Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Тихон Маргарита Эдуардовна
Должность: проректор по учебной работе
Дата подписания: 05.07.2023 10:35:04
Уникальный программный ключ:
f987416c0d8c88835f6e9819ff6afbbc17571051412566fd3dcfcb557f9085d8

Автономная некоммерческая организация высшего образования

Национальный открытый институт г. Санкт-Петербург

Кафедра математических и естественно-научных дисциплин

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
К ВЫПОЛНЕНИЮ КУРСОВОЙ РАБОТЫ
ПО ДИСЦИПЛИНЕ «БАЗЫ ДАННЫХ»
на тему «Проектирование базы данных в среде Access»

Направление подготовки - 09.03.03 Прикладная информатика

Направленность (профиль) подготовки - Прикладная информатика в экономике

Квалификация (степень) - бакалавр

Форма обучения - заочная

2021

1. ОБЩИЕ УКАЗАНИЯ

Целью курсовой работы является развитие у студентов навыков использования перспективных информационных технологий путем освоения программного продукта Access, входящих в состав Microsoft Office работающих в среде операционной систем Windows.

Отчет по курсовой работе выполняется в виде пояснительной записки, образец титульного листа для которой приведен в приложении 1.

Описание каждого задания должно включать:

1. Лист с заданием.
2. Краткий обзор теоретических основ решения задачи.
3. Описание основных этапов задания.
4. Распечатки базы данных (скрины).
5. Файл Access с выполненным заданием.
6. Список литературы.

Пояснительная записка оформляются в документе Word. Оба файла (пояснительная записка и файл с выполненным заданием в Access) должны быть заархивированы.

Варианты индивидуальных заданий приводятся в п. 2 данных Методических указаний (стр. 2), а подробный пример выполнения курсовой работы – в п.3. указаний (стр. 19).

Библиографический список

1. Агальцов, В.П. Базы данных: учебник в двух книгах / В.П. Агальцов. Издательский дом Форум, 2021. – 71 с. (бакалавриат). Режим доступа: https://znanium.com/catalog/document?id=377105.

2. Базы данных : учеб. пособие / О.Л. Голицына, Н.В. Максимов, И.И. Попов. — 4-е изд., перераб. и доп. — М. : ФОРУМ : ИНФРА-М, 2019. — 400 с. — (Высшее образование: бакалавриат). - Режим доступа: http://znanium.com/catalog/product/1019244

3.. Базы данных : учебник / Л.И. Шустова, О.В. Тараканов. — М. : ИНФРА-М, 2019. — 304 с. + Доп. материалы [Электронный ресурс; Режим доступа: http://www.znanium.com]. — (Среднее профессиональное образование).

4. Мартишин, С.А.Базы данных. Практическое применение СУБД SQL и NO SQL для применения проектирования информационныхтсистем. / С.А. Мартишин и др. - Издательский дом ФОРУМ, 2021. – 368 с. (Высшее образование: Бакалавриат). - Режим доступа: https://znanium.com/catalog/document?id=374126

5. Проектирование современных баз данных: Учебно-методическое пособие / Дадян Э.Г. - М.:НИЦ ИНФРА-М, 2017. - 120 с.: 60x90 1/16 ISBN 978-5-16-106529-7 (online) - Режим доступа: http://znanium.com/catalog/product/959294

2. ЗАДАНИЯ НА КУРСОВУЮ

Вариант 1

Разработать информационно-логическую модель предметной области «Каталог склада деталей» с атрибутами

№ детали	количество деталей № склада
Наименование детали материал	№ секции хранения Кладовщик
завод-изготовитель дата выпуска	Вес детали
дата приобретения стоимость	

Создать логическую структуру этой базы данных для СУБД Access.

Заполнить таблицы базы данных записями.

Создать отчёт с параметрами **Наименование детали, № секции хранения, количество деталей.**

Создать запрос о деталях, хранящихся на определённом складе.

Вариант 2

Разработать информационно-логическую модель предметной области «Оборудование лаборатории» с атрибутами

№ лаборатории	дата покупки
№ комнаты	дата поверки
Инвентарный № прибора	стоимость прибора
Наименование прибора	завод-изготовитель
Модель	габариты (размер, вес)
Серия	запчасти (цена, серия, количество)
дата выпуска	

Создать логическую структуру этой базы данных для СУБД Access.

Заполнить таблицы базы данных записями.

Создать отчёт с параметрами **Наименование прибора, № лаборатории, дата поверки.**

Создать запрос о приборах одной из лабораторий.

Вариант 3

Разработать информационно-логическую модель предметной области «Химические реактивы» с атрибутами

№ лаборатории	компоненты реактива
наименование реактива	дата приобретения
цвет реактива	завод-изготовитель
удельный вес	срок годности реактива
стоимость единицы объёма	режим хранения
количество реактива	Ответственный за хранение

Создать логическую структуру этой базы данных для СУБД Access.

Заполнить таблицы базы данных записями.

Создать отчёт с параметрами **Наименование реактива, № лаборатории, режим хранения.**

Создать запрос о реактивах одной из лабораторий

Вариант 4

Разработать информационно-логическую модель предметной области «Участники конференции» с атрибутами

Код участника	страна	должность
имя	почтовый индекс	дата регистрации
фамилия	домашний телефон	дата отъезда
город	рабочий телефон	секция
район	факс	название доклада
область	место работы	тематика доклада

Создать логическую структуру этой базы данных для СУБД Access.

Заполнить таблицы базы данных записями.

Создать отчёт с параметрами **Код участника, фамилия, название доклада**.

Создать запрос об участниках конференции для одной из секций.

Вариант 5

Разработать информационно-логическую модель предметной области «Рецепты» с атрибутами

№ рецепта	компоненты	срок хранения
код рецепта	инструкции	упаковка
название рецепта	дата изготовления	цена
врач	количество доз	№ аптеки

Создать логическую структуру этой базы данных для СУБД Access.

Заполнить таблицы базы данных записями.

Создать отчёт с параметрами **название рецепта, врач, пациент.**

Создать запрос о рецептах, выданных одним врачом.

Вариант 6

Разработать информационно-логическую модель предметной области «Авторы» с атрибутами

Код автора	образование	почтовый адрес
фамилия	стаж работы	редактор
имя	домашний телефон	отдел издательства
отчество	рабочий телефон	дата поступления рукописи
национальность	факс	число страниц
год рождения	мобильный телефон	дата возвращения на доработку
место рождения		

Создать логическую структуру этой базы данных для СУБД Access.

Заполнить таблицы базы данных записями.

Создать отчёт с параметрами **Код автора, фамилия, дата поступления рукописи**.

Создать запрос об авторах, с которыми работает один из редакторов.

Вариант 7

Разработать информационно-логическую модель предметной области «Клиент» с атрибутами

Код клиента	область	место работы
организация	страна	отдел
имя	почтовый индекс	должность
фамилия	домашний телефон	дата заказа
отчество	рабочий телефон	форма оплаты
город	факс	
район	мобильный телефон	

Создать логическую структуру этой базы данных для СУБД Access.

Заполнить таблицы базы данных записями.

Создать отчёт с параметрами **Код клиента, фамилия, организация, дата заказа**

Создать запрос о клиентах из одного города.

Вариант 8

Разработать информационно-логическую модель предметной области «Книги» с атрибутами

Код книги	место публикации	тип обложки
Код темы	язык	цена
автор	переводчик	число страниц
название	издательство	инвентарный номер
год издания	редактор	
номер ISBN	тираж	

Создать логическую структуру этой базы данных для СУБД Access.

Заполнить таблицы базы данных записями.

Создать отчёт с параметрами **автор, название, инвентарный номер**.

Создать запрос о книгах одного издательства.

Вариант 9

Разработать информационно-логическую модель предметной области «Библиотека» с атрибутами

отдел	название	число страниц
секция	год издания	дата выдачи
код книги	издательство	фамилия читателя
автор	цена	дата возврата

Создать логическую структуру этой базы данных для СУБД Access.

Заполнить таблицы базы данных записями.

Создать отчёт с параметрами **Код книги, фамилия читателя, дата возврата**.

Создать запрос об авторах, книги которых взял один из читателей.

Вариант 10

Разработать информационно-логическую модель предметной области «Исполнители» с атрибутами

Код исполнителя	домашний телефон	название отчёта
Код работы	рабочий телефон	год начала работы
Имя, фамилия, отчество	факс	год окончания работы
город	мобильный телефон	форма оплаты
почтовый индекс	код отчёта	

Создать логическую структуру этой базы данных для СУБД Access.

Заполнить таблицы базы данных записями.

Создать отчёт с параметрами **Код исполнителя, фамилия, форма оплаты**.

Создать запрос об исполнителях одного отчёта.

Вариант 11

Разработать информационно-логическую модель предметной области «Контакты» с атрибутами

Код контакта	название компании	день рождения
фамилия	должность	рекомендован
имя	рабочий телефон	семейное положение
обращение	домашний телефон	имя супруга
город	мобильный телефон	имена детей
область	факс	интересы
страна	электронная почта	

Создать логическую структуру этой базы данных для СУБД Access.

Заполнить таблицы базы данных записями.

Создать отчёт с параметрами **Код контакта, фамилия, название компании**.

Создать запрос о контактах из одного города.

Вариант 12

Разработать информационно-логическую модель предметной области «Адреса» с атрибутами

Фамилия, имя, отчество	адрес электронной почты	факс
страна	сайт в Интернете	мобильный телефон
почтовый индекс	домашний телефон	наименование организации
область	служебный телефон	адрес организации
город, улица, дом, квартира		

Создать логическую структуру этой базы данных для СУБД Access.

Заполнить таблицы базы данных записями.

Создать отчёт с параметрами Фамилия, имя, отчество, сайт в Интернете, факс.

Создать запрос о фамилиях людей из одной организации.

Вариант 13

Разработать информационно-логическую модель предметной области «Инвентарная ведомость» с атрибутами

Отдел	дата покупки
номер комнаты	место покупки
код имущества	завод-изготовитель
название имущества	место получения
количество	посредник
изготовитель	цена покупки
модель	гарантия
инвентарный номер	процент износа
серийный номер	

Создать логическую структуру этой базы данных для СУБД Access.

Заполнить таблицы базы данных записями.

Создать отчёт с параметрами **название имущества, количество, процент износа.**

Создать запрос об имуществе, находящемся в одной из комнат.

Вариант 14

Разработать информационно-логическую модель предметной области «Поставщик» с атрибутами

Код поставщика	страна	наименование поставки
Название фирмы	индекс	модель
Обращаться к (фамилия)	рабочий телефон	цена
Имя	мобильный телефон	дата заказа
должность	факс	дата поставки
город	электронная почта	количество документов
область	код поставки	

Создать логическую структуру этой базы данных для СУБД Access.

Заполнить таблицы базы данных записями.

Создать отчёт с параметрами **Код поставщика, фамилия, название фирмы**, **модель**.

Создать запрос о поставщиках из одного города.

Вариант 15

Разработать информационно-логическую модель предметной области «Товары» с атрибутами

Код товара	Серия
Наименование	Цена
Марка	Количество
Код поставщика	Дата поступления на склад
	Номер складской секции
Код клиента	Номер полки
Единица измерения	Минимальный запас
Год выпуска	Поставка прекращена (дата)
	Ожидается поступление (дата)

Создать логическую структуру этой базы данных для СУБД Access.

Заполнить таблицы базы данных записями.

Создать отчёт с параметрами **Наименование, Серия, Минимальный запас**.

Создать запрос о товаре, хранящемся в одной из секций.

Вариант 16

Разработать информационно-логическую модель предметной области «Заказ» с атрибутами

Код заказа	код клиента	дата доставки к клиенту
Код товара	код агента по продаже	стоимость доставки
Наименование товара	организация-продавец	адрес получателя
количество	дата заказа	фирма-получатель
цена	дата исполнения	телефон и факс получателя

Создать логическую структуру этой базы данных для СУБД Access.

Заполнить таблицы базы данных записями.

Создать отчёт с параметрами **Наименование товара, организация-продавец, фирма-получатель.**

Создать запрос о товарах, доставляемых по заказу одной фирмы.

Вариант 17

Разработать информационно-логическую модель предметной области «Сотрудник» с атрибутами

Отдел	Учебное заведение	страна
Код сотрудника	Учёное звание	индекс
Фамилия сотрудника	Учёная степень	рабочий телефон
Имя	Дата рождения	мобильный телефон
Должность	Место рождения (город)	факс
Образование	область	домашний телефон

Создать логическую структуру этой базы данных для СУБД Access.

Заполнить таблицы базы данных записями.

Создать отчёт с параметрами **Код сотрудника, фамилия, Учёная степень**.

Создать запрос о сотрудниках из одного отдела.

Вариант 18

Разработать информационно-логическую модель предметной области «Сведения о заказе» с атрибутами

Код заказа	количество	Розничная цена
Код товара	Год выпуска	налог
Наименование	Цена	Дата поступления на склад
Марка	Скидка	Дата продажи
Единица измерения	Цена доставки	

Создать логическую структуру этой базы данных для СУБД Access.

Заполнить таблицы базы данных записями.

Создать отчёт с параметрами **Наименование, Марка, Розничная цена.**

Создать запрос о товаре, проданном в конкретный день.

Вариант 19

Разработать информационно-логическую модель предметной области «Оплата» с атрибутами

Код оплаты	Код проекта	Номер карточки
Код заказа	Сумма оплаты	Имя владельца карточки
Код клиента	Дата оплаты	Срок действия карточки
Код участника	Метод оплаты	Код карточки
Код регистрации	Тип карточки	Условия оплаты
		Код метода оплаты

Создать логическую структуру этой базы данных для СУБД Access.

Заполнить таблицы базы данных записями.

Создать отчёт с параметрами **Код клиента, Сумма оплаты, Метод оплаты.**

Создать запрос о заказах, оплаченных по карточке.

Вариант 20

Разработать информационно-логическую модель предметной области «Доставка товара» с атрибутам

Код доставки	Телефон доставки
Код клиента	адрес доставки (улица, дом)
Код заказа	город доставки
Код сотрудника	район доставки
Способ доставки	индекс доставки
Дата заказа	цена доставки
Дата доставки	

Создать логическую структуру этой базы данных для СУБД Access.

Заполнить таблицы базы данных записями.

Создать отчёт с параметрами **Код сотрудника, Способ доставки, цена доставки.**

Создать запрос о доставках в один город.

Вариант 21

Разработать информационно-логическую модель предметной области «Основные фонды» с атрибутами

Код основных фондов Штрих-код	Срок амортизации
Название фонда Дата приобретения	Сборка Срок гарантии
Код сотрудника Дата продажи	Модель Сумма страховки
Код отдела Цена покупки	Номер модели Текущая стоимость
Код продавца	Серийный номер

Создать логическую структуру этой базы данных для СУБД Access.

Заполнить таблицы базы данных записями.

Создать отчёт с параметрами **Модель, Серийный номер, Текущая стоимость.**

Создать запрос о моделях, находящихся в одном из отделов.

Вариант 22

Разработать информационно-логическую модель предметной области «Каталог картин» с атрибутами

-автор
-название картины
-год создания
-страховая стоимость
-зал экспозиции
-время работы экспозиции
-название музея
-город, где находится музей
-страна, где находится музей
-год реставрации
-место нахождения реставрационной мастерской
-реставратор

Создать логическую структуру этой базы данных для СУБД Access.

Заполнить таблицы базы данных записями.

Создать отчёт с параметрами ***Автор, Название картины, Название музея.***

Создать запрос о картинах одного из авторов.

Вариант 23

Разработать информационно-логическую модель предметной области Ремонтная мастерская» с атрибутами

Марка автомобиля
Регистрационный номер
Год выпуска
Страна-производитель
Процент износа
Тип поломки
Категория сложности ремонта
Стоимость ремонта
Гарантийный срок ремонта
Номер ремонтной бригады
Фамилия бригадира ремонтников
Дата постановки на ремонт
Срок окончания ремонта

Создать логическую структуру этой базы данных для СУБД Access.

Заполнить таблицы базы данных записями.

Создать отчёт с параметрами ***Марка автомобиля, Тип поломки, Номер ремонтной бригады.***

Создать запрос об автомобилях одной из марок.

Вариант 24

Разработать информационно-логическую модель предметной области «Ремонт телевизоров» с атрибутами

Марка телевизора
Регистрационный номер
Год выпуска
Страна-производитель
Тип кинескопа
Цветной или чёрно-белый
Процент износа
Тип поломки
Категория сложности ремонта
Стоимость ремонта
Гарантийный срок ремонта
Номер ремонтной бригады
Фамилия бригадира ремонтников
Дата постановки на ремонт
Срок окончания ремонта

Создать логическую структуру этой базы данных для СУБД Access.

Заполнить таблицы базы данных записями.

Создать отчёт с параметрами ***Марка телевизора, Тип поломки, Номер ремонтной бригады.***

Создать запрос о телевизорах одной из марок.

Вариант 25

Разработать информационно-логическую модель предметной области «Ремонт холодильников» с атрибутами

- Марка холодильника
- Регистрационный номер
- Год выпуска
- Страна-производитель
- Число морозильных камер
- Объём морозильника
- Процент износа
- Тип поломки
- Категория сложности ремонта
- Стоимость ремонта
- Гарантийный срок ремонта
- Фамилия бригадира ремонтников
- Дата постановки на ремонт
- Дата окончания ремонта

Создать логическую структуру этой базы данных для СУБД Access.

Заполнить таблицы базы данных записями.

Создать отчёт с параметрами ***Марка холодильника, Тип поломки, Фамилия бригадира ремонтной бригады.***

Создать запрос о холодильниках одной из марок.

Вариант 26

Разработать информационно-логическую модель предметной области «Ремонт компьютеров» с атрибутами

Марка компьютера
Год выпуска
Страна-производитель
Тип процессора
Объём винчестера
Оперативная память Процент износа
Тип поломки Категория сложности ремонта
Стоимость ремонта Гарантийный срок ремонта
Дата поступления в ремонт Дата окончания ремонта

Создать логическую структуру этой базы данных для СУБД Access.
Заполнить таблицы базы данных записями.
Создать отчёт с параметрами Марка компьютера, Тип процессора, Тип поломки.
Создать запрос о компьютерах одной из марок.

Вариант 27

Разработать информационно-логическую модель предметной области «Торговая организация» с атрибутами

1. Финансовый отдел:
2. Отдел сбыта:
    2.1. Отдел учёта
    2.2. Отдел розничной торговли
    2.3. Отдел финансового планирования
    2.4. Отдел оптовой торговли
    2.5. Бухгалтерия
3. Отдел логистики:
    3.1. Отдел поставок
    3.2. Склад

(*в каждом отделе предусмотреть информацию о числе сотрудников и номере комнаты*).

Создать логическую структуру этой базы данных для СУБД Access.
Заполнить таблицы базы данных записями.
Создать отчёт с параметрами ***Отдел учёта, Отдел поставок, Склад.***
Создать запрос о числе сотрудников в бухгалтерии.

Вариант 28

Разработать информационно-логическую модель предметной области «Торговое предприятие» с атрибутами

Торговый филиал (наименование, адрес)

Менеджер по продажам (Индивидуальный номер, фамилия, имя, отчество, стаж)

Заказ (код заказа, товар, адрес доставки)

Клиент (Код клиента, наименование фирмы, фамилия заказчика, адрес)

Создать логическую структуру этой базы данных для СУБД Access.

Заполнить таблицы базы данных записями.

Создать отчёт с параметрами ***Адрес филиала, Фамилия, Имя, отчество менеджера.***

Создать запрос о клиенте по его коду.

Вариант 29

Разработать информационно-логическую модель предметной области «Принтеры» с атрибутами

Фирма – изготовитель
Модель
Цветной или чёрно-белый
Технология печати
Скорость печати
Разрешение
Ресурс
Цена картриджа
Цена принтера

Создать логическую структуру этой базы данных для СУБД Access.

Заполнить таблицы базы данных записями.

Создать отчёт с параметрами ***Фирма – изготовитель, Модель, Ресурс***.

Создать запрос о цветных принтерах.

Вариант 30

Разработать информационно-логическую модель предметной области «Компьютеры» с атрибутами

Фирма-изготовитель
Страна - изготовитель
Модель
Год выпуска
Процессор
Оперативная память
CD –ROM
Твёрдый диск
Цена
Тип монитора
Марка принтера
Электронная почта
Подключение к Интернету

Создать логическую структуру этой базы данных для СУБД Access.

Заполнить таблицы базы данных записями.

Создать отчёт с параметрами ***Фирма – изготовитель, Модель, Процессор.***

Создать запрос о компьютерах с определённым типом монитора.

Вариант 31

Разработать информационно-логическую модель предметной области «Запчасти» с атрибутами

Запчасть (маркировка запчасти, № секции хранения, количество запчастей, стоимость запчасти, № запчасти)

Эксплуатационный материал (маркировка материала, № секции хранения, единица измерения, стоимость материала, количество)

Заказ (вид заказанной запчасти, № запчасти, кто заказал, количество заказанных запчастей, дата заказа)

Накладная (№ накладной, № заказа, вид запчастей, количество, стоимость)

Создать логическую структуру этой базы данных для СУБД Access.

Заполнить таблицы базы данных записями.

Создать отчёт с параметрами ***количество запчастей, количество заказанных запчастей, № накладной.***

Создать запрос о количестве запчастей с нужным номером.

Вариант 32

Разработать информационно-логическую модель предметной области «Маршрутная карта АТП» с атрибутами

Фамилия водителя	Марка автомобиля
Имя	Пробег
Отчество	Пункт назначения
Домашний телефон	Расстояние
Класс	Время выезда
Домашний адрес	Время в пути

Создать логическую структуру этой базы данных для СУБД Access.

Заполнить таблицы базы данных записями.

Создать отчёт с параметрами **Фамилия водителя, Марка автомобиля**.

Создать запрос о водителе, направленном в конкретный пункт назначения.

Вариант 33

Разработать информационно-логическую модель предметной области «Гараж» с атрибутами

Фамилия водителя Марка машины
Имя Номер машины
Отчество Позывные машины
Стаж работы Пробег машины
Категория Номер бокса машины в гараже
Телефон Число занятых мест в гараже
Зарплата Число свободных мест в гараже

Создать логическую структуру этой базы данных для СУБД Access.

Заполнить таблицы базы данных записями.

Создать отчёт с параметрами **Фамилия водителя, Номер машины, Номер бокса машины в гараже.**

Создать запрос о водителях с определённой категорией.

Вариант 34

Разработать информационно-логическую модель предметной области «Автобусный парк» с атрибутами

Марка автобуса	Фамилия, имя водителя
Регистрационный номер	Категория водителя
Год выпуска	Стаж работы
Страна-производитель	Оклад
Число посадочных мест	Номер маршрута
Процент износа	Протяжённость маршрута
Гарантийный срок ремонта	Интервал движения

Создать логическую структуру этой базы данных для СУБД Access.

Заполнить таблицы базы данных записями.

Создать отчёт с параметрами **Марка автобуса, Фамилия, имя водителя, Номер маршрута.**

Создать запрос о водителях одного из маршрутов.

Вариант 35

Разработать информационно-логическую модель предметной области «Отдел кадров предприятия» с атрибутами

Цех	Специальность	Стаж работы в должности
Участок	Должность	Разряд по ЕТС
Фамилия, имя сотрудника	Общий стаж работы	Оклад
Образование	Стаж работы на предприятии	Семейное положение
		Дети (имена, возраст)

Создать логическую структуру этой базы данных для СУБД Access.

Заполнить таблицы базы данных записями.

Создать отчёт с параметрами **Цех, Фамилия, имя сотрудника, Оклад, Семейное положение**

Создать запрос о сотрудниках, имеющих общий стаж более десяти лет.

Вариант 36

Разработать информационно-логическую модель предметной области «Бухгалтерия НИИ» с атрибутами

Отдел	учёная степень	Надбавка за степень
Лаборатория	Специальность	Надбавка за звание
Фамилия, имя сотрудника	Должность	Персональная надбавка
Образование	Общий стаж	Льготы
Учебное заведение	Стаж работы в НИИ	Налог
Учёное звание	Оклад	Зарплата

Создать логическую структуру этой базы данных для СУБД Access.

Заполнить таблицы базы данных записями.

Создать отчёт с параметрами **Отдел, Фамилия, имя сотрудника, Зарплата**.

Создать запрос о сотрудниках, имеющих персональные надбавки.

Вариант 37

Разработать информационно-логическую модель предметной области «Билетная касса» с атрибутами

Номер поезда	Время в пути
Категория (скорый, пассажирский)	Наличие прицепных вагонов
Дни отправления	Число свободных мест в купе
Число вагонов	Номера свободных мест в купе
Число купейных вагонов	Число свободных мест в плацкарте
Общее число посадочных мест	Номера свободных мест в плацкарте
Станция назначения	Число свободных мест общих
Время отправления	Номера свободных мест общих

Создать логическую структуру этой базы данных для СУБД Access.

Заполнить таблицы базы данных записями.

Создать отчёт с параметрами **Номер поезда, Станция назначения, Номера свободных мест в купе.**

Создать запрос о числе свободных мест в купе конкретного поезда.

Вариант 38

Разработать информационно-логическую модель предметной области «Сортировочная станция» с атрибутами

Участок	Груз в отцепляемых вагонах
Номер бригады	Число прицепляемых вагонов
Фамилия, имя бригадира	Груз в прицепляемых вагонах
Число работников в бригаде	Число цистерн в составе
Номер состава	Число отцепляемых цистерн
Станция назначения	Груз в отцепляемых цистернах
Число вагонов в составе	Число прицепляемых цистерн
Число отцепляемых вагонов	Груз в прицепляемых цистернах

Создать логическую структуру этой базы данных для СУБД Access.

Заполнить таблицы базы данных записями.

Создать отчёт с параметрами **Участок, Номер состава, Число цистерн в составе.**

Создать запрос о грузе в отцепляемых вагонах конкретного состава.

Вариант 39

Разработать информационно-логическую модель предметной области «Банк» с атрибутами

Отдел	Стаж работы в банке
Секция	Оклад
Фамилия, имя сотрудника	Персональная надбавка
Образование	Льготы
Учебное заведение	Налог
Специальность	Зарплата
Должность	С каким предприятием работает
Общий стаж	Номер счёта предприятия

Создать логическую структуру этой базы данных для СУБД Access.

Заполнить таблицы базы данных записями.

Создать отчёт с параметрами **Отдел, Фамилия, имя сотрудника, Зарплата**.

Создать запрос о сотрудниках, работающих с конкретным предприятием.

3. МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ВЫПОЛНЕНИЮ КУРСОВОЙ РАБОТЫ

3.1. Проектирование баз данных (БД)

3.1.1. Основные понятия

В широком смысле слова ***база данных*** – это совокупность сведений о конкретных объектах реального мира в какой-либо области.

Создавая базу данных, пользователь стремится упорядочить информацию по различным признакам и быстро извлекать выборку с произвольным сочетанием признаков. Сделать это возможно, только если данные структурированы. ***Структурирование*** – это введение соглашений о способах представления данных. Неструктурированными называют данные, записанные, например, в текстовом файле.

На рис. 1 приведен пример неструктурированных данных, содержащих сведения о студентах (номер личного дела, фамилию, имя, отчество и год рождения). Понятно, что организовать поиск необходимых данных, хранящихся в неструктурированном виде сложно, а упорядочить подобную информацию практически не представляется реальным.

Личное дело №16495, Сергеев Петр Михайлович, дата рождения 1 января 1976г.; Л/д №16593, Петрова Анна Владимировна, дата рожд.15 марта 1975г.; № личн. дела 16693, д.р. 14.04.76, Анохин Андрей Борисович

Рис. 1

Чтобы автоматизировать поиск и систематизировать эти данные, необходимо выработать определенные соглашения о способах представления данных, т.е. дату рождения нужно записывать одинаково для каждого студента, она должна иметь одинаковую длину и определенное место среди остальной информации. Эти же замечания справедливы и для остальных данных (номер личного дела, фамилия, имя, отчество). Структурированная запись данных приведена в табл. 1.

Таблица 1

№ личного дела	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

Итак, **база данных** (БД) – это поименованная совокупность структурированных данных, относящихся к определенной предметной области.

Система управления базами данных (СУБД) – это комплекс программных и языковых средств, необходимых для создания баз данных, поддержания их в актуальном состоянии и организации поиска в них необходимой информации.

3.1.1.1. Структурные элементы базы данных

Понятие базы данных тесно связано с такими понятиями, как **поле, запись, отношение, файл, ключ**.

Поле – элементарная единица логической организации данных. В двумерной таблице **поле** – один из столбцов таблицы. При создании БД нужно предварительно описать все поля. Для описания поля используются следующие характеристики:

имя (например, **Фамилия**, **Имя**, **Отчество**, **Дата рождения** – т. е. название столбца таблицы);

тип (например, символьный, числовой, календарный – в памяти компьютера разный тип информации кодируется различным образом. Для поля **Фамилия** задается символьный тип информации, для поля **Дата рождения** – календарный);

длина (например, 15 символов, причем длина будет определяться максимально возможным количеством символов в данном поле и служит для предварительного задания ширины столбца. Так для поля **Фамилия** достаточно задать длину 15 символов, поскольку большинство фамилий не содержит более 15 букв, а для столбца **Пол** достаточно одного символа, так как туда будет вводится признак пола – одна буква “м” или буква “ж”);

точность – необходимо задавать ее только для числовых данных (например, два десятичных знака для отображения дробной части числа 15.28);

Запись (рис. 2) – совокупность логически связанных полей. Экземпляр записи – отдельная реализация записи, содержащая конкретные значения ее полей т.е. одна строка двумерной таблицы. (В табл. 1 один экземпляр записи характеризует студента Сергеева, второй – Петрова).

Имя поля 1	Имя поля 2	Имя поля 3

Запись

Рис. 2

Отношение – совокупность однородных записей (одна таблица, например, табл. 1).

Файл – совокупность объектов одной базы данных.

Обычно база данных включает следующие объекты:

Таблицы (для хранения данных).

Запросы (для отбора информации).

Формы (для просмотра, ввода и вывода информации).

Отчеты (для подготовки информации к печати).

Ключ – это поле, по которому можно однозначно найти любую запись в БД. В табл. 58 ключом является **№ личного дела**. Если, например, выбрать в

качестве ключа поле **Фамилия**, то однозначность поиска нужной записи о студенте не будет обеспечена, т.к. в любой достаточно большой БД обязательно окажется несколько студентов с фамилией Иванов, Смирнов и т.д.

Описание **логической структуры** записи файла содержит последовательность расположения полей записи и их основные характеристики, как это показано в табл. 2.

Таблица 2

Имя файла				
Поле	Признак ключа	Формат поля		
Имя Полное (обозн.) наименов.		Тип	Длина	Точность (для чисел)
Имя 1				
...				
Имя n				

Как видно из табл. 2, кроме уже известных характеристик, в структуре записи файла указываются поля, значения которых являются ключами.

В табл. 3 приведен пример описания логической структуры записи файла (таблицы) СТУДЕНТ, содержимое которого приводится на рис.1.

Таблица 3

Имя файла					
Поле		Признак ключа	Формат поля		
Обозначение в БД	Полное наименование		Тип	Длина	Точность (для чисел)
Номер	№ личн. дела	ключ	Числ.	5	
Фамилия	Фамилия студ		Текст.	15	
Имя	Имя студента		Текст.	10	
Отчество	Отчество студ		Текст.	15	
Дата	Дата рождения		Дата	8	

3.1.1.2. Реляционная модель данных

Существует несколько типов (моделей) организации БД: иерархический, сетевой, реляционный. В настоящее время наиболее распространенной является реляционная модель.

Понятие **реляционный** (англ. relation – отношение) связано с разработ-ками известного американского специалиста в области баз данных Е. Кодда. Эти модели характеризуются простотой структуры данных, удобным для пользователя табличным представлением и возможностью использования формального аппарата алгебры отношений и реляционного исчисления для обработки данных.

Реляционная модель представляет собой организацию данных в виде двумерных 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 одинаковые 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.

Наиболее популярные СУБД (Paradox, FoxPro, Access, Base) основаны на использовании реляционной модели данных.

Реляционной таблицей можно представить информацию о студентах, обучающихся в вузе (табл. 4)

Таблица 4

№ 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

Таблицы БД в реляционных моделях называются **отношениями**, строки соответствуют **записям**, а столбцы – атрибутам отношений (**полям**). Поле, каждое значение которого однозначно определяет соответствующую запись, называется **простым ключом** (**ключевым полем**). Если записи можно однозначно определить только значениями нескольких полей, то такая таблица базы данных имеет **составной ключ**. В примере, показанном в табл. 3, простым ключевым полем таблицы является "**№ личного дела**". Чтобы связать две реляционные таблицы, необходимо ключ первой таблицы ввести в состав ключа второй таблицы (возможно совпадение ключей).

На рис. 3 показан пример реляционной модели, построенной на основе отношений (таблиц): СТУДЕНТ, СЕССИЯ, СТИПЕНДИЯ.

Таблицы, входящие в состав модели рис. 3 могут включать, например, следующие поля:

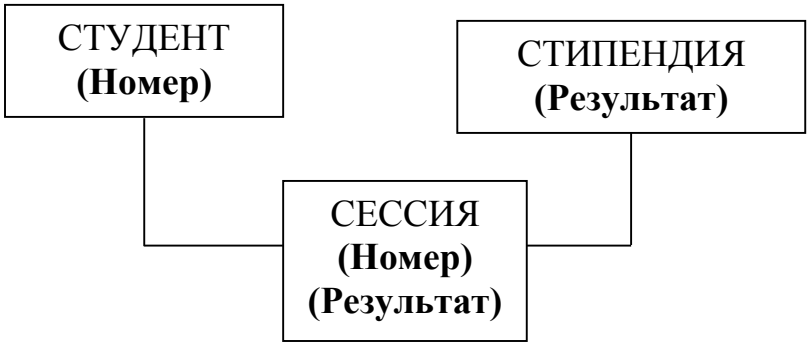


Рис. 3

СТУДЕНТ (**Номер**, Фамилия, Имя, Отчество, Пол, Дата рождения, Группа);
СЕССИЯ (**Номер**, Оценка1, Оценка2, Оценка3, Оценка4, **Результат**);
СТИПЕНДИЯ (**Результат**, Процент).

Выделенные поля в описанных таблицах являются ключевыми. Поле **Номер** – это номер личного дела студента. Поле **Результат** – средний балл студента за сессию.

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.

3.1.1.3. Нормализация отношений

Одни и те же данные могут группироваться в таблицы (отношения) различными способами. Возможная организация различных наборов отношений должна быть рациональной, т.е. исключающей дублирование данных и упрощающей процедуры их обработки и обновления.

Нормализация отношений – формальный аппарат ограничений на формирование отношений (таблиц), который позволяет устранить дублирование, обеспечивает непротиворечивость хранимых в базе данных, уменьшает трудозатраты на ведение (ввод, корректировку) базы данных. Е. Коддом выделены три нормальные формы отношений и предложен механизм, позволяющий любое отношение преобразовать к третьей (самой совершенной) нормальной форме.

Первая нормальная форма

Отношение называется нормализованным или приведенным к первой нормальной форме, если все его атрибуты простые (далее неделимы). Преобразование отношения к первой нормальной форме может привести к увеличению количества реквизитов (полей) таблицы и изменению ключа.

Например, отношение Студент (**Номер**, Фамилия, Имя, Отчество, Дата, Группа) находится в первой нормальной форме. (Каждое поле однозначно описывает какую-то характеристику студента. Ни одно поле не требует дополнительных разъяснений). Отношение Студент1 (**Номер**, Фамилия, Имя, Отчество, Группа, Дети) не находится в первой нормальной форме, т.к. студент может иметь несколько детей и тогда об одном и том же студенте придется ввести в таблицу несколько записей. Например, пусть студент Иванов Иван Иванович, номер личного дела 11111 учится в 215 группе и имеет двоих детей – Максима и Ольгу. Тогда таблица с записью о нем будет иметь вид табл. 5

.

Таблица 5

Номер	Фамилия	Имя студента	Отчество	Группа	Имя ребенка	Возраст ребенка
11111	Иванов	Иван	Иванович	215	Максим	10
11111	Иванов	Иван	Иванович	215	Ольга	3

Чтобы привести отношение СТУДЕНТ1 к первой нормальной форме, нужно разбить эту таблице на две:

СТУДЕНТ1 (**Номер**, Фамилия, Имя, Отчество, Группа),

ДЕТИ (**Номер**, Имя ребенка, Возраст ребенка).

Теперь каждая из таблиц СТУДЕНТ1 и ДЕТИ не содержат повторной информации, т.е. БД приведена к первой нормальной форме.

Вторая нормальная форма

Чтобы рассмотреть вопрос приведения отношений ко второй нормальной форме, необходимо дать пояснения к таким понятиям, как функциональная зависимость и полная функциональная зависимость.

Функциональная зависимость реквизитов – зависимость, при которой в одном экземпляре информационного объекта определенному значению ключа соответствует только одно значение описательного реквизита. В случае составного ключа вводится понятие функционально полной зависимости. Функциональная полная зависимость неключевых атрибутов заключается в том, что каждый неключевой атрибут функционально зависит от ключа, но не находится в функциональной зависимости ни от какой части составного ключа.

Отношение будет находиться во второй нормальной форме, если оно находится в первой нормальной форме, и каждый неключевой атрибут функционально полно зависит от составного ключа.

Отношение Студент (**Номер**, Фамилия, Имя, Отчество, Дата, Группа) находится в первой и во второй нормальной форме одновременно, так как описательные реквизиты однозначно определены и функционально зависят от ключа **Номер**.

Отношение Успеваемость (**Номер**, Фамилия, Имя, Отчество, **Дисциплина,** оценка) находится в первой нормальной форме и имеет составной ключ **Номер** + **Дисциплина**. Это отношение не находится во второй нормальной форме, так как атрибуты Фамилия, Имя, Отчество не находится в полной функциональной зависимости с составным ключом отношения.

Чтобы привести это отношение ко второй нормальной форме, следует разбить его на два: Успеваемость (**Номер**, Фамилия, Имя, Отчество, **Дисциплина**) и Дисциплина (**Дисциплина**, Оценка).

Третья нормальная форма

Понятие третьей нормальной формы основывается на понятии нетранзитивной зависимости. Транзитивная зависимость наблюдается в том случае, если один из двух описательных реквизитов зависит от ключа, а другой описательный реквизит зависит от первого описательного реквизита. Отношение будет находиться в третьей нормальной форме, если оно находится во второй нормальной форме, и каждый неключевой атрибут нетранзитивно зависит от первичного ключа.

Если в состав описательных реквизитов информационного объекта Студент включать фамилию старосты группы (Староста), которая определяется только номером группы, то одна и та же фамилия старосты будет многократно повторяться в разных экземплярах данного информационного объекта. В этом случае наблюдаются затруднения в корректировке фамилии старосты в случае назначения нового старосты, а также неоправданный расход памяти для хранения дублированной информации.

Для устранения транзитивной зависимости описательных реквизитов необходимо провести "расщепление" исходного информационного объекта. В результате расщепления часть реквизитов удаляется из исходного информационного объекта и включается в состав других (возможно, вновь созданных) информационных объектов.

"Расщепление" информационного объекта, содержащего транзитивную зависимость описательных реквизитов, показано на рис. 4. Как видно из рис.4, исходный информационный объект Студент группы представляется в виде совокупности правильно структурированных информационных объектов (Студент и Группа), реквизитный состав которых тождественен исходному объекту. Отношение Студент (Номер, Фамилия, Имя, Отчество, Дата, Группа) находится одновременно в первой, второй и третьей нормальной форме.

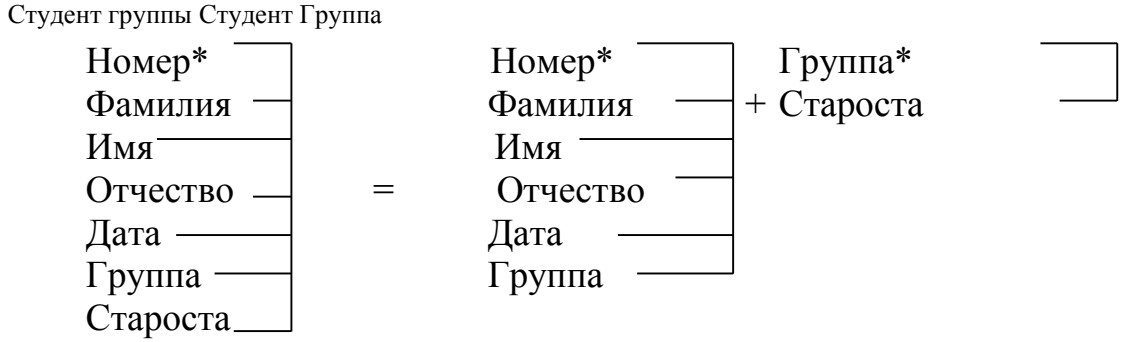


Рис. 4

3.1.1.4. Типы связей

Все информационные объекты предметной области связаны между собой. Различаются связи нескольких типов, для которых введены следующие обозначения:

- один к одному (1:1);
- один ко многим (1:M) или (1:∞);
- многие ко многим (M:M) или (∞:∞).

Рассмотрим эти типы связей на примере.

Дана совокупность информационных объектов, отражающих учебный процесс в вузе:

СТУДЕНТ (**Номер**, Имя, Отчество, Пол, Дата рождения, Группа),

СЕССИЯ (**Номер**, Оценка1, Оценка2, Оценка3, Оценка4, **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. 5 иллюстрирует указанный тип отношений.

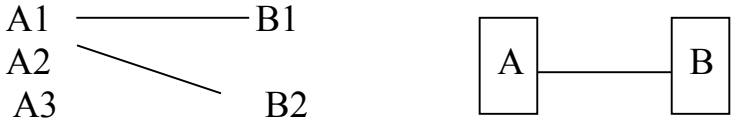


Рис. 5

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 объект А соответствует 0, 1 или больше 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. 6.

Рис. 6

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 со многими (M:M) предполагает, что в каждый момент времени одному экземпляру информационного объекта A соответствует 0, 1 или более экземпляров объекта B и наоборот.

На рис. 7 графически представлено указанное соответствие.

Рис. 7

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

СТУДЕНТ ⟸⟹ ПРЕПОДАВАТЕЛЬ

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

3.1.2. Основные этапы проектирования, создания и ведения баз данных

Задание курсовой работы по проектированию баз данных рекомендуется выполнять в соответствии со следующими этапами (обычный порядок проектирования баз данных).

1. Обследование предметной области и выявление объектов, сведения о которых должны храниться в базе данных (перечень атрибутов).
2. Выбор системы управления базами данных.
3. Конструирование схемы базы данных (разработка информационно-логической модели БД).
4. Создание схем базы данных для машинных носителей.
5. Заполнение таблиц данными (не менее 10 записей).
6. Реализация запросов. Составление программ реализации запросов. Проверка запросов на контрольном примере.
7. Составление инструкции по технологии ведения базы данных.

3.1.3. Пример выполнения курсовой работы по проектированию баз данных

Постановка задачи. Пусть требуется разработать базу данных, в которой бы хранились данные о результатах сдачи экзаменов каждым студентом одного факультета за весь период обучения.

Назовем такую базу данных "Студент".

3.1.3.1. Отбор атрибутов, сведения о которых должны храниться в БД

В соответствии с заданием, в БД должны храниться следующие сведения о студентах:

- номер зачетной книжки;
- фамилия;
- имя;
- отчество;
- дата рождения;
- пол;
- шифр специальности;
- название специальности;
- курс;
- номер семестра;
- название дисциплины;
- название кафедры;
- число учебных часов по дисциплине;
- оценка;
- дата сдачи экзамена.

3.1.3.2. Выбор системы управления базами данных (СУБД).

В последнее время широкое распространение приобрела СУБД Access, входящая в состав Microsoft office. Она реализует реляционную модель данных.

Краткие сведения о СУБД Access

База данных в Access представляет собой единый большой объект, который объединяет такие составляющие, как таблицы, отчеты, запросы, формы и т.д. и позволяет хранить их в одном файле.

Основным структурным элементом БД является таблица. Каждая таблица содержит записи определенного вида, например, о студентах, обучающихся в учебном заведении.

Каждый объект и элемент управления имеет свои свойства, определяя которые можно настраивать объекта и элементы управления.

Таблицы создаются пользователем для хранения данных по одному объекту модели данных предметной области.

Запросы создаются пользователем для выборки нужных данных из одной или нескольких связанных таблиц. С помощью запроса можно также обновить, удалить или добавить данные в таблицы или создать новые таблицы на основе уже существующих.

Формы предназначены для ввода, просмотра и корректировки взаимосвязанных данных базы на экране в удобном виде, который может соответствовать привычному для пользователя документу. Формы также могут использоваться для создания панелей управления в приложении пользователя.

Отчеты предназначены для формирования выходного документа, предназначенного для вывода на печать.

Создание таблицы БД

Создание таблицы в БД состоит из 2-х этапов. На первом этапе определяется ее структура: состав полей, их имена, последовательность размещения в таблице, тип данных, размер, ключевые поля и другие свойства полей. На втором этапе производится создание записей таблицы и заполнение их данными.

Для создания таблицы надо в окне БД выбрать вкладку **Создание** и нажать группа **Таблицы, з**атем выбрать способ создания таблицы: **Режим таблицы, Конструктора** или **Мастера таблиц**. Режим **Конструктора** определяет основной способ, при котором структура таблицы полностью задается пользователем.

Для определения поля задаются **Имя поля, Тип поля, Описание**, а также **Общие свойства.**

Имя поля. Каждое поле в таблице должно иметь уникальное имя, удовлетворяющее соглашениям об именах объектов.

Тип данных. Тип данных определяется значениями, которые предполагается вводить в поле, и операциями, которые будут выполняться с этими значениями.

В БД допускается использование следующих типов данных: текстовый (текст), МЕМО (текст большого размера), числовой (число), денежный, дата/время (Дата, Время), счетчик, логический, поле объекта OLE…

Общие свойства поля:

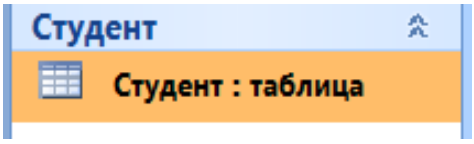
- Размер поля (Длина) – задает максимальный размер данных, сохраняемых в поле.
- Формат поля (Пример формата) – задает формат отображения данных на экране или на печати.
- Число десятичных знаков – задается для числового и денежного типов данных число знаков после запятой.
- Подпись поля – задает текст, который выводится в заголовках таблиц, форм, отчетов.

- Условие на значение – задает ограничения на вводимые значения и позволяет осуществлять контроль ввода.
- Сообщение об ошибке – задает текст сообщения, который выводится на экран при нарушении условия на значение.

Определение ключа – для определения ключа выделяются поля, составляющие ключ и нажимается кнопка **Ключевое поле** (вкладка **Конструктор**). Ключевые поля всегда уникальны и не допускают пустых полей в записях. Если в таблице несколько ключевых полей, необходимо выделить эти поля с зажатой кнопкой **Ctrl**, а затем нажать кнопку **Ключевое поле**.

После сохранения таблицы с нужным именем делается доступным режим таблицы, позволяющий перейти к созданию записей.

Непосредственный ввод данных в таблицу осуществляется в режиме **Таблица**. Для перехода в этот режим в окне БД надо выбрать нужную таблицу и нажать правую кнопку мыши **Открыть** (кнопка) или двойной щелчок по ярлыку таблицы.



В открывшуюся таблицу можно вводить новые записи, заполняя значениями поля.

Замечание! Заполнять БД следует по строкам. Нельзя оставлять пустые поля (столбцы).

3.1.3.3. Проектирование схемы БД (разработка информационно-логической модели)

Согласно п. 3.1.3.1, в БД нужно хранить 15 атрибутов. Свести их в двумерную таблицу невозможно. Отберем атрибуты, которые можно поместить в одной таблице. Очевидно, что атрибуты “Номер зачетной книжки”, “Фамилия”, “Имя”, “Отчество”, “Дата рождения”, “Пол”, “Специальность”, “Шифр специальности”.

Если же ввести в эту таблицу атрибут “Курс”, получается, что все вышеперечисленные данные о студенте придется повторить столько раз, сколько курсов он отучился (т.е. наша таблица из двумерной превращается в трехмерную). На каждом курсе студент изучает много дисциплин и получает по ним оценки. Значит, введение сведений об экзаменах опять приведет к необходимости дублирования информации о каждом студенте (таблица становится четырехмерной). Чтобы избежать этого, проводим нормализацию отношений (см. п. 3.1.2), т.е. разбиваем атрибуты по таблицам так, чтобы избежать повторных записей.

Очевидно, что все атрибуты БД “Студент” можно записать в виде трех двумерных таблиц (рис. 8).

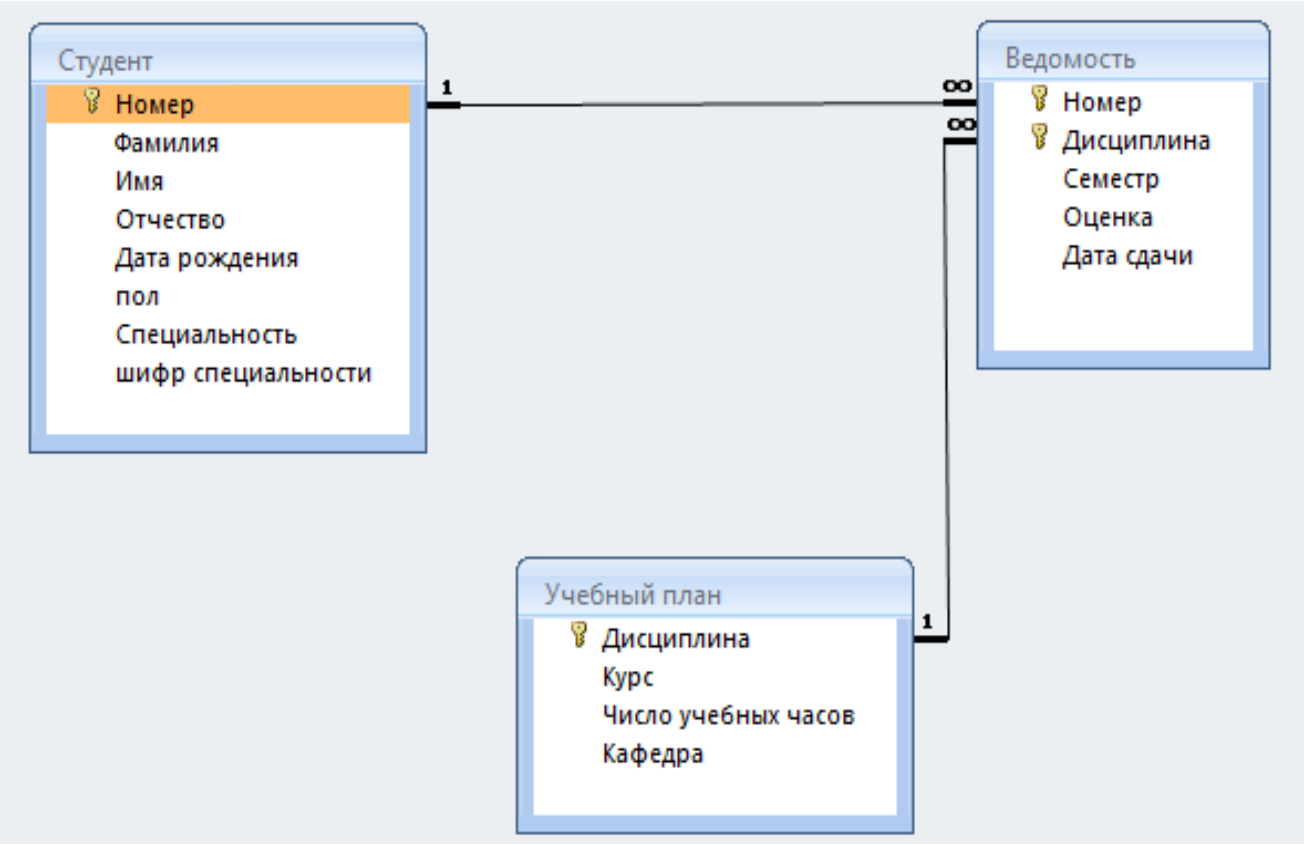


Рис. 8

Первая таблица БД **Студент** имеет ключ **Номер** и связывается со второй таблицей **Ведомость** по этому ключу. Тип связей этих таблиц 1:M или (1:∞), т.к. каждый студент сдает много экзаменов. Таблица **Ведомость** имеет два ключа (составной ключ). Ключ **Номер** используется для связи с таблицей **Студент**, а ключ **Дисциплина** – для связи с таблицей **Учебный план**. Таблица **Учебный**

план имеет ключ **Дисциплина**. Тип связей таблицы **Ведомость** и **Учебный план** 1:1.

Такое распределение всех атрибутов БД по трем таблицам обеспечивает отсутствие повторяющихся записей о каждом студенте, а наличие связей этих таблиц по ключам обеспечивает возможность обращения сразу ко всем таблицам.

3.1.3.4. Создание схем таблиц БД для машинных носителей (разработка логической структуры БД)

Как уже отмечалось, первый этап создания таблицы БД в Access заключается в описании структуры таблицы. Следует указать имена всех полей, тип хранящейся в них информации, размер поля. Для дробных чисел нужно также установить точность (количество знаков после запятой), для данных типа Дата/Время – формат (например, краткий – 20.10.14, полный – 20 октября 2014 г.)

Опишем схему (структуру) таблиц **СТУДЕНТ** (табл. 6), **ВЕДОМОСТЬ** (табл. 7) и **УЧЕБНЫЙ ПЛАН** (табл. 8).

Для создания структуры каждой таблицы следует: в окне БД Access щелкнуть по вкладке **Создать**. В группе **Таблицы** выбрать **Конструктор** и в окне Конструктора с клавиатуры ввести имена полей и их характеристики.

СТУДЕНТ *Таблица 6*

Ключ	Имя поля	Тип информации	Размер	Формат
🔑	Номер	Числовой	Длинное целое	-
	Фамилия	Текстовый	15	-
	Имя	Текстовый	15	-
	Отчество	Текстовый	15	-
	Дата рождения	Числовой	Целое	
	Пол	Текстовый	1	-
	Специальность	Текстовый	50	-
	Шифр специальности	Текстовый	6	-

ВЕДОМОСТЬ *Таблица 7*

Ключ	Имя поля	Тип информации	Размер	Формат
🔑	Номер	Числовой	Длинное целое	-
🔑	Дисциплина	Текстовый	20	-

	Семестр	Текстовый/	6	-
	Оценка	Числовой	Целое	-
	Дата сдачи	Дата/Время	-	Краткий

УЧЕБНЫЙ ПЛАН *Таблица 8*

Ключ	Имя поля	Тип информации	Размер	Формат
🔑	Дисциплина	Текстовый	20	-
	Курс	Числовой	Целое	-
	Число учебных часов	Числовой	Целое	-
	Кафедра	Текстовый	30	-

3.1.3.5. Заполнение таблиц БД данными

После того как создана и сохранена в СУБД структура каждой таблицы, можно заполнить эти таблицы данными. Произведем заполнение данными таблицы СТУДЕНТ (табл. 9), УЧЕБНЫЙ ПЛАН (табл. 10) и ВЕДОМОСТЬ (табл. 11).

Обратите внимание! Ключевые записи во всех таблицах должно быть заполнены одинаковыми данными! В противном случае при объединении этих таблиц в одну БД невозможно будет организовать к ней запросы.

СТУДЕНТ *Таблица 9*

Номер	Фамилия	Имя	Отчество	Дата рождения	Пол	Специальность	Шифр специальности
67136	Егорова	Анна	Ивановна	16.02.1990	ж	Системотехника	220100
96345	Смирнова	Ольга	Павловна	1.10.1991	ж	Менеджмент	061100
87988	Соболев	Алексей	Петрович	30.01.1992	м	Сварка	120500
86712	Иванов	Иван	Иванович	05.05.1991	м	Экономика	060800

УЧЕБНЫЙ ПЛАН *Таблица 10*

Дисциплина	Курс	Число учебных часов	Кафедра
Информатика	1	250	Информатики и математики
Высшая математика	2	160	Информатики и математики
Физика	2	200	Физики
Философия	1	170	Истории и философии

ВЕДОМОСТЬ *Таблица 11*

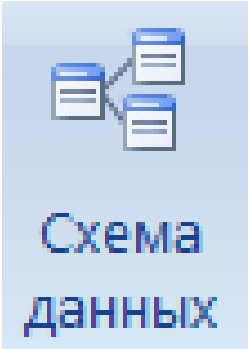
Номер	Дисциплина	Семестр	Оценка	Дата сдачи
67136	Информатика	Первый	Отлично	23.01.2014
96345	Высшая математика	Второй	Хорошо	08.06.2014
87988	Физика	Первый	Неудовл.	25.01.2014

86712	Философия	Второй	Удовл.	19.05.2014

Для заполнения своей БД в Access следует: выделить имя таблицы и дважды щелкнуть левой кнопкой мыши.

3.1.3.6. Создание схемы БД в Access

Выполнив вкладка Работа с базами данных – **Схема данных**, связать все таблицы по ключам (см. рис. 8).



Появится окно **Добавление таблицы** (рис. 9). Добавить все таблицы выделяя их и щелкая по кнопке добавить.

Чтобы соединить таблицы по ключевым полям необходимо:

- щелкнуть по ключевому поля одной таблицы и зажав левую клавишу мыши перетащить его в другую таблицу к соответствующему ключу появится окно рис. 10;
- в окне **Изменение связей** проставить (ОБЯЗАТЕЛЬНО) все флажки в полях как показано на рис. 10;
- щелкнуть по кнопке ОК

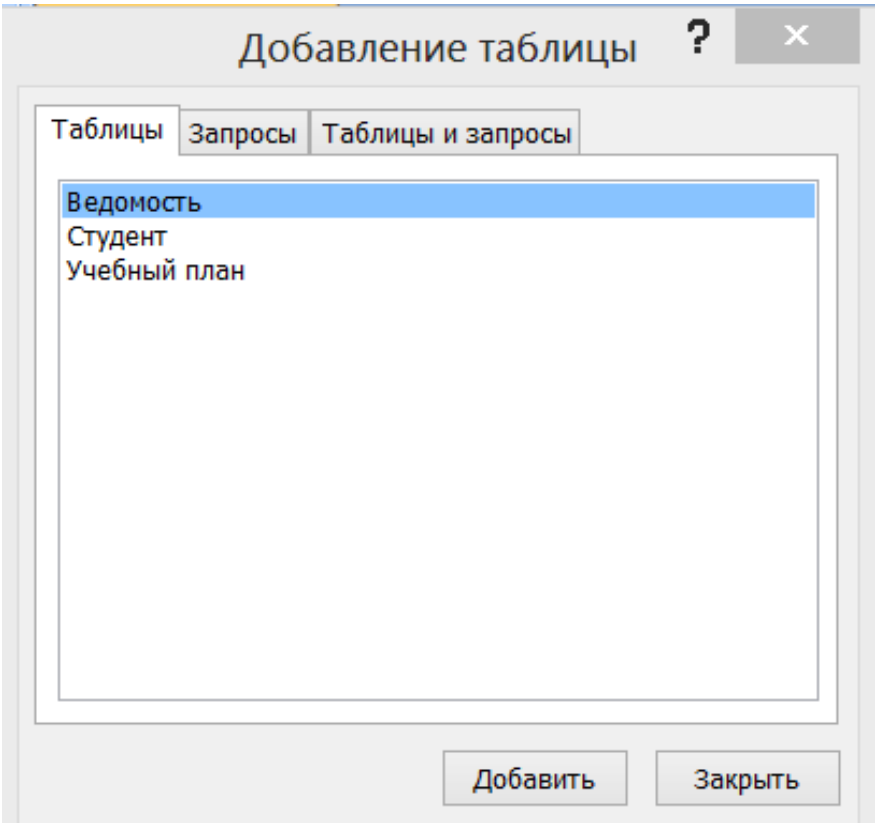

Добавление таблицы
Таблицы
Запросы
Таблицы и запросы
Ведомость
Студент
Учебный план
Добавить
Закрыть

Рис. 9

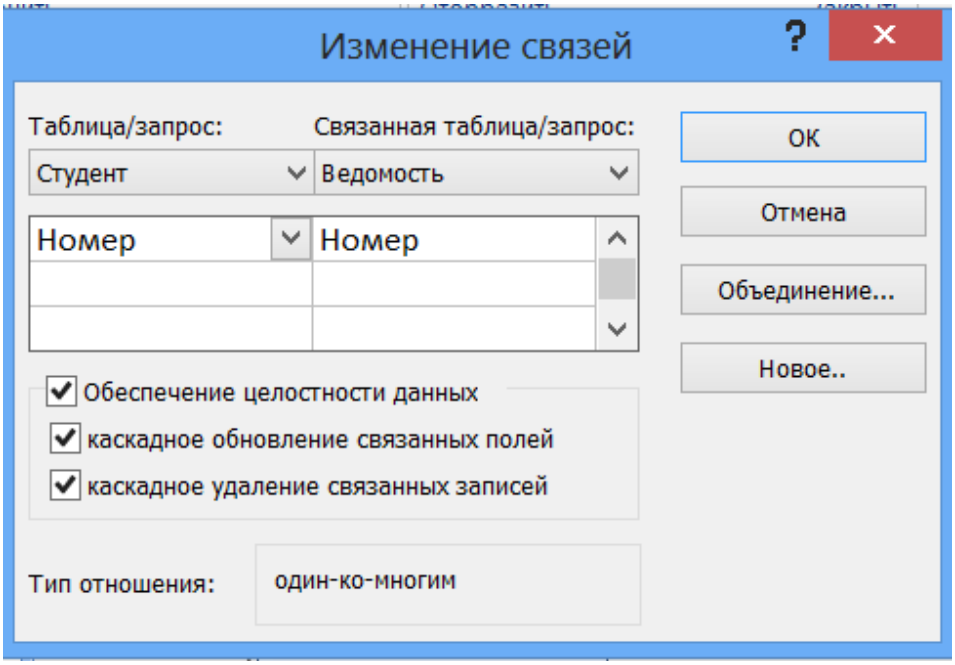

Изменение связей
Таблица/запрос:
Связанная таблица/запрос:
Студент
Ведомость
Номер
Номер
Обеспечение целостности данных
каскадное обновление связанных полей
каскадное удаление связанных записей
Тип отношения:
один-ко-многим
ОК
Отмена
Объединение...
Новое..

Рис.10

3.1.3.7. Формирование запросов

Для разработанной трехтабличной БД можно организовать следующие запросы:

- Выбрать студентов конкретной специальности (например, менеджмент);
- Выбрать студентов, имеющие оценки выше, ниже или равную указанной (например, создать список студентов, имеющих оценки выше “удовлетворительной” или список всех студентов, имеющих оценку “неуд.”);
- Создать список студентов-мужчин;
- Создать список студентов-женщин;
- Создать список студентов, родившихся до 1990 г. и т.д.

Для создания запросов в следует: в окне БД выбрать вкладку **Создание**, группа **Другие**.

1. Создание запроса данных из одной таблицы

1.1. Щелкнуть по кнопке **Конструктор запросов.**

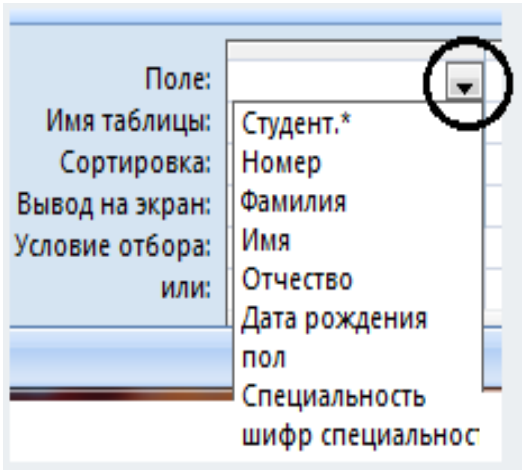
1.2. Появится окно **Добавления таблицы** (рис. 9)

1.3. Добавить таблицу Студент, закрыть окно **Добавление таблицы**. Щелкнуть по кнопке закрыть.

1.4. В нижнюю часть окна Запрос1 добавить поля: Фамилия, Имя, Пол, Специальность. Для этого:

1.4.1. **1 способ**. Щелкнуть по кнопке раскрытия списка в строке **Поле** (рис. 11 а), выбрать поле **Фамилия.**

1.4.2. **2 Способ.** Щелкнуть по полю Фамилия в таблице и зажав левую клавиши мыши перетащить его в строку **Поле** (рис. 11 б).



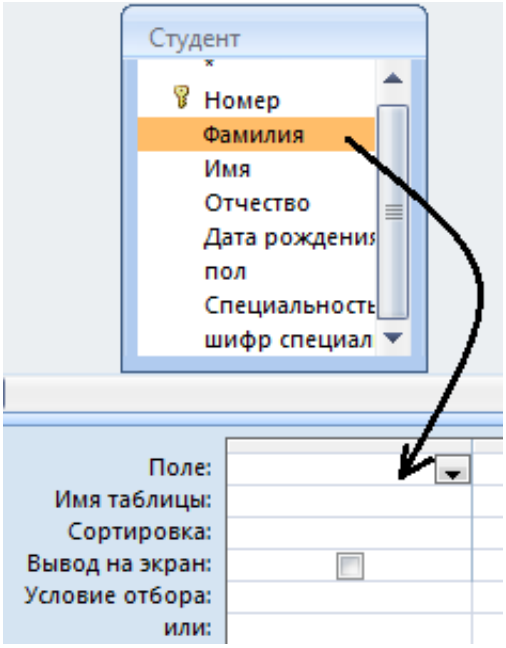


Рис. 11 а.

Рис. 11. б

1.5. В окне **Конструктор запросов** (рис. 12) в строке **Условие отбора** в нужном столбце ввести условие отбора (например, для формирования списка студентов – мужчин в столбце “Пол” ввести “М”) и щелкнуть по пиктограмме ! **Выполнить** на вкладке **Конструктор** (появится таблица, включающая только список студентов – мужчин рис. 13).

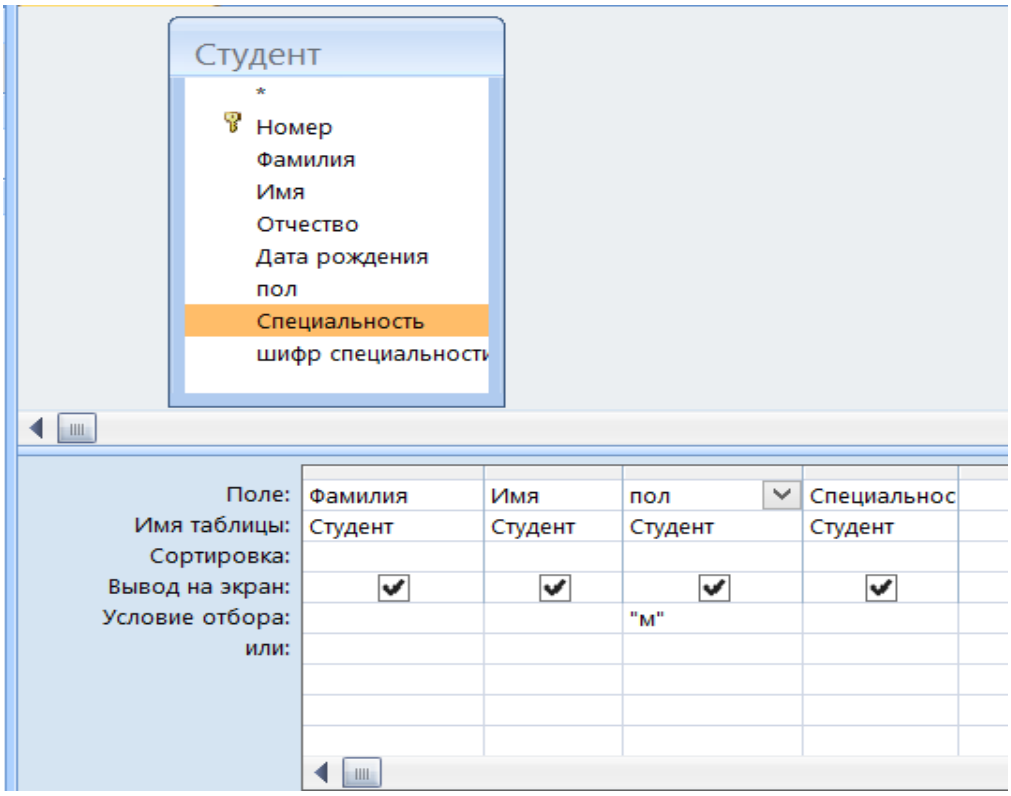


Рис. 12

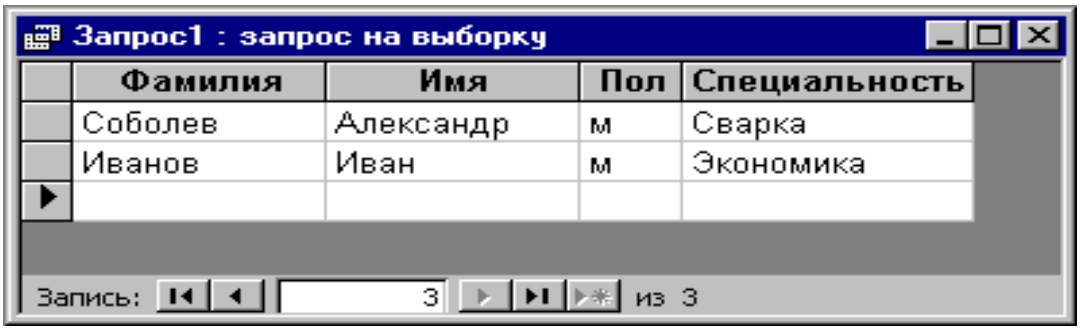


Рис. 13

2. Создания запроса данных, содержащихся в разных таблицах

2.1. Щелкнуть по кнопке **Конструктор запросов.**

2.2. Появится окно **Добавления таблицы** (рис. 9)

2.3. Добавить таблицы Студент, Ведомость и Учебный план, закрыть окно **Добавление таблицы**.

2.4. Добавить необходимые данные, например, Фамилия, Дисциплина и Число часов (рис. 14).

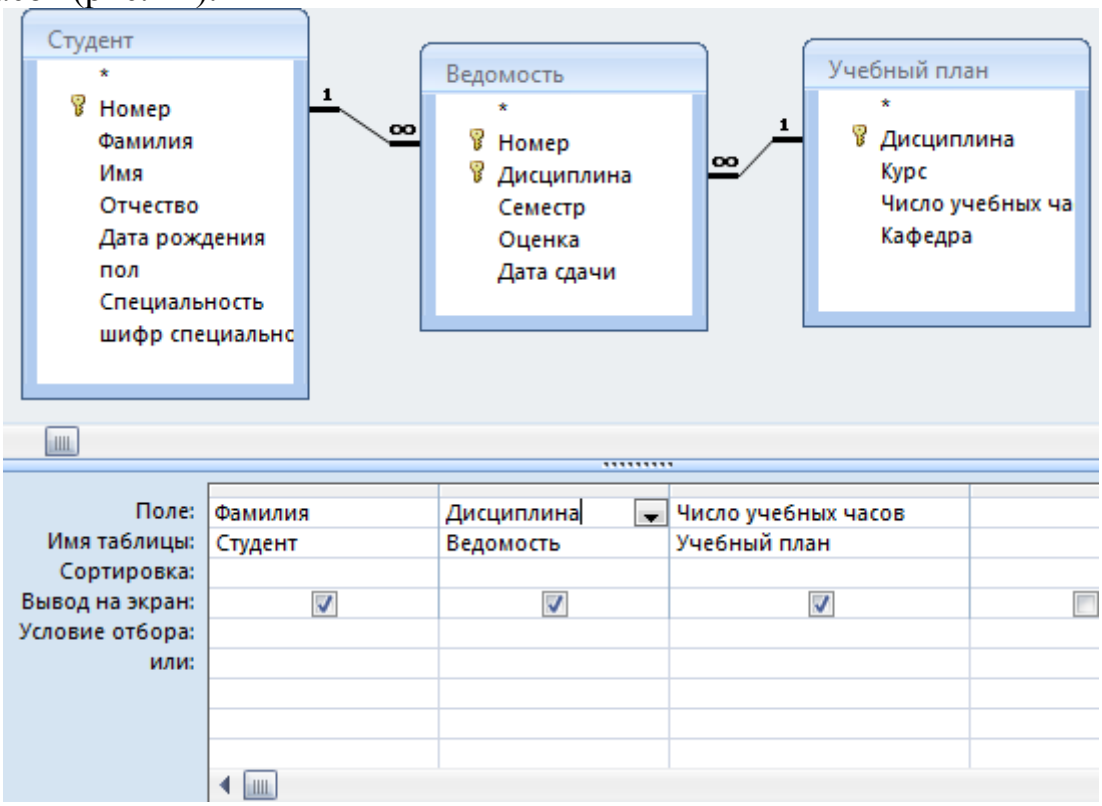


Рис. 14

2.5. щелкнуть по пиктограмме ! или вкладка **Конструктор** – **Выполнить**) (появится таблица, включающая только список студентов – мужчин рис. 15).

Фамилия	Дисциплина	Число учеб
Егорова	Информатика	250
Смирнова	Высшая математика	16
Соболев	Физика	200
Ивнов	Философия	170
*		

Рис. 15

3.1.3.8. Формирование отчетов

Для формирования отчетов используются данные из одной или нескольких связанных таблиц или на основе сформированного ранее запроса.

Например, для формирования отчета созданного на основе запроса в пп. 3.1.3.7 необходимо:

1. Перейти на вкладку **Создание** – группа **Отчеты** выбрать команду **Мастер отчетов** появится окно Создание отчетов рис. 16.
2. На первом шаге мастера в окне **Таблицы и запросы** выбрать Запрос2.

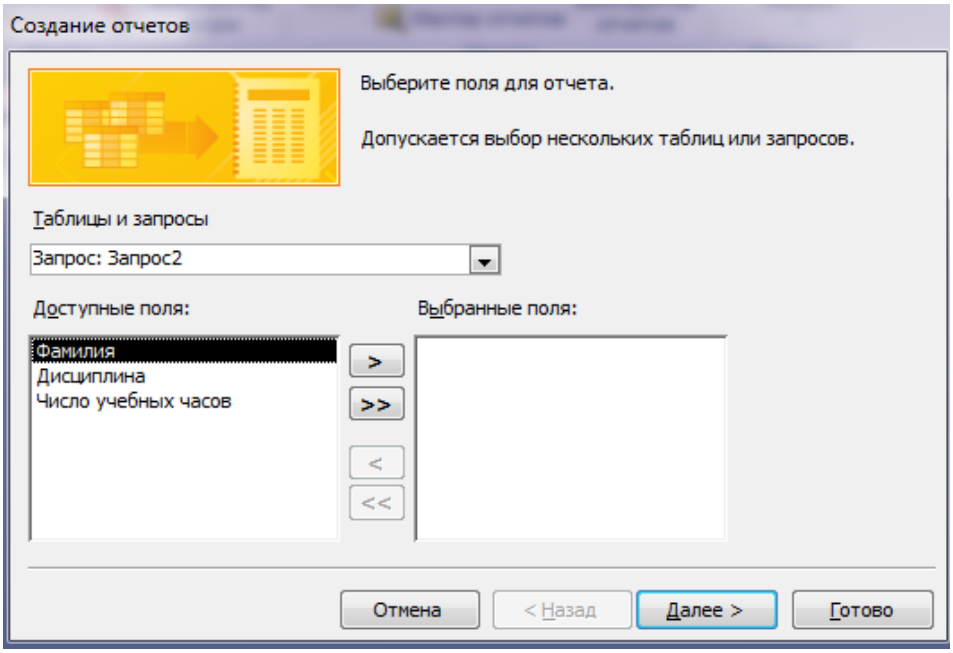


Рис. 16

3. Из окна **Доступные поля:** перенести все поля в окно **Выбранные поля:** нажав кнопке >>. Нажать кнопку **Далее**.
4. Второй шаг мастера Представление данных (рис. 18) в окне **Выберите вид представления данных:** щелкните по таблице Ведомость, нажмите кнопку **Далее** (или этот шаг можно пропустить нажав Далее);

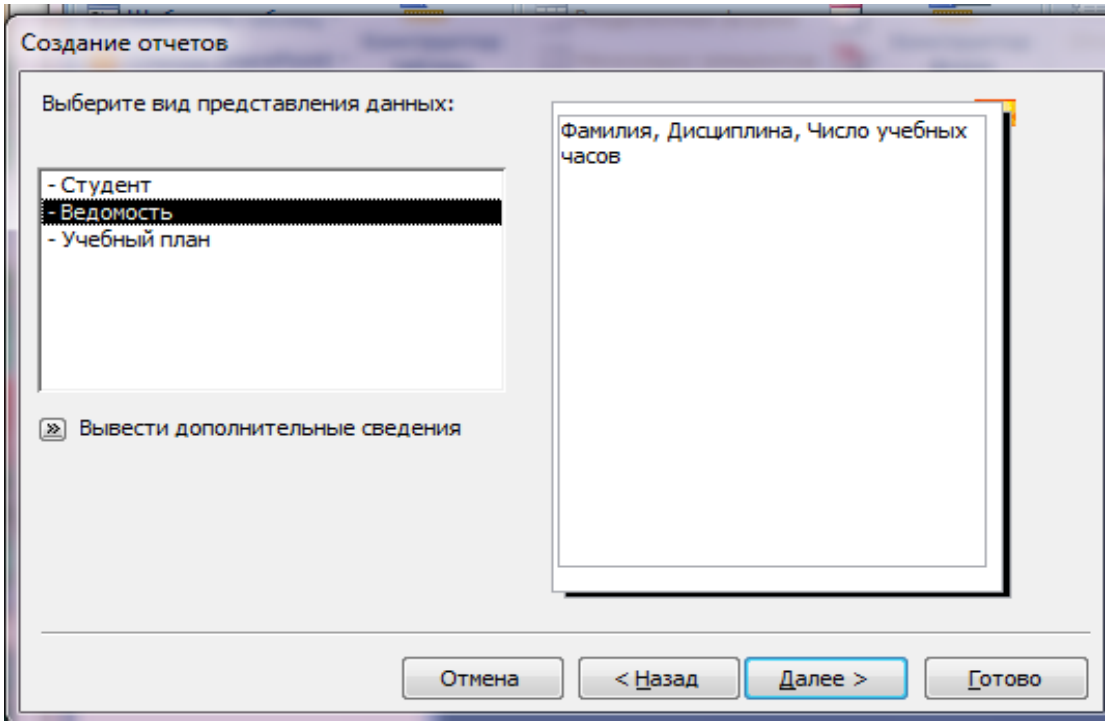


Рис. 18

5. Третий шаг **Добавление уровней группировки** (рис. 19). Это шаг необходим в том случае, если вы подводите итоги в отчете. В нашем случае этот шаг пропускаем нажав кнопке **Далее**.

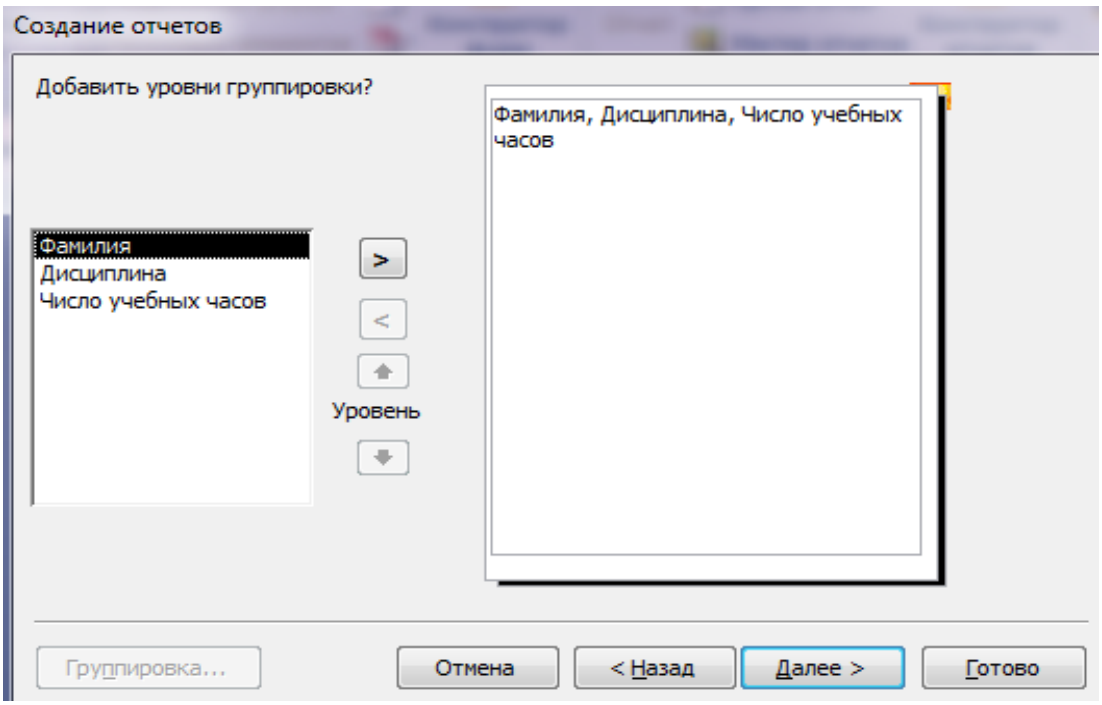


Рис. 19

6. Четвертый шаг. **Задайте требуемый порядок сортировки** (рис. 20). Если вы хотите произвести одноуровневую или многоуровневую сортировки данных в отчете задайте их на этом шаге. Например, отсортируйте данные по полю **Фамилия** в порядке возрастания (рис. 21). Нажмите кнопку **Далее**.

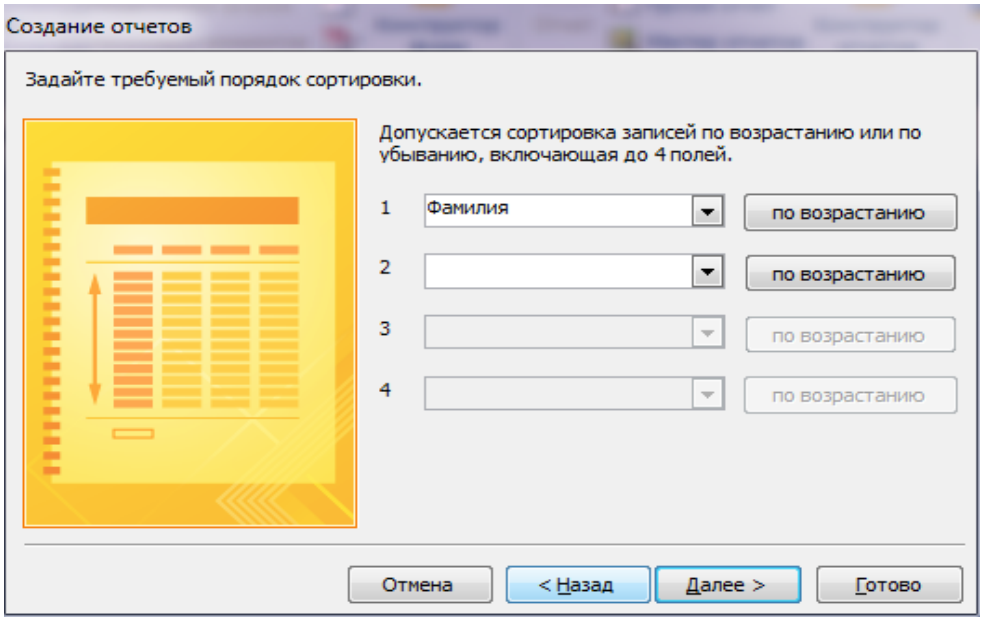


Рис. 21

7. Пятый шаг. **Выберите вид макеты для отчета** (рис. 22). Выбирается макет отчета в нашем случае табличный, вид макета представлен на рисунке в окне слева, также выбирается ориентация страницы, если

данных много или они объемные выберите альбомную ориентацию страницы, в нашем случае Ориентация книжная. Нажмите кнопку **Далее**;

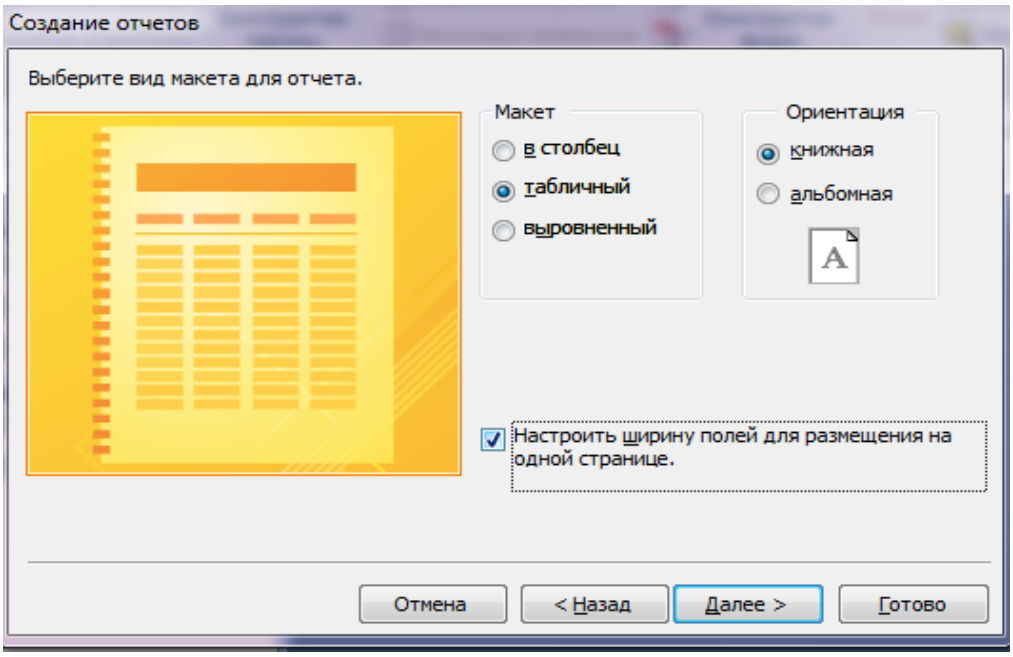


Рис. 22

8. Шестой шаг. **Выберите требуемый стиль** (рис. 23). На этом шаге мастера выберите любой стиль оформления отчета по вашему усмотрению. Вид стиля оформления страницы показан в окне слева, нажмите кнопке **Далее**;
9. Седьмой шаг. На этом последнем этапе задается имя отчета (рис.24) Щелкнуть по кнопке **Готово**.

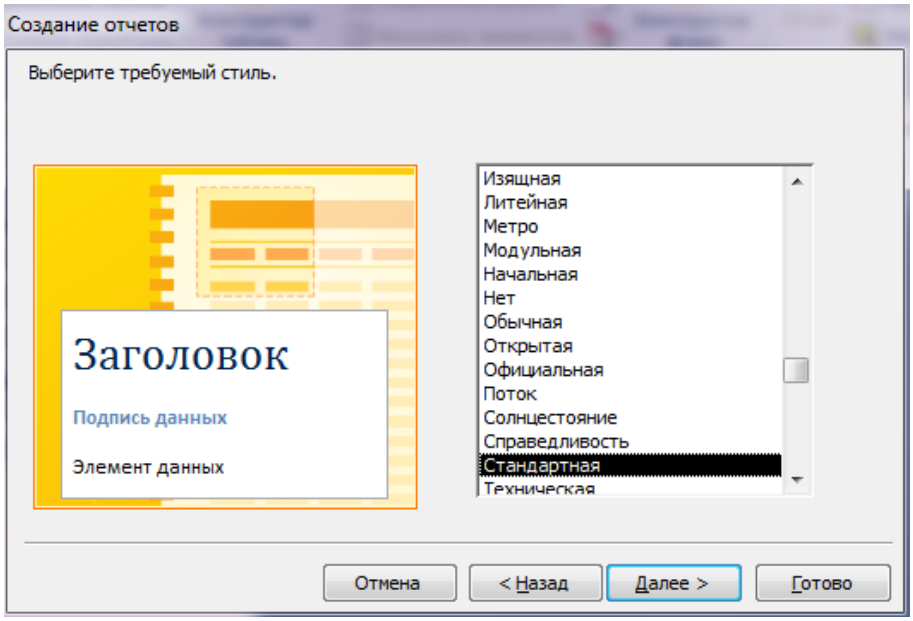


Рис. 23

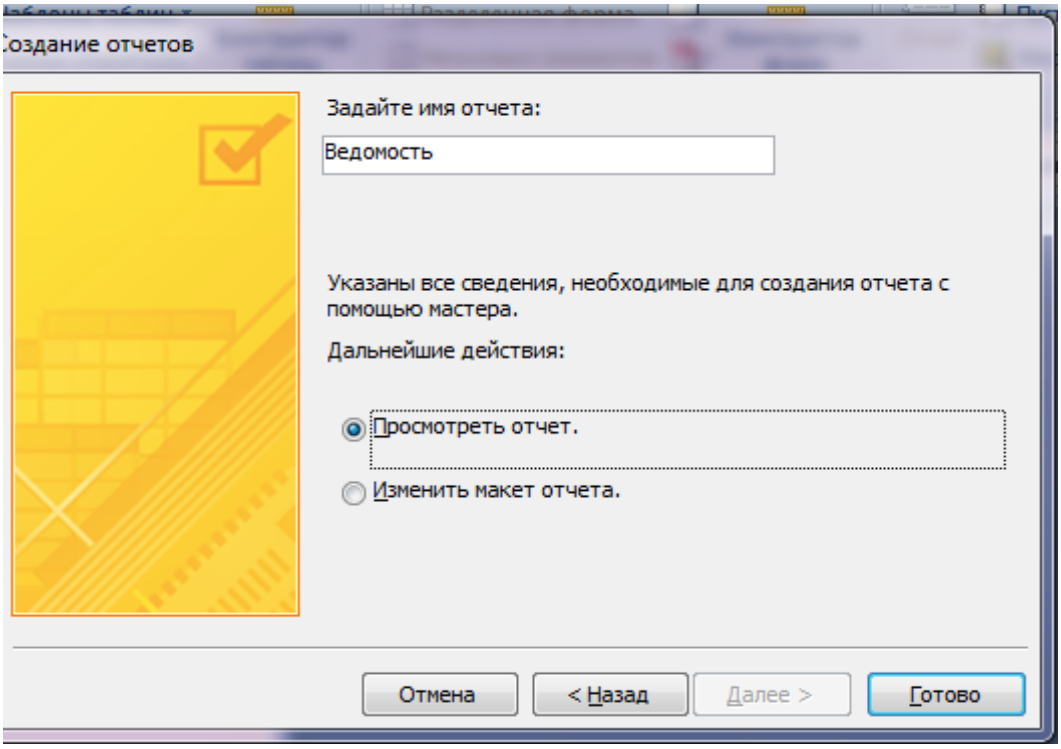

Создание отчетов
Задайте имя отчета:
Ведомость
Указаны все сведения, необходимые для создания отчета с помощью мастера.
Дальнейшие действия:
Просмотреть отчет.
Изменить макет отчета.
Отмена
< Назад
Далее >
Готово

Рис. 24

Приложение

Национальный открытый институт г. Санкт-Петербург

Кафедра Математических и естественно-научных дисциплин

КУРСОВАЯ РАБОТА
по
БАЗАМ ДАННЫХ

Тема: *Конструирование баз данных*

Выполнил студент:
Курс:

Направление подготовки:
Профиль подготовки:

Проверил преподаватель:

Оценка:

Подпись преподавателя:

Дата:

Санкт-Петербург
20__

Оглавление

1. ОБЩИЕ УКАЗАНИЯ 2
Библиографический список 3
2. ЗАДАНИЯ НА КУРСОВУЮ 3
3. МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ВЫПОЛНЕНИЮ КУРСОВОЙ РАБОТЫ 19
3.1. Проектирование баз данных (БД) 19
3.1.1. Основные понятия 19
3.1.1.1. Структурные элементы базы данных 20
3.1.1.2. Реляционная модель данных 21
СТУДЕНТ 22
(Номер) 22
(Результат) 22
(Номер) 22
(Результат) 22
3.1.1.3. Нормализация отношений 23
3.1.1.4. Типы связей 26
3.1.2. Основные этапы проектирования, создания и ведения баз данных 27
3.1.3. Пример выполнения курсовой работы по проектированию баз данных 28
3.1.3.1. Отбор атрибутов, сведения о которых должны храниться в БД 28
3.1.3.2. Выбор системы управления базами данных (СУБД). 28
3.1.3.3. Проектирование схемы БД (разработка информационно-логической модели) 30
3.1.3.4. Создание схем таблиц БД для машинных носителей (разработка логической структуры БД) 32
3.1.3.5. Заполнение таблиц БД данными 33
3.1.3.6. Создание схемы БД в Access 34
3.1.3.7. Формирование запросов 36
3.1.3.8. Формирование отчетов 39
Приложение 44