внешний вид формы» выберем «Ленточный». Далее на этапе «Требуемый стиль» выберите стиль по своему усмотрению. На этапе «Задание имени формы» назовите форму «Штатное расписание + ваша Фамилия».

Откроется форма для ввода данных в таблицу «Должности».

Объект 1. Форма «Штатное расписание + _Фамилия» Ведите данные:

Штатное расписание_Фамилия

Штатное расписание_Фамилия

Код должности	Должность	Оклад
1	Управляющий	50 000,00 сом
2	Заместитель управляющего	40 000,00 сом
3	Ведущий специалист	30 000,00 сом
4	Специалист	20 000,00 сом

Закройте форму и откройте таблицу «Должности». Убедитесь, что все данные внесены в таблицу.

Аналогично создайте форму для заполнения таблицы «Отделы». Назовите ее «Структура организации_Фамилия». Заполните таблицу «Отделы».

Объект 2. Форма «Структура организации_Фамилия» Ведите данные:

Структура организации_Фамилия

Структура организации_Фамилия

Код отдела	Отдел	Телефон отдела
1	Валютный	334
2	Кредитный	335
3	Отдел кадров	337
4	Информационный	338
5	Юридический	336

Создайте форму для заполнения таблицы «Сотрудники». Назовите ее «Карточка сотрудника». В процессе создания формы выберите внешний вид формы «В один столбец».

Объект 3. Форма «Карточка сотрудника». Заполните карточки в соответствии с данными, всего 24 сотрудника. Недостающие данные придумайте самостоятельно.

Сотрудники кредитного отдела

1. Иванов Иван Васильевич, ул. Пушкина, д.2, кв. 15, 0700-35-25-25, 01/01/1960, семейный, 2 детей, Управляющий, премия 3000.

2. Петрова Надежда Юрьевна, ул. Ленина, д.8, кв.46, 0555-25-85-96, 15/06/1965, семейный, 2 детей, Зам. управляющего, премия 2500.

3. Иванова Ольга Дмитриевна, ул. Свободы, д. 456, кв.1, 0772-25-21-14, 02/04/1975, семейный, 1 ребенок, Ведущий специалист, премия 1000.

Сотрудники валютного отдела

1. Управляющий (1 человек)
2. Заместитель управляющего (1 человек)
3. Ведущий специалист (1 человека)
4. Специалист (2 человека)

Сотрудники юридического отдела

1. Управляющий (1 человек)
2. Заместитель управляющего (1 человек)
3. Ведущий специалист (1 человека)
4. Специалист (2 человека)

Сотрудники отдела кадров

1. Управляющий (1 человек)
2. Заместитель управляющего (1 человек)
3. Ведущий специалист (1 человек)
4. Специалист (2 человека)

Сотрудники информационного отдела

1. Управляющий (это вы).
2. Заместитель управляющего (1 человек)
3. Специалист (2 человека)

Карточка сотрудника

Карточка сотрудника

Код сотрудника	<input type="text" value="(№)"/>
Фамилия	<input type="text"/>
Имя	<input type="text"/>
Отчество	<input type="text"/>
Адрес	<input type="text"/>
Телефон	<input type="text" value="I"/>
Дети	<input type="text" value="0"/>
Дата рождения	<input type="text"/>
Семейное положение	<input type="checkbox"/>
Надбавка	<input type="text" value="0,00р"/>
Код отдела	<input type="text"/>
Код должности	<input type="text"/>

Задание 3. Создание запросов.

1 запрос. В появившемся окне мастера запросов «Новый запрос» выберем «Простой запрос». Далее из разных таблиц выберем необходимые данные: *Фамилия, Имя, Отдел, Должность, Надбавка*. Имя запроса «Запрос_Все сотрудники».

2 запрос. Отберите всех сотрудников, которые занимают должности управляющего. Необходимы данные для запроса: *Фамилия, Имя, Должность, Отдел, Телефон*. Имя запроса «Список управляющих» Для этого сделайте отбор нужных полей при помощи мастера запросов. Откройте запрос в режиме конструктора. В строке условие отбора в столбце *Должность* пропишите условие = «Управляющий». Нажмите кнопку *Выполнить*. Сохраните запрос.

3 запрос. Отберите только тех сотрудников, у которых есть дети. Необходимо посчитать количество детей, для закупки подарков на *Новый год*. Имя запроса «Подарки на НГ детям сотрудников». Поля для отбора в запрос: *Фамилия, Имя, Дети, Отдел*. В режиме конструктор для столбца *дети* пропишите условие отбора >0. Нажмите кнопку *Выполнить*. На вкладке Главная нажмите кнопку *Суммы*. Внизу таблицы запроса появится строчка *Итог*. В столбце *дети* из раскрывающегося списка выберите *Сумма*. Сохраните запрос.

4 запрос. Создайте запрос «Ведомость з/п». Поля для отбора: *Отдел, Фамилия, Имя, Надбавка, Оклад*. Перейдите в режим конструктор. Нажмите кнопку *Итоги*.

В последнем пустом столбце в строке *Групповая операция* выберите *Выражение* и в строке *Поле* пропишите: *Итого:[Надбавка]+[Оклад]*. Нажмите кнопку *Выполнить* и посчитайте итоговую сумму для столбца *Итого*.

5 запрос. Создайте запрос «Список сотрудников для командировки». Поля для отбора: *Отдел, Фамилия, Имя, Семейное положение, Должность*. Нужно отобрать только холостых сотрудников. Для этого после создания запроса в режиме конструктор в столбец *семейное положение* в условие отбора вводим =false.

Задание 4. Создание отчетов.

1 отчет. На первом шаге выберите все поля запроса «Все сотрудники». На следующем шаге выберите вид группировки - по отделам. Добавлять уровню группировки на следующем шаге не нужно. Затем задайте сортировку по убыванию надбавки. Вид макета

для отчета - *Ступенчатый*. Стиль отчета - по вашему усмотрению. *Название отчета - Отчет_надбавки сотрудников*. Лист книжный. В режиме макет сделайте настройку ширины столбцов таким образом, чтобы было видно все данные.

2 отчет. Создайте простой отчет для запроса «*Список управляющих*». *Первая группировка по Должности*. Название отчета «*Список управляющих*». Лист книжный. В режиме макет сделайте настройку ширины столбцов таким образом, чтобы было видно все данные.

3 отчет. Создайте простой отчет для запроса «*Подарки на НГ детям сотрудников*». На первом шаге выбираем сотрудники. На следующем шаге добавьте уровень группировки Отдел. На следующем шаге нажимаем кнопку Итоги. В открывшемся окне в строке Дети ставим галочку в столбце sum. Закрываем окно. Макет выбираем *Ступенчатый*. Название отчета «*Подарки на НГ детям сотрудников*». Лист книжный. В режиме макет сделайте настройку ширины столбцов таким образом, чтобы было видно все данные.

4 отчет. Создайте простой отчет для запроса «*Ведомость з/п*». Группировка по отделу. Макет выбираем *Ступенчатый*. Название отчета «*Ведомость з/п*». Лист книжный. В режиме макет сделайте настройку ширины столбцов таким образом, чтобы было видно все данные.

5 отчет. Создайте простой отчет «*Данные сотрудников*». Отберите из таблицы сотрудники: Фамилия, Имя, Отчество, Адрес, Телефон, Дата рождения, Семейное положение; из таблицы Отделы - поле Отдел. Группировка по отделу. Макет выбираем Блок. Лист альбомный. Название отчета «*Данные сотрудников*»

6 отчет. Создайте простой отчет для запроса «*Список сотрудников для командировки*». Первая группировка по отделу. Название отчета «*Список сотрудников для командировки*». Лист книжный. В режиме макет сделайте настройку ширины столбцов таким образом, чтобы было видно все данные.

Задание 5. По образцу базы данных «Фирма» создайте базу данных согласно заданному варианту:

Вариант 1

1) Разработайте базу данных «*Электронная библиотека*», состоящую из трех таблиц со следующей структурой:

- Книги - шифр книги (ключевое поле), автор, название, год издания, количество экземпляров.

- Читатели - читательский билет (ключевое поле), фамилия, имя, отчество, адрес.

- Выданные книги - порядковый номер (счетчик, ключевое поле), шифр книги, читательский билет, дата выдачи, дата возвращения, дата фактического возвращения.

2) Установите связи между таблицами.

3) С помощью запроса отберите все книги, выпущенные с 1991 по 2014 годы.

4) Создайте форму для ввода данных.

5) Создайте отчет.

Вариант 2

1) Разработайте базу данных «Продуктовый магазин», которая состоит из четырех таблиц со следующей структурой:

- Товары - код товара (ключевое поле), наименование товара, количество товара.

- Поступление товаров - номер (счетчик, ключевое поле), код товара, дата поступления, цена приобретения товара за единицу, код поставщика.

- Продажа товаров - номер (счетчик, ключевое поле), код товара, месяц продажи, проданное количество за месяц, цена продажи товара.

- Поставщики - код поставщика (ключевое поле), название поставщика, адрес поставщика, телефон поставщика.

2) Установите связи между таблицами.

3) С помощью запроса отберите товары, цены которых от 100 до 450 сом.

4) Создайте форму для ввода данных.

5) Создайте отчет.

Вариант 3

1) Разработайте базу данных «Сессия», состоящую из трех таблиц со следующей структурой:

- Студенты - шифр студента (ключевое поле), фамилия, имя, отчество, курс, группа.

- Экзамены - номер (счетчик, ключевое поле), шифр студента, дата, шифр дисциплины, оценка.

- Дисциплины - шифр дисциплины (ключевое поле), название дисциплины, количество часов.

2) Установите связи между таблицами.

3) С помощью запроса отберите студентов, сдавших экзамен на 4 или 5.

4) Создайте форму для ввода данных.

5) Создайте отчет.

Вариант 4

1) Разработайте базу данных «Абитуриенты», состоящую из четырех таблиц со следующей структурой:

- Анкета - номер абитуриента (ключевое поле), фамилия, имя, отчество, дата рождения, оконченное среднее учебное заведение (название, номер, населенный пункт), дата окончания учебного заведения, адрес, телефон, шифр специальности.

- Специальности - шифр специальности (ключевое поле), название специальности.

- Дисциплины - шифр дисциплины (ключевое поле), название дисциплины.

- Вступительные экзамены - номер абитуриента, шифр дисциплины, экзаменационная оценка.

2) Установите связи между таблицами.

3) Составьте запрос для отбора студентов, сдавших экзамены без троек.

4) Создайте форму для ввода данных.

5) Создайте отчет.

Вариант 5

1) Разработайте базу данных «Оптовый склад», состоящую из четырех таблиц со следующей структурой:

- Склад - код товара, количество, дата поступления.

- Товары - код товара (ключевое поле), название товара, срок хранения.

- Заявки - код заявки (ключевое поле), название организации, код товара,

- требуемое количество.

- Отпуск товаров - код заявки (ключевое поле), код товара, отпущенное количество, дата отпуска товара.

2) Установите связи между таблицами.

3) С помощью запроса отберите товары, количество которых от 50 до 200 штук.

4) Создайте запрос с параметром для отбора товаров, поступивших на склад какого-либо числа.

5) Создайте формы для ввода данных, отчеты и главную кнопочную форму.

Вариант № 6

1) Разработайте базу данных «Прокат спортивного оборудования», состоящую из трех таблиц со следующей структурой:

- Клиенты - код клиента (ключевое поле), фамилия, имя, отчество, телефон, адрес, паспортные данные, залог.

- Склад - код оборудования (ключевое поле), название, количество, залоговая стоимость, остаток.

- Прокат - № п/п, клиент, оборудование, дата выдачи, срок возврата, отметка о возврате, оплата проката.

2) Установите связи между таблицами.

3) Создайте запрос для отбора оборудования с залоговой стоимостью от 100 до 5000 сом.

4) Создайте запрос с параметром для отбора клиентов, возвративших оборудование.

5) Создайте формы для ввода данных, отчеты и главную кнопочную форму.

Вариант 7

1) Разработайте базу данных «Транспортные перевозки», состоящую из трех таблиц со следующей структурой:

- Транспорт - марка автомобиля, государственный номер (ключевое поле), расход топлива.

- Заявки - код заявки (ключевое поле), дата заявки, название груза, количество груза, пункт отправления, пункт назначения.

- Доставка - № п/п, дата и время отправления, дата и время прибытия, код заявки, государственный номер автомобиля, пройденное расстояние.

2) Установите связи между таблицами.

3) С помощью запроса отберите заявки с количеством груза от 100 до 500 кг.

4) Создайте запрос с параметром для отбора транспорта по марке автомобиля.

5) Создайте формы для ввода данных, отчеты и главную кнопочную форму.

Вариант 8

1) Разработайте базу данных «Банк», состоящую из трех таблиц со следующей структурой:

- Клиенты - код клиента (ключевое поле), фамилия, имя, отчество, паспорт, телефон, адрес, заработная плата.

- Виды кредитов - код кредита (ключевое поле), название кредита, процентная ставка, условия предоставления.

- Предоставленные кредиты — № п/п, клиент, кредит, дата предоставления, срок, дата возврата, сумма, отметка о возврате.

2) Установите связи между таблицами.

3) Создайте запрос для отбора клиентов, взявших кредит от 500 000 до 1 000 000 сом.

4) Создайте запрос с параметром для отбора кредитов по процентной ставке.

5) Создайте формы для ввода данных, отчеты и главную кнопочную форму.

Вариант 9

1) Разработайте базу данных «Туристическая фирма», состоящую из четырех таблиц со следующей структурой:

- Клиенты - код клиента (ключевое поле), фамилия, имя, отчество, телефон, адрес, паспорт.

- Сотрудники - код сотрудника (ключевое поле), фамилия, имя, отчество, должность, телефон, адрес, паспортные данные.

- Туристические маршруты - код маршрута (ключевое поле), название, описание маршрута, страна, стоимость путевки, количество дней, вид транспорта.

- «Заказы» - код заказа (ключевое поле), клиент, маршрут, сотрудник (менеджер, оформивший заказ), дата, отметка об оплате.

2) Установите связи между таблицами.

3) Создайте запрос для отбора маршрутов со стоимостью от 10000 до 20000 сом.

4) Создайте запрос с параметром для отбора клиентов, выбравших определенный вид маршрута.

5) Создайте формы для ввода данных, отчеты и главную кнопочную форму.

Вариант 10

1) Разработайте базу данных «Поликлиника», состоящую из четырех таблиц со следующей структурой:

- Врачи код врача (ключевое поле), ФИО, должность, специализация, стаж работы, адрес, телефон.

- Болезни - № п/п (ключевое поле), название заболевания, рекомендации по лечению, меры профилактики.

- Пациенты - код пациента (ключевое поле), ФИО, адрес, телефон, страховой полис, паспорт.

- Диагноз - № п/п (ключевое поле), пациент, заболевание, лечащий врач, дата обращения, дата выздоровления.

2) Установите связи между таблицами.

3) С помощью запроса отберите врачей-стоматологов и ортопедов.

4) Создайте запрос с параметром для отбора пациентов с определенным видом заболевания.

5) Создайте формы для ввода данных, отчеты и главную кнопочную форму.

Контрольные вопросы.

1. Как формируются отношения между таблицами?
2. Как происходит изменение отношения между таблицами?
3. Как происходит удаление отношения между таблицами?
4. Какие запросы можно делать в Access?
5. Для чего нужны отчеты в Access?
6. Из каких составных частей состоят отчеты?
7. Как происходит создание отчета в Access?

ЛАБОРАТОРНАЯ РАБОТА №4. СОЗДАНИЕ КОНЦЕПТУАЛЬНОЙ МОДЕЛИ ПРЕДМЕТНОЙ ОБЛАСТИ С ПРИМЕНЕНИЕМ CASE — СРЕДСТВ.

Цель работы: получить навыки проектирования базы данных средствами CASE.

Теоретическая часть:

Разработка современных сложных информационных систем требует применения специальных методик и инструментов. Этим объясняется возросший среди разработчиков интерес к CASE - технологиям и инструментальным CASE - средствам. CASE - технология представляет собой совокупность методологий анализа, проектирования, разработки и сопровождения сложных систем программного обеспечения, поддержанных комплексом взаимосвязанных средств автоматизации. CASE - это инструментарий для системных аналитиков, разработчиков и программистов, который позволяет описывать бизнес - процессы на компьютере, используя полученные схемы при разработке или настройке системы.

CASE-средства ERwin и VPwin, разработанные фирмой Computer Associates International, Inc., являются, пожалуй, самыми популярными. Erwin применяется как средство для разработки IDEF1X - модели.

IDEF1X (IDEF1 Extended) — Data Modeling — методология построения реляционных структур (баз данных), относится к типу методологий «Сущность- взаимосвязь» (ER — Entity-Relationship) и, как правило, используется для моделирования реляционных баз данных, имеющих отношение к рассматриваемой системе.

ERwin Data Modeler (ранее: ERwin) позволяет проектировать, документировать и сопровождать базы данных. Создав наглядную модель базы данных, предоставляет возможность оптимизации структур. Визуальное моделирование повышает качество создаваемой базы данных, продуктивность и скорость её разработки. Кроме того, позволяет созданную модель данных перенести в выбранную базу данных в автоматическом режиме.

При создании новой модели пользователю предлагается выбрать тип создаваемой модели: логическая, физическая, логическая/физическая. После чего предоставляется возможность формирования структуры базы данных. Пользуясь возможностями ERwin Data Modeler созданную модель данных можно перенести в

выбранную базу данных в автоматическом режиме. При этом отпадает необходимость вручную создавать структуру таблиц и связей между ними. Возможна и обратная операция - из существующей базы данных сформировать модель данных в программе.

Чтобы добавить в модель новую сущность, необходимо на панели «Toolbox» выбрать инструмент «Entity», после чего щелкнуть мышью на свободном месте рабочей области. Появится рамка новой сущности, и программа перейдет в режим ввода имени сущности. Завершается ввод имени нажатием на клавишу Enter.

После ввода имени сущности в зависимости от режима отображения, который может быть определен с помощью пункта меню «Format»_«Display level», возможны варианты:

- в режиме «Entity» отобразится вновь добавленная сущность;
- в режиме «Attribute» будет предложено ввести имена ключевых атрибутов (этот режим включен по умолчанию), при этом ввод каждого из них завершается нажатием «Enter», при нажатии клавиши «Tab» осуществляется переход для ввода неключевых атрибутов;

- в режиме «Definition» будет предложено ввести определения для вновь добавленной сущности;

- в режиме «Primary key» будет предложено ввести имена ключевых атрибутов;

- в режиме «Icon» добавленная сущность будет представлена пиктограммой. Редактор атрибутов вызывается двойным щелчком мышью на любом из них. Если сущность была создана без атрибутов, то добавить новый атрибут можно кнопкой «New...» в нижней левой части диалога. Если атрибуты уже есть, их можно переименовать кнопкой «Rename...» или удалить кнопкой «Delete».

В редакторе атрибутов можно указать тип атрибута и другие свойства.

При добавлении нового атрибута на экране появляется диалоговое окно, где предлагается указать его тип данных, логическое и физическое имя. Чтобы создать связь, необходимо в панели «Toolbox» выбрать тип связи и последовательно щелкнуть на двух связываемых сущностях. При этом для связей типа «один-ко-многим» сначала щелкнуть на родительской, потом на дочерней сущностях. Для связей типа «многие-ко-многим» в произвольном порядке. Открыв окно редактора связей (двойным щелчком), можно

задать ее свойства. На вкладке «General» можно в разделе «Relationship Cardinality» можно указать конкретный тип связи, который обеспечивает ERwin:

- ноль, один или много;
- один или много;
- ноль или один;
- точное количество.

В поле «Verb Phrase» можно задать роли участия сущностей в связи.

Так же указать, является связь идентифицирующей или нет, допускает наличие пустых значений.

На вкладке «RoleName» можно указать, будет ли переименовываться атрибут, соответствующий внешнему ключу, и под каким именем он будет отображен в дочерней таблице.

Для обеспечения целостности данных при модификации информации в базе данных на вкладке «RI Actions» можно указать, какие действия выполняются с сущностью при выполнении операций (удаления, вставки, обновления) с другой сущностью

На физическом уровне модель выглядит так, как она должна быть реализована средствами СУБД. В раскрывающемся списке в панели инструментов следует выбрать «Physical». При модификации физической модели рекомендуется придерживаться английских наименований атрибутов и сущностей. Это определяется в соответствии с требованиями, предъявляемыми СУБД и приложениями, работающие с разработанной БД. После переименования следует проверить значения типа данных для каждого атрибута, допустимость неопределенных значений. Работа с атрибутами выполняется в окне «Columns», которое можно открыть двойным щелчком по любому из столбцов таблицы. После завершения моделирования разработанная структура данных может быть транслирована в выбранную БД определенной СУБД.

Практическая часть

Задание на лабораторную работу.

Выбрать вариант предметной области из нижеследующей таблицы, пользуясь пакетом Erwin или Oracle SQL Developer Data Modeler, сформировать модель данных для реализации БД.

Номер	Предметные области
-------	--------------------

варианта	
1.	Регистратура лечебного учреждения
2.	Склад-магазин компьютерной техники
3.	Склад-магазин книжной продукции
4.	Издательство
5.	Регистрация транспортных средств
6.	Риэлторское агенство
7.	Юридическая консультация
8.	Расписание занятий
9.	Научная конференция
10.	Вуз
11.	Спортивные соревнования
12.	Сессия
13.	Продажа билетов
14.	Объекты недвижимости
15.	Каталоги фильмов
16.	Каталоги музыкальные
17.	Гостинница
18.	Частный медицинский кабинет
19.	Больница
20.	Аптеки и лекарственные препараты
21.	Ветаптека
22.	Офтальмологическая клиника
23.	Центр спортивной медицины
24.	Стоматология
25.	Перинатальный центр

- 1.1. Описать предметную область;
- 1.2. Выделить сущности (модель должна включать не менее 6-8 сущностей);
- 1.3. Определить связи между ними;
- 1.4. Осуществить формализацию модели, устранить избыточность;
- 1.5. Задать необходимые ограничения целостности.
- 1.6. Оформить отчет, который должен включать: описание предметной области, сущностей, ограничений целостности, а также описание работы

Пример реализации БД для ЛР №1. Предметная область – успеваемость студента.

1. Описание предметной области

Каждому студенту, обучающемуся в университете необходимо 2 раза в год по окончании семестра сдавать сессию. О студенте в базе данных должна храниться следующая информация: ФИО, Год поступления. Группа, Факультет, Адрес. Любой студент имеет уникальный номер зачетной книжки.

2. Выделение сущностей Выделим следующие сущности:

Студент с атрибутами: ФИО. Год поступления, Группа, Факультет, Адрес.

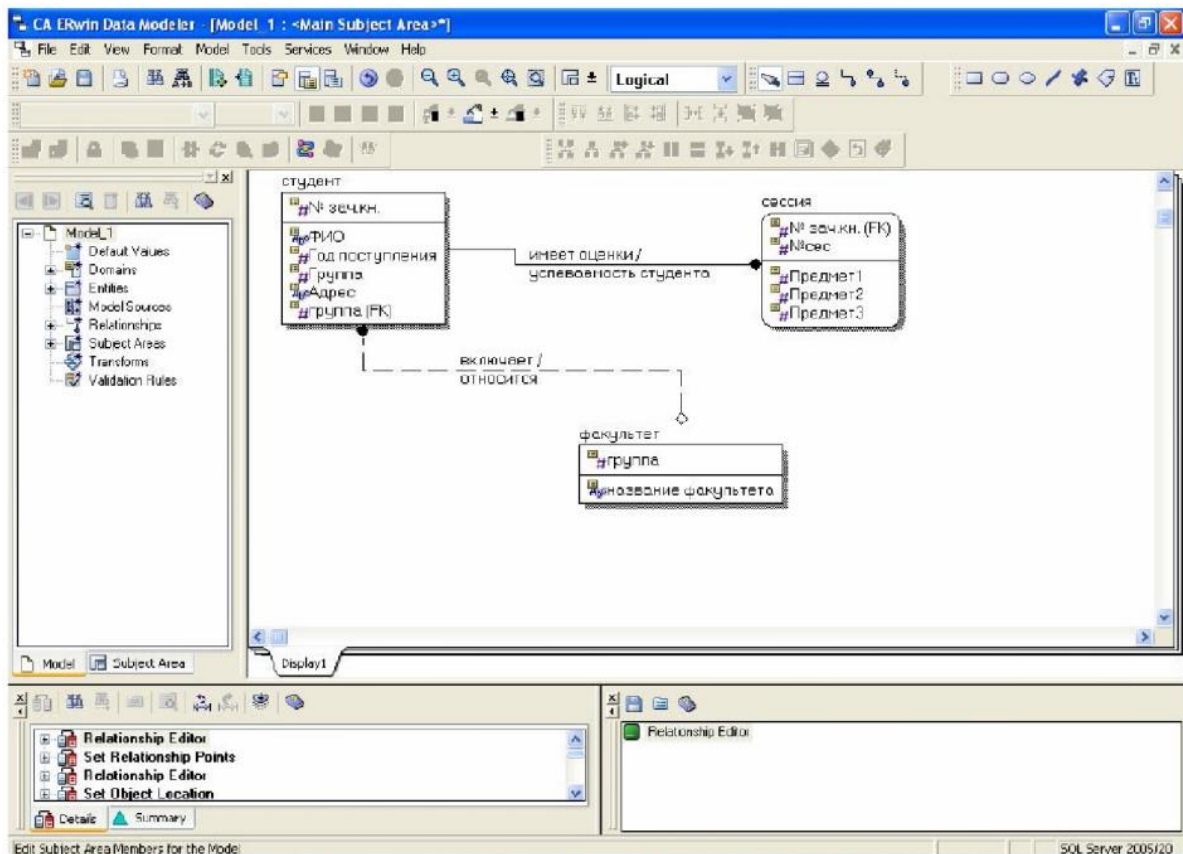
Сессия с атрибутами: № сессии, Предмет1, Предмет2, Предмет3

3. Определить связи между ними

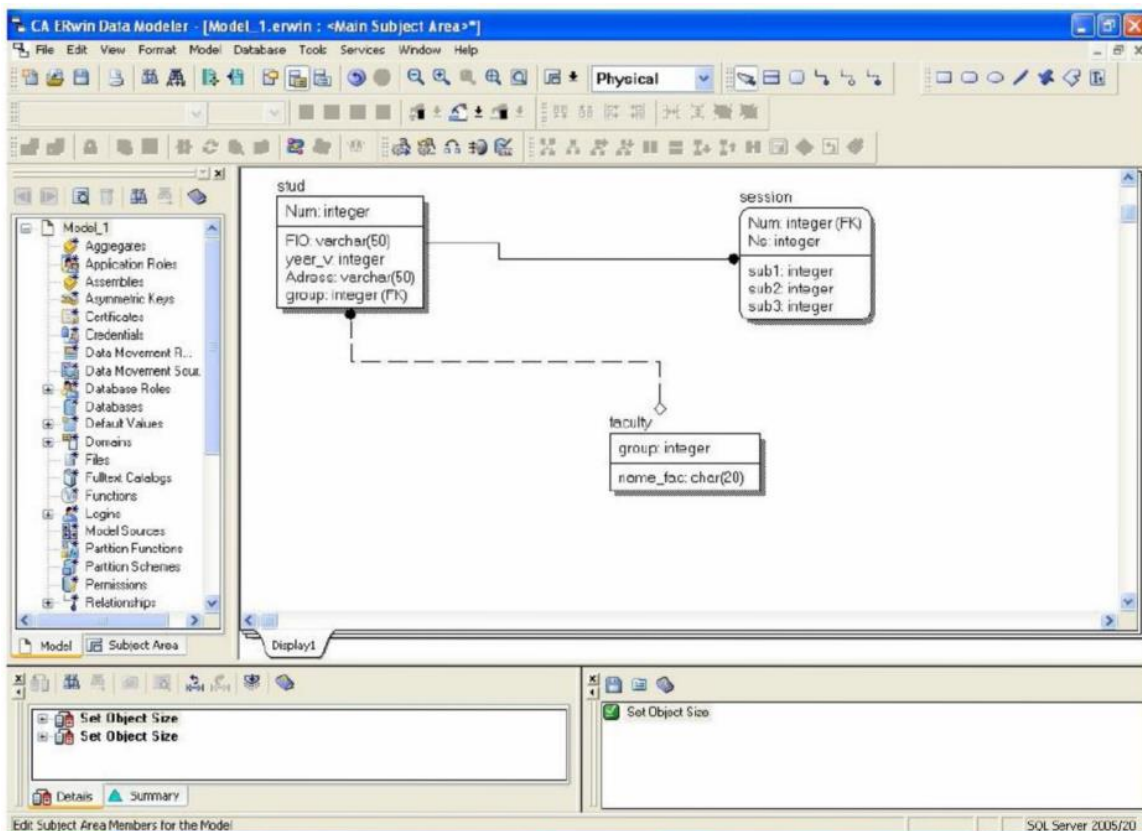
Между сущностями Студент и Сессия устанавливается связь один ко многим.

4. Осуществить формализацию модели, устранить избыточность

5. Для исключения явлений избыточности и аномалий проведем нормализацию до 3 НФ. Получим следующие сущности:



6. На физическом уровне данная модель будет выглядеть следующим образом:



Контрольные вопросы:

1. Что такое БД?
2. Что такое схема БД?
3. Что такое CASE технология, какие инструменты вы знаете?
4. Какие свойства связей и сущностей вы знаете, приведите пример.
5. Что такое предметная область БД?

ЛАБОРАТОРНАЯ РАБОТА №5. РЕАЛИЗАЦИЯ ЛОГИЧЕСКОЙ МОДЕЛИ СРЕДСТВАМИ СУБД.

Цель работы: Получить навыки проектирования и физической реализации реляционной базы данных средствами CASE.

Теоретическая часть:

Реляционная база данных - это совокупность взаимосвязанных таблиц, описывающих некоторую предметную область. Структура таблиц определяет эффективность программ, обрабатывающих эти таблицы, и все приложение в целом. Реляционная модель баз данных основывается на математических принципах теории реляционных наборов.

Создавать и модифицировать таблицы, добавлять, изменять и удалять записи в них возможно с помощью языка SQL (Structured Query Language - язык структурированных запросов). Особенность языка состоит в том, что его команды ориентированы на работу не с отдельными записями, а с их наборами, как свободных, так и реляционно-связанных таблиц и баз данных. В командах SQL формулируется критерий и способы преобразования информации. При этом SQL определяет наиболее эффективные последовательности операций для получения результата.

Для создания таблицы используется оператор **CREATE TABLE:**

```
CREATE TABLE <имя таблицы>  
( {<имя поля> <тип данных> [( <размер> )]  
[<ограничения целостности поля> ...] },  
[<ограничения целостности таблицы> , ... ] );
```

Для обязательных полей устанавливается ограничение not null.

- <имя поля> - имя поля (столбца) таблицы;
- <тип данных> - один из выше перечисленных типов;
- <размер> - размер поля в символах;
- <ограничения целостности поля> - предполагает

использование следующих ограничений:

primary key - первичный ключ (обязательный и уникальный);

unique - уникальное значение поля в пределах таблицы;

[not] null - [не] возможность не указывать значение поля;

check(<условие>) - проверка условия для поля (полей);

default<выражение> - задание значения поля по умолчанию;

foreign key <имя таблицы>[<имя столбца>] внешний ключ;

<ограничения целостности таблицы> - те же ограничения, что и для поля и дополнительно:

foreign key [Описок полей>, ...] references <имя таблицы> [(Описок полей>)] внешний ключ

Например

```
Create table tab (id numeric(6) primary key, class numeric(3), fdata date, group char(6), foreign key (class, fdata) references exam(class, fdata));
```

Для вставки кортежа (записи) применяется оператор **INSERT**:

```
INSERT INTO <имя таблицы> [(<имя поля>), ...] VALUES (<список выражений>)|  
<запрос>;
```

Для создания триггеров используется оператор **CREATE TRIGGER**:

```
CREATE TRIGGER <имя_триггера>  
ON <имя_таблицы>  
FOR {[INSERT] [.UPDATE] [.DELETE] }  
[WITH ENCRYPTING]  
AS  
SQL-операторы (Тело триггера)
```

- Имя триггера является идентификатором во встроенном языке программирования СУБД и должно удовлетворять соответствующим требованиям;
 - В параметре FOR задается одна или несколько операций модификации, которые запускают данный триггер.
 - Параметр WITH ENCRYPTING имеет тот же смысл, что и для хранимых процедур, он скрывает исходный текст тела триггера.
 - Существует несколько правил, ограничивающих набор операторов, используемых в теле триггера.
 - В большинстве СУБД действуют следующие ограничения:
 - Нельзя использовать в теле триггера операции создания объектов БД (новой БД, новой таблицы, нового индекса, новой хранимой процедуры, нового триггера, новых индексов, новых представлений);
 - Нельзя использовать в триггере команду удаления объектов DROP для всех типов базовых объектов БД;
 - Нельзя использовать в теле триггера команды изменения базовых объектов . ALTER TABLE, ALTER DATABASE;

- Нельзя изменять права доступа к объектам БД, то есть выполнять команду GRANT или REVOKE;
- Нельзя создать триггер для представления (VIEW);
- В отличие от хранимых процедур, триггер не может возвращать никаких значений, он запускается автоматически сервером и не может связаться самостоятельно ни с одним клиентом

Практическая часть.

1. В соответствии с моделью из л/р №1 создать необходимые таблицы, учитывая типы данных, характерные для выбранной СУБД, и необходимые внутренние ограничения целостности.

2. Создать необходимые триггеры и хранимые процедуры для обеспечения целостности данных.

3. Заполнить созданные структуры 8-10 записями.

Структура данных, разработанная в л/р №1 может быть транслирована в базу данных СУБД Oracle, MS SQL Server, MySQL и другие.

4. Оформить отчет,

Контрольные вопросы.

1. Что такое кортеж БД?
2. Что такое триггер?
3. Что такое логическая реализация БД?
4. Как обеспечивается право доступа к БД?

ЛАБОРАТОРНАЯ РАБОТА №6. РЕАЛИЗАЦИЯ ЗАПРОСОВ К БД.

Цель работы: Получить навыки организации запросов к данным, хранящимся в базе данных.

Теоретическая часть:

Извлечение данных из отношений выполняется с помощью команды **SELECT** (селекция). Эта команда не изменяет данные в БД.

Результатом выполнения команды *SELECT* является временное отношение, которое помещается в курсор (специальную область памяти СУБД) и обычно сразу выводится на экран. Синтаксис этой команды

```
SELECT * | { [ ALL | DISTINCT ] <список вывода>,...}
FROM {<имя таблицы> [<алиас>] },...
[ WHERE <условие>]
[ GROUP BY {<имя поля> | <целое>}... [ HAVING <условие>] ]
[ ORDER BY {<имя поля> | <целое> [ ASC | DESC ] },... ]
[ UNION [ALL] SELECT ...];
```

- список вывода

Элементом списка вывода может быть выражение и необязательный алиас. Выражение может включать имена полей, знаки операций, вызовы функций и константы. Список элементов, разделяется запятыми.

- алиас - временный синоним имени таблицы, определённый внутри запроса.

- условие - содержит условие-предикат выборки или | и условие соединения таблиц

DISTINCT - предикат удаления из результирующего отношения повторяющихся кортежей.

ALL - предикат, обратный к *DISTINCT* (используется по умолчанию).

Рассмотрим основные предложения команды *SELECT*

SELECT - после этого ключевого слова указывается список вывода

Например:

```
Select поле1 *поле2 as синоним, поле3
```

Если надо вывести все поля из отношений, к которым обращается данный запрос, можно указать символ * (если в отношениях нет полей с одинаковыми именами). В этом случае сначала будут выведены поля таблицы, стоящей первой в предложении *FROM*, затем - второй и т.д. Поля, относящиеся к одной таблице, будут выводиться в том порядке, в каком они были записаны при создании таблицы.

FROM- в этом предложении указывается имя таблицы (имена таблиц), из которых осуществляется выборка.

WHERE - содержит условия выбора отдельных записей.

GROUP BY - группирует записи по значению одного или нескольких полей. Каждой группе в результирующем отношении соответствует одна запись.

HAVING - позволяет указать условия выбора для групп записей. Может использоваться только после **GROUP BY**

ORDER BY - упорядочивает результирующие записи по значению одного или нескольких полей: *ASC* - по возрастанию, *DESC* - по убыванию.

Если во фразе **FROM** указаны две и более таблицы, то эта последовательность действий выполняется для декартова произведения указанных таблиц.

Представление (view, обзор) - это хранимый запрос, создаваемый на основе команды *SELECT*.

Представление реально не содержит данных. Запрос, определяющий представление, выполняется тогда, когда к представлению происходит обращение с другим запросом, например, *SELECT*, *UPDATE* и т.д.

Создание представления выполняется командой **CREATE VIEW**

```
CREATE VIEW <имя представления> [(<имя столбца>,...)]
```

```
AS <запрос>;
```

Запрос, на основании которого создаётся представление, называется **определяющим запросом**, а таблицы, к которым происходит обращение в определяющем запросе - **базовыми таблицами**. Определяющий запрос не может включать предложение *ORDER BY*.

Если не указывать имена столбцов, то они получают названия по именам, перечисленным в списке выбора определяющего запроса. Указывать имена столбцов представления обязательно, если список выбора содержит агрегирующие функции или столбцы с одинаковыми именами из разных таблиц.

Например: Создать представление, содержащее информацию о студентах:

```
create view Wstud
as Num, Name, Group, Year, Bd, Money, Adress
from Stud;
select * from Wstud;
```

Представление может быть обновляемым и не обновляемым. Обновляемым является представление, при обращении к которому можно обновить базовую таблицу. Изменения будут произведены в базовой таблице и отразятся в представлении. По стандарту

SQL-2 представление не является обновляемым, если определяющий запрос:

- содержит ключевое слово *DISTINCT*;
- ссылается на другое необновляемое представление;
- содержит вычисляемые выражения в списке выбора;
- выбирает данные более чем из одной таблицы;
- содержит предложение *GROUP BY*.

Если вносимые изменения выходят за рамки определяющего запроса и поэтому не могут быть отражены в представлении, они могут быть отвергнуты системой (это зависит от реализации).

Практическая часть.

1. Продумать запросы разработанной вами БД согласно предметной области.
2. Создать представление согласно своему варианту предметной области.
3. Оформить отчет,

Контрольные вопросы:

1. Как организуются запасы в БД?
2. Что такое представление?
3. Как выглядит структура SQL-запроса?
4. Приведите примеры SQL-запросов?
5. Перечислите основные агрегатные функции SQL.