



Автономная некоммерческая образовательная организация
высшего образования
«Воронежский экономико-правовой институт»
(АНОО ВО «ВЭПИ»)



УЧЕБНО-МЕТОДИЧЕСКИЙ КОМПЛЕКС ДИСЦИПЛИНЫ (МОДУЛЯ)

Б1.В.06 Ведение баз данных в цифровых системах
(наименование дисциплины (модуля))

38.05.01 Экономическая безопасность
(код и наименование направления подготовки)

Специализация Экономико-правовое обеспечение экономической безопас-
ности в условиях цифровизации
(наименование направленности (профиля))

Квалификация выпускника Специалист
(наименование квалификации)

Форма обучения Очная, заочная
(очная, заочная)

Рекомендована к использованию Филиалами АНОО ВО «ВЭПИ».

Воронеж 2024

Учебно-методический комплекс дисциплины (модуля) рассмотрен и одобрен на заседании кафедры прикладной информатики.

Протокол от « 15 » апреля 2024 г. № 8

Заведующий кафедрой



М.С. Агафонова

Разработчики:

Доцент



В.А. Поздняков

1. Методические рекомендации по выполнению практических работ

Методические указания по выполнению практических работ по дисциплине «Ведение баз данных в цифровых системах» разработаны в соответствии с рабочей программой и предназначены для приобретения необходимых практических навыков и закрепления теоретических знаний, обобщения и систематизации знаний перед экзаменом.

Методические указания предназначены для обучающихся специальности 38.05.01 Экономическая безопасность.

Порядок выполнения практической работы

- записать название работы, ее цель в тетрадь;
- выполнить основные задания в соответствии с ходом работы;
- выполнить индивидуальные задания.

Рекомендации по оформлению практической работы

Задания выполняются обучающимися по шагам. Необходимо строго придерживаться порядка действий, описанного в практической работе.

Результаты выполнения практических заданий необходимо сохранять в своей папке на компьютере или USB – накопителе.

В случае пропуска занятий обучающийся осваивает материал самостоятельно в свободное от занятий время и сдает практическую работу с пояснениями о выполнении.

Критерии оценки практической работы

- наличие цели выполняемой работы, выполнение более половины основных заданий (удовлетворительно);
- наличие цели выполняемой работы, выполнение всех основных и более половины дополнительных заданий (хорошо);
- наличие цели выполняемой работы, выполнение всех основных и индивидуальных заданий (отлично).

Практическая работа № 1 Сбор и анализ информации

Цель практической работы. Получить теоретические знания и практические навыки реализации баз данных (БД). Осуществить анализ предметной области. Освоить концептуальное проектирование и научиться определять сущности и атрибуты БД. Научиться разрабатывать инфологическую модель БД в виде ER-диаграмм. Получить теоретические знания и практические навыки при физическом проектировании баз данных (БД). Научиться создавать даталогическую модель БД.

I. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

1. ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ БАЗЫ ДАННЫХ

Понятие БД и СУБД

Информационная система - система, реализующая автоматизированный сбор, обработку и манипулирование данными и включающая технические средства обработки данных, программное обеспечение и соответствующий персонал.

Цель любой информационной системы - обработка данных об объектах реального мира. Основой информационной системы является **база данных**. В широком смысле

слова **база данных** - это совокупность сведений о конкретных объектах реального мира в какой-либо предметной области.

Под **предметной областью** принято понимать часть реального мира, подлежащего изучению для организации управления и в конечном счете автоматизации, например, предприятие, вуз и т.д.

Создавая базу данных, пользователь стремится упорядочить информацию по различным признакам и быстро производить выборку с произвольным сочетанием признаков. Большое значение при этом приобретает структурирование данных.

Структурирование данных - это введение соглашений о способах представления данных.

Неструктурированными называют данные, записанные, например, в текстовом файле.

Ниже приведен пример неструктурированных и структурированных данных, содержащих сведения о студентах (номер личного дела, фамилию, имя, отчество и год рождения).

Неструктурированные данные:

Личное дело № 16493. Сергеев Петр Михайлович, дата рождения 1 января 1976 г.; Л/д № 16593, Петрова Анна Владимировна, дата рожд. 15 марта 1975 г.; № личн. дела 16693, д.р. 14.04.76, Анохин Андрей Борисович

Легко убедиться, что сложно организовать поиск необходимых данных, хранящихся в неструктурированном виде.

Структурированные данные:

№ личного	Фамилия	Имя	Отчество	Дата рождения
16493	Сергеев	Петр	Михайлович	01.01.76
16593	Петрова	Анна	Владимировна	15.03.75
16693	Анохин	Андрей	Борисович	14.04.76

В современной технологии баз данных предполагается, что создание базы данных, ее поддержка и обеспечение доступа пользователей к ней осуществляются централизованно с помощью специального программного инструментария - **системы управления базами данных (СУБД)**.

База данных (БД) - это поименованная совокупность данных, отражающая состояние объектов и их отношений в рассматриваемой предметной области.

Объектом называется элемент предметной области, информацию о котором мы сохраняем.

Объект может быть **реальным** (например, человек, изделие; или населенный пункт) и **абстрактным** (например, событие, счет покупателя или изучаемый студентами курс).

Так, в области продажи автомобилей примерами **объектов** могут служить **МОДЕЛЬ АВТОМОБИЛЯ, КЛИЕНТ и СЧЕТ**. На товарном складе - это **ПОСТАВЩИК, ТОВАР, ОТПРАВЛЕНИЕ** и т. д.

Понятие базы данных тесно связано с такими понятиями структурных элементов, как **поле, запись, файл (таблица)** (рис.1).



↓
Столбец называется - Поле

Тип товара	Ед.изм	Название товара
Конд. изд.	кг	Печенье

← Строка называется – Запись (Кортеж)

↙ Значение поля

Рис.1 Основные структурные элементы БД

Структурные элементы базы данных

Поле - элементарная единица логической организации данных, которая соответствует неделимой единице информации - реквизиту. Для описания поля используются следующие характеристики:

- **имя**, например, **Фамилия, Имя, Отчество, Дата рождения**;
- **тип**, например, символьный, числовой, денежный;
- **длина**, например, 15 байт, причем будет определяться максимально возможным количеством символов;
- **точность** для числовых данных, например два десятичных знака для отображения дробной части числа,

Запись - совокупность логически связанных полей.

Экземпляр записи - отдельная реализация записи, содержащая конкретные значения ее полей.

Файл (таблица) - совокупность экземпляров записей одной структуры.

Описание логической структуры записи файла содержит последовательность расположения полей записи и их основные характеристики, как это показано на рис. 2 и 3.

Имя файла					
Поле		Признак ключа	Формат поля		
Имя (обозначение)	Полное наименование		Тип	Длина	Точность (для чисел)
имя 1					
имя N					

Рис. 2. Описание логической структуры записи файла

В структуре записи файла указываются поля, значения которых являются ключами: **первичными (ПК) и внешними (ВК)**,

Первичный ключ (ПК) - это одно или несколько полей, однозначно идентифицирующих запись. Если первичный ключ состоит из одного поля, он называется **простым**, если из нескольких полей - **составным** ключом.

Внешний ключ (ВК) - это одно или несколько полей, которые выполняют роль поисковых или группировочных признаков. В отличие от первичного, значение внешнего ключа может повторяться в нескольких записях файла, то есть он не является уникальным. Если по значению первичного ключа может быть найден один единственный экземпляр записи, то по внешнему - несколько.

Имя файла: СТУДЕНТ					
Поле		Признак ключа	Формат поля		
Обозначение	Наименование		Тип	Длина	Точность
Номер	№ личного дела	•	Симв	5	

Фамилия	Фамилия студента		Симв	15	
Имя	Имя студента		Симв	10	
Отчество	Отчество студента		Симв	15	
Дата	Дата рождения		Дата	8	

Рис. 3. Описание логической структуры записи файла **СТУДЕНТ**

Понятие модели данных

Для того, чтобы спроектировать структуру базы данных, необходима исходная информация о предметной области. Желательно, чтобы эта информация была представлена в формализованном виде.

Такое формализованное описание предметной области будем называть **инфологической (infological) моделью предметной области (ИЛМ)** или **концептуальной моделью (КМ)**.

Ядром любой базы данных является модель данных. **Модель данных** представляет собой множество структур данных, ограничений целостности и операций манипулирования данными. С помощью модели данных могут быть представлены объекты предметной области и взаимосвязи между ними.

Модель данных - совокупность структур данных и операций их обработки.

СУБД основывается на использовании **иерархической, сетевой или реляционной модели**, на комбинации этих моделей или на некотором их подмножестве.

Самой распространенной моделью данных является - **реляционная**.

Иерархическая модель данных

Иерархическая модель организует данные в виде древовидной структуры

К основным понятиям иерархической структуры относятся: **уровень, элемент**

(**узел**), **связь**. Дерево представляет собой иерархию элементов, называемых узлами.

Узел - это совокупность атрибутов данных, описывающих некоторый объект. На самом верхнем уровне иерархии имеется один и только один узел - **корень**. Каждый узел, кроме корня, связан с одним узлом на более высоком уровне, называемым **исходным** для данного узла. Ни один элемент не имеет более одного исходного. Каждый элемент может быть связан с одним или несколькими элементами на более низком уровне. Они называются **порожденными** (рис. 4).

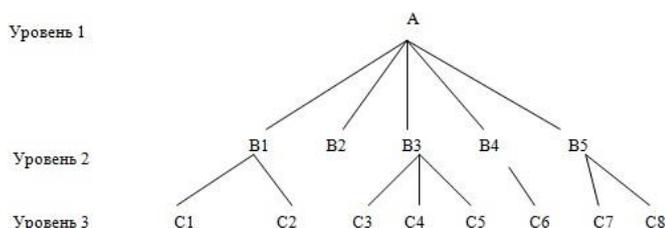


Рис. 4. Графическое изображение иерархической структуры БД

К каждой записи базы данных существует только один (иерархический) путь от корневой записи. Например, как видно из рис. 4, для записи C4 путь проходит через записи A и B3.

Сетевая модель данных

Сетевая модель организует данные в виде сетевой структуры.

Структура называется сетевой, если в отношениях между данными порожденный элемент имеет более одного исходного.

В сетевой структуре при тех же основных понятиях (уровень, узел, связь) каждый элемент может быть связан с любым другим элементом.

На рис. 5 изображена сетевая структура базы данных в виде графа.

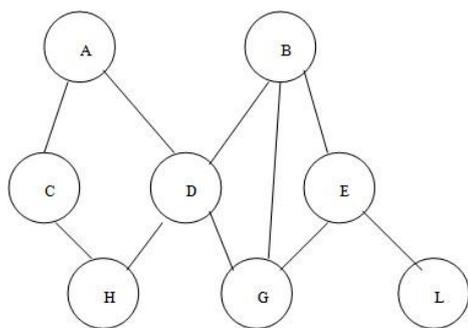


Рис. 5. Графическое изображение сетевой структуры

Реляционная модель данных

Реляционная модель данных является совокупностью взаимосвязанных двумерных таблиц объектов модели.

Например, реляционной таблицей можно представить информацию о студентах, обучающихся в вузе (рис. 6).

№ личного дела	Фамилия	Имя	Отчество	Дата рождения	Группа
16493	Сергеев	Петр	Михайлович	01.01.76	111
16593	Петрова	Анна	Владимировна	15.03.75	112
16693	Анохин	Андрей	Борисович	14.04.76	111

Рис. 6. Пример реляционной таблицы

Связи между двумя логически связанными таблицами в реляционной модели устанавливаются по равенству значений одинаковых атрибутов этих таблиц.

Каждая реляционная таблица представляет собой двумерный массив и обладает следующими **свойствами**:

- - каждый элемент таблицы - один элемент данных;
- - все столбцы в таблице однородные, т.е. все элементы в столбце имеют одинаковый тип (числовой, символьный и т.д.) и длину;
- - каждый столбец имеет уникальное имя;
- - одинаковые строки в таблице отсутствуют;
- - порядок следования строк и столбцов может быть произвольным. При описании реляционной модели часто используют следующие термины:

отношение, кортеж, домен.

Отношения представлены в виде таблиц, строки которых соответствуют **записям (кортежам)**, а столбцы полям, атрибутам отношений (**доменам**).

Поле, каждое значение которого однозначно определяет соответствующую запись, называется простым ключом (ключевым полем). Если записи однозначно определяются значениями нескольких полей, то такая таблица базы данных имеет **составной ключ**.

В примере, показанном на рис.6, ключевым полем таблицы является "№ личного дела".

Между двумя реляционными таблицами могут быть сформированы **связи**.

Различные таблицы, могут быть **связаны между собой через общее поле данных**. На рис. 7 показан пример реляционной модели, построенной на основе отношений:

СТУДЕНТ, СЕССИЯ, СТИПЕНДИЯ.



Рис.7. Пример реляционной модели

Таблица **СТУДЕНТ** имеет поля: Номер, Фамилия, Имя, Отчество, Дата рождения, Группа; **СЕССИЯ** - Номер, Оценка 1, Оценка 2, Оценка 3, Оценка 4, Результат;

СТИПЕНДИЯ - Результат, Процент

Таблицы **СТУДЕНТ** и **СЕССИЯ** имеют совпадающие ключи (**Номер**), что дает возможность легко организовать связь между ними.

Таблица **СЕССИЯ** имеет **первичный ключ Номер** и содержит **внешний ключ Результат**, который обеспечивает ее связь с таблицей **СТИПЕНДИЯ**. Благодаря имеющимся связям достигаются следующие преимущества:

1. Удастся избежать дублирования информации. Все необходимые данные можно хранить только в одной таблице. Так, например, нет необходимости в таблице **СЕССИЯ** хранить номер группы каждого студента, сдающего экзамены, достаточно задать связь с таблицей **СТУДЕНТ**.

2. В реляционных базах данных легко производить изменения. Если в таблице **СЕССИЯ** изменить какие-нибудь значения, то правильная информация автоматически будет связана с другими таблицами, ссылающимися на первую (например, таблица **СТИПЕНДИЯ**).

3. В нереляционных базах данных сложно передать все имеющиеся зависимости, т.е. связать друг с другом данные из различных таблиц. Реляционная база данных выполняет все эти действия автоматически.

4. В реляционных базах данных удастся легко избежать установления ошибочных связей между различными таблицами данных, а необходимый объем памяти сокращен до минимума.

2. ПРОЕКТИРОВАНИЕ БАЗ ДАННЫХ

Анализ и описание предметной области БД

Предметной областью называется фрагмент реальности, который описывается или моделируется с помощью БД и ее приложений. В предметной области выделяются информационные объекты – идентифицируемые объекты реального мира, процессы, системы, понятия и т.д., сведения о которых хранятся в БД.

Анализ предметной области выполняется в следующем порядке: название предприятия, цель деятельности предприятия, структура предприятия, информационные потребности пользователей. Для описания ПО следует привести основные решаемые задачи БД. Дать словесное описание процесса функционирования ПО, проанализировать основные хозяйственные операции, которые совершаются в ПО. Привести анализ структуры предприятия, перечислить задачи, решаемые отдельными подразделениями.

Анализ предметной области целесообразно разбить на три фазы:

1. анализ концептуальных требований и информационных потребностей;
2. выявление информационных объектов и связей между ними;
3. построение концептуальной модели предметной области и проектирование концептуальной схемы БД.

Анализ концептуальных требований и информационных потребностей

Требования пользователей к разрабатываемой БД представляют собой список запросов с указанием их интенсивности и объемов данных. Эти сведения разработчики БД получают в диалоге с ее будущими пользователями. Здесь же выясняются требования к вводу, обновлению и корректировке информации. Требования пользователей уточняются и дополняются при анализе имеющихся и перспективных задач.

Рассмотрим примерный состав вопросника при анализе различных предметных областей. *Пример.* Предлагается разработать БД для учета студентов вуза. Анализ предметной области:

1. Сколько студентов учится в вузе? 2. Сколько факультетов и отделений в вузе? 3. Как распределены студенты по факультетам отделений и курсам? 4. Сколько дисциплин читается на каждом курсе по каждой специальности? 5. Как часто обновляется информация в БД? 6. Сколько преподавателей в вузе? 7. Сколько иногородних студентов живет в общежитии, на частных квартирах? 8. Сколько лекционных аудиторий и аудиторий для проведения практических занятий, лабораторий? 9. Какая преемственность существует между читаемыми курсами? Как информация, представленная в п.п. 1-9, используется в настоящее время (расписание занятий, экзаменов, зачетов и т.д.) и как ее собираются использовать? Сколько раз в день, сколько человек и кто пользуются БД?

Выявление информационных объектов и связей между ними

Вторая фаза анализа предметной области состоит в выборе информационных объектов, задании необходимых свойств для каждого объекта, выявлении связей между объектами, определении ограничений, накладываемых на информационные объекты, типы связей между ними, характеристики информационных объектов. Проанализируем предметную область на примере БД "Видеомагнитофоны".

При выборе информационных объектов постараемся ответить на ряд вопросов:

1. На какие классы можно разбить данные, подлежащие хранению в БД?
2. Какое имя можно присвоить каждому классу данных?
3. Какие наиболее интересные характеристики (с точки зрения пользователя) каждого класса данных можно выделить?
4. Какие имена можно присвоить выбранным наборам характеристик?

Пример. БД "Видеомагнитофоны", рассчитанной на пользователей, которые хотят приобрести данный вид техники.

После беседы с различными пользователями и просмотра каталогов было выяснено, что интерес представляют три информационных объекта: видеомагнитофон, видеоплеер, видеокассета.

Рассмотрим наиболее существенные характеристики каждого информационного объекта.

Объект - ВИДЕОМАГНИТОФОН.

Атрибуты - страна-изготовитель, фирма-изготовитель, № модели, телевизионные системы, число кассетных гнезд, ресурс непрерывной работы, система автопоиска, напряжение в сети, наличие таймера, число программ, габаритные размеры, масса, цена в долларах, год выпуска.

Объект - ВИДЕОПЛЕЙЕР.

Атрибуты - страна-изготовитель, фирма-изготовитель, № модели, телевизионные системы, число воспроизводящих головок, ресурс непрерывной работы, напряжение в сети, наличие таймера, габаритные размеры, масса, цена в долларах, год выпуска.

Объект - ВИДЕОКАССЕТА.

Атрибуты - наименование, страна-изготовитель, фирма-изготовитель, тип кассеты, время проигрывания, цена в долларах.

Далее выделим связи между информационными объектами. В ходе этого процесса постараемся ответить на следующие вопросы:

1. Какие типы связей между информационными объектами?
2. Какое имя можно присвоить каждому типу связей?
3. Каковы возможные типы связей, которые могут быть использованы впоследствии?
4. Имеют ли смысл какие-нибудь комбинации типов связей?

Попытаемся задать ограничения на объекты и их характеристики.

Под **ограничением целостности** обычно понимают логические ограничения, накладываемые на данные. Ограничение целостности - это такое свойство, которое мы задаем для некоторого информационного объекта или его характеристики и которое должно сохраняться для каждого их состояния. Введем следующие ограничения:

1. Значение атрибута "число кассетных гнезд" изменяется от 1 до 2.
2. Значение атрибута "ресурс непрерывной работы" изменяется от 4 до 24.
3. Значение атрибута "напряжение в сети" изменяется от 110 до 240 В.
4. Значение атрибута "число программ" изменяется от 1 до 20 и т.д.

Связи между различными классами объектов.

Помимо классов объектов в ИЛМ отображают связи между различными классами объектов. Такие связи моделируют отношения между объектами различных видов в реальном мире. При отборе связей помещаемых в ИЛМ следует руководствоваться информационными потребностями пользователей базы данных.

Каждая связь характеризуется именем, типом, классом принадлежности и направлением. Имя связи должно быть глагольным оборотом, например

«**Принадлежит**», «**Закреплены за**», «**Входит в**» и т.д.

Все информационные объекты предметной области связаны между собой. Соответствия, отношения, возникающие между объектами предметной области,

называются связями. Различаются связи нескольких типов, для которых введены следующие обозначения: Различают четыре типа связи:

- «**один к одному**» (1:1); □ «**один ко многим**» (1:M);
- «**многие к одному**» (M:1)
- «**многие ко многим**» (M:M).

Рассмотрим эти типы связей на примерах.

Пример. Дана совокупность информационных объектов, отражающих учебный процесс в вузе:

СТУДЕНТ (Номер, Фамилия, Имя, Отчество, Пол, Дата рождения, Группа) СЕССИЯ (Номер, Оценка 1, Оценка 2, Оценка 3, Оценка 4, Результат) СТИПЕНДИЯ (Результат, Процент)

ПРЕПОДАВАТЕЛЬ (Код преподавателя, Фамилия, Имя, Отчество)

Связь один к одному (1:1) предполагает, что в каждый момент времени одному экземпляру информационного объекта А соответствует не более одного экземпляра информационного объекта В и наоборот. Рис. 8 иллюстрирует указанный тип отношений.

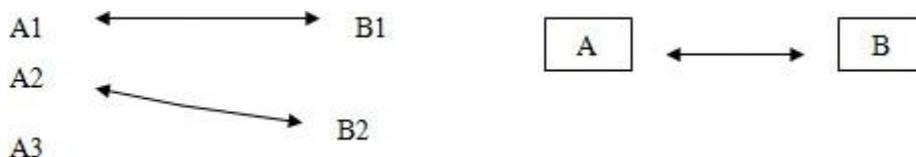


Рис. 8. Графическое изображение реального отношения 1:1

Примером связи 1:1 может служить связь между информационными объектами **СТУДЕНТ** и **СЕССИЯ**:

СТУДЕНТ <-> СЕССИЯ

Каждый студент имеет определенный набор экзаменационных оценок в сессию. При связи **один ко многим (1 : M)** одному экземпляру информационного объекта

А соответствует 0, 1 или более экземпляров объекта В, но каждый экземпляр объекта В связан не более чем с 1 экземпляром объекта А. Графически данное соответствие имеет вид, представленный на рис. 9.

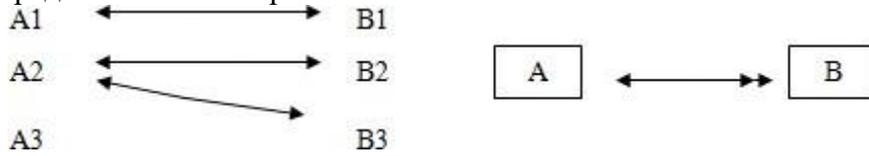


Рис. 9. Графическое изображение реального отношения 1:M

Примером связи 1: M служит связь между информационными объектами **СТИПЕНДИЯ** и **СЕССИЯ**:

СТИПЕНДИЯ <- - >> СЕССИЯ

Установленный размер стипендии по результатам сдачи сессии может повторяться многократно для различных студентов.

Связь **многие ко многим (M:M)** предполагает, что в каждый момент времени одному экземпляру информационного объекта А соответствует 0, 1 или более экземпляров объекта В и наоборот. На рис. 10 графически представлено указанное соответствие.

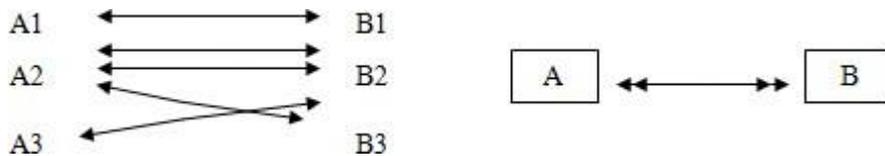


Рис. 10. Графическое изображение реального отношения M : M

Примером данного отношения служит связь между информационными объектами **СТУДЕНТ** и **ПРЕПОДАВАТЕЛЬ**:

СТУДЕНТ <<->>ПРЕПОДАВАТЕЛЬ

Один студент обучается у многих преподавателей, один преподаватель обучает многих студентов.

Связь «многие к одному» при создании БД физически обычно организуется путем введения дополнительного поля в таблицу со стороны «много». Это поле называется **внешний ключ**. На рис. 10. код группы - внешний ключ.

Студент				Группа		
Фамилия	Имя	...	Код группы	Код группы	Название группы	Код факультета
Петров.	Иван	...	2	1	КТ171	
Сидоров	Сергей	...	2			
Лаптева	Марина	...	1			
Киров	Андрей	...	3			
Жоголева	Ирина	...	3			
				2	МТ181	
				3	ЕК382	

Рис. 10. Введение внешнего ключа в БД

Построение инфологической (концептуальной модели) предметной области

Заключительная фаза анализа предметной области состоит в проектировании ее инфологической структуры или концептуальной модели.

Концептуальная модель включает описания объектов и их взаимосвязей, представляющих интерес в рассматриваемой предметной области (ПО) и выявляемых в результате анализа данных.

Концептуальная модель применяется для структурирования предметной области с учетом информационных интересов пользователей системы. Она дает возможность систематизировать информационное содержание предметной области, позволяет как бы "подняться вверх" над ПО и увидеть ее отдельные элементы. При этом уровень детализации зависит от выбранной модели.

Концептуальная модель является представлением точки зрения пользователя на предметную область и не зависит ни от программного обеспечения СУБД, ни от технических решений.

Концептуальная модель должна быть стабильной. Могут меняться прикладные программы, обрабатывающие данные, может меняться организация их физического хранения, концептуальная модель остается неизменной или увеличивается с целью включения дополнительных данных.

Одной из распространенных моделей концептуальной схемы является модель «**сущность - связь**» (**ER-моделей** (или ER-диаграмм)). Основными конструкциями данной модели являются сущности и связи.

Под **сущностью** понимают основное содержание объекта ПО, о котором собирают информацию. В качестве сущности могут выступать место, вещь, личность, явление.

Экземпляр сущности - конкретный объект. **Например:**

<p>сущность (объект) - служащий экземпляр сущности - Иванов А.В.;</p> <p>сущность (объект) - институт экземпляр сущности - МГУ.</p>

Сущность принято определять **атрибутами** - поименованными характеристиками.

Например:

<p>сущность - служащий атрибуты: ФИО, год рождения, адрес, образование и т.д.</p>

Чтобы задать атрибут в модели, ему надо присвоить имя и определить область допустимых значений. Одно из назначений атрибута - идентифицировать сущность.

Связь определяет отношения между сущностями. **Типы связей: один к одному, один ко многим, многие ко многим.**

При построении модели «сущность - связь» используют **графические диаграммы**. При этом обозначают: сущности - прямоугольниками, атрибуты - овалами, связи - ромбами, см. рис.11.

На практике приходится строить несколько вариантов моделей, из которых выбирается одна, наиболее полно отображающая предметную область.

Классом объектов называют совокупность объектов, обладающих одинаковым набором свойств. Например, для объектов класса «СТУДЕНТ» таким набором свойств являются: «ГОД_РОЖДЕНИЯ», «ПОЛ» и др.

Объекты могут быть реальными, как названный выше объект «СТУДЕНТ», и абстрактными, как, например, «ПРЕДМЕТЫ», которые изучают студенты.

Пример. Спроектировать БД "Сессия". База данных должна выдавать оперативную информацию об успеваемости студентов на факультетах в семестре. Результатами сессии считать только экзамены.

По сути дела и БД исходя из формулировки задания можно выделить лишь одно приложение. Речь идет об успеваемости студентов разных факультетов по тем или иным дисциплинам. Более конкретно речь идет о выдаче справок по результатам сессии каждого

студента, учебной группы, курса, факультета, а также об автоматизированном составлении ведомости. Выберем следующие сущности:

ИНСТИТУТ, ФАКУЛЬТЕТ, СТУДЕНТ, ПРЕПОДАВАТЕЛЬ, ДИСЦИПЛИНА.

В данном примере можно выделить сущность ЭКЗАМЕН или ВЕДОМОСТЬ, но можно не выделять, а сформировать ведомость из имеющихся данных посредством связей.

Зададим каждую сущность набором атрибутов:

ИНСТИТУТ (название, подчиненность, адрес, телефон, ФИО ректора)

ФАКУЛЬТЕТ (название, код специальности, данные о кафедрах, число выпускников, декан). **СТУДЕНТ** (ФИО, группа, курс, номер текущего семестра, пол).

ПРЕПОДАВАТЕЛЬ (ФИО, должность, звание, кафедра, стаж).

ДИСЦИПЛИНА (название, число часов, код дисциплины, виды занятий, число читаемых семестров, номера текущих семестров, на каких курсах преподается)

В каждом наборе атрибутов, характеризующих сущность, необходимо выбрать ключевые атрибуты, т.е. атрибуты, делающие сущность уникальной. При задании атрибутов ключевые атрибуты подчеркивались.

После выбора сущностей, задания атрибутов и анализа связей можно перейти к проектированию информационной (концептуальной) схемы БД.

Концептуальная схема БД "Успеваемость» представлена на рис.11 (атрибуты сущностей на диаграмме не показаны).

Рассмотрим некоторые **ограничения** в рассматриваемом примере:

1. Значение атрибута "телефон" (сущность - ИНСТИТУТ) задается целым положительным шестизначным числом.
2. Значение атрибута "код факультета" (сущность - ФАКУЛЬТЕТ) лежит в интервале 1-10.
3. Значение атрибута "курс" (сущность - СТУДЕНТ) лежит в интервале 1 - 6
4. Значение атрибута "семестр" (сущность - СТУДЕНТ, ДИСЦИПЛИНА) лежит в интервале 1-12.
5. Значение атрибута "число часов" (сущность - ДИСЦИПЛИНА) лежит в интервале 1-300.
6. Одному студенту может быть приписана только одна группа.
7. Один студент может учиться только на одном факультете.
8. Один студент в семестре сдает от 3 до 5 дисциплин
9. Один студент изучает в семестре от 6 до 12 дисциплин.
10. Одному преподавателю приписывается только одна кафедра.
11. Один студент может передавать одну дисциплину не более трех раз.
12. Ключи: название института, название факультета, ФИО и группа студента, ФИО и кафедра преподавателя, название дисциплины.

3. ФИЗИЧЕСКОЕ ПРОЕКТИРОВАНИЕ

Даталогическое проектирование

Инфологическая модель является исходной для построения *даталогической* модели БД и служит промежуточной моделью для специалистов предметной области (для которой создается банк данных (БНД)) и администратора БД в процессе проектирования и разработки конкретной БД.

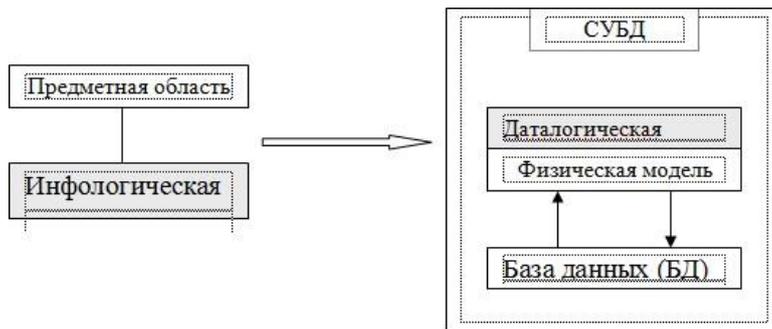


Рис. 13. Структура проектирования БД

Под *даталогической* понимается модель, отражающая логические взаимосвязи между элементами данных безотносительно их содержания и физической организации. При этом даталогическая модель разрабатывается с учетом конкретной реализации СУБД, также с учетом специфики конкретной предметной области на основе ее инфологической модели.

Инфологическая модель предметной области строится первой. Предварительная инфологическая модель строится еще на предпроектной стадии и затем уточняется на более поздних стадиях проектирования баз данных. Затем на ее основе строятся концептуальная (логическая), внутренняя (физическая) и внешняя модели.

Конечным результатом даталогического проектирования является описание логической структуры базы данных на языке программирования. Однако если проектирование выполняется «вручную», то для большей наглядности сначала строится схематическое графическое изображение структуры базы данных. При этом должно быть обеспечено однозначное соответствие между конструкциями языка описания данных и графическими обозначениями информационных единиц и связей между ними.

Графическое представление используется и при автоматизированном проектировании структуры базы данных как интерфейсное средство общения с проектировщиком, и при документировании проекта.

Спроектировать логическую структуру базы данных означает определить все информационные единицы и связи между ними, задать их имена; если для информационных единиц возможно использование разных типов, то определить их тип. Следует также задать некоторые количественные характеристики, например, длину поля.

Описание даталогической модели

Даталогическая модель представляет собой описание базы данных, выполненное в терминах используемой СУБД. Наиболее часто при разработках баз данных применяют реляционные СУБД. Для СУБД этого типа даталогическая удобно представить в виде набора таблиц специальной формы (табл. 1.4.).

Такая таблица составляется для каждого отношения, используемого в базе данных. Отношения в базе соответствуют классам объектов из инфологической модели. Кроме того, отношения могут представлять некоторые связи предметной области.

Каждой таблице нужно поставить в соответствие ее **ключи**. Схема ключа представляет собой перечисление атрибутов отношения, составляющих ключ.

Различают простые и составные ключи. **Простой ключ** строится на основе одного атрибута.

Составной ключ строится на базе использования нескольких атрибутов.

Ключи принято разделять на **первичные, внешние и вспомогательные**.

Первичный индекс должен быть только один для каждой таблицы. Значения атрибутов, используемых для формирования первичного ключа, должны быть уникальными для каждой записи в таблице. Значения первичного ключа уникально

идентифицируют каждую запись. Не может быть двух записей в таблице с одинаковым значением первичного ключа. Например, в качестве первичного ключа для отношения «Сотрудники» можно выбрать атрибут «Табельный номер», значение которого является уникальным для каждой записи о сотруднике.

Внешние ключи используются для реализации связей типа **1:M** между отношениями. Внешний ключ строится для отношения, находящегося на стороне «много» связи 1:M. Для каждого такого отношения на даталогической модели должен быть показан внешний ключ. Внешний ключ всегда должен иметь соответствующий ему первичный ключ отношения, находящегося на стороне «один» связи типа 1:M.

Для связанных ключа «внешний - первичный» соединяются на схеме даталогической модели ломаной линией. Например, если в таблице «Сотрудники» имеется атрибут

«Шифр категории», то этот атрибут можно использовать в качестве внешнего ключа для связи с таблицей «Категории». В последней таблице должен быть первичный ключ по полю «Шифр категории». Внешних ключей может быть несколько для одной таблицы.

Следует отметить, что первичные и внешние ключи строятся как правило на основе целочисленных атрибутов, а не атрибутов, имеющих строковый или вещественный тип.

Кроме первичных и внешних ключей часто используют **вспомогательные индексы**. Они применяются для реализации связей, получения нужного упорядочения при выводе на экран и создании отчетов и т.д. В каждом случае использования вспомогательного индекса, его необходимость должна быть обоснована. Применение большого количества индексов замедляет работу СУБД, т.к. операции над записями отношения требуют корректировки всех индексов.

II. ЗАДАНИЕ ВЫПОЛНЕНИЯ ПРАКТИЧЕСКОЙ РАБОТЫ

Практическая работа №1 выполняется письменно и в конце занятия сдается на проверку. После проверки будет выставлена оценка.

Выбор задания

Выбрать из таблицы «**Варианты заданий для лаб.работы №1.doc**» вариант задания, соответствующий **номеру студента в списке учебной группы**. Для всех последующих практических работ вариант остается неизменным. Каждому студенту предоставляется свой вариант предметной области (ПО), который он будет использовать в процессе выполнения всех практических работ.

Анализ предметной области.

На основании выбранного варианта привести: название предприятия, цель деятельности предприятия, структура предприятия, информационные потребности пользователей (кратко).

Описание основных сущностей ПО.

Здесь следует привести описание основных сущностей (объектов) ПО. Отбор сущностей производится на основе анализа информационных потребностей. Необходимо привести таблицы описания сущностей (сущностей должно быть не менее 3-х)

Таблица 1.1. Список сущностей предметной области.

N п.п.	Наименование сущности	Краткое описание

Здесь же приводится отбор атрибутов (не менее 5-ти) для каждого экземпляра сущности. Отбираются только те атрибуты сущностей, которые необходимы для формирования ответов на регламентированные и непредусмотренные запросы. Для каждого объекта следует привести таблицы его атрибутов.

Таблица 1.2. Список атрибутов.

N п.п.	Наименование атрибута	Краткое описание

На основе анализа информационных запросов следует выявить связи между сущностями. Для выявленных связей также нужно заполнить таблицу 1.3.

Таблица 1.3. Список связей ПО.

N п.п.	Наименование связи	Сущности, участвующие в связи	Краткое описание

Построение инфологической модели.

На основании ранее выбранного варианта и таблиц 1.1-1.3:

- описать классы объектов (сущностей) и их свойства,
- расставить существующие связи между ними,
- на основании табл. 1.3. в письменной форме обосновать типы связей (1:1, 1:M и т.д.).

При графическом построении ИЛМ следует придерживаться единого масштаба для всей схемы. Все прямоугольники, обозначающие классы объектов, должны быть одного размера. Аналогично, все ромбы с именами связей также должны иметь одинаковый размер.

Построение даталогической модели.

На основании ранее выбранного варианта и таблиц 1.1-1.3, инфологической модели и нормализации БД необходимо:

- провести соответствие ключей для каждой таблицы 1.1-1.3,
- заполнить для каждой таблицы БД форму, согласно табл. 1.4. Таблица 1.4. Структура таблицы для даталогической модели.

N п.п.	Наименование реквизита	Идентификатор	Тип	Длина	Формат изображения	Ограничения и комментарий

III. СОДЕРЖАНИЕ ОТЧЕТА

1. Название и цель работы.
2. Словесный и схематический анализ предметной области (ПО), включая схему структуры предприятия.
3. Заполненные таблицы 1.1 - 1.3. с описанием основных сущностей ПО.
4. Инфологическая модель БД, согласно варианту.
5. Обоснование типов связи в инфологической модели данных.
6. Даталогическая модель БД (табл. 1.4.).

Таблица 1. Варианты заданий для практической работы № 1

№ варианта	Условие
Вариант №1	<p>На основании выбранного варианта выполнить следующее:</p> <ol style="list-style-type: none"> 1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации. <p>БД – успеваемость студентов ВУЗА. БД состоит из следующих таблиц: факультеты, кафедры, учебные группы, студенты, ведомости успеваемости.</p>
	<p>Таблица факультеты имеет следующие атрибуты: название факультета, ФИО декана, номер комнаты, номер корпуса, телефон.</p> <p>Таблица кафедры имеет следующие атрибуты: название кафедры, факультет, ФИО заведующего, номер комнаты, номер корпуса, телефон, кол-во преподавателей.</p> <p>Таблица учебные группы имеет следующие атрибуты: название группы, год поступления, курс обучения, кол-во студентов в группе.</p> <p>Таблица студенты имеет следующие атрибуты: студента, фамилия, имя, отчество, группа, год рождения, пол, адрес, город, телефон.</p> <p>Таблица ведомости успеваемости имеет следующие атрибуты: группа, студент, предмет, оценка.</p>

Вариант №2	<p>На основании выбранного варианта выполнить следующее:</p> <ol style="list-style-type: none"> 1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации. <p>БД – информационная система супермаркета. БД состоит из следующих таблиц: отделы, сотрудники, товары, продажа товаров, должности.</p> <p>Таблица отделы имеет следующие атрибуты: название отдела, кол-во прилавков, кол-во продавцов, номер зала.</p> <p>Таблица сотрудники имеет следующие атрибуты: фамилия, имя, отчество, отдел, год рождения, год поступления на работу, стаж, должность, пол, адрес, город, телефон.</p> <p>Таблица должности имеет следующие атрибуты: название должности, сумма ставки.</p> <p>Таблица товары имеет следующие атрибуты: название товара, отдел, страна производитель, условия хранения, сроки хранения .</p> <p>Таблица продажа товаров имеет следующие атрибуты: сотрудника являющегося продавцом, товара дата, время, кол-во, цена, сумма.</p>
Вариант №3	<p>На основании выбранного варианта выполнить следующее:</p> <ol style="list-style-type: none"> 1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации. <p>БД – информационная система военного округа. БД состоит из следующих таблиц: места дислокации, вид войск, части, роты, личный состав.</p> <p>Таблица вид войск имеет следующие атрибуты: название.</p> <p>Таблица места дислокации имеет следующие атрибуты: страна, город, адрес, занимаемая площадь.</p> <p>Таблица части имеет следующие атрибуты: номер части, место дислокации, вид войск, кол-во рот</p>
	<p>Таблица роты имеет следующие атрибуты: название роты, кол-во служащих.</p> <p>Таблица личный состав имеет следующие атрибуты: фамилия, рота, должность, год рождения, год поступления на службу, выслуга лет, награды, участие в военных мероприятиях.</p>

<p>Вариант №4</p>	<p>На основании выбранного варианта выполнить следующее:</p> <ol style="list-style-type: none"> 1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации. <p>БД – информационная система библиотеки. БД состоит из следующих таблиц: библиотеки, фонд библиотеки, тип литературы, сотрудники, пополнение фонда.</p> <p>Таблица библиотеки имеет следующие атрибуты: название, адрес, город.</p> <p>Таблица фонд библиотеки имеет следующие атрибуты: название фонда, библиотека, кол-во книг, кол-во журналов, кол-во газет, кол-во сборников, кол-во диссертаций, кол-во рефератов.</p> <p>Таблица тип литературы имеет следующие атрибуты: название типа. Таблица сотрудники имеет следующие атрибуты: фамилия сотрудника, библиотека, должность, год рождения, год поступления на работу, образование, зарплата.</p> <p>Таблица пополнение фонда имеет следующие атрибуты: фонд, сотрудник, дата, название источника литературы, тип литературы, издательство, дата издания, кол-во экземпляров.</p>
<p>Вариант №5</p>	<p>На основании выбранного варианта выполнить следующее:</p> <ol style="list-style-type: none"> 1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации. <p>БД – информационная система туристического агентства. БД состоит из следующих таблиц: пансионаты, туры, клиенты, путевки, вид жилья. Таблица пансионаты имеет следующие атрибуты: название пансионата, адрес, город, страна, телефон, описание территории, кол-во комнат, наличие бассейна, наличие медицинских услуг, наличие спа-салона, уровень пансионата, расстояние до моря.</p> <p>Таблица вид жилья имеет следующие атрибуты: название (дом, бунгало, квартира, 1-я комната, 2-я комната и т.д.), категория жилья (люкс, полулюкс, и т.д.), пансионат, описание условий проживания, цена за номер в сутки.</p> <p>Таблица туры имеет следующие атрибуты: название тура (Европа, средняя Азия, тибет и т.д.), вид транспорта, категория жилья на ночь (гостиница, отель, палатка и т.д.), вид питания (одноразовое, двухразовое, трехразовое, завтраки), цена тура в сутки.</p>

	<p>Таблица клиенты имеет следующие атрибуты: фамилия, имя, отчество, паспортные данные, дата рождения, адрес, город, телефон.</p> <p>Таблица путевки имеет следующие атрибуты: клиент, пансионата, вид жилья, дата заезда, дата отъезда, наличие детей, наличие мед. страховки, кол-во человек, цена, сумма.</p>
Вариант №6	<p>На основании выбранного варианта выполнить следующее:</p> <ol style="list-style-type: none"> 1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации. <p>БД – информационная система автопредприятия города. БД состоит из следующих таблиц: автотранспорт, водители, маршруты, обслуживающий персонал, гаражное хозяйство.</p> <p>Таблица автотранспорт имеет следующие атрибуты: название транспорта (автобусы, такси, маршрутные такси, прочий легковой транспорт, грузовой транспорт и т.д.), кол-во наработки, пробег, кол-во ремонтов, характеристика.</p> <p>Таблица маршруты имеет следующие атрибуты: название маршрута, транспорт, водитель, график работы.</p> <p>Таблица водители имеет следующие атрибуты: фамилия, имя, отчество, год рождения, год поступления на работу, стаж, должность, пол, адрес, город, телефон.</p> <p>Таблица обслуживающий персонал имеет следующие атрибуты:</p> <p>должность (техники, сварщики, слесари, сборщики и др.), фамилия, имя, отчество, год рождения, год поступления на работу, стаж, пол, адрес, город, телефон.</p> <p>Таблица гаражное хозяйство имеет следующие атрибуты: название гаража, транспорт на ремонте, вид ремонта, дата поступления, дата выдачи после ремонта, результат ремонта, персонал, производящего ремонт.</p>

Вариант №7	<p>На основании выбранного варианта выполнить следующее:</p> <ol style="list-style-type: none"> 1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации. <p>БД – информационная система поликлиники. БД состоит из следующих таблиц: врачи, пациенты, история болезней, отделения, обслуживающий персонал.</p> <p>Таблица отделения имеет следующие атрибуты: название отделения (хирургия, терапия, неврология и т.д.), этаж, номера комнат, ФИО заведующего.</p> <p>Таблица врачи имеет следующие атрибуты: фамилия, имя, отчество, должность, стаж работы, научное звание, адрес, номер отделения, в котором он работает.</p>
-------------------	--

	<p>Таблица пациенты имеет следующие атрибуты: фамилия, имя, отчество, адрес, город, возраст, пол.</p> <p>Таблица диагнозы имеет следующие атрибуты: название диагноза, признаки болезни, период лечения, назначения.</p> <p>Таблица история болезни имеет следующие атрибуты: пациент, врач, диагноз, лечение, дата заболевания, дата вылечения, вид лечения (амбулаторное, стационарное).</p>
--	---

Вариант №8	<p>На основании выбранного варианта выполнить следующее:</p> <ol style="list-style-type: none"> 1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации. <p>БД – информационная система больницы. БД состоит из следующих таблиц: врачи, пациенты, история болезней, операции, лист лечения. Таблица врачи имеет следующие атрибуты: фамилия, имя, отчество, должность, стаж работы, научное звание, адрес.</p> <p>Таблица пациенты имеет следующие атрибуты: фамилия, имя, отчество, адрес, город, возраст, пол.</p> <p>Таблица история болезни имеет следующие атрибуты: пациента врач, диагноз, дата заболевания, дата вылечивания, вид лечения (амбулаторное, стационарное), код операции.</p> <p>Таблица лист лечения имеет следующие атрибуты: дата лечения, история болезни, лекарства, температура, давление, состояние больного (тяжелое, среднее, и т.д.).</p> <p>Таблица операции имеет следующие атрибуты: описание операции (удаление аппендицита, пластическая операция и т.д.), врач, дата операции, пациент, результат операции.</p>
Вариант №9	<p>На основании выбранного варианта выполнить следующее:</p> <ol style="list-style-type: none"> 1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации. <p>БД – информационная система библиотек города. БД состоит из следующих таблиц: библиотеки, читальные залы, литература, читатели, выдача лит-ры.</p> <p>Таблица библиотеки имеет следующие атрибуты: название, адрес, город.</p> <p>Таблица читальные залы имеет следующие атрибуты: название читального зала, библиотека, кол-во единиц лит-ры, кол-во посадочных мест, время работы, этаж, кол-во сотрудников.</p> <p>Таблица читатели имеет следующие атрибуты: фамилия, имя, отчество, категория читателя, место работы или обучения, возраст, дата регистрации в библиотеке.</p>
	<p>Таблица литература имеет следующие атрибуты: название, категория литературы, авторы, издательство, год издательства, кол-во страниц, читальный зал.</p> <p>Таблица выдача литературы имеет следующие атрибуты: читатель, литература, дата выдачи, срок выдачи, вид выдачи, наличие залога.</p>

Вариант №10	<p>На основании выбранного варианта выполнить следующее:</p> <ol style="list-style-type: none"> 1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации. <p>БД – информационная система автосалона. БД состоит из следующих таблиц: автомобили, марка автомобиля, сотрудники, продажа автомобилей, покупатель.</p> <p>Таблица марка автомобиля имеет следующие атрибуты: название марки, страна производитель, завод производитель, адрес.</p> <p>Таблица автомобиля имеет следующие атрибуты: название автомобиля, марка, год производства, цвет, категория, цена.</p> <p>Таблица покупателя имеет следующие атрибуты: фамилия, имя, отчество, паспортные данные, адрес, город, возраст, пол.</p> <p>Таблица сотрудника имеет следующие атрибуты: фамилия, имя, отчество, стаж, зарплата.</p> <p>Таблица продажа автомобилей имеет следующие атрибуты: дата, сотрудник, автомобиль, покупатель.</p>
Вариант №11	<p>На основании выбранного варианта выполнить следующее:</p> <ol style="list-style-type: none"> 1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации. <p>БД – успеваемость студентов кафедры. БД состоит из следующих таблиц: кафедры, дисциплины, преподаватели, студенты, ведомости успеваемости.</p> <p>Таблица кафедра имеет следующие атрибуты: название кафедры, факультет, ФИО заведующего, номер комнаты, номер корпуса, телефон, кол-во преподавателей.</p> <p>Таблица преподаватели имеет следующие атрибуты: фамилия, имя, отчество, кафедра, год рождения, год поступления на работу, стаж, должность, пол, адрес, город, телефон.</p> <p>Таблица студенты имеет следующие атрибуты: фамилия, имя, отчество, кафедра, год рождения, пол, адрес, город, телефон.</p> <p>Таблица дисциплины имеет следующие атрибуты: название дисциплины, кафедра, читаемой эту дисциплину, кол-во часов, вид итогового контроля.</p> <p>Таблица ведомости успеваемости имеет следующие атрибуты: преподаватель, дисциплина, студент, оценка.</p>

Практическая работа № 2 Сбор и анализ информации

Цель работы: выработать практические навыки моделирования предметной области и построения различных видов модели баз данных

Нормализация, функциональные и многозначные зависимости

Нормализация – это разбиение таблицы на две или более, обладающих лучшими свойствами при включении, изменении и удалении данных. Окончательная цель нормализации сводится к получению такого проекта базы данных, в котором каждый факт появляется лишь в одном месте, т.е. исключена избыточность информации. Это делается не столько с целью экономии памяти, сколько для исключения возможной противоречивости хранимых данных.

Каждая таблица в реляционной БД удовлетворяет условию, в соответствии с которым в позиции на пересечении каждой строки и столбца таблицы всегда находится единственное атомарное значение, и никогда не может быть множества таких значений. Любая таблица, удовлетворяющая этому условию, называется нормализованной. Фактически, ненормализованные таблицы, т.е. таблицы, содержащие повторяющиеся группы, даже не допускаются в реляционной БД.

Всякая нормализованная таблица автоматически считается таблицей в первой нормальной форме, сокращенно 1НФ. Таким образом, строго говоря, "нормализованная" и "находящаяся в 1НФ" означают одно и то же. Однако на практике термин "нормализованная" часто используется в более узком смысле – "полностью нормализованная", который означает, что в проекте не нарушаются никакие принципы нормализации.

Теперь в дополнение к 1НФ можно определить дальнейшие уровни нормализации – вторую нормальную форму (2НФ), третью нормальную форму (3НФ) и т.д.

По существу, таблица находится в 2НФ, если она находится в 1НФ и удовлетворяет, кроме того, некоторому дополнительному условию, суть которого будет рассмотрена ниже. Таблица находится в 3НФ, если она находится в 2НФ и, помимо этого, удовлетворяет еще другому дополнительному условию и т.д.

Таким образом, каждая нормальная форма является в некотором смысле более ограниченной, но и более желательной, чем предшествующая. Это связано с тем, что "(N+1)-я нормальная форма" не обладает некоторыми непривлекательными особенностями, свойственным "N-й нормальной форме". Общий смысл дополнительного условия, налагаемого на (N+1)-ю нормальную форму по отношению к N-й нормальной форме, состоит в исключении этих непривлекательных особенностей.

За время развития технологии проектирования реляционных БД были выделены следующие нормальные формы:

- первая нормальная форма (1NF);
- вторая нормальная форма (2NF);
- третья нормальная форма (3NF);
- нормальная форма Бойса-Кодда (BCNF);
- четвертая нормальная форма (4NF);
- пятая нормальная форма, или нормальная форма проекции-соединения (5NF).

Обычно на практике применение находят только первые три нормальные формы.

Теория нормализации основывается на наличии той или иной зависимости между полями таблицы. Определены два вида таких зависимостей: функциональные и многозначные.

Определение. Функциональная зависимость. Поле В таблицы функционально зависит от поля А той же таблицы в том и только в том случае, когда в любой заданный момент времени для каждого из различных значений поля А обязательно существует только одно из различных значений поля В. Отметим, что здесь допускается, что поля А и В могут быть составными.

Другими словами, в отношении R атрибут Y функционально зависит от атрибута X в том и только в том случае, если каждому значению X соответствует одно значение Y.

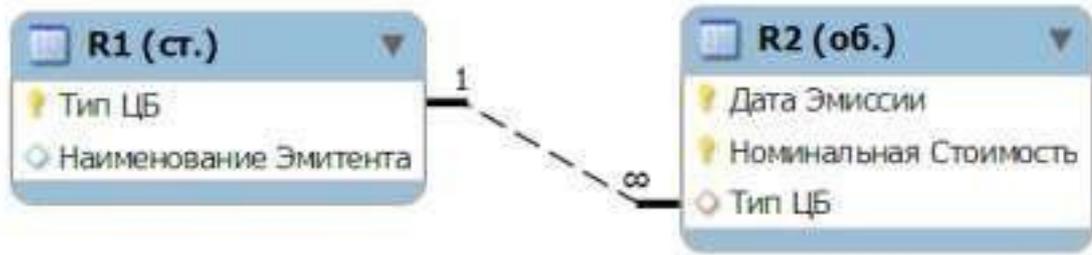


Рисунок 2.1. Схема «Таблица – связь» **Задание для практической работы**

Дан фрагмент отношения (таблицы). Предполагается, что функциональные зависимости, имеющиеся во фрагменте, распространяются на все отношение (таблицу). Для вашего варианта:

1. Определить первичный ключ отношения и все функциональные зависимости отношения.
2. Привести отношение к ЗНФ, указать первичные и внешние ключи полученных отношений, построить схему "Таблица - Связь".

Индивидуальные задания к практической работе

Вариант 1.

Область	Тип	C ₂ N	Количество
А	С	27	5
А	С	27	6
Б	Д	26	7
Б	Д	27	7

Вариант 2.

Наименование	Свойство	Артикул	Количество
А	С	27	9
А	Д	27	9
Б	С	27	9
Б	Д	25	9

Вариант 3.

Тип	Характеристика	Группа	Количество
А	С	13	100
А	Д	13	100
Б	С	13	200
Б	Д	13	100

Вариант 4.

Наименование Эмитента	Тип ЦБ	Дата Эмиссии
ОАО —КрАЗ	акция привилегированная	20.06.1999
ОАО —КрАЗ	акция обыкновенная	20.06.1999
ЗАО —Агат	акция привилегированная	23.06.1999
ТОО —Искра	акция привилегированная	20.06.1999

Практическая работа № 3. Сбор и анализ информации

Цель работы: выработать практические навыки моделирования предметной области и построения нормальных форм баз данных

Нормализация, функциональные и многозначные зависимости

Нормализация – это разбиение таблицы на две или более, обладающих лучшими свойствами при включении, изменении и удалении данных. Окончательная цель нормализации сводится к получению такого проекта базы данных, в котором каждый факт появляется лишь в одном месте, т.е. исключена избыточность информации. Это делается не столько с целью экономии памяти, сколько для исключения возможной противоречивости хранимых данных.

Каждая таблица в реляционной БД удовлетворяет условию, в соответствии с которым в позиции на пересечении каждой строки и столбца таблицы всегда находится единственное атомарное значение, и никогда не может быть множества таких значений. Любая таблица, удовлетворяющая этому условию, называется нормализованной. Фактически, ненормализованные таблицы, т.е. таблицы, содержащие повторяющиеся группы, даже не допускаются в реляционной БД.

Всякая нормализованная таблица автоматически считается таблицей в первой нормальной форме, сокращенно 1НФ. Таким образом, строго говоря, "нормализованная" и "находящаяся в 1НФ" означают одно и то же. Однако на практике термин "нормализованная" часто используется в более узком смысле – "полностью нормализованная", который означает, что в проекте не нарушаются никакие принципы нормализации.

Теперь в дополнение к 1НФ можно определить дальнейшие уровни нормализации – вторую нормальную форму (2НФ), третью нормальную форму (3НФ) и т.д.

По существу, таблица находится в 2НФ, если она находится в 1НФ и удовлетворяет, кроме того, некоторому дополнительному условию, суть которого будет рассмотрена ниже. Таблица находится в 3НФ, если она находится в 2НФ и, помимо этого, удовлетворяет еще другому дополнительному условию и т.д.

Таким образом, каждая нормальная форма является в некотором смысле более ограниченной, но и более желательной, чем предшествующая. Это связано с тем, что "(N+1)-я нормальная форма" не обладает некоторыми непривлекательными особенностями, свойственным "N-й нормальной форме". Общий смысл дополнительного условия, налагаемого на (N+1)-ю нормальную форму по отношению к N-й нормальной форме, состоит в исключении этих непривлекательных особенностей.

За время развития технологии проектирования реляционных БД были выделены следующие нормальные формы: первая нормальная форма (1NF); вторая нормальная форма (2NF); третья нормальная форма (3NF); нормальная форма Бойса-Кодда (BCNF); четвертая нормальная форма (4NF); пятая нормальная форма, или нормальная форма проекции-соединения (5NF).

Обычно на практике применение находят только первые три нормальные формы.

Задание для практической работы

Задание 1. Приведение отношения к первой нормальной форме

1. Открыть файл journal.xls. Лист Ученик. Здесь собрана информация об ученике (см. лист Ученик). Анализ таблицы показывает, что в столбце «Родители» и «Место работы» указаны по 2 значения. Кроме того, в столбце адрес слишком много данных. Возможна ситуация, когда РОНО будет интересоваться район проживания, при этом конкретная квартира – неинтересна. Из-за неоднозначности значений придется привести таблицу к первой нормальной форме.

2. В таблице скопировать строки и сделать так, чтобы в каждой строке был один родитель. 3. Одновременно нужно создать еще один столбец «Контекст адреса»

4. Получим отношение в первой нормальной форме.

5. Выделим в ней первичный ключ: строки отличаются друг от друга столбцами - «№ билета» и «Фамилии родителей». Зальем желтым цветом.

Задание 2. Приведение отношения ко второй нормальной форме

1. Открыть файл journal.xls. Лист Ученик.
2. Сделайте еще одну таблицу. Будет 2 таблицы. В одной будет все, что определяется только столбцом «№ билета», а в другой все, что определяется 2-мя столбцами.
3. Первую таблицу назовем «Личные данные ученика». Первичный ключ «№ билета»
4. Вторую таблицу назовем «Родители». Первичный ключ «№ билета» и «Фамилии родителей».
5. Отношение «Родители» конечно, больше преобразованиям не подлежит.

Задание 3. Приведение отношения к третьей нормальной форме

1. Открыть файл journal.xls. Лист Ученик.
2. Выделим из отношения «Личные данные ученика» таблицу «Класс».
3. Останется таблица «Личные данные». Однако в ней нужно оставить столбец «Класс».
4. В таблице «Класс» ключом является столбец «Класс».

ЗАМЕЧАНИЕ. Название специализации встречается многократно для разных классов. Со временем формулировка может измениться. Поэтому целесообразно сделать справочную таблицу «Справка» и сделать столбец «Код специализации».

Задание 4. Нормализация отношения «Преподаватель».

1. Открыть файл journal.xls. Лист «Преподаватель».
2. Отношение «Преподаватель» привести к первой нормальной форме. Для этого скопировать строки, исправить данные так, чтобы в каждой строке был только один класс и предмет.
3. Выделим первичный ключ. Это столбцы «Фамилия преподавателя», «Класс» и «Предмет».
4. Анализируем и видим, что есть атрибуты, которые зависят только от фамилии преподавателя. Значит, нужно привести отношение ко второй нормальной форме.
5. Получим две таблицы. Первую назовем «Преподаватель – класс – предмет». Она уже находится в третьей нормальной форме, т.к. здесь атрибуты ключевые и все не зависят друг от друга. Вернее здесь наблюдаются связи многие-ко-многим. Отставим пока.
6. Рассмотрим вторую таблицу «Личные данные преподавателя». Первичный ключ «Фамилия». Поскольку данные преподавателя могут меняться (замуж вышла), то целесообразнее сделать столбец «Код преподавателя» и сделать этот столбец ключевым. Но тогда в отношении «Класс» нужно вместо поля «Классный руководитель» поставить «Код классного руководителя».
7. Посмотрим на столбец «Классное руководство».
8. Такая информация у нас уже есть, дублировать ее не надо. Поэтому просто вычеркнем этот столбец.

Задание 5. Приведение отношения к четвертой нормальной форме

1. Открыть файл journal.xls. Лист «Преподаватель». Отношение «Преподаватель – класс – предмет».
2. Разделим его на 2 таблицы «Преподаватель – предмет» и «Преподаватель – класс».
3. Видим, что таблица «Преподаватель – класс» осталась «многие – ко – многим». Оставим ее в покое.

Задание 6. Конечный результат

1. Откройте файл journal_ready.xls.
2. Сравните эти таблицы с теми, которые получились у вас.

Задание 7. Постройте сетевую модель по примеру в программе ERModeler

На листе «Конечный результат» приведена реляционная модель той части задачи школьного журнала, нормализацией которой мы занимались на предыдущих страницах. Приведенную модель сложно считать наглядной и удобной для восприятия, однако именно такой вид представления наиболее удобен для проведения дальнейших этапов проектирования. Понимая это, условимся в дальнейшем проводить этап инфологического проектирования с учетом требований к реляционным моделям.

Практическая работа № 4. Сбор и анализ информации

Цель работы: приобретение практических навыков анализа предметной области, информационных задач и построения концептуальной модели базы данных.

Проектирование базы данных (БД)

Проектирование базы данных (БД) – одна из наиболее сложных и ответственных задач, связанных с созданием информационной системы (ИС). В результате её решения должны быть определены содержание БД, эффективный для всех её будущих пользователей способ организации данных и инструментальные средства управления данными.

Основная цель процесса проектирования БД состоит в получении такого проекта, который удовлетворяет следующим требованиям:

- корректность схемы БД, т.е. база должна быть гомоморфным образом моделируемой предметной области (ПО), где каждому объекту предметной области соответствуют данные в памяти ЭВМ, а каждому процессу – адекватные процедуры обработки данных;
- обеспечение ограничений ;
- эффективность функционирования ;
- защита данных (от аппаратных и программных сбоев и несанкционированного доступа);
- простота и удобство эксплуатации;
- гибкость, т.е. возможность развития и адаптации к изменениям предметной области и/или требований пользователей.

Внимание! Базы данных всегда проектируются под конкретное назначение системы.

Техника проектирования баз данных может измениться в целом и в деталях в зависимости от назначения системы. Например, следует различать проектирование систем складирования данных и проектирование так называемых OLTP-систем, ориентированных на оперативную обработку транзакций. В данном учебном курсе рассматривается проектирование баз данных в основном для OLTP-систем. Именно на таких системах исторически сложилась техника проектирования баз данных.

Этапы проектирования базы данных

Процесс проектирования включает в себя следующие этапы:

Концептуальное проектирование – это процедура конструирования информационной модели, не зависящей от каких-либо физических условий реализации.

Логическое проектирование – это процесс конструирования информационной модели на основе существующих моделей данных, не зависимо от используемой СУБД и других условий физической реализации.

Физическое проектирование – это процедура создания описания конкретной реализации БД с описанием структуры хранения данных, методов доступа к данным.

Концептуальное проектирование

Основными задачами концептуального проектирования являются определение предметной области системы и формирование взгляда на ПО с позиций сообщества будущих пользователей БД, т.е. инфологической модели ПО.

Концептуальная модель ПО представляет собой описание структуры и динамики ПО, характера информационных потребностей пользователей в терминах, понятных пользователю и не зависящих от реализации БД. Это описание выражается в терминах не отдельных объектов ПО и связей между ними, а их типов, связанных с ними ограничений

целостности и тех процессов, которые приводят к переходу предметной области из одного состояния в другое.

Рассмотрим основные подходы к созданию концептуальной модели предметной области.

1. Функциональный подход к проектированию БД

Этот метод реализует принцип "от задач" и применяется тогда, когда известны функции некоторой группы лиц и/или комплекса задач, для обслуживания информационных потребностей которых создаётся рассматриваемая БД.

2. Предметный подход к проектированию БД

Предметный подход к проектированию БД применяется в тех случаях, когда у разработчиков есть чёткое представление о самой ПО и о том, какую именно информацию они хотели бы хранить в БД, а структура запросов не определена или определена не полностью. Тогда основное внимание уделяется исследованию ПО и наиболее адекватному её отображению в БД с учётом самого широкого спектра информационных запросов к ней.

3. Проектирование с использованием метода "сущность-связь"

Метод "сущность-связь" (entity-relation, ER-method) является комбинацией двух предыдущих и обладает достоинствами обоих. Этап инфологического проектирования начинается с моделирования ПО. Проектировщик разбивает её на ряд локальных областей, каждая из которых (в идеале) включает в себя информацию, достаточную для обеспечения запросов отдельной группы будущих пользователей или решения отдельной задачи (подзадачи). Каждое локальное представление моделируется отдельно, затем они объединяются.

Выбор локального представления зависит от масштабов ПО. Обычно она разбивается на локальные области таким образом, чтобы каждая из них соответствовала отдельному внешнему приложению и содержала 6-7 сущностей.

Сущность – это объект, о котором в системе будет накапливаться информация. Сущности бывают как физически существующие (например, СОТРУДНИК или АВТОМОБИЛЬ), так и абстрактные (например, ЭКЗАМЕН или ДИАГНОЗ). Для сущностей различают тип сущности и экземпляр. Тип характеризуется именем и списком свойств, а экземпляр – конкретными значениями свойств. Типы сущностей можно классифицировать как сильные и слабые. Сильные сущности существуют сами по себе, а существование слабых сущностей зависит от существования сильных. Например, читатель библиотеки – сильная сущность, а абонемент этого читателя – слабая, которая зависит от наличия соответствующего читателя. Слабые сущности называют подчинёнными (дочерними), а сильные – базовыми (основными, родительскими).

Для каждой сущности выбираются свойства (атрибуты). Различают:

- Идентифицирующие и описательные атрибуты. Идентифицирующие атрибуты имеют уникальное значение для сущностей данного типа и являются потенциальными ключами. Они позволяют однозначно распознавать экземпляры сущности. Из потенциальных ключей выбирается один первичный ключ (ПК). В качестве ПК обычно выбирается потенциальный ключ, по которому чаще происходит обращение к экземплярам записи. Кроме того, ПК должен включать в свой состав минимально необходимое для идентификации количество атрибутов. Остальные атрибуты называются описательными и включают в себе интересные свойства сущности.

- Составные и простые атрибуты. Простой атрибут состоит из одного компонента, его значение неделимо. Составной атрибут является комбинацией нескольких компонентов, возможно, принадлежащих разным типам данных (например, ФИО или адрес). Решение о том, использовать составной атрибут или разбивать его на компоненты, зависит от характера его обработки и формата пользовательского представления этого атрибута.

- Однозначные и многозначные атрибуты (могут иметь соответственно одно или много значений для каждого экземпляра сущности).

- Основные и производные атрибуты. Значение основного атрибута не зависит от других атрибутов. Значение производного атрибута вычисляется на основе значений других атрибутов (например, возраст студента вычисляется на основе даты его рождения и текущей даты).

Спецификация атрибута состоит из его названия, указания типа данных и описания ограничений целостности – множества значений (или домена), которые может принимать данный атрибут. Далее осуществляется спецификация связей внутри локального представления. Связи могут иметь различный содержательный смысл (семантику). Различают связи типа "сущность-сущность", "сущность-атрибут" и "атрибут-атрибут" для отношений между атрибутами, которые характеризуют одну и ту же сущность или одну и ту же связь типа "сущность-сущность". Каждая связь характеризуется именем, обязательностью, типом и степенью. Различают факультативные и обязательные связи. Если вновь порождённый объект одного типа оказывается по необходимости связанным с объектом другого типа, то между этими типами объектов существует обязательная связь (обозначается двойной линией). Иначе связь является факультативной. По типу различают множественные связи "один к одному" (1:1), "один ко многим" (1:N) и "многие ко многим" (M:N). Степень связи определяется количеством сущностей, которые охвачены данной связью. Пример бинарной связи – связь между отделом и сотрудниками, которые в нём работают. Примером тернарной связи является связь типа экзамен между сущностями ДИСЦИПЛИНА, СТУДЕНТ, ПРЕПОДАВАТЕЛЬ. Из последнего примера видно, что связь также может иметь атрибуты (в данном случае это Дата проведения и Оценка). Пример ER–диаграммы с указанием сущностей, их атрибутов и связей приведен на рис. 1.



Рисунок 1. Пример ER–диаграммы с однозначными и многозначными атрибутами

Пример проектирования реляционной базы данных

В качестве примера возьмем базу данных компании, которая занимается издательской деятельностью. База данных создаётся для информационного обслуживания редакторов, менеджеров и других сотрудников компании. БД должна содержать данные о сотрудниках компании, книгах, авторах, финансовом состоянии компании и предоставлять возможность получать разнообразные отчёты. В соответствии с предметной областью система строится с учётом следующих особенностей:

- каждая книга издаётся в рамках контракта;
- книга может быть написана несколькими авторами;
- контракт подписывается одним менеджером и всеми авторами книги; □ каждый автор может написать несколько книг (по разным контрактам);
- порядок, в котором авторы указаны на обложке, влияет на размер гонорара;
- если сотрудник является редактором, то он может работать одновременно над несколькими книгами;

- у каждой книги может быть несколько редакторов, один из них – ответственный редактор;

- каждый заказ оформляется на одного заказчика;
- в заказе на покупку может быть перечислено несколько книг.

Выделим базовые сущности этой предметной области:

1. Сотрудники компании. Атрибуты сотрудников – ФИО, табельный номер, пол, дата рождения, паспортные данные, ИНН, должность, оклад, домашний адрес и телефоны. Для редакторов необходимо хранить сведения о редактируемых книгах; для менеджеров – сведения о подписанных контрактах.

2. Авторы. Атрибуты авторов – ФИО, ИНН (индивидуальный номер налогоплательщика), паспортные данные, домашний адрес, телефоны. Для авторов необходимо хранить сведения о написанных книгах.

3. Книги. Атрибуты книги – авторы, название, тираж, дата выхода, цена одного экземпляра, общие затраты на издание, авторский гонорар.

4. Контракты будем рассматривать как связь между авторами, книгами и менеджерами. Атрибуты контракта – номер, дата подписания и участники.

5. Для отражения финансового положения компании в системе нужно учитывать заказы на книги. Для заказа необходимо хранить номер заказа, заказчика, адрес заказчика, дату поступления заказа, дату его выполнения, список заказанных книг с указанием количества экземпляров.

ER–диаграмма издательской компании приведена на рис. 2 (базовые сущности на рисунках выделены полужирным шрифтом).

Анализ информационных задач и круга пользователей системы

Система создается для обслуживания следующих групп пользователей:

- администрация (дирекция);
- менеджеры;
- редакторы;
- сотрудники компании, обслуживающие заказы.

Определим границы информационной поддержки пользователей: 1)

Функциональные возможности:

- ведение БД (запись, чтение, модификация, удаление в архив);
- обеспечение логической непротиворечивости БД;
- обеспечение защиты данных от несанкционированного или случайного доступа (определение прав доступа);
- реализация наиболее часто встречающихся запросов в готовом виде;
- предоставление возможности сформировать произвольный запрос на языке манипулирования данными.

2) Готовые запросы:

- получение списка всех текущих проектов (книг, находящихся в печати и в продаже);
- получение списка редакторов, работающих над книгами;
- получение полной информации о книге (проекте);
- получение сведений о конкретном авторе (с перечнем всех книг);
- получение информации о продажах (по одному или по всем проектам); определение общей прибыли от продаж по текущим проектам;
- определение размера гонорара автора по конкретному проекту.

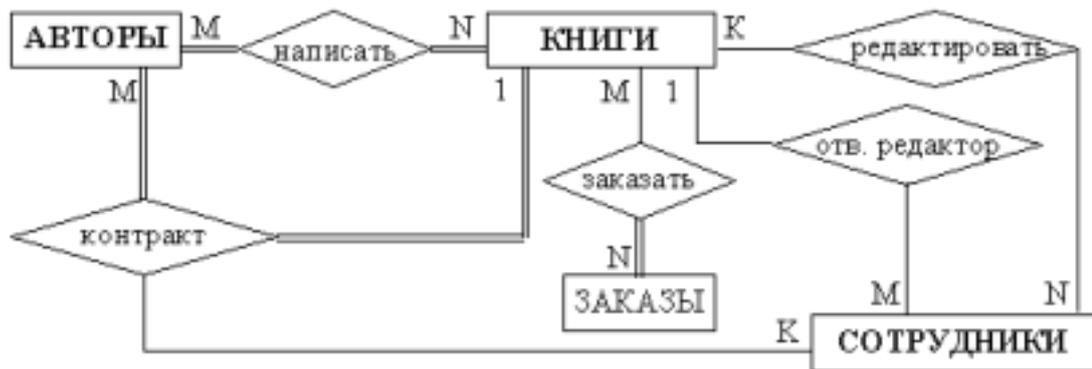


Рисунок 2. ER–диаграмма издательской компании

Задание для практической работы

По заданному описанию предметной области построить концептуальную модель базы данных

Выделите типы сущностей;

- Выделите типы связей и определите для них показатели кардинальности и степень участия сторон;

- Выделите атрибуты и свяжите их типами сущностей и связей; □ Определите потенциальные и первичные ключи сущностей; □ Нарисуйте ER-диаграмму.

- и проанализируйте информационные задачи и группы пользователей.

Индивидуальные задания к практической работе

Вариант 1.

Задача – организация учебного процесса в вузе:

- Студенты: паспортные данные, адрес, дата зачисления, номер приказа, факультет, группа, является ли старостой, кафедра (специализация), изучаемые (изученные) предметы, оценки, задолженности, стипендия.

- Учебные курсы: название, факультет(ы), групп(ы), кафедра, семестр(ы), форма отчётности, число часов.

- Преподаватели: паспортные данные, адрес, телефон, фотография, кафедра, должность, учёная степень, начальник (зав. кафедрой), предмет(ы), число ставок, зарплата.

Вариант 2.

Учет и выдача книг в библиотеке вуза:

- Книги: авторы, название, раздел УДК, раздел (техническая, общественно-политическая и т.п.), место и год издания, издательство, количество страниц, иллюстрированность, цена, дата покупки, номер сопроводительного документа (чек, счёт/накладная), вид издания (книги, учебники, брошюры, периодические издания), инвентарный номер (есть только для книг и некоторых учебников), длительность использования читателями (год, две недели, день), электронная версия книги или ее реферата (отсканированный текст).

- Читатели: номер читательского билета, ФИО, год рождения, адрес, дата записи, вид (студент, аспирант, преподаватель, сотрудник), курс, номер группы, названия взятых книг и даты их выдачи.

Вариант 3.

Отдел кадров некоторой компании.

- Сотрудники: ФИО, паспортные данные, фотография, дом. и моб. телефоны, отдел, комната, раб. телефоны (в т.ч. местный), подчинённые сотрудники, должность, тип(ы) работы, задание(я), проект(ы), размер зарплаты, форма зарплаты (почасовая, фиксированная).

- Отделы: название, комната, телефон(ы), начальник, размер финансирования, число сотрудников.

- Проекты: название, дата начала, дата окончания, размер финансирования, тип финансирования (периодический, разовый), задачи и их исполнители, структура затрат и статьи расходов.

Вариант 4.

Отдел поставок некоторого предприятия.

- Поставщики: название компании, ФИО контактного лица, расчётный счёт в банке, телефон, факс, поставляемое оборудование (материалы), даты поставок (по договорам и реальные), метод и стоимость доставки.

- Сырьё: тип, марка, минимальный запас на складе, время задержки, цена, продукты, при производстве которых используется, потребляемые объёмы (необходимый, реальный, на единицу продукции).

Вариант 5.

Пункт проката видеозаписей (внутренний учёт).

- Видеокассеты: идентификационный номер видеокассеты, тип видеокассет, дата его создания, компания-поставщик, число штук данного типа (общее, в магазине, выдано в настоящее время, выдано всего, выдано в среднем за месяц), общая длительность записей; записи видеокассет: название, длительность, категория, год выпуска и производитель (оригинала).

- Клиенты: ФИО, паспортные данные, адрес, телефон; заказы, т.е. взятые видеокассеты (сейчас и в прошлом): номер, дата выдачи, дата возвращения, общая стоимость заказа.

Вариант 6.

Пункт проката видеозаписей (информация для клиентов).

- Видеокассеты: краткое описание, внешний вид (этикетка), марка (пустой) видеокассеты, цена за единицу прокатного времени (например: 1 день, 3 дня, неделя), есть ли в наличии, общая длительность записей; записи на видеокассете: название, длительность, жанр (категория), тема, год и страна выпуска (оригинала), кинокомпания, описание, актёры, режиссер.

- Заказы: идентификационные номера и названия выданных видеокассет, дата выдачи, дата возвращения (продления), общая стоимость заказа, возвращены ли кассеты заказа.

Вариант 7.

Кинотеатры (информация для зрителей).

- Фильмы: название, описание, жанр (категория), длительность, популярность (рейтинг, число проданных билетов в России и в мире), показывается ли сейчас (сегодня, на текущей неделе), в каких кинотеатрах показывается, цены на билеты (в т.ч. средние).

- Кинотеатры: название, адрес, схема проезда, описание, число мест (в разных залах, если их несколько), акустическая система, широкоэкранный, фильмы и цены на них: детские и взрослые билеты в зависимости от сеанса (дневной, вечерний и т.п.) и от категории мест (передние, задние и т.п.); сеансы показа фильмов (дата и время начала).

Вариант 8.

Ресторан (информация для посетителей).

- Меню: дневное или вечернее, список блюд по категориям.

- Блюда: цена, название, вид кухни, категории (первое, второе и т.п.; мясное, рыбное, салат и т.п.), является ли вегетарианским, компоненты блюда, время приготовления, есть ли в наличии.

- Компоненты блюд: тип (гарнир, соус, мясо и т.п.), калорийность, цена, рецепт, время приготовления, есть ли в наличии, ингредиенты (продукты) и их расходы на порцию.

Вариант 9.

Задача - информационная поддержка деятельности склада.

База данных должна содержать информацию о наименовании товара, его поставщике, количестве, цене товара, конечном сроке реализации, сроке хранения на складе. Торговый склад производит уценку хранящейся продукции. Если продукция хранится на складе дольше 10 месяцев, то она уценивается в 2 раза, а если срок хранения превысил 6 месяцев, но не достиг 10, то в 1,5 раза. Ведомость уценки товаров должна содержать информацию: наименование товара, количество товара(шт.), цена товара до уценки, срок хранения товара, цена товара после уценки, общая стоимость товаров после уценки.

Вариант 10.

Задача – информационная поддержка деятельности адвокатской конторы. БД должна осуществлять:

- ведение списка адвокатов; ведение списка клиентов; ведение архива законченных дел.
- Необходимо предусмотреть:
 - получение списка текущих клиентов для конкретного адвоката;
 - определение эффективности защиты (максимальный срок минус полученный срок) с учётом оправданий, условных сроков и штрафов;
 - определение неэффективности защиты (полученный срок минус минимальный срок);
 - подсчёт суммы гонораров (по отдельным делам) в текущем году;
 - получение для конкретного адвоката списка текущих клиентов, которых он защищал ранее (из архива, с указанием полученных сроков и статей).

Вариант 11.

Задача – информационная поддержка деятельности гостиницы.

БД должна осуществлять:

- ведение списка постояльцев;
- учёт забронированных мест;
- ведение архива выбывших постояльцев за последний год.
- Необходимо предусмотреть:
 - получение списка свободных номеров (по количеству мест и классу);
 - получение списка номеров (мест), освобождающихся сегодня и завтра;
 - выдачу информации по конкретному номеру;
 - автоматизацию выдачи счетов на оплату номера и услуг;
 - получение списка забронированных номеров;
 - проверку наличия брони по имени клиента и/или названию организации

Практическая работа № 5. Приведение БД к нормальной форме 3НФ

Цель работы: освоить основные приемы заполнения и редактирования таблиц; познакомиться с простой сортировкой данных и с поиском записей по образцу; научиться сохранять и загружать базы данных.

Microsoft Office Access

Access является наиболее сложной программой из всех офисных приложений Microsoft Office. Чтобы начать работу с этой программой, вначале необходимо создать структуру базы данных, подробно ее описать, а затем создать различные формы.

ACCESS – это реляционная СУБД. Это означает, что с ее помощью можно работать одновременно с несколькими таблицами базы данных, эти таблицы между собой связаны. Таблицу ACCESS можно связать с данными, хранящимися на другом компьютере. Данные ACCESS очень просто комбинировать с данными EXCEL, WORD и другими программами Office.

Access во многом похож на Excel. Основное различие между таблицей БД и электронной таблицей – в системе адресации: в электронной таблице адресуется каждая ячейка, а в таблице БД – только поля текущей записи.

Таблицы – основные объекты базы данных (БД). В них хранятся данные. Реляционная база данных может иметь много взаимосвязанных таблиц. Сведения по разным вопросам следует хранить в разных таблицах.

Запрос – это средство, с помощью которого извлекается из базы данных информация, отвечающая определенным критериям. Результаты запроса представляют не все записи из таблицы, а только те, которые удовлетворяют запросу.

Формы – Обеспечивают более наглядную работу с таблицами, с помощью форм в базу вводят новые данные или просматривают имеющиеся.

Отчеты – средство представления данных таблиц. Отчеты могут быть оформлены надлежащим образом и распечатаны в том виде, в котором требуется пользователю.

Макросы – набор из одной или более макрокоманд, выполняющих определенные операции (открытие форм, печать отчетов).

Модули - это программы, написанные на языке программирования Visual Basic.

Задание для практической работы

Заполните таблицы по образцу

1. Вызвать программу Access 2007 (Access 2016). 2. В окне системы управления базы данных щелкнуть по значку «Новая база данных». Справа в появившемся окне дать имя новой базе данных «Анкета ГС-31» и щелкнуть по значку папки, находящемуся справа от окна названия . Откроется окно сохранения, найдите свою папку и сохраните в нее новый файл базы данных «Анкета ГС-31» (вместо ГС-31 укажите номер вашей группы). Затем нажмите на кнопку

«Создать».

3. Появится окно «Таблица» (Рисунок 5.1).

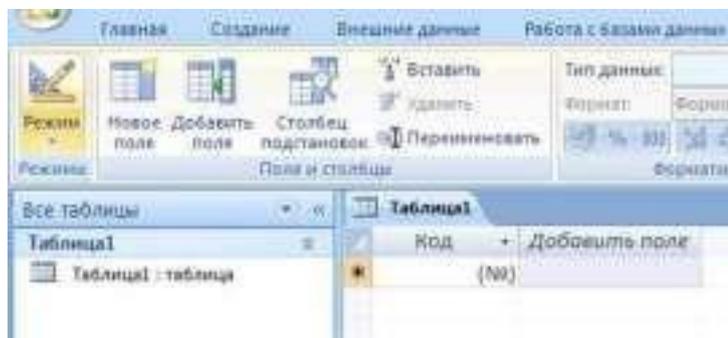


Рисунок 5.1. Окно пустой базы данных

4. В появившемся окне откройте меню команды <Режим> и выберите вариант <Конструктор>  и сохраните будущую таблицу под названием <Ведомость успеваемости>. Появится окно Конструктора.

5. Заполните поля в Конструкторе данными из *рисунка 5.2*. Тип данных можно брать из меню, появившемся при нажатии на кнопку  в ячейке справа.

Обратите внимание: ключевое поле «Счетчик» внесен в таблицу автоматически. Если напротив поля отсутствует значок ключа, то на панели инструментов щелкните по этому значку. 

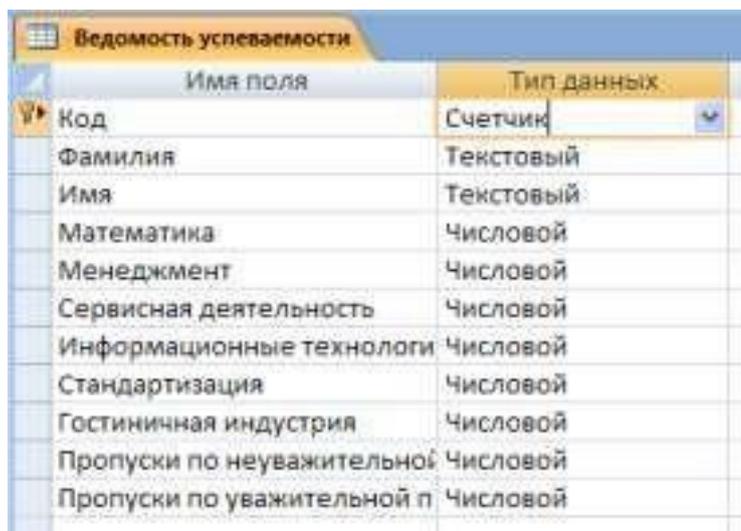


Рисунок 5.2. Создание таблицы через конструктор

6. Перейдите в режим таблицы, щелкнув по кнопке **Режим** на панели инструментов. Введите данные в этом режиме, заполняя клетки таблицы. Значение поля **Код** будет меняться автоматически.

7. Заполните базу данных значениями из *таблицы 5.1*. Напротив каждой фамилии выставьте по всем дисциплинам оценки от 2 до 5.

Таблица 5.1.

Код	Фамилия	Имя	Математика	Менеджмент	Сервисная деятельность	Информационные технологии	Стандартизация	Гостиничная индустрия	Пропуски по неуважительной причине	Пропуски по уважительной причине
1	Иванникова	Анна								
2	Баранова	Ирина								
3	Корнилова	Ольга								
4	Воробьев	Алексей								
5	Воробьев	Олег								
6	Скоркин	Алекс								
7	Володина	Нина								
8	Новоселов	Алексей								
9	Петрова	Елена								
10	Чернова	Кристина								
11	Терещинка	Инна								
12	Истратов	Максим								
13	Бондарь	Ольга								
14	Ревин	Олег								
15	Шарова	Оксана								

Выполните редактирование ячеек:

- Замените фамилию Иванникова на Иванова.

9. Отсортируйте:

а) *фамилии* – по алфавиту (поставьте маркер на любую фамилию в столбце Фамилия и щелкните мышкой по кнопке  на панели инструментов или произведите сортировку с помощью контекстного меню)

б) *имя* – по алфавиту

10. Сохраните текущую таблицу, щелкнув по кнопке «крестик» в правом верхнем углу окна таблицы.

11. Откройте снова свою базу данных.

12. Выполните поиск записей по образцу: *найти студентку по фамилии Володина*.

Для этого установите курсор в поле фамилия, щелкните на кнопке  <Бинокль> на панели инструментов меню Главная и в появившемся диалоговом окне введите в поле <Образец> фамилию *Володина* и щелкните по кнопке <Найти>.

Примечание: Если требуется найти следующую подобную запись, то щелкните мышкой по кнопке <Найти далее>. По окончании работы щелкните по кнопке <Отмена>.

13. Переименуйте поле «Математика» на «Информатика» с помощью контекстного меню.

14. Скройте столбец Пр н/пр., потом отобразите его назад.

15. Войдите в режим *Конструктора* и назначьте полю Пр н/пр и Пр ув/пр. *Маску ввода* 00 «часов». Заполните эти поля данными от 0 до 99.

16. Завершите работу с Access.

Практическая работа № 6 Приведение БД к нормальной форме 3НФ

Цели работы: научиться самостоятельно создавать ключевое поле; закрепить навыки по удалению, добавлению, заполнению и редактированию таблиц; научиться использовать фильтр в таблице.

Microsoft Office Access

Access является наиболее сложной программой из всех офисных приложений Microsoft Office. Чтобы начать работу с этой программой, вначале необходимо создать структуру базы данных, подробно ее описать, а затем создать различные формы.

ACCESS – это реляционная СУБД. Это означает, что с ее помощью можно работать одновременно с несколькими таблицами базы данных, эти таблицы между собой связаны. Таблицу ACCESS можно связать с данными, хранящимися на другом компьютере. Данные ACCESS очень просто комбинировать с данными EXCEL, WORD и другими программами Office.

Access во многом похож на Excel. Основное различие между таблицей БД и электронной таблицей – в системе адресации: в электронной таблице адресуются каждая ячейка, а в таблице БД – только поля текущей записи.

Таблицы – основные объекты базы данных (БД). В них хранятся данные. Реляционная база данных может иметь много взаимосвязанных таблиц. Сведения по разным вопросам следует хранить в разных таблицах.

Запрос – это средство, с помощью которого извлекается из базы данных информация, отвечающая определенным критериям. Результаты запроса представляют не все записи из таблицы, а только те, которые удовлетворяют запросу.

Формы – Обеспечивают более наглядную работу с таблицами, с помощью форм в базу вводят новые данные или просматривают имеющиеся.

Отчеты – средство представления данных таблиц. Отчеты могут быть оформлены надлежащим образом и распечатаны в том виде, в котором требуется пользователю.

Макросы – набор из одной или более макрокоманд, выполняющих определенные операции (открытие форм, печать отчетов)

Модули - это программы, написанные на языке программирования Visual Basic.

Задание для практической работы

Создайте таблицы и схему данных по заданным критериям 1) Откройте учебную базу данных <Анкета ГС-31>.

2) Создайте таблицу <Преподаватели > в *Режиме таблицы*. Для этого в меню Создание выберите кнопку Таблица. В появившейся таблице сделайте следующее:

- Добавьте два поля – Поле 1 и Поле 2, выполнив команду через контекстное меню.
- Переименуйте <Поле 1> на <Предмет>. Для этого поставьте курсор в любую ячейку столбца <Поля 1> и выполните команду *Переименовать столбец* из контекстного меню. Или щелкните два раза по имени поля, удалите старое название и впечатайте новое.
- Переименуйте аналогично <Поле 2> на <Преподаватель>.

3) Сохраните таблицу с именем <Преподаватели>, щелкнув по кнопке <Сохранить> (дискетка на панели инструментов).

4) Перейдите в режим <Конструктор> и удалите строку с ключевым словом Счетчик.

Посмотрите как заданы поля. Сделайте поле <Предмет> ключевым, поместив курсор на имя этого поля и щелкнув по кнопке  - *Ключевое поле*. Тип данных поля задайте *текстовым*.

5) Перейдите в *Режим таблицы* и заполните таблицу <Преподаватели> записями из *Рисунок 6.1*.

предмет	преподаватель
Математика	Бекетова Н.И.
Менеджмент	Казумова Н.С.
Сервисная деятельность	Бессарабова Т.В.
Информационные технологии	Бабич О.А.
Стандартизация	Казарян Г.Г.
Гостиничная индустрия	Казарян Г.Г.

Рисунок 6.1. Таблица «Преподаватели»

6) Закройте таблицу <Преподаватели>, сохранив все изменения.

7) Используя <Шаблон таблиц>, создайте таблицу <Личные данные> студентов с ключевым полем. Для этого:

- Находясь на закладке <Создание> щелкните по кнопке <Шаблоны таблиц>, <Контакты>. Появится таблица уже с готовыми полями.

- Переименуйте предложенные поля на следующие поля: <Код студента>, <Фамилия>, <Имя>, <Город>, <Адрес>, <Телефон>, <Дата рождения>, <Фотография>, <Любимый предмет>, лишние поля удалите.

- Сохраните полученную таблицу под названием <Личные данные>. Ключевое поле задано автоматически.

8) Внесите данные в новую таблицу, заполнив поля <Фамилия>, <Имя>, <Город>, <Адрес>, <Телефон>, <Дата рождения>.

ПРИМЕЧАНИЕ. Поля <Фамилия> и <Имя> можно скопировать из таблицы <Ведомость успеваемости>. В поле <Город> внесите четыре разных города (например, Новороссийск, Геленджик, Анапа, Крымск)

9) Перейдите в режим <Конструктор> и назначьте типы данных: для поля <Телефон> - *числовой*, для поля <Дата рождения> - *дата/время*, для поля <Фотография> – *поле объекта OLE*, для остальных – *текстовый*.

Для поля <Любимый предмет> выполните свойство выбор предмета из списка с помощью *Мастера подстановок*. Для этого в строке <Любимый предмет> в поле *Тип данных – текстовый* щелкните по кнопке и в ниспадающем меню выберите команду <Мастер подстановок>.

- В диалоговом окне <Создание подстановки> поставьте флажок напротив способа <Будет введен фиксированный набор значений> и нажмите <Далее>.

- В следующем окне внесите в столбец все предметы (предметы из таблицы <Преподаватели>), нажмите <Далее>.

- В последнем окне, не изменяя имени столбца нажмите <Готово>.

10) Перейдите в режим таблицы и выберите для каждого студента с помощью кнопки  из списка любимый предмет.

11) Создайте *схему данных*, т.е. установите связи между таблицами.

- Щелкните по кнопке - *Схема данных* на панели инструментов меню <Работа с базами данных>. В окне <Отобразить таблицу> выделите таблицу <Ведомость успеваемости> и щелкните по кнопке <Добавить>. Также добавьте таблицы <Преподаватели> и <Личные данные>. В окне <Схема данных> появится условный вид этих таблиц. Закройте окно <Добавление таблицы>.

- Поставьте мышку на имя поля <Предметы> в таблице <Преподаватели>, и не отпуская кнопку мыши перетащите его на поле <Любимый предмет> таблицы <Личные данные>. Отпустите мышку. Появится диалоговое окно <Связи>, в котором включите значки «Обеспечение целостности данных», «Каскадное обновление связанных полей» и «Каскадное удаление связанных полей». Щелкните по кнопке <Создать>. Появится связь «один-ко-многим».

- Поставьте мышку на имя поля <Код студента> в таблице <Личные данные> и перетащите его, не отпуская мышки, на поле <Код> таблицы <Ведомость успеваемости>. В появившемся окне <Связи> включите значок «Обеспечение целостности данных» и щелкните по кнопке <Создать>. Появится связь «один-к- одному».

- Закройте схему данных, сохранив ее.

12) Произведите фильтрацию данных в таблице <Личные данные> по выделенному.

- Откройте таблицу в режиме таблицы.

- Выберите студентов, проживающих в Новороссийске. Для этого поставьте курсор в одну из первых записей, где есть город Новороссийск и щелкните по кнопке - *Фильтр по выделенному* на панели инструментов. Выберите команду <Равно «Новороссийск» >.

Access отобразит все записи, удовлетворяющие критерию фильтрации. 

- Для отображения всех записей выполните команду <Удалить фильтр> для этого щелкните по соответствующей кнопке на панели инструментов .

13) Закончите работу с базой данных Access.

Практическая работа № 7

Приведение БД к нормальной форме 3НФ

Цели работы: закрепить навыки по редактированию таблиц; познакомиться с основными видами запросов; научиться создавать запросы на выборку различными способами; научиться создавать сложные запросы; научиться создавать перекрестные запросы.

Запрос – это средство, с помощью которого извлекается из базы данных информация, отвечающая определенным критериям. Результаты запроса представляют не все записи из таблицы, а только те, которые удовлетворяют запросу.

Запросы состоят из ряда условий, каждое условие состоит из трех элементов:

1. поле, которое используется для сравнения;
2. оператор, описывающий тип сравнения;
3. величина, с которой должно сравниваться значение поля. Выражения и операторы, применяемые в условиях отбора.

Выражения и операторы	Описание выражений и операторов
Числа	Вводятся без ограничений
Текст	Должен быть заключен в кавычки
Даты	Ограничиваются с двух сторон символами # (например, #01.02.02#)
*; +; -; /; ^	Арифметические операторы, связывающие выражения
<; <=; >; >=; =; <>	Операторы сравнения
And (И); Not (Нет); Or (Или)	Логические операторы
Like	Используется для логики замены в выражениях
In	Для определения, содержится ли элемент данных в списке значений
Between... And...	Для выбора значений из определенного интервала
?	Заменяет один символ (букву или цифру)
*	Заменяет несколько символов

Запросы могут быть простые, сложные перекрестные.

Задание для практической работы

Создайте запросы к вашей базе данных

- 1) Откройте свою учебную базу данных.
- 2) Создайте запрос на выборку студентов, у которых по всем предметам только хорошие оценки с помощью *Мастера запросов*.
 - На панели инструментов выберите команду <Мастер запросов>.
 - В появившемся диалоговом окне выберите <Простой запрос> и щелкните по кнопке <ОК>.
 - В следующем окне выберите таблицу, по которой строится запрос (<Ведомость успеваемости>), и те поля, которые участвуют в запросе. Перенесите их в правую часть окна с помощью кнопки  и нажмите <Далее>., В следующем окне тоже нажмите <Далее>.
 - В другом окне дайте название запроса «Хорошисты» и нажмите <Готово>.
 - Появится таблица <Хорошисты>, в которой отражены фамилии всех студентов и изучаемые предметы.
 - Откройте таблицу «Хорошисты», перейдите в режим <Конструктор>. Здесь в поле <Условия отбора> под каждым предметом поставьте условие >=4 или 4OR5.

Примечание: Галочки в каждом поле означают, что по вашему выбору можно включить или убрать любое поле на выборку.

- Перейдите в режим таблицы, ответив <Да> на вопрос о сохранении запроса. (В таблице должны остаться фамилии «хорошистов»).

3) С помощью <Конструктора запросов> создайте запрос на выборку по таблице <Личные данные>.

- Щелкните по таблице <Личные данные>, зайдите в меню <Создание>, выберите команду <Конструктор запросов >.

- Добавьте нужную таблицу в поле запроса. Выделите её в списке и щелкните по кнопке <Добавить>. Закройте окно <Добавление таблицы>.

- Выберите студентов, чьи фамилии начинаются на букву «В» и которые проживают в Анапе. Для этого:

- добавьте в строку <Поле> два поля <Фамилия> и <Город>;

- в строке <Условия отбора> в первом столбце укажите значение Like —В * l, а во втором столбце с названием <Город> - «Анапа»;

- закройте запрос, сохранив его под названием —ВВВl (у вас должны остаться в списке студенты, проживающие в Анапе). Рисунок 7.1.

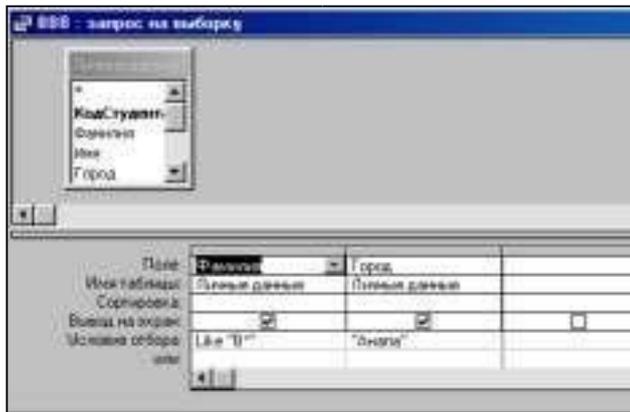


Рисунок 7.1. Запрос на выборку

Индивидуальные задания к практической работе

а) Составьте запрос с названием <Запрос 1> на базе таблицы <Ведомость успеваемости>, в котором будут указаны студенты, имеющие по первым двум предметам оценки не менее «4».

(Выполните запрос или через *Конструктор запросов*, или через *Мастер запросов*)

б) Составьте <Запрос 2> на базе таблицы <Ведомость успеваемости>, в котором будут указаны студенты, имеющие не более 30 часов пропусков по неуважительной причине. Добавьте в этот запрос поле пропуски по уважительной причине в интервале от 30 часов до 45 часов

(используйте оператор *Between... And...*)

в) Составьте <Запрос> на базе таблицы <Личные данные>. Выведите список студентов, которым на данный момент, т.е. на сегодняшнее число, исполнилось уже 17 лет (используйте оператор *Between... And...*)

Примечание: Дата записывается с использованием символа #, например, #01.02.02.#

4) Составьте запрос на базе трех таблиц <Ведомость успеваемости>, <Личные данные> и <Преподаватель>. Выберите студентов, которые проживают в Новороссийске и у которых любимый предмет «Менеджмент». Озаглавьте <Запрос 4>. Используйте <Конструктор запросов>.

- В меню <Создание> выберите <Конструктор запросов>.

- Добавьте все три таблицы в поле запроса. Закройте окно <Добавление таблицы>.

- В первый столбец в строку <Поле> перетащите из первой таблицы с помощью мышки <Фамилия>, из второй таблицы во второй столбец <Город> и из третьей таблицы в третий столбец строки <Поле> - <Предмет> (Рисунок 7.2).

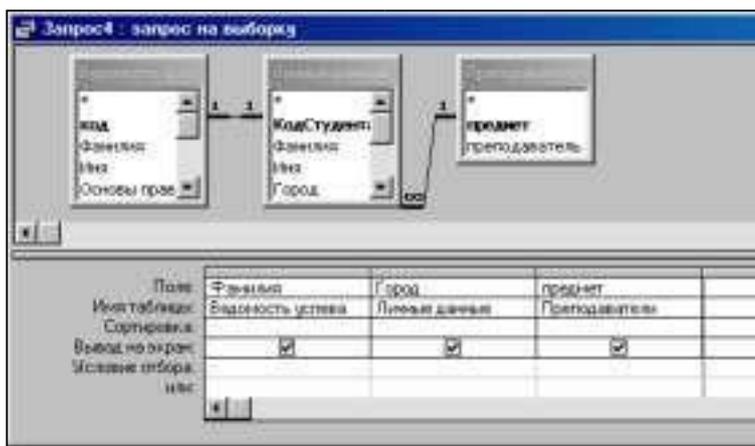


Рисунок 7.2. Запрос на выборку

- В поле <Условия отбора> в столбце <Город> введите город «Новороссийск», в столбце <Предмет> введите «Менеджмент».
- Сохраните запрос под именем <Запрос 4>.
- Откройте запрос и проверьте результат проделанной работы.

5) Выполните запрос на создание новой таблицы, в которой должны быть поля <Фамилия>, <Имя>, <Пропуски по неуважительной причине>, <Город> и <Предмет>.

- В меню <Создание> выберите <Конструктор запросов>.
- Добавьте все три таблицы из списка окна <Добавление таблицы>. Закройте это окно.
- В первую строчку <Поле> из первой таблицы перенесите в первый столбец поля <Фамилия>, во второй <Имя> и в третий <Пропуски по уважительной причине>, в четвертый столбец перетащите поле <Город> из второй таблицы и в последнем столбце будет поле <Предмет> из третьей таблицы.

- Закройте запрос, сохранив его с именем <Запрос 5>.

б) Создайте *перекрестный запрос*.

Допустим, нужно посчитать для ведомости, сколько в группе человек получили по предмету —троек|, —четверок| и —пятерок|. Для этих целей используется *перекрестный запрос*.

- В меню <Создание> выберите <Мастер запросов>.
- В диалоговом окне выберите <Перекрестный запрос>, щелкните по кнопке <ОК>.
- В окне <Создание перекрестных запросов> выделите таблицу <Ведомость успеваемости> и щелкните <Далее>.
- Выберите поля, значения которого будут использоваться в качестве заголовков строк – это <Фамилия> и <Имя>. Щелкните по кнопке <Далее>.
- Выберите поле, значение которого будут использоваться в качестве заголовков столбцов, например <Менеджмент>. Щелкните по кнопке <Далее>.
- Выберите функцию, по которой будут вычисляться значения ячеек на пересечении столбцов и строк (в данном случае Count – количество). Щелкните по кнопке <Далее>.

- Задайте имя запроса <Итог по менеджменту> и щелкните по кнопке <Готово>.

Составьте аналогичные запросы для оценок по трем другим предметам.

7) Предъявите преподавателю все запросы своей базы данных на экране дисплея.

8) Завершите работу с Access.

Практическая работа № 8

Проектирование реляционной схемы базы данных в среде СУБД

Цели работы:

Научиться создавать формы ввода-вывода; научиться создавать кнопочные формы.

Форма – это средство, упрощающее ввод, редактирование и отображение информации, хранящейся в таблицах базы данных. Она представляет собой окно с набором элементов управления.

Форма сама по себе не хранит информацию, она просто обеспечивает удобный способ доступа к информации, хранящейся в одной или нескольких таблицах. Формы по сравнению с обработкой данных в режиме таблицы обладают следующими преимуществами:

- Форма позволяет в каждый момент сфокусировать внимание на отдельной записи;
- Элементы управления на форме можно расположить логичным образом, облегчающим чтение и работу с данными;
- Отдельные элементы управления обладают возможностями облегчить ввод и изменение отдельных данных;
- Некоторые объекты баз данных, такие как рисунки, анимации, звуки и видеоклипы, могут отображаться только в режиме формы, но не в режиме таблицы.

Создание кнопочной формы.

Кнопочное меню представляет собой форму, на которой расположены элементы управления – кнопки с поясняющими надписями. Щелчок на кнопке открывает соответствующую таблицу, запрос, форму или отчет. Меню - удобный инструмент работы с базами данных, и он практически всегда присутствует в базах созданных для предприятий или фирм.

Кнопочное меню создают с помощью Диспетчера кнопочных форм.

Отчет – это гибкое и эффективное средство для организации просмотра и распечатки итоговой информации. В отчете можно получить результаты сложных расчетов, статистических сравнений, а также поместить в него рисунки и диаграммы. Пользователь имеет возможность разработать отчет самостоятельно (в режиме *Конструктора*) или создать отчет с помощью *Мастера*, т.е. полуавтоматически.

Задание для практической работы

Задание 1. Создайте формы к базе данных

- 1) Откройте свою базу данных.
 - 2) Создайте форму с помощью <Мастера форм> на базе таблицы <Ведомость успеваемости>.
 - Откройте таблицу <Ведомость успеваемости>.
 - Выберите закладку <Формы >, щелкните мышкой по кнопке <Другие формы>.
- 
- В появившемся диалоговом окне выберите <Мастер форм>.
 - В поле <Таблицы/Запросы> выберите таблицу <Ведомость успеваемости>, в поле <Доступные поля> выберите поля <Фамилия>, <Имя> и перенесите их стрелкой в поле <Выбранные поля>. Также перенесите поля с названием предметов, щелкните по кнопке <Далее>.
 - Выберите внешний вид формы – Табличный, щелкните по кнопке <Далее>.
 - Выберите требуемый стиль (н-р, Обычная), щелкните по кнопке <Далее>.
 - Задайте имя формы <Успеваемость> и щелкните по кнопке <Готово>. В результате получите форму, в которой можно менять данные и вводить новые значения.
 - Закройте форму.
- 3) Создайте форму на основе таблицы <Преподаватели>.
 - Откройте таблицу <Преподаватели>.



- Выберите закладку <Формы >, щелкните мышкой по кнопке <Другие формы>.
- В появившемся диалоговом окне выберите <Мастер форм> .
- Выберите внешний вид формы - <ленточный>.
- Выберите любой стиль.
- Получите готовую форму. Сохраните ее под именем <Преподаватели>.
- Закройте форму.
- Создайте форму <Личные данные> с помощью инструмента <Пустая форма>
- На вкладке Создание в группе Формы щелкните Пустая форма. 
- Access открывает пустую форму в режиме макета и отображает область Список полей.
 - В области Список полей щелкните знак плюс (+) рядом с таблицей или таблицами, содержащими поля, которые нужно включить в форму.
 - Чтобы добавить поле к форме, дважды щелкните его или перетащите его на форму. Чтобы добавить сразу несколько полей, щелкните их последовательно, удерживая нажатой клавишу CTRL. Затем перетащите выбранные поля на форму.
 - Закройте окно списка полей.
 - Перейдите в режим Конструктора

Примечание 1. Размер окошка для названия поля и для его значений меняются мышкой.

Для этого выделите черный квадратик рамки (рамка станет цветной), установите курсор на границу рамки и с помощью двунаправленной стрелки измените размеры рамки.

Примечание 2. С помощью кнопок панели инструментов Шрифт меняйте соответственно цвет фона, текста, линии/границы и т.д.

- Расположите элементы удобно по полю.
- Задайте размер текста поля <Фамилия> равным 24 пт, шрифт - синего цвета.
- Увеличьте в высоту рамку поля <Фотография>.
- Сохраните форму с именем <Данные студентов>.
- Посмотрите все способы представления форм: в режиме Конструктора, режиме Макета и режиме Форм.
 - Закройте форму.

4) Добавьте в таблицу <Личные данные> логическое поле <Институт> (т.е., собирается ли в дальнейшем учащийся поступать в институт). Значение этого поля <ДА> или <НЕТ>.

- Откройте таблицу <Личные данные> в режиме Конструктор. Добавьте поле с именем <Институт> и типом Логический. Закройте таблицу.

- Перейдите на закладку Формы и откройте форму <Данные студентов> в режиме Конструктор

- Щелкните по кнопке <Список полей> на панели инструментов, выделите название <Институт> и перетащите его мышкой в область данных, появиться значок и надпись <Институт>. 

- Расположите новые элементы по правилам оформления формы (с помощью мыши).

- Закройте <Список полей>

Примечание 3. Если флажок установлен, поле в таблице имеет значение <ДА>, если снят, то <НЕТ>.

- Перейдите в режим <Раздельная форма> и посмотрите записи. Установите флажки у восьми разных учащихся.

- Закройте форму, ответив утвердительно на вопрос о сохранении.

5) Создайте кнопочную форму <Заставка> с помощью Конструктора.

- Щелкните по кнопке <Создать>.
- Выберите <Конструктор>. Появится пустая форма. Задайте мышкой ширину формы, равную 10см, а высоту – 7см.
- Сохраните работу с именем <Заставка>.
- Откройте созданную форму <Заставка> в режиме Конструктора.
- Выберите на панели инструментов <Элементы управления> кнопку Аа – <Надпись>. Курсор мышки примет вид крестика с «приклеенной» буквой А. Щелкните мышкой по месту начала надписи и введите:
База данных
«Гостиница» группа ГС - 31
 (после слов База данных нажмите одновременно комбинацию клавиш Shift+Enter.)
- Нажмите клавишу <Enter>. Выберите размер букв 18, а выравнивание - по центру. Цвет фона – голубой. Растяните мышкой надпись на ширину окна.
- Выберите на панели элементов значок - Кнопка. Щелкните мышкой по тому месту области данных, где должна быть кнопка. Появится диалоговое окно <Создание кнопок>.
- Выберите категорию <Работа с формой>, а действие <Открыть форму>, и щелкните по кнопке <Далее>.
- Выберите форму <Успеваемость>, открываемую этой кнопкой щелкните по кнопке <Далее>. В следующем окне также щелкните по кнопке <Далее>.
- В следующем окне поставьте переключатель в положение <Текст>, наберите в поле слово <Успеваемость> (Рисунок 8.1) и щелкните по кнопке <Далее>.

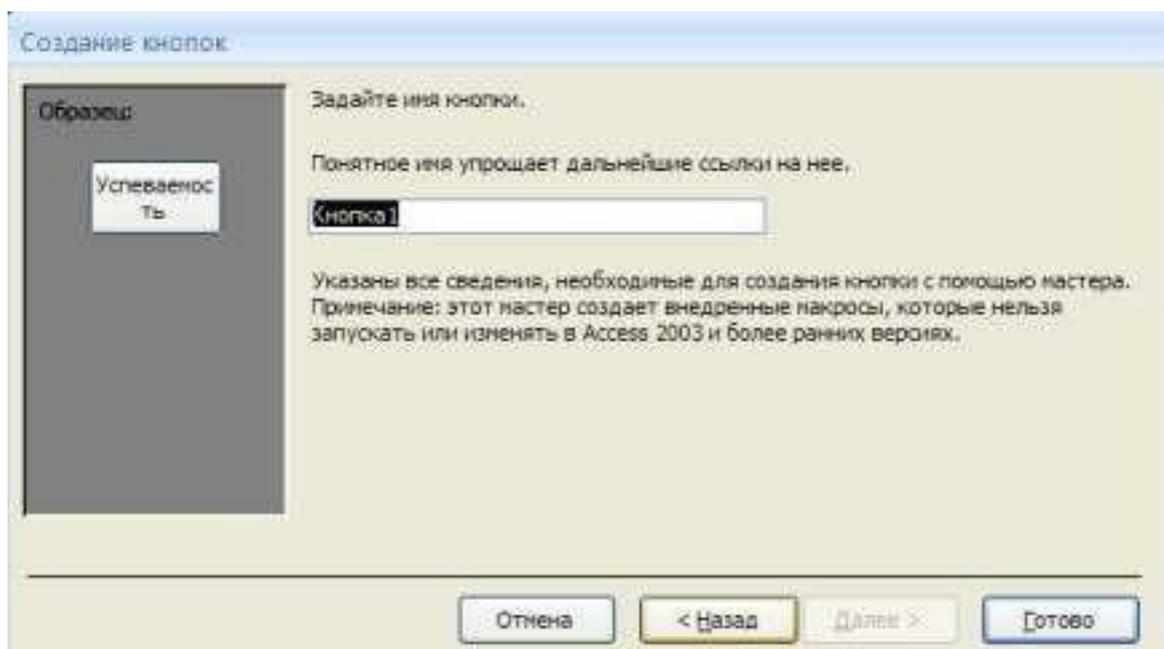


Рисунок 8.1. Создание кнопок

- Задайте имя кнопки <Успеваемость> и щелкните по кнопке <Готово>.

Примечание 4. Размер и расположение кнопок можно менять мышкой в режиме Конструктор.

Самостоятельно создайте кнопки для форм <Личные данные> и <Преподаватели>.

- Перейдите в режим формы (Рисунок 8.2). Теперь при щелчке мышью по соответствующим кнопкам будут открываться соответствующие формы для работы. □ Закройте форму.



Рисунок 8.2. Окно формы

б) Создайте кнопочную форму при помощи Диспетчера кнопочных форм.

- Откройте вкладку Работа с базами данных, команда - Диспетчер кнопочных форм. Вы получите диалоговое окно, представленное на Рисунке 8.3.

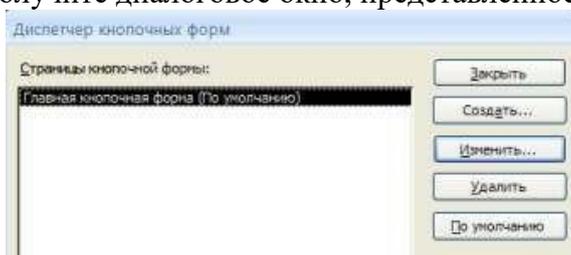


Рисунок 8.3. Диспетчер кнопок

□ Щелкните в этом окне по кнопке <Изменить>.

- В следующем окне щелкните по кнопке <Создать> и в появившемся окне измените содержимое полей в соответствии с Рисунком 8.4 (Команду и Форму выбирайте из списка, а не набирайте вручную). Щелкните по кнопке <ОК>.

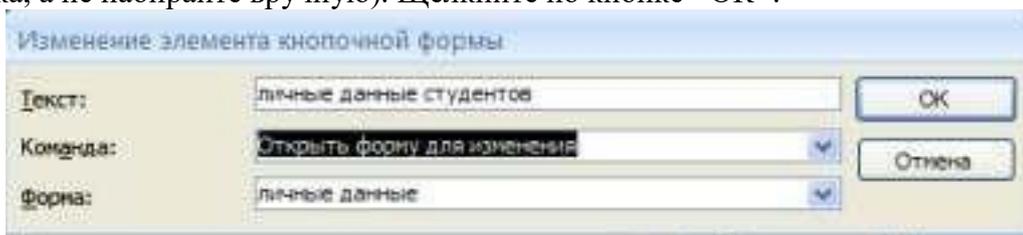


Рисунок 8.4. Изменение элементов кнопочной формы

- Аналогично создайте еще три элемента кнопочной формы: <Успеваемость>, <Преподаватели> и <Заставка>.

• Добавьте кнопку закрытия базы данных. Для этого щелкните по кнопке <Создать>, наберите в поле Текст слово <Выход>, а в поле Команда выберите <Выйти из приложения>. Закройте диалоговые окна.

• Откройте окно <Кнопочная форма> в режиме Конструктора или Макета, измените цвет надписи и название вашей базы данных на ГОСТИНИЦА, сохраните форму.

• Украсьте вашу форму рисунком. Для этого щелкните по значку Эмблема и выберите в открывшемся окне папку с рисунками, выберите понравившийся и вставьте в свою кнопочную форму.

- Перейдите в режим формы, проверьте работу всех кнопок кнопочной формы.

Завершите работу с базой данных, нажав на кнопку <Выход>.

Задание 2. Создайте отчет с помощью Мастера отчетов.

- Откройте вкладку *Создание*, меню *Отчеты*.
- Выберите *Мастер отчетов* и таблицу «Личные данные».
- Выберите нужные поля, которые будут участвовать в отчете, нажмите кнопку «Далее».
- В новом окне выберите поля для группировки так, чтобы сначала было указано поле «Фамилия», нажмите кнопку «Далее».
- На этом шаге отсортируйте данные по алфавиту, нажмите кнопку «Далее».
- Выберите вид макета *Ступенчатый* и щелкните по кнопке «Далее».
- Выберите стиль отчета: *Открытая* и щелкните по кнопке «Далее».
- Задайте имя отчета: «Отчет1» и щелкните по кнопке «Готово». Вы попадете в режим просмотра отчета.
- Закройте отчет согласившись с сохранением.

Самостоятельно. Составьте еще два отчета по запросам – «Запрос 3» и «Запрос 5», выбирая из разных макетов: *блок*; *структура*, выбирая из разных стилей. Сохраните отчеты под именами «Отчет 2» и «Отчет 3».

Задание 3. Создайте Пустой отчет в столбец на базе таблицы «Ведомость успеваемости» и сохраните его с именем «Успеваемость».

С помощью Конструктора измените цвет букв заголовка, их размер и шрифт.

Задание 4. Создайте почтовые наклейки.

- Откройте вкладку *Создание*, меню *Отчеты*.
- Выберите таблицу «Личные данные», команда *Наклейки*.
- В следующем окне щелкните по кнопке «Далее».
- В следующем окне выберите шрифт, размер шрифта, насыщенность и цвет, вновь щелкните по кнопке «Далее».
- В следующем окне создайте прототип наклейки, напечатав слово **ЛИЧНОСТЬ** и выбрав соответствующие поля, щелкните по кнопке «Далее».
- В следующем окне укажите поля для сортировки (Фамилия, Имя), щелкните по кнопке «Далее».
- Введите имя отчета «Наклейки» и щелкните по кнопке «Готово».
- Просмотрите Наклейки (Рисунок 8.5).

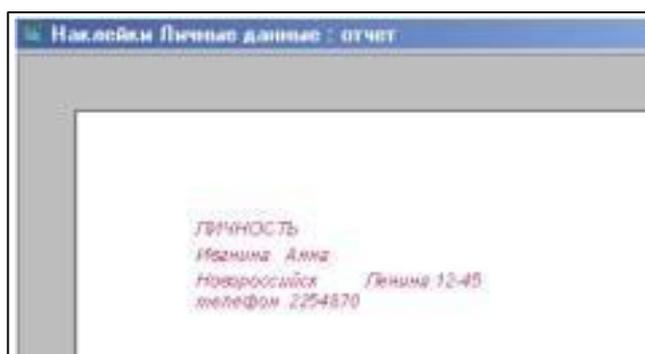


Рисунок 8.5. Отчет

Практическая работа № 9

Проектирование реляционной схемы базы данных в среде СУБД

Цель работы: овладение навыками работы с переменными, обработка табличных файлов в среде VisualFoxPro.

VisualFoxPro

При создании приложения используется проект, который объединяет элементы приложения VisualFoxPro и группирует их по типам. База данных в VisualFoxPro – это совокупность таблиц, отношений между таблицами, индексов, триггеров, хранимых процедур.

База данных является частью проекта, поэтому её целесообразно создавать в окне проекта.

Структурными элементами базы данных являются:

Поле – элементарная единица логической организации данных, которая соответствует неделимой единице информации – реквизиту. Для описания поля используются следующие характеристики: имя, длина, тип и точность (для числовых данных). В структуре записи файла указываются поля, значения которых являются ключами: первичными и внешними.

Первичный ключ - это одно или несколько полей, однозначно идентифицирующих запись. Если первичный ключ состоит из одного поля, он называется простым, если из нескольких полей - составным ключом.

Внешний ключ - это одно или несколько полей, которые выполняют роль поисковых или группировочных признаков. В отличие от первичного, значение внешнего ключа может повторяться в нескольких записях файла, то есть он не является уникальным. Если по значению первичного ключа может быть найден один единственный экземпляр записи, то по внешнему – несколько.

Файл (таблица) – совокупность экземпляров записей одной структуры. Описание логической структуры записи файла содержит последовательность расположения полей записи и их основные характеристики,

Запись – совокупность логически связанных полей.

Экземпляр записи - отдельная реализация записи, содержащая конкретные значения ее полей.

Таблицы составляют основу вашей базы данных. В них будет храниться вся необходимая информация. В дальнейшем данные в таблице будут дополняться новыми данными, редактироваться или исключаться из таблицы. Поля таблицы предназначены для хранения в них данных. Это могут быть числа, текстовая информация, даты, графические файлы и т.д. В VisualFoxPro допустимыми являются типы полей, указанные в таблице 9.1.

Таблица 9.1. Некоторые типы данных, используемых в СУБД FoxPro

Тип	Описание	Пример
Integer	Целые числа	9846
Numeric	Данные, с фиксированной точкой	3.1456
		-56.235
		"01/07/04"
Logical	Поля, содержащие только одно из двух возможных значений (да/нет)	True; False
Date	Дата	01/07/04
DateTime	Дата и время	01/07/04 12:30:00 pm
Memo	Очень длинный текст или комбинация текста и чисел	

Таблица 9.2. Некоторые команды СУБД FoxPro

Команда	Описание
CREATE	создание элементов БД (базы, таблиц, отчётов, запросов и т.д.)
MODIFY (STRUCTURE)	изменение элементов БД (базы, таблиц, отчётов, запросов и т.д.)
BROWSE, EDIT, CHANGE, APPEND	команды редактирования и просмотра содержимого таблиц
OPEN DATABASE	Открытие БД
CLOSE	закрытие элементов БД (базы, таблиц, отчётов, запросов и т.д.)
QUIT	Выход из Visual FoxPro

Задание для практической работы

Создать в служебной папке Мои документы новую папку и присвоить ей имя, пример:

Зкурс4группаfoxlab (указывать свой курс и группу)

Запустить программу Microsoft Visual FoxPro:

Пуск/программы/VisualFoxPro

Ознакомиться с элементами рабочего окна программы.

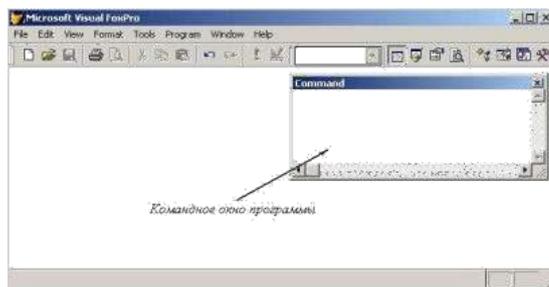


Рисунок 9.1. Главное окно VisualFoxPro 8.0

Создать новый проект: File/New/Project/NewFile, указать созданную нами ранее папку, присвоить имя проекту Информационные системы и сохранить. Все создаваемые в последующем элементы приложения будут храниться в проекте Информационные системы БД создаётся аналогично: File/New/Database/Newfile, присвоить имя Штат и сохранить. Структура проекта и его элементы отражаются в окне программы ProjectManager (Менеджер проекта).

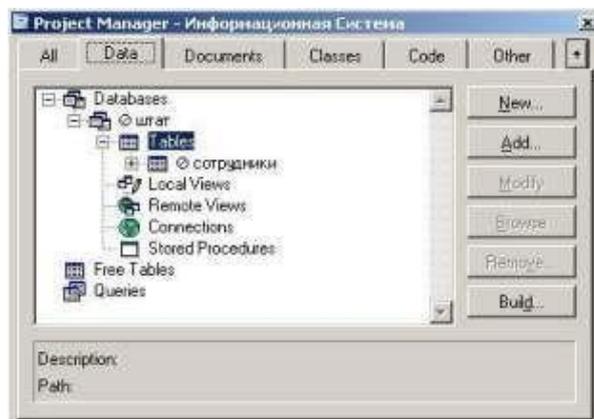


Рисунок 9.2. Окно Project Manager

Добавить БД в проект: в окне ProjectManager щёлкнуть на вкладке Data/Databases/Add, в открывшемся диалоговом окне выбрать БД и нажать ОК

Создать таблицу в БД штат: в окне ProjectManager (см. рис.9.2.) щёлкнуть клавишей мыши на вкладке Data/Databases/штат/Tables/New/Newtable, присвоить имя Сотрудники и сохранить

В появившемся окне TableDesigner(Конструктор таблиц) (см. рис.9.3.) на вкладке Fields (Поля) создать структуру таблицы в соответствии с таблицей 3.

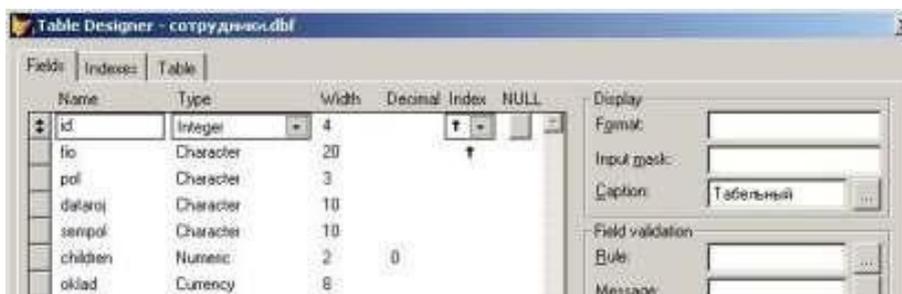


Рисунок 9.3. Окно конструктора таблицы TableDesigner вкладка Fields

Таблица 9.3.Определение полей таблицы Сотрудники в окне конструктора таблицы TableDesigner

(наименование поля)	(тип поля)	(ширина поля)	(индексное поле)	(надпись, определяет заголовок поля)
Name	Type	Width	Index	Caption
id	Integer	4	↑Ascending	Табельный номер
fio	Character	20		ФИО

Перейти на вкладку Indexes(Индексы) окна конструктора таблицы TableDesigner и присвоить созданному индексу значения в соответствии с таблицей 9.4. Это необходимо для создания ключевого поля.

Order	Name	Type	Expression	Filter	Collate
↑	id	Primary	id		Machine

Для ввода и редактирования данных в таблице *Сотрудники* в командном окне программы введите команду APPEND и нажмите клавишу Enter Командное окно программы **FoxPro** (см. рис.9.1.) предназначено для ввода команд с клавиатуры и последующего их выполнения. Любые действия и операции над элементами приложения в СУБД **FoxPro** могут осуществляться при помощи команд, вводимых в программное окно.

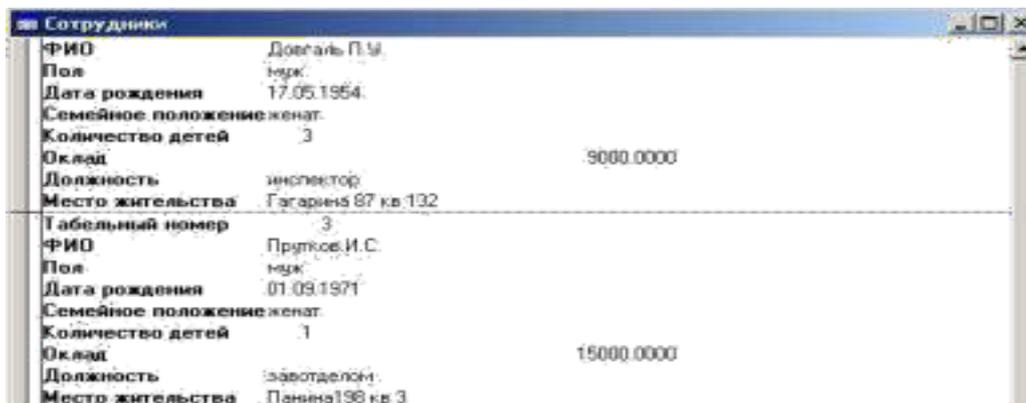


Рисунок 9.4. Просмотр и редактирование содержимого таблицы в режиме APPEND

После ввода данных (см. таблицу 9.4.) в командном окне программы введите команду BROWSE и нажмите клавишу Enter. Результат должен соответствовать рис.9.5.

Рисунок 9.5. Просмотр таблицы в режиме BROWSE

Таблица 9.4. Содержимое таблицы Сотрудники

Таб. номер	ФИО	Пол	Дата рождения	Семейное положение	Количество детей	Оклад	Должность	Место жительства
1	Гуськов Г.Г.	муж	22.02.1976	холост	0	7000	экономист	Липкина 156 кв.43
2	Довгаль П.У.	муж	17.05.1954	женат	3	9000	инспектор	Гагарина 87 кв.132
3	Прутков И.С.	муж	01.09.1971	женат	1	15000	завотделом	Панна 198 кв.3
4	Косыгин Н.И.	муж	28.03.1979	холост	0	7000	экономист	Николева 30 кв.77
5	Петрикеев Е.П.	муж	14.08.1980	холост	0	4500	консультант	Нурдимова 18 кв.84
6	Беркова Б.Ю.	жен	24.11.1959	замужем	4	12000	бухгалтер	Дунчевского 64 кв.9
7	Петренко Е.П.	жен	29.09.1981	незамужем	0	7000	экономист	Левина 101 кв.15
8	Петросова Г.Н.	жен	19.06.1976	замужем	2	4500	консультант	Миро пр. 120 кв.2
9	Орлова К.Д.	жен	31.12.1977	незамужем	0	7000	экономист	Седова 18 кв.112
10	Косыгина А.К.	жен	01.10.1969	замужем	1	6000	менеджер	Панна 198 кв.3

В окне BROWSE установите курсор в нижней части таблицы и нажмите комбинацию клавиш Ctrl + Y на клавиатуре. В таблицу добавится новая строка

Прежде чем удалить строку в VisualFoxPro её сначала необходимо пометить на удаление. Для этого щелкнуть клавишей мыши в ячейке слева от удаляемой записи в области узкого, непоименованного столбца, ячейка станет помеченной чёрным цветом, или выделить необходимую строку и выбрать команду ToggleDeletionMark (Установка

метки на удаление) пункта Table (Таблица) системного меню VisualFoxPro, запись будет помечена на удаление. Удалить запись, выбрав команду RemoveDeletedRecords (Удалить запись) пункта Table (Таблица) системного меню VisualFoxPro. Запись будет удалена.

Для закрытия всех элементов приложения в командном окне программы введите команду CLOSE ALL.

Для завершения работы с программой **VisualFoxPro** в командном окне программы введите команду QUIT.

Создание отношений между таблицами в многотабличной БД Библиотека

Среди требований, предъявляемых к СУБД, основное место занимает возможность быстрого поиска необходимой информации. Средством, позволяющим решить эту проблему, является применение индексов. В VisualFoxPro для создания первичных ключей, определяющих отношения между таблицами и условия целостности данных также предназначены индексы. В этом случае индексы должны быть уникальными, то есть значения индексированного поля должны быть неповторяющимися (уникальными).

Для создания индекса таблицы используется вкладка Indexes (Индексы) окна конструктора таблиц TableDesigner. Все индексы в VisualFoxPro имеют имена, задаваемые в поле Name (Имя).

Для задания типа создаваемого индекса используется список Type (Тип).

Таблица 9.10. Описание типов индекса

Тип индекса	Описание
Regular (Обычный)	Создается индекс, в котором для каждой записи таблицы хранится значение индексного выражения. Если несколько записей имеют одинаковое значение индексного выражения, то каждое значение хранится отдельно и содержит ссылку на связанную с ней запись.
Unique (Уникальный)	Создается индекс, в котором хранятся только неповторяющиеся значения индексного выражения. Если две или более записей содержат одинаковое значение индексного выражения, то будет храниться только одно значение и ссылка на первую из записей с одинаковым значением индексного выражения. Таблица может иметь несколько уникальных индексов.
Candidate (Кандидат)	Создается уникальный индекс, который не содержит полей с пустыми значениями. Этот индекс обладает всеми качествами первичного ключа и не является им только по той причине, что таблица не может содержать более одного первичного ключа.
Primary (Первичный)	Создается уникальный индекс, который используется для связывания таблиц и определения условий целостности данных. Поля, входящие в первичный ключ, не должны допускать ввода пустых значений. В отличие от уникального индекса, таблица может иметь только один первичный ключ.

Обычно, в VisualFoxPro при создании форм, отчетов и запросов используются несколько таблиц, между которыми установлены постоянные отношения. Такие таблицы называются связанными. Из двух связанных таблиц одна является главной (родительской), а другая подчиненной (дочерней). При создании индексов для

родительской таблицы должен быть определён ключ типа Primary (Первичный) или типа Candidate (Кандидат), а для дочерней таблицы – индекс для связи с родительской таблицей типа Regular (Обычный).

Таблица 9.11. Типы отношений между таблицами

Типы отношений	Описание
Отношение "один-к-одному"	Каждая запись в одной таблице соответствует только одной записи в другой таблице
Отношение "один-ко-многим"	Наиболее распространённый тип отношений, каждой записи в одной таблице может соответствовать несколько записей в другой таблице
Отношение "много-к-одному"	Отношение "много-к-одному" можно сравнить с отношением "один-ко-многим", рассматриваемое с другой точки зрения
Отношение "много-ко-многим"	Нескольким записям одной таблицы может соответствовать несколько записей в другой таблице

Одним из самых важных требований, предъявляемых к базам данных, является целостность данных, которую определяют установленные между таблицами отношения. Для определения целостности данных в VisualFoxPro используется окно построителя условий целостности данных ReferentialIntegrityBuilder (Построитель целостности данных), которое содержит перечень всех установленных отношений между таблицами (см. рис.9.3.).

Таблица 9.12. Описание действий VisualFoxPro, в зависимости от выбранной опции, при изменении значения первичного ключа или ключа типа

Наименование опции	Описание
Cascade (Каскадное изменение)	При изменении значений полей первичного ключа или ключа-кандидата в родительской таблице автоматически осуществляется каскадное изменение всех соответствующих значений в дочерней таблице
Restrict (Запрет изменения)	Не позволяет изменять значения полей первичного ключа или ключа-кандидата в родительской таблице, если в дочерней таблице имеется хотя бы одна запись, содержащая ссылку на изменяемую запись
Ignore (Игнорировать)	Позволяет изменять значения полей первичного ключа или ключа-кандидата в родительской таблице независимо от существования связанных записей в дочерней таблице. Целостность данных при этом не поддерживается

Таблица 9.13. Действия VisualFoxPro, в зависимости от выбранной опции, при удалении записи из родительской таблицы

Наименование опции	Описание
Cascade (Каскад)	При удалении записи из родительской таблицы автоматически осуществляется каскадное удаление всех записей из дочерней таблицы, связанных с удаляемой записью.
Restrict (Запрет)	Не позволяет удалять записи в родительской таблице, если в дочерней таблице имеется хотя бы одна запись, содержащая ссылку на удаляемую запись. При попытке удаления записи возникает ошибка, которую вы можете обработать программно.
Ignore (Игнорировать)	Позволяет удалять записи в родительской таблице независимо от существования связанных записей в дочерней таблице. Целостность данных при этом не поддерживается.

Индивидуальные задания к практической работе

Создать многотабличную БД и присвоить имя Библиотека. Создать следующие таблицы:

1. Таблица Книги

(наименование поля)	(тип поля)	(ширина поля)	(индексное поле)	(название, определяет заголовок поля)
Name	Type	Width	Index	Caption
Kodknigi	Integer	4		Код книги
nazvanie	Character	40		Название
razdel	Character	15		Раздел
izdat	Character	20		Издательство
godizdan	Numeric	4		Год издания
mestolran	Character	5		Местопребывания

Определение полей таблицы Книги в окне конструктора таблицы **TableDesigner**

Содержимое таблицы Книги

Код	Название	Раздел	Издательство	Год	Место
книги				издани я	хранени я
1	Практический курс программирования	Информатика	Наука	1983	6-11
2	TURBOPASCAL для школьников	Информатика	Финансы и статистика	1999	6-22
3	Занимательная математика	Математика	Тригон	1998	3-14
4	HTML в действии	Информатика	Питер	1997	5-4
5	Национальное счетоводство	Экономика	Финансы и статистика	1998	4-11
6	Самоучитель VisualFoxPro 8.0	Информатика	БХВ-Петербург	2003	5-34
7	Язык телодвижений	Психология	Наука	1984	2-17
8	Теория машин	Техника	Машиностроение	1957	3-15
9	Теория гипноза	Психология	Наука	1999	2-31
10	Карьера менеджера	Экономика	Парадокс	1998	1-212

2. Таблица Разделы

(наименование поля)	(тип поля)	(ширина поля)	(индексное поле)	(надпись, определяет заголовок поля)
Name	Type	Width	Index	Caption
razdel	Character	15		Раздел

Определение полей таблицы Разделы в окне конструктора таблицы TableDesigner
Содержимое таблицы Разделы

Раздел
Экономика
Информатика
Психология
Математика
Техника

3. Таблица Издательство

Определение полей таблицы Издательство в окне конструктора таблицы TableDesigner

(наименование поля)	(тип поля)	(ширина поля)	(индексное поле)	(надпись, определяет заголовок поля)
Name	Type	Width	Index	Caption
izdat	Character	20		Издательство
gorod	Character	15		Город

Содержимое таблицы *Издательство*

Издательство	Город
Финансы и статистика	Москва
Тригон	Санкт-Петербург
Питер	Санкт-Петербург
Наука	Москва
Машиностроение	Москва
Парадкс	Минск
БХВ-Петербург	Санкт-Петербург

4. Таблица *Автор_книги*

Определение полей таблицы *Автор_книги* в окне конструктора таблицы **TableDesigner**

Имя полей	Тип полей	Ширина полей	Индекс полей	Имя, определяет заголовок полей
Name	Type	Width	Index	Caption
kodevknigi	Integer	4		Код автора книги
kodeknigi	Integer	4		Код книги
kodevtora	Integer	4		Код автора

Содержимое таблицы *Автор_книги*

Код автора книги	Код книги	Код автора
1	1	1
2	1	2
3	2	3
4	3	4
5	4	5
6	5	6
7	6	7
8	7	8
9	8	9
10	9	10
11	10	11

5. Таблица *Авторы*

Определение полей таблицы *Авторы* в окне конструктора таблицы **TableDesigner**

Имя полей	Тип полей	Ширина полей	Индекс полей	Имя, определяет заголовок полей
Name	Type	Width	Index	Caption
kodevtora	Integer	4		Код автора
familya	Character	20		Фамилия
imya	Character	15		Имя

Содержимое таблицы *Авторы*

Код автора	Фамилия	Имя
1	Фролов	Геннадий
2	Илюхин	Виктор
3	Попов	Владимир
4	Ахимова	Светлана
5	Морис	Брюс
6	Кулагина	Галина
7	Омельченко	Людмила
8	Алан	Пиз
9	Мальцев	Анатолий
10	Горин	Дмитрий
11	Яковка	Лиз

6. Открыть БД штат: в окне ProjectManager (Менеджер проекта) вкладка Data/Databases/Библиотека/Open

7. Необходимо модифицировать таблицы БД Библиотека, создать индексированные поля, а также первичные ключи для осуществления связей между таблицами. Для модификации (изменения структуры) таблицы в окне проекта ProjectManager (Менеджер проекта) установить курсор на модифицируемую таблицу Книги и нажать кнопку Modify (Модифицировать) или в окне конструктора БД DatabaseDesigner (Конструктор базы данных) установить курсор в таблицу Книги, вызвать контекстное меню и выбрать команду Modify (Модифицировать). На экране откроется диалоговое окно TableDesigner (Конструктор таблицы)

8. В окне TableDesigner (Конструктор таблицы) перейти на вкладку Indexes (Индексы) и ввести значения в соответствии с таблицей 6., нажать кнопку ОК, система попросит подтвердить сохранение изменений, нажать Yes Таблица 9.15.Определение индексов таблицы Книги на вкладке Indexes (Индексы) окна конструктора таблицы TableDesigner.

Order	Name	Type	Expression	Filter	Collate
1	кодкниг	Primary	кодкниг		Machine
1	razdel	Regular	razdel		Machine
1	izdat	Regular	izdat		Machine



Рисунок 9.5. Вкладка Indexes (Индексы) окна TableDesigner таблицы Книги

Индекс, с присвоенным типом Primary является первичным ключом таблицы

9. Аналогично внести изменения в таблицы БД Библиотека: Разделы, Издательство, Автор_книги, Авторы в соответствии с таблицами 9.16,9.17, 9.18, 9.19.

Таблица 9.16.Определение индексов таблицы Разделы на вкладке Indexes (Индексы) окна конструктора таблицы TableDesigner

Order	Name	Type	Expression	Filter	Collate
1	razdel	Primary	razdel		Machine

Таблица 9.17.Определение индексов таблицы Издательство на вкладке Indexes (Индексы) окна конструктора таблицы TableDesigner

Order	Name	Type	Expression	Filter	Collate
1	izdat	Primary	izdat		Machine

Таблица 9.18.Определение индексов таблицы Автор_книги на вкладке Indexes (Индексы) окна конструктора таблицы TableDesigner

Order	Name	Type	Expression	Filter	Collate
1	kodavknigi	Primary	kodavknigi		Machine
2	kodavtora	Regular	kodavtora		
3	kodknigi	Regular	kodknigi		

Таблица 10. Определение индексов таблицы Авторы на вкладке Indexes (Индексы) окна конструктора таблицы TableDesigner

Order	Name	Type	Expression	Filter	Collate
1	kodavtora	Primary	kodavtora		Machine

10. В окне DatabaseDesigner (Конструктор базы данных) выбрать родительскую таблицу Разделы. Таблицы в конструкторе БД обозначаются прямоугольниками, в нижней части которых после надписи Indexes (Индексы) расположен список индексов, созданных для данной таблицы. Первичный ключ в этом списке выделяется значком ключа, расположенным с левой стороны от наименования индекса. Установить курсор мыши на первичный ключ razdel таблицы Разделы. Нажать кнопку мыши и, не отпуская ее, переместить курсор мыши на индекс razdel дочерней таблицы Книги. Отпустить кнопку мыши. Созданные отношения между таблицами отображаются в виде линий. Аналогичным образом, создать связь между таблицами Издательство и Книги

(Таблица Издательство – родительская, таблица Книги – дочерняя, первичный ключ – izdat).

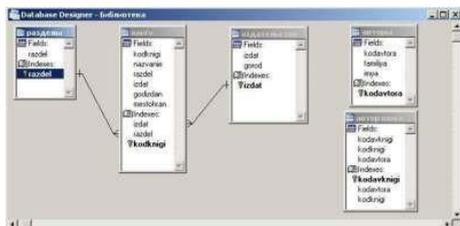


Рисунок 9.6. Отображение связей между таблицами в окне DatabaseDesigner (Конструктор базы данных)

Для редактирования отношений между связанными таблицами дважды щелкнуть левой кнопкой мыши на линии, появится диалоговое окно EditRelationship(Редактирование отношения), в котором слева приведено наименование родительской таблицы и располо-

жен раскрывающийся список индексов таблицы, справа размещена аналогичная информация о дочерней таблице. В этом диалоговом окне указан также тип установленного отношения между таблицами.

9. В окне EditRelationship(Редактирование отношения) нажать кнопку ReferentialIntegrity(Целостность данных), в появившемся диалоговом окне

ReferentialIntegrityBuilder (Построитель целостности данных) выбрать отношение издательство- книги. В полях Update (Изменить), Delete (Удалить), Insert (Заменить) установить тип действий Restrict (Запрет изменения). Провести аналогичные действия для отношения разделы-книги. Результат описанных действий, которые необходимы для обеспечения целостности данных, представлен на рис.9.7.

10. Для сохранения выполненных действий нажать кнопку ОК, система потребует подтверждение сохранения, нажать кнопку Да

11. Нажать кнопку ОК для закрытия диалогового окна EditRelationship.

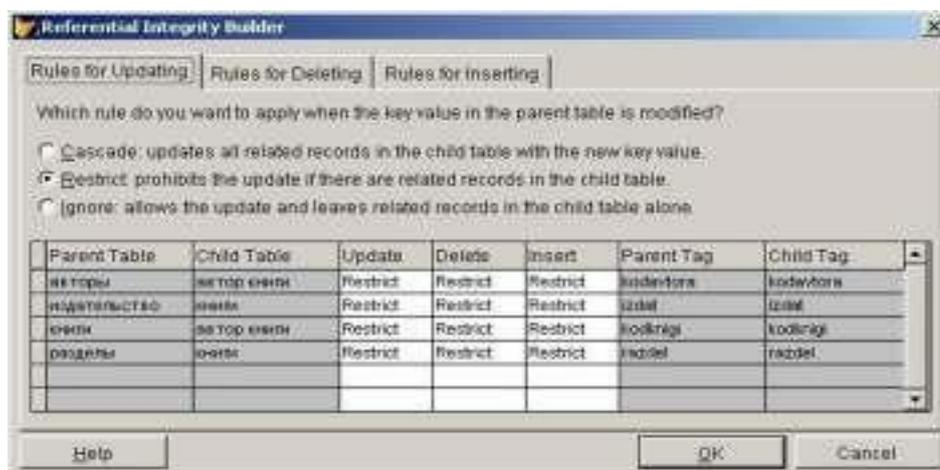


Рисунок 9.7. Диалоговое окно ReferentialIntegrityBuilder(Построитель целостности данных)

12. Создать связь между таблицами Книги и Автор_книги (kodknigi)

13. Создать связь между таблицами Авторы и Автор_книги (kodavtora)

14. Установить целостность данных в созданных отношениях: в полях Update (Изменить), Delete (Удалить), Insert (Заменить) установить тип действий Restrict (Запрет изменения)

Практическая работа № 10. Проектирование реляционной схемы базы данных в среде СУБД

Цель работы: овладение практическими навыками по созданию форм

Формы в VisualFoxPro

В VisualFoxPro для просмотра, ввода и редактирования данных, хранящихся в таблицах, используются формы, являющиеся более наглядным средством представления информации. Важным преимуществом форм является, то, что они позволяют работать не с одной, а с несколькими связанными таблицами, что, в свою очередь, также увеличивает наглядность. При создании форм в VisualFoxPro разработчик может использовать следующие средства: FormWizard (мастер форм), FormBuilder (построитель формы), Builder (построитель объектов формы), AutoFormatBuilder (построитель автоформата), FormDesigner (конструктор форм).

Чтобы создать форму для одной или связанных таблиц с возможностью задания отображаемых в форме полей, стиля их отображения и указания типа кнопок управления, можно использовать FormWizard (мастер создания форм).

Для самостоятельной разработки формы с заданными свойствами или изменения формы, созданной с помощью мастера, вам необходимо использовать FormDesigner (конструктор форм).

Для облегчения размещения в конструкторе форм полей и надписей, оформленных в соответствии с выбранным стилем, можно использовать FormBuilder (Построитель формы).

Создание формы с помощью конструктора форм

Любая форма в VisualFoxPro состоит из объектов, каждый из которых имеет характерные свойства. Процесс создания формы в конструкторе форм состоит в размещении в форме объектов и определении свойств, а также связанных с ними событий и выполняемых действий.

Процесс создания формы включает следующие действия:

- настройка параметров формы
- определение среды окружения, то есть выбор используемых в форме таблиц и установка связей между ними
- размещение в форме объектов: текста, полей различных типов, линий, рисунков, кнопок управления.

Задание для практической работы

Задание 1. Создание формы средствами мастера форм

Запустить программу MicrosoftVisualFoxPro

Открыть созданный проект: File/Open/Информационная система

Открыть БД штат: в окне ProjectManager (Менеджер проекта) вкладка Data/Databases.

Создать форму Сотрудники, щелкнув клавишей мыши на вкладке Documents/Forms/New/FormWizard/FormWizard, нажать клавишуOk.



Рисунок 10.1. Диалоговое окно WizardSelection

В появившемся окне FormWizard в области Databasesandtables (Базы данных и таблицы) выбрать необходимую БД штат и указать таблицу Сотрудники, для которой создается форма. Из области Availablefields (Имеющиеся поля) выбрать поля, которые будут размещены в форме, в соответствии с рис.10.2, используя кнопки расположенные между списками для перехода к следующему шагу нажать кнопку Next(Далее).

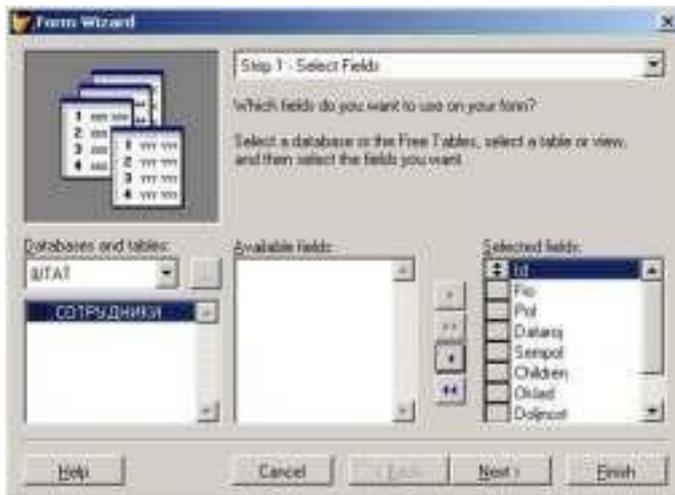


Рисунок 10.2. Диалоговое окно выбора полей для отображения

В появившемся диалоговом окне в области Style установить стиль отображения объектов и в области Button type типы кнопок управления в соответствии с рис.10.3, нажать кнопку Next.



Рисунок 10.3. Окно выбора стиля отображения полей и управляющих кнопок

В появившемся диалоговом окне задать критерий сортировки данных, указав поля по которым будет осуществляться упорядочивание в соответствии рис.10.4. нажать кнопку Next.

На заключительном шаге создания форм в области Typeatitleforyourform(Тип заголовка формы) указать имя формы Сотрудники. Воспользовавшись кнопкой Preview (Просмотр) просмотрите, как будет выглядеть создаваемая форма, после просмотра нажмите кнопку ReturntoWizard.

Если что-то не так, вернитесь к предыдущим шагам, воспользовавшись кнопкой Back. Нажать кнопку Finish (Готово) и в появившемся диалоговом окне SaveAs (Сохранить как) указать папку, в которой будет размещена форма Сотрудники.

Индивидуальные задания к практической работе

Создать форму для таблицы Книги БД Библиотека

1. В диалоговом окне выбора полей для отображения выбрать все поля таблицы Книги

2. В окне выбора стиля отображения полей и управляющих кнопок (см. рис.3.) в области Style (стиль) выбрать стиль Embossed, а в области Buttontype (Типы кнопок управления) выбрать пункт NoButtons
3. Установить критерий упорядочения данных по полю kodknigi
4. В окне задания заголовка формы в области Typeatitleforyourform(Тип заголовка формы) указать имя формы Книги. Просмотреть вид создаваемой формы нажатием кнопки Preview (Просмотр).
5. После сохранения запустить форму Книги, она должна иметь вид как на рис.10.8.

Рисунок 10.8. Форма Книги

Задание 2. Создание формы средствами конструктора форм

Запустить программу Microsoft Visual FoxPro Открыть проект Информационная система

Открыть БД штат Выбрать в окне Project Manager вкладку Documents/Forms/New/New Form. Откроется окно FormDesigner(Конструктор форм) для создания новой формы

Открыть окно окружения формы DataEnvironment (Среда окружения): пункт меню View/ DataEnvironment. Для размещения таблицы в среде окружения выбрать команду Add(Добавить) из меню DataEnvironment (Среда окружения) или вызвать контекстное меню на окне окружения формы DataEnvironment и выбрать пункт Add. В появившемся диалоговом окне AddTableorView (Добавить таблицу или представление данных), выбрать из списка таблиц открытой базы штат таблицу Сотрудники, для которой создается форма, и нажать кнопку ОК

Открыть окно свойств таблицы, размещённой в окне окружения DataEnvironment. Для этого установить на таблицу Сотрудники курсор, вызвать контекстное меню правой клавишей мыши и выбрать команду Properties (Свойства)

В окне Properties (свойства) выделить свойство Order (Упорядочение). Для упорядочения данных в форме в поле коррекции свойства нажмите кнопку раскрытия списка и из списка индексов таблицы выберите индекс (id), по которому хотите упорядочить данные. Закройте окно определения среды окружения DataEnvironment.

Для задания свойств формы установить курсор в форму, вызвать контекстное меню, выбрать команду Properties (Свойства). Откроется окно Properties(Свойства). В его верхнем списке, указывающем название объекта, для которого осуществляется настройка свойств содержится текст Form1 (Форма1)

В окне Properties (Свойства) скорректировать свойство Caption (Надпись), введя в текстовое поле заголовков формы Сотрудники текст

Свойство формы AutoCenter (Автоцентр) должно иметь значение True(Истина), чтобы форма располагалась в центре экрана

Изменить свойства FontName (Наименование шрифта), указав шрифт TimesNewRomani FontSize (Размер шрифта) указав размер шрифта 12 После того как вы

определили параметры формы, разместили в окружении используемые таблицы, можно приступать к размещению объектов в форме. Осуществим размещение полей таблицы Сотрудники и надписей к ним, используя FormBuilder (построитель формы).

Для запуска построителя форм установить курсор в форму, вызвать контекстное меню, выбрать команду Builder (Построитель), откроется диалоговое окно FormBuilder (построитель формы), содержащее две вкладки: FieldSelection (Выбор поля) и Styles (Стиль). В вкладке FieldSelection (Выбор поля) из области Databasesandtables (Базы данных и таблицы) выбрать БД штат и таблицу Сотрудники, затем из списка Availablefields (Имеющиеся поля) перенести в Selectedfields(Выбранные поля) все необходимые поля используя кнопки расположенные между списками. Перейти на вкладку Styles (Стиль), выбрать из списка стилей стиль Embossed и нажать кнопку ОК. На форме будут размещены объекты формы (см. рис. 10.9.).

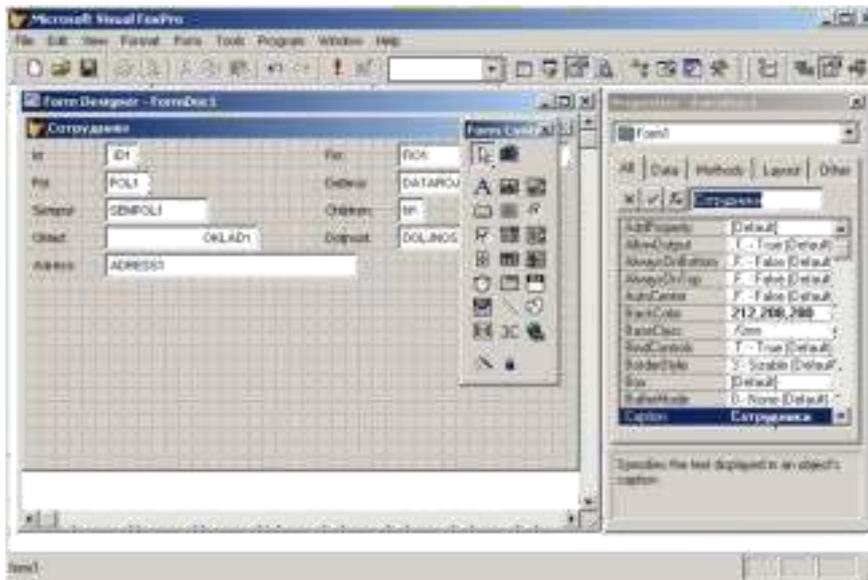


Рисунок 10.9. Окно FormDesigner(Конструктор форм)

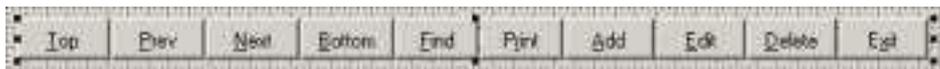


Рисунок 10.10. Объект - блок кнопок управления

Для размещения на создаваемой форме кнопок управления воспользуемся ранее созданной формой Сотрудники. Открыть форму Сотрудники в режиме редактирования: окно программы ProjectManager/Documents/Forms/сотрудники/Modify. В появившемся окне FormDesigner формы Сотрудники нажатием левой клавиши мыши выделить объект BUTTONSET1 (блок кнопок управления, см. рис.10.10.) и нажать комбинацию клавиш Ctrl+С на клавиатуре или установить курсор в форму, вызвать контекстное меню, выбрать команду Copy. Закрыть окно FormDesigner формы Сотрудники.

В окне FormDesigner формы Сотрудники тест установить курсор в форму, вызвать контекстное меню, выбрать команду Paste или нажать комбинацию клавиш Ctrl+V. Блок кнопок управления записями таблицы (см. рис.10.10) в создаваемой форме будет скопирован.

Задать цвет фона формы. Выделить в окне Properties (Свойства) свойство формы BackColor (цвет фона), нажать расположенную с правой стороны поля редактирования свойства кнопку и в открывшемся диалоговом окне Цвет выберите цвет, который вы хотите использовать для фона

Расположить объекты формы в соответствии с рис.10.11. Для перемещения объектов формы можно использовать метод перетаскивания мышью или кнопками управления

курсором на клавиатуре, предварительно выделив необходимый объект, а также комбинацией клавиш Ctrl+[кнопки управления курсором], использовать все выше перечисленные способы, выяснить их различия. Для изменения размеров формы или объектов формы необходимо выделить объект (форму), подвести курсор к углу объекта (формы), когда курсор примет вид разнонаправленной стрелки нажать левую клавишу мыши и удерживая её, изменить размеры объекта (формы).

Запустить отредактированную форму: установить курсор в форму, вызвать контекстное меню, выбрать команду RunForm или нажать кнопку [] на панели инструментов. Система попросит сохранить созданную форму, нажать кнопку Yes, в появившемся окне указать папку и имя файла (Сотрудники тест), сохранить.

Индивидуальные задания к практической работе

Создать кнопки навигации в форме Книги

Открыть форму Книги в окне FormDesigner (Конструктор форм)

Нажать кнопку CommandGroup (Группа кнопок)  на панели инструментов FormControls (элементы управления формы) и щёлкнуть в месте их предполагаемого размещения в форме

В Окне Properties (свойства) размещённого объекта выделить свойство ButtonCount (Количество кнопок) и указать необходимое количество размещаемых в форме кнопок, указав число 5

Увеличить с помощью мыши размеры рамки, окружающей данный объект. Перейти в режим редактирования: установить на объект курсор, вызвать контекстное меню и выбрать команду Edit (Редактировать). Выделяя поочередно кнопки, переместить их, расположив горизонтально, в одну линию

Выйти из режима редактирования, щёлкнув вне области объекта

CommandGroup (Группа кнопок)

Скорректировать размер рамки, окружающей объект: в свойстве AutoSize (Авторазмер) установить значение True (Истина)

Открыть окно свойств объекта CommandGroup (Группа кнопок), нажать кнопку раскрытия списка в верхней части данного окна, поочередно выбрать из списка элементы Command1,

Command2, Command3, Command4, Command5 и, используя свойство

Caption(Надпись) задать названия кнопок соответственно: Первая, Следующая, Предыдущая, Последняя, Выход (см. рис.10.12.).

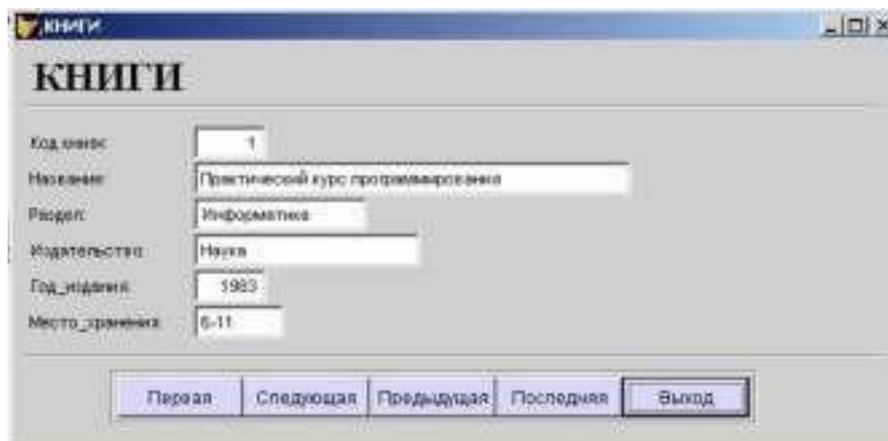


Рисунок 10.12. Форма Книги

Определить команды, которые будут выполняться при нажатии на созданные кнопки. Установить на объект CommandGroup (Группа кнопок) курсор, вызвать контекстное меню и выбрать команду Edit (Редактировать), выделить кнопку Первая,

вызвать контекстное меню выбрать команду Code (Код программы), в появившемся окне ввести следующие команды:

Для кнопки Первая:

Переходим на первую запись и обновляем информацию в форме IF !BOF()

GO TOP

ENDIF

_screen.ActiveForm.Refresh()

После ввода кода программы закрыть окно.

Ввести аналогично команды для остальных кнопок:

Для кнопки Следующая:

Переходим на следующую запись и обновляем информацию в форме IF !EOF()

SKIP

ENDIF

_screen.ActiveForm.Refresh()

Для кнопки Предыдущая:

Переходим на предыдущую запись и обновляем информацию в форме IF !BOF()

SKIP - 1

ENDIF

_screen.ActiveForm.Refresh()

Для кнопки Последняя:

Переходим на последнюю запись и обновляем информацию в форме

IF !BOF()

GO BOTTOM

ENDIF

_screen.ActiveForm.Refresh()

Для кнопки Выход:

Запрашиваем и выходим, если Да

IFMESSAGEBOX("Выходить из формы?", 4+32+256, "Выход")=6

_screen.ActiveForm.Release()

ELSE

_screen.ActiveForm.Refresh()

ENDIF

После ввода команд закрыть окно процедур

Запустить форму на выполнение. Проверить результат проделанной работы.

Практическая работа № 11. Создание базы данных в среде разработки

Цель работы: овладение навыками изменения табличных файлов при помощи запросов и функций

Обработка запросов в VisualFoxPro

Одним из основных назначений разработанного приложения является быстрый поиск информации в БД и получение ответов на разнообразные вопросы. Для этих целей в VisualFoxPro используются средства, называемые запросами.

При создании запросов в VisualFoxPro разработчик может использовать следующие средства:

- QueryWizard (Мастер запросов)
- QueryDesigner (Конструктор запросов)
- Команда SELECT языка Visual FoxPro

Результатом запроса является таблица, которую можно сохранить в массиве, создаваемой новой таблице, отобразить на экране в режиме Browse (Просмотр) или вывести в виде отчёта.

Для создания запросов можно использовать QueryWizard (Мастер запросов), который последовательно запрашивает наименования таблиц, используемых в запросе, перечень

полей таблиц, критерий упорядочения и условия фильтрации данных. С помощью QueryDesigner (Конструктор запросов) можно формировать различной сложности критерии для выбора записей из нескольких таблиц, над полями можно выполнять различные вычисления.

В верхней части окна QueryDesigner (Конструктор запросов) расположена панель, на которой отображаются используемые в запросе таблицы. Ниже находятся вкладки, предназначенные для выбора полей запроса и формирования условий выборки. Открывая в окне QueryDesigner (Конструктор запросов) необходимые вкладки, можно выполнять следующие действия:

- Выбирать поля результирующей таблицы запроса
- Формировать вычисляемые поля
- Задавать критерии для выборки, группировки и упорядочения данных
- Указывать, куда выводить результаты выборки

Таблица 1. Назначение вкладок окна QueryDesigner (Конструктор запросов)

Вкладка	Назначение
Fields (Поля)	Позволяет указать поля исходных таблиц, выбираемые в результирующий запрос
Join(Объединение)	Позволяет задать условия объединения таблиц
Filter (Фильтр)	Позволяет определить фильтры, накладываемые для выбора записей
OrderBy (Упорядочение)	Позволяет задать критерии упорядочения данных
GroupBy (Группировка)	Позволяет задать условия группировки данных
Miscellaneous (Разное)	Позволяет задать дополнительные условия, такие как признак выборки повторяющихся значений, количество или процент выбора данных

Задание для практической работы

Запустить программу MicrosoftVisualFoxPro.

Открыть проект Информационная система.

Открыть БД Штат проекта.

Создать запрос о сотрудниках, имеющих более одного ребёнка и получающих зарплату менее 9000: в окне ProjectManager щёлкнуть клавишей мыши на вкладке Data/Queries/New/QueryWizard, в появившемся диалоговом окне WizardSelection (Выбор мастера) выбрать пункт QueryWizard (Мастер запросов) и нажать кнопку ОК.

В появившемся диалоговом окне выбора исходной таблицы и полей в области Databasesandtables (Базы данных и таблицы) выбрать необходимую БДштат иуказать таблицу Сотрудники. Из области Availablefields (Имеющиеся поля) выбрать поля Fio и Oklad. Для перехода к следующему шагу нажать кнопку Next(Далее).

В появившемся диалоговом окне выбора условий выборки в области Field(Поле) верхней строки выбрать поле Children (Количество детей) таблицы Сотрудники, в области Operator (Оператор Условия) выбрать условие morethen, в области Value (Выражение) установить значение 1, в нижней строке внести следующие данные соответственно: в области Field(Поле) – поле Oklad (Зарплата), в области Operator (Оператор Условия) – lessthan, в области Value (Выражение) – 9000.

Нажать кнопку Next(Далее).

В появившемся диалоговом окне выбора условия сортировки данных из области Availablefields (Имеющиеся поля) выбрать поле Fio. Нажать кнопку Next(Далее).

Нажать кнопку Preview (Просмотр) и посмотреть результат запроса (запрос будет выглядеть в виде простой таблицы см. рис.11.1.), закрыть таблицу запроса. Нажать кнопку Finish (Готово), в появившемся диалоговом окне SaveAs (Сохранить как) указать необходимую папку и присвоить запросу имя Зарплата дети.



Fio	Oklad
Гуськов	7000.0000
Петросова Г.Н.	4500.0000

Рисунок 11.1. Запрос Зарплата дети

Запустить запрос: в окне ProjectManager щёлкнуть клавишей мыши на вкладке Data/Queries/Зарплата дети/Run 10. Закрыть запрос Зарплата дети.

Создать новый запрос, показывающий количество сотрудников организации, получающих определённую зарплату: в окне ProjectManager щёлкнуть клавишей мыши на вкладке Data/Queries/New/NewQuery. В появившемся диалоговом окне AddTableorView (Добавить таблицу или представление данных) в области Database (База данных) выбрать БД Штат, в области TablesinDatabase таблицу Сотрудники. Нажать кнопку Add, при этом в верхней области окна QueryDesigner (конструктор запросов) (см. рис.11.2.), находящегося позади окна AddTableorView (Добавить таблицу или представление данных) появится выбранная таблица. Нажать кнопку Close (Заккрыть) для закрытия окна AddTableorView.



Рисунок 11.2. Окно QueryDesigner (конструктор запросов)

На вкладке Fields (Поля) окна QueryDesigner (конструктор запросов) в области Availablefields (Имеющиеся поля) выделить поле Сотрудники.oklad, оно автоматически перенесётся в область Functionandexpressions (Функции и выражения), нажать расположенную справа от поля кнопку вызова построителя выражения, откроется диалоговое окно ExpressionBuilder (Построитель выражения) (см. рис.11.3.).

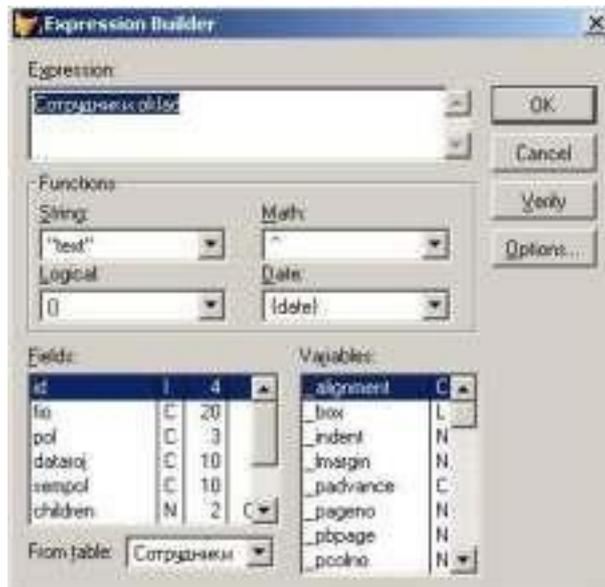


Рисунок 11.3. Диалоговое окно ExpressionBuilder (Построитель выражения)

В поле ввода Expression (Выражение) диалогового окна ExpressionBuilder (Построитель выражения) используя поля таблиц, расположенные в списке Fields (Поля), функции области Functions (Функции), и ввод данных с клавиатуры сформировать следующее выражение: Сотрудники.oklad AS Зарплата.

Для проверки правильности набранного выражения нажать кнопку Verify (Проверить), нажать кнопку OK.

Для переноса созданного выражения из области Functionandexpressions (Функции и выражения) в область SelectedFields (Выбранные поля) нажать находящуюся между данными областями кнопку Add (Добавить).

Руководствуясь предыдущими пунктами 12-15 практической части создать следующее выражение: COUNT(Сотрудники.fio) AS Количество_сотрудников.

Выбрать вкладку OrderBy (Упорядочение) окна QueryDesigner (конструктор запросов) и из области SelectedFields (Выбранные поля) в область Orderingcriteria (Критерий упорядочения) перенести поле Сотрудники.oklad.

Выбрать вкладку GroupBy (Группировка) окна QueryDesigner (конструктор запросов) и из области Availablefields (Имеющиеся поля) в область GroupedFields (Поля группировки) перенести поле Сотрудники.oklad.

Выбрать пункт меню Query (Запрос) окна программы VisualFoxPro, выбрать команду Comments (Комментарии) (см. таблицу 2.). В появившемся окне Comment(Комментарий) в поле ввода области AddComment (Добавить комментарий) ввести строку: Запрос, показывающий количество сотрудников организации, получающих определённую зарплату. Закрыть окно Comment(Комментарий).

Запустить созданный запрос: установить курсор в окне QueryDesigner (конструктор запросов), вызвать контекстное меню, выбрать команду RunQuery, или нажать кнопку [] на панели инструментов. Закрыть запрос.

2. Методические рекомендации по организации образовательного процесса по дисциплине (модулю)

2.1. Методические рекомендации педагогическим работникам Института и (или) лицам, привлекаемым Институтом к реализации образовательных программ на иных условиях

2.1.1. Методические рекомендации по проведению лекций и практических занятий

Особенность преподавания теоретической части дисциплины (модуля) заключается в широком использовании общедидактических методов обучения, основным из которых должен быть выбран метод устного изложения учебного материала. Все лекции должны быть направлены на фундаментальную подготовку обучающихся, обеспечивающую дальнейшую практическую направленность обучения по соответствующему профилю. Поэтому в них основной упор следует делать на сообщение обучающимся специальных знаний, запас которых необходим для решения различных проблем, возникающих как в процессе обучения, так и в будущей практической деятельности в условиях рыночной экономики.

В процессе проведения лекций, наряду с методом монологического изложения материала, необходимо использовать метод рассуждающего (проблемного) изложения. Поэтому педагогическим работникам Института и (или) лицам, привлекаемым Институтом к реализации образовательных программ на иных условиях (далее – педагогический работник, педагогические работники) важно на лекциях активно обращаться к аудитории, как в процессе создания проблемных ситуаций и формулировки проблем, так и в поиске путей их разрешения.

Особенностью преподавания практической части является проведение практических занятий с применением методов показа, совместного выполнения (заданий) упражнений, активного группового взаимодействия. На практических занятиях целесообразно организовывать семинары - дискуссии, деловые игры с разбором конкретных практических ситуаций.

Практические занятия необходимо строить, исходя из потребностей, умения решать типовые и творческие задачи будущей профессиональной деятельности с использованием электронно-вычислительной и другой техники.

Целью проведения практических занятий является углубление теоретических знаний, формирование у обучающихся умений свободно оперировать ими, применять теорию к решению практических задач, и в целом развивать творческое профессиональное мышление обучающихся.

Для углубления теоретических знаний следует осуществлять ориентацию обучающихся на самостоятельное изучение дополнительной

литературы, их участие в научной работе, выполнение НИР отдельными, наиболее подготовленными обучающимися.

Для достижения воспитательных целей учебных занятий необходимо в полной мере использовать возможности содержания дисциплины (модуля), личный пример педагогического работника, индивидуальный подход к обучающимся в образовательном процессе.

2.1.2. Методические рекомендации по проведению интерактивных занятий

Освоение дисциплины (модуля) предполагает использование как традиционных (лекций, практических занятий с использованием методических материалов), так и инновационных образовательных технологий с использованием в учебном процессе интерактивных форм проведения занятий.

Интерактивное обучение — это специальная форма организации познавательной деятельности. Она подразумевает вполне конкретные и прогнозируемые цели. Цель состоит в создании комфортных условий обучения, при которых обучающийся чувствует свою успешность, свою интеллектуальную состоятельность, что делает продуктивным сам процесс обучения, даёт знания и навыки, а также создать базу для работы по решению проблем после того, как обучение закончится.

Другими словами, интерактивное обучение – это, прежде всего, диалоговое обучение, в ходе которого осуществляется взаимодействие между обучающимся и педагогическим работником, между самими обучающимися.

Задачами интерактивных форм обучения являются:

- пробуждение у обучающихся интереса к изучению дисциплины (модуля);
- эффективное усвоение учебного материала;
- самостоятельный поиск обучающимися путей и вариантов решения поставленной задачи (выбор одного из предложенных вариантов или нахождение собственного варианта и обоснование решения);
- установление взаимодействия между обучающимися, обучение работать в команде, проявлять терпимость к любой точке зрения, уважать право каждого на свободу слова, уважать его достоинства;
- формирование у обучающихся мнения и отношения;
- формирование жизненных и профессиональных навыков;
- выход на уровень осознанной компетентности обучающегося.

При использовании интерактивных форм роль педагогического работника резко меняется, перестаёт быть центральной, он лишь регулирует процесс и занимается его общей организацией, готовит заранее необходимые задания и формулирует вопросы или темы для обсуждения в группах, даёт консультации, контролирует время и порядок выполнения намеченного плана. Участники обращаются к социальному опыту – собственному и других людей, при этом им приходится вступать в коммуникацию друг с другом, сов-

местно решать поставленные задачи, преодолевать конфликты, находить общие точки соприкосновения, идти на компромиссы.

Следует обратить внимание на то, что в ходе подготовки занятия на основе интерактивных форм обучения перед педагогическим работником стоит вопрос не только в выборе наиболее эффективной и подходящей формы обучения для изучения конкретной темы, а открывается возможность сочетать несколько методов обучения для решения проблемы, что, несомненно, способствует лучшему осмыслению обучающихся. Представляется целесообразным рассмотреть необходимость использования разных интерактивных форм обучения для решения поставленной задачи.

Принципы работы на интерактивном занятии:

- занятие – не лекция, а общая работа;
- все участники равны независимо от возраста, социального статуса, опыта, места работы;
- каждый участник имеет право на собственное мнение по любому вопросу;
- нет места прямой критике личности (подвергнуться критике может только идея);
- все сказанное на занятии – не руководство к действию, а информация к размышлению.

Алгоритм проведения интерактивного занятия:

1. Подготовка занятия.

Педагогический работник производит подбор темы, ситуации, определение дефиниций (все термины, понятия и т.д. должны быть одинаково поняты всеми обучающимися), подбор конкретной формы интерактивного занятия, которая может быть эффективной для работы с данной темой в данной группе.

При разработке интерактивного занятия рекомендуем обратить особое внимание на следующие моменты:

1) Участники занятия, выбор темы:

- возраст участников, их интересы, будущая специальность;
- временные рамки проведения занятия;
- проводились ли занятия по этой теме в данной учебной группе ранее;
- заинтересованность группы в данном занятии.

2) Перечень необходимых условий:

- должна быть четко определена цель занятия;
- подготовлены раздаточные материалы;
- обеспечено техническое оборудование;
- обозначены участники;
- определены основные вопросы, их последовательность;
- подобраны практические примеры из жизни.

3) Что должно быть при подготовке каждого занятия:

- уточнение проблем, которые предстоит решить;

- обозначение перспективы реализации полученных знаний;
- определение практического блока (чем группа будет заниматься на занятии).

4) Раздаточные материалы:

- программа занятия;
- материал должен быть структурирован;
- использование графиков, иллюстраций, схем, символов.

2. Вступление.

Сообщение темы и цели занятия.

- участники знакомятся с предлагаемой ситуацией, с проблемой, над решением которой им предстоит работать, а также с целью, которую им нужно достичь;

– педагогический работник информирует участников о рамочных условиях, правилах работы в группе, дает четкие инструкции о том, в каких пределах участники могут действовать на занятии;

– при необходимости нужно представить участников (в случае, если занятие межгрупповое);

– добиться однозначного семантического понимания терминов, понятий и т.п. Для этого с помощью вопросов и ответов следует уточнить понятийный аппарат, рабочие определения изучаемой темы. Систематическое уточнение понятийного аппарата сформирует у обучающихся установку, привычку оперировать только хорошо понятными терминами, не употреблять малопонятные слова, систематически пользоваться справочной литературой.

Примерные правила работы в группе:

- быть активным;
- уважать мнение участников;
- быть доброжелательным;
- быть пунктуальным, ответственным;
- не перебивать;
- быть открытым для взаимодействия;
- быть заинтересованным;
- стремиться найти истину;
- придерживаться регламента;
- креативность;
- уважать правила работы в группе.

3. Основная часть.

Особенности основной части определяются выбранной формой интерактивного занятия, и включает в себя:

3.1. Выяснение позиций участников;

3.2. Сегментация аудитории и организация коммуникации между сегментами. Это означает формирование целевых групп по общности позиций каждой из групп. Производится объединение сходных мнений разных участ-

ников вокруг некоторой позиции, формирование единых направлений разрабатываемых вопросов в рамках темы занятия и создается из аудитории набор групп с разными позициями. Затем – организация коммуникации между сегментами. Этот шаг является особенно эффективным, если занятие проводится с достаточно большой аудиторией: в этом случае сегментирование представляет собой инструмент повышения интенсивности и эффективности коммуникации);

3.3. Интерактивное позиционирование включает четыре этапа интерактивного позиционирования:

- 1) выяснение набора позиций аудитории,
 - 2) осмысление общего для этих позиций содержания,
 - 3) переосмысление этого содержания и наполнение его новым смыслом,
 - 4) формирование нового набора позиций на основании нового смысла.
4. Выводы (рефлексия).

Рефлексия начинается с концентрации участников на эмоциональном аспекте, чувствах, которые испытывали участники в процессе занятия. Второй этап рефлексивного анализа занятия – оценочный (отношение участников к содержательному аспекту использованных методик, актуальности выбранной темы и др.). Рефлексия заканчивается общими выводами, которые делает педагогический работник.

Примерный перечень вопросов для проведения рефлексии:

- что произвело на вас наибольшее впечатление?
- что вам помогало в процессе занятия для выполнения задания, а что мешало?
- есть ли что-либо, что удивило вас в процессе занятия?
- чем вы руководствовались в процессе принятия решения?
- учитывалось ли при совершении собственных действий мнение участников группы?
- как вы оцениваете свои действия и действия группы?
- если бы вы играли в эту игру еще раз, чтобы вы изменили в модели своего поведения?

Интерактивное обучение позволяет решать одновременно несколько задач, главной из которых является развитие коммуникативных умений и навыков. Данное обучение помогает установлению эмоциональных контактов между обучающимися, обеспечивает воспитательную задачу, поскольку приучает работать в команде, прислушиваться к мнению своих коллег, обеспечивает высокую мотивацию, прочность знаний, творчество и фантазию, коммуникабельность, активную жизненную позицию, ценность индивидуальности, свободу самовыражения, акцент на деятельность, взаимоуважение и демократичность. Использование интерактивных форм в процессе обучения, как показывает практика, снимает нервную нагрузку обучающихся, дает возможность менять формы их деятельности, переключать внимание на узловые вопросы темы занятий.

Этика педагогического работника включает следующие моменты:

- педагогический работник должен способствовать личному вкладу обучающихся и свободному обмену мнениями при подготовке к интерактивному обучению;
- педагогический работник должен обеспечить дружескую атмосферу для обучающихся и проявлять положительную и стимулирующую ответную реакцию;
- педагогический работник должен облегчать подготовку к занятиям, но не должен сам придумывать аргументы при дискуссиях;
- педагогический работник должен подчеркивать образовательные, а не соревновательные цели обучающихся;
- педагогический работник должен обеспечить отношения между собой и обучающимися, они должны основываться на взаимном доверии;
- педагогический работник должен провоцировать интерес, затрагивая значимые для обучающихся проблемы;
- стимулировать исследовательскую работу;
- заранее подготовить вопросы, которые можно было бы ставить на обсуждение по ходу занятия, чтобы не дать погаснуть дискуссии, обсуждению;
- не допускать ухода за рамки обсуждаемой проблемы;
- обеспечить широкое вовлечение в разговор как можно большего количества обучающихся, а лучше — всех;
- не оставлять без внимания ни одного неверного суждения, но не давать сразу же правильный ответ; к этому следует подключать обучающихся, своевременно организуя их критическую оценку;
- не торопиться самому отвечать на вопросы, касающиеся материала занятия такие вопросы следует переадресовывать аудитории;
- следить за тем, чтобы объектом критики являлось мнение, а не участник, выразивший его;
- проанализировать и оценить проведенное занятие, подвести итоги, результаты. Для этого надо сопоставить сформулированную в начале занятия цель с полученными результатами, сделать выводы, вынести решения, оценить результаты, выявить их положительные и отрицательные стороны;
- помочь участникам занятия прийти к согласованному мнению, чего можно достичь путем внимательного выслушивания различных толкований, поиска общих тенденций для принятия решений;
- принять групповое решение совместно с участниками. При этом следует подчеркнуть важность разнообразных позиций и подходов;
- в заключительном слове подвести группу к конструктивным выводам, имеющим познавательное и практическое значение;
- добиться чувства удовлетворения у большинства участников, т.е. поблагодарить всех обучающихся за активную работу, выделить тех, кто помог в решении проблемы;

- показать высокий профессионализм, хорошее знание материала в рамках учебной программы;
- обладать речевой культурой и, в частности, свободным и грамотным владением профессиональной терминологией;
- проявлять коммуникабельность, а точнее – коммуникативные умения, позволяющие педагогическому работнику найти подход к каждому обучающемуся, заинтересованно и внимательно выслушать каждого, быть естественным, найти необходимые методы воздействия на обучающихся, проявить требовательность, соблюдая при этом педагогический такт;
- обеспечить быстроту реакции;
- способность лидировать;
- уметь вести диалог;
- иметь прогностические способности, позволяющие заранее предусмотреть все трудности в усвоении материала, а также спрогнозировать ход и результаты педагогического воздействия, предвидеть последствия своих действий;
- уметь владеть собой;
- уметь быть объективным.

2.1.3. Методические рекомендации по контролю успеваемости

2.1.3.1. Текущий контроль успеваемости

Текущий контроль уровня освоения содержания дисциплины (модуля) рекомендуется проводить в ходе всех видов учебных занятий методами контроля, предусмотренными рабочей программой дисциплины (модуля).

Качество письменных работ оценивается исходя из того, как обучающиеся:

1. Выбрали и использовали форму и стиль изложения, соответствующие целям и содержанию дисциплины (модуля);
2. Применили связанную с темой информацию, используя при этом понятийный аппарат в соответствующей области;
3. Представили структурированный и грамотно написанный текст, имеющий связное содержание.

Внутрисеместровая аттестация является обязательной формой текущего контроля успеваемости обучающихся очной формы обучения и неотъемлемой частью образовательного процесса и проводится с целью подведения итогов текущей успеваемости обучающихся Института и филиалов.

Проведение внутрисеместровой аттестации по дисциплине (модулю) регулируется локальным нормативным актом Института.

Результаты внутрисеместровой аттестации по дисциплине (модулю) выставляются педагогическим работником в аттестационную ведомость

(система оценки знаний в период внутрисеместровой аттестации – «аттестован», «не аттестован»).

Запись «аттестован» в аттестационную ведомость вносится в случаях, если продемонстрированные обучающимся знания соответствуют оценкам: «отлично», «хорошо», «удовлетворительно». Запись «не аттестован» в аттестационную ведомость вносится в случае, если продемонстрированные обучающимся знания соответствуют оценке «неудовлетворительно», в том числе в случае систематической неявки обучающегося на занятия при отсутствии уважительных причин.

2.1.3.2. Промежуточная аттестация

К промежуточной аттестации допускаются обучающиеся, успешно выполнившие все виды отчетности, предусмотренные рабочей программой дисциплины (модуля). В ходе промежуточной аттестации проверяется степень усвоения материала, умение творчески и последовательно, четко и кратко отвечать на поставленные вопросы, делать конкретные выводы и формулировать обоснованные предложения. Итоговая оценка охватывает проверку достижения всех заявленных целей изучения дисциплины (модуля) и проводится для контроля уровня понимания обучающимися связей между различными ее элементами.

В ходе промежуточной аттестации акцент делается на проверку способностей обучающихся к творческому мышлению и использованию понятийного аппарата дисциплины (модуля) в решении профессиональных задач по соответствующему направлению подготовки.

2.2. Методические указания обучающимся

2.2.1. Методические рекомендации по выполнению самостоятельной работы обучающихся:

СР как вид деятельности обучающихся многогранна. В качестве форм СР при изучении дисциплины (модуля) предлагаются:

- опрос;
- доклад, сообщение.

Задачи СР:

- обретение навыков самостоятельной научно-исследовательской работы на основании анализа текстов литературных источников и применения различных методов исследования;
- выработка умения самостоятельно и критически подходить к изучаемому материалу.

Технология СР должна обеспечивать овладение знаниями, закрепление и систематизацию знаний, формирование умений и навыков.

2.2.2. Рекомендации по работе с научной и учебной литературой

Работа с учебной и научной литературой является главной формой СР и необходима при подготовке к учебным занятиям по дисциплине (модулю). Она включает проработку лекционного материала – изучение рекомендованных источников и литературы по тематике лекций.

Конспект лекции должен содержать реферативную запись основных вопросов лекции, предложенных педагогическим работником схем (при их демонстрации), основных источников и литературы по темам, выводы по каждому вопросу. Конспект должен быть выполнен в отдельной тетради по предмету. Он должен быть аккуратным, хорошо читаемым, не содержать не относящуюся к теме информацию или рисунки.

Конспекты научной литературы при самостоятельной подготовке к занятиям должны быть выполнены также аккуратно, содержать ответы на каждый поставленный в теме вопрос, иметь ссылку на источник информации с обязательным указанием автора, названия и года издания используемой научной литературы. Конспект может быть опорным (содержать лишь основные ключевые позиции), но при этом позволяющим дать полный ответ по вопросу, может быть подробным.

Объем конспекта определяется самим обучающимся. В процессе работы с учебной и научной литературой обучающийся может:

- делать записи по ходу чтения в виде простого или развернутого плана (создавать перечень основных вопросов, рассмотренных в источнике);
- составлять тезисы (цитирование наиболее важных мест статьи или монографии, короткое изложение основных мыслей автора);
- готовить аннотации (краткое обобщение основных вопросов работы);
- создавать конспекты (развернутые тезисы).

Работу с литературой следует начинать с анализа основной и дополнительной литературы, учебно-методических изданиях необходимых для изучения дисциплины (модуля).

Необходимо отметить, что работа с литературой не только полезна как средство более глубокого изучения любой дисциплины (модуля), но и является неотъемлемой частью профессиональной деятельности будущего выпускника.

2.2.3. Методические рекомендации обучающимся по планированию и организации изучения дисциплины (модуля)

Многочисленные исследования бюджета времени обучающихся показывают, что для овладения всеми дисциплинами (модулями), изучаемыми в течение семестра, обучающемуся необходимо самостоятельно заниматься 4-5 часов ежедневно. Особенно важно выработать свой собственный стиль в работе, установить равномерный ритм на весь семестр. Под ритмом понимается

ежедневная работа приблизительно в одни и те же часы, при целесообразности чередования ее с перерывами для отдыха. Правильно организованный, разумный режим работы обеспечит высокую эффективность без существенных перегрузок.

Изучение любой дисциплины (модуля) следует начинать с проработки рабочей программы дисциплины (модуля), особое внимание, уделяя целям и задачам, структуре и содержанию дисциплины (модуля).

При подготовке к занятиям обучающийся должен просмотреть конспекты лекций, рекомендованную литературу по данной теме, подготовиться к ответу на контрольные вопросы.

Успешное изучение любого курса требует от обучающихся посещения лекций, активной работы на практических занятиях, выполнения всех учебных заданий педагогического работника, ознакомления с основной и дополнительной литературой.

В процессе лекционного занятия обучающийся должен выделять важные моменты, выводы, анализировать основные положения. Если при изложении материала педагогическим работником создана проблемная ситуация, необходимо пытаться предугадать дальнейший ход рассуждений. Это способствует лучшему усвоению материала лекции и облегчает запоминание отдельных выводов.

Для более прочного усвоения знаний лекцию необходимо конспектировать. Запись лекции – одна из форм работы обучающихся, требующая навыков и умения кратко, схематично, последовательно и логично фиксировать основные положения, выводы, обобщения, формулировки. В процессе лекции рекомендуется конспектировать только самое важное в рассматриваемой теме: формулировки определений и классификации, выводы и то, что старается выделить лектор. Культура записи лекции – один из важнейших факторов успешного и творческого овладения знаниями. Последующая работа над текстом лекции актуализирует в памяти ее содержание, позволяет развивать аналитическое мышление.

Процесс изучения дисциплин (модулей) учебного плана, как правило, предполагает наличие практических и/или лабораторных занятий.

2.2.4. Методические рекомендации по подготовке обучающихся к контактной работе при проведении учебных занятий по дисциплине (модулю)

2.2.4.1. Методические рекомендации по подготовке обучающихся к лекциям

Подготовка обучающихся к лекциям предполагает:

- работу с имеющимися конспектами лекций;
- чтение основной и дополнительной литературы.

Работу с конспектом лекций лучше начинать с просмотра конспекта в тот же день после занятий и выделения материала конспекта, который вызывает затруднения для понимания. Необходимо найти ответы на затрудни-

тельные вопросы, используя рекомендуемую литературу. Если самостоятельно не удалось разобраться в материале, сформулируйте вопросы и обратитесь за помощью к педагогическому работнику на консультации или ближайшей лекции. Регулярно отводите время для повторения пройденного материала, проверяя свои знания, умения и навыки по контрольным вопросам.

Опыт показывает, что только многообразная, планомерная и целенаправленная обработка лекционного материала обеспечивает его надежное закрепление в долговременной памяти человека. Предсессионный штурм непродуктивен, материал запоминается ненадолго. Необходим систематический труд в течение всего семестра.

2.2.4.2. Методические рекомендации по подготовке обучающихся к практическим занятиям

Практические занятия представляют особую форму сочетания теории и практики. Их назначение – углубление проработки теоретического материала дисциплины (модуля) путем регулярной и планомерной СР на протяжении всего периода изучения дисциплины (модуля). Процесс подготовки к практическим занятиям включает изучение нормативных документов, обязательной и дополнительной литературы по рассматриваемому вопросу.

Практические занятия развивают у обучающихся навыки самостоятельной работы по решению конкретных задач.

При подготовке к практическим занятиям обучающимся рекомендуется: внимательно ознакомиться с тематикой практического занятия; изучить конспект лекции по теме, изучить рекомендованную литературу; составить краткий план ответа на каждый вопрос практического занятия; если встретятся незнакомые термины, обязательно обратиться к словарю и зафиксировать их в конспекте.

2.2.5. Методические рекомендации по составлению плана

План – это схематически записанная совокупность коротко сформулированных мыслей-заголовков. По форме членения и записывания планы могут быть подразделены на простые и сложные. Сложный план в отличие от простого имеет еще и подпункты, которые детализируют или разъясняют содержание частей, основных пунктов. План может быть записан в виде схемы, отражающей взаимосвязь положений.

План представляет собой независимую, самостоятельную форму записи благодаря ряду достоинств:

- краткость записи, что позволяет сравнительно легко переделывать его, совершенствуя как по существу, так и по форме;

- наглядность и обзримость, проявляющиеся в возможности последовательно изложить материал;

- включенность элементов, свидетельствующих об обобщении содержания произведения, что позволит в дальнейшем развить эти положения в тезисах, конспектах, рефератах.

При составлении сложного плана используют два способа работы:

1) разрабатывают подробный простой план, а далее преобразуют его в сложный, группируя части пунктов под общими для них заголовками (основными пунктами сложного плана);

2) составляют краткий простой план и затем, вновь читая текст, преобразуют его в сложный, подыскивая детализирующие пункты. Второй путь требует больших затрат времени и приемлем лишь при продолжительной, заранее запланированной работе.

2.2.6. Методические рекомендации по составлению конспекта

Конспект – это универсальный документ, который должен быть понятен не только его автору, но и другим людям, знакомящимся с ним. Поэтому к конспектам можно с успехом обращаться через несколько (или много) лет после их написания. Основное требование к конспекту отражено уже в его определении – «систематическая, логически связная запись, отражающая суть текста».

В отличие от тезисов, содержащих только основные положения, конспекты при обязательной краткости содержат факты и доказательства, примеры и иллюстрации.

Прежде чем начать конспектировать необходимо уяснить особенности и отличия разных видов конспектов. Конспекты можно условно подразделить на несколько видов.

Вопросно-ответный конспект. Это один из самых простых видов плана-конспекта. На пункты плана, выраженные в вопросительной форме, даются точные ответы.

Этапы работы:

- 1) составьте план прочитанного текста;
- 2) сформулируйте каждый пункт плана в виде вопроса;
- 3) запишите ответы на поставленные вопросы.

Тезисный конспект. Представляет собой сжатый пересказ прочитанного или услышанного. Такой конспект быстро составляется и запоминается; учит выбирать главное, четко и логично излагать мысли, дает возможность усвоить материал еще в процессе его изучения.

Этапы работы:

- 1) составьте план прочитанного текста;
- 2) сформулируйте кратко и доказательно каждый пункт плана в виде тезиса, выберите разумную и эффективную форму записи;
- 3) запишите тезис.

Цитатный конспект. Это конспект, созданный из отрывков подлинника – цитат. При этом цитаты должны быть связаны друг с другом цепью логических переходов.

Цитатный конспект – прекрасный источник дословных высказываний автора, а также приводимых им фактов. Он поможет определить ложность положений автора или выявить спорные моменты, которые значительно труднее найти по пересказу — свободному конспекту.

Этапы работы:

- 1) прочитайте текст, отметьте в нем основное содержание, главные мысли, выделите те цитаты, которые войдут в конспект;
- 2) пользуясь правилами сокращения цитат, вы пишете их в тетрадь;
- 3) прочтите написанный текст, сверьте его с оригиналом;
- 4) сделайте общий вывод.

Свободный (художественный) конспект. Этот вид конспекта представляет собой сочетание выписок, цитат, тезисов. Свободный конспект требует умения самостоятельно четко и кратко формулировать основные положения.

Этапы работы:

- 1) работая с источниками, изучите их и глубоко осмыслите;
- 2) сделайте необходимые выписки основных мыслей, цитат, составьте тезисы;
- 3) используя подготовленный материал, сформулируйте основные положения по теме.

Тематический конспект. Это конспект ответа на поставленный вопрос или конспект учебного материала темы. Составление тематического конспекта учит работать над темой, всесторонне обдумывая ее, анализируя различные точки зрения на один и тот же вопрос.

Большую пользу при составлении тематического конспекта можно извлечь из рабочего каталога, картотеки, наконец, ранее составленных конспектов, других записей.

Можно использовать так называемый обзорный тематический конспект. В этом случае составляется тематический обзор на определенную тему с использованием одного или нескольких источников.

К обзорному тематическому конспекту можно отнести и хронологический конспект, в котором запись подчинена построению в порядке последовательности событий.

Этапы работы:

- 1) изучите несколько источников и сделайте из них выборку материала по определенной теме или хронологии;
- 2) мысленно оформите прочитанный материал в форме плана;
- 3) пользуясь этим планом, кратко, своими словами изложите осознанный материал;
- 4) составьте перечень основных мыслей, содержащихся в тексте, в форме простого плана.

Общие требования ко всем видам конспектов: системность и логичность изложения материала, краткость, убедительность и доказательность.

Приступая к конспектированию внимательно прочитайте текст, отметьте в нем незнакомые вам термины, понятия, не совсем понятные положения, а также имена, даты.

Выберите вид конспекта, который наиболее целесообразен именно для осуществления стоящей перед вами задачи. Познакомьтесь с этапами работы над конспектом и приступайте к их практическому воплощению. Перед тем как сдать работу педагогическому работнику, прочитайте конспект еще раз, при необходимости доработайте его.

2.2.7. Требования к подготовке доклада

Доклад - вид СР, который способствует формированию навыков исследовательской работы, расширяет познавательный интерес, приучает критически мыслить.

При написании доклада по заданной теме составляют план, подбирают основные источники. В процессе работы с источниками систематизируют полученные сведения, делают выводы и обобщения. К докладу по крупной теме могут, привлекаться несколько обучающихся, между которыми распределяются вопросы выступления.

Отличительными признаками доклада являются:

- передача в устной форме информации;
- публичный характер выступления;
- стилевая однородность доклада;
- четкие формулировки и сотрудничество докладчика и аудитории;
- умение в сжатой форме изложить ключевые положения исследуемого вопроса и сделать выводы.