

МИНОБРНАУКИ РОССИИ

**Орский гуманитарно-технологический институт (филиал)
федерального государственного бюджетного образовательного учреждения
высшего образования «Оренбургский государственный университет»
(Орский гуманитарно-технологический институт (филиал) ОГУ)**

Кафедра программного обеспечения

**Методические указания по выполнению и защите лабораторных и практических работ
по дисциплине «Б1.Д.Б.20 Базы данных»**

Уровень высшего образования

БАКАЛАВРИАТ

Направление подготовки

09.03.01 Информатика и вычислительная техника
(код и наименование направления подготовки)

Программное обеспечение средств вычислительной техники и автоматизированных систем
(наименование направленности (профиля) образовательной программы)

Тип образовательной программы

Программа бакалавриата

Квалификация

Бакалавр

Форма обучения

Очная

Год начала реализации программы (набора)

2019

г. Орск 2018

Методические указания предназначены для обучающихся очной формы обучения направления подготовки 09.03.01 Информатика и вычислительная техника профилю Программное обеспечение средств вычислительной техники и автоматизированных систем по дисциплине «Б1.Д.Б.20 Базы данных»

Составитель _____  О.В. Подсобляева

Методические указания рассмотрены и одобрены на заседании кафедры программного обеспечения, протокол № 1 от «01» сентября 2018 г.

Заведующий кафедрой _____  Е.Е. Сурина

Согласовано:

Председатель методической комиссии по направлению подготовки 09.03.01 Информатика и вычислительная техника

_____  Е.Е.Сурина
«12» сентября 2018 г.

© Подсобляева О.В., 2018
© Орский гуманитарно-технологический институт (филиал) ОГУ, 2018

Пояснительная записка

В результате изучения дисциплины «Б1.Д.Б.20 Базы данных» у обучающихся должны быть сформированы знания, умения и навыки:

- сформировать теоретические основы построения и приобретения начальных практических навыков проектирования и эксплуатации баз данных в автоматизированных информационных системах.

- усвоить порядок создания и сопровождения баз данных, настройки и сопровождения СУБД;

- изучить основы языка запросов SQL, развития архитектур баз данных, развития отечественных и зарубежных СУБД.

Одной из наиболее эффективных форм закрепления теоретических знаний и выработки навыков самостоятельной работы являются лабораторные и практические занятия.

Целью проведения лабораторных и практических занятий является:

- закрепление знаний студентов по основам проектной деятельности,

- формирование у студентов навыков использования современных технических средств и технологий для решения проектных и исследовательских задач.

Тематический план

Таблица 1 – Тематический план выполнения лабораторных и практических работ по дисциплине «Б1.Д.Б.20 Базы данных» для обучающихся направления подготовки 09.03.01 Информатика и вычислительная техника профиль подготовки Программное обеспечение средств вычислительной техники и автоматизированных систем

Практические работы

№ раздела	Наименование практических работ	Кол-во часов
3	Проектирование баз данных. Создание проектов Access – клиентских приложений сервера баз данных.	2
3	Разработка форм – интерфейса для ввода, просмотра и корректировки документов. Подготовка новых документов	2
4	Программирование на VBA в среде Access. Программирование обработки событий. Объектная модель Access. Объекты и модели доступ к данным. Свойства, методы и события объектов.	4
4	Создание проектов Access – клиентских приложений сервера баз данных.	2
5	Инструкции языка SQL: создание таблиц и схем базы данных	2
5	Создание базы данных на сервере. Инструментальные средства многопользовательских СУБД. Создание таблиц и схем данных. Типы данных. Определение параметров целостности: ключи, индексы, ограничения на значения, связанная целостность.	2
5	Система безопасности серверных баз данных. Разграничение доступа к данным. Управление учетными записями. Роли сервера. Назначение прав доступа к базам данных и их объектам. Роли баз данных.	2
	Итого:	16

Лабораторные работы

№ раздела	Наименование лабораторных работ	Кол-во часов
5	Работа в MS Access 2010	16
7	Технология доступа к базе данных ADO в среде программирования Delphi	2
7	Навигационная обработка данных в таблицах: просмотр и модификация.	2
7	Фильтрация записей.	2
7	Сортировка и поиск записей по разным критериям.	2
7	Реализация ссылочной целостности БД.	2
7	Динамические запросы с параметрами.	2
7	Организация меню в приложении: главное и контекстное.	2
7	Динамические запросы на модификацию.	2
	Итого:	32

Методические указания по выполнению и оформлению лабораторных и практических работ

Лабораторные и практические работы по дисциплине «Базы данных» предполагают решение задач по темам, представленным в тематическом плане.

В лабораторной и практической работе должны быть выполнены все предусмотренные задания. В работе должна просматриваться логическая последовательность и взаимная увязка основных частей работы.

Рекомендуемая структура лабораторных и практических работ:

- 1) цель работы;
 - 2) задание в соответствии с выбранным вариантом;
 - 3) теоретическая часть, включающая краткое изложение теоретических положений по теме лабораторной работы, формулы для решения задания;
 - 4) практическая часть, включающая решение задания по теме лабораторной работы.
- Дополнительно для наглядности расчетный материал может быть представлен в виде таблиц, графиков;
- 5) выводы по практической работе;
 - 6) список использованной литературы.

Работы могут быть оформлены:

- машинописным текстом на листах формата А4.

Титульный лист оформляется на основе СТО 02069024. 101 – 2014 «РАБОТЫ СТУДЕНЧЕСКИЕ. Общие требования и правила оформления».

Работа защищается устно и принимается к зачету, если нет замечаний по ее выполнению и оформлению. При отсутствии зачетных практических работ студент не допускается к зачету по дисциплине «Б1.Д.Б.14 Базы данных».

Практическая работа № 1, 2, 3, 4 (по вариантам)

Тема: Проектирование баз данных. Создание проектов Access – клиентских приложений сервера баз данных. Разработка форм – интерфейса для ввода, просмотра и корректировки документов. Подготовка новых документов. Программирование на VBA в среде Access. Программирование обработки событий. Объектная модель Access. Объекты и модели доступ к данным. Свойства, методы и события объектов. Создание проектов Access – клиентских приложений сервера баз данных.

Общая часть задания:

ФИО		Годовая зарплата	
Стаж		Отдел	

Отчет

№ отдела	Штат по плану	ФИО	Специальность	Стаж	Годовая зарплата
.....
Итого по отделу средняя зарплата				

Вариант №3

Структура записей таблиц

- 1) Предприятие Продукция Выпуск 1 кв. Выпуск 2 кв. Выпуск 3 кв. Выпуск 4 кв. Средняя цена
- 2) Предприятие Число работников Адрес

Контроль исходных данных: Выпуски и средняя цена – натуральные числа. Число работников – целое от 10 до 100.

Запрос на удаление: удалить данные о произвольно заданной продукции

Запросы на выборку: а) Вывод справки по видам продукции

Продукция Стоимость годового выпуска Количество предприятий

б) Вывод сведений о предприятиях, у которых суммарный квартальный выпуск всей продукции рос от квартала к кварталу

Предприятие % роста Годовой выпуск на 1-ого человека (руб)

Форма

Предприятие		Стоимость годового выпуска	
Штат		Адрес	

Отчет

Предприятие	Адрес	Продукция	Годовой выпуск	Средняя цена
.....
Итого по предприятию выпущено на сумму			

Вариант №4

Структура записей таблиц:

- 1) Предприятие Изделие Цена 1 кв. Цена 2 кв. Цена 3 кв. Цена 4 кв. Годовой выпуск
- 2) Предприятие Адрес Размер основных фондов (руб)

Контроль исходных данных: числовые величины – натуральные числа

Запрос на удаление: удалить данные о произвольно заданном изделии

Запросы на выборку: а) Вывести справку по изделиям вида:

Изделия Стоимость годового выпуска Количество предприятий

б) сведения о предприятиях с минимальным размером основных фондов

Предприятие РазмерОФ Кол-во видов изделий Средняя ст-сть год-го выпуска

Форма

Предприятие		Годовой выпуск	
Средняя цена		Изделие	

Отчет

Предприятие Адрес Изделие Стоимость годового выпуска

 Итого по предприятию

Вариант №5

Структура записей таблиц:

- 1) Предприятие Шифр продукции Цена Выпуск 1 кв. Выпуск 2 кв Выпуск 3 кв Выпуск 4 кв
- 2) Шифр продукции Название продукции

Контроль исходных данных: цена и выпуски – натуральные числа

Запрос на удаление: удалить записи по произвольно заданному предприятию

Запросы на выборку: а) Вывести справку вида:

Предприятие Название продукции Стоимость годового выпуска

б) Для произвольно заданной продукции вывести список выпускающих ее предприятий и их цены на эту продукцию

Форма

Предприятие		
Название продукции		Выпуск по кварталам

Отчет

Предприятие Продукция Цена Годовой выпуск

 Итого по предприятию выпущено на сумму

Вариант №6

Структура записей таблиц:

- 1) Предприятие Изделие Выпуск Цена
 1 кв. 2 кв 3 кв к 4 кв 1 кв. 2 кв 3 кв к 4 кв
- 2) Предприятие Адрес Число работающих

Контроль исходных данных: Выпуск и цены – натуральные числа. Число работающих – целое в диапазоне 30-300

Запрос на удаление: удалить записи по произвольно заданному изделию

Запросы на выборку: а) Вывести справку вида:

Предприятие Адрес Количество видов изделий Стоимость годового выпуска

б) *Изделие Среднегодовой выпуск Средняя цена Максимальная цена за 1 кв.*

Форма

Предприятие	<input type="text"/>
Изделие	<input type="text"/>
Стоимость по кварталам	
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>

Отчет

Предприятие Адрес Изделие Стоимость годового выпуска

 Итого по предприятию выпущено на сумму

Вариант №7

Структура записей таблиц:

1) № отдела ФИО Специальность Стаж Год рождения Зарплата по месяцам
 1 2 3 4 5 6

2) № отдела Название Плановый месячный фонд зарплаты

Контроль исходных данных: № отдела – целое число от 1 до 5, стаж – целое от 1 до 40, зарплата – натуральные числа, большие 600, год рождения – целое от 1954 до 1984

Запрос на изменение: заменить номер отдела с заданным названием на новый, введенный с клавиатуры

Запросы на выборку: а) Вывести справку вида:

Название отдела Число сотрудников Плановый фонд з/п Фактическая з/п за февраль

б) Вывод справки о специальностях на букву «К» вида:

Специальность Минимальный стаж Максимальный стаж Средняя зарплата за полгода

Форма

№ отдела	<input type="text"/>	Название
Средний возраст	<input type="text"/>	<input type="text"/>

Отчет

Название ФИО Специальность Возраст Стаж Зарплата

 Зарплата по отделам.....

Вариант №8

Структура записей таблиц:

1) Группа ФИО Предмет Пропуски по месяцам Оценка
 февраль, март, апрель, май

2) Предмет Количество часов за семестр

Контроль исходных данных: Пропуски – целые числа от 0 до 30 оценка – целое число от 2 до 5, количество часов за семестр – целое от 30 до 80

Запросы на выборку: а) Вывести сведения о студентах, пропустивших более 80% часов по какому – либо предмету:

Группа ФИО Предмет % пропусков за семестр

б) Предмет Количество часов Средний % пропусков за семестр:

в) для произвольно заданной группы вывести сведения о студентах с минимальным общим числом пропусков:

Группа ФИО Общее число пропусков Средний балл

Форма

Группа		Студент	
Общее число пропусков		Средний балл	

Отчет

Группа ФИО Предмет Число пропусков Оценка

.....

По группе общее число пропусков.....

Вариант №9

Структура записей таблиц:

1) Магазин товар Цена Количество продаж по дням недели
1 2 3 4 5 6 7

2) Магазин Адрес Арендная плата Число сотрудников

Контроль исходных данных: величины – натуральные числа

Запрос на изменение: увеличить цену товаров в произвольно заданном магазине на 10%

Запросы на выборку: а) Для произвольно заданного товара вывести сведения о его максимальной и минимальной ценах и о магазинах, где они имеют место

б) Вывести справку вида:

Магазин Арендная плата Число сотр. Число видов товаров Общая стоимость продаж за неделю

Форма

Магазин		Число сотрудников	
Общая стоимость товаров за неделю			

Отчет

Товар Минимальная цена Максимальная цена Общее количество продаж во всех магазинах

.....

Общая сумма.....

Вариант №10

Структура записей таблиц:

1) Предприятие Шифр продукции Выпуск 1 кв. выпуск 2 кв. Выпуск 3 кв. Выпуск 4 кв.

2) Название продукции Шифр продукции Цена за единицу

Контроль исходных данных: числовые величины – натуральные числа

Запрос на удаление: удалить записи, содержащие произвольно заданное предприятие

Запросы на выборку: а) Вывести справку вида

Предприятие Количество видов продукции Средняя стоимость годового выпуска

б) вывод сведений о произвольно заданной своим названием продукции:

Продукция Шифр Максимальный Годовой выпуск Минимальный годовой выпуск

Форма

Продукция		Среднегодово й выпуск
Число выпускающих предприятий		

Отчет

Предприятие Продукция Годовой выпуск Стоимость годового выпуска
.....
Итого по предприятиям.....

Вариант №11

Структура записей таблиц:

1) № автоколонны № автомобиля Шифр груза Объем перевозок по дням недели
B1 B2 B3 B4 B5 B6 B7

2) Шифр груза Название груза Цена за единицу

Контроль исходных данных: Шифр – натуральное 5ти разрядное число. Объемы перевозок – натуральные числа от 1 до 100, цена – натуральное число.

Запрос на изменение: заменить произвольно заданный номер автомобиля на другой произвольно заданный.

Запросы на выборку: а) Вывод сведений о перевозках произвольно заданной автоколонны:
№ автомобиля Название груза Перевезено за неделю

б) для грузов, название которых начинается на букву «Т», а цена больше произвольно заданной величины, вывести справку

Название груза Объем перевозок за неделю стоимость перевезенного груза

Форма

№ автоколонны		№ автомобиля	
Шифр груза		Перевезенное за неделю кол-во	
Название груза			

Отчет

№ автоколонны № автомобиля Название груза Перевезено за неделю
.....
Итого по автоколонне.....

Вариант №12

Структура записей таблиц:

1) Шифр института Курс Группа ФИО Экзаменационные оценки
B1 B2 B3 B4 B5 B6 B7

2) Шифр института Название % бюджетников Число специальностей

Контроль данных: курс – целое от 1 до 5, группа – буквенно – числовая, оценки – целые от 2 до 5, %бюджетников – вещественное от 20 до 90, число специальностей – целое от 1 до 4

Запрос на удаление: удалить записи о студентах, имеющих больше 3 двоек

Запросы на выборку: а) вывести итоги по институтам:

Название института Число специальностей Количество отличников

б) вывести сведения о студентах, средний балл которых равен максимальному среднему баллу в их группе

Шифр института курс Группа ФИО Средний балл

Форма

Институт		Число специальностей	
Курс		Группа	
Средний балл			
Фамилия ИО			

Отчет

Институт Курс Группа Фамилия ИО Средний балл

.....

Средний балл по группе

Вариант №13

Структура записей таблиц:

1) Предприятие Шифр продукции Выпуск 1 кв. Выпуск 2 кв. Выпуск 3 кв. Выпуск 4 кв. цена

2) Шифр продукции Название продукции Название стандарта

Контроль данных: выпуски – целые числа в диапазоне от 1 до 200, цена – натуральное число

Запрос на удаление: удалить данные по произвольно заданному предприятию

Запросы на выборку: а) вывести справку вида:

Название продукции Стандарт Стоимость годового выпуска Число предприятий

б) вывести сведения о предприятиях, непрерывно увеличивавших в течение года стоимость выпуска продукции (суммарную по всем видам на этом предприятии)

Предприятие % роста за год

Форма

Предприятие		Стоимость годового выпуска
Шифр продукции		

Отчет

Предприятие Продукция Годовой выпуск Цена Стоимость

.....

Стоимость выпуска по предприятию.....

Вариант №14

Структура записей таблиц:

1) Институт Курс Группа ФИО Экзаменационные оценки

V1 V2 V3 V4 V5 V6

2) Институт Общее количество студентов Число групп на курсе

Контроль данных: курс – целое от 1 до 5, оценки – целые от 2 до 5. число групп на курсе – целое от 1 до 7.

Запрос на удаление: удалить сведения о студентах. У которых больше 3 двоек

Запросы на выборку: а) вывести справку вида:

Форма

№ отдела		Зарплата за
Фамилия ИО		

Отчет

Название ФИО Специальность Возраст Стаж Средняя зарплата

.....

Средняя зарплата по отделу

Вариант №17

Структура записей таблиц:

1) Предприятие Продукция Прибыль по месяцам
B1 B2 B3 B4 B5 B6

2) Предприятие Размер персонала Адрес

Контроль данных: Прибыль – натуральные числа, размер персонала – натуральное число от 10 до 300

Запрос на удаление: удалить данные о произвольно заданной продукции

Запросы на выборку: а) вывести справку о предприятиях города Орел вида:

Предприятие Число сотрудников Количество видов продукции

б) вывести справку вида:

Продукция Количество выпускающих предприятий среднемесячная прибыль

Форма

Предприятие		Прибыль
Продукция		

Отчет

Предприятие Продукция Прибыль за полугодие

.....

Прибыль по предприятию.....

Вариант №18

Структура записей таблиц:

1) Предприятие Продукция Выпуск за полгода Цена единицы
1 2 3 4 5 6

2) Предприятие Количество сотрудников Адрес

Контроль данных: выпуски, цена – натуральные числа, количество сотрудников – целое число от 30 до 300

Запросы на выборку: а) вывести количество видов продукции, выпускаемых произвольно заданным предприятием.

б) вывести название и адреса предприятий с самым большим количеством сотрудников:

в) вывести справку вида:

Продукция Количество предприятий Средняя цена Максимальная цена

Форма

Фамилия ИО		Специальность	
Курс		Группа	
		Средний балл	

Отчет

Название ФИО Специальность Возраст Стаж Средняя зарплата

 Средняя зарплата по отделу

Вариант №21

Структура записей таблиц:

1) Шифр специальности Курс Группа ФИО Оценки за сессию
 1 2 3 4

2) Шифр специальности Название специальности

Контроль данных: курс – целое от 1 до 5, оценки – целые от 2 до 5.

Запрос на удаление: удалить записи с произвольно заданной группой

Запросы на выборку: а) вывести сведения о группах с максимальным средним баллом в виде:

Специальность Курс Группа Средний балл

б) вывести сведения о неуспевающих произвольно заданного курса:

Специальность Курс Группа ФИО Средний балл

Форма

Фамилия ИО		Специальность	
Курс		Группа	
		Средний балл	

Отчет

Название ФИО Специальность Возраст Стаж Средняя зарплата

 Средняя зарплата по отделу

Вариант №22

Структура записей таблиц:

1) Предприятие Продукция Выпуск по месяцам Цена единицы
 1 2 3 4 5

2) Предприятие Адрес

Контроль данных: числовые величины – натуральные числа

Запрос на удаление: удалить сведения о произвольно заданной продукции

Запросы на выборку: а) вывод справки о предприятиях г. Тулы в виде:

Предприятие Адрес Количество видов продукции Стоимость выпуска за полгода :

б) вывести справку вида:

Предприятие Стоимость выпуска по месяцам Стоимость выпуска за полгода

Форма

Предприятие		Выпуск за полгода
Продукция		

Отчет

Предприятие Продукция Выпуск за полгода

Итого по предприятию

Вариант №23

Структура записей таблиц:

- 1) Предприятие Продукция План выпуска на год Выполнение плана по месяцам Цена
- 2) Предприятие Размер персонала Адрес

Контроль данных: план, цена, размер персонала – натуральные числа.

Запрос на изменение: уменьшить цену произвольно заданной продукции на 10%

Запросы на выборку: а) вывод сведений о предприятиях г. Москвы в виде:

Предприятие Адрес Размер персонала Количество видов продукции:

б) вывод сведений о предприятиях, выпускающих произвольно заданную продукцию по минимальной цене:

Продукция Предприятие Цена

Форма

Предприятие		Стоимость годового выпуска
Продукция		
Адрес		

Отчет

Предприятие Продукция Размер персонала Стоимость фактического выпуска за год

Итого по предприятию

Вариант №24

Структура записей таблиц:

- 1) Предприятие Шифр продукции Выпуск 1кв. Выпуск 2кв. Выпуск 3кв. Выпуск 4кв. Цена
- 2) Шифр продукции Название продукции

Контроль данных: выпуски и цена – натуральные числа.

Запрос на удаление: удалить данные по произвольно заданному предприятию

Запросы на выборку: а) вывести справку вида:

Предприятие Стоимость годового выпуска Количество видов продукции:

б) вывести сведения о произвольно заданной своим названием продукции:

Продукция Шифр Максимальный годовой выпуск Минимальный годовой выпуск

Форма

Предприятие	<input type="text"/>
Продукция	<input type="text"/>
Выпуск за год	<input type="text"/>

Отчет

Предприятие	Продукция	Цена	Стоимость годового выпуска
.....
Итого по предприятию			

Вариант №25

Структура записей таблиц:

- 1) Предприятие Продукции Выпуск 1кв. Выпуск 2кв. Выпуск 3кв. Выпуск 4кв. Цена
- 2) Предприятие Адрес Дата основания

Контроль данных: выпуски и цена – натуральные числа, дата основания с 01.01.2005 по 31.12.2010

Запрос на изменение: увеличить на 10% цену произвольно заданной продукции

Запросы на выборку: а) вывести количество различных видов продукции

б) вывести сведения о самых старых предприятиях:

Предприятие Адрес Возраст Стоимость годового выпуска

Форма

Предприятие	<input type="text"/>
Продукция	<input type="text"/>
Годовой выпуск	<input type="text"/>

Отчет

Предприятие	Продукция	Адрес	Выпущено на сумму
.....
Итого по предприятию			

Практическая работа №5 Инструкции языка SQL: создание таблиц и схем базы данных

Вариант

1. Создать БД по учету кварт платы из двух таблиц:
Жильцы(**Код жильца**, ФИО, Улица, дом, квартира, площадь, кол-во проживающих)
Платеж (**Код платежа**, Код жильца, дата, сумма платежа)
2. Определить ключи, связав таблицы
3. Составить запросы:
 - все сведения о квартиросъемщиках с фамилией, начинающейся с «А»
 - данные по текущему месяцу, с подведением итогов
 - расчет квартплаты по каждому жильцу при следующих условиях – 2р. за кв. метр+ 0,5р. за каждого проживающего

2 Вариант

1. Создать БД по учету успеваемости студентов
Предметы(**Код_предмета**, название предмета, группа, ФИО преподавателя)

Студенты (**Код_студента**, ФИО, факультет, группа)

Экзамены(**Код**, Код предмета, Код_студента, оценка, дата сдачи)

Определить ключи, связав таблицы по заданным полям

2. Составить запросы:

- все сведения о студентах первого курса технологического факультета
- средний балл каждого студента заданной с клавиатуры группы
- сведения о студентах имеющих 2 по предметам
- средний балл по каждому курсу каждого факультета

3 Вариант

1. Создать БД по анализу отгрузки оборудования по заводу

2. Определить ключи, связав таблицы по заданными полями

3. Составить запросы:

- загрузка оборудования каждого цеха по кварталам, ч/з группировку
- данные по текущему месяцу, с подведением итогов
- сведения о заданном оборудовании
- о начислении штрафа цеху в размере 15%, если не выполнен план.

4 Вариант

1. Создать БД по учету поставки продукции по заводу Продукция (Код, Наименование, Кол-во, цена)

Заказчики (Код, ФИО, Адрес), Заказы (Код,

2. Определить ключи, связав таблицы по заданными полями

3. Составить запросы:

- план поставки продукции по заказчикам в разрезе кварталов
- информация о заданном заказчике
- итоговая ведомость оплаты заказчикам
- о расчете долга по уплате поставки

5 Вариант

1. Создать БД с информацией о кол-ве рабочих по цехам, профессиям, разрядам, для анализа средней зарплаты

2. Определить ключи, связав таблицы по заданными полями

3. Составить запросы:

- средняя з/плата по заводу в разрезе профессий и разрядов
- данные по определенной профессии с выводом информации о з/п
- сведения о заданном цехе
- о начислении премий работникам, перевыполнившим план, ч/з функцию.

6 Вариант

1. Создать БД по учету товара на складе и его отправке клиентам

2. Определить ключи, связав таблицы по заданными полями

3. Составить запросы:

- о клиентах сделавших заказ на МАХ сумму
- данные по текущему месяцу, с подведением итогов
- сведения о недостающих на складе товарах
- о начислении стоимости за товары каждого вида и заказа в целом

7 Вариант

1. Создать БД по анализу выполнения плана рабочими на п/п

2. Определить ключи, связав таблицы по заданными полями

3. Составить запросы:

- численность рабочих в определенном цехе
- данные по выполнению плана рабочими всего п/п
- сведения о нормативном плане на 1 рабочего (расчитать)
- о снятии штрафа в размере 1 мин. з/платы если не выполнен план, ч/з функцию

8 Вариант

1. Создать БД по учету выдачи в прокат видео кассет
2. Определить ключи, связав таблицы по заданными полями
3. Составить запросы:
 - о клиенте взявшем более 3 кассет
 - данные по клиентам, которым начисляется штраф, в связи с невозвращением кассет, ч/з функцию
 - сведения о всех клиентах с учетом стоимости
 - о кассетах с фильмами-боевиками

9 Вариант

1. Создать БД по учету выдачи в прокат книг в библиотеке
2. Определить ключи, связав таблицы по заданными полями
3. Составить запросы:
 - о клиенте взявшем более 3 книг
 - данные по клиентам, которым начисляется штраф, в связи с невозвращением книг, ч/з функцию
 - сведения о всех клиентах с учетом стоимости
 - о книгах-детективах

Практическая работа №6 Создание базы данных на сервере. Инструментальные средства многопользовательских СУБД. Создание таблиц и схем данных. Типы данных. Определение параметров целостности: ключи, индексы, ограничения на значения, связная целостность.

Задание

1. Создайте базу данных
2. Создайте новую таблицу
3. Определите поля таблицы
4. Сохраните созданную таблицу

Порядок работы

Для создания новой базы данных:

1) На странице **Приступая к работе с Microsoft Office Access** в разделе **Новая пустая база данных** выберите команду **Новая база данных**.



2) В области **Новая база данных** в поле **Имя файла** введите имя файла. Если имя файла указано без расширения, расширение будет добавлено автоматически (*.accdb). Чтобы сохранить файл в другой папке, отличной от используемой по умолчанию, нажмите кнопку **Открыть** (рядом с полем **Имя файла**), перейдите к нужной папке (с:\Access, папку создать предварительно), и нажмите кнопку **ОК**.

3) Имя базы данных задайте *Деканат*, а тип файла оставьте прежним, так как другие типы файлов нужны в специальных случаях.

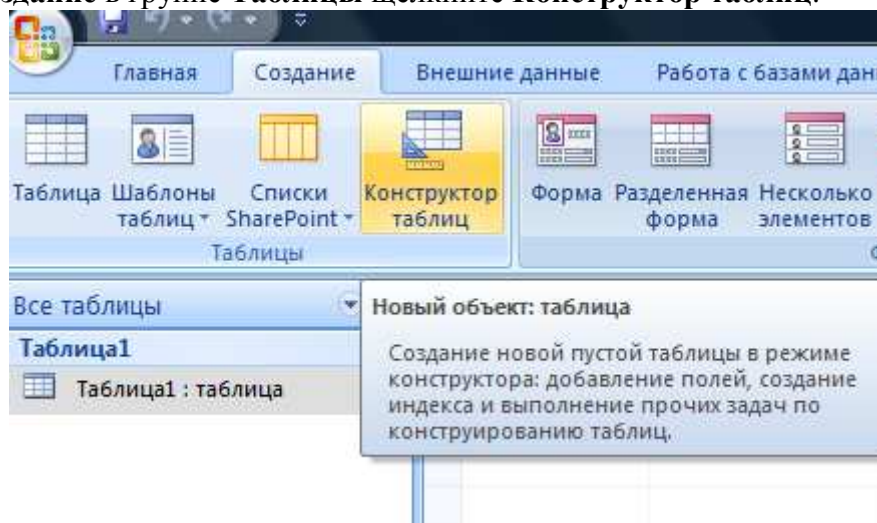
4) Нажмите кнопку **Создать**.

2. Для создания таблицы базы данных:

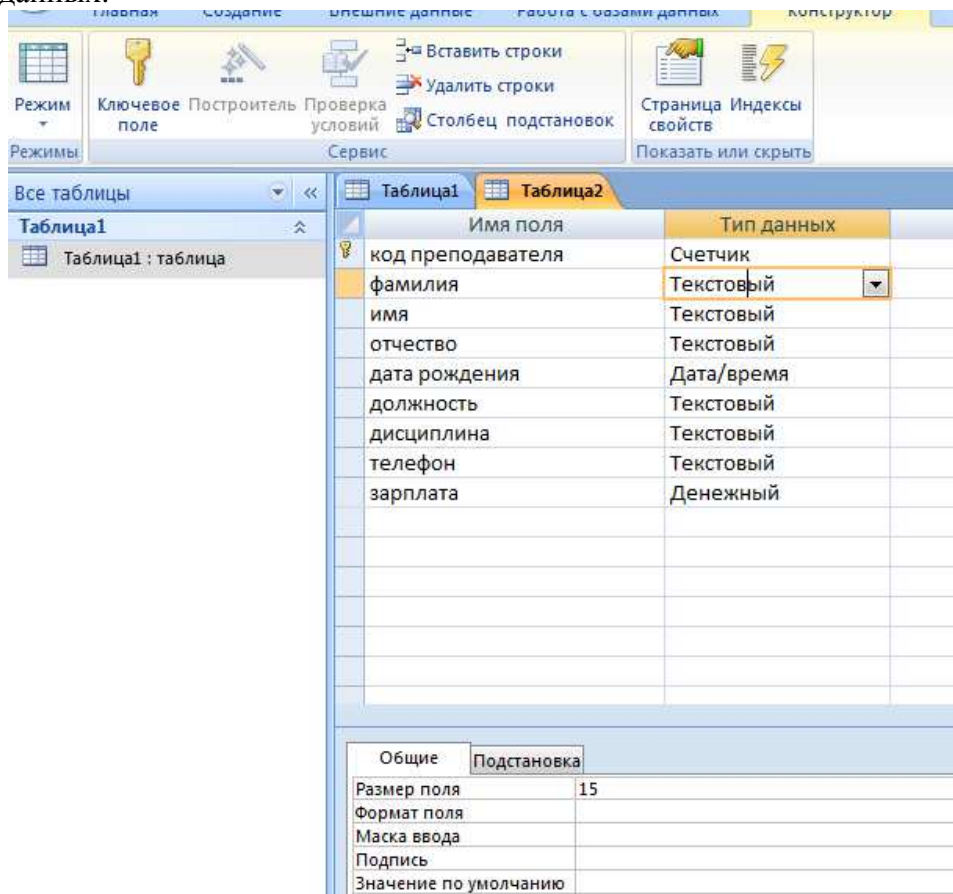
1) Приложение Access создаст базу данных с пустой таблицей с именем «Таблица1» и откроет эту таблицу в режиме таблицы. Курсор находится в первой пустой ячейке столбца, **Добавить поле**.

Создание таблицы в режиме конструктора В режиме конструктора сначала создается структура новой таблицы. Затем можно переключиться в режим таблицы для ввода данных или ввести данные, используя другой метод, например вставку или импорт.

2) На вкладке **Создание** в группе **Таблицы** щелкните **Конструктор таблиц**.



3) Для каждого поля в таблице введите имя в столбце **Имя поля**, а затем в списке **Тип данных** выберите тип данных.



ПРИМЕЧАНИЕ. Если столбцы **Имя поля** и **Тип данных** не отображаются, возможно, вместо режима конструктора выбран режим таблицы. Чтобы переключиться в режим конструктора, в области переходов щелкните таблицу правой кнопкой мыши, а затем выберите команду **Режим конструктора**. Будет предложено ввести имя для новой таблицы, а затем произойдет переключение в режим конструктора.

4) Когда все необходимые поля будут добавлены, сохраните таблицу: Щелкните значок **Кнопка Microsoft Office**, а затем выберите команду **Сохранить**.

5) Чтобы добавить данные, переключитесь в режим таблицы начните их вводить, щелкнув первую пустую ячейку.

3. Для определения полей таблицы:

- введите в строку столбца «Имя поля» имя первого поля *Код преподавателя*;
- в строке столбца «Тип данных» щелкните по кнопке списка и выберите тип данных *Счетчик*. Поля вкладки *Общие* оставьте такими, как предлагает Access.

4. Для сохранения таблицы:

- выберите пункт меню **Файл, Сохранить**;
- в диалоговом окне «Сохранение» введите имя таблицы *Преподаватели*;
- щелкните по кнопке <ОК>.

Примечание. В предыдущих версиях офиса результатом щелчка по кнопке <ОК> Access предлагал вам задать ключевое поле (поле первичного ключа), т.е. поле, однозначно идентифицирующее каждую запись. В нашем случае это поле «Код преподавателя». Встаньте указателем на название поля и нажмите на кнопку с изображением ключа на панели инструментов. При создании новой таблицы в режиме таблицы приложение Access 2007 автоматически создает первичный ключ и присваивает ему тип данных «Счетчик». Если нужно добавить поле первичного ключа к уже созданной таблице, следует открыть таблицу в режиме конструктора.


Заполнение базы данных

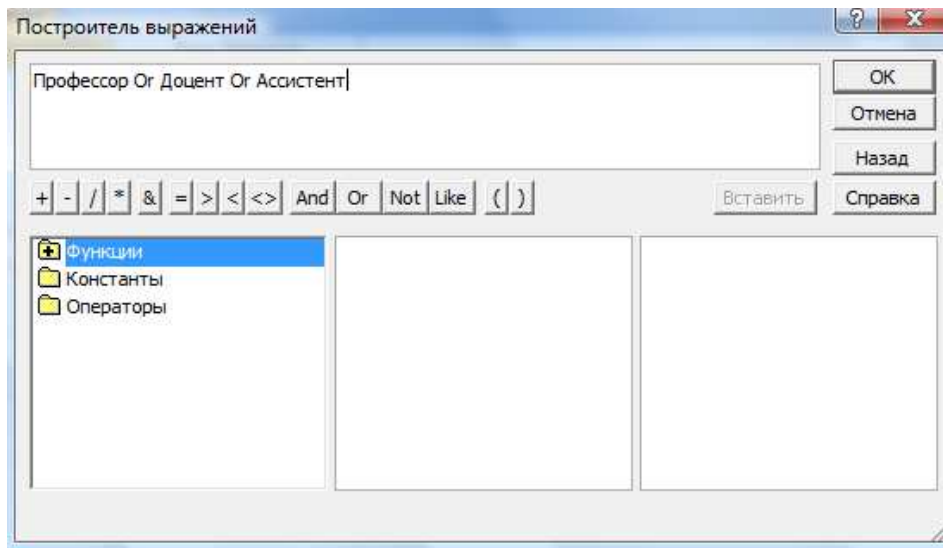
Задание

1. Введите ограничения на данные, вводимые в поле «Должность»; должны вводиться только слова *Профессор, Доцент или Ассистент*.
2. Задайте текст сообщения об ошибке, который будет появляться на экране при вводе неправильных данных в поле «Должность».
3. Задайте значение по умолчанию для поля «Должность» в виде слова *Доцент*.
4. Введите ограничения на данные в поле «Код преподавателя»; эти данные не должны повторяться.
5. Создайте столбец подстановок для поля «Дисциплина».
6. Заполните таблицу данными в соответствии с таблицей 4 и проверьте реакцию системы на ввод неправильных данных в поле «Должность».
7. Измените ширину каждого поля таблицы в соответствии с шириной данных.
8. Произведите поиск в таблице преподавателя Миронова.
9. Произведите замену данных: измените заработную плату ассистенту Сергеевой с 4500 р. на 4700 р.
10. Произведите сортировку данных в поле «Год рождения» по убыванию.
11. Произведите фильтрацию данных по полям «Должность» и «Дисциплина».
12. Просмотрите созданную таблицу, как она будет выглядеть на листе бумаги при печати.

Порядок работы

1. Для задания условия на значение для вводимых данных:

- войдите в режим *Конструктор* для проектируемой таблицы. Если вы находитесь в окне базы данных, то выберите вкладку *Главная* и щелкните по группе команд <Режим>. Если вы находитесь в режиме таблицы, то щелкните по кнопке *Конструктор*;
- в верхней части окна щелкните по полю «Должность»;
- в нижней части окна щелкните по строке параметра *Условие на значение*;
- щелкните по кнопке  для определения условий на значение при помощи построителя выражений;



- в появившемся окне напишите слово *Профессор*, затем щелкните по кнопке **Or** (эта кнопка выполняет функцию ИЛИ), напишите *Доцент*, снова щелкните по этой же кнопке, напишите *Ассистент* и щелкните по кнопке <ОК>. Таким образом вы ввели условие, при котором в поле «Должность» могут вводиться только указанные значения.

2. В строке Сообщение об ошибке введите предложение "Такой должности нет, правильно введите данные".

3. В строке Значение по умолчанию введите слово "Доцент".

4. Введите ограничения на данные в поле «Код преподавателя». Здесь ограничения надо вводить не совсем обычным способом. Дело в том, что коды преподавателей не должны повторяться, а также должна быть обеспечена возможность их изменения (из-за последнего условия в этом поле нельзя использовать тип данных *Счетчик*, в котором данные не повторяются). Для выполнения второго условия измените тип поля «Код преподавателя» со «Счетчика» на «Числовой», а для выполнения первого условия сделайте следующее:

- щелкните по строке параметра *Индексированное поле*;

Примечание. Индекс — это средство Access, ускоряющее поиск и сортировку данных в таблице. Ключевое поле (поле первичного ключа) таблицы индексируется автоматически. Не допускается создание индексов для полей типа *МЕМО* и *Гиперссылка* или полей объектов *OLE*. Свойство *Индексированное поле* определяет индекс, создаваемый по одному полю. Индексированное поле может содержать как уникальные, так и повторяющиеся значения. Допускается создание произвольного количества индексов.

- выберите в списке пункт **Да (совпадения не допускаются)**;

Внимание! Убедитесь, что эту операцию Вы проделали именно для поля «Код преподавателя», а не для поля «Должность» !

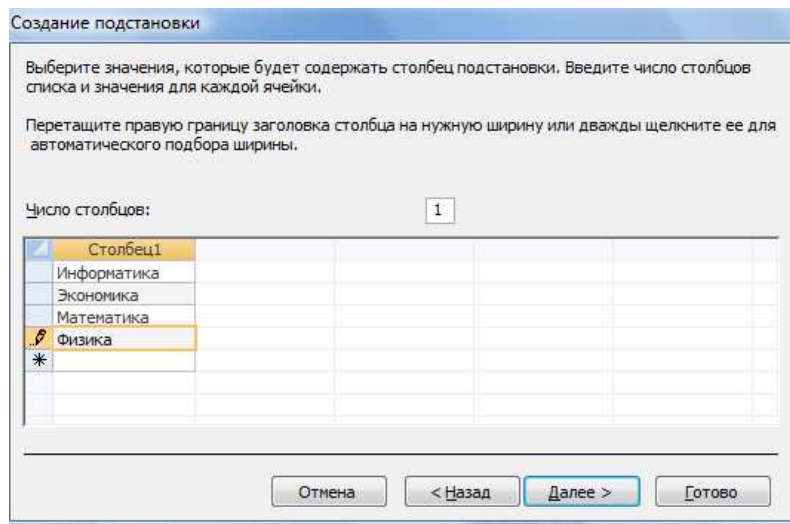
5. Для удобного заполнения таблицы данными воспользуемся возможностями Мастера подстановок:

- измените тип поля «Дисциплина» с *текстового* на *Мастер подстановок*(последняя запись в выпадающем списке);

- отметьте пункт «*Будет введен фиксированный набор значений*»;

- щелкните по кнопке *Далее*;

- число столбцов – 1, в столбец1 введите значения: Информатика, Экономика, Математика, Физика;



- щелкните по кнопке «Далее»; подпись оставьте прежней - «Дисциплина».
- Готово;
- перейдите в режим **Таблица**, щелкнув по кнопке на линейке во вкладке *Главная*, в группе команд Режим, выбрав команду *Режим таблицы*. На вопрос о сохранении таблицы щелкните по кнопке <Да>.

6. Введите данные в таблицу.

- Попробуйте в поле <Должность> любой записи ввести слово *Лаборант*. Посмотрите, что получилось. На экране должно появиться сообщение: "Такой должности нет, правильно введите данные". Введите правильное слово.

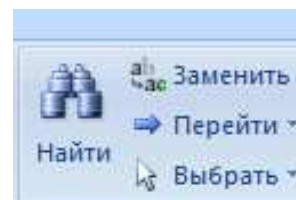
код препод	фамилия	имя	отчество	дата рожде	должность	дисциплина	телефон	зарплата	Доба
1	Истомин	Ремин	Евгеньевич	23.10.1954	Доцент	Информатика	110-22-33	8 500,00р.	
2	Миронов	Павел	Юрьевич	25.07.1960	Профессор	Экономика	110-33-44	12 000,00р.	
3	Гришин	Евгений	Сергеевич	12.09.1974	Доцент	Математика	269-77-88	7 600,00р.	
4	Сергеева	Ольга	Ивановна	12.02.1984	Ассистент	Математика	123-45-67	4 500,00р.	
5	Емец	Татьяна	Ивановна	16.02.1950	Доцент	Экономика	546-78-90	6 500,00р.	
6	Игнатъева	Наталья	Павловна	30.05.1970	Доцент	Информатика	125-67-32	7 900,00р.	
7	Миронов	Алексей	Николаевич	17.06.1949	Доцент	Физика	100-00-33	8 500,00р.	
*					Доцент				

7. Для изменения ширины полей таблицы в соответствии с шириной данных:

- щелкните в любой строке поля «Код преподавателя»;
- выполните команду на вкладке ленты на вкладке ленты **Главная, Запись, Дополнительно, Ширина столбца**;
- в появившемся окне щелкните по кнопке <По ширине данн...>
- проделайте эту операцию с остальными полями.

8. Для поиска в таблице преподавателя Миронова:

- переведите курсор в первую строку поля «Фамилия»;
- выполните команду на вкладке ленты **Главная, Найти**;



- в появившейся строке параметра *Образец* введите *Миронов*;
- в строке параметра *Просмотр* должно быть слово *ВСЕ* (имеется в виду искать по всем записям);
- в строке параметра *Совпадение* выберите из списка *С любой частью поля*;
- в строке параметра *Только в текущем поле* установите флажок (должна стоять "галочка");
- щелкните по кнопке <Найти далее>. Курсор перейдет на вторую запись и выделит слово *Миронов*;
- щелкните по кнопке <Найти далее>. Курсор перейдет на седьмую запись и также выделит слово *Миронов*;

- щелкните по кнопке <Заккрыть> для выхода из режима поиска.


9. Для замены заработной платы ассистенту Сергеевой с 4500 р. на 4700 р.(изменить формат поля Денежный, десятичная часть установить 0):

- переведите курсор в первую строку поля «Зарплата»;
- выполните команду на вкладке ленты **Главная, Заменить**;
- в появившемся окне в строке *Образец* введите 4500;
- в строке *Заменить на* введите 4700. Обратите внимание на остальные опции — вам надо вести поиск по всем записям данного поля;
- щелкните по кнопке <Найти далее>. Курсор перейдет на четвертую запись;
- щелкните по кнопке <Заменить>. Данные будут изменены;
- щелкните по кнопке <Заккрыть>.

10. Для сортировки данных в поле «Год рождения» по убыванию:

■ щелкните по любой записи поля «Год рождения»;




■ щелкните по кнопке  на панели управления или выполните команду на вкладке *Главная: Сортировка и фильтр, Сортировка по убыванию*. Все данные в таблице будут отсортированы в соответствии с убыванием значений в поле «Год рождения».

11. Для фильтрации данных по полям «Должность» и «Дисциплина»:


- щелкните по записи *Доцент* поля «Должность»;




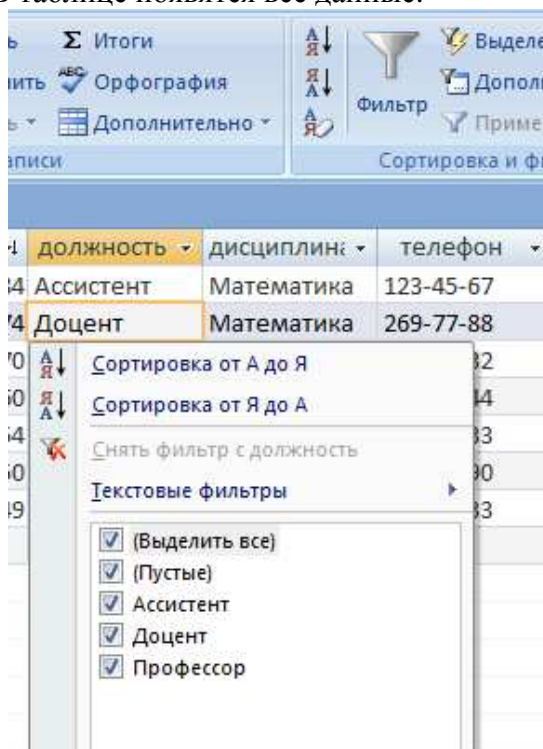
- щелкните по кнопке  или выполните команду **Сортировка и фильтр, Фильтр по выделенному**. В таблице останутся только записи о преподавателях — доцентах;

- щелкните по записи *Информатика* поля «Дисциплина»;



- щелкните по кнопке  или выполните команду **Сортировка и фильтр, Фильтр по выделенному**. В таблице останутся только записи о преподавателях — доцентах кафедры информатики;

- для отмены фильтрации щелкните по кнопке  на панели инструментов или выполните команду **Сортировка и фильтр, Удалить фильтр**. В таблице появятся все данные.



дата рожде	должность	дисциплина	тел
12.02.1984	ассистент	Математика	123-
12.09.1974	Доцент	Математика	269-
30.05.1970	Доцент	Информатика	125-
25.07.1960	Профессор	Экономика	110-
23.10.1954	Доцент	Информатика	110-
16.02.1950	Доцент	Экономика	546-

12. Для просмотра созданной таблицы:

- Щелкните значок **Кнопка Microsoft Office**, а затем выберите команду **Печать, Предварительный просмотр**.
- закройте окно просмотра.

Практическая работа №7 Система безопасности серверных баз данных. Разграничение доступа к данным.

1. Создайте структуру таблицы *Студенты*.
2. Создайте структуру таблицы *Дисциплины*.
3. Измените структуру таблицы *Преподаватели*.
4. Создайте структуру таблицы *Оценки*.
5. Разработайте схему данных, т.е. создайте связи между таблицами.

Структура таблицы Студенты

Имя поля	Тип данных	Размер поля
Код студента	Числовой	Целое
Фамилия	Текстовый	15
Имя	Текстовый	12
Отчество	Текстовый	15
Номер группы	Числовой	Целое
Телефон	Текстовый	9
Стипендия	Логический	Да/Нет

Структура таблицы *Преподаватели* уже создана в работе 1 и заполнена данными, поэтому для работы используйте эту таблицу с одним лишь изменением — в структуру таблицы надо добавить поле «Код дисциплины» (в режиме конструктора) и заполнить его.

Структура таблицы Дисциплины

Имя поля	Тип данных	Размер поля
Код дисциплины	Числовой	Целое
Название дисциплины	Текстовый	30

Структура таблицы Оценки

Имя поля	Тип данных	Размер поля
Код студента	Числовой	Целое
Код дисциплины	Числовой	Целое
Оценки	Числовой	Байт

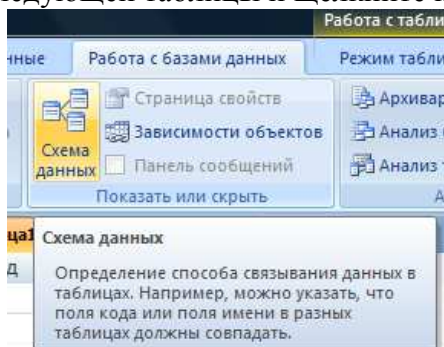
В этой таблице задавать ключевое поле не надо, так как данные во всех полях могут повторяться.

Разработайте схему данных, т.е. создайте связи между таблицами.

Для этого:

- щелкните по кнопке **Схема данных** на вкладке *Работа с базами данных*. На экране появится окно «Схема данных»;
- через контекстное меню добавьте таблицы: щелкните по кнопке **Добавить таблицу**;
- в появившемся окне будет выделено название одной таблицы. Щелкните по кнопке <Добавить>;

- переведите выделение на имя следующей таблицы и щелкните по кнопке <Добавить>.



Аналогично добавьте оставшиеся две таблицы;

- закройте окно, щелкнув по кнопке <Закрывать>;

- создайте связь между таблицами *Дисциплины* и *Оценки*. Для этого подведите курсор мыши к полю «Код дисциплины» в таблице *Дисциплины*, щелкните левой кнопкой мыши и, не отпуская ее, перетащите курсор на поле «Код дисциплины» в таблицу *Оценки*, а затем отпустите кнопку мыши. На экране откроется окно «Связи»;

- установите флажок ("галочку") в свойстве *Обеспечение целостности данных*, щелкнув по нему;

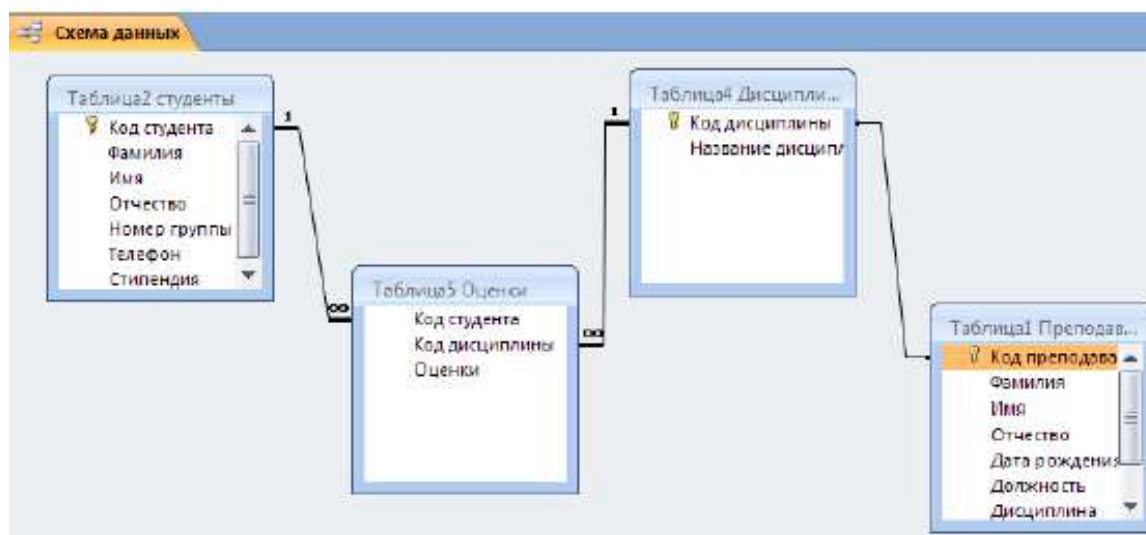
- установите флажок в свойствах *Каскадное обновление связанных полей* и *Каскадное удаление связанных записей*;

Примечание. Задание каскадного обновления связанных полей и каскадного удаления связанных записей позволит вам отредактировать записи только в таблице *Дисциплины*, а в таблице *Оценки* эти действия будут со связанными записями выполняться автоматически. Например, если вы удалите из таблицы *Дисциплины* один предмет, то в таблице *Оценки* удалятся все строки, связанные с этим предметом.

- щелкните по кнопке <Создать>. Связь будет создана;

- аналогично создайте связи между полем «Код дисциплины» в таблице *Дисциплины* и полем «Код дисциплины» в таблице *Преподаватели*, а также между полем «Код студента» в таблице *Студенты* и полем «Код студента» в таблице *Оценки*.

- закройте окно схемы данных, ответив *ДА* на вопрос о сохранении макета.



Задание №2

1. Создайте форму *Студенты*.
2. Заполните данными таблицу *Студенты* посредством формы *Студенты*.
3. Создайте форму *Дисциплины*.
4. Заполните данными таблицу *Дисциплины* посредством формы *Дисциплины*.
5. Создайте форму *Оценки*.

6. Заполните данными таблицу *Оценки* посредством формы *Оценки*.

Таблица Студенты

Код студента	Фамилия	Имя	Отчество	Номер группы	Телефон	Стипендия
1	Арбузов	Николай	Николаевич	151	260-15-63	Да
2	Киршин	Петр	Валерьевич	151	110-67-82	Да
3	Кривинский	Сергей	Николаевич	151	172-97-21	Нет
4	Крылова	Елена	Петровна	151	130-31-87	Да
5	Кульчий	Григорий	Викторович	151	269-53-75	Да
6	Патрикеев	Олег	Борисович	152	234-11-63	Нет
7	Перлов	Кирилл	Николаевич	152	312-21-33	Нет
8	Соколова	Наталья	Петровна	152	166-87-24	Нет
9	Степанская	Ольга	Витальевна	152	293-43-77	Да
10	Тимофеев	Сергей	Иванович	152	260-11-57	Да

Таблица Дисциплины

Код дисциплины	Название дисциплины
1	Информатика
2	Математика
3	Физика
4	Экономика

Аналогично заполните формы *Оценки*

Лабораторная работа №1
Тема: Работа в MS Access 2010

Задание 1.

На АЗС продают бензин нескольких марок. Известна стоимость литра бензина каждой марки. Создать БД, содержащую сведения о продаже бензина и позволяющую совершать перечисленные ниже операции.

1. Выписывать чек при продаже бензина каждому клиенту.
2. Выдавать по запросу сведения:
 - об объемах продаж по марке бензина;
 - о количестве клиентов, обслуженных на АЗС, в заданном интервале дат.
3. Выдавать отчет о продажах бензина по дням, по каждой марке отдельно и итоговый объем продажи.

Задание 2.

Обменный пункт производит продажу и обмен валюты нескольких видов (доллары, марки, лиры и т. д.). Известны курсы покупки и продажи каждой валюты. Создать БД, содержащую сведения об операциях с валютой и позволяющую совершать перечисленные ниже операции.

1. Выписывать справку при продаже/покупке валюты.
2. Выдавать по запросу сведения:
 - об объемах продаж/покупок по заданной валюте (в валюте и рублевом эквиваленте);
 - о количестве клиентов, обслуженных обменным пунктом, в заданном интервале дат.
3. Выдавать отчет об операциях с валютой по дням, по каждому

виду отдельно и итоговый объем продаж/покупок.

Задание 3.

Создать БД библиотеки, содержащую сведения об имеющейся литературе и позволяющую совершать перечисленные ниже операции.

1. Вести учет поступающей литературы.
2. Списывать книги, изданные более 10 лет назад.
3. Вести учет выдачи книг читателям.
4. Удерживать штраф в размере 3-х кратной стоимости книги при ее потере.
5. Выдавать по запросу сведения о наличии книг:
 - по фамилии автора;
 - по количеству изданий (например, книги выдержавшие более 2-х изданий);
 - по издательству.
6. Выдавать по запросу сведения о количестве потерянных книг и сумму полученных штрафов.
7. Выдавать отчет о наличии книг в библиотеке с группировкой по издательствам.

Задание 4.

На оптовой базе реализуют продовольственные товары. Известна стоимость каждого товара. Создать БД, содержащую сведения о продаже товаров и позволяющую совершать перечисленные ниже операции.

1. Выписывать чек при продаже товара каждому клиенту.
2. Выдавать по запросу сведения:
 - об объемах продаж по каждому наименованию;
 - о количестве клиентов, обслуженных на оптовой базе, в заданном интервале дат.
3. Выдавать отчет о продажах товара по дням, по каждому наименованию отдельно и итоговый объем продаж.

Задание 5.

Создать БД о движении поездов, содержащую сведения об имеющихся маршрутах и позволяющую совершать перечисленные ниже операции.

1. Вести учет маршрутов поездов.
2. Учет проданных билетов.
3. Выдавать по запросу сведения о наличии проданных билетов:
 - по фамилии пассажира;
 - по маршруту следования;
 - по датам продажи.
4. Выдавать отчет о наличии билетов в кассе с группировкой по маршрутам следования и итоговый объем продаж

Задание 6.

Создать БД продажи ЭВМ, содержащую сведения об имеющихся наименованиях и позволяющую совершать перечисленные ниже операции.

1. Вести учет поступившей техники.
2. Списывать морально-устаревшие ЭВМ.
3. Вести учет продажи ЭВМ.
4. Выдавать по запросу сведения о наличии ЭВМ:
 - по имеющемуся количеству на складе;
 - по дате поступления на склад;
 - по заводу производителю.
8. Выдавать отчет о наличии технических средств на складе с группировкой по производителям.

Задание 7.

Создать БД проката CD-дисков, содержащую сведения об имеющихся в прокате дисках и позволяющую совершать перечисленные ниже операции.

1. Вести учет, поступающих в прокат, CD-дисков.
2. Списывать диски, имеющие повреждения.

3. Вести учет выдачи дисков клиентам.
4. Удерживать штраф в размере 3-х кратной стоимости диска при его потере.
5. Выдавать по запросу сведения о наличии CD-дисков:
 - по названиям;
 - по количеству;
 - по производителю.
9. Выдавать по запросу сведения о количестве потерянных дисков и сумму полученных штрафов.
10. Выдавать отчет о наличии дисков в прокате с группировкой по производителям.

Задание 8.

Создать БД проката видеокассет, содержащую сведения, об имеющихся в прокате кассетах, и позволяющую совершать перечисленные ниже операции.

6. Вести учет о новых поступлениях кассет.
7. Списывать кассеты, имеющие повреждения.
8. Вести учет выдачи кассет клиентам.
9. Удерживать штраф в размере 3-х кратной стоимости кассеты при ее потере.
10. Выдавать по запросу сведения о наличии кассет:
 - по названиям;
 - по количеству;
 - по заводу производителю.
11. Выдавать по запросу сведения о количестве потерянных кассет и сумму полученных штрафов.
12. Выдавать отчет о наличии кассет в прокате с группировкой по производителям.

Задание 9

В аптеке продают лекарства различных наименований. Известна стоимость лекарства каждого наименования. Создать БД, содержащую сведения о продаже лекарств и позволяющую совершать перечисленные ниже операции.

1. Выписывать чек при продаже лекарства каждому покупателю.
2. Списывать просроченные лекарства.
3. Выдавать по запросу сведения:
 - об объемах продаж по наименованию лекарства.
 - о поступлении нового лекарства в аптеку.
4. Выдавать отчет о продажах лекарства по дням, по каждому наименованию отдельно и итоговый объем продаж.

Задание 10

Создать БД о погрузке различного груза, содержащую сведения об имеющихся наименованиях грузов и позволяющую совершать перечисленные ниже операции.

5. Вести учет объема отгрузки.
6. Выписывать накладную на отгрузку.
7. Выдавать по запросу сведения о наличии груза:
 - по имеющемуся количеству на складе;
 - по дате поступления на склад;
 - по дате отгрузки со склада.
13. Выдавать отчет о наличии технических средств на складе с группировкой по производителям.

Задание 11.

На АЗС продают бензин нескольких марок. Известна стоимость литра бензина каждой марки. Создать БД, содержащую сведения о продаже бензина и позволяющую совершать перечисленные ниже операции.

3. Выписывать чек при продаже бензина каждому клиенту.
4. Выдавать по запросу сведения:
 - об объемах продаж по марке бензина;
 - о количестве клиентов, обслуженных на АЗС, в заданном интервале дат.

3. Выдавать отчет о продажах бензина по дням, по каждой марке отдельно и итоговый объем продажи.

Задание 12.

Обменный пункт производит продажу и обмен валюты нескольких видов (доллары, марки, лиры и т. д.). Известны курсы покупки и продажи каждой валюты. Создать БД, содержащую сведения об операциях с валютой и позволяющую совершать перечисленные ниже операции.

3. Выписывать справку при продаже/покупке валюты.
4. Выдавать по запросу сведения:
 - об объемах продаж/покупок по заданной валюте (в валюте и рублевом эквиваленте);
 - о количестве клиентов, обслуженных обменным пунктом, в заданном интервале дат.
3. Выдавать отчет об операциях с валютой по дням, но каждому виду отдельно и итоговый объем продаж/покупок.

Задание 13.

Создать БД библиотеки, содержащую сведения об имеющейся литературе и позволяющую совершать перечисленные ниже операции.

6. Вести учет поступающей литературы.
7. Списывать книги, изданные более 10 лет назад.
8. Вести учет выдачи книг читателям.
9. Удерживать штраф в размере 3-х кратной стоимости книги при ее потере.
10. Выдавать по запросу сведения о наличии книг:
 - по фамилии автора;
 - по количеству изданий (например, книги выдержавшие более 2-х изданий);
 - по издательству.
14. Выдавать по запросу сведения о количестве потерянных книг и сумму полученных штрафов.
15. Выдавать отчет о наличии книг в библиотеке с группировкой по издательствам.

Задание 14.

На оптовой базе реализуют продовольственные товары. Известна стоимость каждого товара. Создать БД, содержащую сведения о продаже товаров и позволяющую совершать перечисленные ниже операции.

3. Выписывать чек при продаже товара каждому клиенту.
4. Выдавать по запросу сведения:
 - об объемах продаж по каждому наименованию;
 - о количестве клиентов, обслуженных на оптовой базе, в заданном интервале дат.
3. Выдавать отчет о продажах товара по дням, по каждому наименованию отдельно и итоговый объем продажи.

Задание 15.

Создать БД о движении поездов, содержащую сведения об имеющихся маршрутах и позволяющую совершать перечисленные ниже операции.

5. Вести учет маршрутов поездов.
6. Учет проданных билетов.
7. Выдавать по запросу сведения о наличии проданных билетов:
 - по фамилии пассажира;
 - по маршруту следования;
 - по датам продажи.
8. Выдавать отчет о наличии билетов в кассе с группировкой по маршрутам следования и итоговый объем продаж

Задание 16.

Создать БД продажи ЭВМ, содержащую сведения об имеющихся наименованиях и позволяющую совершать перечисленные ниже операции.

8. Вести учет поступившей техники.
9. Списывать морально-устаревшие ЭВМ.

10. Вести учет продажи ЭВМ.
11. Выдавать по запросу сведения о наличии ЭВМ:
 - по имеющемуся количеству на складе;
 - по дате поступления на склад;
 - по заводу производителю.
16. Выдавать отчет о наличие технических средств на складе с группировкой по производителям.

Лабораторная №2 Технология доступа к базе данных ADO в среде программирования Delphi

Создайте в BDE Administrator новый псевдоним со своей фамилией. Укажите драйвер БД по умолчанию Paradox, а в параметре Path укажите путь к созданной вами папке.

Для этого необходимо выбрать команду Пуск – Программы – Delphi 7 – BDE Administrator.

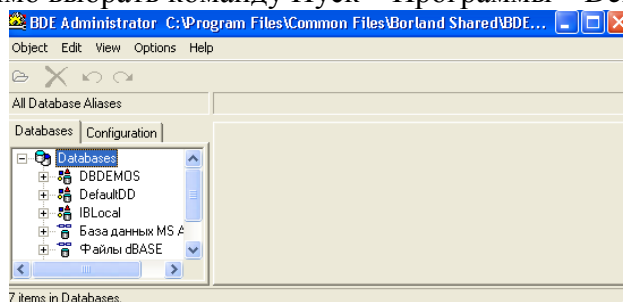


Рисунок 1

Для работы с таблицами БД при проектировании приложения удобно использовать Database Desktop, которая позволяет:

- создавать таблицы
- изменять структуры
- редактировать записи.

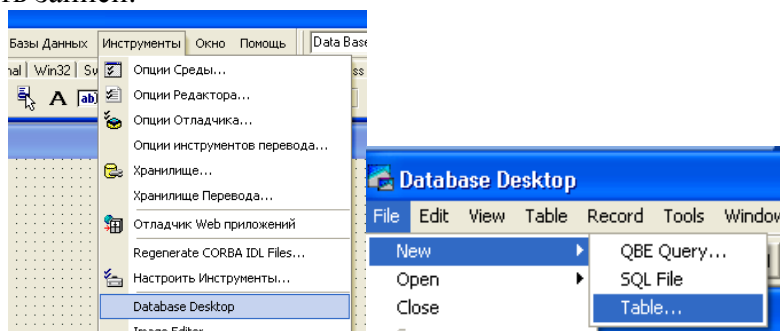


Рисунок 2

Кроме того, с помощью Database Desktop можно выполнять и другие действия над БД (создание, редактирование и выполнение визуальных и SQL – запросов, операций с псевдонимами).

Процесс создания новой таблицы начинается с вызова команды File/New/Table (Файл/Новая/Таблица) и происходит в интерактивном режиме. При этом разработчик должен:

- выбрать формат (тип) таблицы
- задать структуру таблицы.

В начале создания новой таблицы в окне Create Table (создание таблицы) выбирается ее формат. По умолчанию предлагается формат таблицы Paradox, который и будем использовать. Для таблиц других форматов, например dBase IV, действия по созданию таблицы практически не отличаются.

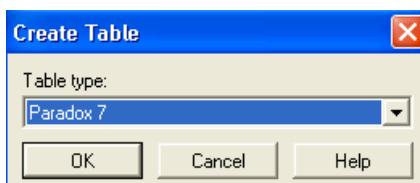


Рисунок 3

После выбора формата таблицы появляется окно определения структуры таблицы, в котором выполняются следующие действия:

- описание полей
- задание ключа
- задание индексов
- определение ограничений на значения полей
- определение условий ссылочной целостности
- задание паролей
- задание языкового драйвера
- задание таблицы для выбора значений

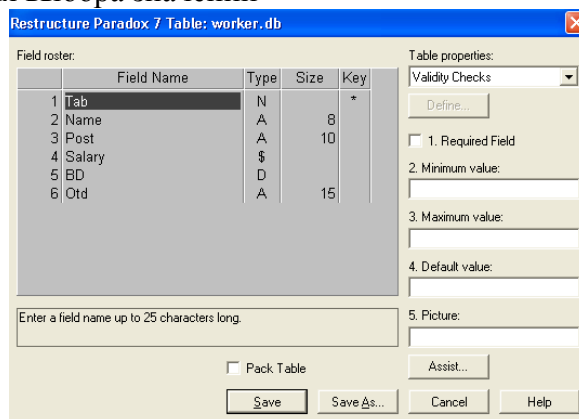


Рисунок 4

В этом списке обязательным является первое действие, т.е. каждая таблица должна иметь хотя бы одно поле. Остальные действия выполняются при необходимости. Часть действий, такие как задание ключа и паролей, производится только для таблиц определенных форматов, например, для таблиц Paradox.

При создании новой таблицы, сразу после выбора ее формата можно не создавать структуру таблицы, а скопировать ее из другой таблицы: при нажатии кнопки Borrow (Взаймы) открывается окно Select Borrow Table (выбор таблицы для заимствования).

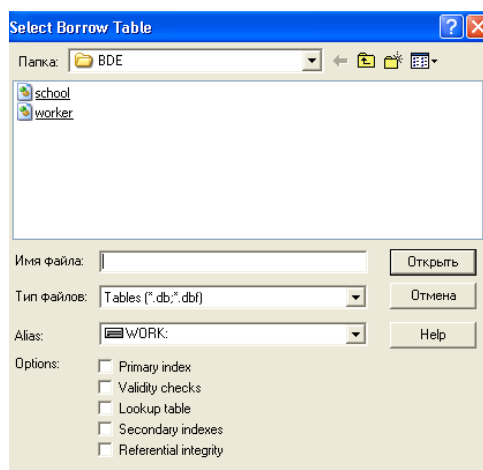


Рисунок 5

В этом окне можно выбрать таблицу и указать копируемые элементы структуры, установив соответствующий флажок, например, Primary index (первичный индекс) для ключа. После нажатия кнопки Open из выбранной таблицы в новую копируются описания полей, а также те элементы, для которых установлен флажок. Если какой-либо элемент в структуре копируемой таблицы отсутствует, то состояние флажка не имеет значения. Например, если в выбранной таблице не определены ограничения ссылочной целостности, то в новой таблице они не появятся, даже если установлен флажок Referential integrity (ссылочная целостность).

Впоследствии скопированную структуру можно настраивать, изменяя, добавляя или удаляя ее отдельные элементы.

После определения структуры таблицы ее необходимо сохранить, нажав кнопку Save as и указав расположение таблицы на диске и ее имя. В результате на диск записывается новая таблица, первоначально пустая, при этом все необходимые файлы создаются автоматически.

Описание полей

Центральной частью окна определения структуры таблицы является список Field roster (список полей), в котором указываются поля таблицы. Для каждого поля задаются:

- имя поля – в столбце Field Name
- тип поля – в столбце Type
- размер поля - в столбце Size

Имя поля вводится по правилам, установленным для выбранного формата таблиц.

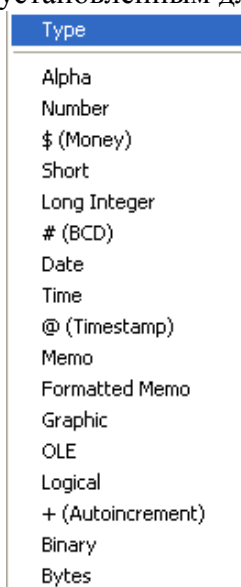


Рисунок 6

Тип поля можно задать непосредственно указав соответствующий символ, выбрав его в списке. Список содержит все типы полей, допустимые для заданного формата таблицы.

Размер поля задается не всегда, необходимость его указания зависит от типа поля. Для полей определенного типа, например, автоинкрементного (+) или целочисленного, размер поля не задается. Для поля строкового типа размер определяет максимальное число символов, которые могут храниться в поле. Добавление к списку полей новой строки выполняется переводом курсора вниз на несуществующую строку, в результате чего эта строка появляется в конце списка.

Для указания ключевых полей в столбце ключа (Key) нужно установить символ *, переведя в эту позицию курсор и нажав любую клавишу.

Для выполнения остальных действий по определению структуры таблицы используется комбинированный список Table Properties (свойства таблицы).

Создание приложения BDE

Для примера рассмотрим создание приложения, использующего механизм доступа BDE и позволяющего перемещаться по записям таблицы БД, просматривать и редактировать поля, удалять записи из таблицы, а также вставлять новые. Файл проекта приложения обычно не требует от разработчика выполнения каких – либо действий. Поэтому при создании приложения главной задачей является конструирование форм, в простейшем случае – одной формы.

Вид формы приложения на этапе проектирования показан на рисунке, где в форме размещены компоненты Table1, DataSource1,DBGrid, DBNavigator1.

Компонент Table1 обеспечивает взаимодействие с таблицей БД. Для связи с требуемой таблицей нужно установить в соответствующее значения свойство DataBaseName, указывающее путь к БД, и свойство TableName, указывающее имя таблицы. После задания таблицы для открытия набора данных свойство Active должно быть установлено в значение True.

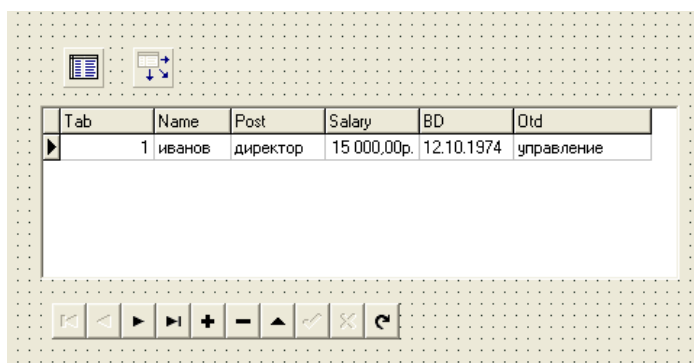


Рисунок 7

Компонент DBGrid отображает содержимое таблицы БД в виде сетки, в которой столбцы соответствуют полям, а строки – записям таблицы. По умолчанию пользователь может просматривать и редактировать данные. Компонент DBNavigator позволяет пользователю перемещаться по таблице, редактировать, вставлять и удалять записи. Компоненты DBGrid и DBNavigator связываются со своим источником данных – компонентом DataSource1 – через свойства DataSource.

Порядок выполнения задания

1) Создайте форму приложения в Delphi. На форму поместите компоненты Table1 (вкладка BDE) и DataSource1(вкладка Data Access), которые обеспечивают доступ к файлу данных, и установите следующие значения их свойств:

Свойство	Значение
Table1.DatadaseName	***
Table1.TableName	***.db
Table1.Active	True
DataSource1.Dataset	Table1

2) Для просмотра и редактирования данных в режиме таблицы в форму надо добавить компонент DBGrid1 (вкладка Data Controls).

3) Чтобы задать, какая информация будет отображаться в таблице DBGrid1 , необходимо задать источник данных для таблицы (значение свойства DataSource = DataSource1). Свойство Options.dgTitles разрешает вывод строки заголовка столбцов, Options.dgIndicator разрешает вывод колонки индикатора во время работы с базой данных (треугольник – текущая запись, звездочка – новая запись).

4) Добавить к форме компонент DBNavigator, настроив его на работу с таблицей-источником данных (свойству DataSource нужно присвоить значение DataSource 1).

5) Откомпилируйте и запустите программу. Следует обратить внимание, что для того чтобы после запуска программы в окне появилась информация или, если база данных пустая, можно было вводить новую информацию, свойство Active таблицы-источника данных должно иметь значение True.

Используя клавиши перемещения курсора вверх и вниз, а также клавиши листания текста страницами (<Page Up> и <Page Down>), можно, перемещаясь от строки к строке, просматривать записи базы данных. Нажав клавишу <Ins>, можно добавить запись, а нажав клавишу — удалить запись. Для того чтобы внести изменения в поле записи, нужно, используя клавиши перемещения курсора влево и вправо, выбрать необходимое поле и нажать клавишу <F2>.

7) Сохраните созданную базу данных для дальнейшей работы

Лабораторная №3 Навигационная обработка данных в таблицах: просмотр и модификация.

Сортировка заключается в упорядочивании записей по определенному полю в порядке возрастания или убывания содержащихся в нем значений. Сортировка наборов данных Table и Query выполняется различными способами. Рассмотрим сортировку набора данных Table. Сортировка выполняется автоматически по текущему индексу. При смене индекса происходит автоматическое переупорядочивание записей. Таким образом, сортировка возможна по полям, для которых создан индекс. Задать индекс, по которому выполняется сортировка, можно с помощью свойств IndexName или IndexFieldNames. Эти свойства являются взаимоисключающими, и установка значения одного из них приводит к автоматической очистке значения другого. В качестве значения свойства IndexName указывается имя индекса, установленное при его создании. При использовании свойства IndexFieldNames указываются имена полей, образующих соответствующий индекс.

1) Откройте созданную на предыдущей лабораторной работе таблицу и измените ее.

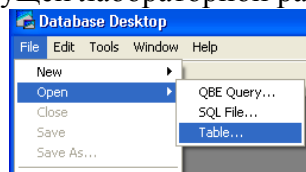


Рисунок 1

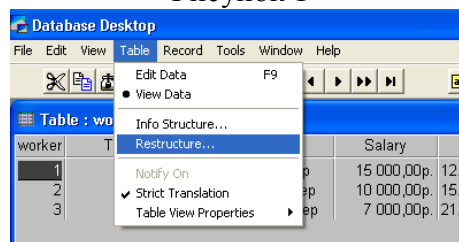


Рисунок 2

2) Задать индексы полей можно следующим образом:

- с помощью определения свойств таблицы (Properties Table) определите вторичные индексы (Secondary Indexes - Define).
- добавьте индексы по фамилии (indName), по дате рождения (indDate) и по отделу (indOtd).

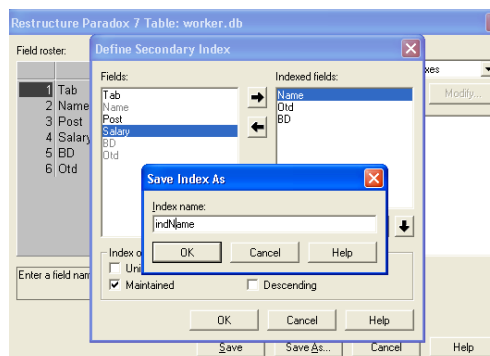


Рисунок 3

3) На форме расположите компонент RadioGroup, который представляет собой группу переключателей RadioButton. Такая группа переключателей создана для упорядочения переключателей и упрощения организации их взаимодействия по сравнению с добавлением их вручную к обычной группе.

Управление числом и названиями переключателей производится с помощью свойства Items типа TString, которое позволяет получить доступ к отдельным переключателям в группе. Это свойство содержит строки, отображаемые как заголовки переключателей. Отсчет строк в массиве переключателей начинается с нуля. Доступ к отдельному переключателю можно получить через свойство ItemIndex типа Integer, содержащее позицию переключателя, выбранного в группе в текущий момент. Это свойство используется для выбора отдельного переключателя или для определения, какой из переключателей является выбранным. Ниже на рисунке приведен внешний вид формы. Процедура, которая выполняется по нажатию кнопки.

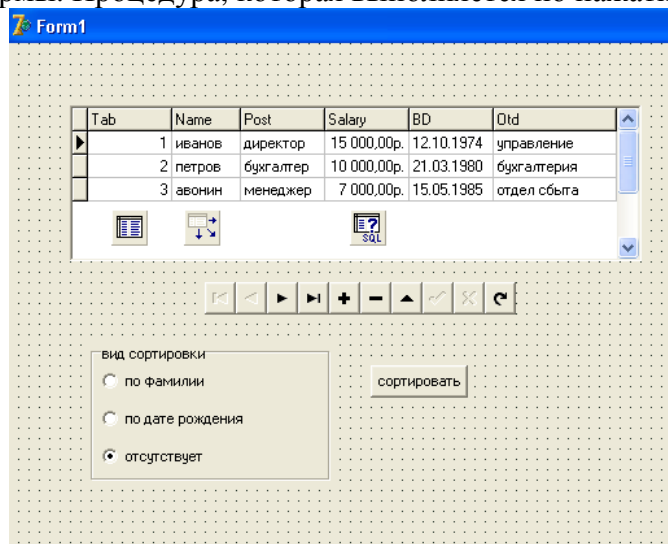


Рисунок 4

```

procedure TForm1.Button2Click(Sender: TObject);
begin
case RadioGroup1.ItemIndex of
  0: Table1.IndexName:= 'indName' ;
  1: Table1.IndexName:= 'indBD';
  2: Table1.IndexName:= '';
end;

```

Лабораторная №4 Фильтрация записей.

Фильтрация – это задание ограничений для записей, отбираемых в набор данных. Набор данных представляет собой записи, выбранные из одной или нескольких таблиц. Состав записей в наборе данных в данный момент времени зависит от установленных ограничений, в том числе от фильтров.

Система Delphi дает возможность осуществлять фильтрации записей:

- по выражению
- по диапазону

Достоинством фильтрации по выражению является то, что она применима к любым полям, в том числе к индексированным. В связи с тем, что в процессе отбора просматриваются все записи таблицы, фильтрация по выражению эффективна при небольшом количестве записей.

Для задания выражения фильтра используется свойство Filter типа String.

Параметры фильтрации задаются с помощью свойства FilterOptions типа TFilterOptions. Это свойство принадлежит к множественному типу и может принимать комбинации двух значений:

- foCaseIntensitive – регистр букв не учитывается.
- foNoPartialCompare – выполняется проверка на полное соответствие содержимого поля и значения, заданного для поиска.

По умолчанию все параметры фильтра выключены, и свойство FilterOptions имеет значение [].

Рассмотрим пример обработчика событий формы приложения, в которой пользователю предоставлена возможность управлять фильтрацией по двум полям, либо совсем отключить фильтрацию.

Пользователь может задавать условия фильтрации по полю Salary. Фильтрация записей происходит при нажатии кнопки Фильтровать.

Ниже приведены обработчики событий модуля формы Form1 приложения.

```
procedure TForm1.Button3Click(Sender: TObject);
begin
// фильтровать по окладу
if rbFilterSalary.Checked then Table1.Filter := 'Salary >' +
edtSalaryMin.Text + 'AND Salary <' + edtSalaryMax.Text;
// отключить фильтр
if rbNoFilter.Checked then Table1.Filter := '';
end;
```

Для ограничения значений оклада используются два однострочных редактора edtSalaryMin и edtSalaryMax

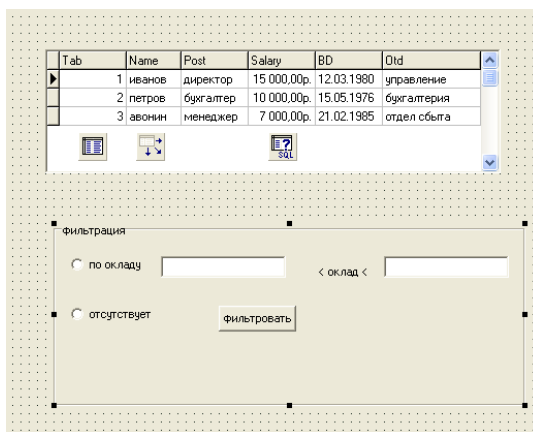


Рисунок 1

Лабораторная №5 Сортировка и поиск записей по разным критериям.

Поиск записи, удовлетворяющей определенным условиям. Означает переход на эту запись. Поиск во многом похож на фильтрацию, так как в процессе поиска также выполняется проверка полей записей по некоторым условиям. Отличие заключается в том, что в результате поиска количество записей в наборе данных не изменяется.

При организации поиска записей важное значение имеет наличие индекса для полей, по которым ведется поиск. Индексирование значительно повышает скорость обработки данных, кроме того. Ряд методов может работать только с индексированными полями.

Функция Locate (const KeyField:String; const KeyValues: Variant; Options: TLocateOptions): Boolean ищет запись с заданными значениями полей. Если удовлетворяющие условиям поиска записи существуют, то указатель текущей записи устанавливается на первую из них. Если запись найдена, то функция возвращает значение True.

Список полей, по которым ведется поиск, задается в параметре KeyFields. Параметр KeyValues типа Variant указывает значения полей для поиска. Параметр TLocateOptions по сути похож на тип TFilterOptions.

Рассмотрим в качестве примера обработчика событий формы приложения, используемой для последовательного поиска записей (см. рисунок).

Ниже приведен фрагмент кода программы.

```
procedure TForm1.edtSearchChange(Sender: TObject);
var strField: String;
begin
```

```

//если режим поиска выключен, то выйти из процедуры
if not cbSearch.Checked then exit;
//выбрать поле для поиска
case RadioGroup2.ItemIndex of
0: strField:= 'Name';
1: strField:= 'Otd';
end;
//выполнить поиск
Table1.Locate(strField,edtSearch.Text,[loCaseInsensitive,loPartialKey]);
end;

procedure TForm1.cbSearchClick(Sender: TObject);
begin
edtSearchChange(Sender);
end;
procedure TForm1.RadioGroup2Click(Sender: TObject);
begin
edtSearchChange(Sender);
end;

```

Здесь при установке флажка cbSearch включается режим поиска, и указатель текущей записи перемещается на запись, удовлетворяющую условиям поиска. Поле, по которому ведется поиска выбирается с помощью переключателей. Процесс поиска запускается каждый раз при изменении содержимого редактора edtSearch .

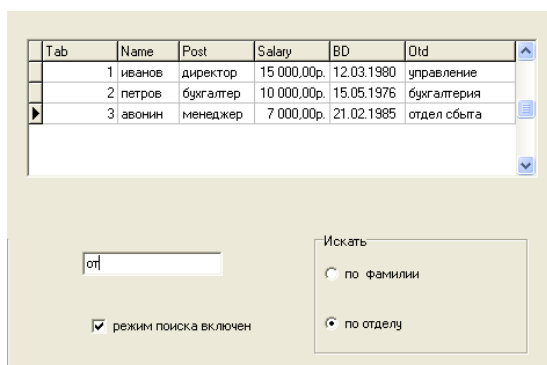


Рисунок 1

Лабораторная №6 Реализация ссылочной целостности БД. Лабораторная №7 Динамические запросы с параметрами.

Обычно визуальные компоненты предназначены для редактирования, добавления и удаления записей. Группируются на форме и работают взаимосвязано. Вместе с этими компонентами часто располагают элементы управления сортировкой, фильтрацией и поиском. Тем самым для пользователя обеспечивается выполнения различных операций с данными.

Для примера рассмотрим форму (см. рисунок 1), с помощью которой можно изменять записи таблицы, содержащей сведения о товарах. В информацию о товаре входят уникальный код записи, название товара, единица измерения. Цена единицы и примечания. Поле названия и поле товара обязательно быть заполненными, поле кода – автоинкрементное, и его значение при добавлении новой записи формируется автоматически.

Ниже приводится код модуля формы приложения.

```

unit Unit11;
interface
uses

```

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, Grids, DBGrids, DB, DBTables, ExtCtrls, DBCtrls, Mask;
type

```
TForm1 = class(TForm)
  btnEdit: TButton;
  btnInsert: TButton;
  btnDelete: TButton;
  cbCanEdit: TCheckBox;
  btnChangeOK: TButton;
  btnChangeCancel: TButton;
  btnClose: TButton;
  dbeName: TDBEdit;
  dbePrice: TDBEdit;
  dbeNote: TDBEdit;
  dbcbUnit: TDBComboBox;
  lblRecordCount: TLabel;
  lblChangeKind: TLabel;
  Label1: TLabel;
  Label2: TLabel;
  Label3: TLabel;
  Label4: TLabel;
  DataSource1: TDataSource;
  DBGrid1: TDBGrid;
  Table1: TTable;
  procedure StateBrowse(Sender:TObject);
  procedure StateChange(sender:TObject);
  procedure FormCreate(Sender: TObject);
  procedure cbCanEditClick(Sender: TObject);
  procedure Table1BeforeEdit(DataSet: TDataSet);
  procedure btnEditClick(Sender: TObject);
  procedure Table1BeforeInsert(DataSet: TDataSet);
  procedure btnInsertClick(Sender: TObject);
  procedure btnDeleteClick(Sender: TObject);
  procedure Table1BeforePost(DataSet: TDataSet);
  procedure btnChangeOKClick(Sender: TObject);
  procedure btnChangeCancelClick(Sender: TObject);
  procedure btnCloseClick(Sender: TObject);
  procedure lblChangeKindClick(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;
var
  Form1: TForm1;
implementation
  {$R *.dfm}
  // режим просмотра
  procedure TForm1.StateBrowse(Sender: TObject);
  begin
    lblRecordCount.Caption:= 'всего записей' + intToStr(Table1.RecordCount);
    cbCanEditClick(Sender);
```

```

btnChangeOK.Enabled:= False;
btnChangeCancel.Enabled:=False;
lblChangeKind.Caption:='просмотр записи' ;
end;
//режим изменения
procedure TForm1.StateChange(Sender:TObject);
begin
btnEdit.Enabled:=False;
btnInsert.Enabled:=False;
btnDelete.Enabled:=False;
btnChangeOK.Enabled:=True;
btnChangeCancel.Enabled:=True;
end;
procedure TForm1.FormCreate(Sender:TObject);
begin
// исходное состояние элементов управления
StateBrowse(Sender);
// первоначально изменение записей запрещено
cbCanEdit.Checked:=False;
// запрет автоматического перехода в режим редактирования
DataSource1.AutoEdit:=False;
// формирование списка единиц измерения товара
dbcbUnit.Items.Clear;
dbcbUnit.Items.Add('шт. ');
dbcbUnit.Items.Add('упак. ');
dbcbUnit.Items.Add('кор. ');
end;
procedure TForm1.cbCanEditClick(Sender: TObject);
var bm1:TBookmark;
begin
// запоминание положения текущей записи
bm1:=Table1.GetBookmark;
// отключение отображения изменений данных в визуальных компонентах
Table1.DisableControls;
if not cbCanEdit.Checked then begin
Table1.Active:=False;
Table1.ReadOnly:=True;
Table1.Active:=True;
// блокирование элементов, связанных с переходом
// в режим изменения записей
btnEdit.Enabled:=False;
btnInsert.Enabled:=False;
btnDelete.Enabled:=False;
end
else begin
Table1.Active:=False;
Table1.ReadOnly:=False;
Table1.Active:=True;
// разблокирование элементов, связанных с переходом
// в режимы изменения записей
btnEdit.Enabled:=True;
btnInsert.Enabled:=True;

```



```

// если набор данных пуст, то удаление записей запрещено
if Table1.RecordCount>0
then btnDelete.Enabled:=True
else btnDelete.Enabled:=False;
end;
// возврат к текущей записи
if Table1.BookmarkValid(bm1) then Table1.GotoBookmark(bm1);
if Table1.BookmarkValid(bm1) then Table1.FreeBookmark(bm1);
// включение отображения изменений данных в визуальных компонентах
Table1.EnableControls;
end;
procedure TForm1.Table1BeforeEdit(DataSet: TDataSet);
begin
// запрос на подтверждение перехода в режим редактирования
if MessageDlg ('будете редактировать?',mtConfirmation, [mbYes,mbNo],0)<>mrYes then Abort;
end;
// переход в режим редактирования
procedure TForm1.btnEditClick(Sender: TObject);
begin
Table1.Edit;
lblChangeKind.Caption:= 'редактирование записи';
stateChange(Sender);
if dbename.CanFocus then dbename.SetFocus;
end;
procedure TForm1.Table1BeforeInsert(DataSet: TDataSet);
begin
// подтверждение перехода в режим вставки
if messageDlg('добавить запись?',mtconfirmation,[mbYes,mbNo],0)<>mrYes then Abort;
end;
// переход в режим вставки
procedure TForm1.btnInsertClick(Sender: TObject);
begin
Table1.Insert;
lblChangeKind.Caption:='вставка записи';
StateChange(Sender);
if dbename.CanFocus then dbename.SetFocus;
end;
// переход в режим просмотра удаляемой записи
procedure TForm1.btnDeleteClick(Sender: TObject);
begin
// подтверждение перехода в режим просмотра удаляемой записи
if MessageDlg ('удалить запись',mtConfirmation, [mbYes,mbNo],0)<>mrYes then Exit;
lblchangeKind.Caption:='удаление записи';
StateChange(Sender);
if btnChangeCancel.CanFocus then btnChangeCancel.SetFocus;
end;
procedure TForm1.Table1BeforePost(DataSet: TDataSet);
begin
// проверка, задано ли значение для названия товара
if dbename.Text=' ' then begin
Beep;
MessageDlg('не задано название товара',mtError,[mbOK],0);

```

```

if dbeName.CanFocus then dbeName.SetFocus;
Abort;
end;
// проверка, задано ли значение цены товара
if dbePrice.Text=' ' then begin
Beep;
messageDlg('не задана цена товара',mtError,[mbOK],0);
if dbePrice.CanFocus then dbePrice.SetFocus;
Abort;
end;
end;
procedure TForm1.btnChangeOKClick(Sender: TObject);
begin
// утверждение изменений в текущей записи
// или удаление текущей записи
if Table1.State in [dsEdit,dsInsert]
then Table1.Post
else if lblChangeKind.Caption='удаление записи' then Table1.Delete;
StateBrowse(Sender);
end;
procedure TForm1.btnChangeCancelClick(Sender: TObject);
begin
// если набор данных находился в режиме просмотра,
// то никаких действий метод не выполняет
Table1.Cancel;
StateBrowse(Sender);
end;
//закрытие формы
procedure TForm1.btnCloseClick(Sender: TObject);
begin
close;
end;
end.

```

В качестве набора данных используется компонент Table1, записи которого отображает сетка DBGrid1. Значения полей текущей записи отображаются в компонентах dbeName, dbePrice, dbeNote, cbCanEdit. Установка свойств, определяющих взаимные отношения компонентов, связанных с набором данных, выполнена при разработке приложения через Инспектор объектов.

Пользователь не может переводить набор данных в режимы изменения с помощью визуальных компонентов, так как свойство AutoEdit источника данных DataSource1 установлено в значение False.

Для перевода набора данных в режимы редактирования и вставки служат кнопки Изменить (btnEdit) и Добавить (btnInsert). Переход в эти режимы осуществляется вызовом методов Edit и Insert, после чего название режима отображается в надписи lblChangeKind, блокируются кнопки, связанные с переходом в режимы изменения, и разблокируются кнопки btnChangeOK и btnChangeCancel, позволяющие принять или отменить изменения, сделанные при редактировании или вставке записи. Для подтверждения или отмены изменений вызываются методы Post или Cancel, после чего кнопки подтверждения снова блокируются, а кнопки перехода в режимы изменения разблокируются. Изменение и задание значений полей выполняются с помощью компонентов dbeName, dbePrice, dbeNote, cbCanEdit.

Флажок cbCanEdit с заголовком Редактирование разрешено включает и отключает возможность изменения в наборе данных. Управление этой возможностью осуществляется через свойство ReadOnly набора данных. Свойство переключается только при закрытом наборе данных.

При разрешении изменений в наборе данных проверяется число его записей, и если набор данных пуст, то кнопка btnDelete блокируется.

Рисунок 1

Лабораторная №8 Организация меню в приложении: главное и контекстное.

В качестве примера работы со связанными таблицами рассмотрим приложение, предназначенное для автоматизации складского учета.

При организации складского учета используются две таблицы формата Paradox: Store для хранения информации о товарах и Cards для хранения карточек товара, в которой отмечается движение (приход-расход) каждого товара. Структура таблиц приведена на рисунке 1.

Field roster:				
	Field Name	Type	Size	Key
1	S_Code	+		*
2	S_Name	A	20	
3	S_Unit	A	7	
4	S_Price	\$		
5	S_Quality	N		
6	S_Note	A	30	

Field roster:				
	Field Name	Type	Size	Key
1	C_Number	+		*
2	C_Code	I		
3	C_Move	N		
4	C_Date	D		

Рисунок 1.

Между таблицами устанавливается связь «главный - подчиненный», при которой таблица Store склада является главной, а таблица Cards движение товара – подчиненной. Для организации связи в качестве поля связи главной таблицы берется автоинкрементное поле S_Code уникального кода товара. По этому полю построен ключ, значение которого автоматически формируется при добавлении новой записи и в пределах таблицы является уникальным. В подчиненной таблице полем связи является целочисленное поле C_Code, по которому построен индекс.

Приложение для работы со складом включает главную форму (рисунок 2) и форму ввода данных о новом товаре (рисунок 3)

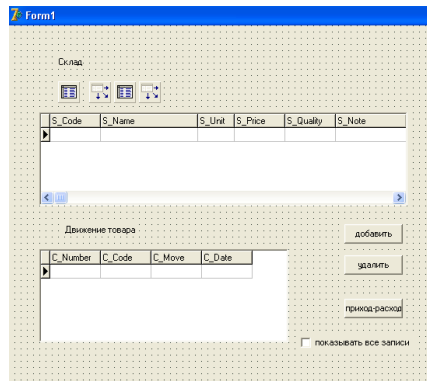


Рисунок 2.

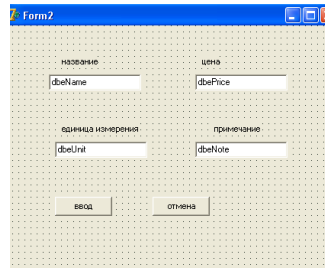


Рисунок 3.

В верхней части главной формы выводится информация о состоянии склада, в нижней части – сведения о движении товара. При выборе в таблице склада записи о товаре в таблице движения товара автоматически отображаются только записи, соответствующие движению именно этого товара. Для наглядности в наборы данных включены все поля таблиц, которые отображаются в компонентах DBGrid. При этом названия заголовков столбцов совпадают с названиями полей.

Модификация данных таблиц с помощью компонента DBGrid запрещена, для этого их свойства AutoEdit установлены в значение False. Для модификации таблиц используются кнопки. Связь между полями таблицы и элементами формы 2 устанавливается через источник данных dsStore, расположенный на главной форме. Чтобы такая связь стала возможной, в модулях формы имеются ссылки для на друга.

Ниже приведены коды модулей форм приложения. Установка свойств большинства компонентов выполнена в обработчиках событий создания форм приложения.

//модуль главной формы

```
unit Unit12;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, Grids, DBGrids, DB, DBTables, StdCtrls;
```

```
type
```

```
TForm1 = class(TForm)
```

```
TableStore: TTable;
```

```
TableCard: TTable;
```

```
dsStore: TDataSource;
```

```
dsCard: TDataSource;
```

```
DBGridStore: TDBGrid;
```

```
DBGridCard: TDBGrid;
```

```
Label1: TLabel;
```

```
Label2: TLabel;
```

```
btnNew: TButton;
```

```

btnDelete: TButton;
btnMove: TButton;
cbMoveAll: TCheckBox;
procedure FormCreate(Sender: TObject);
procedure btnNewClick(Sender: TObject);
procedure btnDeleteClick(Sender: TObject);
procedure btnMoveClick(Sender: TObject);
procedure cbMoveAllClick(Sender: TObject);
private
{ Private declarations }
public
{ Public declarations }
end;
var
Form1: TForm1;
implementation
uses unit2;
{$R *.dfm}
procedure TForm1.FormCreate(Sender: TObject);
begin
dsStore.AutoEdit:=False;
dsCard.AutoEdit:=False;
Tablecard.MasterSource:= dsStore;
cbMoveAll.Checked:=False;
cbMoveAllClick(Sender);
end;
procedure TForm1.btnNewClick(Sender: TObject);
begin
TableStore.Append;
if Form2.ShowModal = mrOK
then begin
TableStore.FieldByName('S_Quality').AsFloat:=0;
TableStore.Post;
end
else TableStore.Cancel;
end;
procedure TForm1.btnDeleteClick(Sender: TObject);
var n: longint;
begin
if TableStore.RecordCount=0 then exit;
if MessageDlg('удалить запись?',mtConfirmation,[mbOK,mbNo],0)= mrOK
then begin
// удаление записей в карточке движения товара
TableCard.Last;
for n:= 1 to TableCard.RecordCount do TableCard.Delete;
// удаление карточки движения товара
TableStore.Delete;
end;
end;
procedure TForm1.btnMoveClick(Sender: TObject);
var sMove:string;
nmove:double;

```

```

begin
if InputQuery ('поступление товара' + TableStore.FieldName('S_Name').AsString,'приход-
расход',smove)
then begin
// проверка введенного приходного или расходного кол-ва товара
try
nmove:=StrToFloat(smove);
except;
beep;
messageDlg('неправильно введен приход-расход'+smove,mterror,[mbOK],0);
exit;
end;
// добавление новой записи в карточку движения товара
TableCard.Append;
// поле C_Code заполняется автоматически S_Code
TableCard.FieldName('c_Move').AsFloat:=nmove;
TableCard.FieldName('c_Date').AsDateTime:=Now;
TableCard.Post;
// пересчет наличного количества товара
TableStore.Edit;
TableStore.FieldName('S_Quality').AsFloat:=
TableStore.FieldName('S_Quality').AsFloat+nmove;
TableStore.Post;
end;
end;
procedure TForm1.cbMoveAllClick(Sender: TObject);
begin
if not cbMoveAll.Checked then begin
TableCard.IndexName:= 'indC_Code';
Tablecard.MasterFields:= 'S_Code';
btnDelete.Enabled:=True;
end
else begin
TableCard.IndexName:=' ';
TableCard.IndexFieldNames:=' ';
Tablecard.MasterFields:=' ';
btnDelete.Enabled:=False;
end;
end;
end.
// модуль формы ввода
unit Unit2;
interface
uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, Mask, DBCtrls;
type
TForm2 = class(TForm)
dbeName: TDBEdit;
dbeUnit: TDBEdit;
dbePrice: TDBEdit;
dbeNote: TDBEdit;

```

```

Label1: TLabel;
Label2: TLabel;
Label3: TLabel;
Label4: TLabel;
btnOK: TButton;
btnCancel: TButton;
procedure FormCreate(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;
var
  Form2: TForm2;
implementation
  uses unit12;
  {$R *.dfm}
procedure TForm2.FormCreate(Sender: TObject);
begin
  dbeName.DataSource:=form1.dsStore;
  dbeName.DataField:='S_name';
  dbeunit.DataSource:= Form1.dsStore;
  dbeUnit.DataField:='S_Unit';
  dbePrice.DataSource:=form1.dsStore;
  dbePrice.DataField:='s_Price';
  dbeNote.DataSource:=Form1.dsStore;
  dbeNote.DataField:='S_Note';
  btnOK.ModalResult:=mrOK;
  btnCancel.ModalResult:=mrCancel;
end;
end.

```

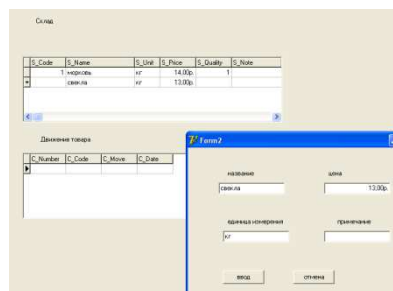
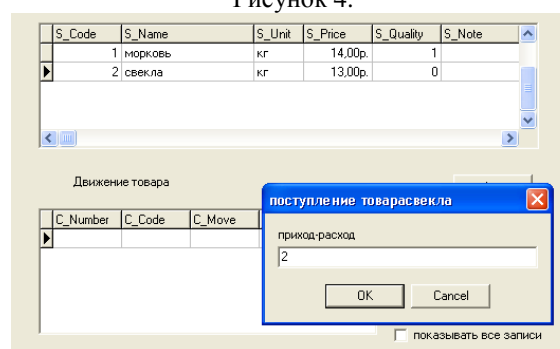


Рисунок 4.



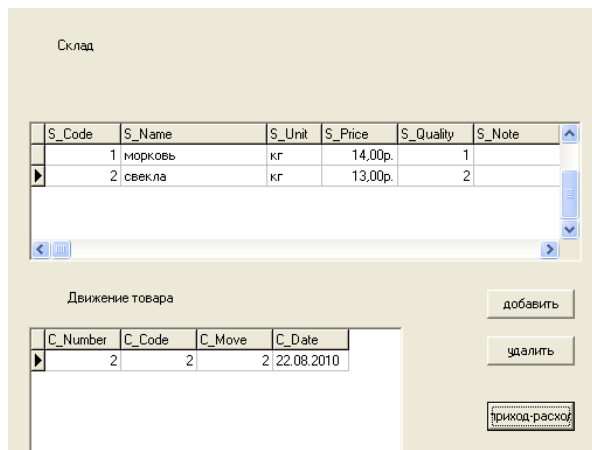


Рисунок 5.

В рассмотренном примере связь между таблицами устанавливается уже при выполнении приложения. Обычно же таблицы связываются на этапе разработки через Инспектор Объектов. При этом для установки свойств IndexName и MasterFields удобно использовать специальный редактор полей связи (Field Link Designer), вызываемый двойным щелчком в области значений свойства MasterFields в окне Инспектора Объектов. В списке Available Indexes выбирается индекс подчиненной таблицы, после чего составляющие его поля отображаются в списке Detail Fields. В этом списке необходимо выбрать поле подчиненной таблицы, а в списке Master Fields – поле главной таблицы. После нажатия кнопки Add выбранные поля связываются между собой, что отображается в списке Joined Fields. При этом оба поля пропадают из своих списков.

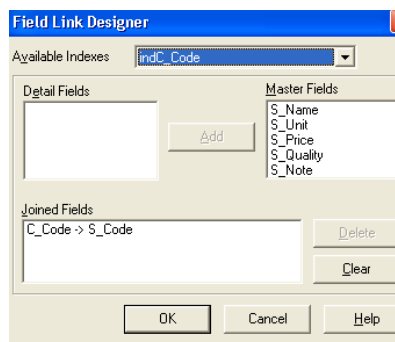


Рисунок 6.

Лабораторная №9 Динамические запросы на модификацию.

Пользователь управляет сортировкой с помощью групп переключателей.

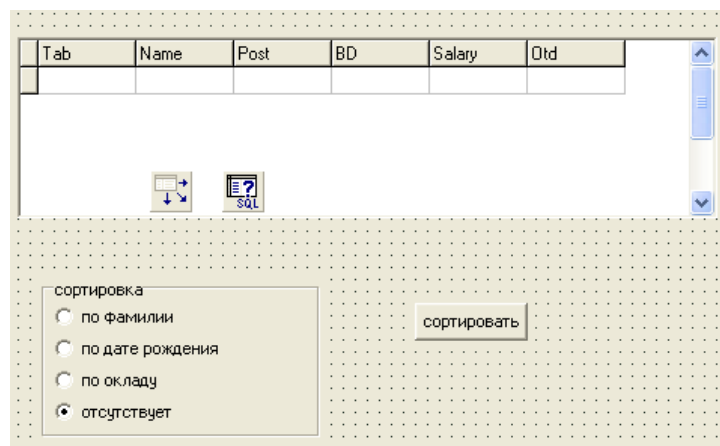


Рисунок 1.

Ниже приводится код обработчика события нажатия кнопки «сортировка»


```

procedure TForm1.btnSortClick(Sender: TObject);
var s:string;
begin
Query1.Close;
Query1.SQL.Clear;
Query1.SQL.Add('SELECT *From worker.DB ');
case RadioGroup1.ItemIndex of
0: s:='ORDER BY Name';
1: s:='ORDER BY BD';
2: s:='ORDER BY Post';
3: s:='';
end;
Query1.SQL.Add(s);
Query1.Open;
end;

```

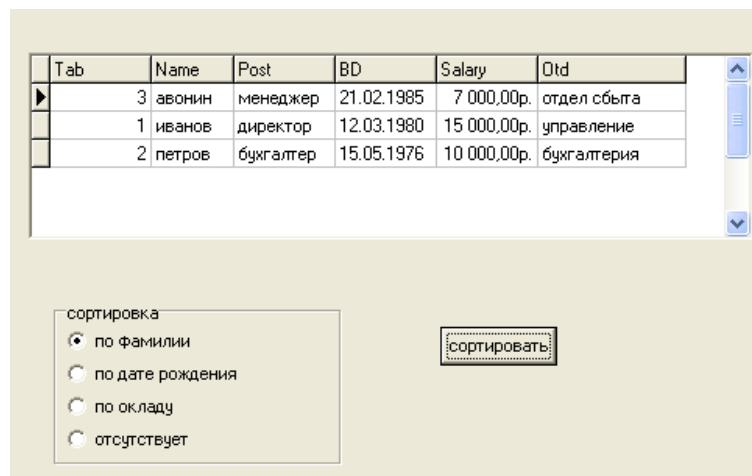


Рисунок 2.

Список полей, по которым выполняется сортировка, указывается в операторе ORDER BY. Поля в списке обозначаются именами или номерами, которые соответствуют номерам в списке полей после слова SELECT. По умолчанию сортировка происходит в порядке возрастания значений полей. Для указания обратного порядка сортировки по какому-либо полю нужно указать после имени этого поля DESC.

Задание: предусмотреть возможность сортировки по возрастанию и убыванию.

Рекомендуемая литература

Основная литература

1. Гущин, А.Н. Базы данных : учебник / А.Н. Гущин. - М. : Директ-Медиа, 2014. - 266 с. : ил.,табл., схем. - ISBN 978-5-4458-5147-9 ; То же [Электронный ресурс]. - URL: [//biblioclub.ru/index.php?page=book&id=222149](http://biblioclub.ru/index.php?page=book&id=222149) , коэффициент книгообеспеченности 0,5
2. Хомоненко, А. Д. Delphi 7 [Текст] / А. Д. Хомоненко, В. Э. Гофман, Е. В. Мещеряков.- 2-е изд., [перраб. и доп.]. - Санкт-Петербург : БВХ-Петербург, 2013. - 1136 с. : ил. + 1 электрон. опт. диск (CD-ROM). - ([В подлиннике]). - Предм. указ. : с. 1109. - аб.ТБ-10

Дополнительная литература

1. Щелоков, С.А. Базы данных : учебное пособие / С.А. Щелоков ; Министерство образования и науки Российской Федерации, Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Оренбургский государственный университет», Кафедра программного обеспечения вычислительной техники и

автоматизированных систем. - Оренбург : Оренбургский государственный университет, 2014. - 298 с. : ил. - Библиогр. в кн. ; То же [Электронный ресурс]. - URL: [//biblioclub.ru/index.php?page=book&id=260752](http://biblioclub.ru/index.php?page=book&id=260752), коэффициент книгообеспеченности 1.

2. Абросимова, М.А. Базы данных: проектирование и создание программного приложения в СУБД MS Access : практикум / М.А. Абросимова ; Министерство образования и науки Российской Федерации, Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Уфимский государственный университет экономики и сервиса», Кафедра «Информатика и ИКТ». - Уфа : Уфимский государственный университет экономики и сервиса, 2014. - 56 с. - Библиогр. в кн. ; То же [Электронный ресурс]. - URL: [//biblioclub.ru/index.php?page=book&id=272367](http://biblioclub.ru/index.php?page=book&id=272367), коэффициент книгообеспеченности 1.

3. Медведкова, И.Е. Базы данных / И.Е. Медведкова, Ю.В. Бугаев, С.В. Чикунов ; Министерство образования и науки РФ, ФГБОУ ВПО «Воронежский государственный университет инженерных технологий» ; науч. ред. Г.В. Абрамов. - Воронеж : Воронежский государственный университет инженерных технологий, 2014. - 105 с. : ил. - Библиогр. в кн. - ISBN 978-5-00032-060-0 ; То же [Электронный ресурс]. - URL: [//biblioclub.ru/index.php?page=book&id=336039](http://biblioclub.ru/index.php?page=book&id=336039), коэффициент книгообеспеченности 1.

4. Гущин, А.Н. Базы данных : учебно-методическое пособие / А.Н. Гущин. - 2-е изд., испр. и доп. - М. ; Берлин : Директ-Медиа, 2015. - 311 с. : ил. - Библиогр.: с. 226-228. - ISBN 978-5-4475-3838-5 ; То же [Электронный ресурс]. - URL: [//biblioclub.ru/index.php?page=book&id=278093](http://biblioclub.ru/index.php?page=book&id=278093), коэффициент книгообеспеченности 1.

Периодические издания

1. Журнал «Вестник компьютерных и информационных технологий»
2. Журнал «Информационные технологии и вычислительные системы»
3. Журнал «Стандарты и качество»
4. Журнал «Прикладная информатика»