

МИНИСТЕРСТВО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ИНСТИТУТ
РАДИОТЕХНИКИ, ЭЛЕКТРОНИКИ И АВТОМАТИКИ
(ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ)

Подлежит возврату № 0000

ИНФОРМАТИКА:

РЕЛЯЦИОННЫЕ БАЗЫ ДАННЫХ

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ
ЛАБОРАТОРНЫХ РАБОТ

**1. ИНФОРМАЦИОННО-ЛОГИЧЕСКИЕ МОДЕЛИ
И МНОГОТАБЛИЧНЫЕ БАЗЫ ДАННЫХ**

**2. ОБРАБОТКА ИНФОРМАЦИИ
В СИСТЕМАХ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ**

Для студентов специальностей

072000, 190400, 190700,
200100, 200300

МОСКВА 2002

Составители: Парамонов Н.Б.,
Певцов Е.Ф.

Редактор Певцов Е.Ф.

Учебно-методические указания для выполнения лабораторных работ по предмету «Информатика», обучение которому проводится по специальностям 072000, 190400, 190700, 200100, 200300 факультета «Электроника» в I и II семестрах. Рассмотрены основные методы работы с системами управления базами данных при создании реляционных информационных систем, пользовательских форм, запросов и отчетов, а также методы обработки информации в СУБД.

Печатаются по решению редакционно-издательского совета Московского Государственного института радиотехники, электроники и автоматики (технического университета)

Рецензенты: к. т. н., доц. Т.Г. Ситникова
к. т. н., доц. А.В. Штыков

© Московский Государственный
институт радиотехники,
электроники и автоматики
(технический университет)
2002

РЕЛЯЦИОННЫЕ БАЗЫ ДАННЫХ

Введение

Наряду с текстовыми процессорами и электронными таблицами системы управления базами данных (СУБД) используются буквально во всех сферах компьютерного обеспечения информационных процессов.

Информационными системами (ИС) называют специально организованные массивы или базы данных, а также программно-аппаратные средства, предназначенные для хранения и обработки информации.

Условно различают документальные и фактографические ИС. *Документальные ИС* содержат базу из неструктурированных документов, снабженную тем или иным аппаратом поиска. Пример такой системы - справочник по Windows, вызываемый пользователем в процессе работы. *Фактографические* системы служат для выдачи однозначных ответов на ограниченный перечень возможных запросов. Для достижения этой цели данные в системе подлежат строгому структурированию и записи в определенном формате. Пример – бухгалтерские документы. Снабдив неструктурированную систему специальными описателями, мы можем привести ее к виду, позволяющему автоматически находить в базе данных нужную информацию.

Основополагающие понятия ИС – *объект, атрибут* и *связь*.

Объект (сущность) – это нечто существующее и различимое, т.е. все то, что имеет название, и для чего имеется определенный способ отличать один подобный объект от другого (факты жизни, произведения искусства, книги, математические функции, числа, философские теории и т.д.). Конкретный объект в такой группе называют *экземпляром* объекта (примеры: продукция завода, команды высшей лиги в футболе, жители определенной местности, экономические теории, алгебраические уравнения второй степени).

Атрибут (данное) – это некоторый показатель, который характеризует некий объект и принимает для конкретного экземпляра объекта некоторое значение (назначение продукции, состав команды, национальность жителя, автор теории, решения кон-

кретного уравнения). Атрибут некоторого набора объектов сам может быть набором объектов, имеющим собственные атрибуты (игроки команды, детали биографии автора, мнимые или действительные части решения уравнения).

Списки возможных значений атрибутов называют *классификаторами* (справочники, словари, тезаурус). Например, множество всех выпускаемых заводом изделий можно классифицировать по атрибутам комплекс, устройство, блок, элемент, узел, деталь. Эти атрибуты в свою очередь классифицируются по внутреннему устройству (механические, электрические ...) или назначению и т.д.

Существует строгое математическое доказательство того, что любую структуру данных можно преобразовать в простую двумерную таблицу. Базы данных, которые состоят из двумерных таблиц, называют *реляционными* (от англ. *relation* - отношение). Основная идея реляционного подхода состоит в том, чтобы представить произвольную структуру данных в виде простой двумерной таблицы (*нормализовать* структуру). Любые совокупности данных в теории и практике баз данных представляются в виде таких двумерных таблиц, связанных между собой по определенному признаку. Каждая таблица состоит из фиксированного числа столбцов и некоторого переменного количества строк. Описание столбцов составляет разработчик базы данных, его принято называть *макетом* таблицы. Каждый столбец представляет конкретное данное (например, код фирмы, код файла или папки, код продукции, десятичный номер схемы, цена). В терминологии баз данных столбцы таблицы называются *полями*.

При выборе данного необходимо приписать ему и сообщить системе точное название (*имя*), с помощью которого в дальнейшем можно будет манипулировать значениями данных. Следует различать *значение данного* и его *имя*. Для каждого поля, имеющего уникальное имя, разработчик обязательно также должен указать тип данных и их характеристики-спецификации (длина, формат и т.п.).

Отдельную строку таблицы называют *записью (record)*. Система нумерует записи по порядку, причем в отличие от количест-

ва полей в таблице, количество записей может как угодно меняться в процессе работы с базой данных. Изменение количества полей или их имен или типов представляет собой *отдельную* операцию, называемую *изменением макета таблицы* или *модификацией базы данных*.

Для удобства работы с таблицами, для некоторых полей вводят цифровые или буквенные коды. Одновременно в базу данных следует включить таблицы расшифровки кодов – словари или классификаторы. Один код должен употребляться только для обозначения определенного атрибута объекта и не может быть применен для обозначения других атрибутов.

Некоторая совокупность имен данных образует **ключ** поиска. Каждая запись в таблице должна иметь **первичный ключ**, т.е. идентификатор или адрес, значение которого однозначно определяет только эту запись. Таким образом, каждое *значение* первичного ключа *уникально*, в противном случае невозможно будет отличить одну запись от другой. Ключ может состоять из нескольких полей (т.е. задаваться как совокупность кодов). Следует всегда исключать из таблицы поля, которые не связаны непосредственно с первичным ключом таблицы.

Для проверки правильности назначения первичного ключа следует убедиться, что он обладает двумя основными признаками: 1) запись идентифицируется однозначно, т.е. значение ключа определяет только одну запись; 2) избыточность записи отсутствует, т.е. никакое поле нельзя удалить из ключа, не нарушая при этом свойства однозначной идентификации.

Среди многих программных продуктов, обеспечивающих переработку данных, весьма популярна система управления базой данных *Access*.

Объекты Access. Шесть вкладок исходного окна Access представляют шесть видов объектов, с которыми работает программа.

Таблицы - основные формы базы данных. В них собственно хранятся данные. Реляционная база данных может иметь много взаимосвязанных таблиц.

Запросы - специальные структуры, предназначенные для

обработки данных. Запросы позволяют представить данные в упорядоченном виде. С помощью запросов данные фильтруют, отбирают, изменяют, объединяют.

Формы - это объекты, с помощью которых в базу вводят новые данные или просматривают имеющиеся.

Отчеты - это обратные по отношению к формам структуры. Они позволяют выдавать на печать данные в удобном для пользователя виде.

Макросы - это *макрокоманды*. Если какие-то операции с базой данных производятся особенно часто, то имеет смысл сгруппировать последовательность этих команд в один макрос и запускать на исполнение выделенной комбинацией клавиш.

Модули - это программные процедуры, написанные на языке Visual Basic. Если стандартных средств Access недостаточно для решения специальных задач заказчика и пользователя базы данных, то программист может расширить возможности системы, написав соответствующие программные модули.

Связи. Каждое поле может входить в несколько таблиц. База данных состоит из элементов данных и связей между ними. В базе данных много различных типов элементов данных, поэтому необходима специальная схема, позволяющая изобразить связи между типами элементов данных. Такую схему называют *моделью данных*.

Модель данных представляет собой таблицу типов используемых данных содержащую имена объектов и их атрибуты, и определяющую существующие между ними связи. Связи между информационными объектами отображаются реальными отношениями.

Определены следующие *типы реальных отношений*:

1:1 (Один-к-одному) - здесь *одному* экземпляру первого информационного объекта соответствует *один* экземпляр второго информационного объекта.

Примером такого отношения может служить связь между информационными объектами

СТУДЕНТ \longleftrightarrow СЕССИЯ

1:M (Один-ко-многим) - в этом случае одному экземпляру первого объекта соответствует множество экземпляров второго объекта, а каждому экземпляру второго объекта соответствует один экземпляр первого объекта.

Примером данного отношения служит связь между информационными объектами

СТИПЕНДИЯ $\leftarrow \rightarrow$ СЕССИЯ

M:M (Многие-ко-многим) - каждому экземпляру первого объекта соответствует множество экземпляров второго объекта, и каждому экземпляру второго объекта соответствует множество экземпляров первого объекта.

Пример - связь между информационными объектами

СТУДЕНТ $\leftarrow \leftarrow \rightarrow$ ПРЕПОДАВАТЕЛЬ

При установлении связей важно уяснить, смысл понятий *обеспечение целостности данных* и *каскадное обновление данных*. Обеспечение целостности данных предполагает защиту от случаев удаления записей из одной таблицы, при котором нарушается связь с данными другой таблицы, т.е. данные другой таблицы остаются без связей. Каскадное обновление предполагает одновременное изменение данных во всех связанных таблицах.

Лабораторная работа А1 **ИНФОРМАЦИОННО-ЛОГИЧЕСКИЕ МОДЕЛИ** **И МНОГОТАБЛИЧНЫЕ БАЗЫ ДАННЫХ**

В результате выполнения данного задания студенты должны научиться:

- Создавать структуру однотабличной базы данных.
- Вводить и редактировать данные.
- Разрабатывать пользовательские формы ввода информации в однотабличную и реляционную базы данных.
- Создавать отчеты для вывода данных.
- Формировать простые запросы для поиска и отбора данных.

- Разрабатывать схему данных, создавать структуру реляционной базы данных и конструировать запросы из нескольких связанных таблиц.

Общее задание: Для иллюстрации возможностей программы Access и получения начальных навыков работы с ней составьте базу данных "**Students**", позволяющую автоматизировать обработку информации по успеваемости студентов и назначению стипендий по результатам сдачи сессии.

Выполнение лабораторной работы А1 включает несколько связанных заданий.

1. Анализ задачи.

При создании любой таблицы в базе данных необходимо строго определить состав, тип и размеры полей, составляющих таблицу. Если база данных содержит несколько таблиц, необходимо также определение ключа для каждой таблицы.

Задание 1. Проведите анализ задачи, определите необходимые таблицы БД "Students", их атрибуты, спецификации данных и разработайте макеты необходимых таблиц.

Выполнение задания 1.

Выберем сначала атрибуты студентов, как объектов базы данных. Для выполнения данного задания будем считать, что минимально необходимые сведения о каждом включают следующие данные: *фамилия, имя, отчество, номер личного дела, номер группы, дату рождения, пол*. Эти данные хранятся в личных делах студентов.

Пример данных для таблицы "Студенты" приведен на рис. 1.

Перечисленные атрибуты целесообразно одновременно назначить именами полей. В Access имя может быть любым, но должно быть уникальным, не может содержать более 64 символов и не должно включать символы ".", "!", "[", "]"

На следующем этапе следует определить *типы* полей и при необходимости - их размеры. Тип данных указывает на то, как их следет обрабатывать.

Студенты факультета "Электроника"

<i>Номер личного дела</i>	<i>Фамилия</i>	<i>Имя</i>	<i>Отчество</i>	<i>Пол</i>	<i>Дата рождения</i>	<i>Группа</i>
ЭМ-001234	Петров	Иван	Петрович	м	12.02.85	ЭМ-1-01
ЭМ-004321	Сидорова	Ольга	Борисовна	ж	25.04.85	ЭМ-2-01
ЭМ-005321	Ивашова	Екатрина	Васильевна	ж	01.01.84	ЭМ-3-01
ЭМ-001232	Кравцов	Сергей	Павлович	м	24.06.83	ЭМ-1-01
ЭМ-002234	Пугачев	Алексей	Львович	м	31.12.84	ЭМ-2-01
ЭМ-001444	Суворов	Александр	Васильевич	м	05.07.84	ЭМ-3-01
ЭМ-004512	Игнатов	Николай	Федорович	м	17.09.85	ЭМ-2-01
ЭМ-001238	Максимова	Елена	Алексеевна	ж	03.04.84	ЭМ-1-01
ЭМ-004467	Кротова	Ирина	Васильевна	ж	23.11.85	ЭМ-2-01

6

Рис.1 Данные о студентах из таблицы "Студенты" БД "Students.mdb".

Можно использовать следующие типы данных:

- *текстовый* – для ввода текстовой информации и чисел без выполнения математических расчетов (до 255 символов);
- поле *МЕМО* – для хранения произвольного текста (до 65535 символов);
- *числовой* – для выполнения над числами математических операций;
- *денежный* – для специальных операций с денежными данными (рекомендуется также для хранения чисел с одним-четырьмя знаками после запятой, т.к. при этом применяются более быстрые процедуры вычислений с фиксированной точкой и автоматическим округлением);
- *дата/время* – для хранения информации о дате и времени;
- *счетчик* – специальное числовое поле, в котором “Access 97” автоматически присваивает уникальный порядковый номер каждой записи (удобно использовать для задания первичного ключа);
- *логический* – для задания атрибута, принимающего только одно из двух возможных значений типа "Да" или "Нет";
- *поле объекта OLE* (Object Linking and Embedding – связь и внедрение объектов) – объект, созданный другим приложением, например, электронная таблица или рисунок (максимальный размер – примерно 1Гбайт).

Назначение типов данных осуществляется в специальном окне - *конструкторе таблиц*. При работе с конструктором таблиц существуют некоторые приемы, облегчающие ввод данных в таблицы. Один из них – выбирать значения из раскрывающегося поля со списком, т.е. использовать подпрограмму "*Мастер подстановки*". Другой – использование маски ввода, т.е. задание пользовательского формата для ввода данных.

Примеры масок ввода: 1) маска (000) 000-0000 – означает, что на месте нулей, заполняющих поле по умолчанию, должна обязательно введена цифра, например: (206) 555-0248; 2) маска (000) ААА-АААА – означает, что на месте нулей, заполняющих поле по умолчанию, должна обязательно введена цифра, а на месте символа "А", м.б. как цифра так и буква, например: (206) 555-

TELE. Существует еще довольно большое количество служебных символов для обозначения маски: <, !, #, L, & и т.д.

Если введенные данные не соответствуют формату, то с помощью конструктора таблиц можно задать вывод на экран определенного пользователем сообщения об ошибке.

Пример задания атрибутов (имена, типы и размеры полей) для таблицы "Студенты" приведен в табл.1.

Таблица 1. Структура таблицы "Студенты"

Поле	Тип поля	Размер поля
Номер личного дела	Текстовый	5
Фамилия	Текстовый	15
Имя	Текстовый	10
Отчество	Текстовый	15
Пол	Текстовый	1
Дата рождения	Дата	Краткий формат
Группа	Текстовый	8

Приступая к разработке макета таблицы "Студенты", для создания *новой* базы данных воспользуйтесь пунктом меню "**Файл-Создать...**", и выберите имя файла с БД: "**Students**".

Создание макетов таблиц выполните, пользуясь режимом *конструктора* таблиц (команды "**Таблица-Создать**", "**Создание таблицы**", "**Новая таблица**").

Определите поля таблицы. Для определения первого поля введите в ячейку столбца "**Поле**" имя первого поля "**Номер личного дела**", после чего в ячейке столбца "**Тип данных**" выберете из раскрывающегося списка значение "**Текстовый**". Если значение типа "**Текстовый**" не подходит, то нажмите кнопку раскрытия списка и выберите нужный тип данных. Сохраните, выбрав пункт меню "**Файл- Сохранить как...**". Закройте базу данных, выполнив команду "**Файл- Заккрыть**". В результате все готово для ввода и редактирования данных.

Для разработки макетов остальных таблиц "Сессия" и "Стипендия" базы данных *Students.mdb* выполните аналогичные действия. Данные по успеваемости в текущей сессии удобнее хра-

нить и редактировать в отдельной таблице "Сессия". Атрибутами каждого студента, идентифицируемого по атрибуту "Номер личного дела" в ней будут предметы, которые вынесены в экзаменационную сессию, а данными – результаты сдачи экзаменов в виде целых чисел. Пусть, для определенности эти предметы называются: "Мат_Анализ", "Информатика", "Лин_Алгебра" и "Физика". В этой же таблице целесообразно обобщать успешность сдачи сессии по критерию, установленному внутренними правилами данного учебного заведения (атрибут "Результат"). Для определенности будем считать, что эти данные принимают следующие значения:

- если все оценки отличные – значение "отл.";
- экзамены сданы на 5 и 4 – значение "хор.";
- все экзамены сданы на 4 и 3 – значение "удовл.";
- есть хотя бы один не сданный экзамен – значение "плохо".

Критерии назначения стипендии по результатам экзаменов целесообразно выделить в третью независимую таблицу "Стипендия" (см. табл.2), устанавливающую однозначную связь между значением объекта "Результат" и суммой стипендии.

Таблица 2. Классификатор стипендий.

Результат	Сумма
отл.	400 р.
хор.	300 р.
удовл.	200 р.
плохо	0

2. Ввод и редактирование данных и создание пользовательских форм.

Задание 2.1. Введите записи о студентах, взяв данные о для таблицы "Студенты" из примера на рис.1.

Задание 2.2. Для таблицы "Сессия" назначьте произвольно оценки за экзамены, но с условием, чтобы в ней содержались все возможные значения атрибута "Результат".

Задание 2.3. Таблица "Стипендия" идентична табл. 2, приведенной выше.

Задание 2.4. При помощи "Мастера форм" создайте простую пользовательскую форму для ввода и редактирования данных таблицы "Студенты", как это показано на рис.2. Примените созданную форму для ввода новых данных в таблицу "Студенты".

Выполнение заданий 2.1.- 2.3.

В созданную таблицу данные могут быть введены как непосредственно в табличной форме по умолчанию, так и с использованием специально разработанной пользовательской экранной формы. Редактирование записей и исправление ошибок в данных таблицы возможно также в каждом из двух этих режимов. При любом способе ввода и редактирования данных таблицы Access сохраняют введенную или исправленную запись в той папке, в которой создан макет таблицы БД. В режиме таблицы показ записей в формате строк и столбцов обеспечивает возможность одновременного просмотра нескольких записей. Допускается также добавление и изменение данных в режиме таблицы.

Вдоль верхнего края окна расположены имена полей таблицы. Каждое поле соответствует определенному столбцу в таблице. Каждая запись занимает одну строку таблицы. Ввод в определенную ячейку таблицы (выделенную курсором) осуществляется путем набора информации на клавиатуре и последующим нажатием клавиши {Enter} или {Tab}. При окончании ввода данных в последнее поле записи программа сама переходит на первое поле новой записи и ожидает ввода данных. Заполнения с клавиатуры требуют все поля, кроме тех, тип которых определен как "Счетчик".

Перемещение в таблице. Для быстрого просмотра данных, введенных в таблицу, а также необходимого позиционирования в таблице нужно обратить внимание на возможности быстрого перемещения, общие для всех программ Microsoft:

Переход к первой записи - щелчок мышью по кнопке {◀◀}, расположенной в строке состояний внизу таблицы;

- Последняя запись - {▶▶};
- Первый столбец таблицы - клавиша {Home};
- Последний столбец таблицы - клавиша {End};
- Следующий столбец справа - одна из клавиш {→}, {Enter} или {Tab};
- Следующий столбец слева — клавиша {←}, {Right}, или {Shift+Tab};
- На строку вверх – {Up};
- На строку вниз - {Down};
- Вверх на 26 строк - {PageUp};
- Вниз на 26 строк - {PageDown};
- В левый верхний угол таблицы – {Ctrl+Home};
- В правый нижний угол таблицы – {Ctrl+End}.

Редактирование данных. Редактировать данные ячейки таблицы можно как с полной, так и с частичной их заменой. Для полной замены данных необходимо подвести курсор к редактируемой ячейке так, чтобы все ее содержимое было высвечено в реверсивном виде, а затем набрать (ввести) заменяемую информацию. Частичную замену данных можно осуществить двумя способами: во-первых, щелкнуть в нужной ячейке, и она автоматически откроется для редактирования; во-вторых, используя клавиши, переместиться в нужную ячейку, а затем нажать функциональную клавишу {F2}. Удобно также пользоваться копированием в буфер обмена и вставкой из него (клавиши {Ctrl+C} и {Ctrl+V}).

Удаление записи. Для удаления записи ее необходимо выделить (щелкнуть по области маркировки записи) и либо нажать клавишу {Del}, либо выполнить команду меню "**Правка-Удалить**".

Данные в таблицу БД вводить и редактировать намного удобнее, если воспользоваться экраном в виде специального бланка (*формы*). Такой способ ввода позволяет видеть на экране все данные одной записи и вводить дополнительный текст, поясняющий значение каждого поля. Можно создать форму, напоминающую печатную страницу, расположить в ней окна списков, фотографии, графики и др.

Access располагает мастером по разработке форм пяти видов:

- *В один столбец* — поля выводятся на экран в виде последовательности строк;
- *Табличная форма* - поля выводятся в виде строк и столбцов;
- *Диаграмма* - для ее создания выбирается таблица, содержащая числовые значения, которые можно представить в графическом виде;
- *Составная форма* - объединяет в себе данные более одной таблицы БД. Позволяет просмотреть и изменить данные в нескольких таблицах одновременно;
- *Простая форма* - единственная форма, которую Access создает автоматически, включает каждое поле таблицы и использует стандартный шаблон.

Выполнение задания 2.4.

В окне **"База данных- Students"** выберите вкладку **"Форма"**. В том же окне нажмите кнопку **"Создать"**, затем в окне **"Новая форма"** в диалоге **"Создание формы"** в открывшееся окно **"Источник таблицы/запроса"** введите или выберите из списка имя таблицы **"Студенты"**. После этого создайте простую форму: выберите функцию **"Автоформа: в столбец"**. Обратите внимание, что того же результата можно добиться, воспользовавшись интерактивным режимом работы с подпрограммой **"Мастер форм"**.

Примерный вид полученной в результате проделанных операций формы представлен на рис.2. Добавьте в таблицу записи в режиме формы. Содержание добавляемых записей возьмите из примера, приведенного на рис.1. Ознакомьтесь с возможностями ввода данных в форму, сохраните созданную форму и убедитесь, что все введенные записи внесены в таблицу **"Студенты"**.

3. Вывод данных

Access выводит информацию из базы данных в виде отчета (распечатки содержимого базы данных). Для создания требуемого отчета служит программа, называемая **"Мастер отчетов"**.

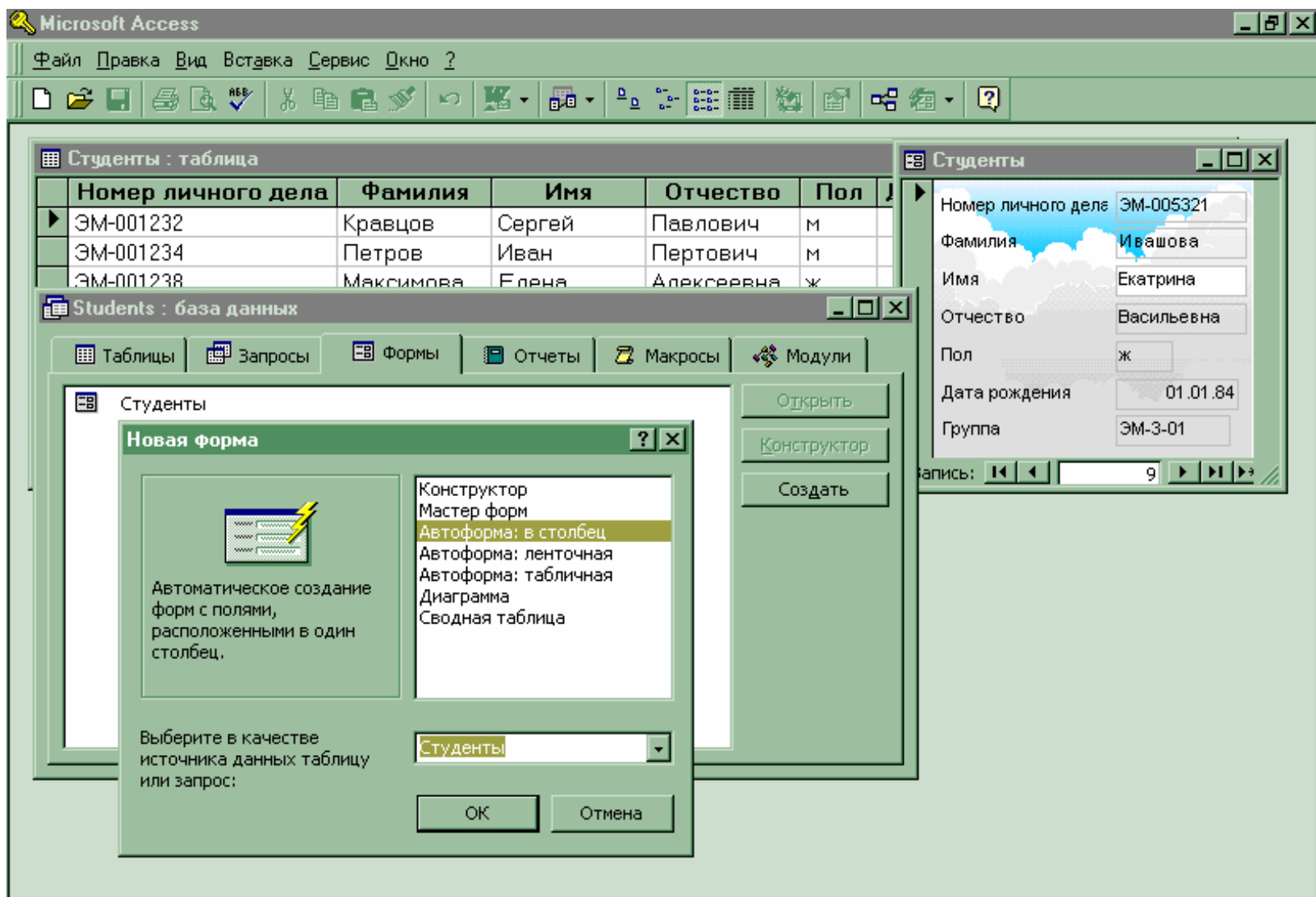


Рис. 2. Типичное окно Access с примером формы.

Варианты форм:

- *В один столбец;*
- *Группировка данных и вычисление итогов;*
- *Почтовая наклейка;*
- *Групповые вычисления;*
- *Табличный отчет;*
- *Простой отчет;*
- *Слияние с MS WORD.*

При работе с "*Мастером отчетов*" требуется определить стиль отчета (Строгий, Доклад, Табличный), его ориентацию на странице (книжная или альбомная), присвоить имя отчету и ввести заголовок отчета. В специальном режиме *конструктора* предусмотрена возможность редактирования вида созданного отчета. Пользователь может добавить свои оригинальные стили, которые "*Мастер отчетов*" будет использовать при создании следующих отчетов.

Следует помнить, что при распечатке данных с использованием формы, СУБД Access выводит на странице столько записей, сколько на ней может поместиться, поэтому возможен вариант, когда часть одной записи разместится в конце одной страницы, а другая ее часть — в конце следующей страницы.

Задание 3. Создайте детальный отчет для вывода на печать данных таблицы "Студенты".

Выполнение задания 3.

Воспользуйтесь командами и функциями: "**Создать**", "**Новый отчет**" "**Мастер отчетов**" и т.д. В соответствующих окнах, вкладках и раскрывающихся списках выберите таблицы и их поля для включения в отчет, уровни группировки не выбирайте (щелкните по кнопке {Далее}), выводимую в отчете информацию упорядочите по полям "*Группа*" и "*Фамилия*". Пример отчета, в котором назначен табличный тип представления данных, строгий стиль и книжная ориентация, но не проведена группировка данных, представлен на рис. 1.

Для более глубокого изучения СУБД рассмотрите возможности работы в режиме *конструктора отчетов* (при открытом отчете для перехода в режим конструктора воспользуйтесь функ-

цией **Вид-Конструктор...** панели инструментов **Стандартная**). Измените расположение и формат элементов отчета и проконтролируйте сделанные изменения в режиме просмотра.

4. Поиск и отбор данных.

Простейшим способом *поиска* информации в базе данных является использование директивы "**Поиск**". Этот поиск может проводиться как в одном из указанных полей, так и во всех полях таблицы БД. Возможно изменение порядка просмотра записей в таблице. Обычно поиск по этой директиве начинается с активного места таблицы (активной записи, активного поля). Для просмотра всей таблицы необходимо перейти к первой записи, а затем начать поиск.

Для того чтобы записи в таблице выстраивались при выводе в удобном для пользователя порядке, используется *сортировка*. Access может проводить сортировку по одному полю, по нескольким полям, по возрастанию или по убыванию значений ключевого признака.

Для вывода только определенных записей таблицы (отбора) используется *фильтрация*.

Задание 4.1. *Поиск данных.* Для данных, содержащихся в таблице "студент", в режиме формы осуществите *поиск* одной из записей, например, найдите запись о студенте с заданной фамилией.

Выполнение задания 4.1.

- находясь в форме "Студенты", щелкните в строке поля "Фамилия", затем щелкните на кнопке {Найти} (или выполните команду **Правка - Найти**);
- в диалоговом окне "**Поиск**" в поле "*Фамилия*" введите в строку "*Образец для поиска*" искомую фамилию;
- щелкните на кнопке {Найти}.

Задание 4.2. *Сортировка.* В режиме таблицы сгруппируйте записи по возрастанию значений одного из полей, например, расположите записи о студентах в таблице по алфавиту.

Выполнение задания 4.1.

- откройте соответствующую таблицу;
- щелкните на столбце "Фамилия";

- щелкните по кнопке пиктографического меню {По возрастанию} (или выберите пункт меню **"Записи, Быстрая сортировка, По возрастанию"**).

Задание 4.3. *Фильтрация.* Отфильтруйте данные в таблице соответствии с заданным критерием, например, используйте фильтрацию для вывода на экран только записей, относящихся к студентам, родившимся после 1985 г.

Выполнение задания 4.3.

- в окне с таблицей "Студенты" щелкните на кнопке {Расширенный фильтр} (или выберите пункт меню **"Записи, Фильтр, Расширенный фильтр, ..."**);
- в окне фильтра в строке *"Поле"* выберите поле с именем *"Дата рождения2"*;
- в строке *"Условие отбора"* наберите выражение *">31.12.85"*, щелкните на кнопке {Применить фильтр} (или выберите пункт меню **"Записи - Применить фильтр"**).
- удалите фильтр (выведите снова все записи таблицы), для чего щелкните по кнопке {Показать все записи} (или выберите пункт меню **"Записи - Удалить фильтр"**).

5. Формирование запросов.

Запросы предоставляют бóльшие возможности для поиска и отбора нужной информации, чем рассмотренные ранее средства. Запрос представляет собой вопрос о данных, хранящихся в таблицах, или инструкцию на отбор записей, подлежащих изменению. Как и формы, запросы могут применяться для закрытия доступа ко всей БД, ограничивая определенным пользователям информацию в соответствии выполняемой задачей. В результате действия запроса формируется результирующая таблица, содержащая лишь часть БД и, возможно, некоторую дополнительную информацию, полученную в результате обработки исходной. Запросы позволяют обрабатывать информацию, т.е.: упорядочивать и фильтровать данные, объединять содержимое из нескольких таблиц, разделять содержимое одной таблицы, проводить дополнительные вычисления в таблицах. При этом исходная БД не изменяется, если операция изменения БД в результате применения

запроса не обусловлена заранее (запросы на изменение).

В Access имеется возможность самостоятельно создать запрос или воспользоваться мастером по разработке запросов. Самым распространенным типом запроса является *запрос на выборку*. В результате формируются таблицы, в которых отображаются только нужные по условию задачи данные из всей базы. Вообще для формирования запросов в базах данных применяется *специальный язык запросов SQL (Structured Query Language)*. Для облегчения работы Access предоставляет пользователю и разработчику дополнительное средство - *Бланк запроса по образцу*.

Для подготовки запроса на выборку необходимо определить: а) поля, по которым будет проводиться поиск; б) искомое значение; в) поля, выводимые в результате выполнения запроса.

Задание 5. Сформируйте запрос-выборку, позволяющий получить из таблицы "Студенты" данные о студентах мужского пола, родившихся после 1984 г.

Выполнение задания 5.

Выбор базовых таблиц для формирования запроса, выполняется через функции: "**Запрос-Создать-Новый запрос**", "**Конструктор**" (т.е. ручной режим разработки запроса), "**Добавление таблиц**" из всех имеющихся в данной базе для использования в данном запросе. Выберите необходимую таблицу:

- в окне "База данных: Students" откройте вкладку {Запрос};
- нажмите кнопку {Создать};
- в диалоге "**Новый запрос**" выберите режим {Конструктор};
- на фоне появившегося окна "**Запрос1. Запрос на выборку**" в диалоге "**Добавление таблицы**" выберите таблицу "Студенты" и нажмите на кнопку {Добавить};

Рабочее окно "*Бланка запроса по образцу*" имеет две части. В верхней находится список таблиц и их поле, которые были выбраны и на которых должен основываться данный запрос. Нижняя часть представляет собой специальный элемент управления в виде бланка-таблицы и предназначается для определения структуры запроса. Перетаскиванием при помощи мыши поля из верхней части в нижнюю осуществляется автоматическое заполнение строк "*Имя таблицы*" и "*Поле*". Следующая строка бланка служит для сортировки данных по определенному полю. Вывод поля

в результирующую таблицу запроса можно отключить, если сбросить соответствующий флажок в следующей строке (например, не нужно выводить информацию о дате рождения, а сортировка или условие применены именно в этом поле). Последние две строки служат для задания условий отбора записей в запрос. Они могут быть достаточно сложными *выражениями* и задаваться различными комбинациями выражений. Условий для формирования запроса может быть несколько (по нескольким полям).

- после появления в окне **"Запрос1. Запрос на выборку"** списка полей таблицы "Студенты" в диалоговом окне "Добавление таблицы" щелкните на кнопке {Закреть};
- в первую ячейку строки *"Поле"* перетащите из списка полей таблицы "Студенты" поле *"Фамилия"*, во вторую — *"Имя"*, в третью — *"Отчество"* в четвертую — *"Дата рождения"*, в пятую — *"Пол"*;
- в пятую ячейку строки *"Условие отбора"* поместите выражение : ="м" и уберите признак вывода на экран информации из этого поля;
- в четвертую ячейку строки *"Условие отбора"* поместить выражение: >#31.12.85# и установите признак вывода на экран информации из данного поля.
- выполните запрос, для чего щелкните на кнопке пиктографического меню {!} (или выберите пункт меню **"Запрос - Запуск"**).
- сохраните запрос командой **"Файл - Сохранить запрос"**, в появившемся после этих действий окне **"Сохранение"** введите имя запроса, например, *"Пол_Возраст"*.

6. Разработка информационно-логической модели и создание многотабличной базы данных

При проектировании БД целесообразно строить информационно-логическую модель (ИЛМ) предметной области, которая определяет совокупность информационных объектов, их атрибутов и структурных связей. Информационный объект должен обладать следующими свойствами: иметь имя, уникальный идентификатор, состав атрибутов, количество экземпляров. В качестве идентификатора используется один или несколько атрибутов.

Реляционный подход к проектированию ИЛМ базируется на

понятии нормализации. В частности, спроектированные таблицы должны содержать только простые, далее неделимые данные (находятся в первой нормальной форме). Для них должно выполняться условие функционально-полной зависимости неключевых атрибутов от ключа (таблицы находятся во второй нормальной форме). Для таблиц реляционной базы также должна отсутствовать транзитивная зависимость неключевых атрибутов от ключевых или зависимости между неключевыми атрибутами (таблицы находятся в третьей нормальной форме).

СУБД Access может обрабатывать данные различных таблиц базы данных. Access обеспечивает корректную связь между таблицами базы данных за счет ключей (значений эквивалентных полей). При этом создается индекс для ключевого поля таблицы и использует его для поиска записей и объединения таблиц в запросе. Ключевое поле не может содержать пустых и повторяющихся значений. Таблицу, в которой не определен ключ, нельзя использовать при установке связей, кроме того, поиск и сортировка в такой таблице выполняются медленнее. При включении в запрос связанных таблиц базы данных в окне "**Запрос-выборка**" между ними автоматически возникает соединительная линия.

При проектировании оперативных таблиц следует соблюдать следующие правила:

- 1) Четко определить, что есть первичный ключ таблицы, т.е. убедиться, что двух записей с одинаковым значением ключа в таблице быть не может.
- 2) Если первичный ключ не просматривается, подумать, правильно ли подобран состав полей.
- 3) Если первичный ключ выбран правильно, то к нему можно подписывать любые атрибуты, зависящие *только* от ключа. Например, в словарь клиентов можно включить фамилию директора, код банка клиента, но адрес банка включать не следует - это атрибут банка, а не клиента и его место в таблице, где первичным ключом является сам банк или его код.
- 4) Если при просмотре подготовленной базы данных в паре таблиц обнаруживается одноименное поле, которое не входит в первичный ключ ни одной из таблиц, - это ошибка нормализации. Система не сможет контролировать согласованность значений таких полей.

Если между таблицами, включенными в запрос, нет связи, и она не возникает автоматически, можно соединить таблицы в окне "**Запрос-выборка**". Для этого необходимо наличие в них полей с *совпадающими*, т.е. полностью идентичными данными. Надо учесть, что такое соединение сохраняется лишь для данного запроса и при использовании этих же таблиц в новом запросе требуется соединять их заново.

Задание 6. На основании информации, содержащейся в таблицах "Студенты", "Сессия", "Стипендия" подготовьте проект приказа о назначении студентов на стипендию.

Выполнение задания 6.

В ходе решения поставленной задачи следует выделить следующие информационные объекты и их ключи (ключ — это подчеркнутый атрибут):

- Студенты (Номер, Фамилия, Имя, Отчество, Пол, Дата рождения, Группа)
- Сессия (Номер, Оценка1, Оценка2, Оценка3, Оценка4, Результат)
- Стипендия (Результат, Сумма)

Информационно-логическую модель (ИЛМ) данной задачи можно представить в графическом виде (рис.3).

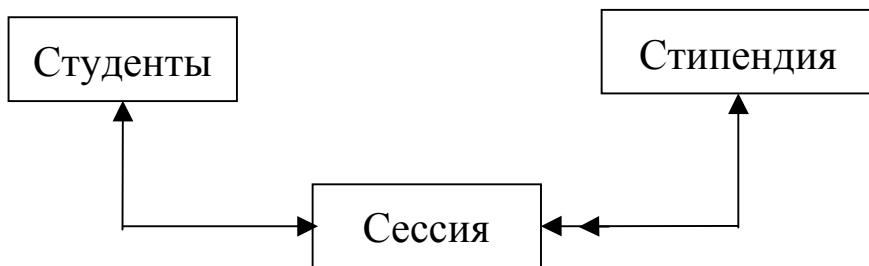



Рис.3. Пример графического представления информационно-логической модели (ИЛМ)

- Откройте ранее созданную базу данных "Students.mdb". В диалоговом окне "База данных: Students" появляется список таблиц: "Сессия", "Стипендия", "Студенты".
- Создайте новый запрос на основе связанных таблиц. Для этого: выберите вкладку "**Запрос**" и нажмите кнопку { Создать }; в диалоговом окне "**Создание запроса**" нажмите кнопку "**Конструктор**", после чего появляется окно "Запрос-

- выборка"; в окне "**Добавление таблицы**" выделите в списке "**Таблица/Запрос**" таблицу "Студенты" и щелкните на кнопке "**Добавить**", в том же списке выделите и добавьте таблицы "Сессия" и "Стипендия"; закройте диалог щелчком по кнопке {Закреть}.
- Списки полей всех выбранных таблиц появляются в верхней части окна "**Запрос-выборка**". Между этими списками автоматически возникает соединительная линия, так как между таблицами уже установлена связь. (Если связь не установлена, то установите ее, выбрав необходимые поля и отношения между ними. Для этого нужно воспользоваться пунктом меню "**Сервис-Схема данных...**". В соответствующем окне следует добавить требуемые таблицы, соединить, удерживая левую кнопку мыши, поля таблиц по требуемой связи и назначить тип связи в контекстном меню для данной связи).
 - Включите поля из трех таблиц в запрос. Из таблицы "Студенты" в бланк запроса по образцу (в строку *Поле*) перетащите следующие поля: "*Фамилия*", "*Имя*", "*Отчество*", "*Группа*". В следующее поле в запросе перетащите поле "*Сумма*" из таблицы "Стипендия".
 - Установите условие отбора. Для отбора студентов, подлежащих назначению на стипендию, необходимо в строке "*Условие отбора*" под полем "*Сумма*" ввести выражение >0 .
 - Упорядочите выводимые в запросе данные по полю "*Фамилия*" в алфавитном порядке. Щелкните ячейку в строке "**Сортировка**" под полем "*Фамилия*" и в появившемся списке выберите "**По возрастанию**". Окно сформированного запроса принимает вид, показанный на рис.4.
 - Посмотрите сформированную запросом информацию и убедитесь, что он работает правильно. Для этого щелкните по кнопке {Режим таблицы} меню **Вид** (другой вариант – кнопка с пиктограммой ).
 - Присвойте запросу имя. Для этого выберите команду меню **Файл, Сохранить...** и в окне введите имя "Проект приказа".
 - На основании запроса "Проект приказа" сформируйте соответствующий отчет.

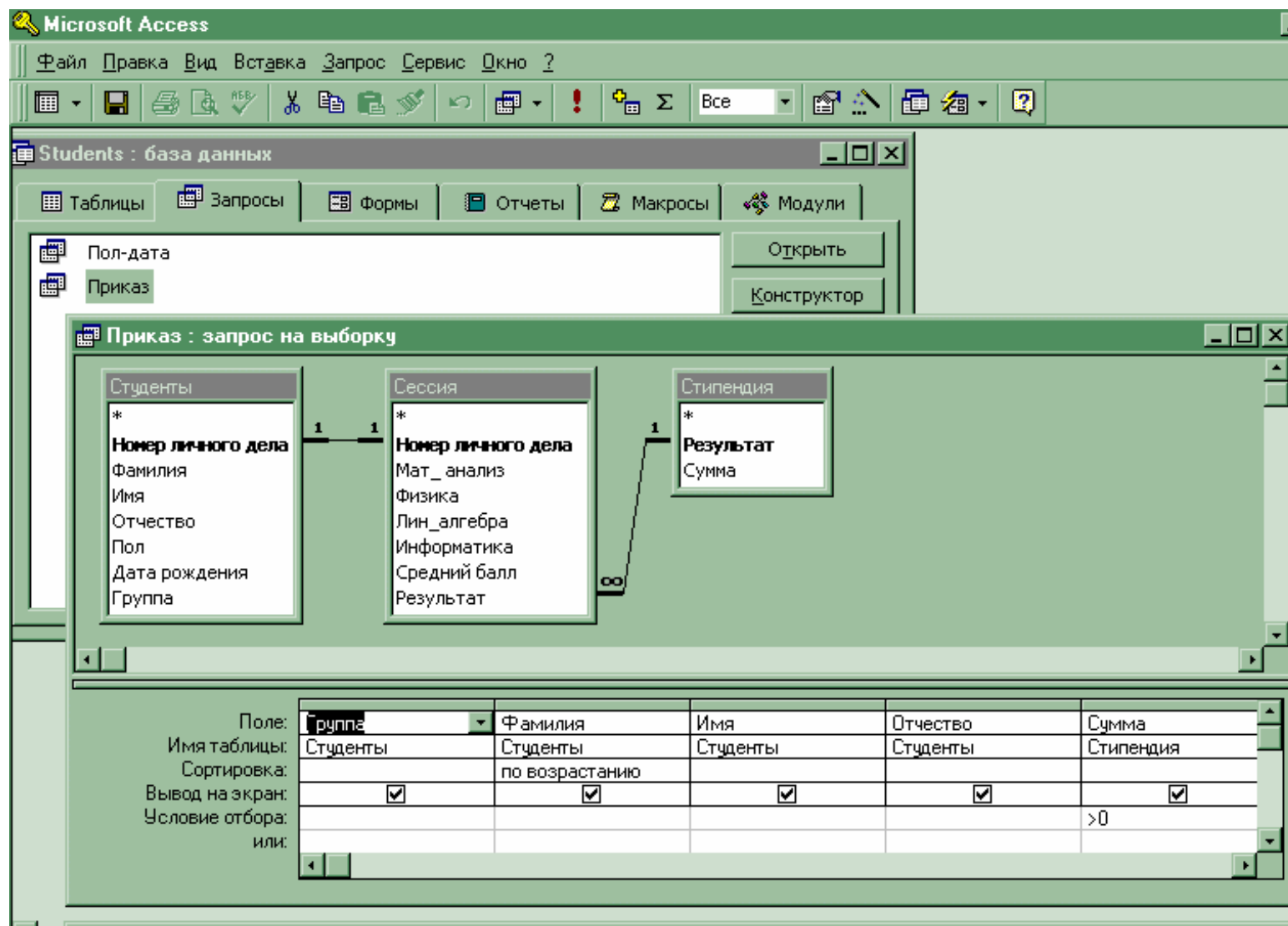


Рис.4. Окно конструктора запроса при подготовке приказа о назначении стипендии.

Лабораторная работа А2
ОБРАБОТКА ИНФОРМАЦИИ
В СИСТЕМАХ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ

Общие замечания

Результатом выполнения запроса является таблица с временным набором данных (динамический набор). Записи динамического набора могут включать поля из одной или нескольких таблиц. Запросы используются аналогично таблицам. На основе запроса можно построить отчет или форму. При обновлении данных в динамическом наборе возможно включение этих изменений в таблицы. Использование запросов позволяет осуществить различные формы доступа к одной и той же информации.

Механизм запросов позволяет не только отобразить необходимые сведения из БД, но и расширяет возможности обработки информации за счет подключения группировки и отбора записей и расчетных операций. Виды обработки данных с помощью запросов:

- включение в результирующую таблицу заданных полей;
- выбор записей по условиям отбора (И, ИЛИ и другие);
- вычисления в каждой из полученных записей;
- группировка записей с одинаковыми значениями в одном или нескольких полях для выполнения над ними групповых функций;
- обновление полей в выбранных записях;
- удаление или добавление выбранного подмножества записей из таблицы БД;
- создание новой таблицы БД на основе запроса;
- анализ информации, на основе перекрестного запроса, сформированного на двух и более полях из одной или более таблиц (создание кросс - таблицы).

Выражения в запросах. Для указания условий отбора данных и для создания вычисляемых полей в запросах используются выражения. Выражения представляют собой формулы, по которым вычисляются необходимые значения. Для построения выражений при формировании условий отбора применяются: *идентификаторы, литералы, операторы, символы-заменители,*

арифметические вычисления и функции.

Идентификатор — это имя, введенное в выражение для резервирования места под значение, которое хранится в поле или элементе управления. На основе использования идентификаторов можно создавать выражения, которые используют информацию, хранящуюся в таблицах, формах, отчетах. Для обращения к полям таблицы между идентификатором и именем поля ставится разделительный знак "!". Например, обращение к полю "*Номер личного дела*" в таблице "*Студенты*" выглядит так: [Студенты]![Номер личного дела]. Знак точки в идентификаторе служит для отделения имени объекта от его свойств.

Литерал — это точное значение, которое БД использует именно в том виде, как оно вводится. При записи литерала используются специальные символы-ограничители, которые указывают на тип данных литерала.

Если литерал — число, то он вводится без ограничителей, (например: **465.8**). Текстовый литерал должен иметь в качестве ограничителя кавычки " или '. ("**Иванов**" или '**Иванов**'). В литералах типа "Дата" используется ограничитель #. (например: **#12.11.96#**). В случае литерала типа поле или элемента управления вводятся ограничители []. (**[Фамилия]**).

Несколько специальных литералов называют *константами*. Примеры: *Да, Нет, Истина, Ложь*. Константа *Null* используется для пустых полей. Целесообразно вводить эту константу в условие отбора для поиска записей, не имеющих данных в этом поле.

Операторы - обозначают отношения между данными полей и применяются для поиска более, чем по одному значению. Оператор указывает действие, которое должно быть выполнено с элементами выражения.

Примеры операторов:

- разные варианты сравнения т.е.: =, >, <, >=, <=, <>.
- логические операторы используются для задания нескольких условий отбора:
 - **And (И)** - оба условия отбора должны принимать значение *Истина*;
 - **Or (ИЛИ)** – хотя бы одно из условий отбора должно прини-

мать значение Истина;

- **Xor** (исключающее ИЛИ)- одно из условий отбора, но не оба должно принимать значение Истина;
- **Not (НЕ)** - условие отбора не должно принимать значение Истина.

При использовании логических операторов полезно помнить, что логическое **"И"** есть операция исключения, когда добавление каждого нового условия уменьшает число записей, которые должны удовлетворять всем условиям отбора. Напротив, логическое **"ИЛИ"**, есть операция добавления или включения, поскольку увеличивает число записей, удовлетворяющих хотя бы по одному условию отбора.

- оператор для соединения частей текста **&**. Пример: **=[Фамилия] & " "&[Имя];**
- операторы языка SQL:
 - **Between...and** - для выбора значений из определенного интервала. Пример: в БД "Students" оператор **Between#01.10.85#And#31.12.85#** служит для выборки данных о студентах, родившихся в четвертом квартале 1985 г.
 - оператор **Like** служит для работы с символами-заменителями **"* "** и **"?"**. Примеры: а) запись в условии отбора **Like"A*"** (или просто **"A*"**) означает отбор всех записей, начинающихся с буквы "А"; б) запись ***Запад*** после запуска запроса выведет в данном поле все данные соответствующего столбца, в которых содержится комбинация букв **...Запад...** (**Юго-Западный, Западный ветер** и т.п.).
 - оператор **In** служит для определения, содержится ли элемент данных в списке значений.
- арифметические операторы: *****(умножение), **+**(сложение), **-** (вычитание), **/** (деление), **^**(возведение в степень).

Функция — это специальное имя, которое используется для выполнения какой-либо операции и может применяться в выражениях. В Access встроено несколько десятков функций. Аргументы функции должны заключаться в круглые скобки (). Скобки могут быть опущены только при нулевом аргументе. Примерами функций, используемых при построении выражений в за-

просах, могут служить: *Avg()* (т.е. среднее арифметическое значений), *Count()* (т.е. количество записей), *Sum()* (т.е. сумма всех записей) и т. д.

Пример применения функции *Date()*, которая возвращает системное значение даты компьютера и применяется для идентификации записей с временными ограничениями, которые уже прошли или должны вступить в силу через некоторое время: "*Between Date()And(Date()+7)*" - означает выбор всех записей с датой на предстоящую неделю.

Вычисления в запросах. В запросы можно добавлять поля, содержимое которых есть результат *вычислений*, основанных на данных из других полей (или даже таблиц). Эти поля называют *вычисляемыми полями*. В таких случаях в *Бланке запроса по образцу* на месте имени поля указывается название нового имени вычисляемого поля и через двоеточие - формула для вычислений. Имена полей с данными в формуле заключают в квадратные скобки. Как уже отмечалось, в общем случае перед именем поля следует указывать еще и *название таблицы*, содержащей это поле, затем знак "!" и *имя поля*.

Пример вычислений в запросах: вычисление среднего балла из 4-х экзаменов в таблице "Сессия":

Средний балл:([Оценка1]+[Оценка2]+[Оценка3]+[Оценка4])/4

Для работы с длинными формулами и выражениями полезно применять специальную форму Access которая вызывается на экран комбинацией клавиш {Shift+F2}. По вычисляемому полю также как и по другим полям можно проводить операции сортировки, назначать условия и проводить другие операции. ***Построитель выражений*** - специальный инструмент Access, позволяющий облегчить ввод данных для выражений. Вывести на экран его можно, вызвав в *Бланке запроса* в строке ***Условие...*** контекстное меню и затем выбрав функцию ***"Построить"*** (комбинация клавиш {Ctrl+F2}).

Частным случаем запросов на выборку являются *запросы с параметрами*.

Для формирования запроса с параметрами применяется спе-

циальная команда **Like**. В этом случае формат команды должен быть следующий:

Like [Текст для обращения к пользователю из специального окна запроса].

Эту команду следует поместить в строке "Условие отбора" в том поле, по которому нужно проводить выбор. Пример: запрос о результатах сдачи сессии по группам.

Особый вид запросов - **итоговые запросы**. Они содержат не только вычисляемые поля, но и итоговые вычисления по какой-то определенной группе записей. Для одной записи итоговые функции не имеют смысла, поэтому предварительно записи следует сгруппировать по какому-либо признаку.

Для добавления в *Бланк запроса по образцу* итоговых вычислений следует включить в него дополнительную строку "Группировка". Для этого:

– вызовите функцию меню **Запрос - Групповые операции** (кнопка со знаком суммы на соответствующей панели **Конструктора Запросов**);

– затем в добавленной строке "*Групповые операции*" в тех полях, по которым производится группировка (их может быть несколько), установите из выпадающего списка функцию "**Группировка**";

– в тех полях, где надо провести вычисление, выберите из раскрывающегося списка одну из возможных функций:

- **Группировка** - означает, что данное поле используется как основа для группировки данных.
- **Sum** - суммирование всех значений по данному полю.
- **Avg** - среднее значение для данного поля и для всех записей, отобранных запросом.
- **Min** или **Max** - определяет минимальное или максимальное значение для всех значений, отобранных запросом.
- **Count** - определяет количество записей, которые содержат некоторое значение в этом поле для всех записей, отобранных запросом.
- **Выражение** - позволяет ввести выражение, а не название поля в ячейке *Поле* (см. Построитель выражений).
- **Условие** - показывает, что поле используется для того, что-

бы в нем содержалось условие отбора, которое определяет, какие именно записи включать в данный запрос; это поле не включается в результат работы запроса.

Если по одному полю нужно провести несколько итоговых расчетов, то поле следует ввести в бланк несколько раз. Операция группировки может проводиться одновременно по нескольким полям. В этом случае производится вычисление итоговых результатов отдельно, для каждого набора записей, имеющих одинаковые значения в тех полях, по которым, осуществляется группировка.

Задание 1. Пользуясь БД "Students" организуйте вывод информации о студентках факультета, родившихся, позже 1 июля 1985 г. и раньше 31 марта 1986 г.

Указание. Это задание можно выполнить несколькими способами: а) дважды применив условия отбора *Between...And* в полях "Дата рождения" и "Пол"; б) добавив два условия по признаку $I > \dots I < \dots$ в поле "Дата рождения" (дважды выведенное) и одно - в поле "Пол" и в) применив групповые операции с группировкой в поле "Дата рождения" и "Пол".

Задание 2. Пользуясь БД "Students" организуйте вывод информации о результатах сдачи сессии по группам.

Указание: Целесообразно применение оператора *Like* в поле "Группа" (запрос с параметром).

Задание 3. Пользуясь БД "Students" организуйте вывод информации о результатах сдачи сессии по группам с выводом среднего балла каждого студента. Сгруппируйте дополнительно информацию по признаку пола. Включите результаты запроса в таблицу "Сессия".

Указание: Для ввода выражения, рассчитывающего средний балл, используйте {Shift+F2}, или {Ctrl+F2}.

Задание 4. Пользуясь БД "Students" организуйте вывод информации о среднем балле по каждому предмету.

Указание: Просто вычисление среднего дает неверный ответ (среди студентов могут оказаться имеющие 0-е оценки из-за пропуска экзамена). Возможны два пути решения: а) ввести в групповые операции дополнительное поле *Count* и условия, что все

оценки больше 0 в комбинации "Номер группы", объединенных условием OR; б) лучшее решение - создание 4-х промежуточных запросов с вычислением отдельного среднего по каждому предмету с условием не нулевой оценки и определением еще дополнительного поля Count.

Задание 5. Пользуясь БД "Students" организуйте вывод информации о результатах сдачи сессии по группам с выводом среднего балла в каждой группе.

Указание: Для выполнения этого задания следует создать промежуточные отчеты с групповыми операциями, дополнительным полем *Count* и условиями на группировку по признаку >0 и по группам (например, ="ЭМ-1-01").

Задание 6. Пользуясь БД "Students" организуйте вывод информации о результатах сдачи сессии по группам с вычислением среднего балла по группам и общего на факультете.

Указание: Для выполнения этого задания следует, создать промежуточные запросы с вычислением среднего по каждому предмету как в задании 5 и затем использовать их как основу нового запроса.

Задание 7. Создайте новые таблицы, содержащие списки групп с сортировкой по среднему баллу по итогам сессии в каждой группе.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- [1.] Симонович С., Евсеев Г., Алексеев А. Специальная информатика: учебное пособие / М.: АСТ-ПРЕСС: Информком-Пресс, 1999. – 480 с.
- [2.] Шафрин Ю. Основы компьютерной технологии (Учебное пособие для 7-11 классов по курсу «Информатика и вычислительная техника») / М.: АБФ, 1998. - 655 с.
- [3.] Сигел Ч. Изучи сам Access 97 / Перев. с англ. Ю.В. Климец. – Мн.: ООО "Попурри", 1998. – 352 с.