

Санкт-Петербургский государственный технологический институт
(технический университет)
Кафедра систем автоматизированного проектирования и управления

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
к лабораторной работе 1 по курсу

БАЗЫ ДАННЫХ

ст. преподаватель
Иванов А.А.

2023

Лабораторная работа № 1.

Разработка информационной системы в среде Microsoft Access.

Теоретическая часть

Концептуальные модели и их использование при проектировании ИС.

Разработчики баз данных берут концептуальную модель за основу для проектирования БД. В концептуальной модели используются три типа связей для описания отношений между данными: *один-ко-многим*, *многие-ко-многим* и *один-к-одному*. Проектировщики баз данных обычно используют для этого сокращенные обозначения — 1:M, M:N и 1:1 соответственно (хотя стандартным обозначением для связи «многие-ко-многим» является M:N, можно использовать и обозначение M:M). Следующие примеры показывают, в чем состоит *разница между этими связями*.

- **Один-ко-многим.** Художник может написать много различных картин, но каждая из них написана одним художником. Следовательно, художник («один») связан с картинами («многие»). Поэтому проектировщик баз данных обозначает связь «ХУДОЖНИК пишет КАРТИНУ» как 1:M. Точно так же в учетной записи клиента («один») может содержаться множество счетов, но данные счета («многие») относятся к единственному клиенту. Связь «КЛИЕНТУ выписан СЧЕТ» также относится к типу 1:M.
- **Многие-ко-многим.** Студент может изучать несколько предметов, и на занятиях по каждому предмету может присутствовать много студентов. Следовательно, студенты («многие») связаны с предметами («многие»). Поэтому проектировщики баз данных обозначают связь «СУДЕНТЫ изучают ПРЕДМЕТЫ» как M:N.
- **Один-к-одному.** Структура отдела продаж предприятия может быть организована так, что каждым магазином заведует один сотрудник. В свою очередь каждый заведующий (естественно, являющийся сотрудником) управляет одним магазином. Поэтому связь «СОТРУДНИК заведует МАГАЗИНОМ» обозначается как 1:1.

Модель «сущность-связь»

Поскольку проще исследовать структуру графически, чем описывать ее в текстовой форме, проектировщики БД посчитали удобным использовать соответствующие изобразительные средства, в которых логические объекты (сущности, entity) и их связи (relationship) могли изображаться графическими средствами.

С этой целью была разработана модель «сущность-связь» (entity-relationship, ER), которая является, самым широко распространенным и простым графическим инструментом моделирования данных.

ER-модели обычно представляются в виде диаграмм «сущность-связь» (ER-диаграмма, ERD). В ER-диаграмме используется графическое представление модели компонентов базы данных.

Основу ER-модели составляют следующие компоненты.

- **Сущность** (логический объект) – персона, местоположение или предмет, сведения о которых подлежат сбору и хранению. В ER-модели сущность представлена в виде прямоугольника. Название сущности (имя существительное) записывается в центре прямоугольника, как правило, заглавными буквами и предпочтительнее в единственном числе на русском

языке: ХУДОЖНИК лучше, чем ХУДОЖНИКИ и СТУДЕНТ лучше, чем СТУДЕНТЫ. Обычно если ER-диаграмма связана с реляционной моделью, сущность отображается на реляционную таблицу. Каждая строка реляционной таблицы соответствует *экземпляру сущности*.

Сущность описывается набором *атрибутов*. Каждый атрибут описывает отдельное свойство сущности. Например, сущность СТУДЕНТ имеет такие атрибуты, как номер студенческого билета, фамилию, имя, отчество и т.д.

- **Связь** описывает соединение между данными. Большинство связей описывает соединение между двумя сущностями. Разработчики ER-диаграмм для обозначения типа связи используют термин связность (на ER-диаграммах связность записывается рядом с прямоугольником, соответствующим сущности). Связь изображается на ER-диаграмме ромбом, соединенным с соответствующей сущностью. Название связи (в глагольной форме) записывается внутри ромба.

На рис. 1 проиллюстрированы ER-диаграммы с использованием связей, которые мы обсуждали выше.

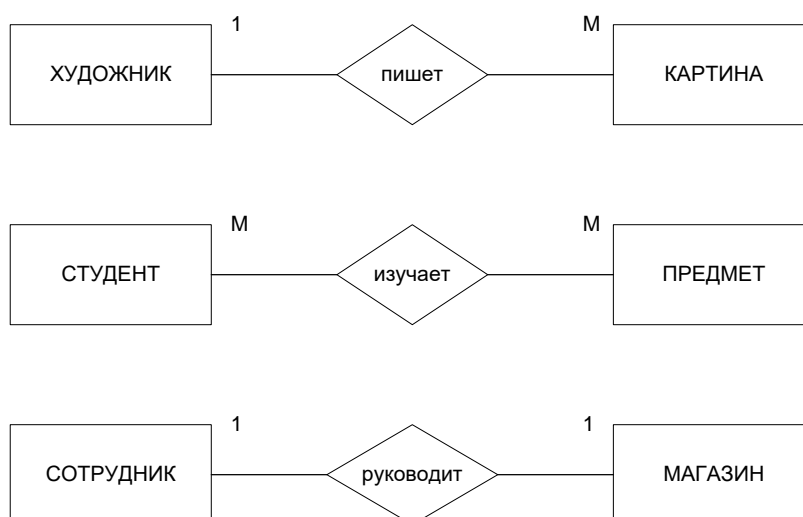


Рисунок 1 - Представление отношений на ER-диаграмме (модель Чена)

ER-диаграмма, представленная на рисунке 1, построена на основе так называемой *модели Чена*, описанной Питером Ченом в одной из его работ. На ER-диаграммах, созданных на базе модели Чена, связи обозначаются ромбами, а сущности — прямоугольниками.

Другой широко известный способ построения ER-диаграмм, называемый «птичья лапка», представлен на рисунке 2. Название «птичья лапка» произошло от специфического вида символа, которым обозначается связь в направлении «многие».

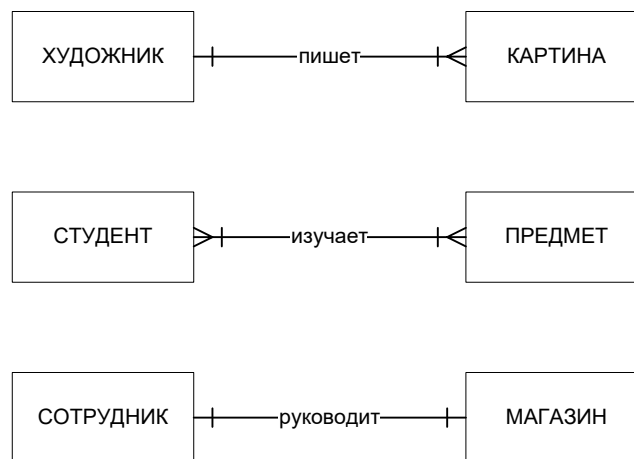


Рисунок 2 - Представление отношений на ER-диаграмме (модель «птичья лапка»)

Этапы проектирования информационных систем

При проектировании структур данных для информационных систем можно выделить три основных подхода:

1. Формулирование знаний о системе (определение типов исходных данных и их взаимосвязей) и требований к обработке данных, получение с помощью CASE-системы (системы автоматизации проектирования и разработки баз данных) готовой схемы БД или даже готовой прикладной информационной системы.
2. Сбор информации об объектах решаемой задачи в рамках одной таблицы (одного отношения) и последующая декомпозиция ее на несколько взаимосвязанных таблиц на основе процедуры нормализации отношений.
3. Структурирование информации для использования в информационной системе в процессе проведения системного анализа на основе совокупности правил и рекомендаций.

Чаще всего для проектирования используют первый подход, поэтому остановимся на нем более подробно. Этот подход охватывают первые два уровня проектирования информационных систем: концептуальный и логический.

Концептуальное (инфологическое) проектирование – построение формализованной модели предметной области. Такая модель строится без ориентации на какую-либо конкретную СУБД.

Инфологическая модель дает формализованное описание предметной области независимо от структур данных, исключая неоднозначность за счет использования средств формальной логики. Одним из главных понятий является понятие объекта. Целью инфологического моделирования является формализация объектов реального мира предметной области и методов обработки информации в соответствии с поставленными задачами обработки и требованиями представления данных естественными для человека способами сбора и представления информации. Основными компонентами инфологической модели являются: описание предметной области; описание методов обработки; описание информационных потребностей пользователя.

Основные элементы данной модели:

1. Описание объектов предметной области и связей между ними.
2. Описание информационных потребностей пользователей (описание основных запросов к БД).
3. Описание алгоритмических зависимостей между данными.

4. Описание ограничений целостности, т.е. требований к допустимым значениям данных и к связям между ними.

Для инфологического проектирования часто используют объектно-ориентированные модели, в которых главным компонентом выступает объект с набором атрибутов и функций, которые могут обращаться к атрибутам этого объекта. При объектно-ориентированном подходе в рассматриваемой предметной области выделяют классы объектов, для которых определяются методы обработки, что, в конечном счете, обеспечивает лучшую реализацию динамического поведения информационной системы. При объектно-ориентированном подходе используют графические методы моделирования предметной области. Классическим подходом является проектирование информационных систем с помощью диаграммы «сущность-связь» (entity-relation diagram или ER-диаграммы), которая представлена на рисунке 3.

Разработка объектно-ориентированных моделей, как правило, выполняется с помощью case-средств. К таким средствам относятся BPwin (PLATINUM technology), Silverrun (Silverrun technology), Oracle Designer (Oracle), Rational Rose (Rational Software), Toad Data Modeler (Quest Software) и др.

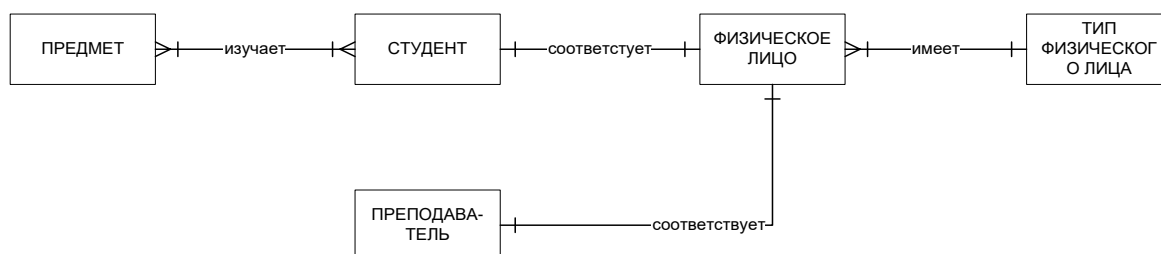


Рисунок 3 - Пример инфологической модели.

Логическое (дatalogическое) проектирование – отображение инфологической модели на модель данных, используемую в конкретной СУБД, например, на реляционную модель данных. Для реляционных СУБД дatalogическая модель – набор таблиц, обычно с указанием ключевых полей, связей между таблицами. Если инфологическая модель построена в виде ER-диаграмм (или других формализованных средств), то дatalogическое проектирование представляет собой построение таблиц по определённым формализованным правилам, а также нормализацию этих таблиц. Пример логической модели представлен на рисунке 4.

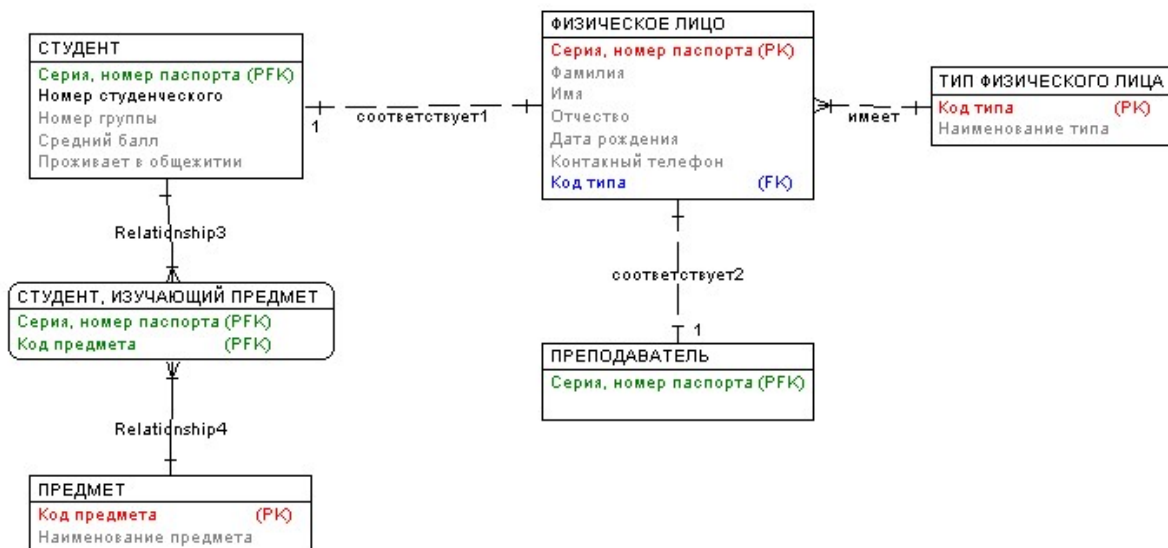


Рисунок 4 - Пример логической модели.

В Microsoft Access логическую модель можно создать на вкладке «Работа с базами данных» в пункте меню «Схема данных».

Наконец, третьим уровнем проектирования ИС является физическое проектирование.

Физическое проектирование – реализация логической модели средствами конкретной СУБД, а также выбор решений, связанных с физической средой хранения данных: выбор типов данных, определение индексов.

Реляционная модель

Реляционная модель данных, предложенная сотрудником фирмы IBM Эдгаром Коддом (E.F. Codd) основывается на понятии отношение (relation) и состоит из трех частей:

- структурной части;
- целостной части;
- манипуляционной части.

Структурная часть описывает, какие объекты рассматриваются реляционной моделью. Постулируется, что единственной структурой данных, используемых в реляционной модели, являются нормализованные n-арные отношения. Отношения (*relation*) в реляционной модели представляются в виде таблиц.

Под *таблицей* понимается двумерная структура, состоящая из строк и столбцов, содержащая некоторые данные. Таблица содержит группу *связанных сущностей*, т. е. набор сущностей.

Физическое размещение данных в реляционных базах на внешних носителях легко осуществляется с помощью обычных файлов. Примерами реляционных СУБД являются: DB2, Firebird, Informix, Ingres, InterBase, MSDE, MS SQL Server, MongoDB, MySQL, mSQL, Oracle Database, Pervasive SQL, PostgreSQL, Sybase, ЛИНТЕР.

Свойства отношений.

Отношение характеризуется следующими параметрами:

Степень отношения – это число его атрибутов (столбцов). Отношение степени один называют унарным, степени два – бинарным, степени три – тернарным, ..., а степени n – n -арным.

Кардинальное число или *мощность отношения* – это число его кортежей (строк). Кардинальное число отношения изменяется во времени в отличие от его степени.

Рассмотрим свойства реляционной таблицы на примере таблицы STUDENT:

STU_NUM	STU_LNAME	STU_FNAME	STU_PNAME	STU_DOB	STU_GROUP	STU_PHONE	STU_AVM	STU_HOST
20236	Иванов	Петр	Сидорович	7.11.1984	823	1234567	4.5	No
20237	Петров	Сидор	Иванович	7.1.1985	824	7654321	4.6	Yes
20238	Сидоров	Иван	Петрович	1.5.1985	825	1726354	4.7	No

1. Таблица представляет собой двумерную структуру, состоящую из строк и столбцов. Таблица STUDENT представляет собой двумерную структуру, составленную из трёх строк (кортежей) и девяти столбцов. Можно также считать, что таблица состоит из трёх записей и девяти атрибутов (полей). Также можно сказать, что степень отношения STUDENT – 9, а мощность – 3.
2. Каждая строка таблицы (*кортеж*, tuple) представляет собой отдельную сущность внутри набора сущностей. Данная таблица содержит набор сущностей STUDENT, включающий 3 отдельные сущности (строки).
3. Каждый столбец таблицы представляет собой атрибут, и у каждого столбца есть свое имя
4. На каждом пересечении строки и столбца имеется единственное значение.
5. Каждая таблица должна иметь атрибут или несколько атрибутов, уникально идентифицирующих каждую строку.
6. Все значения в столбце должны отображаться в одинаковом формате. Например, если атрибуту присваивается формат целого, то все значения в столбце, представляющем данный атрибут, должны быть целыми
7. Каждый столбец имеет определенный диапазон значений, называемый доменом *атрибута* (attribute domain).
8. Порядок следования строк и столбцов для СУБД не существенен.

Связывание таблиц

При проектировании реальных СУБД информацию обычно размещают в нескольких таблицах. Таблицы при этом связаны семантикой информации. В реляционных СУБД для указания связей таблиц производят операцию их связывания. Выигрыш, обеспечиваемый в результате связывания таблиц заключается в следующем:

1. Обычно СУБД выполняет контроль целостности вводимых данных в соответствии с установленными связями, что повышает достоверность хранимой в БД информации.
2. Установление связи между таблицами облегчает доступ к данным. Связывание таблиц при выполнении таких операций, как поиск, просмотр, редактирование, выборка и подготовка отчетов, обычно обеспечивает возможность обращения к произвольным полям связанных записей. Это уменьшает количество явных

обращений к таблицам данных и число манипуляций в каждой из них.

Между таблицами могут устанавливаться бинарные (между 2-мя таблицами), тернарные (между 3-мя таблицами) и, в общем случае, n-арные связи. Рассмотрим наиболее часто встречающиеся бинарные связи.

Логическое связывание таблиц производится с помощью ключа связи. Ключ связи по аналогии с обычным ключом таблицы, состоит из одного или нескольких полей, которые в данном случае называют полями связи.

Суть связывания состоит в установлении соответствия полей связи между двумя таблицами, т.е. связь создается, когда две таблицы совместно используют один атрибут с общим значением. Конкретнее, первичный ключ одной таблицы появляется в качестве внешнего ключа (foreign key) в связанной таблице.

В зависимости от того, как соотносятся ключевые поля с полями связи между двумя таблицами в реляционной базе данных могут устанавливаются все рассмотренные нами основные виды связи.

Связь вида 1:1 является частным случаем связи 1:M и образуется когда все поля связи одной и другой таблицы являются ключевыми. Поскольку значение в ключевых полях обеих таблиц не повторяются обеспечивается взаимно однозначное соответствие записей из этих таблиц. На практике связи вида 1:1 используют сравнительно редко, так как хранимую в двух таблицах информацию легко объединить в одну таблицу, которая занимает гораздо меньше физической памяти. Возможны случаи, когда действительно удобнее иметь не одну, а две и более таблицы. Причинами этого может быть необходимость ускорить обработку, повысить удобство работы нескольких пользователей с общей информацией, обеспечить более высокую степень защиты информации и т.д.

Связь 1:M имеет место в случае, когда одной записи первой (иногда ее еще называют главной) таблицы соответствует несколько записей второй (подчиненной) таблицы.

И, наконец, самый общий вид связи M:N возникает в случаях, когда несколькими записями одной таблицы, соответствует несколько записей другой таблицы. При этом каждая сторона отношения выглядит как отношение один-ко-многим. Однако если рассматривать взаимосвязь таблиц с двух сторон, становится очевидным, что ни одна из таблиц не может быть главной и для их связывания необходима третья таблица. Связующая таблица представляет собой промежуточную таблицу, которая служит «мостом» между двумя таблицами в отношении многие-ко-многим. Ее ключ состоит из ключевых полей этих таблиц, с каждой из которых она связана отношением многие-к-одному. Помимо ключевых полей, связующая таблица может содержать дополнительные поля, которых нет в связываемых таблицах, но которые могут иметь значение для каждой из них. Таким образом, отношение многие-ко-многим складывается из отношений один-ко-многим и многие-к-одному.

Практическое задание

1. Откройте Microsoft Access и создайте новую базу данных
2. В режиме конструктора создайте следующие 4 таблицы.
 - **таблицу «ФАКУЛЬТЕТ» с атрибутами:**
Код факультета (тип данных – счетчик);
Наименование факультета.
 - **таблицу «ГРУППА» с атрибутами:**
Номер группы
Специальность
Факультет (тип данных – числовой, размер поля – длинное целое);
 - **таблицу «СТУДЕНТ» с атрибутами:**
Номер зачетки;
Фамилия;
Имя;
Отчество;
Дата рождения (тип данных – дата и время);
Группа (тип данных должен соответствовать типу данных атрибута «Номер группы» в таблице «ГРУППА»);
 - **таблицу «ПРЕДМЕТ» с атрибутами:**
Код предмета (тип данных – счетчик);
Наименование предмета;
 - **таблицу «ОБУЧЕНИЕ» с атрибутами:**
Студент (тип данных должен соответствовать типу данных атрибута «Номер зачетки» в таблице «СТУДЕНТ»);
Предмет (тип данных – числовой, размер поля – длинное целое);
Оценка;
Дата сдачи (тип данных – дата и время);

При создании поля «Оценка» выберите тип данных «Мастер подстановок...» и укажите, что будет введен фиксированный список значений. Введите значения для столбца «Столбец1»: 5; 4; 3; 2; н/а (каждое в отдельной ячейке). При создании полей «Дата рождения» и «Дата сдачи» установите формат поля «Краткий формат даты», а затем выберите соответствующую маску ввода. Для тех атрибутов, для которых не указан тип данных, задайте тип данных сами, исходя из той информации, которая должна храниться в данном поле.

В таблице ФАКУЛЬТЕТ задайте первичный ключ «Код факультета».

В таблице ГРУППА задайте первичный ключ «Номер группы», поля «Специальность» и «Факультет» установите как обязательные.

В таблице СТУДЕНТ задайте первичный ключ «Номер зачетки», поля «Фамилия», «Имя», «Дата рождения» и «Номер группы» установите как обязательные.

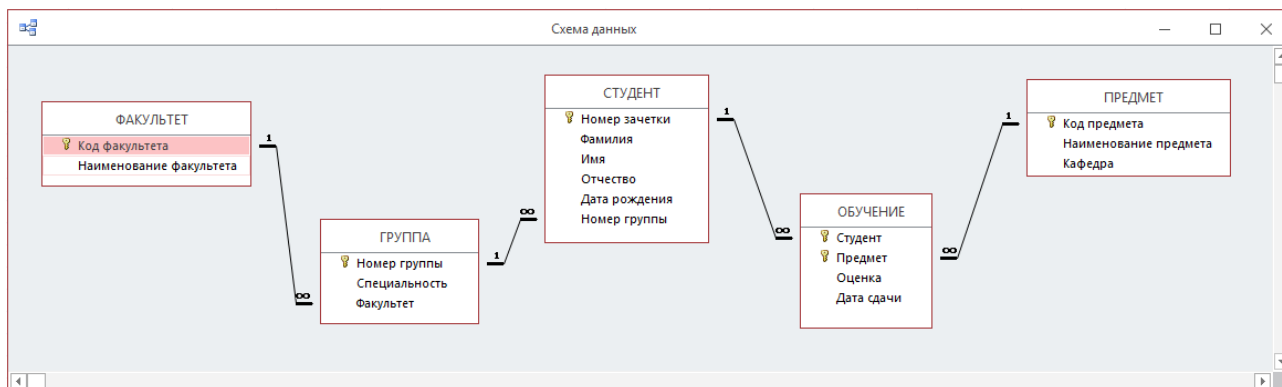
В таблице ПРЕДМЕТ задайте первичный ключ «Код предмета»

В таблице ОБУЧЕНИЕ задайте первичный составной ключ «Студент, Предмет»

Для создания первичного ключа необходимо в режиме конструктора таблицы выделить один или несколько атрибутов и нажать кнопку «Ключевое поле».

3. Откройте схему данных (на вкладке «Работа с базами данных»).

Добавьте на схему данных все созданные таблицы и установите между ними связи, как представлено на рисунке:



При создании связей установите свойство «Обеспечение целостности данных» для всех связей, как показано на рисунке:

The dialog box shows the relationship between 'ФАКУЛЬТЕТ' and 'ГРУППА' on the 'Факультет' attribute. The 'Обеспечение целостности данных' (Data integrity) checkbox is checked. Below it, 'каскадное обновление связанных полей' (cascade update) and 'каскадное удаление связанных записей' (cascade delete) are unchecked. The 'Тип отношения' (Relationship type) is set to 'один-ко-многим' (one-to-many).

4. Снова откройте таблицу «Группа» в режиме конструктора и настройте подстановку для поля внешнего ключа «Факультет» следующим образом:

Общие	Подстановка
Тип элемента управления	Поле со списком
Тип источника строк	Таблица или запрос
Источник строк	ФАКУЛЬТЕТ
Присоединенный столбец	1
Число столбцов	2
Заголовки столбцов	Нет
Ширина столбцов	0см
Число строк списка	16
Ширина списка	Авто
Ограничиться списком	Да
Разрешить изменение списка	Нет
Форма изменения элемента	
Только значения источника	Да

Для правильной настройки подстановки необходимо:

- 1) Задать тип элемента управления «Поле со списком»;
- 2) Задать тип источника строк «Таблица или запрос»;
- 3) В качестве источника строк необходимо выбрать ту таблицу, данные из которой необходимы для заполнения данного поля. Эта таблица находится на стороне «один» связи «один-ко-многим», связывающей исходную таблицу (для которой определяется подстановка) и таблицу – источник строк.
- 4) В качестве присоединенного столбца необходимо выбрать порядковый номер ключевого столбца в таблице – источнике строк.
- 5) Число столбцов необходимо задать таким образом, чтобы в него входил как столбец с первичным ключом таблицы-источника строк, так и все столбцы, которые необходимо отобразить в поле со списком пользователю системы. В данном случае число столбцов равно двум.
- 6) Ширина столбцов выбирается таким образом, чтобы скрыть не существенные для пользователя столбцы, установив их ширину в 0. В данном случае ширина 0см установлена для первого столбца таблицы «ФАКУЛЬТЕТ». Таким образом, в раскрывающемся списке пользователь будет видеть только наименования факультетов. Если бы наименование факультетов находилось в первом столбце, а код факультета - во втором, для того чтобы скрыть код факультета, было бы необходимо установить ширину столбцов следующим образом: 3см; 0см

Настройте аналогичным образом подстановку для полей «Группа» (источник данных – таблица «ГРУППА») в таблице «СТУДЕНТ», а также «Студент» (источник данных – таблица «СТУДЕНТ») и «Предмет» (источник данных – таблица «ПРЕДМЕТ») в таблице «ОБУЧЕНИЕ», правильно выбрав параметры подстановки.

5. Разработайте формы для таблиц «Факультет», «ГРУППА», «ПРЕДМЕТ», «СТУДЕНТ». Проще всего создать форму следующим образом. Закройте все открытые вкладки с конструкторами таблиц и со схемой данных. Выделите в инспекторе объектов слева таблицу, для которой необходимо создать форму. В пункте меню «Создание» нажмите на кнопку «Форма». Форма создастся

автоматически. Можно также воспользоваться мастером создания форм или конструктором форм. В форме «Факультет» удалите подчиненную форму (с группами), поле с надписью «Код факультета», а также надпись для поля «Наименование факультета» (только надпись, но не само поле). В свойствах формы установите режим по умолчанию «Ленточные формы», а высоту самой формы сделайте минимальной, немного превышающей высоту поля «Наименование факультета».

6. С помощью диспетчера кнопочных форм (вкладка «Работа с базой данных» или найдите её с помощью функции «Поиск») разработайте главную кнопочную форму, которая должна открываться автоматически при открытии базы данных и содержать кнопки для открытия созданных ранее форм в режиме изменения. Для этого воспользуйтесь кнопкой «Изменить» на странице «Главная кнопочная форма (по умолчанию)», а затем кнопкой «Создать», выбрав соответствующую команду в раскрывающемся списке команд.
7. Заполните с помощью форм несколько записей в каждой таблице (в таблице «ФАКУЛЬТЕТ» – не менее 2-х записей, в таблице «ГРУППА» – не менее 3-х записей, в таблице «ПРЕДМЕТ» – не менее 4-х записей, в таблице студент – не менее 10 записей о студентах из 3-х групп). При заполнении одиночных форм перемещайтесь по записям с помощью навигатора в левой нижней части окна:



Здесь «1 из 2» означает номер текущей записи из общего количества записей. Следующий значок обозначает переход к следующей записи. Следующий значок обозначает переход к последней записи, и наконец, последний значок – создание новой записи.

8. Разработайте следующие отчеты с помощью «Мастера отчетов»:
 - а. По предметам, включив в него поле «Наименование предмета» из таблицы «ПРЕДМЕТ», поле «Оценка» из таблицы «ОБУЧЕНИЕ», поля «Фамилия», «Имя», «Отчество», «Номер группы» из таблицы «СТУДЕНТ». Установите дополнительный уровень группировки по номеру группы, сортировку по фамилии и устраивающий вас макет отчета.
 - б. По студентам, включив в него поля «Фамилия», «Имя», «Отчество», «Номер группы» из таблиц «СТУДЕНТ», «Оценка», «Дата сдачи» из таблицы «ОБУЧЕНИЕ» и «Наименование предмета» из таблицы «ПРЕДМЕТ», установив в отчете группировку по номеру группы.
 - с. По группам, включив в него поля Номер группы», «Специальность» из таблицы «ГРУППЫ», а также «Номер зачетки», «Фамилия», «Имя», «Отчество», из таблиц «СТУДЕНТ». Измените в конструкторе макет отчета таким образом, чтобы информация

помещалась и легко воспринималась на листе А4 портретной ориентации. Также измените группировку таким образом, чтобы итог по количеству студентов в каждой группе отображался в примечании группы.

9. Добавьте кнопку для открытия разработанных отчетов на соответствующие формы (в примечание формы).
10. Добавьте еще одну страницу для отчетов в кнопочную форму, разместив на ней кнопки для открытия созданных отчетов. На странице «Главная кнопочная форма» добавьте еще одну кнопку для открытия страницы отчетов. Измените надпись (в верхней части кнопочной формы) и рисунок (в левой части) на форме так, чтобы они соответствовали предметной области разработанного приложения.

Результатом работы являются файл базы данных и краткий отчет, в котором необходимо указать краткие характеристики информационной системы: размер файла базы данных, количество и наименования таблиц, форм, отчетов, общее количество записей в базе данных. В отчет также необходимо включить скриншот окна Microsoft Access с главной кнопочной формой, а также первую страницу каждого разработанного отчета.