

Министерство образования и науки Российской Федерации
Федеральное агентство по образованию
Ярославский государственный университет им. П. Г. Демидова
Кафедра компьютерной безопасности и математических методов
обработки информации

О. В. Власова

Системы управления базами данных

Лабораторный практикум

*Рекомендовано
Научно-методическим советом университета для студентов,
обучающихся по специальностям Прикладная математика
и информатика, Компьютерная безопасность*

Ярославль 2010

УДК 004.65
ББК 3 973.233–018.2
В 58

*Рекомендовано
Редакционно-издательским советом университета
в качестве учебного издания. План 2009/10 года*

Рецензент – кафедра компьютерной безопасности
и математических методов обработки информации

Власова, О. В. Системы управления базами данных: лабораторный практикум / О. В. Власова; Яросл. гос. ун-т им. П. Г. Демидова. – Ярославль : ЯрГУ, 2010. – 76 с.

Лабораторный практикум посвящен важнейшей составляющей широко разрабатываемых и используемых информационных систем организационного управления – базам данных (БД), создаваемым и функционирующим на основе систем управления базами данных (СУБД).

Содержит концептуальные представления об основных принципах построения БД и СУБД, принципах проектирования БД, а также анализ основных технологий реализации БД. Особое внимание уделяется представлению фундаментальных понятий и математических моделей, лежащих в основе реляционных БД и СУБД.

Предназначено для студентов, обучающихся по специальностям 010501.65 Прикладная математика и информатика, 090102.65 Компьютерная безопасность (дисциплина «Базы данных и экспертные системы, СУБД», блок ОПД), очной формы обучения.

УДК 004.65
ББК 3 973.233–018.2

© Ярославский государственный университет им. П. Г. Демидова, 2010

Учебное издание

Власова Ольга Владимировна
Системы управления базами данных

Лабораторный практикум

Редактор, корректор И. В. Бунакова

Верстка И. Н. Иванова

Подписано в печать 29.04.10. Формат 60×84 ¹/₁₆. Бум. офсетная.
Гарнитура «Times NewRoman». Усл. печ. л. 4,42. Уч.-изд. л. 3,09.

Тираж 100 экз. Заказ

Оригинал-макет подготовлен в редакционно-издательском отделе Ярославского государственного университета им. П. Г. Демидова.

Отпечатано на ризографе.

Ярославский государственный университет им. П. Г. Демидова.
150000, Ярославль, ул. Советская, 14.

Предисловие

Последние десятилетия в области программирования характеризуются резким ростом количества создаваемых информационных систем организационного управления. Практически в каждой организации функционирует (или создается) такая система (или её элементы). Важнейшей структурной частью информационных систем являются БД, создаваемые и функционирующие на основе использования специализированных программных систем – СУБД. Все это обуславливает большую потребность в квалифицированных кадрах, способных как создавать информационные системы на основе СУБД, так и обслуживать соответствующие информационные системы и БД.

Цель данного учебного пособия состоит в формировании концептуальных представлений об основных принципах построения БД, СУБД; о математических моделях, описывающих БД; о принципах проектирования БД.

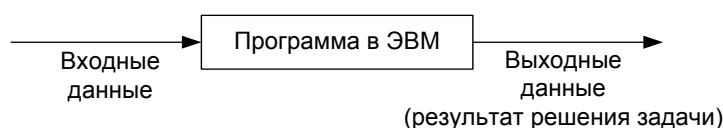
Предварительные знания

Курс «Базы данных» опирается на материалы следующих курсов: «Основы построения ЭВМ»; «ЭВМ и программирование»; «Дискретная математика».

Теоретические сведения

1.1. Развитие основных понятий представления данных

Любой вычислительный процесс представляет собой отображение (по определенному алгоритму) некоторых входных данных в выходные.



Соотношение сложности представления обрабатываемых данных и алгоритма вычислений определяет два класса задач:

- вычислительные задачи – достаточно простое представление данных и сложный, многооперационный процесс вычислений;
- задачи обработки данных (невыхислительные задачи) – простой алгоритм обработки данных и сложное представление обрабатываемых данных.

На начальной стадии обучения программированию основное внимание уделяется разработке алгоритма решения задачи. Однако часто оказывается, что возможность (или невозможность) решения конкретной задачи зависит не только от выбранного алгоритма, но и от того, какие понятия используются для представления обрабатываемых данных.

Рассмотрим простейший пример нахождения корней квадратного уравнения:

$$A * X^2 + B * X + C = 0,$$

где X , A , B и C – числа, которые являются здесь элементарными единицами данных (элементами данных).

При программировании алгоритма решения этой задачи используется простейший вид данных – простая переменная. Заметим, что каждая простая переменная характеризуется определенным типом значений, который должен выбираться при программировании. Даже в этом простейшем случае необходимо правильно выбрать тип переменной, причем от этого выбора

может зависеть возможность или невозможность решения конкретной задачи.

Рассмотрим другой пример:

$$S = a_1 + a_2 + \dots + a_N.$$

Решение этой задачи в общем случае невозможно получить, используя только простые переменные. Здесь обрабатывается не отдельное число, а последовательность чисел. В этом случае при программировании используется такой вид данных, как массив – совокупность элементов, с каждым из которых связан упорядоченный набор целых чисел, называемых индексами. Все элементы должны иметь одинаковый тип значений, который и будет типом массива. В этом случае числа a_1, a_2, \dots, a_N представляются в программе массивом $A[1], A[2], \dots, A[N]$.

Приведенные примеры показывают, что изменение вида задач обуславливает необходимость использования других типов данных.

Ранние языки программирования (ФОРТРАН, АЛГОЛ-60) были предназначены для решения научно-технических вычислительных задач. В этих языках использовались только вышеуказанные типы данных (простые переменные и массивы), что было вполне достаточно.

Начиная с конца 1960-х годов компьютеры начинают интенсивно использоваться для решения так называемых невычислительных задач, связанных с обработкой различного рода документов. Рассмотрим появление новых типов данных на примере упрощенных задач обработки данных.

Задача 1. Оплата услуг телефонной компании.

Рассматриваем задачу при двух упрощающих предположениях:

- клиент вносит абонентскую плату за K минут разговора;
- клиенту начисляется оплата в зависимости от количества потраченных минут.

Необходимые для решения этой задачи сведения о клиенте представлены в следующей карточке НАЧИСЛЕНИЕ: