

Т. В. Юрченко

**ИНФОРМАЦИОННЫЕ СИСТЕМЫ В ЭКОНОМИКЕ И
УПРАВЛЕНИИ**

Учебное пособие



Нижний Новгород

2013

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Нижегородский государственный архитектурно-строительный
университет»

Кафедра прикладной информатики и статистики

Т. В. Юрченко

**ИНФОРМАЦИОННЫЕ СИСТЕМЫ В ЭКОНОМИКЕ И
УПРАВЛЕНИИ**

Учебное пособие

Нижний Новгород

ННГАСУ

2013

Юрченко, Т. В. Информационные системы в экономике и управлении: учебное пособие / Т. В. Юрченко; Нижегород. гос. архит.-строит. ун-т – Н.Новгород: ННГАСУ, 2013. – 114 с.

ISBN

В учебное пособие включен теоретический материал об одной из важнейших задач прикладной информатики по созданию и обеспечению функционирования информационных систем в экономике и управлении. Подробно рассмотрены вопросы информационного обеспечения в современной экономической ситуации. Пособие содержит практикум по созданию баз данных на основе СУБД MS Access.

Рекомендуется для студентов направлений «Менеджмент» и «Прикладная информатика».

ISBN

© Юрченко Т. В., 2013

© ННГАСУ, 2013

Содержание

| | |
|--|-----|
| Введение..... | 4 |
| 1. Теоретические аспекты создания и развития информационных систем | 5 |
| 1.1. Методология создания информационных систем..... | 5 |
| 1.2. Основные понятия теории информационных систем..... | 10 |
| 1.3. Классификация информационных систем..... | 15 |
| 1.4. Структура и виды информационных систем..... | 18 |
| 1.5. Инструментальные средства ИС управления..... | 37 |
| 2. Лабораторный практикум..... | 46 |
| 2.1. Базы данных как основная составляющая информационного обеспечения системы управления..... | 46 |
| 2.1.1. Основные понятия теории баз данных..... | 46 |
| 2.1.2. Microsoft Access как настольная СУБД реляционного типа..... | 49 |
| 2.1.3. Объекты MS Access..... | 54 |
| 2.1.4. Схема создания БД с помощью MS Access..... | 56 |
| 2.1.5. Таблицы базы данных в СУБД Access..... | 60 |
| 2.1.6. Сущность и виды запросов..... | 66 |
| 2.1.7. Параметрические запросы..... | 71 |
| 2.1.8. Создание перекрестного запроса с помощью мастера..... | 74 |
| 2.1.9. Создание перекрестного запроса с помощью конструктора..... | 75 |
| 2.1.10. Корректировка данных средствами запроса..... | 75 |
| 2.2. База данных «Учебный процесс»..... | 82 |
| 2.3. База данных «Склад»..... | 100 |
| 2.4. Темы для самостоятельных работ..... | 111 |
| Литература..... | 113 |

Введение

Современный руководитель должен быть не только грамотным с точки зрения информационных технологий и применения вычислительной техники, но и знать методические основы создания информационных систем и функционирования информационных технологий. Другими словами, он должен обладать системной информационной компетентностью.

Материал учебного пособия разбит на две главы: теоретическую и практическую. В теоретической части освещены вопросы истории и методологии информационных систем в экономике и управлении, архитектуры и структуры информационных систем. Особое внимание уделено вопросам информационного обмена и базовым информационным процессам. Отдельный раздел посвящен вопросам создания базы данных, как структурного элемента информационной системы, с помощью СУБД MS Access. Рассмотрены назначение и основные функции баз данных в информационных системах организаций.

Практическая часть представляет собой цикл лабораторных работ по созданию баз данных двух типов: «Учебный процесс» и «Склад». Каждая из них, по сути, является сквозной задачей, решение которой строится по принципу нарастания сложности, благодаря чему происходит постепенное усвоение материала и приобретение устойчивых навыков и знаний. Темы сквозных задач выбраны с тем расчетом, чтобы студенты могли тренироваться в построении информационных моделей в различных между собой, но в то же время достаточно типичных предметных областях.

В завершение практического курса приводятся контрольные вопросы и задания для курсового проектирования.

1. Теоретические аспекты создания и развития информационных систем

1.1. Методология создания информационных систем

Исторически сложилось так, что переход общества от индустриального к информационному начался в середине прошлого столетия в связи с развитием компьютерной техники и науки. При этом сам по себе научно-технический прогресс привел к усложнению технологий во всех отраслях промышленности, к появлению сложных многоэтапных производств. В этих условиях вопрос об эффективности деятельности предприятия встал особенно остро.

В то же время произошел прорыв в области теории управления экономическими объектами. Наиболее известны в мире труды Норберта Винера (США), которому принадлежит первенство в раскрытии возможностей науки кибернетики с позиций управления экономическими объектами и информационными потоками. Его труды стали фундаментом науки о функционировании информации в сложных управляющих системах.

Научные исследования в области системного применения ЭВМ для решения задач управления привели к развитию теории информации, теории кодирования, информатика сформировалась как новое научное направление. Ее теоретические исследования стали основой для методологии применения технических и программных средств для решения прикладных (практических) задач. Первоначально областью приложения служили военные и тактические задачи. Постепенно произошел переход и в область экономики.

Назначением информационных систем (ИС) и информационных технологий (ИТ) стало не только информационное сопровождение производственных и хозяйственных процессов, решение функциональных

задач управления внутри организации, но и информационное взаимодействие между различными организациями, связанными между собой в производственном, хозяйственном, информационном аспектах.

Стало очевидным, что создание информационных систем в экономике и управлении предприятиями открывает для них большие перспективы в плане повышения эффективности их деятельности. Но достижение этих целей могло стать возможным только при перестройке и координации производственных и хозяйственных процессов в совокупности с формированием методических основ создания и функционирования информационных систем. Таким образом, еще с 1970 годов началось интенсивное развитие научной мысли в направлении разработки методик создания и функционирования информационных систем и информационных технологий. Постепенно были выработаны методические подходы, положившие начало созданию единых регламентов и стандартов проектирования информационных систем в экономике, постепенному переходу от использования технических и программных средств для решения отдельных экономических задач к комплексному охвату автоматизацией функций управления по конкретным направлениям производственной и хозяйственной деятельности организаций.

Первоначально эти методические подходы были сформулированы академиком В.М.Глушковым как научно-методические положения и практические рекомендации, которые он назвал основополагающими принципами создания информационной системы: системность, развитие, информационность, совместимость, стандартизация и унификация, декомпозиция и эффективность.

Принцип системности является важнейшим при создании, функционировании и развитии информационной системы. Он позволяет подойти к исследуемому экономическому объекту (организации) как к

единому целому, выявить на этой основе многообразные типы связей между его структурными элементами, обеспечивающими целостность системы, установить направления производственно-хозяйственной деятельности организации и реализуемые ею конкретные функции.

Практическое значение применения системного принципа состоит в том, что он позволяет в доступной для анализа форме не только выявить интересующее создателей системы все существенное в деятельности организаций, но и использовать компьютерное моделирование для исследования поведения проектируемой системы в конкретных, заданных экспериментаторами условиях. Поэтому в основе создания информационных систем лежит *метод моделирования*, позволяющий находить наиболее приемлемые и обоснованные проектные решения, варианты построения системы и тем самым обеспечивать наибольшую эффективность функционирования организации.

Принцип развития заключается в том, что информационная система создается с учетом возможности постоянного пополнения и обновления функций системы и видов ее обеспечения. Предусматривается, что развиваются производственные и управленческие процессы, усложняются и перестраиваются организационные структуры экономических объектов, что вызывает необходимость масштабирования информационных систем, расширения информационного фонда, создаваемого в виде баз и хранилищ данных, баз знаний.

Информационный принцип направлен на детальное и всестороннее изучение информации и информационных процессов, сопровождающих процессы управления экономическими объектами. Имеется в виду изучение информации в семантическом (содержательном), синтаксическом (знаковом) и прагматическом (практическом) аспектах. Изучение информации необходимо для проектирования автоматизированных рабочих мест (АРМ), систем передачи, хранения, обработки данных и

защиты информации, где знания объема, содержания, полезности сведений являются основополагающими.

На информационном подходе к анализу управленческих процессов и проектированию информационных потоков в виде электронного документооборота основывается широко применяемый в настоящее время объектно-ориентированный метод моделирования информационных процессов и автоматизации проектировочных работ.

Принцип совместимости заключается в обеспечении способности взаимодействия информационных систем различных видов, назначений, уровней в процессе функционирования экономических объектов. Поэтому в процессе проектирования должно быть обеспечено системное единство методических подходов в решении проблем информационной, технической, программной совместимости вновь создаваемых и вводимых в эксплуатацию информационных систем управления. Единство методических подходов отражается в нормативно-правовых документах, регламентирующих процесс разработки, документирования, приемки и эксплуатации информационной системы. Это международные и отечественные стандарты, отраслевые и ведомственные нормативные материалы, регламенты, протоколы, стандарты организаций.

Примером комплекса государственных стандартов России может служить комплекс стандартов и руководящих документов на автоматизированные системы «Информационная технология» (ГОСТ 34.602-99), единая система программной документации (ЕСПД), представленная серией ГОСТ 19.XXX, и другие. Значительный вклад в создание единых подходов к методике проектирования и функционирования информационных систем в условиях интенсивно протекающей информатизации вносят международные организации, в частности, международная организация по стандартизации ISO (International Organization for Standardization), которая по праву считается

основателем широкого комплекса международных стандартов открытых систем OSI (Open Systems Interconnection), или открытая модель межсетевого взаимодействия. Широко используются стандарты, регламентирующие языковые средства информационной обработки, коммуникационные технологии и организацию вычислений, межобъектное взаимодействие и другие.

Принцип стандартизации и унификации заключается в необходимости применения типовых, унифицированных и стандартизированных элементов функционирования ИС. Это прежде всего относится к составляющим информационного, технического, программного и других обеспечивающих подсистем информационной технологии. Внедрение в практику создания и развития ИС этого принципа позволяет сократить временные, трудовые и стоимостные затраты на создание ИС при максимально возможном использовании накопленного опыта в формировании проектных решений и внедрении автоматизации проектировочных работ, обеспечивает многоаспектное взаимодействие ИС.

Принцип декомпозиции используется как при изучении особенностей свойств элементов и системы в целом, так и при создании ИС на новой информационно-технологической базе. Этот принцип основан на разделении системы на части и выделении отдельных комплексов работ, что создает условия для более эффективного анализа существующего состояния управленческой деятельности, изучения особенностей решения функциональных задач для последующего моделирования конкретных аспектов управленческой деятельности и перехода на автоматизацию этой деятельности.

Принцип эффективности заключается в достижении рационального соотношения между затратами на создание ИС и целевым эффектом, получаемым при ее функционировании.

Как правило, кроме основных принципов для эффективного управления выделяют также ряд частных принципов, детализирующих общие, а также организационно-технологических принципов (новых задач, автоматизации информационных потоков и документооборота, автоматизации проектирования), применение которых направлено на совершенствование методики проектирования и организации функционирования информационных технологий. Используемые при этом регламенты направлены не только на внедрение формализованных методов описания бизнес-процессов, структурирование данных, предоставление интерфейсных удобств пользователю для работы в новой информационно-технологической среде, но и на обеспечение информационно-технологического взаимодействия между различными ИС при минимизации затрат на их создание и соблюдении критерия «стоимость/качество» при функционировании информационных систем и технологий.

1.2. Основные понятия теории информационных систем

Информационная система – организационно-техническая система, которая предназначена для выполнения информационно-вычислительных работ или предоставления информационно-вычислительных услуг, удовлетворяющих потребности системы управления и ее пользователей – управленческого персонала, внешних пользователей (инвесторов, поставщиков, покупателей) путем использования и/или создания информационных продуктов.

ИС существуют в рамках систем управления и полностью подчинены целям их функционирования.

Информационно-вычислительная работа – деятельность, связанная с использованием информационных продуктов. Типичным примером

информационной работы является поддержка информационных технологий управления.

Информационно-вычислительная услуга – разовая информационно-вычислительная работа.

Информационный продукт – вещественный или нематериальный результат интеллектуального человеческого труда, обычно материализованный на каком-либо носителе, например, разнообразные программные продукты – приложения, выходная информация в виде документов управления, базы данных, базы знаний (хранилища).

Наряду с приведенным выше определением ИС, которое в данном пособии будем считать основным, существует и ряд других определений, в которых ставится акцент на другие аспекты предназначения, функционирования и структуры информационных систем. Приведем некоторые из них.

Согласно Федеральному закону Российской Федерации от 27 июля 2006 г. №149-ФЗ «Об информации, информационных технологиях и защите информации», *информационная система* – организационно упорядоченная совокупность документов (массив документов) и информационных технологий, в том числе с использованием средств вычислительной техники и связи, реализующих информационные процессы (сбор, обработка, накопление, хранение, поиск и распространение информации).

Информационная система – это совокупность внутренних и внешних потоков прямой и обратной информационной связи экономического объекта, а также методов, средств, специалистов, участвующих в процессе обработки информации и выработке управленческих решений.

Информационная система – взаимосвязанная совокупность средств, методов и персонала, используемых для хранения, обработки и выдачи информации в интересах достижения поставленной цели.

В структуре любой экономической системы можно выделить объект и субъект управления, а также информационные потоки между внешней средой, объектом и субъектом управления, которые поддерживаются ИС (рис. 1) [7].

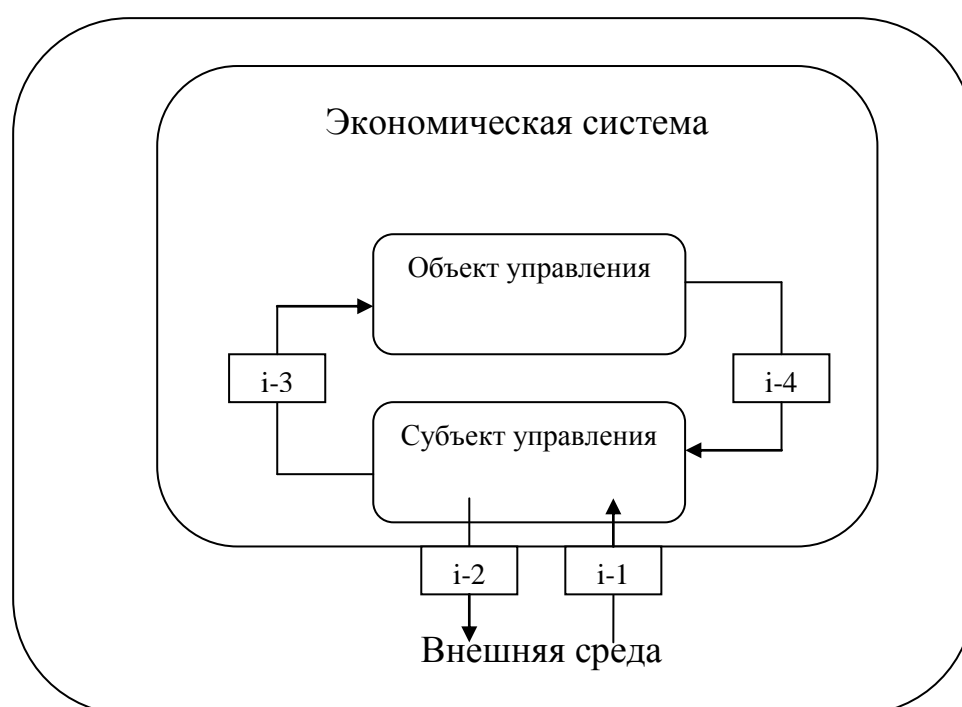


Рис. 1. Структура системы управления

Объект управления – подсистема материальных элементов экономической деятельности (сырье и материалы, оборудование, готовая продукция, работники и др.) и хозяйственных процессов (основное и вспомогательное производство, снабжение, сбыт и др.).

Субъект управления – совокупность взаимодействующих структурных подразделений экономической системы (дирекция, финансовый отдел, производственный отдел и др.), осуществляющих следующие функции управления:

– *планирование* – определяет цель функционирования экономической системы на различные периоды времени (стратегическое, тактическое и оперативное планирование);

– *учет* – отображает состояние объекта управления в результате выполнения хозяйственных процессов;

– *контроль* – фиксирует отклонение учетных данных от плановых целей и нормативов;

– *регулирование* – осуществляет оперативное управление всеми хозяйственными процессами для исключения возникающих отклонений между плановыми и учетными данными;

– *анализ* – определяет тенденции в работе экономической системы и резервы, которые учитываются при планировании на следующий временной период.

ИС представляет собой совокупность функциональной структуры, информационного, математического, технического, организационного и кадрового обеспечения, которые объединены в единую систему в целях сбора, хранения, обработки и выдачи необходимой информации для выполнения функций управления. Она обеспечивает информационную систему управления, формируя следующие информационные потоки:

– $i-1$ – информационный поток из внешней среды в систему управления, который с одной стороны представляет собой поток нормативной информации, создаваемый государственными учреждениями в части законодательства, а с другой стороны – поток информации о конъюнктуре рынка, создаваемый конкурентами, потребителями и поставщиками;

– $i-2$ – информационный поток из системы управления во внешнюю среду (отчетная информация, прежде всего финансовая, в государственные органы, инвесторам, кредиторам, потребителям, маркетинговая информация потенциальным потребителям);

– *i-3* – информационный поток из системы управления на объект представляет собой совокупность *плановой, нормативной и распорядительной* информации для осуществления хозяйственных процессов;

– *i-4* – информационный поток от объекта в систему управления, который отражает *учетную* информацию о состоянии объекта управления экономической системой (сырья, материалов, денежных, энергетических, трудовых ресурсов, готовой продукции и выполненных услугах) в результате выполнения хозяйственных процессов.

ИС накапливает и перерабатывает поступающую учетную информацию и имеющиеся нормативы и планы в аналитическую информацию, служащую основой для прогнозирования развития экономической системы, корректировки ее целей и создания планов для нового цикла ее функционирования.

К потокам информации, функционирующим в рамках ИС, предъявляются следующие требования:

– полнота и достаточность информации для реализации функций управления;

– своевременность предоставления информации;

– обеспечение необходимой степени достоверности информации в зависимости от уровня управления;

– экономичность обработки информации (затраты на обработку информации не должны превышать получаемый эффект);

– адаптивность к изменяющимся информационным требованиям пользователей.

Свойства информационных систем:

– любая информационная система может быть подвергнута анализу, построена и управляема на основе общих принципов построения систем;

- информационная система является динамичной и развивающейся;
- при построении информационной системы необходимо использовать системный подход;
- выходной продукцией информационной системы является информация, на основе которой принимаются решения;
- информационную систему следует воспринимать как человеко-компьютерную систему обработки информации.

1.3. Классификация информационных систем

Данная классификация достаточно условна, так как признаки ИС взаимосвязаны. Рассмотрим наиболее характерные из них [5].

1. Сфера деятельности объекта управления:
 - промышленное предприятие;
 - сфера обращения (торговля, банки и кредитные организации);
 - образование;
 - социальная сфера.
2. Функциональная структура ИС:
 - автоматизация технической подготовки производства;
 - маркетинг и стратегия развития предприятия;
 - технико-экономическое планирование;
 - финансы (бухучет и финансовый анализ);
 - материально-техническое обеспечение;
 - оперативно-календарное управление производством;
 - управление сбытом готовой продукции;
 - управление персоналом.

3. Организационная структура ИС:
 - автоматизированное рабочее место (АРМ) управленческого персонала;
 - комплекс взаимосвязанных АРМ.
4. Границы ИС:
 - ИС предприятия (организации);
 - ИС отрасли;
 - государственная ИС;
 - международная ИС.
5. Степень интеграции ИС:
 - локальная ИС (изолированное информационное пространство);
 - частично интегрированная ИС (общее информационное пространство);
 - полностью интегрированная корпоративная ИС.
6. Информационно-технологическая архитектура ИС:
 - ИС централизованной архитектуры построения (один центр хранения и обработки данных);
 - ИС распределенной архитектуры (компьютерные сети, наличие множества центров обработки и хранения информации).
7. Специализация ИС:
 - ИС менеджмента (или организационно-экономического управления, Information Management System – IMS);
 - информационно-поисковые системы (Information Retrieval System – IRS);
 - системы автоматизированного обучения (Education Information System – EIS) и другие.

Наибольшее распространение получили ИС менеджмента, среди которых выделяют:

- АСУП – автоматизированные системы управления ресурсами предприятий и организаций;
- АСУ ТП – автоматизированные системы управления технологическими процессами производства продукции;
- САПР – системы автоматизированного проектирования конструкций и технологий производства продукции и другие.

ИС менеджмента в качестве компонентов включает в себя другие специализированные ИС, предназначенные для следующих целей:

- автоматизация делопроизводства (Office Automation System – OAS);
- поддержка принятия решений (Design Support System – DSS);
- формирование знаний системы управления (Knowledge Base System – KBS) и другие.

Для выработки стратегии развития предприятия (перспективные направления, планирование, инвестиционное проектирование и пр.) создаются специализированные системы поддержки принятия решений, использующие методы статистического анализа и прогнозирования, моделирования данных и бизнес-процессов, имитационного моделирования, так называемые корпоративные стратегические системы (Enterprise Strategic System – ESS). В ИС поддержки принятия решений нашли применение технологии оперативного анализа и обработки данных, полученных из хранилищ данных (Data Warehouse), технологии извлечения информации из данных (Data Mining), моделирования бизнес-процессов.

В современных ИС менеджмента значительная роль отводится ИС искусственного интеллекта (Artificial Intelligence System – AIS). Эти ИС поддерживают естественно-языковой интерфейс для пользователей (специалистов по формализации знаний), предоставляют методы искусственного интеллекта для решения слабоструктурированных и плохо

формализованных задач. Ядром AIS является база знаний (Knowledge Base – KB), которая используется для формирования новой информации путем логического вывода. Наибольшее распространение среди AIS получили экспертные системы, с помощью которых на основе реальных данных выдвигается и дается оценка некоторой гипотезы.

Другие примеры AIS:

- ИС полнотекстового поиска (объединяются с реляционными СУБД и образуют новый класс постреляционных СУБД);
- нейронные сети;
- ИС аналитических вычислений на основе методов исследования операций, математического моделирования, статистического анализа и прогнозирования и другие.

1.4. Структура и виды информационных систем

Структуру ИС составляет совокупность отдельных ее частей, называемых подсистемами.

Подсистема — это часть системы, выделенная по какому-либо признаку. Общую структуру ИС можно рассматривать как совокупность подсистем независимо от сферы применения. В этом случае говорят о структурном признаке классификации, а подсистемы называют обеспечивающими. Структура любой ИС может быть представлена совокупностью обеспечивающих подсистем (рис. 2) [1].



Рис. 2. Обеспечивающие подсистемы ИС

Среди обеспечивающих подсистем обычно выделяют:

- информационное обеспечение;
- техническое обеспечение;
- математическое обеспечение;
- программное обеспечение;
- организационное обеспечение;
- правовое обеспечение.

Назначение подсистемы информационного обеспечения состоит в своевременном формировании и выдаче достоверной информации для принятия управленческих решений.

Информационное обеспечение – совокупность единой системы классификации и кодирования информации, унифицированных систем документации, схем информационных потоков, циркулирующих в организации, а также методология построения баз данных.

Рассмотрим каждую составляющую информационного обеспечения более подробно.

Названия объектов и понятий (например, названия материалов, готовой продукции, профессий и т.д.), с которыми сталкивается экономист в своей работе, складываются чаще всего произвольно. Такие названия слишком громоздки, различны по длине и форме представления. Они не обеспечивают часто полной характеристики отражаемого объекта и поэтому не могут быть использованы при передаче и обработке данных с помощью технических средств. Поэтому при проектировании ИС важное значение приобретает классификация и кодирование информации.

Классификация – распределение множества объектов на подмножества в соответствии с установленными признаками их сходства и различия.

Классификация служит основой для кодирования информации.

Кодирование – процесс присвоения условных обозначений объектам классификации и классификационным группировкам.

Условное обозначение, присвоенное конкретному объекту, называется кодовым словом или *кодом*. Код может состоять из одного знака или системы знаков, образованной по определенным правилам.

Цель кодирования – представление информации в более компактной и удобной форме при записи ее на машинный носитель; приспособление информации к обработке на компьютере и передаче по каналам связи; упрощение логической обработки информации с использованием специальных методов.

Совокупность правил, по которым строится кодовое обозначение объекта, называется *системой классификации*.

Унифицированные системы документации создаются на государственном, республиканском, отраслевом и региональном уровнях. Главная цель – это обеспечение сопоставимости показателей различных сфер общественного производства. Разработаны стандарты, где устанавливаются требования:

- к унифицированным системам документации;
- к унифицированным формам документов различных уровней управления;
- к составу и структуре реквизитов и показателей;
- к порядку внедрения, ведения и регистрации унифицированных форм документов.

Схемы информационных потоков отражают маршруты движения информации и ее объемы, места возникновения первичной информации и использования результатной информации. За счет анализа структуры подобных схем можно выработать меры по совершенствованию всей системы управления.

Построение схем информационных потоков, позволяющих выявить объемы информации и провести ее детальный анализ, обеспечивает:

- исключение дублирующей и неиспользуемой информации;
- классификацию и рациональное представление информации.

При этом подробно должны рассматриваться вопросы взаимосвязи движения информации по уровням управления. Следует выявить, какие показатели необходимы для принятия управленческих решений, а какие нет. К каждому исполнителю должна поступать только та информация, которая используется.

Методология построения баз данных базируется на теоретических основах их проектирования. Для понимания концепции методологии приведем основные ее идеи в виде двух последовательно реализуемых на практике этапов:

1-й этап – обследование всех функциональных подразделений фирмы с целью:

- понять специфику и структуру ее деятельности;
- построить схему информационных потоков;
- проанализировать существующую систему документооборота;
- определить информационные объекты и соответствующий состав реквизитов (параметров, характеристик), описывающих их свойства и назначение.

2-й этап – построение концептуальной информационно-логической модели данных для обследованной на 1-м этапе сферы деятельности. В этой модели должны быть установлены и оптимизированы все связи между объектами и их реквизитами. Информационно-логическая модель является фундаментом, на котором будет создана база данных [3].

Для создания информационного обеспечения необходимо:

- ясное понимание целей, задач, функций всей системы управления организацией;

- выявление движения информации от момента возникновения и до ее использования на различных уровнях управления, представленной для анализа в виде схем информационных потоков;
- совершенствование системы документооборота;
- наличие и использование системы классификации и кодирования;
- владение методологией создания концептуальных информационно-логических моделей, отражающих взаимосвязь информации;
- создание массивов информации на машинных носителях, что требует наличия современного технического обеспечения.

Техническое обеспечение – комплекс технических средств, предназначенных для работы ИС, а также соответствующая документация на эти средства и технологические процессы.

Комплекс технических средств составляют:

- компьютеры любых моделей;
- устройства сбора, накопления, обработки, передачи и вывода информации;
- устройства передачи данных и линий связи;
- оргтехника и устройства автоматического съема информации;
- эксплуатационные материалы.

Документацией оформляются предварительный выбор технических средств, организация их эксплуатации, технологический процесс обработки данных, технологическое оснащение. Документацию можно условно разделить на три группы:

- общесистемную, включающую государственные и отраслевые стандарты по техническому обеспечению;
- специализированную, содержащую комплекс методик по всем этапам разработки технического обеспечения;

– нормативно-справочную, используемую при выполнении расчетов по техническому обеспечению.

К настоящему времени сложились две основные формы организации технического обеспечения (формы использования технических средств): централизованная и частично или полностью децентрализованная.

Централизованное техническое обеспечение базируется на использовании в информационной системе больших ЭВМ и вычислительных центров.

Децентрализация технических средств предполагает реализацию функциональных подсистем на персональных компьютерах непосредственно на рабочих местах.

Перспективным подходом следует считать, по-видимому, частично децентрализованный подход – организацию технического обеспечения на базе распределенных сетей, состоящих из персональных компьютеров и большой ЭВМ для хранения баз данных, общих для любых функциональных подсистем.

Математическое и программное обеспечение – совокупность математических методов, моделей, алгоритмов и программ для реализации целей и задач ИС, а также нормального функционирования комплекса технических средств.

К средствам математического обеспечения относятся:

- средства моделирования процессов управления;
- типовые задачи управления;
- методы математического программирования, математической статистики, теории массового обслуживания.

В состав *программного обеспечения* входят общесистемные и специальные программные продукты, а также техническая документация.

К общесистемному программному обеспечению относятся комплексы программ, ориентированных на пользователей и предназначенных для решения типовых задач обработки информации. Они служат для расширения функциональных возможностей компьютеров, контроля и управления процессом обработки данных.

Специальное программное обеспечение представляет собой совокупность программ, разработанных при создании конкретной ИС. В его состав входят пакеты прикладных программ (ППП), реализующие разработанные модели разной степени адекватности, отражающие функционирование реального объекта.

Техническая документация на разработку программных средств должна содержать описание задач, задание на алгоритмизацию, экономико-математическую модель задачи, контрольные примеры.

Организационное обеспечение – совокупность методов и средств, регламентирующих взаимодействие работников с техническими средствами и между собой в процессе разработки и эксплуатации ИС.

Организационное обеспечение реализует следующие функции:

- анализ существующей системы управления организацией, где будет использоваться ИС, и выявление задач, подлежащих автоматизации;
- подготовку задач к решению на компьютере, включая техническое задание на проектирование ИС и технико-экономическое обоснование ее эффективности;
- разработку управленческих решений по составу и структуре организации, методологии решения задач, направленных на повышение эффективности системы управления.

Организационное обеспечение создается по результатам предпроектного обследования на 1-м этапе построения баз данных.

Правовое обеспечение – совокупность правовых норм, определяющих создание, юридический статус и функционирование

информационных систем, регламентирующих порядок получения, преобразования и использования информации.

Целью правового обеспечения является укрепление законности.

В состав правового обеспечения входят законы, указы, постановления государственных органов власти, приказы, инструкции и другие нормативные документы министерств, ведомств, организаций, местных органов власти. В правовом обеспечении можно выделить общую часть, регулирующую функционирование любой ИС, и локальную часть, регулирующую функционирование конкретной системы.

Правовое обеспечение этапов разработки ИС включает нормативные акты, связанные с договорными отношениями разработчика и заказчика и правовым регулированием отклонений от договора.

Правовое обеспечение этапов функционирования ИС включает:

- статус ИС;
- права, обязанности и ответственность персонала;
- правовые положения отдельных видов процесса управления;
- порядок создания и использования информации.

Структура ИС может быть представлена как совокупность ее функциональных подсистем, а функциональный признак может быть использован при классификации ИС. Функциональный признак определяет назначение подсистемы, а также ее основные цели, задачи и функции [6].

В хозяйственной практике производственных и коммерческих объектов типовыми видами деятельности, которые определяют функциональный признак классификации ИС, являются: производственная, маркетинговая, финансовая, кадровая.

Производственная деятельность связана с непосредственным выпуском продукции и направлена на создание и внедрение в производство научно-технических новшеств.

Маркетинговая деятельность включает в себя:

- анализ рынка производителей и потребителей выпускаемой продукции, анализ продаж;
- организацию рекламной кампании по продвижению продукции;
- рациональную организацию материально-технического снабжения.

Финансовая деятельность связана с организацией контроля и анализа финансовых ресурсов фирмы на основе бухгалтерской, статистической, оперативной информации.

Кадровая деятельность направлена на подбор и расстановку необходимых фирме специалистов, а также ведение служебной документации по различным аспектам.

Указанные направления деятельности определили типовой набор ИС:

- производственные системы;
- системы маркетинга;
- финансовые и учетные системы;
- системы кадров (человеческих ресурсов);
- прочие типы, выполняющие вспомогательные функции в зависимости от специфики деятельности фирмы.

В крупных фирмах основная ИС функционального назначения может состоять из нескольких подсистем для выполнения подфункций (табл. 1).

Тип ИС зависит от того, чьи интересы она обслуживает и на каком уровне управления.

На рис. 3 показан один из возможных вариантов классификации ИС по функциональному признаку с учетом уровней управления и уровней квалификации персонала [1].

Функции ИС

| Система маркетинга | Производственные системы | Финансовые и учетные системы | Система кадров (человеческих ресурсов) | Прочие системы, например ИС руководства |
|---|--|---|--|--|
| Исследование рынка и прогнозирование продаж Управление продажами Рекомендации по производству новой продукции Анализ и установление цены Учет заказов | Планирование объемов работ и разработка календарных планов Оперативный контроль и управление производством Анализ работы оборудования Участие в формировании заказов поставщикам Управление запасами | Управление портфелем заказов Управление кредитной политикой Разработка финансового плана Финансовый анализ и прогнозирование Контроль бюджета Бухгалтерский учет и расчет зарплаты | Анализ и прогнозирование потребности в трудовых ресурсах Ведение архивов записей о персонале Анализ и планирование подготовки кадров | Контроль за деятельностью фирмы Выявление оперативных проблем Анализ управленческих и стратегических ситуаций Обеспечение процесса выработки стратегических решений |

Из рис. 3 видно, что чем выше по значимости уровень управления, тем меньше объем работ, выполняемых специалистом и менеджером с помощью ИС.

Однако при этом возрастают сложность и интеллектуальные возможности ИС и ее роль в принятии менеджером решений. Любой уровень управления нуждается в информации из всех функциональных систем, но в разных объемах и с разной степенью обобщения.

Основание пирамиды составляют ИС, с помощью которых сотрудники-исполнители занимаются операционной обработкой данных, а менеджеры низшего звена оперативным управлением. Наверху пирамиды на уровне стратегического управления ИС изменяют свою роль и становятся стратегическими, поддерживающими деятельность менеджеров

высшего звена по принятию решений в условиях плохой структурированности поставленных задач.

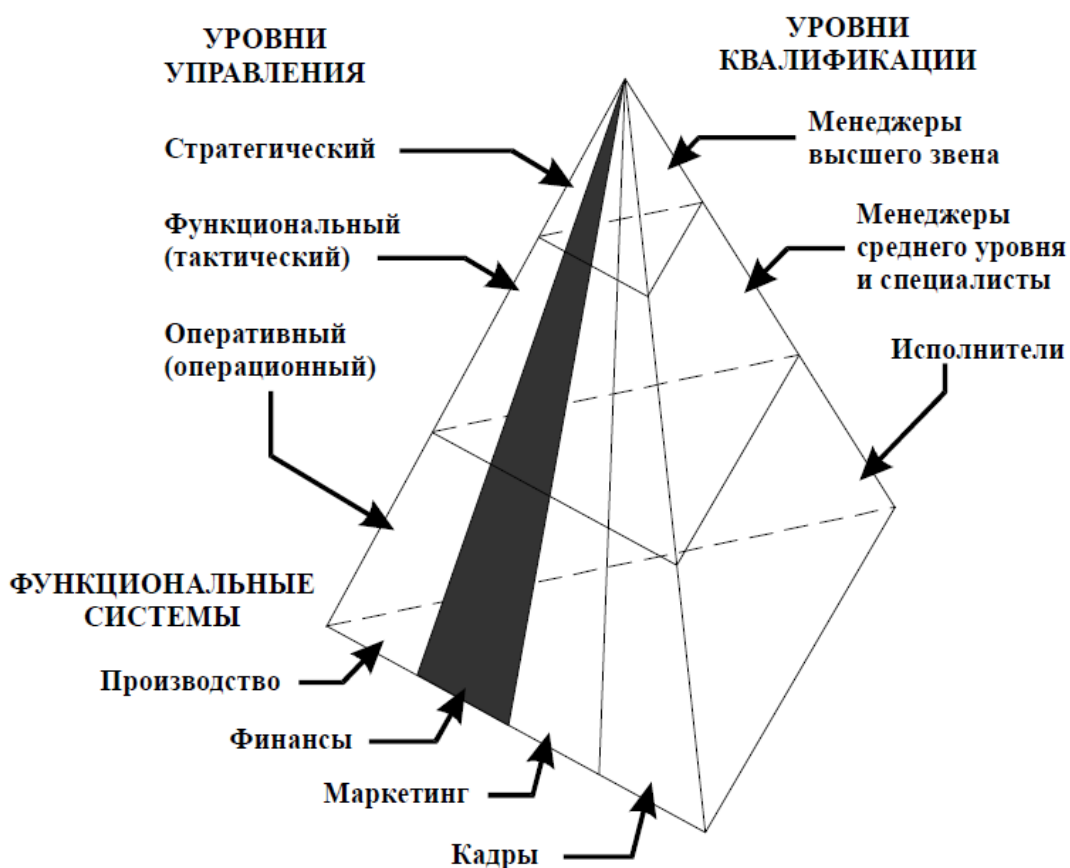


Рис. 3. Взаимосвязь функций и уровней управления ИС

В каждой организации существует система управления, имеются различные уровни управления, на которых циркулируют соответствующие информационные потоки. Для обработки информации используются различные информационные технологии, которые реализуются в ИС.

По уровням управления можно выделить:

- стратегические системы (ESS);
- аналитические системы (KWS);
- системы поддержки процесса принятия решений (DSS);
- системы оперативного управления (MIS);
- системы автоматизации делопроизводства и коммуникации (OAS);
- системы диалоговой обработки запасов (TPS).

Стратегические ИС (ESS) предназначены для оказания помощи высшему руководству компании (Top Managers) в процессе принятия стратегических решений. Эти системы учитывают долгосрочные изменения, происходящие в окружающей среде и деловом окружении предприятия, интегрируют в себе знания и данные всех ИС предприятия и строятся, как правило, на базе систем искусственного интеллекта (экспертных систем).

Для функционирования ESS необходимо:

- создание единого информационного пространства и эффективной развитой коммуникационной инфраструктуры;
- внедрение новых форм и методов управления на основе современных информационных технологий и концепции управления качеством;
- кардинальное сокращение времени, необходимого на прохождение информации, требующейся для принятия решения;
- введение единого стандарта работы с электронными документами, учитывающего существующую нормативную базу и обеспечивающего защищенность, управляемость и доступность документов
- автоматизация и повышение эффективности работы сотрудников и подразделений путем внедрения специализированных приложений и средств поддержки групповой работы
- создание инфраструктуры управления корпоративными отраслевыми знаниями.

Системы обработки данных предназначены для учета и оперативного регулирования хозяйственных операций, подготовки стандартных документов для внешней среды (счетов, накладных, платежных поручений). Выполнение действий оперативного управления производится в срок от одного до нескольких дней и реализует

регистрацию и обработку событий, например оформление и мониторинг заказов, приход и расход материальных ценностей на складе, ведение табеля учета рабочего времени и т.д. Эти задачи имеют итеративный, регулярный характер, выполняются непосредственными исполнителями хозяйственных процессов (рабочими, кладовщиками, администраторами и т.д.) и связаны с оформлением и пересылкой документов в соответствии с четко определенными алгоритмами. Результаты выполнения хозяйственных операций через экранные формы заносятся в базу данных.

ИС управления ориентированы на тактический уровень управления: среднесрочное планирование, анализ и организацию работы в течение нескольких месяцев – например, анализ и планирование поставок, составление производственных программ. Решение подобных задач предназначено для руководителей различных служб предприятия (отделов материально-технического снабжения и сбыта, цехов и т.д.). Задачи решаются на основе базы накопленных оперативных данных.

Системы поддержки принятия решений (СППР) используют в основном на верхнем уровне управления, имеющего стратегическое долгосрочное значение в течение года или нескольких лет. К таким задачам относятся формирование стратегических целей, планирование привлечения ресурсов, источников финансирования. Реже задачи СППР решаются на тактическом уровне, имеют нерегулярный характер.

ИС, которая включает в себя все три типа перечисленных ИС, называется *стратегической ИС*.

Различные уровни управления организацией представлены ИС различного типа, так как никакая единственная система не может полностью обеспечивать потребности организации во всей информации.

Рассмотрим такие основные типы ИС как:

- системы эксплуатационного уровня;
- системы уровня знания;

- системы уровня управления;
- системы стратегического уровня.

Системы эксплуатационного уровня предназначены для специалистов-исполнителей. Основной целью таких систем являются ответы на обычные вопросы и проведение потоков транзакций через организацию. Информация в такой системе должна быть легко доступна, оперативна и точна.

Системы уровня знания поддерживают работников, чья деятельность связана с созданием новой информации и знания, обработчиков данных в организации. Целью систем такого уровня является интеграция новых знаний в бизнес и помощь организации в управлении потоками документов.

Системы уровня управления предназначены для обслуживания таких функций как контроль, анализ, принятие решений и выполнение административных функций менеджеров среднего звена управления.

Системы стратегического уровня предназначены для помощи руководителям высшего уровня, которые подготавливают и принимают стратегические решения. Основное назначение таких систем – подготовка стратегических исследований и предсказание коммерческих трендов как внутри организации, так и в её деловом окружении.

Рассмотрим подробнее виды ИС, соответствующих каждому организационному уровню.

Системы диалоговой обработки запросов – TPS (Transaction Processing Systems) – основные деловые системы, которые обслуживают эксплуатационный уровень организации.

Эти системы выполняют и рассчитывают рутинные транзакции, необходимые для ведения бизнеса. Задачи, ресурсы и цели заранее определены и хорошо структурированы. Примерами такого рода систем

могут служить коммерческие расчеты продаж, системы бронирования мест в гостиницах, хранение отчетов служащих и другие.

Системы автоматизации делопроизводства – OAS (Office Automation Systems) предназначены для увеличения производительности труда обработчиков данных в офисе. Основная категория пользователей таких систем – секретари, бухгалтеры, делопроизводители, менеджеры среднего звена управления, чья работа должна главным образом использовать или распространять информацию. Системы OAS относятся к системам уровня знания.

Системы обработки знаний – KWS (Knowledge Work Systems) помогают специалистам, работа которых связана прежде всего с созданием новой информации и знания. Системы подобного класса представляют собой автоматизированные рабочие места (АРМ) специалистов и способствуют созданию новых знаний, а также гарантируют, что новые знания и технический опыт должным образом интегрируются в бизнес. Системы KWS относятся к системам уровня знания.

Управляющие ИС – MIS (Management Information Systems) обслуживают управленческий уровень организации, обеспечивая менеджерам доступ к текущей работе организации и историческим отчетам. MIS-системы обычно обслуживают менеджеров, заинтересованных в отчетах различной периодичности, и ориентируются на внутренние, не относящиеся к окружающей среде, результаты.

Эти системы имеют ограниченные аналитические возможности и используют простую установившуюся практику типа резюме и сравнения в противоположность сложным математическим моделям или статистическим методам. Можно выделить следующие особенности MIS-систем:

- поддержка структурированных и слабо структурированных решений;

- ориентация на выдачу отчетов и контроль для обеспечения текущего учета действий;
- отсутствие гибкости и ограниченные аналитические возможности.

Системы поддержки принятия решений – DSS (Decision Support Systems) могут быть определены как человеко-машинные системы, поддерживающие итеративный процесс и позволяющие менеджерам использовать данные, знания и модели для анализа и решения слабоструктурированных и неструктурированных проблем.

DSS-системы начали развиваться еще с 70-х годов и призваны помочь принятию решений управления, объединяя данные, сложные аналитические модели и удобное для пользователя программное обеспечение в единую мощную систему.

DSS-системы подсказывают или помогают выбрать руководящему персоналу обоснованное решение, приносящее успех предприятию. Они предназначены для:

- анализа данных, оценки сложившейся ситуации для выработки решения;
- выявления ограничений на принимаемое решение, противоречивых требований, формируемых внутренней и внешней средой;
- генерации списка возможных решений (альтернатив);
- оценки альтернатив с учетом ограничений и противоречивых требований для выбора решения;
- анализа последствий принимаемого решения;
- окончательного выбора решения.

Такие задачи относятся к классу слабо структурированных и неструктурированных задач, где невозможно без вмешательства человека дать четкие алгоритмы зависимостей между данными. В этих задачах количественные или качественные зависимости показателей либо

неизвестны, либо заранее не определены. В хорошо структурированных задачах можно найти алгоритм построения количественных или качественных зависимостей, что упрощает их автоматизацию.

Решение слабо структурированных задач основано на использовании экономико-математических моделей, методов экспертных оценок, многопроходного анализа данных.

Пользователями систем поддержки принятия решений являются руководители высших уровней управления предприятием и менеджеры аналитических служб.

Отличие систем поддержки принятия решений от аналитических систем заключается в следующем. Аналитические системы подготавливают аналитическую информацию. Руководитель может на ее основе принять решение. Системы поддержки принятия решений проводят дальнейший анализ аналитической информации для выработки подсказки, списка решений или единственного обоснованного решения.

Системы поддержки принятия решений помогают менеджерам решать многочисленные задачи:

- анализ примеров (case analysis) – оценка значений выходных величин для заданного набора значений входных переменных;
- параметрический («что, если?») анализ – оценка поведения выходных величин при изменении значений входных переменных;
- анализ чувствительности – исследование поведения результирующих переменных в зависимости от изменения значений одной или нескольких входных переменных;
- анализ возможностей – нахождение значений входной переменной, которые обеспечивают желаемый конечный результат;
- анализ влияния – выявление для выбранной результирующей переменной всех входных переменных, влияющих на её значение, и оценка

величины изменения результирующей переменной при заданном изменении входной переменной;

- анализ данных – прямой ввод в модель ранее имевшихся данных и манипулирование ими при прогнозировании;

- сравнение и агрегирование – сравнение результатов двух или более прогнозов, сделанных при различных входных предположениях, или сравнение предсказанных результатов с действительными, или объединение результатов, полученных при различных прогнозах или для разных моделей;

- командные последовательности (sequences) – возможность записывать, исполнять, сохранять для последующего использования регулярно выполняемые серии команд и сообщений;

- анализ риска – оценка изменения выходных переменных при случайных изменениях входных величин;

- оптимизация – поиск значений управляемых входных переменных, обеспечивающих наилучшее значение одной или нескольких результирующих переменных.

Исполнительные информационные системы – EIS (Executive Information Systems) обслуживают стратегический уровень организации и ориентированы на решение неструктурированных задач. Данные системы разрабатываются для фильтрации, сжатия и выявления критических данных, сокращая время и усилия, требуемые для получения полезной для высших руководителей информации.

EIS-системы выбирают агрегированные данные из внутренних MIS- и DSS-систем и проводят системный анализ окружающей среды лучше, чем любые прикладные и специфические системы. Данные системы используют наиболее продвинутое графическое программное обеспечение и могут поставлять графики и данные из многих источников немедленно в офис старшего менеджера или в зал заседаний.

Так как EIS разработаны, чтобы использоваться старшими менеджерами, которые часто имеют немного прямых контактов с компьютерными ИС, такие системы имеют лёгкий в использовании интерфейс.

В отличие от других типов ИС EIS не предназначены для решения определенных проблем. Данные системы обеспечивают лишь обобщенные вычисления и передачу данных, которые могут применяться к изменяющемуся набору проблем. EIS-системы используют меньшее количество аналитических моделей, чем DSS-системы.

EIS помогают найти ответы на следующие вопросы:

1. В каком бизнесе мы должны быть?
2. Что делают конкуренты?
3. Какие новые приобретения защитили бы нас от циклических деловых колебаний?
4. Какие подразделения мы должны продать, чтобы увеличить наличность?

В настоящее время наблюдаются следующие тенденции:

- ощутима бизнес-потребность в доступе к неструктурируемой информации: текстам, графической, аудио-, видеоинформации. Ее интеграция со структурируемыми данными приведет к появлению нового класса инструментов;
- наблюдается тенденция слияния OLAP-систем с инструментами интеллектуального выбора данных;
- большинство информационных хранилищ обеспечиваются средствами получения аналитических данных;
- системы поддержки принятия решений проникают во все сферы экономической и финансовой деятельности: банковские, маркетинговые, финансовые системы, электронный бизнес, торговлю, корпоративные ИС.

1.5. Инструментальные средства ИС управления

Сегодня для решения задач управления применяются следующие инструментальные средства – информационные технологии:

- технологии аналитической обработки данных;
- технологии «добычи данных»;
- технологии искусственного интеллекта;
- геоинформационные технологии;
- технологии виртуальной реальности.

Технологии аналитической обработки данных (OLAP-технологии, от англ. On-line Analytical Processing) представляют собой инструмент для динамического многомерного анализа текущих и исторических данных в интерактивном режиме.

Они обеспечивают:

- агрегирование и детализацию данных по запросу;
- выдачу данных в терминах предметной области;
- анализ деловой информации по множеству параметров (например, поставщик, его местоположение, поставляемый товар, цены, сроки поставки);
- многопроходный анализ информации, который позволяет выявить не всегда очевидные тенденции в исследуемой предметной области;
- произвольные срезы данных по наименованию, выбираемых из разных внутренних и внешних источников (например, по наименованию товара);
- выполнение аналитических операций с использованием статистических и других методов;
- согласование данных во времени для использования в прогнозах, трендах, сравнениях (например, согласование курса рубля).

Концепция технологии OLAP была сформулирована Эдгаром Коддом в 1993 году. Она стала ключевым компонентом организации данных в информационных хранилищах и их применении. Эта технология основана на построении многомерных наборов данных – OLAP-кубов. Целью использования технологий OLAP является анализ данных и представление этого анализа в виде, удобном для восприятия и принятия решений.

Основные требования, предъявляемые к приложениям для многомерного анализа:

- предоставление пользователю результатов анализа за приемлемое время (не более 5 сек.);
- осуществление логического и статистического анализа, его сохранение и отображение в доступном для пользователя виде;
- многопользовательский доступ к данным;
- многомерное представление данных;
- возможность обращаться к любой информации независимо от места ее хранения и объема.

Многомерный анализ данных может быть произведен посредством клиентских приложений и серверных OLAP-систем.

Клиентские приложения, содержащие OLAP-средства, выполняют вычисление агрегатных данных. Агрегатные данные размещаются в кэш внутри адресного пространства такого OLAP-средства. Кэш – быстродействующий буфер большой емкости, работающий по специальному алгоритму. При этом, если исходные данные находятся в реляционной базе, вычисления производятся OLAP-средствами клиентского приложения. Если исходные данные размещаются на сервере баз данных, то OLAP-средства приложений посылают SQL-запросы на сервер баз данных и получают агрегатные данные, вычисленные сервером.

Примерами клиентских приложений, содержащими OLAP-средства, являются приложения статистической обработки данных SEWSS (Statistic Enterprise – Wide SPS System) фирмы StatSoft и MS Excel. Excel позволяет создать и сохранить небольшой локальный многомерный OLAP-куб и отобразить его двух или трехмерные сечения.

Многие средства проектирования позволяют создавать простейшие OLAP-средства. Например, Borland Delphi и Borland C++ Builder.

Отметим, что клиентские приложения применяются при малом числе измерений (не более шести) и небольшом разнообразии значений этих измерений.

Серверные OLAP-системы развили идею сохранения кэш с агрегатными данными.

В них сохранение и изменение агрегатных данных, поддержка содержащего их хранилища осуществляется отдельным приложением (процессом), называемым OLAP-сервером.

Клиентские приложения делают запросы к OLAP-серверу и получают требуемые агрегатные данные. Применение OLAP-серверов сокращает трафик сети, время выполнения запросов, сокращает требования к ресурсам клиентских приложений.

В масштабе предприятия обычно используются OLAP-серверы типа Oracle Express Server, MS SQL Server 2000 Analysis Services и др.

Заметим, что MS Excel позволяет делать запросы к OLAP-серверам.

Самые современные аналитические системы основываются на информационных хранилищах и обеспечивают весь спектр аналитической обработки.

Под хранилищем данных (Data Warehouse) понимают автоматизированную информационно-технологическую систему, которая

собирает данные из существующих баз и внешних источников, а также формирует, хранит и эксплуатирует информацию как единую.

Хранилище данных предоставляет единую среду хранения корпоративных данных, организованных в структурах, оптимизированных для выполнения аналитических операций.

Доступ к информационным хранилищам реализован посредством транзакций. В отличие от OLAP систем все остальные системы, работающие с транзакциями, называют OLTP системами (On-Line Transaction Processing). OLTP системы обеспечивают обработку запросов, выдачу отчетов, но не обеспечивают функций анализа данных.

Технологии «добычи данных» (Data Mining) разработаны для поиска и выявления в данных скрытых связей и взаимозависимостей с целью предоставления их руководителю в процессе принятия решений. Для этого используются статистические методы корреляции, оптимизации и другие, позволяющие находить эти зависимости и синтезировать дедуктивную (обобщающую) информацию.

Инструменты Data Mining обеспечивают:

- поиск зависимых данных (реализацию интеллектуальных запросов);
- выявление устойчивых бизнес-групп (выявление групп объектов, близких по заданным критериям);
- ранжирование важности признаков при классификации объектов для проведения анализа;
- прогнозирование бизнес-показателей (например, ожидаемые продажи, спрос);
- оценку влияния принимаемых решений на достижение успеха предприятия;
- поиск аномалий.

Где могут быть использованы результаты применения технологий Data Mining?

В рыночной сегментации технологии «добычи данных» могут быть полезны для идентификации общих характеристик клиентов, которые покупают одинаковые изделия у компании, а также для предсказания, какие клиенты, вероятно, могут оставить компанию и уйти к конкурентам.

При анализе потребительской корзины данные технологии позволяют выявлять, какие изделия или услуги обычно приобретаются вместе.

При прямом маркетинге указанные технологии могут дать ответ на вопрос, какие проспекты должны быть включены в список рассылки, чтобы получить максимальную эффективность.

Добыча данных требует разработанной и хорошо построенной базы (хранилища) данных. Прежде чем любая организация подумает относительно добычи данных, нужно сначала убедиться, что необходимые данные имеются и что они являются полными и точными.

Технологии искусственного интеллекта (AI) основаны на моделировании мыслительной деятельности человека с использованием специальных программных средств и баз знаний.

В литературе дано определение базы знаний как набора эмпирических правил истинности заключений (высказываний) по данной теме (проблеме), базы эмпирических данных и описания проблемы, а также вариантов её решения.

Технологии искусственного интеллекта используются для решения задач, относящихся к классу неформализованных и слабо структурированных. Алгоритмизированное решение таких задач или не существует в силу неполноты, неопределенности, неточности, расплывчатости рассматриваемых ситуаций и знаний о них, или же такие решения неприемлемы на практике в силу сложности разрешающих

алгоритмов. Применительно к управлению указанные технологии в настоящее время находят применение в таких областях, как планирование и оперативное управление производством, выработка оптимальной стратегии поведения в соответствии со сложившейся ситуацией и т.д.

Развитие технологий искусственного интеллекта в управлении идет в направлении разработки экспертных систем.

Экспертные системы – это системы, которые используют логику принятия решения человека-эксперта и способны в сложных условиях дать квалифицированную консультацию (совет, подсказку, ориентацию) на основе логической переработки данных с целью получения новой информации, которая в явном виде в базу знаний не вводилась.

Экспертная система моделирует мыслительный процесс человека-эксперта, который является специалистом по решению определенного типа проблем. Такая система должна задавать вопросы пользователю, производить оценку ситуации и получать решение, представляемое в каком-либо виде пользователю. Кроме того, от экспертной системы могут потребоваться демонстрация способа, которым получено решение, и его обоснование.

Экспертные системы, являющиеся в настоящее время наиболее распространенным классом систем искусственного интеллекта, обладают способностью рассматривать большее число вариантов, чем это доступно человеку, при доскональном анализе ситуаций в той или иной предметной области и выдавать «интеллектуальные» решения в сложных ситуациях.

Другим направлением эффективного развития технологий искусственного интеллекта в управлении является развитие нейронных сетей.

Нейронные сети устроены по аналогии с тем, как работает человеческая нервная система, и способны распознавать модели, не достаточно ясные для людей, чтобы люди могли их обнаружить.

Фактически нейронные сети используют статистический анализ, чтобы распознавать модели из большого количества информации посредством адаптивного обучения.

Механизм работы нейронной сети следующий. Программе нейронных сетей сначала дается набор данных, состоящих из многих переменных, связанных с большим количеством случаев или исходов, в которых результаты известны.

Программа анализирует данные и обрабатывает все корреляции, а затем выбирает набор переменных, которые строго соотнесены с частными известными результатами, как начальную модель. Далее программа многократно изменяет модель, регулируя параметры переменных или даже заменяя их, пробуя предсказать результаты различных случаев и сравнивая предсказанные результаты с известными результатами. Данная процедура называется обучением сети. По окончании обучения программа готова делать предсказания для будущих случаев.

Геоинформационные технологии предназначены для обработки пространственно-временных данных, основой интеграции которых служит географическая информация. Указанные технологии обеспечивают принятие решений по оптимальному управлению землями и ресурсами, городским хозяйством, транспортом, розничной торговлей, экологией и другими сферами деятельности человека на основе интеграции данных географического характера.

На основе геоинформационных технологий разрабатывается широкий класс геоинформационных систем (ГИС), включающих такие типы, как системы картографической информации, автоматизированные системы картографирования, автоматизированные фотограмметрические системы, земельные информационные системы, автоматизированные кадастровые системы, системы информационного обеспечения администрации городов и т.п.

Инструментальные средства ГИС-технологий базируются на моделях представления пространственных данных, таких как векторная, топологическая и нетопологическая модели, квадродеревья, растровые модели, линейные сети. Их отличает ориентация на пользовательские модели данных с учетом предметной области и особенностей приложений, образующих единую информационную пользовательскую среду, дополняемую возможностями трехмерного проектирования, генерации планов, автоматического документирования проектов и выбора оптимальных вариантов. Данные технологии позволяют осуществлять векторно-топологическое моделирование, буферизацию объектов, анализ сетей, построение цифровых моделей местности и т.д.

Современные ГИС-технологии предназначены для формирования и ведения банков земельных данных о состоянии жилого и нежилого фондов, ведения кадастра недвижимости, анализа, оценки и планирования городских территорий, управления коммунальным хозяйством и т.д.

Такие различные области как управление природными ресурсами, государственная служба, военное и градостроительное проектирование используют ГИС-технологии уже более 30 лет. В начале 90-х годов указанные технологии привлекли внимание деловых пользователей. По мнению многих бизнесменов и аналитиков, сфера приложений ГИС-технологий безгранична.

Сфера применения ГИС-технологий в бизнесе охватывает разные области:

- анализ и отслеживание текущего состояния и тенденций изменения рынка;
- планирование деловой активности;
- оптимальный выбор местоположения новых филиалов фирмы или банка, торговых точек, складов, производственных мощностей;

- выбор кратчайших или наиболее безопасных маршрутов перевозок и путей распределения продукции;
- анализ риска материальных вложений и урегулирование разногласий;
- демографические исследования, проводимые в целях изучения спроса на продукцию;
- географическая привязка баз данных о земле- и домовладении.

В последнее время всё чаще встречается новое понятие геомаркетинг (geomarketing), связывающее в неразрывное целое бизнес и геоинформационные технологии.

В разных странах ГИС-технологии находят применение в бизнесе.

В перспективе особенно успешное и выгодное использование ГИС-технологий представляется при массовых перевозках грузов и людей, при создании сетей оптимально размещенных торговых точек, анализе существующих и потенциальных рынков и районов сбыта продукции, в нефтяных, газовых и электрических компаниях, а также в коммерческих фирмах, занимающихся операциями с недвижимостью, для обоснования, расширения и поддержки банковских операций, в работе авиакомпаний и телекоммуникационных корпораций, ряде других сфер деловой активности.

Технологии виртуальной реальности (VR) предназначены для создания окружающей среды, которая кажется реальной пользователю-человеку. Существует множество различных применений технологий виртуальной реальности в обороне, медицине, индустрии развлечений и т.п.

Использование виртуальной реальности в управлении и бизнесе разделяется на две основные категории: обучение и проектирование.

2. Лабораторный практикум

Лабораторный практикум предусматривает работу в СУБД MS Access. Вначале приводятся краткие сведения о базах данных и об особенностях работы в СУБД MS Access, а потом следуют сквозные задачи и задания для самостоятельного проектирования баз данных.

2.1. Базы данных как основная составляющая информационного обеспечения системы управления

2.1.1. Основные понятия теории баз данных

Базы данных (БД) – это совокупность сведений (о реальных объектах, процессах, событиях или явлениях), относящихся к определенной теме или задаче, организованная таким образом, чтобы обеспечить удобное представление этой совокупности как в целом, так и любой ее части. *Реляционная база данных* представляет собой множество взаимосвязанных таблиц, каждая из которых содержит информацию об объектах определенного типа. Каждая строка таблицы включает данные об одном объекте (например, клиенте, автомобиле, документе), а столбцы таблицы содержат различные характеристики этих объектов – *атрибуты* (например, наименования и адреса клиентов, марки и цены автомобилей). Строки таблицы называются *записями*; все записи имеют одинаковую структуру – они состоят из *полей*, в которых хранятся *атрибуты* объекта. Каждое поле записи содержит одну *характеристику* объекта и имеет строго определенный *тип данных* (например, текстовая строка, число, дата). Все записи имеют одни и те же поля, только в них содержатся разные значения атрибутов [4].

Для работы с данными используются системы управления базами данных (СУБД). Основные функции СУБД – это *определение данных* (описание структуры баз данных), *обработка данных*, *управление данными и администрирование*.

Прежде чем заносить данные в таблицы, нужно определить структуру этих таблиц. Под этим понимается не только описание наименований и типов полей, но и ряд других характеристик (например, формат, критерии проверки вводимых данных). Кроме описания структуры таблиц, обычно задаются *связи между таблицами*. Связи в реляционных базах данных определяются по совпадению значений полей в разных таблицах. Например, клиенты и заказы связаны отношением «один-ко-многим» (1:M), так как одной записи в таблице, содержащей сведения о клиентах, может соответствовать несколько записей в таблице заказов этих клиентов. Если же рассмотреть отношение между преподавателями и курсами лекций, которые они читают, это будет отношение «многие-ко-многим» (M:M), так как один преподаватель может читать несколько курсов, но и один курс может читаться несколькими преподавателями. И последний тип связей между таблицами – это отношение «один-к-одному» (1:1). Такой тип отношений встречается гораздо реже. Как правило, это бывает в случаях, когда нужно определить дополнительные атрибуты для некоторого количества записей в таблице, тогда создается отдельная таблица для этих дополнительных атрибутов, которая связывается отношением «один-к-одному» с основной таблицей.

Любая СУБД позволяет выполнять четыре простейшие операции с данными:

- добавлять в таблицу одну или несколько записей;
- удалять из таблицы одну или несколько записей;
- обновлять значения некоторых полей в одной или нескольких записях;
- находить одну или несколько записей, удовлетворяющих заданному условию.

Для выполнения этих операций используется *механизм запросов*. Результатом выполнения запросов является либо отобранное по

определенным критериям множество записей, либо изменения в таблицах. Запросы к базе формируются на специальном языке, который так и называется язык структурированных запросов (SQL – Structured Query Language).

Нужно отметить, что для работы с БД используются специальные языки, в целом называемые языками БД. В ранних СУБД поддерживалось несколько специализированных по своим функциям языков. Чаще всего выделялись два языка – язык определения схемы БД (SDL – Schema Definition Language) и язык манипулирования данными (DML – Data Manipulation Language). SDL служил главным образом для определения логической структуры БД, т.е. той структуры БД, какой она представляется пользователям. DML содержал набор операторов манипулирования данными, т.е. операторов, позволяющих заносить данные в БД, удалять, модифицировать или выбирать существующие данные.

В современных СУБД обычно поддерживается единый интегрированный язык, содержащий все необходимые средства для работы с БД, начиная от ее создания, и обеспечивающий базовый пользовательский интерфейс с базами данных. Стандартным языком наиболее распространенных в настоящее время реляционных СУБД является язык SQL. Язык SQL сочетает средства SDL и DML, т.е. позволяет определять схему реляционной БД и манипулировать данными.

И последняя функция СУБД – это управление данными и администрирование. Под управлением данными обычно понимают защиту данных от несанкционированного доступа, а под администрированием – поддержку многопользовательского режима работы с данными и обеспечение целостности и согласованности данных.

Защита от несанкционированного доступа обычно позволяет каждому пользователю видеть и изменять только те данные, которые ему разрешено видеть или менять. Средства, обеспечивающие

многопользовательскую работу, не позволяют нескольким пользователям одновременно изменять одни и те же данные. Средства обеспечения целостности и согласованности данных не дают выполнять такие изменения, после которых данные могут оказаться несогласованными. Например, когда две таблицы связаны отношением «один-ко-многим», нельзя внести запись в таблицу на стороне «многие» (ее обычно называют подчиненной), если в таблице на стороне «один» (главной) отсутствует соответствующая запись.

2.1.2. Microsoft Access как настольная СУБД реляционного типа

Microsoft Access является настольной СУБД реляционного типа, которая имеет все необходимые средства для выполнения перечисленных выше функций. Достоинством Access является то, что она имеет интуитивно понятный графический интерфейс, который позволяет не только создавать собственную базу данных, но и разрабатывать простые и сложные приложения.

В отличие от других настольных СУБД, Access хранит все данные в одном файле, хотя и распределяет их по разным таблицам. Пользователь может создать сколько угодно таблиц, используя те правила, которые были описаны выше. Самым важным правилом, которое необходимо соблюдать, является то, что в базе данных нужно хранить только необходимую информацию, при этом данные не должны дублироваться.

Общее правило при проектировании таблиц: если определяется какое-либо поле в таблице, нужно проверить, не хранятся ли уже эти данные в какой-нибудь другой таблице. Если хранятся, то их, как правило, можно, так или иначе, извлечь.

Чем же неудобно иметь в таблицах поля с повторяющимися данными? Дело в том, что это очень неэффективный способ хранения данных. И не только потому, что они занимают лишнее место в памяти.

Этот аргумент в последнее время не является таким сильным, как раньше из-за значительного снижения цен на микросхемы памяти. Основная причина – это то, что такие данные долго вводить и трудно анализировать. Если случайно при вводе значения пользователь сделал грамматическую ошибку или даже просто ввел лишний пробел, то при запросах и группировках такое значение будет рассматриваться как самостоятельное, и строка, содержащая это значение, не попадет в нужную группу или просто не будет выведена на экран. Именно поэтому при проектировании структуры баз данных стараются избегать повторения данных и создают для них отдельные таблицы. Этот процесс называется нормализацией.

Просматривать, изменять и удалять данные можно прямо в таблицах. Однако часто удобнее использовать для этого специально созданные экранные формы, которые могут иметь как табличный вид, когда в форме таблицы отображается сразу несколько записей, так и показывать только одну запись в виде некоторого бланка. В формах можно отображать информацию сразу из нескольких таблиц. В этом случае в основе формы будет не таблица, а запрос. Запрос позволяет объединять данные из нескольких связанных таблиц, выполнять вычисления над данными из других столбцов таблицы, добавлять, изменять и удалять записи в таблицах. Для создания запросов используется язык SQL. Но Access позволяет автоматически формировать запросы в результате интерактивного процесса конструирования. Поэтому в большинстве случаев не нужно досконально знать язык SQL, так как всегда можно посмотреть тот запрос, который сформирует Access.

Для того чтобы можно было создавать печатные копии информации, хранящейся в базе данных Access, предназначены отчеты, которые позволяют извлекать необходимые данные, группировать и сортировать их в нужном виде, вычислять итоговые значения по группам и в целом по всем обработанным записям. Отчеты могут быть дополнены

рисунками, диаграммами, содержательными комментариями, могут быть выведены на печать, преобразованы в документ Word или опубликованы на Web.

Другим удобным средством, которым можно пользоваться, чтобы публиковать данные в сети интранет или интернет, являются страницы доступа к данным.

Они могут содержать не только статическую информацию для просмотра отчетов, но могут также иметь вид динамических страниц, позволяющих пользователю создавать запросы к базе данных, которые извлекают или изменяют нужные данные.

Для выполнения всех перечисленных операций Access предлагает большое количество Мастеров и Построителей, которые помогут быстро научиться работать с данными и добиваться желаемых результатов, а также избежать рутинных действий, которые присутствуют всегда, даже в такой творческой работе, как разработка приложений в Access.

СУБД Access является системой управления базой данных реляционной структуры, включающей все необходимые инструментальные средства для создания локальной базы данных, общей базы данных в локальной сети с файловым сервером или создания приложения пользователя, работающего с базой данных на SQL-сервере.

Реляционная база данных представляет собой множество взаимосвязанных двумерных таблиц – *реляционных таблиц*, в каждой из которых содержатся сведения об одной сущности.

В БД должны храниться данные, логически связанные между собой. Для того чтобы данные можно было связать между собой, и связать так, чтобы эти связи соответствовали реально существующим в данной предметной области, последнюю подвергают детальному анализу, выделяя сущности или объекты. Сущность или объект — это то, о чем необходимо хранить информацию. Сущности имеют некоторые характеристики,

называемые атрибутами, которые тоже необходимо сохранять в БД. Атрибуты по своей внутренней структуре могут быть простыми, а могут быть сложными. Простые атрибуты могут быть представлены простыми типами данных. Различного рода графические изображения, являющиеся атрибутами сущностей, — это пример сложного атрибута. Определив сущности и их атрибуты, необходимо перейти к выявлению связей, которые могут существовать между некоторыми сущностями. Связь — это то, что объединяет две или более сущностей. Связи между сущностями также являются частью данных, и они также должны храниться в базе данных. Обычно все вышперечисленное отображается в совокупности на ER-диаграмме или диаграмме «сущность-связь» (рис. 4).



Рис. 4. Диаграмма «сущность-связь»

На рис. 4 сущности отображены в прямоугольниках, их атрибуты в овалах, а связь между ними – в многоугольнике.

Структура реляционной таблицы определяется составом и последовательностью полей, соответствующих ее столбцам, с указанием типа элементарного данного, размещаемого в поле. Каждое поле отражает определенную характеристику сущности (атрибут), а соответствующий столбец содержит данные определенного типа.

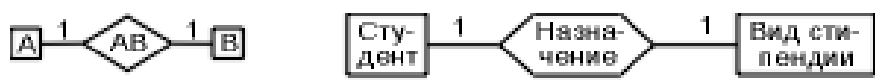
Содержание таблицы заключено в ее строках. Каждая строка таблицы содержит данные о конкретном экземпляре сущности и называется записью. Для однозначного определения каждой записи таблица должна иметь *уникальный (первичный) ключ*, который может

состоять из одного или нескольких полей. По значению ключа отыскивается единственная запись.

Связи между таблицами дают возможность совместного использования данных из разных таблиц. В нормализованной реляционной базе данных связи характеризуются отношениями типа «один-к-одному» (1:1) или «один-ко-многим» (1:M). Связь каждой пары таблиц обеспечивается одинаковыми полями в них – *ключом связи*. Ключом таких связей всегда является уникальный ключ главной таблицы в связи. В подчиненной таблице он называется *внешним ключом*.

Между двумя сущностями, например, А и В возможны четыре вида связей.

Первый тип – связь ОДИН-К-ОДНОМУ (1:1): в каждый момент времени каждому представителю (экземпляру) сущности А соответствует 1 или 0 представителей сущности В:



Студент может не «заработать» стипендию, получить обычную или одну из повышенных стипендий.

Второй тип – связь ОДИН-КО-МНОГИМ (1:M): одному представителю сущности А соответствуют 0, 1 или несколько представителей сущности В.



Квартира может пустовать, в ней может жить один или несколько жильцов.

Так как между двумя сущностями возможны связи в обоих направлениях, то существует еще два типа связи МНОГИЕ-К-ОДНОМУ (M:1) и МНОГИЕ-КО-МНОГИМ (M:N).

Например, две таблицы ГРУППА с перечнем студенческих групп и СТУДЕНТ со списком студентов логически связаны между собой по

одинаковому полю номер группы связью типа 1:M. Размещение сведений о каждой сущности в отдельной таблице и связывание таблиц позволяет избежать повторения значений данных в разных таблицах и упрощает процесс их обновления и поиска в базе. При этом обеспечивается однократный ввод данных при загрузке и корректировке. Если же данные двух таблиц в приведенном примере разместить в одной таблице, то каждая запись должна соответствовать одному студенту, причем данные о группе (количество студентов, проходной балл) будут повторяться во всех записях о студентах одной группы.

Очень важным моментом при создании реляционной БД является построение схемы данных (рис. 5). Она наглядно отображает таблицы и связи между ними, обеспечивает использование связей при обработке данных. Здесь устанавливаются параметры обеспечения целостности связей в БД.

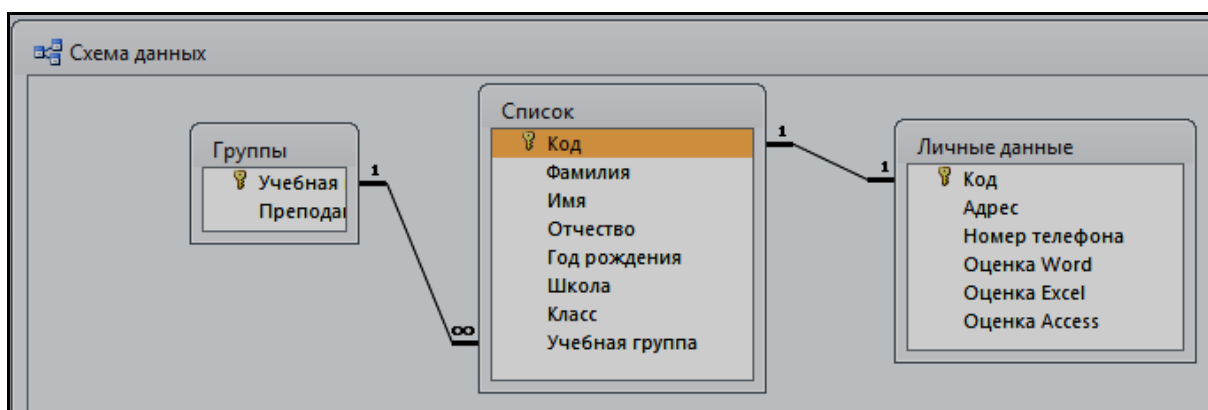


Рис. 5. Схема базы данных

2.1.3. Объекты MS Access

Access ориентирована на работу с объектами, к которым относятся: таблицы БД, запросы, а также объекты приложений для работы с базой данных: формы, отчеты, страницы, макросы и модули. Объекты представлены в окне БД Access. Все операции по работе с объектами БД и приложений начинаются в этом окне (рис. 6).

Таблицы создаются пользователем для хранения данных об одном информационном объекте модели данных предметной области. Таблица состоит из полей (столбцов) и записей (строк). Каждое поле содержит одну характеристику объекта предметной области. В записи собраны сведения об одном экземпляре этого объекта.

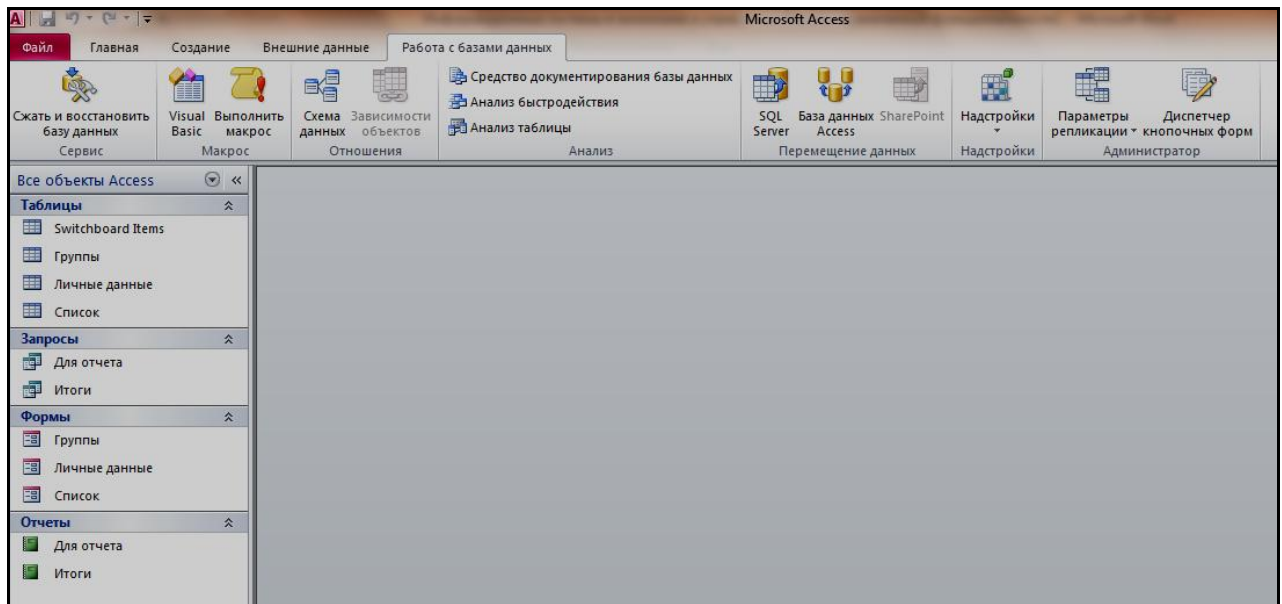


Рис. 6. Окно базы данных MS Access

Запросы создаются пользователем для выборки данных из одной или нескольких связанных таблиц. Результатом выполнения запроса является таблица, которая может быть использована наряду с другими таблицами БД при обработке данных. С помощью запросов можно обновлять, удалять, добавлять данные в таблицы или создавать новые таблицы на основе существующих таблиц и запросов.

Формы являются основным средством создания диалогового интерфейса приложения пользователя. Форма создается для ввода и просмотра взаимосвязанных данных на экране в удобном виде, соответствующем и привычном для пользователя.

Отчеты предназначены для формирования выходных документов, содержащих результаты решения задач пользователя, и вывода их на печать.

Страницы – это страницы доступа к данным; являются диалоговыми Web-страницами, которые поддерживают динамическую связь с базой данных и позволяют просматривать, редактировать и вводить данные в базу, работая в окне браузера.

Макросы. Макрос является программой, которая содержит описание последовательности действий, выполняемых при наступлении некоторого события в объекте или элементе управления приложения. Каждое действие реализуется макрокомандой.

Модули содержат процедуры на языке VBA. Пользователь может разрабатывать процедуры-функции для реализации нестандартных функций или процедуры для обработки событий.

Все таблицы БД, а также другие объекты Access: формы, запросы, отчеты, макросы и модули, построенные для этой базы, и внедренные объекты могут размещаться на диске в одном файле формата *.accdb.

2.1.4. Схема создания БД с помощью MS Access

Перед созданием базы данных с помощью Access пользователь должен определить, из каких таблиц должна состоять база данных, какие данные нужно поместить в каждую таблицу, как связать таблицы. Эти вопросы решаются на этапе проектирования базы данных.

В результате проектирования должна быть определена логическая структура базы данных, то есть состав реляционных таблиц, их структура и межтабличные связи.

Инфологическая модель отображает данные предметной области в виде совокупности информационных объектов и связей между ними. Эта модель представляет данные, подлежащие хранению в базе данных.

Информационный объект – описание некоторой сущности предметной области – реального объекта, процесса, явления или события. Он образуется совокупностью реквизитов, логически взаимосвязанных и представляющих качественные и количественные характеристики сущности. Реквизиты каждого информационного объекта должны отвечать *требованиям нормализации*: информационный объект должен содержать уникальный идентификатор – *ключ*:

- ключ является *простым*, если состоит из одного реквизита или *составным*, если состоит из нескольких реквизитов;

- все описательные реквизиты (т.е. зависимые от ключевых) должны быть взаимонезависимыми, то есть между ними не должно быть функциональных зависимостей;

- все реквизиты, входящие в составной ключ, должны быть взаимонезависимыми;

- каждый описательный реквизит должен зависеть от ключа, т.е. каждому значению ключа должно соответствовать только одно значение описательного реквизита;

- при составном ключе описательные реквизиты должны зависеть целиком от всей совокупности реквизитов, образующих ключ;

- каждый описательный реквизит не должен зависеть от ключа транзитивно, то есть через другой промежуточный реквизит;

- в случае транзитивной зависимости между реквизитами можно выполнить расщепление совокупности реквизитов с образованием двух информационных объектов вместо одного.

Связи между объектами модели данных реализуются одинаковыми реквизитами – *ключами связи* в соответствующих таблицах. При этом ключом связи всегда является уникальный (первичный) ключ главной таблицы. Ключом связи в подчиненной таблице является либо некоторая часть уникального ключа в ней, либо поле, не входящее в состав

первичного ключа (например, код кафедры в таблице ПРЕПОДАВАТЕЛЬ на рис. 7). Ключ связи в подчиненной таблице называют *внешним ключом*.

В процессе непосредственного создания БД сначала конструируются ее таблицы, соответствующие информационным объектам построенной модели данных. Далее создается схема данных, в которой фиксируются существующие логические связи между таблицами (рис. 8). Эти связи должны соответствовать связям информационных объектов. В схеме данных могут быть заданы параметры для поддержки целостности БД, если модель была разработана в соответствии с требованиями нормализации.

| Табельный | ФИО препо | Уч. степень | Уч. звание | Код кафедры |
|-----------|---------------|-------------|------------|-------------|
| 101 | Игнатьев В.В. | д. техн. н. | профессор | 01 |
| 102 | Антонов К.Е. | к.техн.н. | доцент | 01 |
| 103 | Глухова Ж.А. | нет | ст. преп. | 01 |
| 201 | Кораблев П.И | д.фил.н. | профессор | 02 |
| 202 | Чернова Н.А. | к.фил.н. | доцент | 02 |
| 301 | Жданова А.Е. | д.пед.н. | профессор | 03 |
| 302 | Новиков П.Н. | нет | ст.преп. | 03 |
| 401 | Смирнова И.В | к.ист.н. | доцент | 04 |
| 402 | Романова А.В. | нет | ассистент | 04 |
| 501 | Ломоносов М. | д.ф.-м.-н. | профессор | 05 |
| 502 | Львова П.Я. | к.ф.-м.н. | доцент | 05 |
| 503 | Плутков О.Н. | нет | ст.преп. | 05 |
| 601 | Бондаренко В | к.ф.н. | доцент | 06 |
| 602 | Кудряшова Т.Г | нет | ассистент | 06 |
| * | | | | |

Рис. 7. Таблица «Преподаватель»

Целостность данных означает, что в БД установлены и корректно поддерживаются взаимосвязи между записями разных таблиц при загрузке, добавлении и удалении записей в связанных таблицах, а также при изменении значений ключевых полей. При этом должны соблюдаться условия:

- в подчиненную таблицу не может быть добавлена запись с несуществующим в главной таблице значением ключа связи;

- в главной таблице нельзя удалить запись, если не удалены связанные с ней записи в подчиненной таблице;
- изменение значений ключа связи в записи главной таблицы невозможно, если в подчиненной таблице имеются связанные в ней записи.

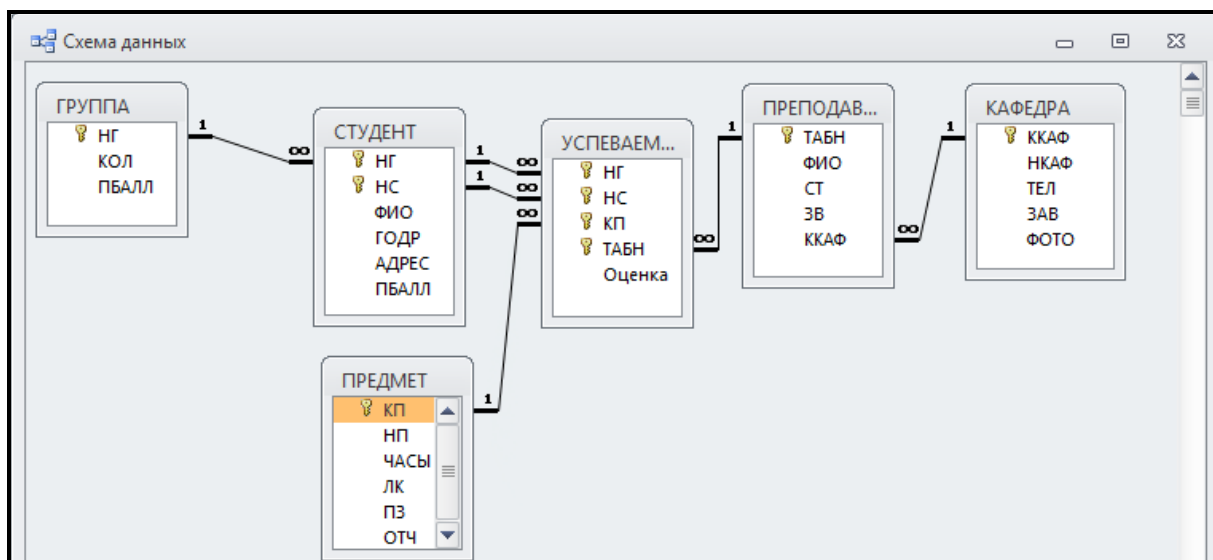


Рис. 8. Схема данных базы данных MS Access

При попытках пользователя нарушить эти условия Access выводит соответствующее сообщение и не допускает выполнение некорректных операций.

Установление между двумя таблицами связей типа 1:1 или 1:М и задание параметров целостности возможно только при следующих условиях:

- связываемые поля имеют одинаковый тип данных, причем имена полей при этом могут быть различными;
- обе таблицы сохраняются в одной базе данных Access;
- главная таблица связывается с подчиненной по первичному простому или составному ключу (уникальному индексу) главной таблицы.

Если для выбранной связи обеспечивается поддержка целостности, можно задать режим каскадного обновления связанных полей и режим каскадного удаления связанных записей.

Создание БД завершается процедурой загрузки, т.е. заполнения таблиц конкретными данными. Удобным инструментом загрузки данных во взаимосвязанные таблицы являются формы ввода-вывода, обеспечивающие интерактивный интерфейс для работы с данными базы. Формы позволяют создать экранный аналог документа-источника, через который можно вводить данные в несколько взаимосвязанных таблиц.

2.1.5. Таблицы базы данных в СУБД Access

Создание реляционной БД начинается с формирования *структуры таблиц*. При этом формируется состав полей и задается их описание. После определения структуры таблиц создается *схема данных*, в которой устанавливаются связи между таблицами. Access запоминает и использует эти связи при заполнении таблиц и обработке данных.

Очень важно при этом задать параметры, в соответствии с которыми Access будет автоматически поддерживать целостность данных. Для этого при определении структуры таблиц должны быть указаны *ограничения* на допустимые значения данных, а при создании схемы данных на основе нормализованных таблиц должны быть заданы параметры поддержания *целостности связей* БД.

Создание таблицы БД состоит из двух этапов. *На первом этапе* определяется ее структура: состав полей, их имена, последовательность размещения полей в таблице, тип данных каждого поля, размер поля, ключи и другие свойства полей. *На втором этапе* производится создание записей таблицы, т.е. заполнение таблицы данными.

Для создания новой таблицы в режиме конструктора надо в окне БД выбрать Создание/Конструктор таблиц. Это позволяет сразу перейти в нужный режим создания таблицы.

При выборе *режима конструктора* таблиц появляется окно *Таблица 1*, в котором определяется структура таблицы БД. При этом меняется состав команд меню и панель инструментов.

Для определения каждого поля нужно задать *Имя поля*, *Тип данных*, *Описание* (краткий комментарий – по желанию) и *Свойства поля* (раздел имеет две вкладки *Общие* – здесь задаются размер, формат, подпись, маска ввода и другие и *Подстановка* – здесь выбирается тип элемента управления: поле, список или поле со списком).

Имя поля. Каждое поле в таблице должно иметь уникальное имя, удовлетворяющее требованиям Access. Это может быть комбинация из букв и цифр, а также пробелов и специальных символов, за исключением символов: «.», «!», «'», «[», «]». Имя не может начинаться с пробела. Максимальная длина 64 символа.

Тип данных. Тип данных определяется значениями, которые предположительно будут храниться в поле, и операциями, которые будут выполняться с этими значениями. Список возможных типов вызывается нажатием кнопки раскрывающегося списка при выборе типа данных в каждом поле (всего их 9):

- *Текстовый* – тип данных по умолчанию. Текст или цифры, не участвующие в расчетах. Число символов не должно превышать 255;
- *Поле МЕМО* – Текст большого объема, некоторое описание, примечание. Максимальная длина 64 000 символов;
- *Числовой* – числовые данные, используемые в математических вычислениях. Более конкретные варианты числового типа и их длина задаются в свойстве Размер поля. Для проведения денежных расчетов не используется;
- *Денежный* – денежные значения или числовые данные, используемые в расчетах, проводящихся с точностью до 15 знаков в целой и до 4 знаков в дробной части. Длина поля 8 байт;

- *Дата/Время* – значения даты или времени, относящиеся к годам с 100 по 9999 включительно. Длина поля 8 байт;

- *Счетчик* – тип данных поля, в которое для каждой новой записи автоматически вводятся уникальные целые, последовательно возрастающие (на 1), или случайные числа. Значения этого поля нельзя изменить или удалить. В таблице не может быть более одного поля такого типа. Используется для определения уникального ключа таблицы;

- *Логический* – логические данные, которые могут иметь одно из двух возможных значений Да/Нет, Истина/Ложь;

- *Поле объекта OLE* – объект, связанный или внедренный в таблицу Access. Это может быть электронная таблица Excel, документ Word, рисунок, звукозапись или другие данные в двоичном формате. Длина поля до 1 Гб. Для полей типа OLE и MEMO не допускается сортировка и индексирование;

- *Гиперссылка* – в качестве гиперссылки можно указывать путь к файлу на жестком диске, в сети, в глобальной сети;

- *Мастер подстановок* – запускает мастер подстановок, который строит для поля список значений, подставляемый из полей другой таблицы. Соответственно, фактически тип данных поля определяется типом данных поля списка.

Как уже говорилось ранее, каждая таблица в реляционной базе данных должна содержать уникальный (первичный) ключ, который может быть простым или составным, то есть включающим несколько полей (до 10). Для определения ключа нужно выделить поля, составляющие ключ, и нажать кнопку *Ключевое поле*.

Если в таблице не задано ключевое поле, то при попытке ее сохранения Access обязательно спросит о необходимости создания такого ключа. Получив ответ «Да», он автоматически создаст поле с типом *Счетчик* и присвоит ему свойство *Первичного ключа*.

После определения структуры таблицы ее надо сохранить. Для этого используется команда *Файл/Сохранить* или кнопка на панели инструментов *Сохранить*. В окне *Сохранение* таблице нужно присвоить имя. При сохранении таблицы происходит обновление файла базы данных, в которую помещается созданная таблица.

После сохранения таблицы становится доступным режим, позволяющий перейти ко второму этапу – созданию записей таблицы. Этот режим называется *режим таблицы*. Он осуществляется нажатием кнопки **Вид** на панели инструментов в разделе *Главная*.

Предположим, нам нужно создать простейшую базу данных для учета успеваемости студентов, состоящую из трех таблиц, связанных между собой (рис. 9).

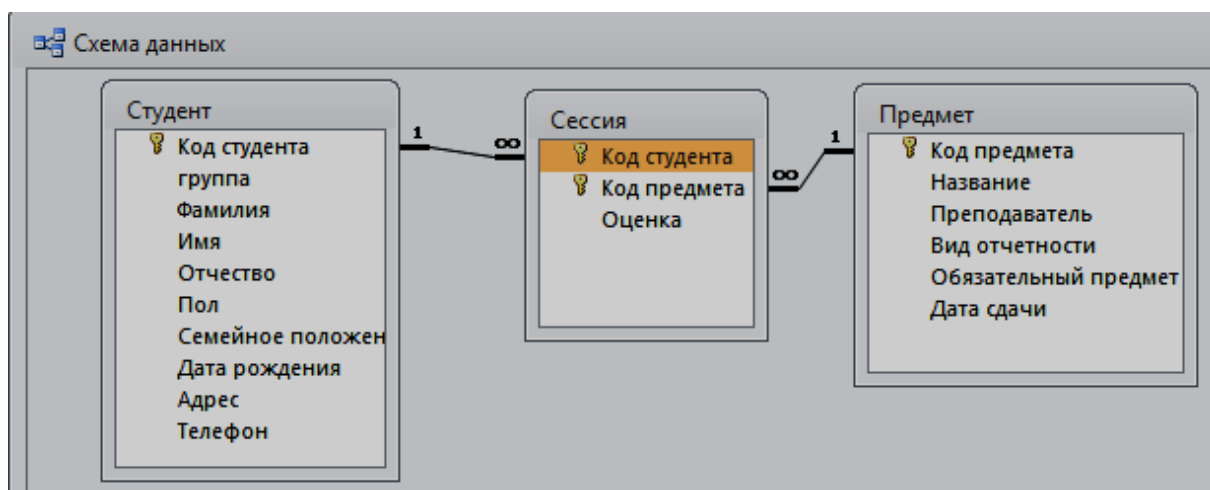


Рис. 9. Схема данных, ключевые поля и связи таблиц

Чтобы создать такую базу данных, нужно последовательно создать таблицы Предмет, Студент и Сессия. Сначала в режиме Конструктора задается структура каждой таблицы (рис. 10, 12, 14), а затем в режиме Таблицы она наполняется соответствующим содержанием (рис. 11, 13, 15).

| Предмет | |
|----------------------|------------|
| Имя поля | Тип данных |
| Код предмета | Числовой |
| Название | Текстовый |
| Преподаватель | Текстовый |
| Вид отчетности | Текстовый |
| Обязательный предмет | Логический |
| Дата сдачи | Дата/время |

Рис. 10. Конструктор таблицы «Предмет»

| Предмет | | | | | | |
|---------|------------|--------------|---------------|------------|-------------------------------------|------------|
| | Код предме | Название | Преподават | Вид отчетн | Обязательн | Дата сдачи |
| + | 21 | Экономика | Бобов И.А. | экзамен | <input checked="" type="checkbox"/> | 12.06.2007 |
| + | 32 | Культурологи | Приборов И.С | экзамен | <input checked="" type="checkbox"/> | 23.06.2007 |
| + | 33 | Философия | Никулин П.А. | экзамен | <input type="checkbox"/> | 30.06.2007 |
| + | 34 | Правоведени | Маслова Е.А. | зачет | <input checked="" type="checkbox"/> | 29.05.2007 |
| + | 35 | Физкультура | Куров П.Л. | зачет | <input checked="" type="checkbox"/> | 20.05.2007 |
| + | 36 | Математика | Львова Т.В. | зачет | <input checked="" type="checkbox"/> | 25.05.2007 |
| + | 39 | Финансовая м | Арбузова Р.О. | экзамен | <input checked="" type="checkbox"/> | 10.06.2007 |
| + | 43 | Статистика | Северов Н.П. | экзамен | <input type="checkbox"/> | 18.06.2007 |
| + | 46 | Проектирова | Додонов А.И. | зачет | <input checked="" type="checkbox"/> | 24.05.2007 |
| + | 48 | Экология | Володин И.И. | зачет | <input type="checkbox"/> | 31.05.2007 |
| + | 50 | Макроэконом | Макарычева И | экзамен | <input checked="" type="checkbox"/> | 19.06.2007 |
| * | 0 | | | | <input type="checkbox"/> | |

Рис. 11. Содержание таблицы «Предмет»

| Студент | |
|--------------------|------------|
| Имя поля | Тип данных |
| Код студента | Числовой |
| группа | Текстовый |
| Фамилия | Текстовый |
| Имя | Текстовый |
| Отчество | Текстовый |
| Пол | Текстовый |
| Семейное положение | Логический |
| Дата рождения | Текстовый |
| Адрес | Текстовый |
| Телефон | Текстовый |

Рис. 12. Конструктор таблицы «Студент»

| Студент | | | | | | | | | | |
|-------------|--------|------------|---------|-------------|-----|-------------------------------------|------------|----------------|----------|--|
| Код студент | группа | Фамилия | Имя | Отчество | Пол | Семейное г | Дата рожде | Адрес | Телефон | |
| 123 | МН-10 | Комаров | Иван | Павлович | м | <input type="checkbox"/> | 12.03.1987 | ул. Академич | 67-17-80 | |
| 125 | БУ-10 | Мухин | Андрей | Афанасьевич | м | <input type="checkbox"/> | 23.05.1988 | ул.Лермонтов | 35-15-67 | |
| 127 | МН-10 | Маркина | Елена | Анатольевна | ж | <input type="checkbox"/> | 30.12.1987 | ул. Героев По | 78-18-00 | |
| 128 | ЭС-25 | Белоконь | Тамара | Игнатьевна | ж | <input type="checkbox"/> | 20.09.1988 | пр. Революци | 55-32-07 | |
| 129 | ЭС-25 | Серодуб | Евгений | Эдуардович | м | <input checked="" type="checkbox"/> | 24.10.1987 | Зеленский съ | 34-23-34 | |
| 130 | ЭС-25 | Дубова | Алена | Ренатовна | ж | <input type="checkbox"/> | 04.04.1988 | ул.Космическ | 45-56-67 | |
| 131 | ЭС-25 | Николаева | Ольга | Игоревна | ж | <input type="checkbox"/> | 10.10.1988 | ул. Ленина, д | 60-33-09 | |
| 132 | ЭС-25 | Самарин | Илья | Тимофеевич | м | <input checked="" type="checkbox"/> | 31.12.1987 | пер. Приволь | 21-12-21 | |
| 133 | БУ-10 | Максимова | Роза | Зиновьевна | ж | <input type="checkbox"/> | 15.05.1988 | ул. Минина, д | 32-16-78 | |
| 134 | БУ-10 | Суслов | Олег | Николаевич | м | <input type="checkbox"/> | 09.09.1988 | ул. Родионов | 38-57-09 | |
| 135 | МН-10 | Козловский | Михаил | Дмитриевич | м | <input type="checkbox"/> | 29.03.1987 | ул.Белинског | 35-65-90 | |
| 136 | МН-10 | Оськина | Людмила | Петровна | ж | <input checked="" type="checkbox"/> | 07.06.1988 | ул.Лопатина, | 77-09-87 | |
| 137 | МН-10 | Фокин | Алексей | Иванович | м | <input type="checkbox"/> | 17.07.1987 | ул.Бекетова, д | 65-87-33 | |
| 138 | МН-10 | Кулакова | Анна | Витальевна | ж | <input checked="" type="checkbox"/> | 27.08.1988 | ул. Минина, д | 36-05-12 | |
| * | 0 | | | | | <input type="checkbox"/> | | | | |

Рис. 13. Содержание таблицы «Студент»

| Сессия | |
|--------------|------------|
| Имя поля | Тип данных |
| Код студента | Числовой |
| Код предмета | Числовой |
| Оценка | Числовой |

Рис. 14. Конструктор таблицы «Сессия»

Чтобы код студента и код предмета при заполнении подставлялись автоматически, нужно выбрать тип данных «Мастер подстановки» и подставить нужное поле из соответствующей таблицы. При таком типе данных происходит автоматическое связывание таблиц базы данных, что можно наблюдать, если загрузить схему данных.

| Сессия | | | |
|-------------|------------|--------|--|
| Код студент | Код предмк | Оценка | |
| 123 | 21 | 5 | |
| 123 | 32 | 4 | |
| 123 | 33 | 4 | |
| 125 | 34 | 2 | |
| 125 | 35 | 3 | |
| 125 | 39 | 3 | |

Рис. 15. Содержание таблицы «Сессия»

2.1.6. Сущность и виды запросов

Одним из основных инструментов выборки и обработки данных в СУБД являются запросы. В Access имеется удобное для пользователя диалоговое графическое средство формирования *запроса по примеру* (Query By Example, QBE), с помощью которого легко может быть построен сложный запрос.

Запрос строится на основе одной или нескольких взаимосвязанных таблиц. При этом могут быть использованы как таблицы БД, так и сохраненные таблицы, полученные в результате ранее созданных запросов. Кроме того, запрос может строиться непосредственно на другом запросе, используя его временную таблицу с результатами.

В Access может быть создано несколько видов запросов:

1. *Запрос на выборку* – выбирает данные из взаимосвязанных таблиц и других запросов. Результатом его является таблица, которая существует до закрытия запроса. На основе этого вида запроса создаются другие запросы.

2. *Запрос на создание таблицы* – использует запрос на выборку, но в отличие от него, результат запроса сохраняется в новой таблице.

3. *Запросы на обновление, добавление, удаление* – являются запросами действия (изменения), в результате выполнения которых изменяются данные в таблице.

Последовательное выполнение ряда *запросов по примеру* (QBE) позволяет решать достаточно сложные задачи, не прибегая к программированию.

Основные принципы конструирования различных запросов QBE заложены в технике конструирования запроса на выборку, являющегося основой всех видов запросов.

Для выполнения действий по созданию запроса используются команды меню и панель инструментов *Конструктора запросов*. Они

становятся доступными после выполнения команды в окне БД: *Создание/Конструктор запросов*. Затем в открывшемся диалоговом окне *Добавление таблицы* нужно выделить используемые в запросе таблицы (или ранее созданные запросы) и нажать кнопку *Добавить*. По окончании добавления нажать кнопку *Заккрыть*.

Появится основное окно *Конструктор запросов*. Оно разделено на две панели. *Верхняя панель* содержит схему данных запроса, которая включает выбранные для данного запроса таблицы. Между ними обязательно должны быть установлены связи! Каждая таблица представлена списком полей, с выделением ключевых. Первая строка в записи полей помечена звездочкой *, она обозначает все множество полей таблицы. Access может автоматически установить связи между полями таблиц, если они имеют поля с одинаковыми именами и типами данных, даже если эти связи не были установлены в схеме данных. *Связи-объединения*, которые не установил Access, могут быть установлены пользователем путем перетаскивания мышкой названий полей из одной таблицы в другую.

Нижняя панель является *бланком запроса QBE*, который нужно заполнить. Он является таблицей, которая предназначена для конструирования структуры таблицы результата запроса и условий выборки данных из исходных таблиц. Первоначально эта таблица пуста.

Каждый столбец бланка относится к одному полю, которое будет включено в таблицу запроса. Поля могут быть использованы для выполнения сортировки, задания условий отбора, проведения необходимых изменений и вычислений. При заполнении бланка запроса:

- в строку *Поле* включаются имена полей, используемых в запросе;
- в строке *Сортировка* выбирается порядок сортировки записей результата;

- в строке *Вывод на экран* отмечаются поля, которые должны быть включены в результирующую таблицу;
- в строке *Условие отбора* задаются условия отбора записей;
- в строке *или* задаются альтернативные условия отбора записей.

Для включения имен полей в строку *Поле* пользуются следующими приемами:

- в строке *Поле* щелчком мыши вызывают появление кнопки списка и выбирают из него нужное поле;
- перетаскивают нужное поле из списка полей таблицы в схеме данных запроса;
- дважды щелкают мышкой по имени поля таблицы в схеме данных запроса;
- для включения всех полей таблицы дважды щелкнуть по символу звездочки в списке полей таблицы в схеме данных запроса.

Условие отбора записей может создаваться для одного или нескольких полей в строке *Условие отбора*. Условием отбора является *выражение*, которое состоит из *операторов сравнения* и *операндов*, используемых для сравнения.

В качестве *операндов* выражения используются *литералы*, *константы*, *идентификаторы (ссылки)*.

Литералами являются конкретные значения, воспринимаемые Access так, как они записаны. Это могут быть числа, текстовые строки, даты. Текстовые строки заключаются в двойные кавычки, даты - в символы #. Например, 567, «Информатика», #1-Января-04#.

Константами называют неизменяющиеся значения, которые определены в Access, например «True», «False», «Да», «Нет», «Null».

Идентификатор представляет собой ссылку на поле, элемент управления или свойство. Идентификаторами могут быть имена полей, таблиц, форм, отчетов. Они должны заключаться в квадратные скобки [...].

Если указывается ссылка на поле конкретной таблицы, то перед именем поля ставится имя таблицы, также заключенное в квадратные скобки, и отделенное от имени поля знаком «!». Например, [Студент]![Год рождения].

Операторами сравнения и логическими операторами могут быть:

- =, <, >, <>;
- Between - And позволяют задать интервал для числовых значений или дат;
- Or, And, Not;
- Like – позволяет использовать образцы с символами шаблона * при поиске в текстовых полях;
- In – позволяет выполнить проверку на равенство любому значению из списка, который задается в круглых скобках. Например, In («Математика», «Информатика», «Экономика»).

Построитель выражений. Условие отбора можно формировать с помощью *Построителя выражений*, который вызывается путем нажатия кнопки *Построить* на панели инструментов. Курсор мышки при этом должен быть предварительно установлен в ячейке ввода условия отбора. После создания выражения нажимается <ENTER>, а Access автоматически анализирует синтаксис созданного выражения и отображает его с необходимыми исправлениями.

Вычисляемые поля. В запросе над полями могут производиться вычисления. Результат вычисления образует вычисляемое поле в таблице, создаваемой по запросу. При каждом выполнении запроса производится вычисление с использованием текущих значений полей.

При вычислениях могут использоваться арифметические выражения и встроенные функции Access.

Арифметические выражения. Выражение вводится в бланк запроса в пустую ячейку строки Поле. Затем, после нажатия клавиши <ENTER>

или перевода курсора в другую ячейку перед выражением в этой ячейке добавляется имя поля **ВыражениеN**, где **N** – целое число, увеличивающееся на единицу для каждого нового создаваемого вычисляемого поля в запросе. Имя вычисляемого поля, стоящее перед выражением, отделяется от него двоеточием. Например, **<Выражение1>:[Цена]*[Количество]**, где **Цена** и **Количество** – имена полей. Имя вычисляемого поля (**Выражение1**) становится заголовком столбца в таблице с результатами выполнения запроса. Это имя можно изменить.

Для вычисляемых полей допускается сортировка, задание условий отбора и расчет итоговых значений, как и для любых других полей.

Встроенные функции. В Access имеются встроенные функции, которые можно использовать в вычисляемых полях:

- функция **Date** формирует текущую дату;
- функция **Month** выделяет месяц из значения поля, содержащего дату;
- функция **DLookup** возвращает значение конкретного поля из записи связанной таблицы, не участвующей в запросе;
- статистические функции над полями подмножества записей, вычисляющие среднее значение, сумму, минимальное, максимальное значение.

Для записи выражения также может быть использован **Построитель выражений**.

Использование групповых операций в запросах

Групповые операции позволяют выделить группы записей с одинаковыми значениями в указанных полях и использовать для этих групп одну из статистических функций, предусмотренных в Access:

- **Sum** – сумма значений некоторого поля для группы;
- **Avg** – среднее от всех значений поля в группе;

- **Max, Min** – максимальное, минимальное значения поля в группе;
- **Count** – число значений поля в группе без учета пустых значений;
- **Stdev** – среднеквадратичное отклонение от среднего значения поля в группе;
- **Var** – дисперсия значений поля в группе;
- **First и Last** - значение поля из первой или последней записи в группе.

Для создания запроса с использованием групповых операций формируется запрос на выборку. В бланк запроса включаются поля, по которым надо произвести группировку, и поля, по которым надо произвести статистические вычисления.

Выполняется команда Вид/Групповые операции или на панели инструментов конструктора запросов нажимается кнопка Групповые операции. После этого в бланке запроса появится строка Групповая операция, в которой для всех полей записана Группировка.

Для групповых вычислений по некоторому полю нужно заменить в нем слово Группировка на нужную статистическую функцию. Выбрать нужную функцию можно через раскрывающийся в поле список.

2.1.7. Параметрические запросы

Конкретное значение поля в условиях отбора может вводиться непосредственно в бланк запроса. Однако в процессе решения задач может возникнуть необходимость вводить значение поля в диалоге с пользователем в процессе выполнения запроса. Для того, чтобы выводилось диалоговое окно, обеспечивающее ввод конкретного значения из поля в условия отбора, необходимо определить *параметр запроса*. Это позволит также избежать большого количества однотипных запросов –

заменить их одним – параметрическим. Имя параметра запроса может задаваться непосредственно в строке «Условие отбора» в квадратных скобках (рис. 16).

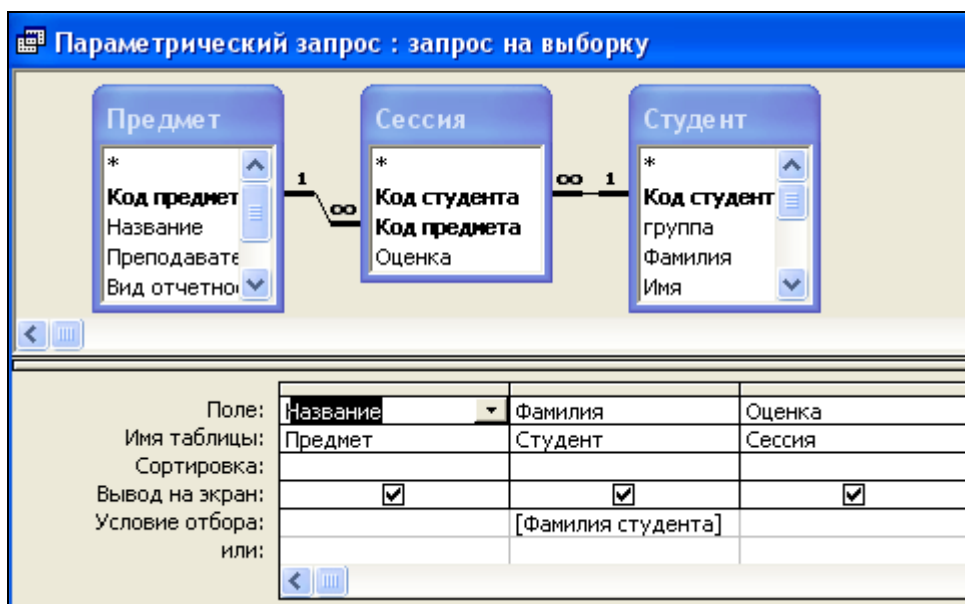


Рис. 16. Параметрический запрос

При выполнении запроса это имя появится в диалоговом окне «Введите значение параметра» (рис. 17).

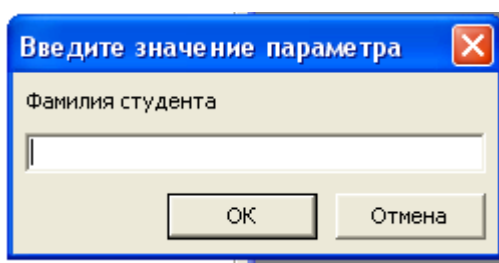


Рис. 17. Диалоговое окно «Введите значение параметра»

Временная таблица запроса будет доступна после ввода конкретной фамилии. В таблице будут отражены только те записи, которые относятся к введенному значению параметра (рис.18).

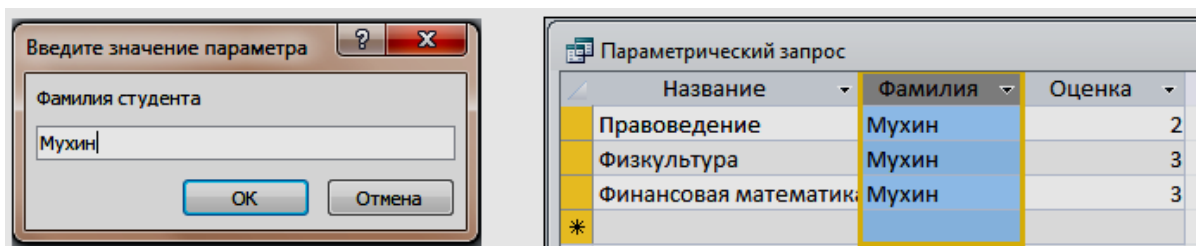


Рис.18. Временная таблица параметрического запроса

Параметры запроса могут быть использованы не только для отбора записей по значению поля, но и для ввода значения операнда в выражениях условий отбора или вычисляемых полей. Рассмотрим запрос для отбора студентов, родившихся ранее задаваемого года. Условие отбора с параметром запроса может быть задано как на рис. 19.

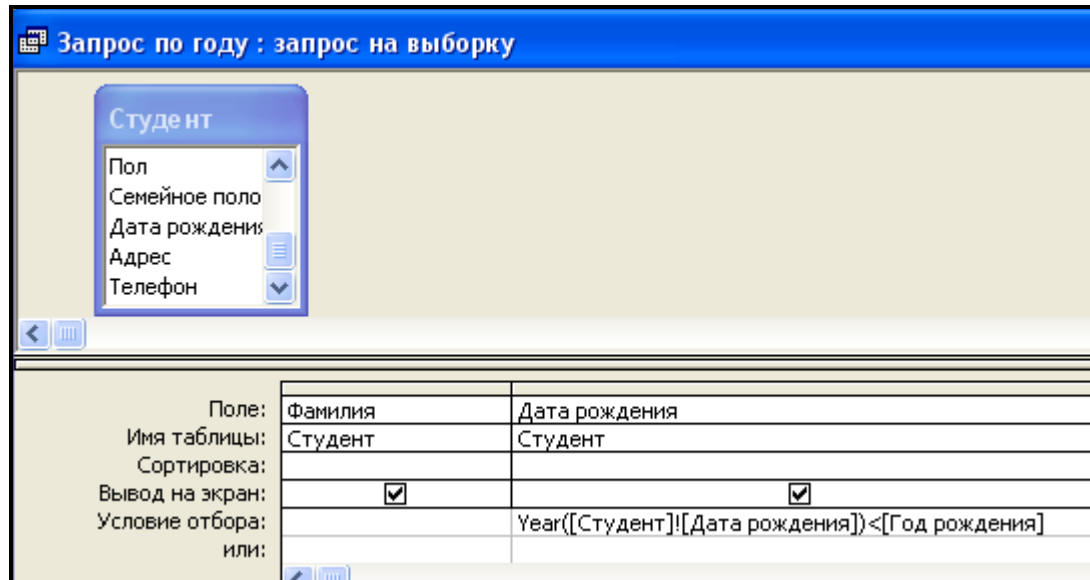


Рис. 19. Параметрический запрос отбора студентов

Здесь в условии отбора используется функция Year, которая выделяет год из даты рождения. Параметр запроса – [Год рождения] – является величиной, с которой сравнивается значение данной функции. Выражение в *Условии отбора* строится с помощью *Построителя выражений*. Если после выполнения запроса, попробовать открыть его в режиме конструктора, то можно увидеть, что Access произвел в нем небольшие изменения (рис. 20).

Сравнив рисунки 19 и 20 нетрудно понять, что Access автоматически создал поле, не отображаемое на экране, условие отбора в котором задается пользователем с помощью ввода значения параметра года рождения.

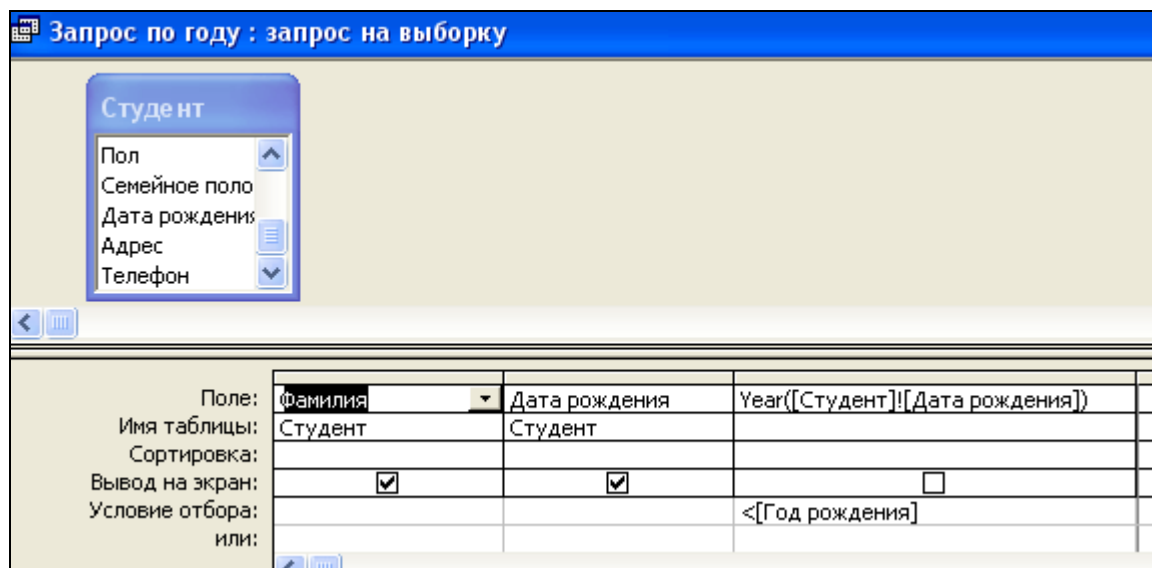



Рис. 20. Конструктор запроса после его выполнения

2.1.8. Создание перекрестного запроса с помощью мастера

В *перекрестном запросе* мастер формирует таблицу, в которой левый столбец образует заголовки строк из значений одного поля, верхняя строка образует заголовки столбцов из значений другого поля, а на пересечении строк и столбцов размещаются итоговые значения, вычисленные по значениям третьего поля. Для получения итогового значения записи группируются по полям, используемым в качестве заголовков строк и столбцов, а для значений третьего поля в полученных группах записей применяется одна из выбранных статистических функций.

Для создания перекрестного запроса с помощью мастера необходимо выбрать объект БД – запросы и нажать кнопку Создать – выбрать пункт Перекрестный запрос и следовать по шагам мастера.

Например, мы хотим получить сравнительную таблицу по группам: сколько в каждой группе учится студентов мужского и женского пола. Для этого вызовем мастер перекрестных запросов и на первом шаге в верхнем окошке выберем таблицу Студент (нажмем Далее). На следующем шаге поле *Группа* стрелочкой перенести в раздел Выбранные поля и нажать Далее – значения этого поля будут использованы в качестве заголовков

строк. На следующем шаге в качестве заголовков столбцов выберите значения поля Пол и нажмите далее. На следующем шаге выберите функцию Число – для подсчета количества студентов по полам. Это аналог Count. Затем задайте имя запроса (или согласитесь с предложенным) и просмотрите результат. Закройте запрос и обратите внимание, что значок у этого запроса отличается от значка запросов на выборку  .

2.1.9. Создание перекрестного запроса с помощью конструктора

Первоначально создается запрос на выборку, затем его тип меняется на перекрестный.

Допустим, мы хотим подсчитать сколько оценок каждого предмета получено за сессию по каждому предмету. Для этого в бланк запроса добавим таблицы Сессия и Предмет (рис. 21). Перенесем в бланк запроса поля Предмет и Оценка. Выполним команду Запрос/Перекрестный. Бланк запроса изменится. Автоматически будут включены групповые операции, и добавится строка Перекрестная таблица. В строке Перекрестная таблица напротив поля Оценка в раскрывающемся списке укажем *Заголовки строк*, напротив поля Предмет – *Заголовки столбцов*. Затем добавим еще раз поле Оценка, в строке Перекрестная таблица укажем *Значение*, в строке Групповые операции – Count. Вернемся в режим конструктора и добавим вычисляемое поле для подсчета итогов.

2.1.10. Корректировка данных средствами запроса

Корректировка данных возможна с помощью создания запросов-изменений. Строятся запросы-изменения так же, как и другие рассмотренные ранее запросы, на основе запроса на выборку.

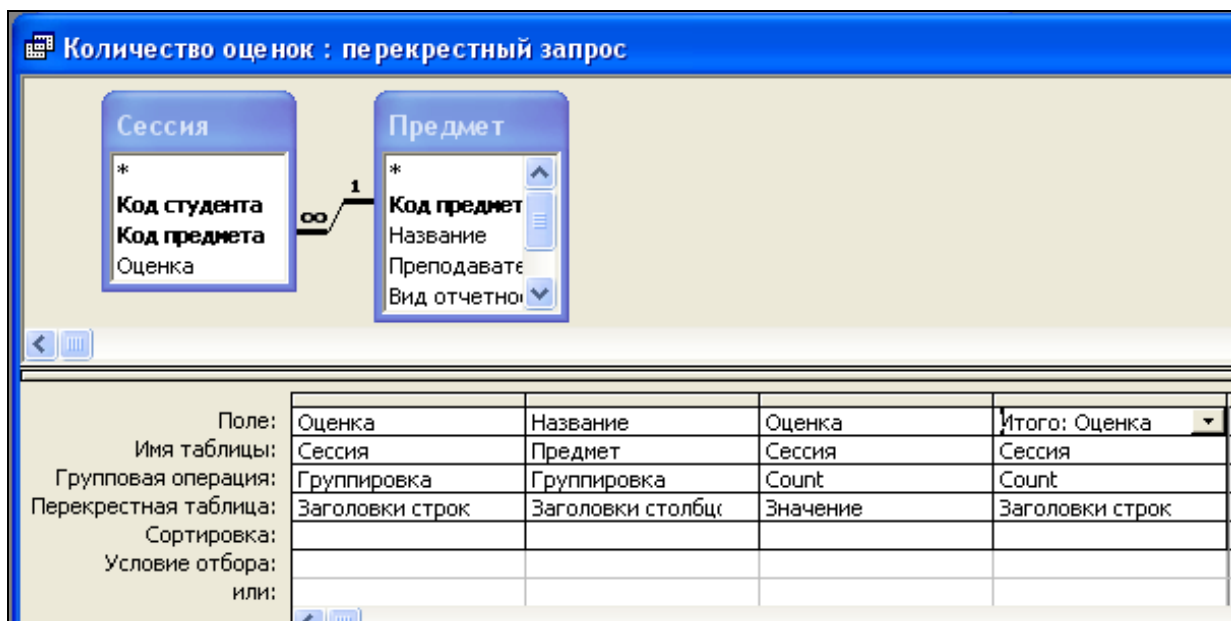


Рис. 21. Конструктор перекрестного запроса

К запросам-изменениям относятся:

- запрос на создание таблицы;
- запрос на удаление данных из таблицы;
- запрос на обновление записей;
- запрос на добавление записей.

Запрос на создание таблицы используется для сохранения результата запроса не во временной, а в реальной таблице базы данных. Допустим, нам необходимо выяснить, кто из студентов подлежит отчислению по итогам сессии. Это те студенты, у которых 2 и более двоек на экзаменах (соответствующие записи должны быть сделаны в таблице «Сессия»). Полученные данные должны быть помещены в отдельную таблицу базы данных.

Данный запрос мы будем создавать на основе запроса на выборку, основанного на другом запросе на выборку. Первоначально необходимо создать вспомогательный запрос на выборку, назвать его «Для создания таблицы». При создании запроса в окне конструктора добавить таблицы «Студент», «Предмет» и «Сессия» (рис. 22). В бланк запроса добавить поля «Фамилия», «Оценка» и «Вид отчетности» из соответствующих

таблиц. Затем задать в поле «Оценка» условие отбора – «2». Во временной таблице этого запроса будут отражаться те записи, где отражено получение студентами двоек, причем предмет не имеет значения. Вернуться в окно конструктора и добавить условие отбора в поле «Вид отчетности» - «экзамен». Во временной таблице запроса будут отражены только двойки, полученные на экзаменах. Сохранить запрос и закрыть его.

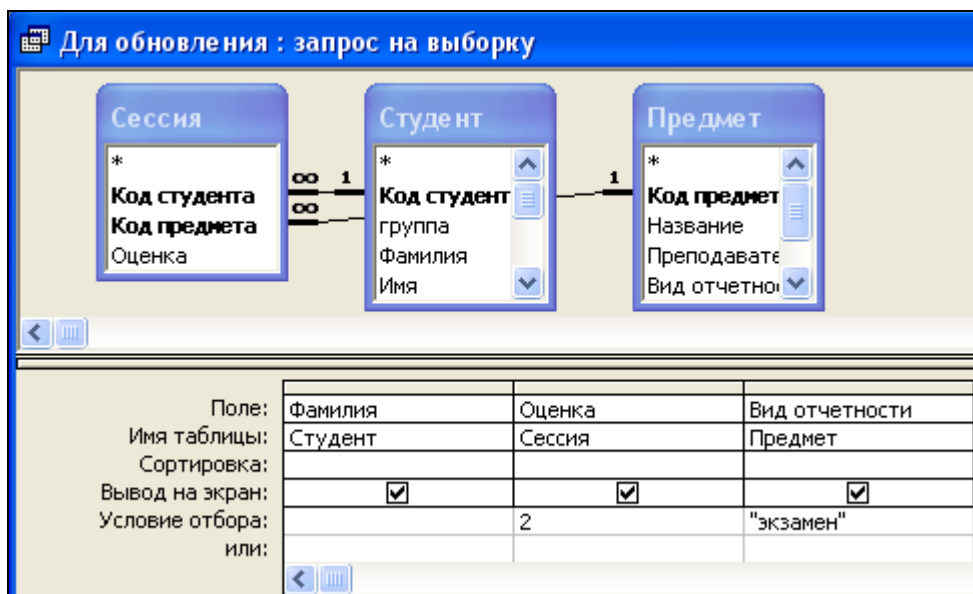



Рис. 22. Конструктор вспомогательного запроса на выборку

Начать создание запроса, который позже будет преобразован в запрос на создание таблицы. В окне конструктора в диалоговом окне «Добавление таблицы» перейти на вкладку «Запросы» и добавить только что созданный запрос «Для создания таблицы» (рис. 23).

Перенести в бланк запроса поля «Оценка» и еще раз «Оценка». Включить групповые операции. Во втором поле «Оценка» задать групповую операцию Count и условие отбора – «>1». Добавить в бланк третье поле – «Фамилия». Просмотреть результаты запроса в его временной таблице. Вернуться в конструктор запроса и в поле Тип запроса произвести преобразование из запроса на выборку в запрос на создание таблицы . В окне «Создание таблицы» нужно ввести ее имя

– «Отчисленные студенты», оставить остальные параметры окна без изменений и нажать ОК.

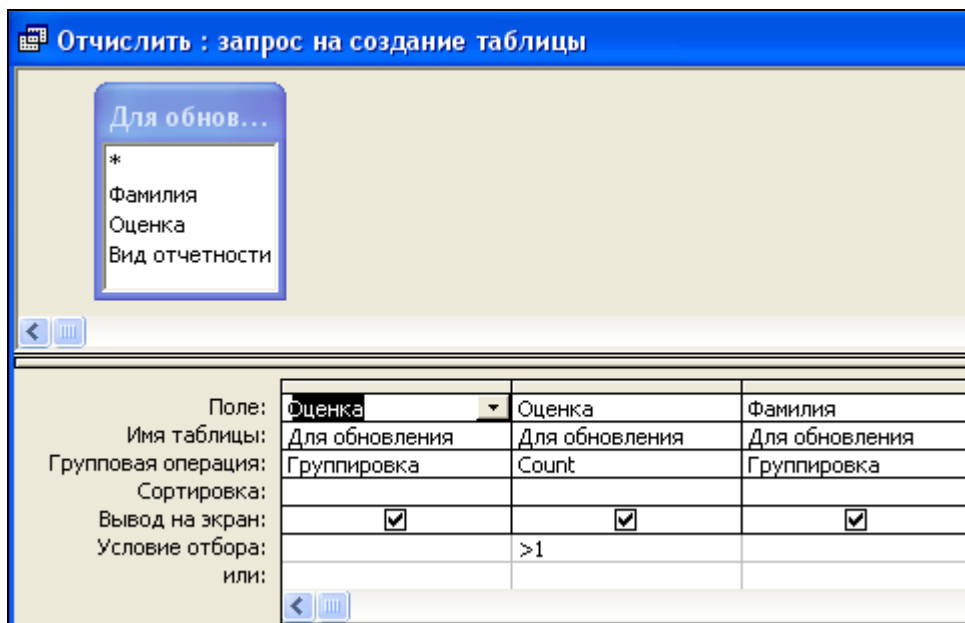



Рис. 23. Конструктор запроса на создание таблицы

Сохранить запрос и запустить его. Это можно сделать двумя способами:

1) из окна конструктора запроса, нажав на кнопку запуска на панели инструментов -  (красный восклицательный знак);

2) в окне базы данных на вкладке «Запросы», дважды щелкнув на имени сохраненного запроса на создание таблицы.

При любом из способов будет один и тот же результат: Access предупредит о том, что он создает таблицу и помещает в нее определенное число записей. Нужно ответить согласием на его действия и перейти на вкладку «Таблицы» базы данных. Если все прошло успешно, то вместо трех таблиц будет четыре, новая будет называться «Отчисленные студенты». Просмотрите ее записи и убедитесь, что они такие же, как во временной таблице запроса на выборку, который был переделан в запрос-изменение.


Запрос на удаление позволяет удалить записи из одной или нескольких взаимосвязанных таблиц. В запросе должны быть указаны

таблицы, из которых производится удаление и критерий отбора записей, подлежащих удалению.

Первоначально создается *запрос на выборку*, который в окне конструктора запросов преобразуется в *запрос на удаление* путем выбора пункта **Удаление** из списка **Тип запроса** или команды меню **Запрос/Удаление**.

После преобразования запроса в запрос на удаление в его бланке появляется строка «Удаление». Затем формируется бланк запроса. Для задания таблицы, записи из которой надо удалить, с помощью мыши перенести символ (*) из списка ее полей в бланк запроса. При этом в строке «Удаление» появится значение «Из». Чтобы задать условие отбора удаляемых записей, надо переместить с помощью мыши в бланк запроса поле (поля), для которого ставится условие отбора. При этом в строке «Удаление» в этих полях появится значение «Условие». Строку «Условие отбора» для этих полей необходимо заполнить критериями отбора.

Для предварительного просмотра удаляемых записей, необходимо нажать кнопку «Вид». Для возврата в режим конструктора используется та же кнопка.

После того, как удаляемые записи просмотрены, можно запустить запрос кнопкой «Запуск» на панели инструментов .

Например, нам нужно удалить из списка студентов всех отчисленных студентов. Значит в таблице «Студент» мы должны удалить только те записи, которые относятся к студентам-двоечникам.

Основываясь на созданной таблице «Отчисленные студенты», определите для себя, какие записи необходимо удалить. Например, по итогам сессии двое студентов – Журавлева и Симонов – подлежат отчислению. Начинаем создание запроса на удаление с того, что создаем запрос на выборку, в бланк которого добавляем таблицу «Студент». Переносим (*) все ее поля в бланк запроса (рис. 24). Командой

Запрос/Удаление преобразуем запрос на выборку в запрос на удаление. Перенесем двойным щелчком в бланк запроса поле «Фамилия». В строке «Условие отбора» задаем соответствующие фамилии, то есть пишем условие «Журавлева» От «Симонов». Просмотрим с помощью кнопки «Вид», что отображены именно те записи, которые нужно.

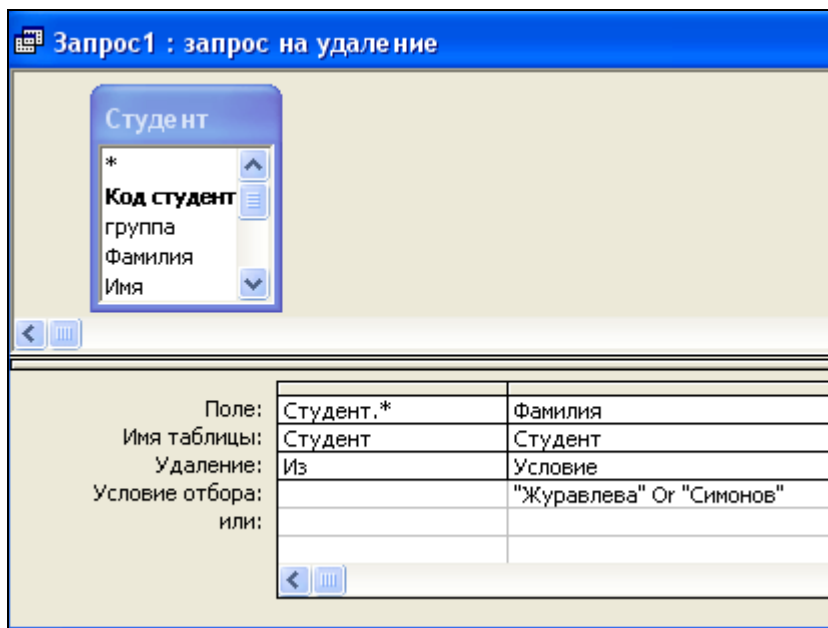


Рис. 24. Конструктор запроса на удаление

Если все правильно, то можно запускать запрос: вернуться в режим конструктора и нажать кнопку «Запуск». Появится диалоговое окно предупреждения (рис. 25).

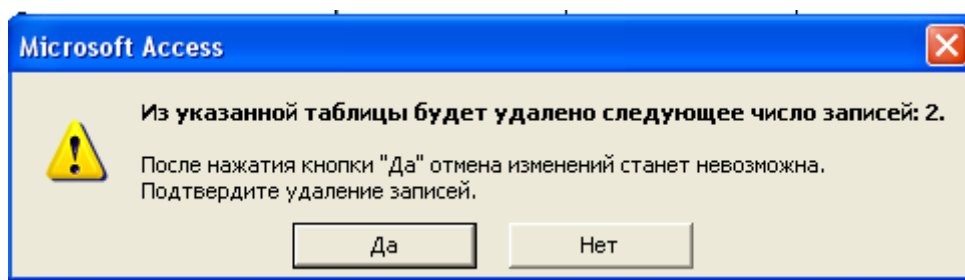


Рис. 25. Диалоговое окно «Предупреждение об удалении записей»

Нажать кнопку «Да». Закрыть запрос, предварительно сохранив его под именем «Отчисление студентов». В окне базы данных перейдите на вкладку «Таблица», откройте таблицу «Студент» и убедитесь, что записи о данных студентах удалены.

Запрос на обновление. Для обновления данных в полях базовых таблиц может быть использован *Запрос на обновление*. Изменения вносятся в группу записей, отбираемых с помощью указанных пользователем условий отбора. Значения для изменений в полях определяются в бланке запроса в строке **Обновление**.

Первоначально создается *запрос на выборку*, который в окне конструктора запросов преобразуется в *запрос на обновление* путем выбора пункта **Обновление** из списка **Тип запроса** или команды меню **Запрос/Обновление**. После выполнения этой команды в бланке запроса появится строка **Обновление**.

Для отбора обновляемых записей нужно включить в бланк запроса поля, требующие обновления, а также поля, по которым задаются условия отбора записей. Условия отбора задаются аналогично тому, как это делается при создании запросов на выборку.

Для обновляемого поля в строку **Обновление** надо ввести значение или выражение, определяющее новое значение поля. Выражение можно создать с помощью построителя выражений.

После выполнения команды **Запрос/Запуск** или нажатия соответствующей кнопки панели инструментов открывается диалоговое окно с сообщением о числе обновляемых записей и вопросом о продолжении операции обновления.

Переключившись в режим таблицы после выполнения запроса, можно просмотреть содержание только что обновленных полей.

Запрос на добавление предназначен для добавления результата запроса данных из таблицы в другую таблицу БД. Поэтому надо, чтобы в запросе были сформированы записи с полями, соответствующими полям в дополняемой записями таблице БД.

2.2. База данных «Учебный процесс»

Задание 1. Создайте новую БД «Учебный процесс» и сконструируйте таблицы с информацией о студентах и студенческих группах экономического факультета.

Порядок работы:

- вызовите программу MS Access с помощью меню или ярлыка на Рабочем столе;

- выберите пункт Создать/Новая база данных. Выполняя предварительное сохранение, в поле Имя файла в качестве имени базы данных введите <Фамилия>_Учебный процесс, где вместо слова «фамилия» впишите свою фамилию. Это будет название вашей личной базы данных. Перед нажатием кнопки Создать не забудьте открыть свою папку;

- в меню выберите пункт **Создание** и выполните команду **Конструктор таблиц**;

- заполните поля в Конструкторе данными из табл. 2.

Т а б л и ц а 2

Описание свойств полей таблицы ГРУППА

| Имя поля | Обязательное поле | Тип данных | Размер | Число десятичных знаков | Подпись поля |
|----------|-------------------|------------|--------------------------|-------------------------|----------------------|
| НГ | Да | Текстовый | 3 | | Номер группы |
| КОЛ | Нет | Числовой | Байт | | Количество студентов |
| ПБАЛЛ | Нет | Числовой | Одинарное с плав. точкой | 2 | Проходной балл |

- сделайте поле НГ ключевым;
- сохраните таблицу, дав ей имя ГРУППА;

- перейдите в режим таблицы, используя панель инструментов;
- заполните таблицу данными из таблицы 3;

Т а б л и ц а 3

Данные таблицы ГРУППА

| Номер группы | Количество студентов | Проходной балл |
|--------------|----------------------|----------------|
| 101 | 30 | 4,50 |
| 102 | 32 | 4,50 |
| 103 | 29 | 4,80 |
| 104 | 35 | 4,40 |
| 105 | 35 | 4,80 |
| 201 | 35 | 4,00 |

- сохраните полученную таблицу;
- закройте таблицу;
- выполните команду Создание/Конструктор таблиц и заполните поля в Конструкторе данными из таблицы 4;
- сохраните таблицу, дав ей имя СТУДЕНТ;
- для облегчения заполнения таблицы выполните подстановку в поле НГ. Для этого в столбце Тип данных в раскрывающемся списке выберите Мастер подстановок, затем нажмите кнопку Далее, с помощью единичной стрелки выберите поле НГ и нажмите Далее, снова Далее и Готово;
- в данной таблице в составной первичный (уникальный) ключ входят поля НГ и НС. Для определения этого ключа в режиме Конструктора надо выделить оба этих поля, щелкая кнопкой мышки на области маркировки при нажатой клавише <Ctrl>. Затем нажать кнопку панели инструментов Ключевое поле;

Описание свойств полей таблицы СТУДЕНТ

| Имя поля | Обязательное поле | Тип данных | Размер | Число десятичных знаков | Подпись поля |
|----------|-------------------|------------|--------------------------|-------------------------|-------------------------|
| НГ | Да | Текстовый | 3 | | Группа |
| НС | Да | Текстовый | 2 | | Номер студента в группе |
| ФИО | Да | Текстовый | 25 | | ФИО |
| ГОДР | Нет | Числовой | Целое | | Год рождения |
| АДРЕС | Нет | Текстовый | 25 | | Адрес |
| ПБАЛЛ | Нет | Числовой | Одинарное с плав. точкой | 2 | Проходной балл |

- перейдите в режим таблицы, используя панель инструментов;
- заполните таблицу данными из табл. 5.

Данные таблицы СТУДЕНТ

| Группа | Номер студента в группе | ФИО | Год рождения | Проходной балл |
|--------|-------------------------|-----------------|--------------|----------------|
| 101 | 01 | Аристов Р.П. | 1982 | 4,25 |
| 101 | 02 | Бондаренко С.А. | 1981 | 4,50 |
| 101 | 03 | Борисова Е.И. | 1982 | 4,25 |
| 101 | 04 | Макарова Н.В. | 1980 | 4,75 |
| 102 | 01 | Бакунина Н.Д. | 1982 | 4,50 |
| 102 | 02 | Фадеев Д.К. | 1982 | 4,25 |
| 102 | 03 | Петров М.Н. | 1982 | 4,50 |
| 103 | 01 | Афанасьев Г.М. | 1981 | 4,25 |
| 103 | 02 | Пирогова С.А. | 1982 | 4,75 |
| 104 | 01 | Иванов К.К. | 1981 | 4,50 |

Задание 2. Выполните редактирование ячеек созданной таблицы СТУДЕНТ.

Порядок работы:

- откройте таблицу в режиме таблицы;
- замените фамилию Бондаренко на Бондарчук, для чего выделите редактируемую ячейку и наберите новую фамилию;
- у студентки Макаровой Н.В. замените год рождения на 1981.

Задание 3. Сортировка и фильтрация записей таблицы СТУДЕНТ.

Порядок работы:

- отсортируйте фамилии студентов по возрастанию, для чего поставьте курсор на любую ячейку столбца ФИО и нажмите соответствующую кнопку на панели инструментов;
- отсортируйте Год рождения по убыванию;
- отсортируйте Номер группы по возрастанию;
- покажите только записи, относящиеся к 101 группе. Для этого выделите в столбце Номер группы ячейку, содержащую номер 101 и нажмите на панели инструментов кнопку Фильтр по выделенному – будут показаны только нужные записи, а остальные временно скрыты;
- усложните фильтр, нажав кнопку Изменить фильтр и задав условие на Год рождения – 1982. Чтобы увидеть записи, соответствующие заданным условиям, нажмите кнопку Применение фильтра;
- просмотрев появившиеся записи, удалите фильтр, нажав на кнопку Удалить фильтр;
- закройте таблицу.

Задание 4. Работа со схемой данных.

Порядок работы:

- щелкните по кнопке Схема данных;
- выделите правой кнопкой мышки связь между таблицами ГРУППА и СТУДЕНТ. Выберите пункт «Изменить связь»;

– включите флажки **Обеспечение целостности данных, Каскадное обновление связанных полей и Каскадное удаление связанных полей**;

– нажмите ОК, сохраните изменения и закройте схему данных.

Задание 5. Создание формы для ввода данных в таблицу СТУДЕНТ.

Порядок работы:

– в окне БД выберите вкладку **Формы**;

– щелкните по кнопке **Создать**;

– в диалоговом окне **Новая форма** выберите в верхней части **Мастер форм**, а в нижней с помощью стрелки раскрывающегося меню – источник данных, т.е. таблицу СТУДЕНТ. Нажмите ОК;

– с помощью двойной стрелки переместите все поля из отдела **Доступные поля** в отдел **Выбранные поля**. Нажмите кнопку **Далее**;

– выберите один из трех видов будущей формы (в один столбец, ленточный, выровненный) и нажмите кнопку **Далее**;

– выберите стиль будущей формы (по личному усмотрению) и нажмите **Далее**;

– мастер предложит дать новой форме имя СТУДЕНТ и открыть ее для заполнения. Согласитесь, нажав кнопку **Готово**. В результате получите форму, в которой можно менять существующие данные и вводить новые.

Задание 6. Внесение изменений в таблицу СТУДЕНТ с помощью формы СТУДЕНТ.

Порядок работы:

– открыть форму СТУДЕНТ;

– удалите запись о студентке Пироговой С.А. Для этого найдите запись о ней, выполните команду **Правка/Выделить запись**, а затем команду **Правка/Удалить** (или нажмите клавишу Delete на клавиатуре).

Будьте внимательны, это действие нельзя отменить;

- внесите записи о новых студентах (табл. 6);

Т а б л и ц а 6

Новые данные таблицы СТУДЕНТ

| Группа | Номер студента в группе | ФИО | Год рождения | Проходной балл |
|--------|----------------------------|-----------------|-----------------|----------------|
| 103 | 02 | Киселева Т.А. | 1982 | 4,75 |
| 103 | 03 | Никитина И.О. | 1982 | 4,75 |
| 104 | 02 | Григорьева Т.П. | 1981 | 4,60 |

- закройте форму СТУДЕНТ, перейдите в окне БД на вкладку Таблицы, откройте таблицу СТУДЕНТ и убедитесь, что все внесенные изменения присутствуют в ней.

Задание 7. Создание новой таблицы КАФЕДРА.

Порядок работы:

- в окне БД выберите команду Создание/Конструктор таблицы;
- заполните поля в Конструкторе данными из табл. 7.

Т а б л и ц а 7

Описание свойств полей таблицы КАФЕДРА

| Имя поля | Обязательное поле | Тип данных | Размер | Подпись поля |
|----------|-------------------|------------------|--------|------------------------|
| ККАФ | Да | Текстовый | 2 | Код кафедры |
| НКАФ | Нет | Текстовый | 30 | Название |
| ТЕЛ | Нет | Текстовый | 15 | Телефон |
| ЗАВ | Нет | Текстовый | 25 | ФИО зав.каф. |
| ФОТО | Нет | Поле объекта OLE | | Фотография заведующего |

- сделайте поле ККАФ ключевым;
- сохраните таблицу под именем КАФЕДРА и закройте окно конструктора.

Задание 8. Заполнение таблицы КАФЕДРА с помощью формы КАФЕДРА.

Порядок работы:

- создайте форму КАФЕДРА так же, как и форму СТУДЕНТ (см. задание 6). При создании формы выберите вид формы «в один столбец»;
- заполните данными из табл. 8;

Т а б л и ц а 8

Данные таблицы КАФЕДРА

| Код | Название | Телефон | ФИО зав.каф. |
|-----|--------------------|----------|-----------------|
| 01 | Информатики | 30-52-50 | Игнатъев В.В. |
| 02 | Иностранного языка | 30-67-90 | Кораблев П.И. |
| 03 | Экономики | 33-87-09 | Жданова А.Е. |
| 04 | Истории | 33-71-22 | Смирнова И.В. |
| 05 | Высшей математики | 30-78-88 | Ломоносов М.А. |
| 06 | Философии | 30-00-22 | Бондаренко В.В. |

– для создания «фото» заведующих кафедрами необходимо щелкнуть по пустому окну поля «Фотография заведующего» правой кнопкой мыши и выбрать в контекстном меню пункт Добавить объект, указать тип объекта – рисунок Paintbrush и нажать ОК;

– с помощью средств Paintbrush нарисовать «портрет», затем щелкнуть левой кнопкой мышки в произвольном месте вне рисунка – «портрет» будет вставлен;

– после заполнения всех полей закройте форму КАФЕДРА и перейдите на вкладку Таблицы;

– откройте таблицу КАФЕДРА и убедитесь в наличии в ней данных. Обратите внимание, что в поле «Фотография заведующего» вместо рисунка присутствует надпись «Точечный рисунок».

Задание 9. Создание новой таблицы ПРЕПОДАВАТЕЛЬ.

Порядок работы:

– в окне БД выберите вкладку Таблицы и выполните команду Создание/Конструктор таблиц;

– заполните поля в Конструкторе данными из табл. 9;

Т а б л и ц а 9

Описание свойств полей таблицы ПРЕПОДАВАТЕЛЬ

| Имя поля | Обязательное поле | Тип данных | Размер | Подпись поля |
|----------|-------------------|------------|--------|-------------------|
| ТАБН | Да | Текстовый | 4 | Табельный номер |
| ФИО | Да | Текстовый | 35 | ФИО преподавателя |
| СТ | Нет | Текстовый | 15 | Уч. Степень |
| ЗВ | Нет | Текстовый | 10 | Уч. Звание |
| ККАФ | Да | Текстовый | 2 | Код кафедры |

– сделайте поле ТАБН ключевым;

– сохраните таблицу под именем ПРЕПОДАВАТЕЛЬ;

– в поле ККАФ выберите тип данных «Мастер подстановок» и выполните последовательность шагов мастера, связав данное поле с одноименным полем в таблице КАФЕДРА;

– перейдите на вкладку Формы и создайте форму ПРЕПОДАВАТЕЛЬ. Заполните с ее помощью таблицу ПРЕПОДАВАТЕЛЬ данными, приведенными в табл. 10;

– убедитесь, что в схеме данных установлены связи между таблицами ПРЕПОДАВАТЕЛЬ и КАФЕДРА. Прodelайте для данных таблиц действия, аналогичные указанным в задании 4.

Данные таблицы ПРЕПОДАВАТЕЛЬ

| Таб.номер | ФИО преподавателя | Уч. Степень | Уч. звание | Код кафедры |
|-----------|--------------------|-------------|------------|-------------|
| 101 | Игнатьев В.В. | д. техн. н. | профессор | 01 |
| 102 | Антонов К.Е. | к. техн. н. | доцент | 01 |
| 103 | Глухова Ж.А. | Нет | ст. преп. | 01 |
| 201 | Кораблев П.И. | д. фил. н. | профессор | 02 |
| 202 | Чернова Н.А. | к. фил. н. | доцент | 02 |
| 301 | Жданова А.Е. | д. пед. н. | профессор | 03 |
| 302 | Новиков П.Н. | | ст. преп. | 03 |
| 401 | Смирнова И.В. | к. ист. н. | доцент | 04 |
| 402 | Романова А.В. | | ассистент | 04 |
| 501 | Ломоносов М.А. | д. ф.-м. н. | профессор | 05 |
| 502 | Львова П.Я. | к. ф.-м. н. | доцент | 05 |
| 503 | Глухов О.Н. | | ст. преп. | 05 |
| 601 | Бондаренко В.В. | к. ф. н. | доцент | 06 |
| 602 | Кудряшова Т.П. | | ассистент | 06 |

Задание 10. Создание новой таблицы ПРЕДМЕТ.

Порядок работы:

- в окне БД выполните команду **Создание/Конструктор таблиц**;
- заполните поля в Конструкторе данными из табл. 11;
- сделайте поле КП ключевым;
- сохраните таблицу под именем ПРЕДМЕТ;

Описание свойств полей таблицы ПРЕДМЕТ

| Имя поля | Обязательное поле | Тип данных | Размер | Подпись поля | Условие на значение | Сообщение об ошибке |
|----------|-------------------|------------|--------|----------------------|---------------------|--------------------------------------|
| КП | Да | Текстовый | 4 | Код предмета | | |
| НП | Нет | Текстовый | 35 | Название предмета | | |
| ЧАСЫ | Нет | Числовой | Целое | Всего часов | >0 And < 201 | Число часов должно быть не более 200 |
| ЛК | Нет | Числовой | Целое | Лекции | | |
| ПЗ | Нет | Числовой | Целое | Практические занятия | | |

- перейдите на вкладку Формы и создайте форму ПРЕДМЕТ;
- заполните с ее помощью таблицу ПРЕДМЕТ данными, приведенными в табл. 12;

Данные таблицы ПРЕДМЕТ

| Код предмета | Название предмета | Всего часов | Лекции | Практические занятия |
|--------------|-------------------|-------------|--------|----------------------|
| 11 | Информатика | 48 | 30 | 18 |
| 12 | Иностранный язык | 60 | 0 | 60 |
| 13 | Экономика | 50 | 20 | 30 |
| 14 | История | 38 | 20 | 18 |
| 15 | Математика | 50 | 14 | 36 |
| 16 | Философия | 40 | 18 | 22 |

– перейти в режим конструктора таблиц и добавить поле ОТЧ (текстовое, разм.-20, назв. -Вид отчетности). Заполнить его данными для предметов: 11 – зачет, 12 – экзамен, 13 – зачет, 14 – зачет, 15 – экзамен, 16 – экзамен.

Задание 11. Создание новой таблицы УСПЕВАЕМОСТЬ.

Порядок работы:

- в окне БД выполните команду Создание/Конструктор таблиц;
- заполните поля в Конструкторе данными из табл. 13;

Т а б л и ц а 13

Описание свойств полей таблицы УСПЕВАЕМОСТЬ

| Имя поля | Обязательное поле | Тип данных | Размер | Подпись поля |
|----------|-------------------|------------|--------|-----------------|
| НГ | Да | Текстовый | 3 | Номер группы |
| НС | Да | Текстовый | 2 | Номер студента |
| КП | Да | Текстовый | 4 | Код предмета |
| ТАБН | Да | Текстовый | 4 | Табельный номер |
| ОЦЕНКА | Нет | Числовой | Целое | |

- назначьте составной ключ из полей НГ, НС, КП, ТАБН;
- сохраните таблицу под именем УСПЕВАЕМОСТЬ;
- в полях НГ, НС, КП, ТАБН выполните подстановку значений из одноименных полей таблиц соответственно СТУДЕНТ, ПРЕДМЕТ, ПРЕПОДАВАТЕЛЬ;

– закройте таблицу и перейдите в схему данных базы. Установите обозначившиеся в схеме данных связи вида 1:М, обеспечив целостность данных, каскадное обновление и удаление записей. Вид схемы данных, которая должна после этого получиться, представлен на рис. 26;

– перейдите на вкладку Формы и создайте форму УСПЕВАЕМОСТЬ. Заполните с ее помощью данными таблицу УСПЕВАЕМОСТЬ следующим образом: студенты 101 и 104 групп сдают

предметы 11 – 13 преподавателям соответственно 102 и 103, 201 и 202, 301 и 302; студенты 102 и 103 групп сдают предметы 14 – 16 преподавателям соответственно 401 и 402, 501 и 502, 601 и 602. Оценки на экзаменах и зачетах придумать самостоятельно, предусмотрев при этом в каждом потоке наличие двоечников (имеющих 1 или 2 двойки), отличников и студентов, сдающих без троек.

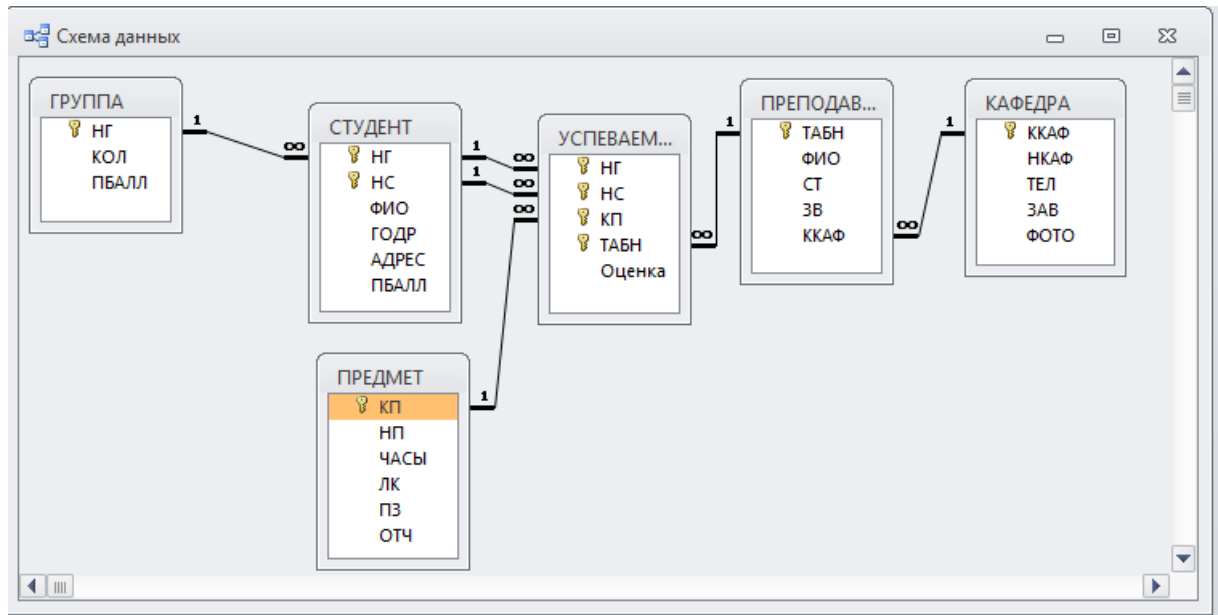


Рис. 26. Схема данных

Задание 12. Создание простейших запросов на выборку.

Порядок работы:

- выполните команду **Создание/Конструктор запросов**;
- добавьте в верхнее поле конструктора таблицу СТУДЕНТ и закройте окно **Добавление таблицы**;
- перенесите с помощью мышки поля НГ, НС, ФИО, ГОДР из макета таблицы в верхнем поле **Конструктора** в нижнее поле конструктора. С помощью кнопки **Вид (Режим таблицы)** на панели инструментов просмотрите временную таблицу запроса;
- вернитесь в режим **Конструктора** и добавьте в поле ГОДР в строке **Условие отбора** условие 1982. С помощью кнопки **Вид** просмотрите полученную выборку. Вернитесь в режим Конструктора и

добавьте в поле НГ условие «102» Or «103». Просмотрите полученную выборку. Вернитесь в Конструктор и снимите флажок вывода на экран в поле ГОДР. Просмотрите. Сохраните запрос под именем 1982;

- выполните команду **Создание/Конструктор запросов**;
- добавьте в верхнее поле конструктора таблицу СТУДЕНТ и закройте окно **Добавление таблицы**;

- перенесите с помощью мышки поля НГ, НС, ФИО, ГОДР из макета таблицы в верхнем поле **Конструктора** в нижнее поле конструктора;

- в поле ФИО введите условие для того, чтобы выборка отображала студентов, чья фамилия начинается на букву А (Like «А*»). Просмотрите. Добавьте условие для отображения студентов с фамилиями на А или на Б (Like «А*» Or Like «Б»). Просмотрите выборку. Сохраните запрос под именем «Студенты на А и Б».

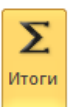
Задание 13. Создание запроса на выборку с использованием групповых операций.

Порядок работы:

- выполните команду **Создание/Конструктор запросов**;
- добавьте в верхнее поле конструктора таблицы УСПЕВАЕМОСТЬ и ПРЕПОДАВАТЕЛЬ;

- перенесите в таблицу конструктора поля НГ из таблицы УСПЕВАЕМОСТЬ и ФИО из таблицы ПРЕПОДАВАТЕЛЬ;

- просмотрите полученную временную таблицу запроса (она имеет неудобный вид);

- вернитесь в конструктор запроса и нажмите на панели инструментов кнопку **Итоги** . Снова просмотрите временную таблицу запроса. Убедитесь в произошедших изменениях;

- сохраните запрос под именем ПРЕПОДАВАТЕЛИ;

- вернитесь в конструктор запроса ПРЕПОДАВАТЕЛИ;
- добавьте в схему данных запроса таблицу ПРЕДМЕТ;
- добавьте в нижнее поле конструктора следующие поля этой таблицы: НП, ОТЧ. Просмотрите полученную таблицу запроса. Получился список преподавателей, принимающих зачеты и экзамены у каждой учебной группы по каждому предмету.

Задание 14. Создание запроса СЕССИЯ.

Порядок работы:

- выполните команду **Создание/Конструктор запросов**;
- необходимо создать запрос, таблица которого содержала бы следующие поля: ФИО студента, название сданного предмета, оценка. Самостоятельно выполните добавление необходимых таблиц в схему данных запроса и необходимых полей в бланк конструктора. Просмотрев результат, сохраните запрос под именем СЕССИЯ;

- вернитесь в конструктор и выполните необходимые действия для получения информации об успеваемости двух конкретных студентов: Бондаренко С.А. и Никитиной И.О. Просмотрите полученный результат. **Изменения по данному пункту не сохраняйте!** Закройте запрос без изменений;

- создайте запрос, используя те же таблицы и поля, что и в запросе СЕССИЯ. Для удобства просмотра записей по конкретному студенту используйте параметры. Для этого в условии отбора поля ФИО вместо конкретной фамилии введите название параметра, по которому будет запрашиваться фамилия при выполнении запроса. Название параметра введите как текст, заключенный в квадратные скобки: [Фамилия и инициалы студента];

- при попытке просмотра таблицы данного запроса появляется диалоговое окно «Введите значение параметра». Заполните его

диалоговую строку какой-либо фамилией с инициалами из списка студентов (например, Аристов Р.П.). Просмотрите полученный результат;

- сохраните запрос под именем СЕССИЯ-2.

Задание 15. Создание запроса для подсчета фактического количества студентов в группе.

Порядок работы:

- выполните команду **Создание/Конструктор таблиц**;
- добавьте в конструктор запроса таблицу СТУДЕНТ;
- в бланк запроса добавьте поле НГ 2 раза;
- активизируйте команду **Итоги**;
- во втором поле НГ выберите групповую операцию Count;
- просмотрите полученный результат;
- сохраните запрос под именем КОЛИЧЕСТВО СТУДЕНТОВ.


Задание 16. Создание запроса для подсчета количества отчисленных студентов в каждой группе (создание вычисляемого поля в запросе).

Порядок работы:

- выполните команду **Создание/Конструктор запросов**;
- добавьте в конструктор запроса таблицу ГРУППА и запрос КОЛИЧЕСТВО СТУДЕНТОВ;

- в бланк запроса добавьте поля НГ, КОЛ, Count НГ (просмотрите временную таблицу запроса и вернитесь в режим конструктора). Сохраните полученный запрос под именем ОТЧИСЛЕННЫЕ СТУДЕНТЫ;

- добавьте новое поле ОТЧИСЛЕНО (название наберите самостоятельно в следующем после Count НГ столбце и сразу же поставьте двоеточие);

- для построения выражения нажмите на кнопку «**Построитель**» (для вызова **Построителя выражений**) . Окно **Построителя**

выражений состоит из *верхнего и нижнего разделов*. В *верхнем* разделе отображается то выражение, которое строит пользователь. В *нижнем* разделе, который состоит из трех окон, находится необходимый «материал» для построения выражения: таблицы, запросы, формы, отчеты, встроенные и созданные пользователем функции, константы, операторы, выражения. Пользователь последовательно мышкой выбирает необходимые объекты и нажимает кнопку «**Вставить**». Когда выражение закончено, пользователь нажимает ОК;

– нам необходимо построить выражение, с помощью которого вычислялась бы разность между значениями полей КОЛ таблицы ГРУППА и Count НГ временной таблицы запроса КОЛИЧЕСТВО СТУДЕНТОВ. Для этого выбираем в левой части нижнего раздела Построителя выражений Таблицы/Группа, в средней части – поле КОЛ, нажимаем кнопку «Вставить». Нажимаем кнопку «минус» в разделе операторов. Затем аналогично вставляем поле Count НГ из запроса КОЛИЧЕСТВО СТУДЕНТОВ. Если все сделано правильно, то в верхнем разделе окна Построителя выражений появится запись: «Отчислено: [ГРУППА]![КОЛ]-[КОЛИЧЕСТВО СТУДЕНТОВ]![Count НГ]». Нажимаем ОК. (Пользователь, хорошо знакомый с синтаксисом выражений, может проделать эту операцию без помощи Построителя выражений в окне конструктора запроса);

– убедитесь в том, что для вновь созданного поля активен флажок вывода на экран и просмотрите временную таблицу запроса. Сохраните изменения, сделанные в запросе.

Задание 17. Создание запроса, являющегося основой для другого запроса или отчета.

Порядок работы:

- создайте запрос на выборку, в конструктор которого добавьте таблицы СТУДЕНТ, УСПЕВАЕМОСТЬ и ПРЕДМЕТ. Добавьте из них поля: НГ, ФИО, НП, ОТЧ, Оценка;

- просмотрите временную таблицу запроса и сохраните запрос под именем ДЛЯ ОТЧЕТА;

- закройте запрос.

Задание 18. Создание перекрестного запроса на основе ранее созданного запроса.

Порядок работы:

- выполните команду: Создание/Мастер запросов/Перекрестный запрос/ОК;

- в открывшемся диалоговом окне **Создание перекрестных таблиц** активизируйте флажок **Запросы** и выберите ранее созданный запрос ДЛЯ ОТЧЕТА. Нажмите кнопку Далее;

- выделите поле ФИО и нажмите стрелку, нажмите кнопку Далее;

- выделите поле НП и нажмите кнопку Далее;

- выделите поле Оценка, функцию «Сумма» и нажмите Далее;

- задайте имя запроса ПЕРЕКРЕСТНЫЙ и нажмите Готово;

- перейдите в конструктор этого запроса и удалите поле «Итоговое значение». Просмотрите таблицу полученного запроса, сохраните изменения и закройте запрос.

Задание 19. Создание отчета на основе ранее созданного запроса.

Порядок работы:

- выполните команду Создание/Мастер отчетов;
- в раскрывающемся списке «Таблицы и запросы» найдите запрос ДЛЯ ОТЧЕТА, перенесите в «Выбранные поля» все поля этого отчета и нажмите кнопку Далее;
- в окне «Выберите вид представления данных» должна быть выделена таблица СТУДЕНТ; нажмите Далее;
- добавьте уровни группировки (выделите поле НГ и нажмите стрелочку >); нажмите Далее;
- на следующей вкладке в первом раскрывающемся списке выберите название поля «Оценка», нажмите кнопку «Итоги» и выберите тип итогов Avg; нажмите ОК; нажмите Далее;
- выберите вид макета для отчета (по своему усмотрению); нажмите Далее;
- выберите один из возможных стилей оформления; нажмите Далее;
- задайте имя отчета ИТОГИ СЕССИИ; нажмите Готово;
- перейдите в конструктор отчета и внесите необходимые коррективы (вместо Avg – напишите «Средний балл», уберите сведения о количестве записей для каждого студента). Закройте отчет, сохранив изменения.

Задания для самостоятельной работы

Задание 1. Создание запроса.

Создайте запрос, отражающий сдаваемые во время сессии предметы с их кодами, общим количеством часов, типом отчетности и фамилиями преподавателей, принимающих зачет или экзамен для каждой группы.

Задание 2. Создание сложного запроса.

Создайте запрос с указанием фамилий и номеров групп студентов, получивших двойки. Создайте на его основе запрос, позволяющий увидеть количество двоек по каждому двоечнику.

Задание 3. Создание отчета.


Создайте отчет ВЕДОМОСТЬ ИНОСТРАННЫЙ ЯЗЫК на основе предварительно созданного запроса, позволяющий вывести сведения о сдаче экзамена по иностранному языку с указанием фамилий студентов, номеров групп, оценки и фамилии преподавателя, принимавшего экзамен. Создайте аналогичные отчеты по истории и математике.

2.3. База данных «Склад»

Задание 1. Создание таблиц базы данных.

Порядок работы:

1. Создайте новую базу данных, назовите ее по образцу: Склад - <Фамилия>, определив ее расположение в *своей* папке.

2. Перейдите в режим Конструктора таблицы (кнопка ) , присвойте таблице имя «Товары» и создайте структуру новой таблицы (табл. 14).

Т а б л и ц а 1 4

Состав полей таблицы «Товары»

| Имя поля | Тип данных | Свойства поля |
|---------------------------|------------|--|
| Код товара | Числовой | Ключевое |
| Наименование | Текстовый | Подпись: Товар |
| Поставщик | Текстовое | |
| Цена | Денежный | |
| Количество | Числовой | |
| Дата поступления на склад | Дата/время | Подпись: Дата склад, Маска ввода: Краткий формат даты |

3. Закройте конструктор таблицы.

4. Выполните команду: Создание/Конструктор таблиц, перейдите в режим Конструктора, присвойте новой таблице имя «Поставщики» и создайте ее структуру (табл. 15).

Т а б л и ц а 1 5

Состав полей таблицы «Поставщики»

| Имя поля | Тип данных | Свойства поля |
|---------------------|------------|-------------------------------|
| Код поставщика | Счетчик | Ключевое |
| Название поставщика | Текстовый | Подпись: Поставщик |
| Адрес | Текстовый | |
| Телефон | Текстовый | Маска ввода: (9000) 900-00-00 |
| Контакт | Текстовый | |

5. Закройте таблицу «Поставщики». Откройте таблицу «Товары» в режиме Конструктора (правая кнопка мыши, щелчок по имени таблицы). В Типе данных поля Поставщик внесите изменения. Выберите тип «Мастер подстановок». Следуя указаниям мастера, подставьте поле Название поставщика из таблицы «Поставщики». Закройте таблицу.

6. Выполните команду: *Работа с базой данных/ Схема данных*. Убедитесь в том, что созданные таблицы связаны между собой. Закройте схему данных.

7. Создайте таблицу «Покупатели» со следующим перечнем полей (табл. 16).

Т а б л и ц а 1 6

Состав полей таблицы «Покупатели»

| Имя поля | Тип данных | Свойства поля |
|----------------|------------|-------------------------------|
| Код покупателя | Счетчик | Ключевое |
| Покупатель | Текстовый | |
| Адрес | Текстовый | |
| Телефон | Текстовый | Маска ввода: (9000) 900-00-00 |
| Контакт | Текстовый | |

8. Создайте таблицу «Сотрудники» со следующей структурой (табл. 17).

Т а б л и ц а 1 7

Состав полей таблицы «Сотрудники»

| Имя поля | Тип данных | Свойства поля |
|-----------------|------------|------------------------|
| Табельный номер | Числовое | Ключевое |
| Фамилия | Текстовый | |
| Имя | Текстовый | |
| Отчество | Текстовый | |
| Отдел | Текстовый | |
| Телефон отдела | Текстовый | Маска ввода: 000-00-00 |

9. Создайте таблицу «Личные данные сотрудников» (табл. 18).

Т а б л и ц а 1 8

Состав полей таблицы «Личные данные сотрудников»

| Имя поля | Тип данных | Свойства поля |
|--------------------|------------|---|
| Табельный номер | Числовой | Ключевое |
| Дата рождения | Дата/время | |
| Образование | Текстовый | Тип элемента управления: Список, Тип источника строк: Список значений; Источник строк (элементы списка): высшее, среднее, среднее специальное, неполное высшее, неполное среднее; значение по умолчанию – высшее. |
| Семейное положение | Логический | |
| Домашний адрес | Текстовый | |
| Домашний телефон | Текстовый | |

10. Создайте таблицу «Заказы» следующей структуры (табл. 19).

Т а б л и ц а 19

Состав полей таблицы «Заказы»

| Имя поля | Тип данных | Свойства поля |
|-------------|--------------------|---------------|
| Код заказа | Счетчик | Ключевое |
| Покупатель | Мастер подстановок | |
| Сотрудник | Мастер подстановок | |
| Товар | Мастер подстановок | |
| Количество | Числовой | |
| Дата заказа | Дата/время | |

11. Откройте схему базы данных. При необходимости добавьте в нее недостающие таблицы (всего 6). Оформите связи между таблицами, согласно образцу (рис. 27).

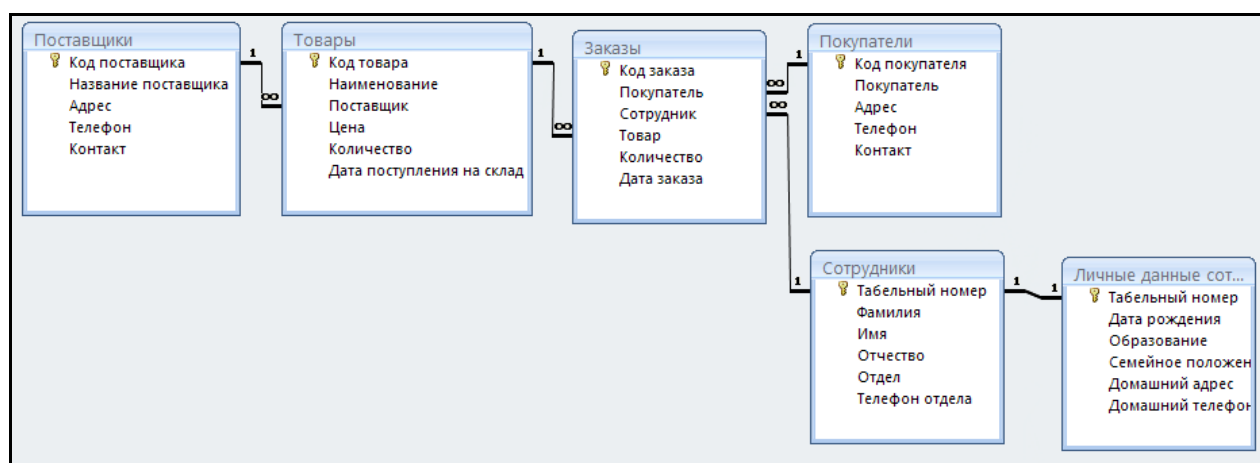


Рис. 27. Схема данных базы данных «Склад»

Задание 2. Создание запросов на выборку.

Порядок работы:

1. На основе таблиц «Сотрудники» и «Личные данные»:
 - 1) список сотрудников (ФИО) и их домашних адресов и телефонов;
 - 2) список сотрудников (табельный номер, ФИО, отдел), имеющих *высшее образование* (использовать условие отбора в поле «Образование»);

3) список сотрудников (табельный номер, ФИО, отдел), имеющих *специальное образование* (использовать условие отбора в поле «Образование»);

4) список сотрудников (ФИО, отдел), чья фамилия (или имя) начинается на определенную букву (использовать условие отбора в поле «фамилия» (или «имя») Like «А*»). Составить минимум 2 списка;

5) список семейных сотрудников;

6) список несемейных сотрудников из любых двух отделов одновременно (использовать условие отбора в двух полях);

7) список сотрудников (ФИО, отдел, дата рождения), родившихся в определенном году;

8) сколько сотрудников работает в каждом отделе? (использовать групповые операции, функция Count);

9) сколько сотрудников каждого типа образования работает в фирме? (использовать групповые операции, функция Count);

10) сколько сотрудников каждого типа семейного положения работает в фирме? использовать групповые операции, функция Count);

11) определить возраст каждого сотрудника (создать вычисляемое поле «Возраст» и использовать построитель выражений, встроенные функции Now – отображает сегодняшнюю дату, Round – округляет до нужного разряда).

12) на основе созданного запроса о возрасте сотрудников создайте список самых молодых сотрудников в каждом отделе.

2. На основе таблиц «Заказы», «Сотрудники» создать запрос, показывающий, какое количество сделок провел каждый из сотрудников.

3. На основе таблиц базы данных составьте следующие запросы на выборку:

1) сколько заказов сделал каждый покупатель?

2) сколько сделок провел каждый сотрудник?

3) какие заказы были сделаны покупателями в определенный период времени (между определенными датами)?

4) используя вычисляемое поле, подсчитайте, на какую сумму были сделаны заказы за весь рассматриваемый период.

5) на основе предыдущего запроса создайте запрос – сведения об обобщенной сумме заказов каждого клиента.

6) используя вычисляемое поле, выясните величину оборота каждого товара (по примеру двух предыдущих запросов).

7) выясните обобщенную сумму сделок каждого сотрудника. Кто принес наибольший оборот?

8) выясните обобщенную сумму сделок по товару каждого поставщика. Кто из поставщиков поставляет самый реализуемый товар?

9) сколько товаров каждого типа осталось на складе?

Задание 3. Создание параметрических запросов на выборку.

В процессе решения задач может возникнуть необходимость вводить значение поля в диалоге с пользователем в процессе выполнения запроса. Для того, чтобы выводилось диалоговое окно, обеспечивающее ввод конкретного значения из поля в условия отбора, необходимо определить параметр запроса. Это позволит также избежать большого количества однотипных запросов – заменить их одним – параметрическим. Имя параметра запроса может задаваться непосредственно в строке «Условие отбора» в квадратных скобках (рис. 28).

При выполнении запроса это имя появится в диалоговом окне «Введите значение параметра». Временная таблица запроса будет доступна после ввода конкретной информации. В таблице будут отражены только те записи, которые относятся к введенному значению параметра. Параметры запроса могут быть использованы и для ввода значения операнда в выражениях условий отбора или вычисляемых полей (рис. 29).

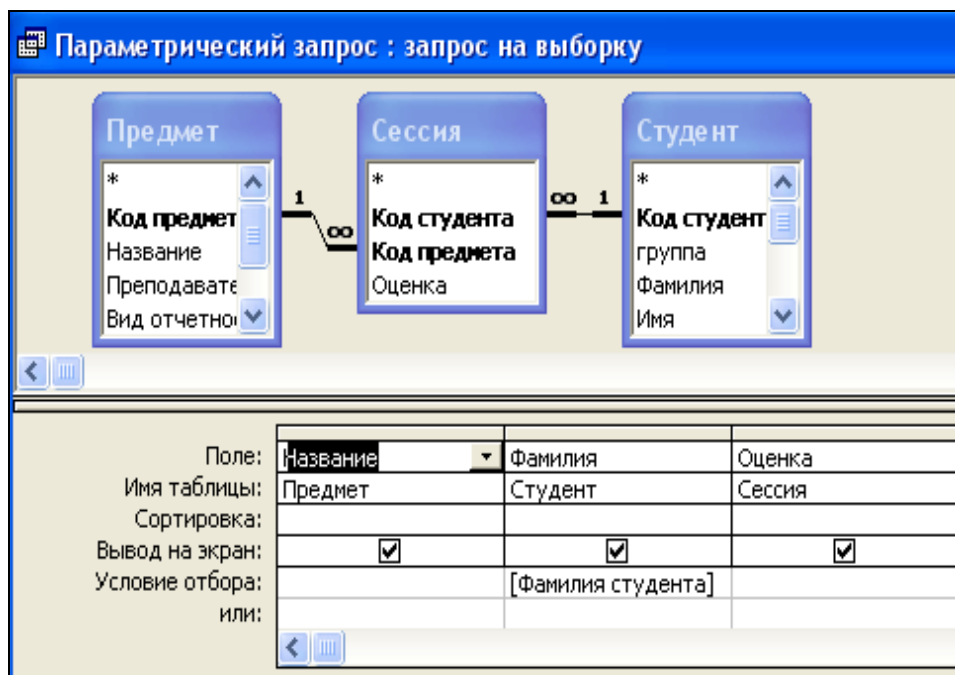


Рис. 28. Конструктор параметрического запроса

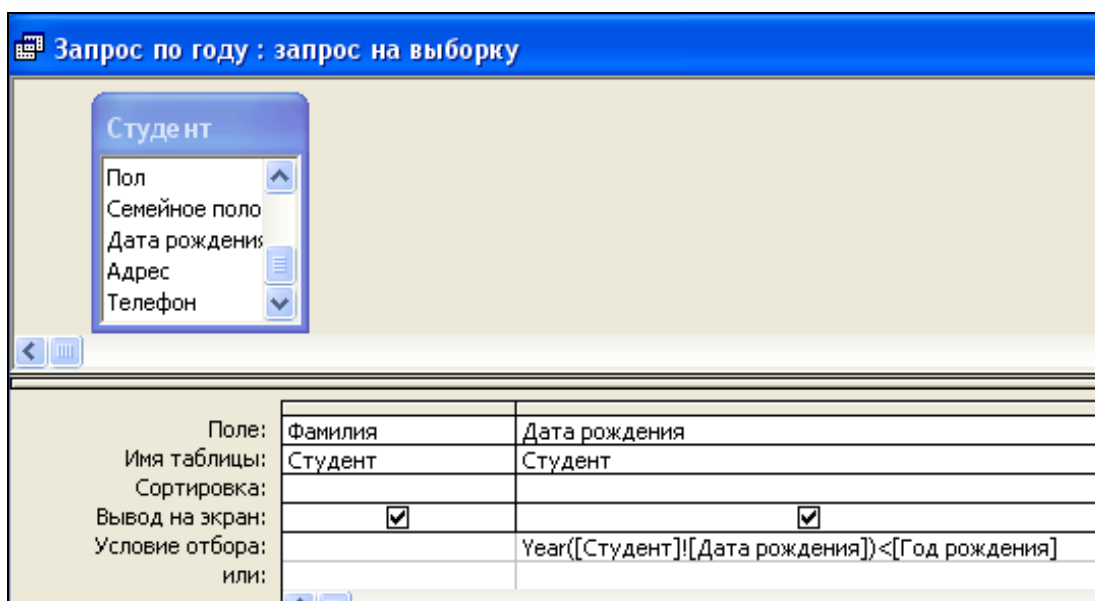


Рис. 29. Параметрический запрос с вычисляемым полем

Создайте параметрические запросы, обеспечивающие отбор следующей информации:

1. Список сотрудников, работающих в *определённом отделе*, с их личными данными.
2. Список товаров на складе, поставляемых *определённым поставщиком*, с ценами и датой поступления на склад.

3. Список покупателей, купивших *определенный товар*, содержащий количество данного товара, сумму заказа, фамилию сотрудника фирмы, обслужившего данный заказ.
4. Список сотрудников фирмы моложе *определенного года* с указанием их семейного положения.
5. Список заказов *определенного покупателя*.
6. Остаток на складе *определенного товара*.

Задание 4. Создание перекрестных запросов.

Перекрестные запросы создаются на основе запросов на выборку. Нужно задать поля, которые будут являться заголовками столбцов, строк и значений будущего запроса. Начать создание запросов данного типа лучше с использованием Мастера запросов. Следуя указаниям Мастера, выберите в качестве основы перекрестного запроса запрос с вычисляемым полем о стоимости заказанных всеми покупателями товаров. Заголовками строк и столбцов выберите поля Покупатель и Товар, а значениями будет Стоимость с операцией Сумма. Сохраните запрос и просмотрите результат.

Постройте перекрестный запрос, показывающий количество заказов товаров всех поставщиков фирмы (то есть, сколько раз покупатели обращались за заказами тех или иных поставщиков). Подумайте, какой запрос на выборку должен быть основой для такого перекрестного запроса.

Задание 5. Создание запросов-изменений.

Для их создания выполняются действия, аналогичные действиям по созданию запросов на выборку. После добавления в бланк запросов нужных таблиц необходимо переключить запрос на выборку в нужный тип запроса нажатием на соответствующую кнопку на панели инструментов. После завершения работы в конструкторе запрос запускают, нажав на кнопку запуска (на панели инструментов) или после того, как запрос будет сохранен двойным щелчком по его имени в перечне запросов.

1. *Запрос на создание таблицы.* Создать таблицу заказов за определенный период времени (например, в течение одного месяца или одной недели), назвав ее «Отправленные заказы». В таблице запроса должны быть поля с информацией о покупателе, товаре, количестве товара и сотруднике, обслуживающем заказ.

2. *Запрос на удаление записей.* С помощью запроса на удаление удалите из таблицы Заказы те записи, которые до этого были помещены в таблицу «Отправленные заказы» - то есть записи того ж периода времени, что и в созданной таблице. **Помните, что отменить данную операцию невозможно, поэтому перед запуском запроса проверьте правильность своих действий.**

3. *Запрос на обновление данных в таблицах.* Предположим, что один из поставщиков поднимает цены на свои товары на 50%. Создайте запрос с условием отбора, который бы соответствующим образом менял цены одного из поставщиков. Помните, что каждый раз при запуске данного запроса будет выполняться операция обновления данных, поэтому его запуск производите только один раз!

Задание 6. Работа с отчетами в режиме Конструктора.

Порядок работы:

1. *Отчет-Наклейки.*

Создайте запрос на выборку «**Рассылка**» для отправления сделанных заказов. В конструктор запроса добавьте таблицы: Покупатели, Заказы, Сотрудники, Товары. В таблицу запроса добавьте поля (рис. 30).

| | | | | | | | |
|-----------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| Поле: | Покупатель | Адрес | Контакт | Количество | Наименование | Фамилия | Телефон отдела |
| Имя таблицы: | Покупатели | Покупатели | Покупатели | Заказы | Товары | Сотрудники | Сотрудники |
| Сортировка: | | | | | | | |
| Вывод на экран: | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Условие отбора: | | | | | | | |
| или: | | | | | | | |

Рис. 30. Конструктор запроса «Рассылка»

Просмотрите временную таблицу запроса, сохраните его и закройте.

Перейдите на вкладку Создание. Выберите Отчеты/Наклейки. Следуйте указаниями Мастера наклеек (нажимайте Далее) вплоть до шага «Прототип наклейки». Здесь сделайте нужные надписи, а имена полей (на рис. 31 в фигурных скобках) выделите и перенесите стрелкой.

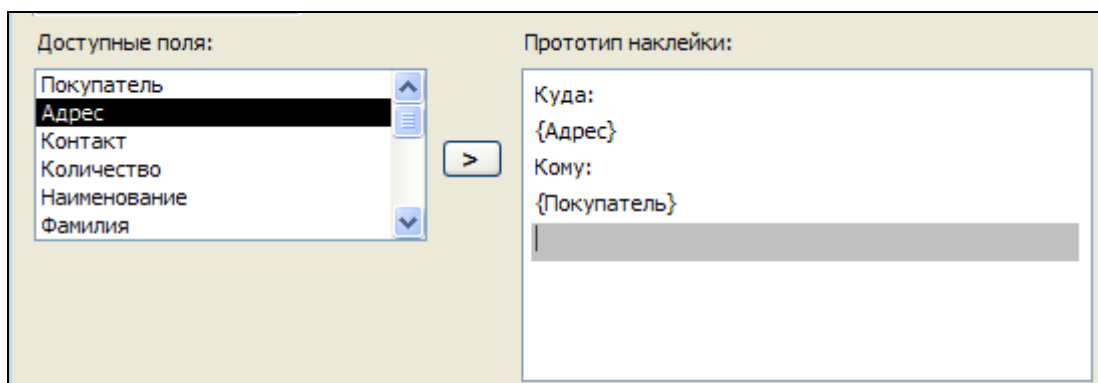
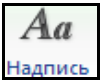


Рис. 31. Создание отчета «Наклейки»

Нажмите Готово.

Закройте окно предварительного просмотра. Вы в режиме конструктора. Добавьте после последнего поля надпись (кнопка ) по образцу (рис. 32) и, нажав кнопку «Добавить поле», откройте список полей запроса «Рассылка» и перенесите поле Фамилия.

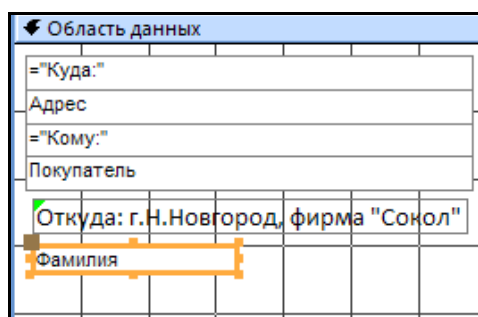


Рис. 32. Конструктор отчета «Наклейки»

Перейдите в режим просмотра отчета, сохраните его и закройте.

2. *Отчет-Письмо покупателю* создаем исходно с помощью мастера отчетов на основе запроса «Рассылки». В макет отчета включите все доступные поля. Проведите группировку по полю

Контакт, выберите оформление и нажмите Готово. Перейдите в режим Конструктора и внесите изменения. На рис. 33 сделаны надписи «Уведомление об отправленных заказах», «Уважаемый господин (госпожа)», «Фирма», «Заказала следующие товары (см. перечень ниже)». «Уведомляем Вас ...». Удалены подписи полей Контакт, Покупатель. Изменена подпись поля Адрес на «по адресу».

| Заголовок отчета | | | | | | | | | |
|---|--|--|--|--|--|--|--|--|--|
| | | | | | | | | | |
| Верхний колонтитул | | | | | | | | | |
| | | | | | | | | | |
| Заголовок группы 'Контакт' | | | | | | | | | |
| Уведомление об отправленных заказах | | | | | | | | | |
| | | | | | | | | | |
| Уважаемый господин (госпожа) | | | | | | | | | |
| Контакт | | | | | | | | | |
| Заголовок группы 'Покупатель' | | | | | | | | | |
| Фирма | | | | | | | | | |
| Покупатель | | | | | | | | | |
| по адресу Адрес заказала следующие товары (см перечень ниже) | | | | | | | | | |
| Уведомляем Вас, что данные товары были высланы Вам по указанному адресу. Просим в установленные сроки оплатить выставленный Вам счет. | | | | | | | | | |
| Количество Товар Фамилия Телефон отдела | | | | | | | | | |
| Область данных | | | | | | | | | |
| Количество Наименование Фамилия Телефон отдела | | | | | | | | | |
| Нижний колонтитул | | | | | | | | | |
| =Now() = "Страница " & [Page] & " из " & [Pages] | | | | | | | | | |
| Примечание отчета | | | | | | | | | |

Рис. 33. Конструктор отчета «Письмо покупателю»

2.4. Темы для самостоятельных работ

1. Создать базу данных для учета продаж автомобилей разных моделей в 2 филиалах фирмы, в фирму обращаются 10 покупателей в разных филиалах. В каждом филиале их обслуживают определенные сотрудники фирмы (менеджеры). Заполните базу, предварительно создав формы для ввода данных. Продумайте систему возможных запросов по созданной базе данных. Создайте отчеты.

2. Создать базу данных для учета учебной литературы и определения количества выданных учебников при условии, что библиотекой пользуются 20 студентов из трех групп, список литературы состоит из 15 названий, причем каждая книга имеет стоимость, которую студент должен возместить в случае потери книги. Каждый студент берет разные учебники, в библиотеке хранится от 6 до 10 экземпляров каждой книги. Обеспечьте ввод данных с помощью форм. Продумайте систему возможных запросов по созданной базе данных. Например, запрос для подсчета количества книг каждого вида, оставшихся в библиотеке. Создайте отчеты, например, отчет о том, какие книги находятся на руках у студентов каждой группы.

3. Создайте базу данных для учета вкладов клиентов банка. Виды вкладов могут быть, например, до востребования с доходом 5% годовых; срочный – 13% годовых; пенсионный – 18% годовых. В банк внесли деньги 10 клиентов на разные виды вкладов (несколько клиентов вносит деньги на вклады более одного вида). При внесении вклада клиентов обслуживают сотрудники банка, информация о которых также должна присутствовать в базе данных. Создайте формы для ввода данных и заполните базу. Продумайте систему возможных запросов по созданной базе данных. Создайте отчеты, например отчет о работе банка, в котором

нужно отразить сумму к выдаче каждого клиента через год по каждому виду вкладов.

4. Создать базу данных «Отдел кадров» (принимается на работу не менее 30 сотрудников): ФИО, должность, оклад, дата поступления на работу, стаж к моменту поступления, отдел, адрес. О каждом сотруднике необходимо собрать максимальное количество информации, разбив ее на две таблицы – служебная и личная информация. Заполните базу, предварительно создав формы. Продумайте систему возможных запросов по созданной базе данных. Например, в процессе работы ряд сотрудников уволились или были изменены их должности и/или оклады, семейное положение, количество детей. Создайте отчеты, например, список лиц с указанием фамилий, должностей и отделов, имеющих оклад выше среднего по предприятию.

5. Создайте базу данных «Регистратура», содержащую информацию о врачах и пациентах. Создать формы для ввода данных. Продумайте систему возможных запросов по созданной базе данных, например создать запрос, выводящий фамилию и специальность врача и список его пациентов. Создайте отчеты, например отчет, отражающий список пациентов, сгруппированный по полю Врач и отсортированный по убыванию возраста его пациентов.

6. Создайте базу данных для подведения итогов вступительных экзаменов. При поступлении 20 абитуриентов сдают 3 экзамена: математика, история, английский язык. Факультет может принять 15 человек. Создайте формы для ввода данных. Продумайте систему возможных запросов по созданной базе данных. Создайте отчеты, например отчет об абитуриентах, принятых на факультет. Разошлите им официальные уведомления (отчеты).

Литература

1. Гаспарян, М.С., Лихачева, Г.Н. Информационные системы и технологии: учебно-методический комплекс / М.С.Гаспарян, Г.Н.Лихачева. – М.: Издат. Центр ЕАОИ, 2008. – 384 с.
2. Информатика в экономике: учеб. пособие / Под ред. проф. Б.Е.Одинцова, проф. А.Н.Романова. – М.: Вузовский учебник, 2008. – 478 с.
3. Информационные системы и технологии в экономике и управлении: учеб. Пособие/ под ред. Проф. В.В.Трофимова. – 2-е изд., перераб. и доп. – М.: Высшее образование, 2007. – 480 с.
4. Информационные системы в экономике: учебник / Под ред. Г.А.Титоренко. – М.: Юнити – Дана, 2008. – 463 с.
5. Коноплева, И.А., Денисов, А.В., Хохлова, О.А. Информационные технологии: учеб. пособие. – М.: ТК Велби, изд-во Проспект, 2009 – 304 с.
6. Соболев, Б.В. и др. Информатика: учебник. – Ростов-на-Дону: Феникс, 2007. – 446 с.
7. Советов, Б.Я., Цехановский, В.В. Информационные технологии: Учеб. для вузов. – М.: Высшая школа, 2006. – 236 с.

Юрченко Татьяна Владиславовна

**ИНФОРМАЦИОННЫЕ СИСТЕМЫ В ЭКОНОМИКЕ И
УПРАВЛЕНИИ**

Учебное пособие

Редактор

С.А.Елизарова

Подписано в печать _____ Формат 60x90 1/16 Бумага газетная. Печать трафаретная.

Уч. изд. л. ____ Усл. печ. л. ____ Тираж 1000 экз. Заказ № _____.

Федеральное государственное образовательное учреждение высшего профессионального образования

«Нижегородский государственный архитектурно-строительный университет»

603950, Н.Новгород, Ильинская, 65

Полиграфцентр ННГАСУ, 603950, Н.Новгород, Ильинская, 65