

УДК 004.9
ББК 32.973
К 63

Автор-составитель С. М. Мовшович, канд. техн. наук, доцент

Рецензенты: О. А. Кравченко, канд. физ.-мат. наук, доцент кафедры информационных технологий Гомельского государственного технического университета им. П. О. Сухого;
Л. П. Авдашкова, канд. физ.-мат. наук, доцент кафедры информационно-вычислительных систем Белорусского торгово-экономического университета потребительской кооперации

Рекомендовано научно-методическим советом учреждения образования «Белорусский торгово-экономический университет потребительской кооперации». Протокол № 2 от 13 декабря 2011 г.

К 63 **Компьютерные** информационные технологии. Технологии баз данных и знаний : пособие для студентов всех специальностей заочной формы получения высшего образования / авт.-сост. : С. М. Мовшович. – Гомель : учреждение образования «Белорусский торгово-экономический университет потребительской кооперации», 2012. – 44 с.
ISBN 978-985-461-934-7

УДК 004.9
ББК 32.973

ISBN 978-985-461-934-7

© Учреждение образования «Белорусский торгово-экономический университет потребительской кооперации», 2012

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Пособие предназначено для студентов заочной формы получения высшего образования. В пособии изложены краткие теоретические сведения и представлены две лабораторные работы по разделу «Технологии баз данных и знаний» курса «Компьютерные информационные технологии» для студентов всех специальностей.

Теоретический материал пособия включает следующие вопросы:

- свойства и характеристики экономической информации;
- информационное обеспечение автоматизированных информационных систем;
- определение и общие свойства баз данных;
- основные понятия реляционной модели данных;
- этапы проектирования базы данных;
- правила логического проектирования.

В пособии подробно рассмотрен пример логического проектирования базы данных, предназначенной для автоматизации работы управленческого персонала фирмы по ремонту бытовой техники. Данный пример иллюстрирует практическое использование сформулированных правил логического проектирования

Каждая лабораторная работа предусматривает проектирование базы данных, заполнение ее таблиц исходной информацией и создание запросов на выборку. Описание первой лабораторной работы сопровождается подробными пояснениями и рисунками, иллюстрирующими процесс логического проектирования и технологию работы в СУБД Access.

В качестве заданий для самостоятельной работы студентам предлагается создать следующие виды запросов к базе данных:

- простой запрос на выборку;
- запрос на выборку с параметром;
- запрос с вычисляемым полем;
- перекрестный запрос.

1. СВОЙСТВА И ХАРАКТЕРИСТИКИ ЭКОНОМИЧЕСКОЙ ИНФОРМАЦИИ

Существует несколько определений экономической информации (ЭИ), практически эквивалентных друг другу. Приведем некоторые из них:

1. Экономическая информация – это информация, характеризующая производственные отношения в обществе.

2. Экономическая информация – совокупность данных, используемых при осуществлении функций организационно-экономического управления экономикой государства и ее отдельными звеньями.

К экономической информации предъявляются следующие требования:

- *точность* обеспечивает ее однозначное восприятие всеми потребителями;

- *достоверность* определяет допустимый уровень искажения как поступающей, так и результативной информации, при которой сохраняется эффективность функционирования системы;

- *оперативность* отражает актуальность информации для необходимых расчетов и принятия решений в изменяющихся условиях.

Экономическая информация классифицируется следующим образом:

1. В зависимости от участия ЭИ в функциях управления. С этой точки зрения ЭИ подразделяется на следующие виды:

- плановая;
- оперативно-учетная;
- нормативно-справочная;
- отчетно-статистическая.

Плановая ЭИ включает директивные значения планируемых и контролируемых показателей бизнес-процессов на некоторый период в будущем: план выпуска продукции в натуральном и стоимостном выражении, планируемый спрос и прибыль, план приема в университет и т. д.

Оперативно-учетная ЭИ отражает фактические значения экономических показателей за определенный период времени: количество выпущенных деталей, начисление сумм в ведомости на получение заработной платы, данные о приеме студентов на первый курс.

Примерами документов, содержащих оперативно-учетную информацию, являются следующие: товарно-транспортная накладная, экзаменационная ведомость, ведомость на получение заработной платы, акт на списание материалов, банковская выписка о движении денежных средств на счете за последний период.

Нормативно-справочную ЭИ составляют различные справочные и нормативные данные, связанные с производственными процессами и отношениями: расценки, тарифы, справочные данные о поставщиках, список возможных оценок.

Отчетно-статистическая ЭИ отражает результаты фактической деятельности фирмы, представляемые вышестоящим органам управления, налоговой инспекции: годовой бухгалтерский отчет о деятельности фирмы.

На практике, в том числе на лабораторных занятиях по курсу «Компьютерные информационные технологии», выделяют только два из перечисленных вида ЭИ – нормативно-справочную и оперативно-учетную.

2. В зависимости от места возникновения. С этой точки зрения ЭИ подразделяется на входную и выходную информацию.

Входная информация – это информация, поступающая в фирму (подразделение, рабочее место специалиста) и используемая как первичная информация для реализации задач управления.

Выходная информация – информация, подготовленная фирмой, подразделением или специалистами и передаваемая в другое звено системы управления.

Ясно, что одна и та же информация может выступать и как входная, и как выходная.

Форма представления ЭИ может быть текстовая, графическая, табличная.

Носителем ЭИ может выступать бумага, магнитный носитель, изображение на экране, звуковое сообщение и т. д.

2. ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ АВТОМАТИЗИРОВАННЫХ ИНФОРМАЦИОННЫХ СИСТЕМ

Автоматизированная информационная система (АИС) – совокупность информации, экономико-математических методов и моделей, технических, программных и технологических средств и специалистов, предназначенная для обработки информации и принятия управленческих решений.

В информационном обеспечении АИС можно выделить следующие элементы:

- системы показателей;
- системы классификации и кодирования;

- документация;
- варианты организации документооборота;
- файлы.

Информационное обеспечение (АИС) подразделяется на немашинное и внутримашинное. Первые четыре из перечисленных выше элементов относятся к немашинному информационному обеспечению, т. е. существующему вне компьютеров и вне зависимости от компьютеров. Файлы – это внутримашинное информационное обеспечение (АИС), т. е. данный элемент хранится в памяти компьютера.

Знание структурных единиц немашинной информации необходимо для ее отображения в базах данных, т. е. во внутримашинном представлении.

Выделяют следующие виды структурных единиц немашинной ЭИ:

1. *Реквизит* – элементарная неделимая на смысловом уровне единица информации, выражающая определенные свойства объекта. Если реквизит дает качественную характеристику объекта или явления, то его называют реквизит-признак (ФИО, место действия, наименование товара и т. д.). Реквизит-основание дает количественную характеристику объекта или явления и выражается в определенных единицах измерения (вес, стоимость, размер, ставка налога и т. д.).

2. *Составная единица информации* – логически взаимосвязанная совокупность реквизитов (ФИО студента + № группы + дисциплина + оценка).

3. *Показатель* – минимальная составная единица информации, отражающая данные одного объекта. Показатель содержит, как правило, один реквизит-основание и несколько реквизитов-признаков (ФИО студента + оценка, наименование товара + вес).

4. *Документ* – составная единица информации, имеющая самостоятельное значение или, другими словами, документ, содержащий совокупность показателей (ГТН, экзаменационная ведомость).

Форма документа отображает структуру информации, определяет состав, название и размещение реквизитов, входящих в документ.

Для проектирования и создания базы данных очень важно знать, что в документе выделяют две части: заголовочную (шапка), содержательную (табличная).

В шапке документа указываются данные, общие для всего документа: номер и дата составления, название документа и пр.

Содержательная часть включает названия реквизитов и место, отведенное для их значений. Содержательная часть, как правило, имеет

вид многострочной таблицы с наименованиями реквизитов в заголовке колонок и множественном значении реквизита в столбце.

3. БАЗЫ ДАННЫХ. ОПРЕДЕЛЕНИЕ И ОБЩИЕ СВОЙСТВА

Существует несколько определений базы данных (БД), которые практически эквивалентны. Приведем некоторые из них:

1. Базы данных – это совместно используемый набор логически связанных данных и описание этих данных, предназначенный для удовлетворения информационных потребностей организации.

2. Базы данных – совокупность данных о реальных объектах, процессах, событиях или явлениях, относящихся к определенной теме или задаче, и организованная таким образом, чтобы обеспечить удобное представление этой совокупности как в целой, так и любой ее части.

Базы данных характеризуются следующими свойствами:

1. Базы данных – это большое хранилище данных, которое однократно определяется и многократно используется многими пользователями из различных подразделений организации.

2. Данные в БД собраны вместе с *минимальной долей избыточности*. Избыточность – это повторение одних и тех же данных в различных частях одной БД.

3. База данных не принадлежит одному подразделению организации, а является *общим корпоративным ресурсом*.

4. База данных хранит не только данные, но и их *описания*. Совокупное описание данных называется системным каталогом или словарем данных, элементы словаря называются *метаданными*.

5. Данные в БД *отделены от приложений*, т. е. от программ, разработанных для нужд конкретных пользователей (например, программ печати какого-то специального отчета). Это означает, что структура данных может быть изменена (например, добавлены новые поля), а пользователь этого не заметит, т. е. приложение, которое он использует, будет по-прежнему правильно работать. Однако если в БД будут удалены некоторые поля, то необходимо внести изменения и в приложение.

6. Связи между данными в БД должны обеспечивать *целостность данных*. Целостность данных – это их корректность. Пример: если в справочнике студентов отсутствует студент с шифром 48 000, то ни одна таблица БД не должна содержать сведения об этом студенте. Целостность данных обеспечивается как программными средствами (это

одна их функций СУБД), так и организационно-административными мерами, устанавливаемыми для пользователей данной организации.

4. МОДЕЛИ ДАННЫХ

Модель данных – это интегрированный набор понятий для описания данных, связей между ними и ограничений, накладываемых на данные в некоторой организации.

Цель построения модели данных – представление данных в удобном и понятном (как для пользователей, так и для разработчиков) виде.

Существует три модели данных: сетевая, иерархическая, реляционная.

Сетевая модель – данные представляются в виде коллекций записей, а связи между ними организуются в виде указателей. При этом, одна запись может указывать на несколько подчиненных ей записей и на одну запись могут указывать несколько стоящих над ней записей.

Иерархическая модель – ограниченный подтип сетевой модели, когда на каждую запись может указывать только одна вышестоящая запись, но каждая запись может иметь в подчинении несколько записей.

При *реляционной модели* данные и связи представляются в виде таблиц. Связь не выражается явно (т. е. в виде указателя), а ее можно увидеть по одинаковым атрибутам в разных таблицах.

Далее речь будет идти только о реляционной модели, принятой в подавляющем большинстве реально работающих систем.

5. ОСНОВНЫЕ ПОНЯТИЯ РЕЛЯЦИОННОЙ МОДЕЛИ ДАННЫХ

Реляционная БД – это набор отношений. С ней связаны следующие понятия:

1. *Отношение* – плоская таблица, состоящая из столбцов и строк.
2. *Атрибут* – поименованный столбец отношения.
3. *Домен* – набор допустимых значений для одного или нескольких атрибутов. Например, для атрибутов Телефон домашний и Телефон служебный существует один домен – набор из 6 цифр (для г. Гомеля).
4. *Кортеж* – строка отношения.

Альтернативные варианты терминов реляционной модели представлены в таблице 1.

Таблица 1 – Основные термины реляционной модели

Официальный термин	Альтернативный вариант 1	Альтернативный вариант 2
Отношение	Таблица	Файл
Кортеж	Строка	Запись
Атрибут	Столбец	Поле

В каждой таблице есть *первичный ключ* – это минимальный набор атрибутов (или один атрибут) отношения, который может быть использован для однозначной идентификации конкретного кортежа. Если первичный ключ состоит из одного атрибута, то он называется *простым первичным ключом*. Если первичный ключ состоит из двух или более атрибутов, то он называется *составным первичным ключом*.

Из приведенного определения следует, что значения первичного ключа не могут повторяться в данной таблице, и, наоборот, если в каком-то атрибуте таблицы есть повторяющиеся значения, то этот атрибут не может быть простым первичным ключом таблицы.

Во всех кортежах первичного ключа обязательно должны быть значения. Это свойство называется *целостностью сущности* базы данных.

Степень отношения определяется количеством атрибутов, которое оно содержит.

Кардинальность отношения – количество кортежей, которое содержит отношение.

Атрибут или набор атрибутов в таблице *A* называется *внешним ключом*, если он является первичным ключом в другой таблице *B*. В этом случае говорят, что таблицы *A* и *B* *связаны* по данному атрибуту, причем для рассматриваемой *связи* таблица *B* является *главной*, а таблица *A* – *подчиненной*. Также говорят, что таблица *A* ссылается на таблицу *B* по внешнему ключу.

Связь между первичным ключом таблицы *B* и внешним ключом таблицы *A* имеет тип «*один-ко-многим*» в том случае, когда конкретное значение атрибута, по которому организована связь, может встретиться в таблице *B* только один раз, а это же значение в таблице *A* может находиться в нескольких строках.

Если в таблице имеется внешний ключ, то его значение должно соответствовать значению первичного ключа в каком-либо кортеже главной таблицы. Другими словами, в таблице *A* не должно быть таких значений внешнего ключа, которых нет в первичном ключе главной таблицы *B*. Это свойство называется *ссылочной целостностью* базы данных.

Если база данных обладает целостностью сущностей и ссылочной целостностью, то говорят, что эта база данных обладает *реляционной целостностью*.

Кроме связи «*один-ко-многим*» существуют связи «*один-к-одному*» и «*многие-ко-многим*», но последние два типа довольно редко используются в БД.

6. ЭТАПЫ ПРОЕКТИРОВАНИЯ БД

Создание БД в среде системы управления базами данных Access (СУБД Access) предполагает выполнение следующих основных этапов:

1. Концептуальное проектирование.
2. Логическое проектирование.
3. Физическое проектирование.
4. Использование БД – заполнение БД оперативной информацией и формирование запросов и отчетов.

Концептуальное проектирование – процедура конструирования информационной модели предметной области, не зависящей от условий реализации БД. Таким образом, сконструированная на данном этапе информационная модель не зависит ни от СУБД, ни от средств вычислительной техники.

Концептуальное проектирование БД выполняется на основе:

- анализа информационных потоков предприятия;
- использования классификаторов и систем кодирования;
- определения области применения БД;
- выяснения состава ее пользователей;
- сбора и анализа требований пользователей.

В нашем курсе не рассматривается методика проведения концептуального проектирования. Мы будем считать его выполненным и, таким образом, предполагается, что к моменту начала логического проектирования БД сконструирована информационная модель рассматриваемой предметной области.

На этапе логического проектирования информационная модель предметной области уточняется с учетом реляционной модели БД. Здесь, в частности, определяются отношения и атрибуты.

Процесс физического проектирования БД предполагает выполнение в среде выбранной СУБД следующих работ:

- 1) описание логической структуры каждой таблицы;
- 2) описание связей между таблицами, входящими в одну БД;

3) первоначальное заполнение справочников БД необходимой нормативно-справочной информацией.

7. ЛОГИЧЕСКОЕ ПРОЕКТИРОВАНИЕ

В результате логического проектирования необходимо определить следующие объекты базы данных:

- набор таблиц, составляющих базу данных;
- набор атрибутов каждой таблицы;
- первичный ключ каждой таблицы;
- домен для каждого атрибута;
- связи между таблицами.

Процесс логического проектирования можно проводить в соответствии со следующими упрощенными правилами:

1. *Определяются набор сущностей, информация о которых должна быть в разрабатываемой базе данных, их атрибуты и ключи.*

Сущность – это объект реального мира или бизнес-процесс. К сущностям можно отнести следующее: товары, сотрудники, процесс закупок, товарно-транспортная накладная и т. п. Каждая сущность состоит из экземпляров (например, конкретные товары) и характеризуется различными атрибутами (например, наименование товара, единица измерения, габариты и т. д.). Среди атрибутов должен быть ключевой атрибут (или совокупность атрибутов). Значения ключевого атрибута не повторяются и служат для однозначной идентификации экземпляров сущности.

2. *Как правило, для каждой сущности, не являющейся документом, в базе данных выделяют одну таблицу (одно отношение). Атрибуты сущности становятся атрибутами таблицы, а ключевой атрибут сущности – первичным ключом таблицы.*

3. *Если сущность представляет собой документ, то для нее выделяют в БД две таблицы: одну для информации из заголовочной части, а вторую – из содержательной, при этом во вторую таблицу включают атрибут, являющийся первичным ключом первой таблицы. Таким образом, во второй таблице появляется внешний ключ, обеспечивающий связь между двумя таблицами.*

4. *Если сущность А через некоторый атрибут ссылается на сущность В, то в таблицу, соответствующую сущности А, надо для связи включить первичный ключ таблицы В. Другими словами, в таблицу А помещают внешний ключ для связи с таблицей В и не включают атрибуты таблицы В, которые определяются (зависят) от ключа.*

Таким образом, связь между таблицами *A* и *B* организуется только по одному атрибуту, являющемуся первичным ключом в таблице *B* и внешним ключом в таблице *A*.

5. В таблицы БД не включают атрибуты для вычисляемых реквизитов документов, так как их значения в любой момент могут быть определены при выполнении запроса к БД.

8. ПРИМЕР ЛОГИЧЕСКОГО ПРОЕКТИРОВАНИЯ

Задание. В фирме по ремонту бытовой техники регистрируются и выписываются клиентам квитанции. Пример квитанции представлен на рисунке 1.

Квитанция № 175 от 12.04.2011 г.				
Клиент ОАО «Мир»				
Наименование бытовой техники	Вид ремонта	Количество	Тариф	Стоимость
ПК IP 2,6/2/64	Устранение обрыва кабеля	1	1500	1500
ПК IP 2,6/2/64	Замена чипа памяти	2	20000	40000
Холодильник «Атлант»	Регулировка ножек	4	500	2000
			Итого	43500
			За срочность (10%)	4350
			Всего к оплате	47850
			Отметка об оплате	Оплачено

Рисунок 1 – Пример квитанции, составляемой в фирме по ремонту бытовой техники

При выполнении задания необходимо учесть следующее:

- номера квитанций не повторяются за весь период учета;
- квитанция выписывается одному клиенту;
- в одной квитанции могут быть упомянуты несколько видов техники и видов ремонта;

- один и тот же вид бытовой техники может быть упомянут в одной квитанции с различными видами ремонта.

Требуется спроектировать БД *Ремонт бытовой техники*, которая будет содержать всю информацию, указанную в квитанциях. В процессе использования БД к ней могут обращаться, например, со следующими запросами:

- выдать список должников на заданное число;
- получить общую сумму задолженности на заданное число;
- вычислить доход за определенный период.

Порядок выполнения задания

1. Определение набора таблиц и их атрибутов

В соответствии с первым правилом логического проектирования необходимо выделить сущности, информация о которых должна находиться в БД.

На основе вида квитанции и примеров запросов к БД можно выделить сущность *Бытовая техника*, которая должна содержать сведения обо всех объектах, принимаемых в ремонт данной фирмой. В бланке квитанции приведен только один атрибут этой сущности – *Наименование бытовой техники*, но, по сути, здесь соединены два атрибута: собственно наименование (холодильник) и модель («Атлант»). Поэтому целесообразно в сущность включить два указанных атрибута. Атрибут *Наименование бытовой техники* не может быть использован в качестве первичного ключа сущности, т. к. могут встретиться объекты ремонта с одним наименованием и разными моделями (холодильники «Атлант» и «Минск»). С целью удобства однозначной идентификации различных объектов ремонта введем атрибут *Шифр бытовой техники*. Необходимо придумать процесс кодирования, т. е. присвоения шифров различным объектам. Например, холодильники различных моделей можно кодировать Х001, Х002 и т. д., а печи СВЧ – П001, П002 и т. д. Заметим, что дополнение сущности атрибутом *Шифр* является традиционным приемом проектирования БД.

Таким образом, для сущности *Бытовая техника* характерны следующие атрибуты: *Шифр бытовой техники*, *Наименование бытовой техники*, *Модель*.

Здесь и далее первичный ключ выделен подчеркиванием сплошной линией.

Вторая сущность, информация о которой должна храниться в проектируемой базе данных, – *Виды ремонта*. Эта сущность, по сути, представляет собой прейскурант фирмы, т. е. перечень видов выполняемых работ и их тарифы. Наименование работы может служить первичным ключом сущности, но это неудобно, т. к. оно содержит много символов. Поэтому целесообразно ввести атрибут *Шифр вида ремонта* и разработать процесс кодирования. Таким образом, для сущности *Виды ремонта* характерны следующие атрибуты: *Шифр вида ремонта*, *Наименование вида ремонта*, *Тариф*.

Третья сущность, составляющая базис проектируемой БД, – *Клиенты*. Несмотря на то что в квитанции упомянуто только наименование клиента, можно предположить о необходимости ведения переписки или телефонных переговоров с клиентом. Возможно, данный клиент будет обращаться в фирму неоднократно, а, следовательно, желательно хранить в базе данных максимально большое число данных о клиенте. Будем считать, что для сущности *Клиенты* характерны следующие атрибуты: *Код Клиента*, *Наименование Клиента*, *Адрес*, *Телефон*, *Ответственное Лицо*.

Введение атрибута *Код Клиента* в качестве первичного ключа сущности обусловлено теми же причинами, о которых говорилось для первых двух сущностей.

Последняя сущность, на которой основывается проектируемая БД, – документ *Квитанция*.

Атрибутами данной сущности являются все реквизиты документа: Номер Квитанции, Дата Квитанции, Наименование Клиента, Наименование бытовой техники, Наименование вида ремонта, Количество, Тариф, Стоимость, Итого, Срочность, Всего к оплате, Оплачено.

Атрибут *Номер Квитанции* может быть использован в качестве первичного ключа, т. к. номера квитанций не повторяются за весь период учета.

Выделенным сущностям надо поставить в соответствие таблицы БД.

Первые три сущности не ссылаются на другие сущности, поэтому на основе второго правила проектирования в БД им будут соответствовать следующие таблицы:

1. *Бытовая техника* (*ШифрБТ*, *НаимБТ*, *Модель*).
2. *Виды ремонта* (*ШифрВР*, *НаименВР*, *Тариф*).
3. *Клиенты* (*КодКл*, *НаимКл*, *Адрес*, *Телефон*, *ОтвЛицо*).

Обратите внимание, что названия атрибутов таблицы записаны без пробелов. Это связано с последующей реализацией БД на компьютере – некоторые программные комплексы (системы управления БД) не допускают использование пробелов в именах атрибутов.

Сущность *Квитанция* является документом, поэтому для нее (правило 3 логического проектирования) в БД определим две таблицы – *Квитанция* и *СтрокаКв*.

В первую таблицу следует включить реквизиты заголовочной части документа. К этим реквизитам относятся следующие: *Номер Квитанции*, *Дата Квитанции*, *Наименование Клиента*, *Итого*, *Срочность*, *Всего к оплате*, *Оплачено*. Не все эти реквизиты надо включать в качестве атрибутов таблицы *Квитанция* – реквизиты *Итого* и *Всего к оплате* являются вычисляемыми и поэтому не используются в БД (правило 5 логического проектирования). Реквизит *Наименование Клиента* ссылается на таблицу *Клиенты* и поэтому вместо него в таблицу *Квитанция* надо включить первичный ключ таблицы *Клиенты* атрибут *КодКл* (правило 4). Таким образом, атрибут *НаимКл*, зависящий от первичного ключа *КодКл*, не включается в таблицу *Квитанция*. В результате получаем следующий набор атрибутов четвертой таблицы БД:

4. *Квитанция* (*НомерКв*, *ДатаКв*, *КодКл*, *Срочность*, *Оплачено*).

В таблице *Квитанция* атрибут *КодКл* является внешним ключом. Здесь и далее внешние ключи будут отмечаться штриховым подчеркиванием.

В таблице *СтрокаКв* должны быть атрибуты, соответствующие следующим реквизитам документа: *Наименование бытовой техники*, *Наименование вида ремонта*, *Количество*, *Тариф*, *Стоимость*. Вместо реквизита *Наименование бытовой техники* в соответствии с правилом 4 логического проектирования следует включить первичный ключ таблицы *Бытовая техника* атрибут *ШифрБТ*. Аналогично, вместо реквизитов *Наименование вида ремонта* и *Тариф* надо включить в таблицу *СтрокаКв* атрибут *ШифрВР*, являющийся первичным ключом таблицы *Виды ремонта*. Атрибут *Стоимость* не нужен в таблице *СтрокаКв*, т. к. он является вычисляемым (правило 5). По правилу 3 в таблицу *СтрокаКв* необходимо включить первичный ключ главной таблицы *Квитанция* атрибут *НомерКв*. Таким образом, получим следующий набор атрибутов таблицы *СтрокаКв*:

5. *СтрокаКв* (*НомерКв*, *ШифрБТ*, *ШифрВР*, *Количество*).

Первичным ключом таблицы *СтрокаКв* является совокупность трех атрибутов *НомерКв*, *ШифрБТ* и *ШифрВР* (каждый из которых является внешним ключом), т. к. в одной квитанции один и тот же вид бытовой техники может быть упомянут только с различными видами ремонта.

2. Определение доменов для атрибутов БД

Будем считать, что атрибуты *ШифрБТ*, *ШифрВР* и *КодКл*, введенные нами для однозначной идентификации экземпляров сущностей, имеют *текстовый* тип с длиной, не превышающей семи символов. Правила кодирования в данном примере не рассматриваются и принимаются произвольными.

Атрибуты *НаимБТ*, *Модель*, *НаименВР*, *НаимКл*, *Адрес* и *Отв.Лицо* имеют *текстовый* тип с длиной, не превышающей 50 символов.

Атрибут *Тариф* – *денежный* тип, 0 цифр после запятой (без копеек).

Атрибут *Телефон* – *текстовый* тип, не более 13 цифр.

Атрибут *НомерКв* – *текстовый* тип, не более пяти цифр.

Атрибут *ДатаКв* – тип *Дата и Время*, краткий формат даты.

Атрибут *Срочность* – тип *логический* (*Да* – срочный заказ, *Нет* – обычный заказ).

Атрибут *Оплачено* – тип *логический* (*Да* – оплачено, *Нет* – не оплачено).

Атрибут *Количество* – *числовой* тип, вещественный, две цифры после запятой.

Отметим, что значения атрибутов *ШифрБТ*, *ШифрВР*, *КодКл*, *Телефон* и *НомерКв* хоть и представляются последовательностью цифр, но имеют не числовой, а текстовый тип, т. к. над ними не производятся арифметические операции.

3. Установление связей между таблицами

В спроектированных таблицах БД имеется четыре внешних ключа. Следовательно, устанавливаются следующие связи между таблицами типа «один-ко-многим»:

1. Таблицы *Клиенты* и *Квитанция* связаны по атрибуту *КодКл*.

2. Таблицы *Квитанция* и *СтрокаКв* связаны по атрибуту *НомерКв*.

3. Таблицы *Бытовая техника* и *СтрокаКв* связаны по атрибуту *ШифрБТ*.

4. Таблицы *Виды ремонта* и *СтрокаКв* связаны по атрибуту *ШифрВР*.

На рисунке 2 приведена спроектированная структура искомой БД. Первичные ключи таблиц (за исключением трех атрибутов таблицы *СтрокаКв*) выделены полужирным шрифтом, тип связи «один-ко-многим» обозначен символами 1 и ∞ на концах линии связи.



Рисунок 2 – Структура базы данных Ремонт бытовой техники

9. СИСТЕМА УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ

Система управления базами данных (СУБД) – это совокупность программных и языковых средств для создания, ведения и совместного использования базы данных многими пользователями. СУБД поддерживает один из возможных типов моделей данных – сетевую, иерархическую или реляционную, обеспечивает многоцелевой характер использования базы данных, защиту и восстановление данных. Наличие развитых диалоговых средств и языка запросов высокого уровня делает СУБД удобным средством для пользователя.

Основными средствами СУБД являются следующие:

- средства задания (описания) структуры базы данных;
- средства конструирования экранных форм, предназначенных для удобного ввода данных, просмотра и их обработки в диалоговом режиме;
- средства создания запросов для выборки данных при заданных условиях, а также выполнения операций по их обработке;
- средства создания отчетов из базы данных для вывода на печать результатов обработки в удобном для пользователя виде;
- языковые средства – макросы, встроенный алгоритмический язык (например, Visual Basic), язык запросов (например, QBE, SQL), которые используются для реализации нестандартных алгоритмов обработки данных, а также процедур обработки событий в задачах пользователя;
- средства создания приложений пользователя (генераторы приложений, средства создания меню и панелей управления приложениями), позволяющие объединить различные операции работы с базой данных в единый технологический процесс.

В своей работе СУБД используют следующие группы специалистов:

1. *Администраторы данных* отвечают за управление данными организации (предприятия). Они выполняют следующие виды работ:

- планирование БД;
- разработка и сопровождение стандартов организации;
- руководство концептуальным и логическим проектированием БД;
- консультации и рекомендации руководству высшего звена организации;
- контроль соответствия направления развития БД корпоративным целям.

2. *Администраторы базы данных* отвечают за физическую реализацию БД, т. е. они курируют следующие вопросы:

- физическое проектирование БД;
- безопасность и целостность данных;
- сопровождение операционной системы;
- обеспечение максимальной производительности приложений и пользователей.

Обязанности администраторов БД носят более технический характер по сравнению с обязанностями администраторов данных.

3. *Разработчики БД* разделяются на разработчиков логической БД, разработчиков физической БД.

Первые должны обладать всесторонним и полным пониманием структуры данных фирмы и ее бизнес-правил, т. е. правил, описывающих основные характеристики данных с точки зрения фирмы.

Вторые получают готовую логическую модель и занимаются ее физической реализацией. Разработчики физической БД выполняют следующие виды работ:

- преобразование логической модели данных в набор таблиц и ограничений целостности данных;
- выбор конкретных структур хранения и методов доступа к данным, обеспечивающих необходимый уровень производительности при работе с БД;
- проектирование любых требуемых мер защиты данных.

4. *Прикладные программисты* разрабатывают приложения, предоставляющие пользователям необходимые им функциональные возможности.

5. *Пользователи-клиенты БД* выполняют следующие виды работ:

- заполнение таблиц БД нормативно-справочной и оперативно-учетной информацией;

- корректировка информации в таблицах БД;
- создание и выполнение запросов к БД;
- формирование и получение отчетов.

10. КРАТКАЯ ХАРАКТЕРИСТИКА СУБД MS ACCESS

СУБД Access входит в состав пакета Microsoft Office и представляет собой мощное средство для хранения и обработки данных. Основное назначение СУБД Access – создание баз данных и управление ими. С помощью СУБД Access можно выбрать из БД разнообразную информацию для просмотра и анализа, представить данные на экране в виде таблиц или специально сконструированных форм, вывести на принтер отчеты, выполнить различные вычисления в процессе отбора данных и подготовки отчетов.

Одним из достоинств СУБД Access является возможность создания баз данных без знания программирования. Система предлагает два основных режима работы:

- конструирование БД с максимальным учетом требований пользователя;
- использование программ-мастеров, создающих БД по определенному алгоритму.

Как и все приложения пакета Microsoft Office, для выполнения многих действий СУБД Access предлагает альтернативные способы:

- основное меню;
- кнопки панели инструментов;
- контекстное меню;
- комбинации клавиш.

База данных в СУБД Access – некоторая совокупность информации, организованная и упорядоченная определенным образом и хранящаяся в одной или нескольких связанных между собой таблицах.

Файл БД служит для хранения данных, сгруппированных в таблицах, и объектов СУБД Access. Файл имеет расширение *.mdb.

К объектам СУБД Access относятся следующие:

1. *Таблица* – хранилище данных, основной объект БД. Для совместной работы с данными из нескольких таблиц устанавливаются связи между таблицами.

2. *Запрос* – объект БД, который позволяет проанализировать данные с помощью групповых операций, вычислений и отбора данных, удовлетворяющих некоторым условиям.

3. *Форма* – объект БД, который обеспечивает представление данных на экране в виде, удобном пользователю. Формы позволяют вводить, просматривать, изменять данные и являются средством поиска данных.

4. *Отчет* – объект БД, предназначенный для отображения данных при выводе на печать.

5. *Макрос* – набор из одной или нескольких макрокоманд, обеспечивающих автоматическое выполнение процедур по управлению базой данных, ее объектами и данными. Макросы часто привязываются к событиям, возникающим при определенных действиях пользователя.

6. *Модуль* – средство для работы с программой (приложением), написанной на языке VBA (Visual Basic for Application).

Конструктор – режим создания объектов БД. Конструктор имеет свое окно, в котором расположен бланк для задания параметров объектов БД.

Мастер – программа пошагового решения определенной узкой задачи, например, создания объектов БД. Мастер ведет диалог с пользователем с целью получения необходимых для решения задачи сведений.

11. ЭТАПЫ РАБОТЫ В СУБД ACCESS

При создании новой БД в СУБД Access надо выполнить следующие действия:

1. Запуск СУБД Access с выбором режима создания новой БД и указанием пути сохранения файла БД.

2. Описание таблиц БД (создание структуры таблиц) с определением для каждого поля типа, значения размера и формата, для каждой таблицы указывается ключевое поле (ключевые поля) и для внешних ключей указываются параметры подстановки его значений из соответствующего ключевого поля другой таблицы.

3. Установление связей между таблицами.

4. Заполнение таблиц нормативно-справочной и оперативно-учетной информацией.

5. Использование БД, т. е. создание запросов и отчетов.

При описании таблиц и заполнении их информацией следует иметь в виду, что сначала проводятся действия с главной таблицей, а затем с подчиненной.

ЛАБОРАТОРНАЯ РАБОТА 1

Цель работы: получение навыков проектирования и реализации базы данных в СУБД Access.

Задание. Создайте базу данных *Акты передачи*, основанную на документе «Акт передачи материальных ценностей», примеры которых представлен на рисунке 3.

Акт № 175 от 12.02.2011 г. передачи материальных ценностей <u>Грушину А.Б.</u> <u>Плановый отдел</u> (ФИО МОЛ) (подразделение)			
Инвентарный номер	Название инвентарного объекта	Наименование инвентарной группы	Балансовая стоимость, р.
ИПО123	Принтер LBP-810	Вычислительная техника	520000
ИПО348	Стул	Мебель	18050
ИПО349	Кресло	Мебель	36800
Акт № 199 от 22.02.2011 г. передачи материальных ценностей <u>Деминой С.Г.</u> <u>Отдел кадров</u> (ФИО МОЛ) (подразделение)			
Инвентарный номер	Название инвентарного объекта	Наименование инвентарной группы	Балансовая стоимость, р.
ИОК023	Телефонный аппарат	Приборы	51000
ИОК074	Стул	Мебель	18050
ИОК101	Светильник	Приборы	8800

Рисунок 3 – Примеры документов «Акт передачи материальных ценностей»

При выполнении задания необходимо учесть следующие обстоятельства (условия применения базы данных):

- номера актов не повторяются на протяжении всего периода учета;
- каждый инвентарный объект идентифицируется уникальным инвентарным номером;
- один и тот же объект может упоминаться в разных актах;
- в одном акте могут быть отражены поступления нескольких объектов, каждый из которых относится к своей инвентарной группе;

- все объекты одного акта принимаются одним материально ответственным лицом (МОЛ);
 - в один день может быть оформлено несколько актов.
- После заполнения таблиц информацией из актов создайте различные запросы на выборку.

Порядок выполнения работы

1. Определение структур таблиц

На основе вида актов и условий применения БД можно выделить следующие сущности и их атрибуты:

- а) *Инвентарные группы* (ШифрИГ, НаимИГ);
- б) *Инвентарные объекты* (ИнвНомер, НазВИО, НаимИГ, БалансСтоим);
- в) *МОЛ* (ТабелНомер, ФИОМОЛ, Подразделение);
- г) *Акты* (НомерАкт, ДатаАкт, ФИОМОЛ, Подразделение, ИнвНомер, НазВИО, НаимИГ, БалансСтоим).

Первичные ключи сущностей подчеркнуты сплошной линией.

Можно выделить еще одну сущность – *Подразделения*, но здесь предполагается, что из БД требуется получать информацию по различным *МОЛ*, а не по подразделениям.

Каждой из первых трех сущностей в проектируемой БД выделим одну таблицу (правило 2 логического проектирования):

1. *Инвентарные группы* (ШифрИГ, НаимИГ).
2. *Инвентарные объекты* (ИнвНомер, НазВИО, БалансСтоим, ШифрИГ).
3. *МОЛ* (ТабелНомер, ФИОМОЛ, Подразделение).

Обратите внимание, что в таблице *Инвентарные объекты* атрибут *НаимИГ* заменен первичным ключом главной таблицы *Инвентарные группы* атрибутом *ШифрИГ* (правило 4 логического проектирования). Таким образом, атрибут *ШифрИГ* является внешним ключом.

Сущность *Акты* является документом, поэтому для нее в БД определим две таблицы – *Акты* и *СтрокиАктов* (правило 3 логического проектирования).

В первую таблицу следует включить в качестве атрибутов реквизиты заголовочной части документа. К этим реквизитам относятся следующие: НомерАкт, ДатаАкт, ФИОМОЛ и Подразделение. Вместо последних двух реквизитов, являющихся ссылками на таблицу *МОЛ*, включим атрибут ТабелНомер (первичный ключ *МОЛ*). Таким образом, атрибут ТабелНомер является внешним ключом таблицы *Акты*. В результате получаем следующий набор атрибутов четвертой таблицы БД.

4. *Акты* (НомерАкт, ДатаАкт, ТабелНомер).

В таблице *СтрокиАктов* должны быть атрибуты, соответствующие следующим реквизитам документа: *ИнвНомер*, *НазвИО*, *НаимИГ*, *БалансСтоим*. Однако в таблицу достаточно включить только атрибут *ИнвНомер*, так как остальные три реквизита однозначно определяются по инвентарному номеру объекта (правило 4). В таблицу необходимо включить атрибут *НомерАкт*, который будет ссылаться на первичный ключ таблицы *Акты* (правило 3). Таким образом, получим следующий набор атрибутов таблицы *СтрокиАктов*:

5. *СтрокиАктов* (НомерАкт, ИнвНомер).

Первичным ключом таблицы *СтрокиАктов* является совокупность двух атрибутов – *НомерАкт* и *ИнвНомер*, каждый из которых является внешним ключом.

2. **Определение доменов для атрибутов БД**

Будем считать, что атрибуты *ШифрИГ* и *ТабелНомер* имеют *текстовый* тип с длиной, не превышающей трех символов. Правила кодирования в данном примере не рассматриваются и принимаются произвольными.

Из приведенных примеров актов следует, что атрибут *ИнвНомер* имеет *текстовый* тип с длиной шесть символов, а *НомерАкт* – *текстовый* тип с длиной четыре символа.

Атрибутам *НаимИГ*, *НазвИО*, *ФИОМОЛ* и *Подразделение*, назначим *текстовый* тип с длиной, не превышающей 50 символов.

Атрибут *БалансСтоим* – *денежный* тип, 0 цифр после запятой (без копеек).

Атрибут *ДатаАкт* – тип *Дата* и *Время*, краткий формат даты.

3. **Определение связей между таблицами**

В спроектированных таблицах БД имеется четыре внешних ключа. Следовательно, устанавливаются следующие связи между таблицами типа «один-ко-многим»:

1. Таблицы *Инвентарные группы* и *Инвентарные объекты* связаны по атрибуту *ШифрИГ*.

2. Таблицы *МОЛ* и *Акты* связаны по атрибуту *ТабелНомер*.

3. Таблицы *Инвентарные объекты* и *СтрокиАктов* связаны по атрибуту *ИнвНомер*.

4. Таблицы *Акты* и *СтрокиАктов* связаны по атрибуту *НомерАкт*.

4. Создание таблиц в СУБД Access

Вызовите СУБД Access и выберите режим создания новой БД и укажите путь сохранения файла БД: *D:\STUD\шифр группы\Акты передачи.mdb*.

В режиме *Таблицы\Создание таблицы в режиме конструктора* (рисунок 4) задайте структуру каждой таблицы БД.

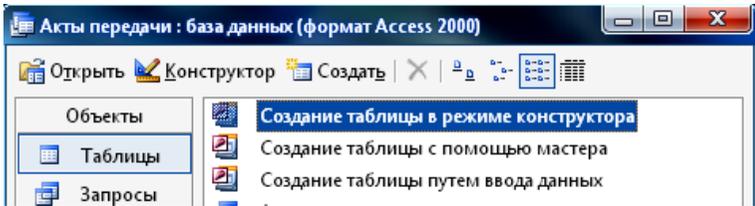


Рисунок 4 – Запуск режима создания таблиц

Для каждой таблицы перечислите атрибуты (поля), выберите тип поля и укажите ключевое поле. Для полей таблиц на вкладке *Общие* укажите значения размера и формата в соответствии с ранее определенными доменами.

На рисунке 5 приведены структуры таблиц *Инвентарные группы* и *Инвентарные объекты*.

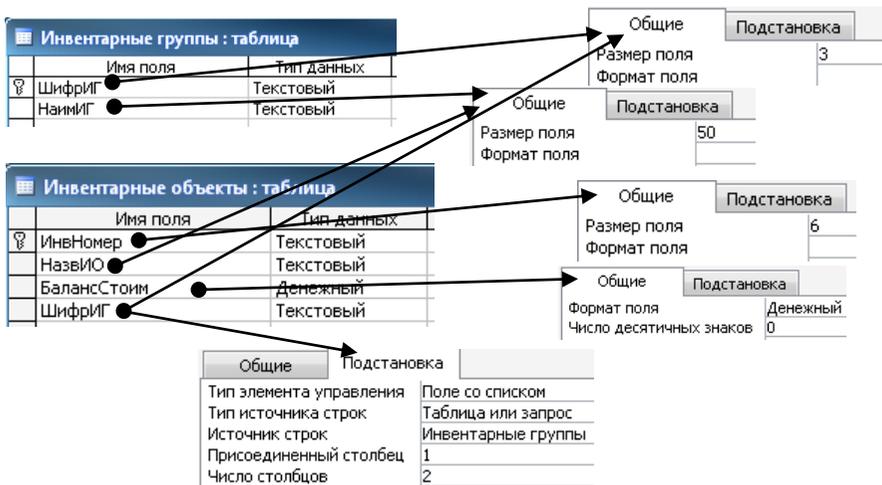


Рисунок 5 – Структура таблиц *Инвентарные группы* и *Инвентарные объекты*

Атрибут *ШифрИГ* таблицы *Инвентарные объекты* надо объявить внешним ключом. Для этого на вкладке *Подстановка* устанавливаются параметры, приведенные на рисунке 5.

На рисунке 6 представлены структуры таблиц *МОЛ*, *Акты* и *СтрокиАктов*.

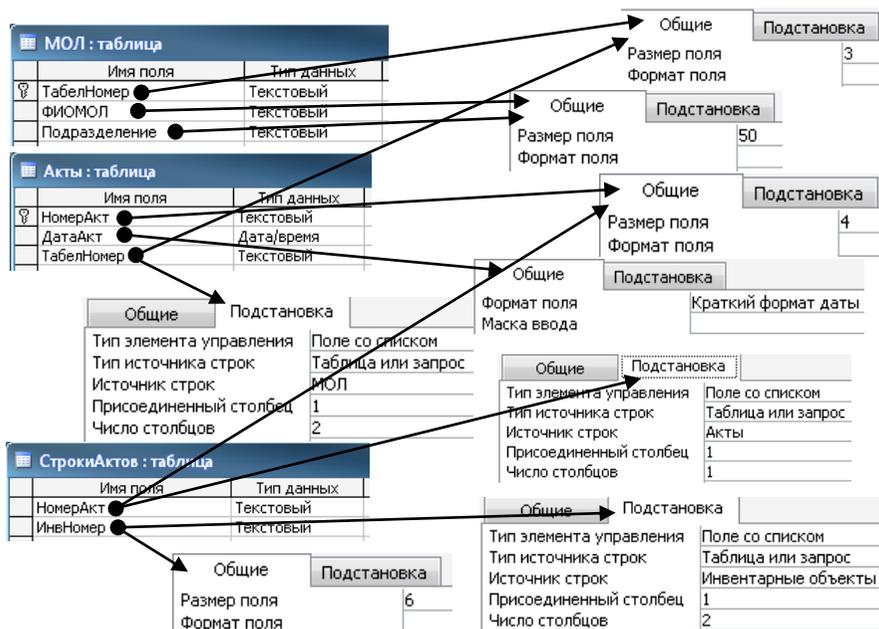


Рисунок 6 – Структура таблиц *МОЛ*, *Акты* и *СтрокиАктов*

5. Установление связей между таблицами в СУБД Access

Выполните команду *Сервис\Схема данных* и в диалоговом окне *Добавление таблицы* (рисунок 7) выберите все созданные таблицы.

Для создания связи между двумя таблицами надо зафиксировать курсор на первичном ключе главной таблицы протянуть его (при нажатой левой кнопки мыши) до соответствующего внешнего ключа подчиненной таблицы. После чего в появившемся диалоговом окне установить два флажка: *Обеспечение целостности данных* и *Каскадное обновление данных* (рисунок 8).

После установления всех связей получим схему данных, приведенную на рисунке 9.

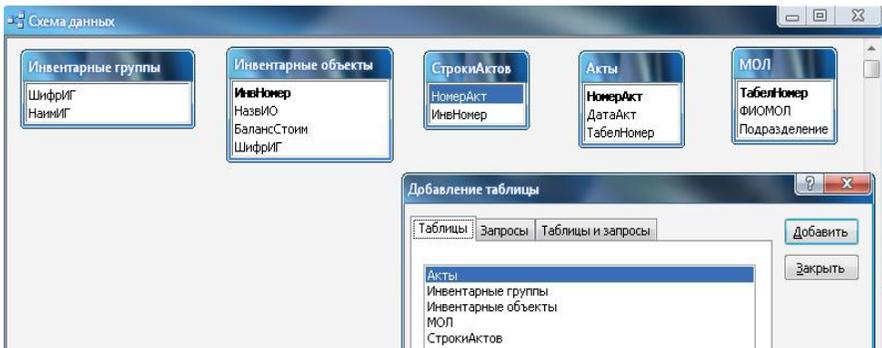


Рисунок 7 – Начальный этап установления связей между таблицами

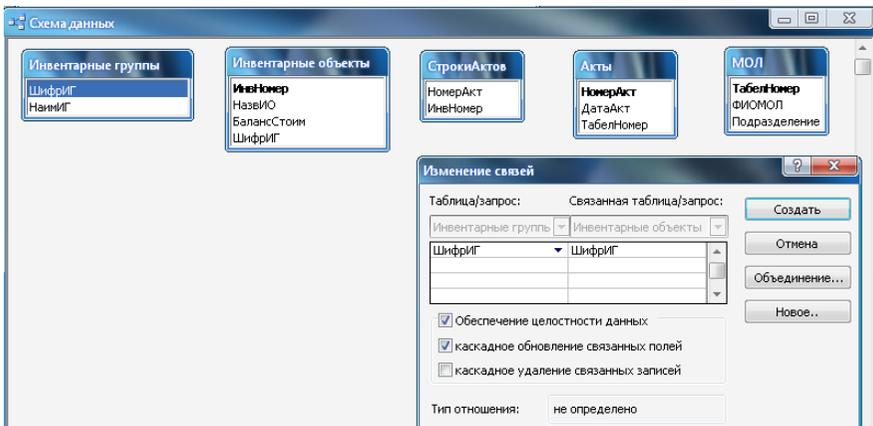


Рисунок 8 – Задание параметров связи между первичным и внешним ключами

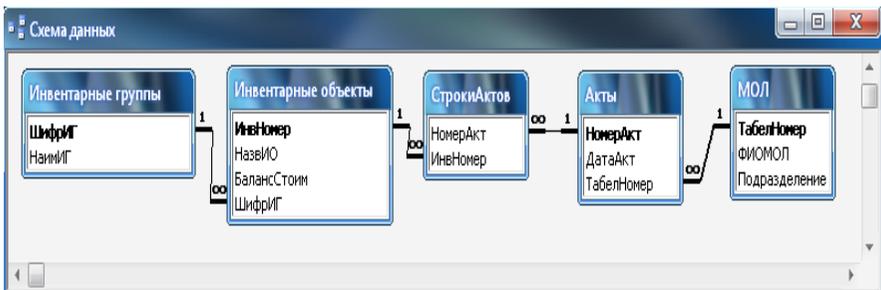


Рисунок 9 – Схема данных БД *Акты передачи*

6. Заполнение таблиц исходной информацией

Заполните таблицы информацией, содержащейся в актах, приведенных на рисунке 3. Таблицы целесообразно заполнять в следующем порядке: *Инвентарные группы*, *МОЛ*, *Инвентарные объекты*, *Акты*, *СтрокиАктов*.

На рисунке 10 приведены таблицы *Инвентарные группы* и *МОЛ*, заполненные информацией из двух актов.

Инвентарные группы : таблица		
	ШифрИГ	НаимИГ
+	101	Вычислительная техника
+	102	Мебель
+	103	Приборы

МОЛ : таблица			
	ТабелНомер	ФИОМОЛ	Подразделение
+	158	Грушин А.Б.	Плановый отдел
+	203	Демина С.Г.	Отдел кадров

Рисунок 10 – Таблицы *Инвентарные группы* и *МОЛ*

При заполнении поля, являющегося внешним ключом, надо не набирать значения на клавиатуре, а выбирать их из списка, созданного свойством *Подстановка*. Пример заполнения поля *ШифрИГ* для таблицы *Инвентарные объекты* приведен на рисунке 11.

Инвентарные объекты : таблица				
	ИнвНомер	НазвИО	БалансСтоим	ШифрИГ
▶ +	ИОК023	Телефонный аг	51 000р.	103
+	ИОК074	Стул	18 050р.	101
+	ИОК101	Светильник	8 800р.	102
+	ИПО123	Принтер LBP-8	520 000р.	103
+	ИПО348	Стул	18 050р.	
+	ИПО349	Кресло	36 800р.	102

Рисунок 11 – Пример заполнения таблицы *Инвентарные объекты*

Аналогично заполняются таблицы *Акты* и *СтрокиАктов* (рисунок 12).

Акты : таблица				
	НомерАкт	ДатаАкт	ТабелНомер	
▶ +	175	12.02.2009	158	
+	199	22.02.2009	158	Грушин А.Б.
*			203	Демина С.Г.

СтрокиАктов : таблица		
	НомерАкт	ИнвНомер
▶	175	ИПО123
	175	ИПО348
	199	ИПО349
	199	ИОКО23
	199	ИОКО74
	199	ИОК101
*		

ИПО123	
ИОКО23	Телефонный аппа
ИОКО74	Стул
ИОК101	Светильник
ИПО123	Принтер LBP-810
ИПО348	Стул
ИПО349	Кресло

Рисунок 12 – Примеры заполнения таблиц *Акты* и *СтрокиАктов*

7. Создание простого запроса на выборку

Создайте запрос, в результате выполнения которого на экран будет выведен список всех инвентарных объектов, находящихся на учете, с указанием их номера, названия, наименования соответствующей инвентарной группы и балансовой стоимости.

Для создания запроса на выборку выберите объект *Запросы* и перейдите в режим *Создание запроса в режиме конструктора* (рисунок 13).

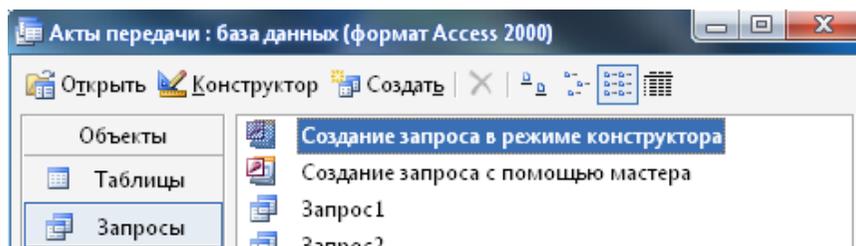


Рисунок 13 – Режим создания запроса

В диалоговом окне *Добавление таблицы* выберите таблицы БД, из которых требуется извлечь нужную информацию (рисунок 14).

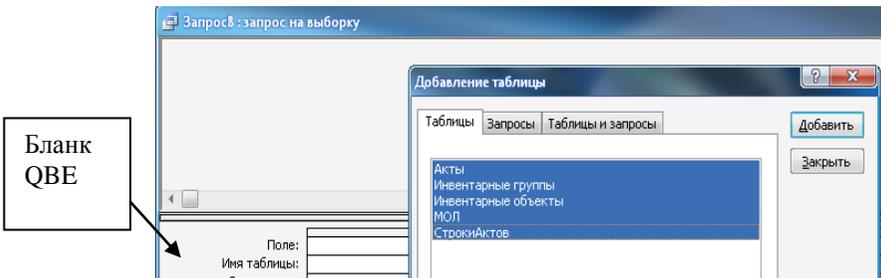


Рисунок 14 – Выбор таблиц для запроса

Для реализации данного запроса достаточно задать таблицы *Инвентарные группы* и *Инвентарные объекты*. Затем требуемые поля надо перетянуть из таблиц в строку *Поле* бланка QBE. Вид искомого запроса в режиме конструктора приведен на рисунке 15.

Запуск запроса на выполнение можно осуществить несколькими способами.

Во-первых, при открытом окне конструктора запросов можно щелкнуть по кнопке *Запуск* или раскрыть список кнопки *Вид* и выбрать команду *Режим таблицы* (рисунок 16). С помощью кнопки *Вид* можно переключаться из режима конструктора в режим таблицы и наоборот.

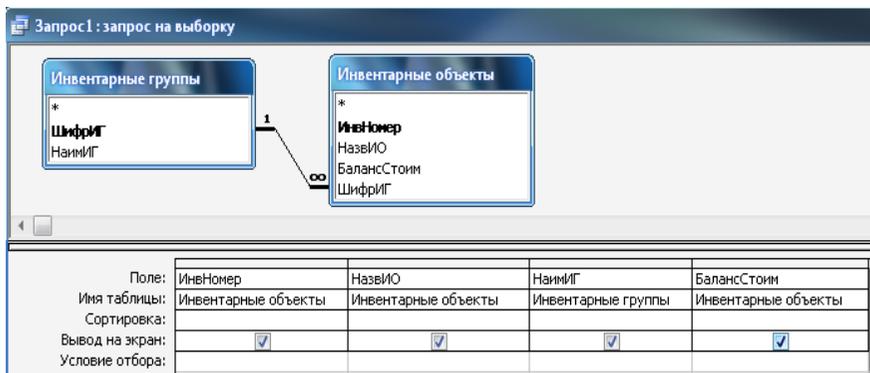


Рисунок 15 – Реализация простого запроса на выборку

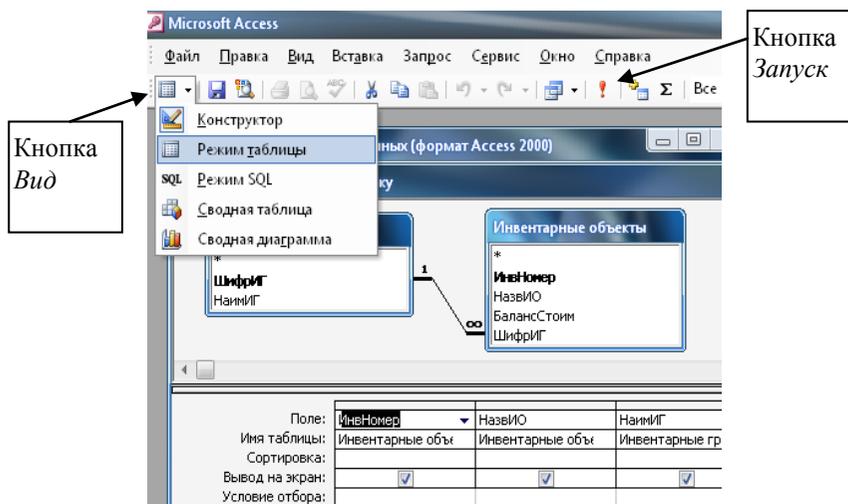


Рисунок 16 – Запуск запроса из окна конструктора

Во-вторых, после закрытия окна конструктора запросов можно дважды щелкнуть по пиктограмме запроса или выполнить команду *Открыть* (рисунок 17).

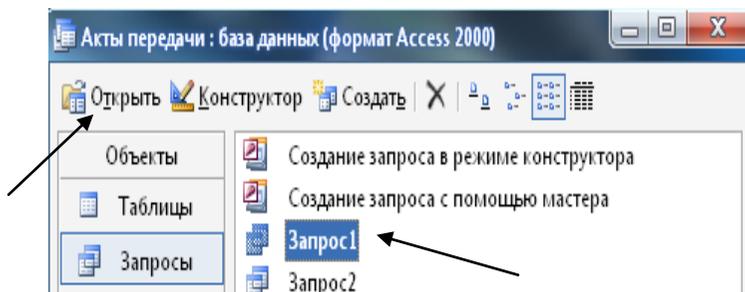


Рисунок 17 – Запуск запроса на выполнение из окна БД

8. Создание запроса на выборку на основе существующего запроса

Создайте запрос, в результате выполнения которого на экран будет выведен список инвентарных объектов, включающий три самых дорогих объекта.

В качестве исходных данных для искомого запроса целесообразно взять *Запрос1* и перетащить в бланк QBE все поля запроса. Далее надо установить в поле *БалансСтоим* и в строке *Сортировка* значение *по убыванию*. Наконец, в поле *Набор значений* панели инструментов надо набрать 3 вместо установленного по умолчанию значения *Все*.

Искомый запрос в режиме конструктора приведен на рисунке 18.

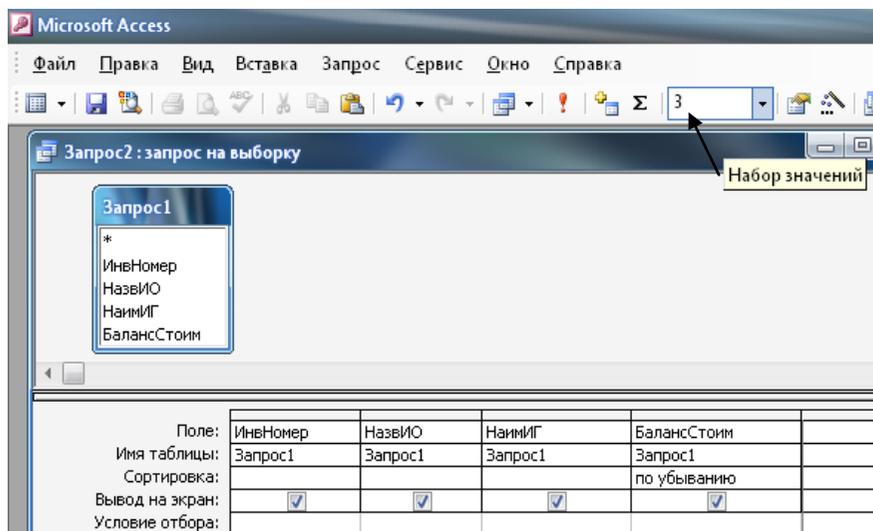


Рисунок 18 – Реализация запроса на основе ранее созданного

9. Создание запроса с параметром текстового типа

Создайте запрос, в результате выполнения которого на экран будет выведен список инвентарных объектов, принадлежащих заданной инвентарной группе.

Характерной особенностью запроса с параметром является появление при выполнении запроса диалогового окна, в котором пользователь может задать некоторые параметры.

Для данного запроса параметром является имя инвентарной группы. Параметр задается в строке *Условие отбора*. При наборе параметра удобно воспользоваться *Построителем*.

На рисунке 19 приведен вид искомого запроса в режиме конструктора.

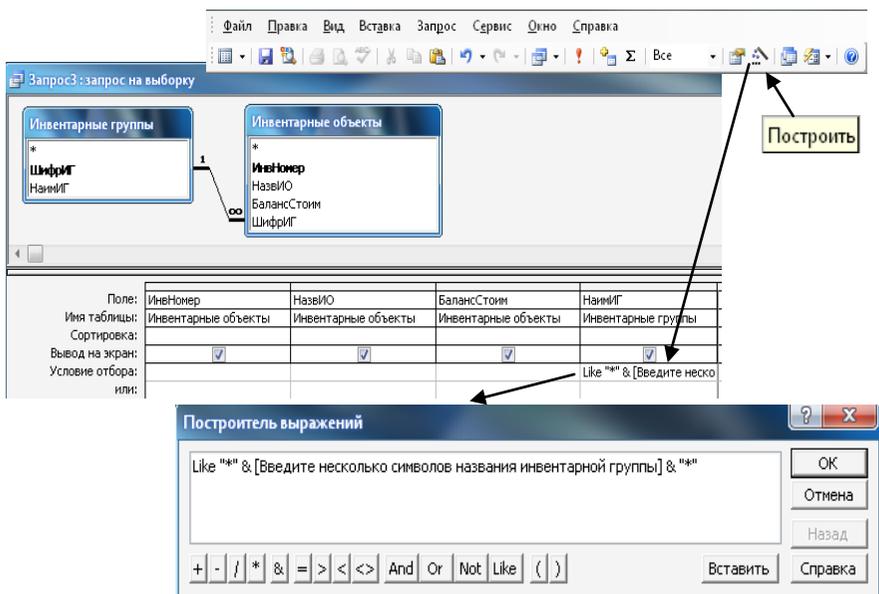


Рисунок 19 – Реализация запроса с параметром текстового типа

Построитель вызывается либо кнопкой *Построить* на панели инструментов, либо одноименной командой из контекстного меню поля *Условие отбора*.

Оператор *Like*, использованный для задания параметра, позволяет не вводить имя инвентарной группы полностью, а ограничиться вводом лишь нескольких символов наименования. Символ «*» заменяет произвольное число любых символов. Все, что написано в квадратных скобках, появляется в диалоговом окне после запуска запроса на выполнение (рисунок 20).

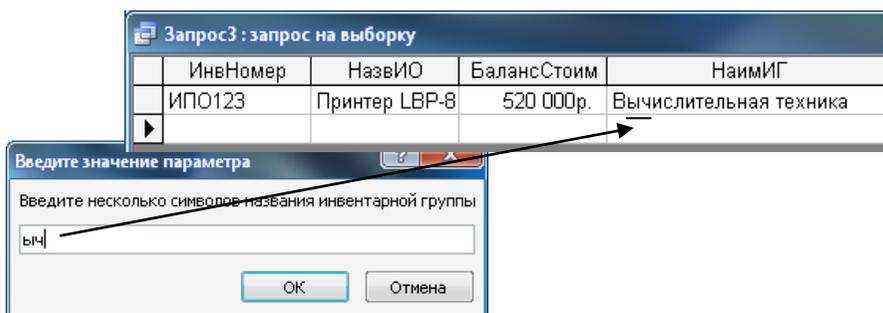


Рисунок 20 – Пример выполнения запроса с параметром

При выполнении запроса СУБД Access отыскивает среди названий инвентарных групп те, которые содержат символы, введенные в диалоговом окне (в примере на рисунке 20 введены символы «ыч»).

10. Дополнение базы данных информацией

Придумайте три акта передачи инвентарных объектов. Два акта адресуйте новым материально ответственным лицам, а третий – уже имеющемуся в таблице *МОЛ*. Общее число строк в трех актах должно быть не менее шести. Инвентарные объекты, упоминаемые в новых актах, должны относиться как к имеющимся инвентарным группам, так и к двум новым. Новые акты должны датироваться мартом.

Дополните таблицы БД информацией из новых актов. Помните, что сначала заполняются главные таблицы, а затем подчиненные.

11. Создание запроса с параметром типа Даты

Создайте запрос, в результате выполнения которого на экран будет выведен список объектов, принятых материально ответственными лицами в заданный период.

Технология создания данного запроса аналогична предыдущему. Для задания параметра в этом случае надо использовать оператор *Between*, который вызывается в окне *Построитель выражений* из раздела *Операторы* и подраздела *Сравнения*, как это показано на рисунке 21.

12. Создание запроса с обобщением (группировкой)

Создайте запрос, в результате выполнения которого для каждой инвентарной группы на экран будет выведена общая стоимость объектов, находящихся на учете.

При создании запроса данного типа надо щелкнуть по кнопке *Групповые операции* на панели инструментов. В результате в бланке QBE появляется строка *Групповые операции*, в которой в столбце *НаимИГ* надо путем выбора из списка установить значение *Группировка*, а в столбце *БалансСтоим* – значение *Sum* (суммирование).

Искомый запрос в режиме конструктора приведен на рисунке 22.

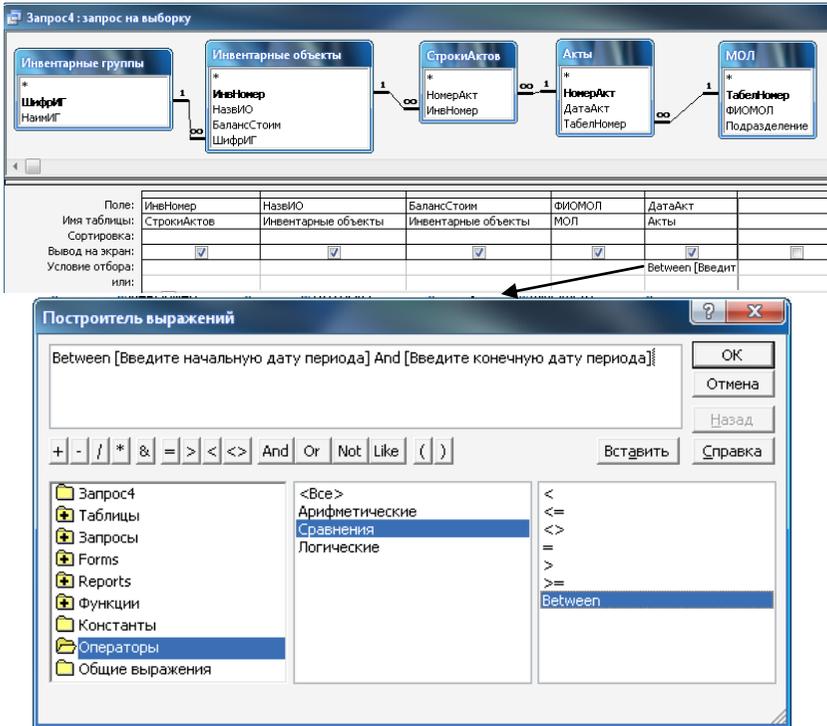


Рисунок 21 – Реализация запроса с параметром типа Даты

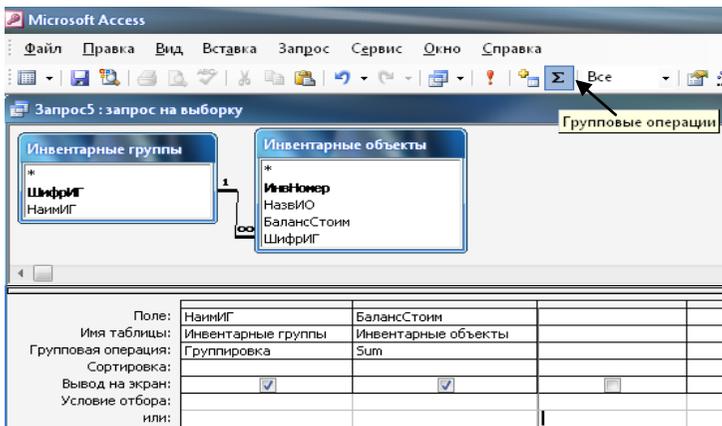


Рисунок 22 – Реализация запроса с обобщением (группировкой)

13. Создание перекрестного запроса

Создайте перекрестный запрос, в котором для каждого материально ответственного лица и для каждой инвентарной группы будет указано общее количество инвентарных объектов, находящихся на учете.

Результатом перекрестного запроса является таблица, в качестве заголовков строк и столбцов которой используются значения некоторых полей. Для создаваемого запроса заголовками строк будут фамилии материально ответственных лиц, а заголовками столбцов получаемой таблицы – наименования инвентарных групп. В ячейках этой таблицы должно быть указано количество инвентарных объектов, относящихся к соответствующей инвентарной группе и находящихся на учете соответствующего лица.

Для создания перекрестного запроса надо (после размещения в бланке запроса нужных таблиц) в списке кнопки *Тип запроса* выбрать тип *Перекрестный*. В результате в бланке QBE появятся строки *Групповая операция* и *Перекрестная таблица* (рисунок 23).

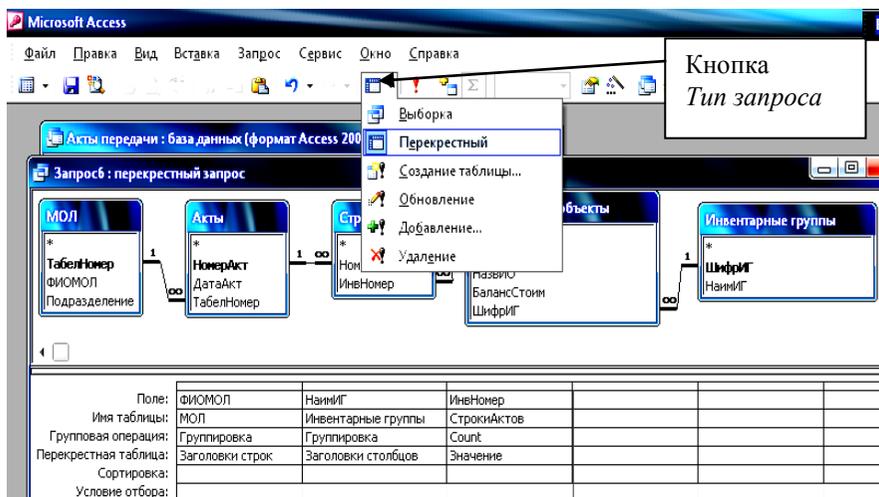


Рисунок 23 – Реализация перекрестного запроса

В строке *Перекрестная таблица* надо указать, значения какого поля будут заголовками строк, а какого – заголовками столбцов. В столбце *ИнвНомер* надо установить *Значение*.

В строке Групповая операция для столбца *ИнвНомер* надо выбрать функцию *Count* (количество). Для остальных столбцов в рассматриваемой строке надо выбрать значение *Группировка*.

14. Задания для самостоятельной работы

1. Создайте запрос, в результате выполнения которого на экран будет выведен список объектов, принятых заданным материально ответственным лицом за весь период учета.

2. Создайте запрос, в результате выполнения которого на экран будет выведен список объектов, находящихся на учете в заданном подразделении.

3. Создайте запрос, в результате выполнения которого для каждого подразделения будет указано количество и стоимость всех объектов, находящихся на учете.

4. Создайте перекрестный запрос, в котором для каждого материально ответственного лица и для каждой инвентарной группы будет указана общая стоимость инвентарных объектов, находящихся на учете.

ЛАБОРАТОРНАЯ РАБОТА 2

Цель работы: получение навыков проектирования и реализации базы данных в СУБД Access.

Задание. Создайте базу данных *Издержки обращения*, которая основана на документах «Ведомость издержек обращения в торговле», примеры которых приведен на рисунке 24.

При выполнении задания необходимо учесть следующие обстоятельства (условия применения базы данных):

- номера ведомостей не повторяются на протяжении всего периода учета;
- в одной ведомости может быть оценено несколько статей издержек;
- не все статьи из классификатора должны быть оценены в одной ведомости;
- некоторые статьи издержек могут быть не оценены ни в одной из ведомостей;
- дважды одна статья может быть оценена в одной ведомости для разных подразделений;
- в один день может быть оформлена только одна ведомость.

Ведомость издержек обращения № 24 от 14.02.2011 г.						
Статья издержек	Единица измерения	Объем (кол-во)	Тариф, р.	Сумма, р.	Подразделение	Примечание
Упаковка	м ²	650	100	65000	Торговый зал	Упаковочные материалы
Электроэнергия	кВт·ч	350	300	105000	Торговый зал	Освещение, кассовые аппараты
Транспорт	км	150	200	30000	Склад	Перевозка продуктов
Электроэнергия	кВт·ч	400	300	120000	Склад	Освещение, холодильные установки

Ведомость издержек обращения № 38 от 15.02.2011 г.						
Статья издержек	Единица измерения	Объем (кол-во)	Тариф, р.	Сумма, р.	Подразделение	Примечание
Транспорт	км	250	200	50000	Склад	Перевозка продуктов
Электроэнергия	кВт·ч	500	300	150000	Склад	Освещение, холодильные установки
Спецодежда	шт.	2	30000	60000	Склад	Спецодежда для грузчиков
Электроэнергия	кВт·ч	200	300	60000	Торговый зал	Освещение, кассовые аппараты
Транспорт	км	45	200	9000	Кабинет директора	Поездки по городу
Электроэнергия	кВт·ч	20	300	6000	Кабинет директора	Освещение

Рисунок 24 – Примеры документов «Ведомость издержек обращения в торговле»

Порядок выполнения работы

1. Определение структур таблиц

На основе вида ведомостей и условий применения БД можно выделить следующие сущности и их атрибуты:

- а) *СтатьиИздержек* (КодСтИзд, НаимСтИзд, ЕдИзм, Тариф);
- б) *Подразделения* (КодПодразд, НаимПодразд);
- в) *Ведомости* (НомерВед, ДатаВед, НаимСтИзд, ЕдИзм, Объем, Тариф, Сумма, НаимПодразд, Примечание).

Первичные ключи сущностей подчеркнуты сплошной линией.

Каждой из первых двух сущностей в проектируемой БД выделим по одной таблице:

1. *СтатьиИздержек* (КодСтИзд, НаимСтИзд, ЕдИзм, Тариф).
2. *Подразделения* (КодПодразд, НаимПодразд).

Сущность *Ведомости* является документом, поэтому для нее в БД определим две таблицы *Ведомости* и *СтрокиВед*.

В первую таблицу следует включить в качестве атрибутов реквизиты заголовочной части документа. К этим реквизитам относятся *НомерВед* и *ДатаВед*. В результате получаем следующий набор атрибутов третьей таблицы БД:

3. *Ведомости* (*НомерВед*, *ДатаВед*).

В таблице *СтрокиВед* должны быть атрибуты, соответствующие следующим реквизитам документа: *НаимСтИзд*, *ЕдИзм*, *Тариф*. Однако в таблицу достаточно включить только атрибут *КодСтИзд*, так как все три реквизита однозначно определяются по коду статьи издержек. В таблицу необходимо включить атрибут *НомерВед*, который будет ссылаться на первичный ключ таблицы *Ведомости*. Вместо реквизита *НаимПодразд* в проектируемую таблицу надо включить атрибут *КодПодразд*, являющийся первичным ключом таблицы *Подразделения*. Реквизит *Сумма* не надо отображать каким-либо атрибутом, так как он является вычисляемым. Таким образом, получим следующий набор атрибутов таблицы *СтрокиВед*:

4. *СтрокиВед* (*НомерВед*, *КодСтИзд*, *Объем*, *КодПодразд*, *Примечание*).

Первичным ключом таблицы *СтрокиВед* является совокупность трех атрибутов *НомерВед*, *КодСтИзд* и *КодПодразд*. Все эти три атрибута являются внешними ключами таблицы *СтрокиВед*.

2. Определение доменов для атрибутов БД

Будем считать, что атрибуты *КодСтИзд*, *ЕдИзм* и *КодПодразд* имеют *текстовый* тип с длиной, не превышающей 5 символов, а атрибут *НомерВед* – *текстовый* тип с длиной, не превышающей 3 символов. Правила кодирования в данном примере не рассматриваются и принимаются произвольными.

Атрибутам *НаимСтИзд*, *НаимПодразд* и *Примечание* назначим *текстовый* тип с длиной, не превышающей 50 символов.

Атрибут *Тариф* – *денежный* тип, 0 цифр после запятой (без копеек).

Атрибут *ДатаВед* – тип *Дата* и *Время*, краткий формат даты.

Атрибут *Объем* – *числовой* тип, длинное целое.

3. Определение связей между таблицами

В спроектированных таблицах БД имеется три внешних ключа. Следовательно, устанавливаются следующие связи между таблицами типа «один-ко-многим»:

1. Таблицы *СтатьиИздержек* и *СтрокиВед* связаны по атрибуту *КодСтИзд*.

2. Таблицы *Подразделения* и *СтрокиВед* связаны по атрибуту *КодПодразд*.

3. Таблицы *Ведомости* и *СтрокиВед* связаны по атрибуту *НомерВед*.

4. Создание таблиц в СУБД Access

Вызовите СУБД Access и выберите режим создания новой БД и укажите путь сохранения файла БД: *D:\STUD\шифр груп\Исдержки обращения.mdb*.

В режиме *Таблицы\Создание таблицы в режиме конструктора* задайте структуру каждой таблицы БД.

Для каждой таблицы перечислите атрибуты (поля), выберите тип поля и укажите ключевое поле. Для полей таблиц на вкладке *Общие* укажите значения размера и формата в соответствии с ранее определенными доменами.

На рисунке 25 приведены структуры таблиц базы данных.

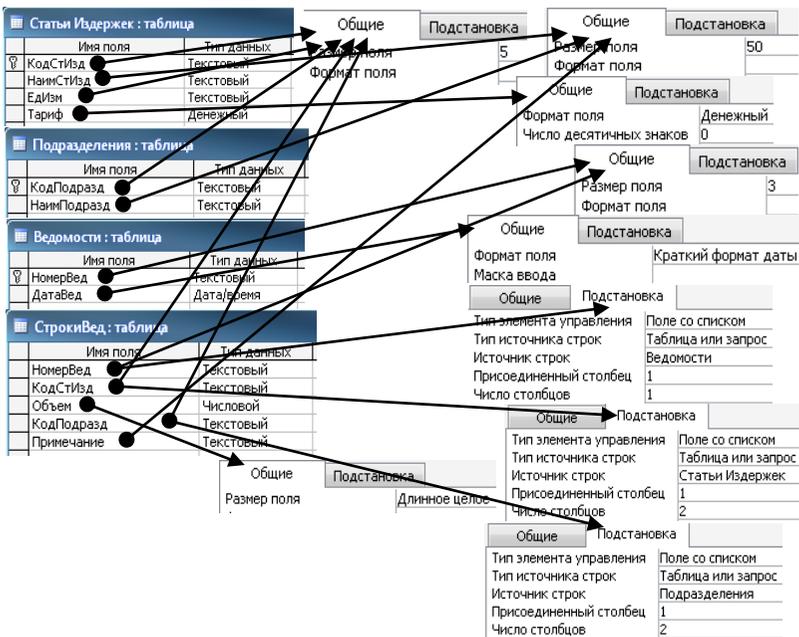


Рисунок 25 – Структура таблиц базы данных *Исдержки обращения*

Обратите внимание, что атрибуты *НомерВед*, *КодСтИзд* и *КодПодразд* таблицы *СтрокиВед* объявлены внешними ключами с помощью вкладки *Подстановка*.

5. Установление связей между таблицами в СУБД Access

С помощью команды *Сервис\Схема данных* установите связи между созданными таблицами БД. Полученная схема данных приведена на рисунке 26.



Рисунок 26 – Схема данных БД Издержки обращения

6. Заполнение таблиц исходной информацией

Заполните таблицы информацией, содержащейся в ведомостях, приведенных на рисунке 24. В первую очередь заполняются таблицы *СтатьиИздержек*, *Подразделения*, *Ведомости* (взаимный порядок заполнения этих трех таблиц произволен). Последней должна заполняться таблица *СтрокиВед*.

7. Создание запросов на выборку

Сконструируйте нижеперечисленные запросы и проверьте их выполнение. Перед выполнением четвертого запроса придумайте две ведомости, в которых упоминаются те же статьи издержек, что и в первых документах, и фигурирует, по крайней мере, одно новое подразделение. Общее число строк в новых документах должно быть не менее шести. Новые ведомости должны датироваться мартом. Дополните таблицы БД информацией из новых документов. Помните, что сначала заполняются главные таблицы, а затем подчиненные.

Запросы на выборку информации из БД:

1. Вывести список всех статей издержек с указанием их названия, единицы измерения и тарифа.

2. Вывести список ведомостей с указанием номера ведомости, даты и общей стоимости издержек по каждой ведомости, при этом список должен быть упорядочен по убыванию стоимости.

3. Вывести список ведомостей, оформленных в заданный период, с указанием номера ведомости, даты и общей стоимости издержек по каждой ведомости.

4. Вывести список подразделений, для которых оформлены ведомости в заданный период.

5. Для каждого подразделения укажите общую стоимость издержек за весь период учета.

6. Создайте перекрестный запрос, в котором для каждого подразделения и для каждой статьи издержек будет указана общая стоимость затрат за весь период учета.

7. Создайте перекрестный запрос, в котором для каждой даты и для каждой статьи издержек будет указана общая стоимость затрат по всем подразделениям.

СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ

Вейскас Дж. Эффективная работа Microsoft Office Access 2003 : [пер. с англ.] / Дж. Вейскас. – СПб. : Питер, 2005. – 1168 с.

Гарсия-Молина Г. Системы баз данных. Полный курс : [пер. с англ.] / Г. Гарсия-Молина, Ульман Дж., Уидом Дж. – М. : Вильямс, 2003. – 1088 с.

Гетц К. Access 2000. Руководство разработчика. Том 1. Настольные приложения : [пер. с англ.] / К. Гетц, П. Литвин, М. Гилберт. – Киев : ВНУ, 2000. – 1264 с.

Дейт, К. Дж. Введение в системы баз данных : [пер. с англ.] / К. Дж. Дейт. – М. : Вильямс, 2005. – 1328 с.

Джексон, Г. Проектирование реляционных баз данных для использования с микроЭВМ : [пер. с англ.] / Г. Джексон. – М. : Мир, 1991. – 252 с.

Коннолли, Т. Базы данных. Проектирование, реализация и сопровождение. Теория и практика : [пер. с англ.] / Т. Коннолли, К. Бегг. – М. : Вильямс, 2003. – 1440 с.

Левчук, Е. А. Технологии организации, хранения и обработки данных : учеб. пособие для вузов / Е. А. Левчук. – Минск : Выш. шк., 2005. – 239 с.

Литвин, П. Access 2000. Руководство разработчика. Том 2. Корпоративные приложения : [пер. с англ.] / П. Литвин, К. Гетц, М. Гилберт. – Киев : ВНУ, 2001. – 912 с.

Литвин Л. Г. Access 2002: Разработка настольных приложений : [пер. с англ.] / Л. Г. Литвин, К. Гетц, М. Гунделой. – СПб. : Питер; Киев : ВНУ, 2002. – 1008 с.

Михеева В. Д. Microsoft Access 2000. – СПб. : БХВ-Петербург, 1999. – 1088с.

Роб, П. Системы баз данных: проектирование, реализация, управление : [пер. с англ.] / П. Роб, К. Коронел. – СПб. : БХВ-Петербург, 2004. – 1040 с.

Рудикова, Л. В. Проектирование баз данных : учеб. пособие. – Минск : ИВЦ Минфина, 2009. – 352 с.

Система управления базами данных Microsoft Access. – М. : ЭКОМ, 1995. – 272 с.

Системы баз данных. Реляционные базы данных и работа с ними в среде СУБД MS Access : пособие для студентов специальности 1-26 03 01 «Управление информационными ресурсами» / С. М. Мовшович, К. Г. Сулейманов. – Гомель : Бел. торгово-экон. ун-т потребит. кооп., 2010. – 120 с.

ER-метод проектирования баз данных и его реализация в среде СУБД Access : пособие для студентов экономических специальностей. / С. М. Мовшович, К. Г. Сулейманов. – Гомель : Бел. торгово-экон. ун-т потребит. кооп., 2003. – 140 с.

СОДЕРЖАНИЕ

Пояснительная записка.....	3
1. Свойства и характеристики экономической информации	4
2. Информационное обеспечение автоматизированных информационных систем	5
3. Базы данных. Определение и общие свойства	7
4. Модели данных	8
5. Основные понятия реляционной модели данных	8
6. Этапы проектирования БД	10
7. Логическое проектирование	11
8. Пример логического проектирования.....	12
9. Система управления базами данных.....	17
10. Краткая характеристика СУБД MS Access.....	19
11. Этапы работы в СУБД Access.....	20
Лабораторная работа 1	21
Лабораторная работа 2	36
Список рекомендуемой литературы	42

Учебное издание

**КОМПЬЮТЕРНЫЕ
ИНФОРМАЦИОННЫЕ
ТЕХНОЛОГИИ**

Технологии баз данных и знаний

Пособие

**для студентов экономических специальностей
заочной формы получения высшего образования**

Автор-составитель

Мовшович Семен Михайлович

Редактор Н. В. Коптелова

Технический редактор И. А. Козлова

Компьютерная верстка Д. А. Петренко

Подписано в печать 11.04.12. Бумага типографская № 1.

Формат 60 × 84 ¹/₁₆. Гарнитура Таймс. Ризография.

Усл. печ. л. 2,56. Уч.-изд. л. 2,6. Тираж 140 экз.

Заказ № 33-04-12

Учреждение образования

«Белорусский торгово-экономический университет
потребительской кооперации».

246029, г. Гомель, просп. Октября, 50.

ЛИ № 02330/00494302 от 04.03.2009 г.

Отпечатано в учреждении образования

«Белорусский торгово-экономический университет
потребительской кооперации».

246029, г. Гомель, просп. Октября, 50.

**БЕЛКООПСОЮЗ
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
«БЕЛОРУССКИЙ ТОРГОВО-ЭКОНОМИЧЕСКИЙ
УНИВЕРСИТЕТ ПОТРЕБИТЕЛЬСКОЙ КООПЕРАЦИИ»**

Кафедра информационно-вычислительных систем

**КОМПЬЮТЕРНЫЕ
ИНФОРМАЦИОННЫЕ
ТЕХНОЛОГИИ**
Технологии баз данных и знаний

Пособие

**для студентов экономических специальностей
заочной формы получения высшего образования**

Гомель 2012