

Лекция №1

Базы данных и файловые системы

Стуколов Сергей Владимирович
кафедра Юнеско по НИТ

просьба старостам отметить
отсутствующих

Цели и задачи курса

- Цели и задачи курса – ознакомить студентов с элементами реляционных баз данных (БД), системами управления БД и вопросами проектирования БД, привить навыки создания БД и манипулирования данными средствами ORACLE.

Лекционный курс

- В лекционном курсе излагаются основные понятия и методы организации реляционных баз данных и манипулирования ими, а также описываются базовые подходы к проектированию реляционных баз данных. Вводится понятие реляционной модели данных, обсуждаются структурная, манипуляционная и целостная составляющие модели. Обсуждаются важные аспекты теории баз данных, связанные с функциональными зависимостями. Описываются процесс проектирования реляционных баз данных на основе принципов нормализации, а также подходы к проектированию реляционных баз данных с использованием диаграммных семантических моделей данных.

Лабораторный практикум

- На лабораторных занятиях, проходящих в компьютерных классах, студенты изучают операторы языка SQL, выполняют семестровое занятие по созданию собственной БД. В результате успешного прохождения курса студент приобретает навыки создания и сопровождения базы данных ORACLE. Курс рассчитан на один семестр. В конце семестра проводится экзамен. Проверка знаний осуществляется компьютерным тестированием, причем вопросы тестирования содержат как теоретический, так и практический материал.

Семестровая работа

- Семестровая работа направлена на самостоятельную разработку семантической и физической модели базы данных на заданную задачу, реализация построенных моделей в виде набора SQL-скриптов. Требования к моделям: следование стандарту при построении ER-диаграммы; поддержка целостности данных. Студент должен без затруднений добавлять, обновлять, удалять записи в БД, а также производить выборки данных.
- Семестровые задания даются каждому студенту индивидуально преподавателем, ведущим лабораторные занятия.

Учебники, учебные пособия и монографии

- Мейер М. Теория реляционных баз данных. М.: Мир, 1987. - 608 с.
- Дейт К. Введение в системы баз данных. М.: Наука, 1980
- Озкарахан Э. Машины баз данных и управление базами данных. - М.: Мир, 1989
- Ульман Д. Основы систем баз данных. - М.: Финансы и статистика, 1983. - 334 с.
- Хансен Г., Хансен Д. Базы данных: разработка и управление. Пер. с англ. – М.: БИНОМ, 2000. 704с.
- Грабер М. SQL. Справочное руководство. – М.: Лори, 2001. 643с.
- [Гудов А.М., Шмакова Л.Е. Введение в язык структурированных запросов SQL / Учебное пособие. – Кемерово, Кемеровский госуниверситет, 2001.- 118с.](#)
- Кириллов В.В. Структуризованный язык запросов (SQL). - СПб.: ИТМО, 1994. - 80 с.
- Кириллов В.В. Основы проектирования реляционных баз данных. Учебное пособие. - СПб.: ИТМО, 1994. - 90 с.

Электронные учебные пособия

- 1. Гудов А.М. Введение в язык структурированных запросов SQL
<http://oldunesco.kemsu.ru/metod/sql/index.html>
- 2. Зеленков Ю.А. Введение в базы данных.
<http://www.mstu.edu.ru/education/materials/zelenkov/toc.html>
- 3. Пушников А.Ю. Введение в системы управления базами данных
<http://citforum.ru/database/dblearn/index.shtml>
- 4. Кузнецов С.Д. Основы современных баз данных
<http://citforum.ru/database/osbd/contents.shtml>
- 5. Кузнецов С.Д. Введение в реляционные базы данных
<http://www.intuit.ru/department/database/rdbintro/>
- 6. Полякова Л.Н. Основы SQL
<http://www.intuit.ru/department/database/sql/>

Экзамен

- Проверка знаний студентов по курсу “Базы данных” основана на тестирующей программе (<http://tests.kemsu.ru>). Данная программа позволяет разбить общее количество вопросов на подгруппы по темам, а затем, в процессе тестирования случайным образом выбирает вопросы из каждой темы, пропорционально подготовленным вопросам по этой теме. Каждый тест представлен 20 вопросами. После прохождения тестирования проводится собеседование с каждым студентом с разбором результатов тестирования. Итоговая оценка выставляется исходя из: результатов беседы со студентом, рекомендации преподавателя лабораторных занятий, оценки выполнения семестровой работы, качества и обоснованности ответов на дополнительные вопросы.

Вопросы?

- Итак – первая лекция посвящена рассмотрению общего смысла понятий БД и СУБД, файловым системам.

УРА!

ПЕРЕМЕНА!

Два основных направления использования ВТ

- Первое направление - применение вычислительной техники для выполнения численных расчетов, которые слишком долго или вообще невозможно производить вручную. Становление этого направления способствовало интенсификации методов численного решения сложных математических задач, развитию класса языков программирования, ориентированных на удобную запись численных алгоритмов, становлению обратной связи с разработчиками новых архитектур ЭВМ.

Два основных направления использования ВТ

- Второе направление, которое непосредственно касается темы нашего курса, это использование средств вычислительной техники в автоматических или автоматизированных информационных системах.
- В самом широком смысле информационная система представляет собой программный комплекс, функции которого состоят в поддержке надежного хранения информации в памяти компьютера, выполнении специфических для данного приложения преобразований информации и/или вычислений, предоставлении пользователям удобного и легко осваиваемого интерфейса.
- Обычно объемы информации, с которыми приходится иметь дело таким системам, достаточно велики, а сама информация имеет достаточно сложную структуру. Классическими примерами информационных систем являются банковские системы, системы резервирования авиационных или железнодорожных билетов, мест в гостиницах и т.д.

Два основных направления использования ВТ

На самом деле, второе направление возникло несколько позже первого. Это связано с тем, что на заре вычислительной техники компьютеры обладали ограниченными возможностями в части памяти. Понятно, что можно говорить о надежном и долговременном хранении информации только при наличии запоминающих устройств, сохраняющих информацию после выключения электрического питания. Оперативная память этим свойством обычно не обладает. В начале использовались два вида устройств внешней памяти: магнитные ленты и барабаны. При этом емкость магнитных лент была достаточно велика, но по своей физической природе они обеспечивали последовательный доступ к данным. Магнитные же барабаны (они больше всего похожи на современные магнитные диски с фиксированными головками) давали возможность произвольного доступа к данными, но были ограниченного размера.

Два основных направления использования ВТ

- Легко видеть, что указанные ограничения не очень существенны для чисто численных расчетов. Даже если программа должна обработать (или произвести) большой объем информации, при программировании можно продумать расположение этой информации во внешней памяти, чтобы программа работала как можно быстрее.
- С другой стороны, для информационных систем, в которых потребность в текущих данных определяется пользователем, наличие только магнитных лент и барабанов неудовлетворительно. Представьте себе покупателя билета, который стоя у кассы должен дожидаться полной перемотки магнитной ленты. Одним из естественных требований к таким системам является средняя быстрота выполнения операций.
- Как кажется, именно требования к вычислительной технике со стороны нечисленных приложений вызвали появление съемных магнитных дисков с подвижными головками, что явилось революцией в истории вычислительной техники. Эти устройства внешней памяти обладали существенно большей емкостью, чем магнитные барабаны, обеспечивали удовлетворительную скорость доступа к данным в режиме произвольной выборки, а возможность смены дискового пакета на устройстве позволяла иметь практически неограниченный архив данных.

Два основных направления использования ВТ

С появлением магнитных дисков началась история систем управления данными во внешней памяти. До этого каждая прикладная программа, которой требовалось хранить данные во внешней памяти, сама определяла расположение каждой порции данных на магнитной ленте или барабане и выполняла обмены между оперативной и внешней памятью с помощью программно-аппаратных средств низкого уровня (машинных команд или вызовов соответствующих программ операционной системы). Такой режим работы не позволяет или очень затрудняет поддержание на одном внешнем носителе нескольких архивов долговременно хранимой информации. Кроме того, каждой прикладной программе приходилось решать проблемы именования частей данных и структуризации данных во внешней памяти.

Файловые системы

- Историческим шагом явился переход к использованию централизованных систем управления файлами. С точки зрения прикладной программы файл - это именованная область внешней памяти, в которую можно записывать и из которой можно считывать данные. Правила именования файлов, способ доступа к данным, хранящимся в файле, и структура этих данных зависят от конкретной системы управления файлами и, возможно, от типа файла. Система управления файлами берет на себя распределение внешней памяти, отображение имен файлов в соответствующие адреса во внешней памяти и обеспечение доступа к данным.

Структуры файлов

- Практически во всех современных компьютерах основными устройствами внешней памяти являются магнитные диски с подвижными головками, и именно они служат для хранения файлов. Такие магнитные диски представляют собой пакеты магнитных пластин (поверхностей), между которыми на одном рычаге двигается пакет магнитных головок. Шаг движения пакета головок является дискретным, и каждому положению пакета головок логически соответствует цилиндр магнитного диска. На каждой поверхности цилиндр "высекает" дорожку, так что каждая поверхность содержит число дорожек, равное числу цилиндров.
- При разметке магнитного диска каждая дорожка размечается на одно и то же количество блоков таким образом, что в каждый блок можно записать по максимуму одно и то же число байтов. Таким образом, для произведения обмена с магнитным диском на уровне аппаратуры нужно указать номер цилиндра, номер поверхности, номер блока на соответствующей дорожке и число байтов, которое нужно записать или прочитать от начала этого блока.

Именованние файлов

- Все современные файловые системы поддерживают многоуровневое именованние файлов за счет поддержания во внешней памяти дополнительных файлов со специальной структурой - каталогов. Каждый каталог содержит имена каталогов и/или файлов, содержащихся в данном каталоге. Таким образом, полное имя файла состоит из списка имен каталогов плюс имя файла в каталоге, непосредственно содержащем данный файл. Разница между способами именованния файлов в разных файловых системах состоит в том, с чего начинается эта цепочка имен.

Именованние файлов

- В этом отношении имеются два крайних варианта. Во многих системах управления файлами требуется, чтобы каждый архив файлов (полное дерево справочников) целиком располагался на одном дисковом пакете (или логическом диске, разделе физического дискового пакета, представляемом с помощью средств операционной системы как отдельный диск). В этом случае полное имя файла начинается с имени дискового устройства, на котором установлен соответствующий диск.
- Можно назвать эту организацию поддержанием изолированных файловых систем.

Именованние файлов

- Другой крайний вариант был реализован в файловых системах операционной системы Multics. Эта система заслуживает отдельного большого разговора, в ней был реализован целый ряд оригинальных идей, но мы остановимся только на особенностях организации архива файлов. В файловой системе Multics пользователи представляли всю совокупность каталогов и файлов как единое дерево. Полное имя файла начиналось с имени корневого каталога, и пользователь не обязан был заботиться об установке на дисковое устройство каких-либо конкретных дисков. Сама система, выполняя поиск файла по его имени, запрашивала установку необходимых дисков.
- Такую файловую систему можно назвать полностью централизованной.

Защита файлов

- Поскольку файловые системы являются общим хранилищем файлов, принадлежащих, вообще говоря, разным пользователям, системы управления файлами должны обеспечивать авторизацию доступа к файлам. В общем виде подход состоит в том, что по отношению к каждому зарегистрированному пользователю данной вычислительной системы для каждого существующего файла указываются действия, которые разрешены или запрещены данному пользователю. Существовали попытки реализовать этот подход в полном объеме. Но это вызывало слишком большие накладные расходы как по хранению избыточной информации, так и по использованию этой информации для контроля правомочности доступа.

Защита файлов

- В большинстве современных систем управления файлами применяется подход к защите файлов, впервые реализованный в ОС UNIX.
- В этой системе каждому зарегистрированному пользователю соответствует пара целочисленных идентификаторов: идентификатор группы, к которой относится этот пользователь, и его собственный идентификатор в группе.
- Соответственно, при каждом файле хранится полный идентификатор пользователя, который создал этот файл, и отмечается, какие действия с файлом может производить он сам, какие действия с файлом доступны для других пользователей той же группы, и что могут делать с файлом пользователи других групп.
- Эта информация очень компактна, при проверке требуется небольшое количество действий, и этот способ контроля доступа удовлетворителен в большинстве случаев.

Режим многопользовательского доступа

- Если операционная система поддерживает многопользовательский режим, вполне реальна ситуация, когда два или более пользователей одновременно пытаются работать с одним и тем же файлом. Если все эти пользователи собираются только читать файл, ничего страшного не произойдет. Но если хотя бы один из них будет изменять файл, для корректной работы этой группы требуется взаимная синхронизация.

Области применения файлов

- Ну Вы это и сами знаете лучше меня

Потребности информационных систем

- Эти системы главным образом ориентированы на хранение, выбор и модификацию постоянно существующей информации. Структура информации зачастую очень сложна, и хотя структуры данных различны в разных информационных системах, между ними часто бывает много общего. На начальном этапе использования вычислительной техники для управления информацией проблемы структуризации данных решались индивидуально в каждой информационной системе. Производились необходимые надстройки над файловыми системами (библиотеки программ), подобно тому, как это делается в компиляторах, редакторах и т.д.

Потребности информационных систем

- Но поскольку информационные системы требуют сложных структур данных, эти дополнительные индивидуальные средства управления данными являлись существенной частью информационных систем и практически повторялись от одной системы к другой.
- Стремление выделить и обобщить общую часть информационных систем, ответственную за управление сложно структурированными данными, явилось первой побудительной причиной создания СУБД.
- Очень скоро стало понятно, что невозможно обойтись общей библиотекой программ, реализующей над стандартной базовой файловой системой более сложные методы хранения данных.

Пример

DEPART

Dept_Id	Dept_Name	Dept_Kol
1	Кафедра алгебры	3
2	Кафедра ЮНЕСКО	2

PERSON

Pers_Id	Pers_Name	Dept_Id
1	Иванов	1
2	Петров	2
3	Сидоров	1
4	Пушников	2
5	Шарипов	1

Спасибо за внимание!
ВОПРОСЫ?

Следующая лекция:
Функции СУБД.
Дореляционные СУБД