

ФГБОУ ВПО «Саратовский государственный университет
имени Н.Г. Чернышевского»

Факультет нелинейных процессов

А.А. Короновский, О.И. Москаленко, А.Е. Храмов

**УЧЕБНО-МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ
ДЛЯ СТУДЕНТОВ**

по курсу
«Принципы построения
и функционирования баз данных»

Саратов – 2014

Основные сведения об учебном курсе

В настоящем учебном курсе излагаются основы использования компьютерных технологий в науке и производстве на примере разработки баз данных, их администрирования и применения на производстве и в научных исследованиях. В курсе уделено большое внимание разработке моделей баз данных и реализации их на практике. Программа курса построена так, чтобы показать тесную связь теории построения и функционирования баз данных с проблемами, возникающими при практической деятельности.

Учебный курс состоит из лекций и практических занятий, взаимодополняющих друг друга. Основной целью лекций является ознакомление студентов с основными сведениями о принципах построения, и функционирования баз данных, в то время как главной задачей практических занятий является выработка и закрепление практических навыков по работе с базами данных. Очевидно, что материал лекций и практических занятий взаимосвязан друг с другом. В то же самое время, **пожалуйста, обратите особое внимание на тот факт, что часть материала будет рассматриваться либо только на лекциях, либо только на практических занятиях.**

Формой промежуточной аттестации по итогам освоения дисциплины по данному курсу является экзамен, который может проводиться в форме компьютерного тестирования или в устной форме, по решению кафедры, обеспечивающей данную дисциплину. Итоговый тест состоит из 25 вопросов, на прохождение теста студенту отводится 30 минут. За каждый полный и правильный ответ студенту начисляется два балла, всего после прохождения теста студент может набрать от нуля до 50 баллов.

В случае проведения экзамена в устной форме, максимальная сумма баллов, которую может получить студент за прохождение промежуточной аттестации, также составляет 50 баллов. При этом используется следующая система оценивания:

41 – 50 баллов:

Студент демонстрирует всестороннее, систематическое и глубокое знание учебного материала, умение свободно выполнять задания, предусмотренные программой, творческие способности в понимании, изложении и использовании материала.

31 – 40 баллов:

Студент демонстрирует полное знание учебного материала, правильно выполняет задания, предусмотренные программой, показывает систематический характер знаний по дисциплине.

11 – 30 баллов:

Студент демонстрирует знания основного учебного материала в объеме, необходимом для дальнейшей учебы и предстоящей работы по специальности, однако допускает погрешности в ответе на экзамене и при выполнении

экзаменационных заданий, но способен их устранить под руководством преподавателя.

0 – 10 баллов:

Студент демонстрирует «отрывочные» знания основного учебного материала, допускает принципиальные ошибки в ответе на экзамене и при выполнении экзаменационных заданий.

Экзаменационная оценка по дисциплине выставляется на основании рейтинга по дисциплине. Перевод полученных баллов в оценку производится по следующей шкале:

	«неудовлетв.»	«удовлетв.»	«хорошо»	«отлично»
Рейтинг по дисциплине (Сумма баллов, набранных студентом по итогам изучения дисциплины)	0–39	40–59	60–79	80–100

Текущая успеваемость обучающихся контролируется с помощью проведения контрольных работ, проверки выполнения заданий практических занятий, а также с помощью средств контроля самостоятельной работы. Работа студента в течение семестра оценивается преподавателями, ведущими эти занятия, в баллах, от нуля до шестидесяти, включая баллы за самостоятельную внеаудиторную работу:

Цикл практических занятий условно может быть разделен на четыре части:

1. Реляционная модель данных. Нормализация.
2. Работа с объектами SQL Server
3. Работа с Transact SQL
4. Проектирование реляционных баз данных

Каждой части посвящена отдельная глава методического пособия, содержащая (в большинстве случаев) краткие теоретические сведения по данной тематике, иллюстративные примеры, методические указания и требования к знаниям и умениям студентов, контрольные вопросы и задания для закрепления материала. Выходные данные методического пособия по курсу приведены ниже:

А.А. Короновский, О.И. Москаленко. Хранение и обработка информации для специалистов в области нелинейных колебательных и волновых физических процессов. Учебное пособие. Издательство «Новый ветер», 2009.

Каждый студент выполняет индивидуальное задание. Узнать номер своего варианта студент может у преподавателя во время проведения занятий. Порядок выполнения заданий четко соответствует структуре курса практических занятий. Максимальное число баллов, которое может получить студент за практические задания, составляет 50 баллов. Распределение баллов по темам является следующим:

Тема	Общее кол-во баллов	Распределение по разделам	Кол-во баллов		
Реляционная модель данных	10	Теория нормальных форм	3		
		Нормализация универсальных отношений	7		
Работа с объектами SQL Server	10	Знание теории	3		
		Создание структуры БД	2		
		Ввод, изменение и удаление данных	2		
		Оператор ALTER TABLE	3		
Работа с Transact SQL	15	Знание теории	3		
		Задание 1	Простая выборка данных (задания 1-2)	1	
			Объединение таблиц	2	
			Работа с функциями агрегирования, группировки	1	
			Представления	1	
			Хранимые процедуры	1	
			Задание 2	1	
		Проектирование реляционных баз данных	15	Знание теории	3
				Концептуальное проектирование	6
				Логическое проектирование	3
Физическое проектирование	3				

При проставлении баллов за практические задания будет учитываться как владение теоретическим материалом, так и умение его применять на практике. Важно обратить внимание на то, что **зачет по заданию вовсе не означает получение им максимального балла по этому заданию.**

Настоящие учебно-методические материалы призваны помочь студенту в освоении материала учебного курса и подготовке к экзамену. В них содержится информация о требованиях к умениям и знаниям студентов, содержании учебного курса, структуре итогового теста. Также в этих материалах содержатся перечень рекомендуемой литературы. **Особое внимание рекомендуется уделить лекционному материалу: студент найдет в настоящем сборнике слайды всех лекций с местом для заметок и примечаний. Студенту рекомендуется распечатывать соответствующие страницы перед каждой лекцией и использовать их для конспектирования основного содержания лекций.** Это должно облегчить работу студента и повысить ее эффективность.

Требования к умениям и знаниям студентов

Студент обязан знать основные понятия теории разработки и администрирования баз данных, принципы построения и функционирования баз данных, уметь применять полученные знания на практике при решении задач.

Студент должен иметь четкое представление об уровнях проектирования и разработки баз данных и уметь проводить необходимые действия на каждом уровне. На уровне концептуального проектирования студент должен обладать навыками анализа соответствующей предметной области, выявлять необходимый набор сущностей, требуемый набор атрибутов для каждой сущности, классифицировать атрибуты, выделять ключи (потенциальные, первичные, альтернативные), определять связи между сущностями, включая связи «подкласс-суперкласс», где это необходимо; формализовать связи между сущностями (показатель кардинальности, степень участия), проводить процесс специализации/генерализации.

На этапе логического проектирования студент должен быть способен выполнить преобразование концептуальной модели в логическую, для чего он должен уметь осуществлять удаление связей типа «многие ко многим», удаление связей с атрибутами; преобразовывать связи «суперкласс/подкласс»; удалять множественные атрибуты; выделять атрибуты с фиксированным набором значений; удалять избыточные связи. На этом этапе студент должен также уметь осуществлять проверку логической модели данных с помощью правил нормализации и, в случае необходимости, проводить процесс нормализации вплоть до нормальной формы Бойса-Кодда. Студент должен иметь представление об универсальном отношении, и нормальных формах, владеть терминологией теории нормальных форм, уметь находить детерминанты, знать, что такое транзитивная зависимость и т.п. Студент должен знать также о способах приведения универсального отношения к первой нормальной форме, уметь осуществлять преобразования первой нормальной формы во вторую и третью нормальную форму и нормальную форму Бойса-Кодда.

На этапе физического проектирования студент должен уметь осуществить переход от логической модели к физической и манипулировать базой данных посредством Transact-SQL: уметь создавать и удалять базу данных, делать ее активной, работать с таблицами и другими объектами SQL (видами, хранимыми процедурами). Студент должен знать основные команды Transact-SQL и их базовый синтаксис, уметь получать информацию об имеющихся в базе данных пользовательских и системных таблицах, создавать, удалять и изменять таблицы, определять в них первичные и внешние ключи как на этапе создания, так и при изменении структуры, знать основные типы данных, поддерживаемые SQL, а также дополнительные опции атрибутов, уметь изменять тип заданного столбца, добавлять в таблицу новый столбец, знать основные ограничения на то или иное изменение и уметь от них избавиться. Кроме того, он должен уметь вводить данные в

таблицы, знать основные трудности, которые могут возникнуть при вводе данных в случае наличия тех или иных опций атрибутов, а также вносить изменения во введенные данные и удалять их в случае возникновения необходимости как полностью, так и частично. Студент должен уметь осуществлять выборку данных из таблиц, знать синтаксис оператора SELECT, уметь делать как простую выборку всех данных из одной таблицы, так и выполнять сложные запросы одновременно к множеству связанных между собой таблиц, уметь создавать представления и хранимые процедуры и работать с ними.

Содержание учебного курса

Введение. Основные сведения. Применение баз данных. Файловые системы, их достоинства и недостатки. Базы данных (БД) и системы управления базами данных (СУБД). Преимущества и недостатки СУБД. Архитектура многопользовательских СУБД: телеобработка, файловый сервер, клиент-сервер. Преимущества технологии «клиент-сервер». Двух- и трехуровневая архитектура технологии «клиент-сервер». Преимущества трехуровневой архитектуры. Модели данных: иерархическая, сетевая, реляционная.

Раздел 1. Реляционная модель данных. Свойства реляционной таблицы. Используемая терминология: отношение, атрибут, составной атрибут, домен, кортеж, ключ. Требования, предъявляемые к потенциальному ключу. Первичный, альтернативный и внешний ключи. Требования, предъявляемые к реляционным базам данных. Нормализация. Теория нормальных форм. Первая, вторая и третья нормальные формы. Нормальная форма Бойса-Кодда. Четвертая и пятая нормальные формы.

Раздел 2. Основы проектирования реляционных баз данных. Жизненный цикл базы данных. Проектирование базы данных. Средства для проектирования (CASE-средства) Три уровня проектирования: концептуальный, логический, физический. Проектирование базы данных на концептуальном уровне. Этапы концептуального проектирования. Понятия «сущность», «связь», «атрибут», «домен». Виды связей между сущностями: один к одному, один ко многим, многие ко многим. Специализация или генерализация типов сущностей. Переход от концептуального уровня проектирования к логическому. Этапы логического проектирования: преобразование концептуальной модели в логическую, проверка модели с помощью правил нормализации, проверка модели в отношении транзакций пользователей, определение требований поддержки целостности данных, обсуждение логических моделей данных с конечными пользователями, документирование. Физический уровень проектирования.

Раздел 3. Microsoft SQL Server. Объекты SQL Server: таблицы, представления (виды), индексы, ключи, умолчания, правила, ограничения целостности, хранимые процедуры, триггеры, определяемые пользователем типы данных, определяемые пользователем функции. Именованные объекты. Физическая архитектура баз данных. Журнал транзакций. Управление таблицами с помощью Transact SQL и Enterprise Manager.

Раздел 4. Работа с объектами SQL Server. Создание базы данных и работа с ней. Операторы CREATE DATABASE и USE. Удаление базы данных. Оператор DROP DATABASE. Работа с таблицами. Ограничения целостности. Первичные и внешние ключи. Автонумерация. Создание, модификация, удаление таблиц. Управление таблицами с помощью Transact SQL. Операторы CREATE TABLE, ALTER TABLE, DROP TABLE.

Раздел 5. Основы T-SQL. Операторы SELECT, INSERT, DELETE, UPDATE. Извлечение данных из одной и нескольких таблиц. Объединение таблиц. Оператор JOIN. Понятие транзакции. Управление транзакциями. Создание и работа с другими объектами SQL Server: виды, хранимые процедуры.

Раздел 6. Индексирование баз данных. Хранение данных. Страницы и группы страниц. Индексы. Использование Индексов. Классификация индексов: кластерный индекс, некластерный индекс, уникальный индекс. Фактор заполнения. Индексирование представлений (видов). Управление индексами: создание, перестроение, переименование и удаление индекса. Фрагментация.

Раздел 7. Возможные неприятности и пути их преодоления. Классификация возможных неприятностей, возникающих при использовании баз данных. Система безопасности. Права доступа. Пользователи и роли. Создание отказоустойчивой системы: Резервный сервер, кластер, технология RAID, резервное копирование. Типы резервного копирования: полная копия, разностная копия, копия журнала транзакций, резервное копирование файлов и групп. Возможная стратегия для резервного копирования больших баз данных. Восстановление баз данных.

Раздел 8. Другие системы управления базами данных. Целесообразность выбора того или иного типа системы управления базами данных. Microsoft Access – настольная система управления базами данных. Создание таблиц и запросов в MS Access. Формы и отчеты в MS Access. Инструменты для создания таблиц, запросов, форм и отчетов. Элементы управления и их свойства в MS Access. Доступ к гетерогенным источникам данных в Microsoft SQL Server. Экспорт и импорт данных.

Рекомендуемая литература

1. *Гарсиа М.Ф. и др.* Microsoft SQL Server. Справочник администратора. ЭКОМ: Москва, 2002.
2. *Хоторн Р.* Разработка баз данных Microsoft SQL Server 2000 на примерах. М.: Издат. дом «Вильямс», 2001.
3. *Баженова И.Ю.* Основы проектирования приложений баз данных. Изд-во: Интернет-университет информационных технологий. 2006.
4. *Кузнецов С.Д.* Основы баз данных. Изд-во: Интернет-университет информационных технологий. 2005.
5. *Уилтон П., Колби Дж.* Язык запросов SQL для начинающих. М.: Диалектика, 2006.
6. *Шпенник М., Слендж О.* Руководство администратора баз данных Microsoft SQL Server 2000. М.: Издат. дом «Вильямс», 2001.
7. *Мейер М.* Теория реляционных баз данных. - М.: Мир, 1987. - 608 с.
8. *Мартин Дж.* Организация баз данных в вычислительных системах. М.: Мир, 1980. - 662 с.
9. *Мартин Дж.* Планирование развития автоматизированных систем. М.: Финансы и статистика, 1984. - 196 с.
10. *Ульман Дж.* Основы систем баз данных. М.: Финансы и статистика, 1983. - 334 с.
11. *Тиори Т., Фрай Дж.* Проектирование структур баз данных: В 2-х кн. М.: Мир, 1985. - Кн. 1. - 287 с.; Кн. 2. - 320 с.
12. *Хаббард Дж.* Автоматизированное проектирование баз данных. М.: Мир, 1984. - 294 с.
13. *Дейт К.Дж.* Введение в системы баз данных. Киев: Диалектика, 1998. - 784 с.
14. *Бойко В.В., Савинков В.М.* Проектирование баз данных информационных систем. - М.: Финансы и статистика, 1989. - 351 с.
15. *Кириллов В.В.* Основы проектирования реляционных баз данных: Учеб. пособие / Санкт-Петербургский государственный институт точной механики и оптики - СПб., 1994 - 88 с.
16. *Гринь А.М.* Логическое проектирование и реализация баз данных: Учеб. пособие. - Новосибирск: Изд-во НГТУ, 1994.
17. *Грабер М.* Введение в SQL. - М.: ЛОРИ, 1996. - 380 с.
18. *Стасьшин В.М.* Язык структурных запросов SQL: Учеб. пособие. - Новосибирск: Изд-во НГТУ, 1996. - 33 с.
19. *Конноли Т., Бегг К., Страчан А.* Базы данных. Проектирование, реализация и сопровождение. Теория и практика. М.: Издательский дом «Вильямс», 2000. - 1120 с.

ИННОВАЦИОННО-ОБРАЗОВАТЕЛЬНАЯ ПРОГРАММА
 Саратовского государственного университета им. Н.Г.Чернышевского

Принципы построения и функционирования баз данных

Лекция первая.
Основные сведения

Факультет нелинейных процессов
 Физика открытых нелинейных систем
 Саратов - 2008

2

О чем сегодняшняя лекция?

- Назначение и применение баз данных;
- Какие бывают базы данных (файловые системы, СУБД);
- Архитектура многопользовательских СУБД (телеобработка, файл-сервер, клиент-сервер);
- Модели данных (иерархическая, сетевая, реляционная).

3

Применение баз данных

- Супермаркет;
- Банк;
- Транспорт (авиационный, железнодорожный и т.д.);
- Библиотека;
- Университет;
- Налоговая инспекция;
- Предприятие (отдел кадров, бухгалтерия, производство и т.д.);
- Интернет-магазин.

4



База данных

Совместно используемый набор логически связанных данных (и описание этих данных), предназначенный для удовлетворения информационных потребностей организации.

Файловые системы

Набор программ, которые выполняют для пользователей некоторые операции. Каждая программа определяет свои собственные данные и управляет ими.

5



Систематизация информации

- Записная книжка;
- Картотека;
- Подшивка документов.

Позволяют работать с большим числом объектов, хранить и извлекать информацию, но совершенно не подходят для тех случаев, когда нужно установить перекрестные связи или выполнить обработку сведений.

6



Файловые системы были разработаны в ответ на потребность в получении более эффективных способов доступа к данным. При этом был использован децентрализованный подход, когда сотрудники каждого отдела при помощи специалистов по обработке данных работают со своими собственными данными и хранят их в своем отделе.



8

Пример данных, хранимых в файловых системах

- **Бухгалтерия:** Фамилия, Имя, Отчество, дата рождения, пол, подразделение, должность, разряд, ставка, оклад, ИНН, номер пенсионного страхования, ...
- **Отдел кадров:** Фамилия, Имя, Отчество, дата рождения, пол, подразделение, должность, разряд, ставка, образование, стаж работы, домашний адрес, телефон, ...
- ...

9

Недостатки файловых систем

- Разделение и изоляция данных;
- Дублирование данных;
- Зависимость от программ и данных;
- Несовместимость файлов;
- Фиксированные запросы/ быстрое увеличение количества приложений.

10

Недостатки файловых систем являются следствием того, что

- Данные хранятся в разных местах;
- Определение данных содержится внутри приложений, а не хранится отдельно и независимо от них;
- Кроме приложений не предусмотрено никаких других инструментов доступа к данным и их обработки.

Для преодоления вышеупомянутых недостатков и повышения эффективности работы необходимо использовать новый подход, а именно базу данных (БД) и систему управления базами данных (СУБД).

11

Определения:

БД - совместно используемый набор логически связанных данных (и описание этих данных), предназначенный для удовлетворения информационных потребностей организации.

СУБД - с это программное обеспечение, с помощью которого пользователи могут определять, создавать и поддерживать базу данных, а также осуществлять к ней контролируемый доступ.



Преимущества СУБД



- Контроль за избыточностью, целостностью и непротиворечивостью данных;
- Увеличение полезной информации при том же объеме хранимых данных;
- Повышенная безопасность;
- Совместное использование данных;

Преимущества СУБД (2)



- Улучшение показателей производительности;
- Упрощение сопровождения системы за счет независимости от данных;
- Улучшенное управление параллельностью;
- Развитые службы резервного копирования и восстановления данных;
- Повышение эффективности с ростом масштабов системы.

Недостатки СУБД



- Сложность;
- Размер;
- Стоимость;
- Дополнительные затраты на аппаратное обеспечение;
- Затраты на преобразование;
- Более серьезные последствия при выходе системы из строя.

16

Стоимость (на 01.07.2004)

MS SQL Server 2000	IBM DB2 Version 8.1	Oracle 10g
Standard Edition \$4,999 US	Workgroup Edition \$7,500 US	Standard Edition \$21,000 US
Enterprise Edition \$19,999 US	Enterprise Edition \$123,600 US	Enterprise Edition \$96,000 US

17

Архитектура многопользовательских СУБД

- Телеобработка;
- Файловый сервер (файл-сервер);
- Клиент-сервер (двухуровневая и трехуровневая архитектуры);

18

Телеобработка

```

    graph TD
      A[Центральная ЭВМ] --- B1[Терминал]
      A --- B2[Терминал]
      A --- B3[Терминал]
      A --- B4[Терминал]
      A --- B5[Терминал]
  
```



- 20
- ### Недостатки файлового сервера
- Большой объем сетевого трафика;
 - На каждой рабочей станции должна находиться полная копия СУБД;
 - Управление параллельностью, восстановлением и целостностью усложняется, поскольку доступ к одним и тем же файлам могут осуществлять сразу несколько экземпляров СУБД.



22

Функции, выполняемые участниками взаимодействия

Клиент	Сервер
Управляет пользовательским интерфейсом	Принимает и обрабатывает запросы к БД от клиентов
Принимает и проверяет запрос пользователя	Проверяет полномочия пользователей
Выполняет приложение	Поддерживает целостность
Генерирует запрос к БД и передает его серверу	Выполняет запросы и возвращает результаты клиенту
Отображает полученные данные пользователю	Обеспечивает параллельный доступ к БД
	Обеспечивает управление восстановлением

23

Преимущества технологии «клиент-сервер»

- Более широкий доступ к существующим БД;
- Повышается общая производительность системы;
- Снижается стоимость аппаратного обеспечения;
- Сокращаются коммуникационные расходы;
- Повышается уровень непротиворечивости данных;
- Согласованное изменение данных множеством пользователей.

24

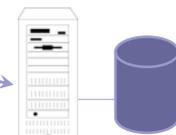
Двухуровневая архитектура

Первый уровень:
Клиент

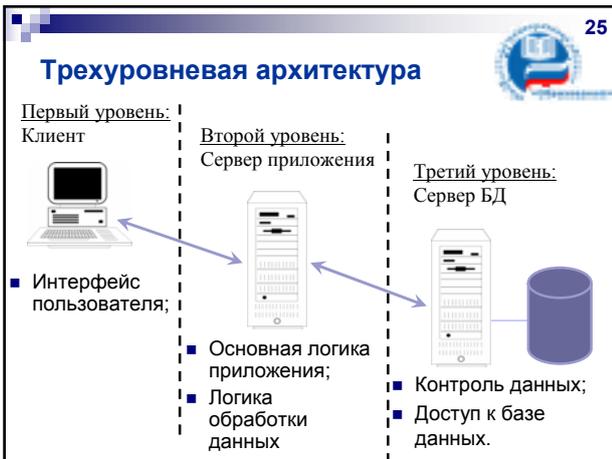


- Интерфейс пользователя;
- Основная логика обработки данных.

Второй уровень:
Сервер БД



- Контроль данных на серверной стороне;
- Доступ к базе данных.



- 26
- ### Преимущества трёхуровневой архитектуры
- «Тонкий» клиент, для которого требуется менее дорогостоящее аппаратное обеспечение;
 - Централизация бизнес-логики на одном сервере приложения;
 - Дополнительная модульность упрощает модификацию/замену ПО каждого уровня без оказания влияния на другие уровни;
 - Органично подходит для Web-технологий.

- 27
- ### Модели данных
- Иерархическая модель данных;
 - Сетевая модель данных;
 - Реляционная модель данных.

28

Иерархическая модель

```

    graph TD
      A[A] --- B1[B1]
      A --- B2[B2]
      A --- B3[B3]
      B2 --- C1[C1]
      B2 --- C2[C2]
      C2 --- D1[D1]
      D1 --- E1[E1]
  
```

29

Сетевая модель

```

    graph TD
      N1[ ] --- N2[ ]
      N1 --- N3[ ]
      N2 --- N4[ ]
      N3 --- N4[ ]
      N4 --- N5[ ]
  
```

30

Реляционная модель

```

    graph LR
      T1[ ] --> T2[ ]
      T3[ ] --> T2[ ]
  
```

ИННОВАЦИОННО-ОБРАЗОВАТЕЛЬНАЯ ПРОГРАММА
 Саратовского государственного университета им. Н.Г.Чернышевского

Принципы построения и функционирования баз данных

**Лекция вторая.
 Реляционная модель**

Факультет нелинейных процессов
 Физика открытых нелинейных систем
 Саратов - 2008

О чем сегодняшняя лекция?

- Что такое «реляционная модель данных»;
- Используемая терминология;
- Требования, предъявляемые к реляционным объектам;
- Ключи и требования, предъявляемые к ним;
- Нормальные формы и процесс нормализации.

Реляционная модель данных

В основе реляционной модели данных лежит идея о том, что любой набор данных может быть представлен в виде двумерной таблицы или набора связанных таких таблиц.

4

Свойства реляционной таблицы

- Любой элемент таблицы является минимальным элементом данных;
- Элементы в пределах столбца имеют одинаковый тип (числовой, символьный и т.п.);
- Все столбцы таблицы имеют уникальное имя;
- В таблице отсутствуют совпадающие строки;
- Порядок следования строк и столбцов в таблице может быть произвольным.

5

Терминология

- Отношение – плоская таблица, состоящая из столбцов и строк (таблица);
- Атрибут – поименованный столбец отношения (столбец, колонка, поле);
- Составной атрибут – совокупность столбцов
- Домен – набор допустимых значений для одного (или нескольких) атрибутов;
- Кортеж – это строка отношения (строка, запись).
- Ключ – множество атрибутов, задание значений которых позволяет однозначно определить значение остальных атрибутов таблицы (потенциальный ключ)

6

Терминология (пример)

целое	строка		целое		Типы данных
номер	имя	должность	деньги		Домены
Табельный номер	Имя	Должность	Оклад	Премия	Атрибуты
2934	Иванов	инженер	112	40	Кортежи
2935	Петров	вед. инженер	144	50	
2936	Сидоров	бухгалтер	92	35	

Ключ (под номером 2934)
Отношение (под таблицей)

Требования, предъявляемые к потенциальному ключу



7

- Уникальность. Никакие два кортежа не имеют одинаковых значений для всех атрибутов, входящих в потенциальный ключ;
- Минимальность (неприводимость). Ни один из атрибутов потенциального ключа не может быть из него исключен без нарушения уникальности.

Терминология (2)



8

- Первичный ключ (*Primary Key – PK*) – потенциальный ключ, который выбран для уникальной идентификации кортежей внутри отношения;
- Альтернативный ключ (*Alternative Key – AK*) – потенциальный ключ, который не выбран в качестве первичного;
- Внешний ключ (*Foreign Key – FK*) – атрибут или множество атрибутов внутри отношения, которое соответствует потенциальному ключу другого (или того же) отношения.

Требования к реляционным БД



9

- Каждая таблица в базе данных имеет уникальное имя в пределах БД;
- Все строки в таблице однотипны;
- Строки таблицы обязательно отличаются друг от друга хотя бы одним значением;
- Каждый столбец таблицы имеет уникальное имя в пределах таблицы и служит для хранения данных строго определенного типа;
- При обращении к данным можно обращаться к любой строке или столбцу данных.

10

Нормализация

- Нормализация – процесс приведения структур данных в состояние, обеспечивающее лучшие условия выборки, добавления, изменения и удаления данных.

Каждый факт появляется и хранится лишь в одном месте, то есть исключена избыточность информации.

- Универсальное отношение – таблица, в которую включены все интересующие атрибуты.

11

Теория нормальных форм

- Ненормализованная форма (NNF) – таблица, содержащая одну или несколько повторяющихся групп данных.
- Первая нормальная форма (1NF) – отношение, в котором ни одно поле строки не содержит более одного значения и любое ключевое поле не пусто.

12

Модельная ситуация

Владелец



Менеджер





Объект недвижимости



Арендатор

13

Атрибуты

- **CNo** – номер арендатора;
- **CName** – ФИО арендатора;
- **ObjNo** – номер объекта недвижимости;
- **Address** – адрес объекта недвижимости;
- **RStart** – дата начала аренды;
- **RFinish** – дата окончания аренды;
- **Rent** – стоимость аренды;
- **OwnNo** – номер владельца объекта недвижимости;
- **OName** – ФИО владельца недвижимости.

14

Ненормализованная таблица

CNo	CName	ObjNo	Address	RStart	RFinish	Rent	OwnNo	OName
CR76	Петров Петр Петрович	PG4	Саратов, Чапаева, 15;	01.07.94;	31.08.96;	1500;	CO40;	Еремин Антон Павлович;
		PG16	Саратов, Соляная, 24	01.09.96	01.09.98	2000	CO93	Дурова Анна Семеновна
CR56	Змеев Игорь Иванович	PG4	Саратов, Чапаева, 15;	01.09.92;	10.06.94;	1500;	CO40;	Еремин Антон Павлович;
		PG36	Саратов, Азина, 25;	10.10.94;	01.12.95;	1700;	CO93;	Дурова Анна Семеновна;
		PG16	Саратов, Соляная, 24	01.01.96	15.08.96	2000	CO93	Дурова Анна Семеновна


Повторяющаяся группа сведений

15

Первая нормальная форма (1NF)

Lease

CNo	ObjNo	CName	Address	RStart	RFinish	Rent	OwnNo	OName
CR76	PG4	Петров Петр Петрович	Саратов, Чапаева, 15	01.07.94	31.08.96	1500	CO40;	Еремин Антон Павлович;
CR76	PG16	Петров Петр Петрович	Саратов, Соляная, 24	01.09.96	01.09.98	2000	CO93	Дурова Анна Семеновна
CR56	PG4	Змеев Игорь Иванович	Саратов, Чапаева, 15	01.09.92	10.06.94	1500	CO40;	Еремин Антон Павлович;
CR56	PG36	Змеев Игорь Иванович	Саратов, Азина, 25	10.10.94	01.12.95	1700	CO93	Дурова Анна Семеновна
CR56	PG16	Змеев Игорь Иванович	Саратов, Соляная, 24	01.01.96	15.08.96	2000	CO93	Дурова Анна Семеновна


Повторяющаяся группа сведений

16

Проблемы 1NF

- Избыточность данных;
- Потенциальная противоречивость данных (аномалии обновления);
- Аномалии вставки;
- Аномалии удаления.

17

Функциональная зависимость

- Функциональная зависимость. Атрибут В функционально зависит от атрибута А (обозначается $A \rightarrow B$), когда каждое значение атрибута А связано только с одним значением атрибута В.
- Детерминант – атрибут (или группа атрибутов), расположенный на диаграмме функциональной зависимости слева от символа стрелки: $A \rightarrow B$.

18

Функциональная зависимость (пример)

№	ФИО сотрудника	Должность
567	Иванов И.И.	менеджер
568	Петров П.П.	консультант
569	Сидоров И.П.	менеджер
570	Веселов А.М.	исп. директор
571	Захаров Ю.Е.	консультант
572	Спиридонов И.М.	менеджер

↑
PK

№ Сотрудника → Должность Должность ✗ № сотрудника
 № Сотрудника → ФИО сотрудника

19

Информация к размышлению

№	ФИО сотрудника	Должность	Месяц	Выплата	Сумма
567	Иванов И.И.	менеджер	июнь	аванс	13400
568	Петров П.П.	консультант	июнь	аванс	13200
569	Сидоров И.П.	менеджер	июнь	аванс	13400
567	Иванов И.И.	менеджер	июнь	зарплата	13600
568	Петров П.П.	консультант	июнь	зарплата	13400
569	Сидоров И.П.	менеджер	июнь	зарплата	13600
569	Сидоров И.П.	менеджер	июнь	премия	8500

20

Вторая нормальная форма (2NF)

- Полная функциональная зависимость – атрибут В функционально зависит от полного значения составного атрибута А и не зависит ни от какого подмножества этого атрибута.
- Вторая нормальная форма (2NF) – отношение, которое находится в 1NF и каждый атрибут которого, не входящий в состав первичного ключа, характеризуется полной функциональной зависимостью от этого ключа.

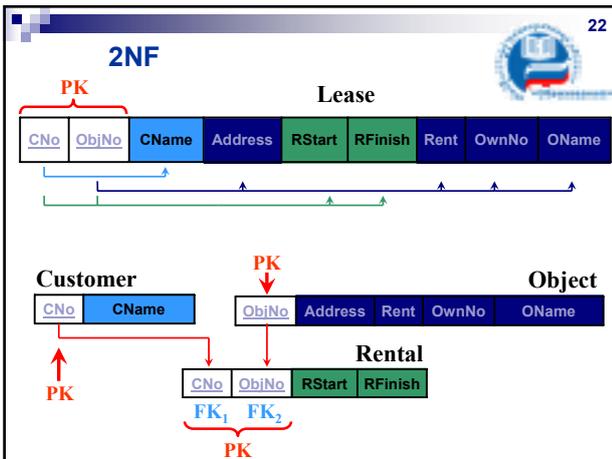
21

1NF → 2NF

PK

CNo	ObjNo	CName	Address	RStart	RFinish	Rent	OwnNo	OName
Lease								

- FD1: CNo → CName
- FD2: ObjNo → Address, Rent, OwnNo, OName
- FD3: CNo, ObjNo → RStart, RFinish (PK)



23

2NF

Customer

CNo	CName
CR76	Петров Петр Петрович
CR56	Змеев Игорь Иванович

Object

CNo	ObjNo	RStart	RFinish
CR76	PG4	01.07.94	31.08.96
CR76	PG16	01.09.96	01.09.98
CR56	PG4	01.09.92	10.06.94
CR56	PG36	10.10.94	01.12.95
CR56	PG16	01.01.96	15.08.96

Rental

ObjNo	Address	Rent	OwnNo	OName
PG4	Саратов, Чапаева, 15	1500	CO40	Еремин Антон Павлович
PG16	Саратов, Соляная, 24	2000	CO93	Дурова Анна Семеновна
PG36	Саратов, Азина, 25	1700	CO93	Дурова Анна Семеновна

24

Транзитивная зависимость

- Транзитивная зависимость. Если для атрибутов A, B и C существуют функциональные зависимости $A \rightarrow B$ и $B \rightarrow C$, то атрибут C транзитивно зависит от атрибута A через атрибут B (при условии, что атрибут A функционально не зависит ни от атрибута B, ни от атрибута C).

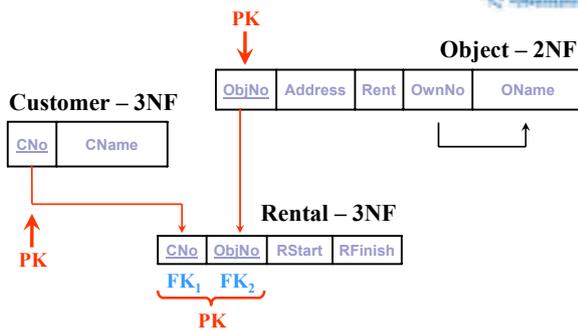
Третья нормальная форма

25

- Третья нормальная форма (3NF) – отношение, которое находится в первой и второй нормальных формах и не имеет не входящих в первичный ключ атрибутов, которые находились бы в транзитивной функциональной зависимости от этого первичного ключа.

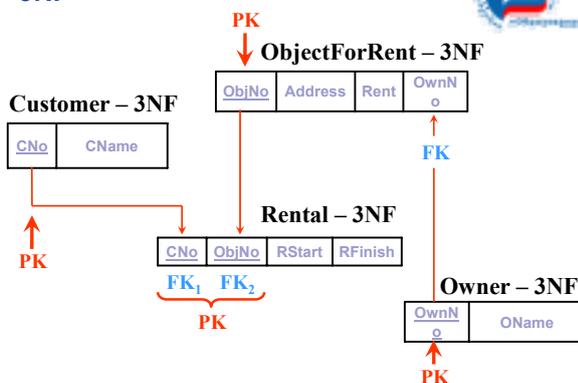
2NF → 3NF

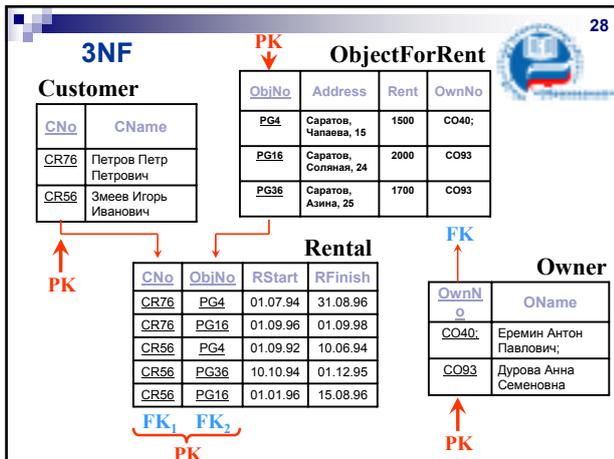
26

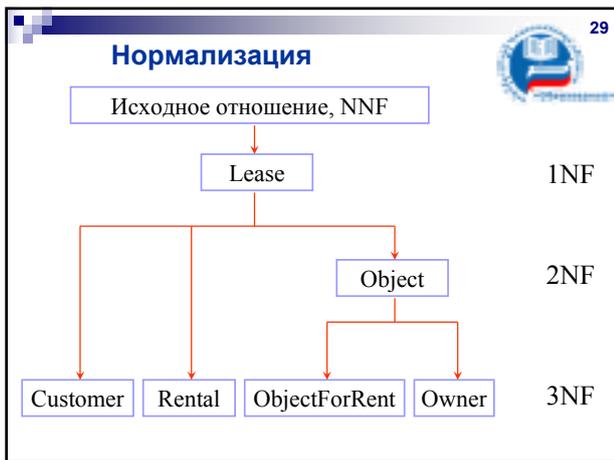


3NF

27







Нормальная форма Бойса-Кодда

- Нормальная форма Бойса-Кодда (НФБК) – отношение, которое находится в третьей нормальной форме и каждый детерминант которого является потенциальным ключом.

31

Модельная ситуация

Менеджер

Кабинет

Арендатор

32

Универсальное отношение

CNo	Date	Time	StaffNo	RoomNo
CR56	14.08.03	11-00	CS05	12
CR76	14.08.03	12-00	CS14	13
CR234	14.08.03	11-00	CS14	13
CR769	14.08.03	14-00	CS05	12
CR87	14.08.03	12-00	CS05	12
CR107	14.08.03	16-30	CS14	13
CR76	17.08.03	15-00	CS05	13

33

3NF → NFBK

CNo	Date	Time	StaffNo	RoomNo
-----	------	------	---------	--------

PK (Primary Key) is indicated over CNo and Date.

FD1: CNo, Date → Time, StaffNo, RoomNo

FD2: StaffNo, Date, Time → CNo, RoomNo

FD3: Date, Time, RoomNo → StaffNo, CNo

Потенциальные ключи:

- **FD1:** CNo, Date → Time, StaffNo, RoomNo;
- **FD2:** StaffNo, Date, Time → CNo, RoomNo;
- **FD3:** Date, Time, RoomNo → StaffNo, CNo

NFBK

34

Date	StaffNo	RoomNo
14.08.03	CS05	12
14.08.03	CS14	13
17.08.03	CS05	13

CNo	Date	StaffNo	Time
CR56	14.08.03	CS05	11-00
CR76	14.08.03	CS14	12-00
CR234	14.08.03	CS14	11-00
CR769	14.08.03	CS05	14-00
CR87	14.08.03	CS05	12-00
CR107	14.08.03	CS14	16-30
CR76	17.08.03	CS05	15-00

PK (Primary Key) is indicated over the Date column in the second table.

FK (Foreign Key) is indicated between the Date and StaffNo columns of the second table and the Date and StaffNo columns of the first table.

Устранение избыточности данных

35

Многозначная зависимость

36

В случае многозначной зависимости, существующей между атрибутами А, В и С некоторого отношения, для каждого значения атрибута А имеется набор значений атрибута В и набор значений атрибута С. Однако, входящие в эти наборы значения атрибутов В и С не зависят друг от друга.

A → B
A → C

Четвертая нормальная форма



Отношение в нормальной форме Бойса-Кодда, которое не содержит нетривиальных многозначных зависимостей, является четвертой нормальной формой (4NF)

Пятая нормальная форма



- Зависимость соединения – свойство декомпозиции, которое вызывает генерацию сложных строк при обратном соединении декомпозированных отношений с помощью операции естественного соединения.
- Пятая нормальная форма (5NF) – отношение без зависимостей соединения

ИННОВАЦИОННО-ОБРАЗОВАТЕЛЬНАЯ ПРОГРАММА
 Саратовского государственного университета им. Н.Г.Чернышевского

Принципы построения и функционирования баз данных

Лекция третья.
Проектирование баз данных

Факультет нелинейных процессов
 Физика открытых нелинейных систем
 Саратов - 2008

О чем сегодняшняя лекция?

- Жизненный цикл базы данных
- Этапы проектирования БД и средства проектирования
- Концептуальное проектирование
 - Теоретические сведения
 - Пример проектирования
- Логическое проектирование
 - Теоретические сведения
 - Пример проектирования

Жизненный цикл базы данных

- Планирование разработки базы данных;
- Определение требований к системе;
- Сбор и анализ требований пользователей;
- Проектирование базы данных;
- Разработка приложений;
- Создание прототипов (необязательно);
- Реализация БД и приложений;
- Конвертирование и загрузка данных;
- Тестирование
- Эксплуатация и сопровождение.

4

Этапы проектирования БД

- ✓ Концептуальное проектирование;
- ✓ Логическое проектирование;
- ✓ Физическое проектирование;

5

CASE-средства (Computer Aided Software Engineering)

- Стандарты;
- Интеграция;
- Поддержка стандартных методов;
- Непротиворечивость;
- Автоматизация;
- Интегрированность с современными СУБД

6

Sybase PowerDesigner

7

Computer Associates ERwin

8

Концептуальное проектирование

Процедура конструирования информационной модели, не зависящей от каких-либо физических условий реализации:

- выбранный тип СУБД,
- состав программ приложения,
- используемый язык программирования,
- операционная система
- другие физические особенности.

9

Этапы концептуального проектирования

- Определение типов сущностей;
- Определение типов связей;
- Определение атрибутов и связывание их с типами сущностей и связей;
- Определение доменов атрибутов;
- Определение потенциальных и первичных ключей;
- Специализация или генерализация типов сущностей;
- Обсуждение локальных концептуальных моделей данных с конечными пользователями.
- Документирование.

10

Модельная ситуация

Владелец  Кабинет 

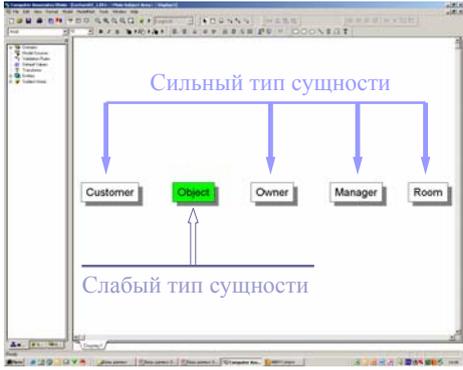
Менеджер 

Объект недвижимости  

Арендатор

11

1. Определение типов сущностей



12

2. Определение типов связей

- Показатель кардинальности (количество возможных связей для каждой из сущностей-участниц):
 - Один к одному (1:1);
 - Один ко многим (1:N);
 - Многие ко многим (N:N).
- Степень участия:
 - полная;
 - частичная;

13

2. Определение типов связей

14

3. Определение атрибутов

Атрибут – свойство сущности

- Простой – состоящий из одного компонента с независимым существованием;
- Составной – состоит из нескольких компонентов с независимым существованием;
- Однозначный – содержит одно значение для одной сущности;
- Многочисленный – содержит несколько значений;
- Производный – его значение определяется на основе значений других атрибутов.

15

3. Определение атрибутов

16

Домен атрибута

Домен атрибута – набор значений, которые могут быть присвоены атрибуту.

Пример:

Атрибут: «месяц»
 Домен: «январь», «февраль», «март», «апрель», «май», «июнь», «июль», «август», «сентябрь», «октябрь», «ноябрь», «декабрь»

17

4. Определение доменов атрибутов

The screenshot shows a UML class diagram with the following classes and attributes:

- Customer**: CustNumber, CustName, CustPhone
- Room**: RoomNumber, Category
- Object**: ObjNumber, Address, Rent
- Owner**: OwnNumber, OwnName, OwnPhone
- Manager**: MngNumber, MngName, MngPhone

Relationships:

- Customer *visit / is visited* Room
- Customer *rent / is rented* Object
- Object *owns / is owned* Owner
- Room *occupies / is occupied* Manager

18

5. Определение ключей

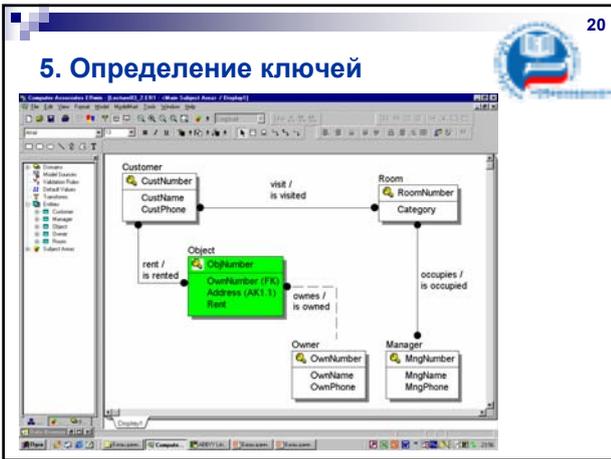
- Потенциальные ключи;
- Первичные ключи;
- Альтернативные ключи.

Выбор первичного ключа



- Потенциальный ключ с минимальным набором атрибутов;
- Потенциальный ключ, вероятность изменения значений которого минимальна;
- Потенциальный ключ, который имеет минимальную вероятность потери уникальности в будущем;
- Потенциальный ключ, значения которого имеют минимальную длину (текстовый тип);
- Потенциальный ключ, с которым проще всего работать пользователю.

5. Определение ключей



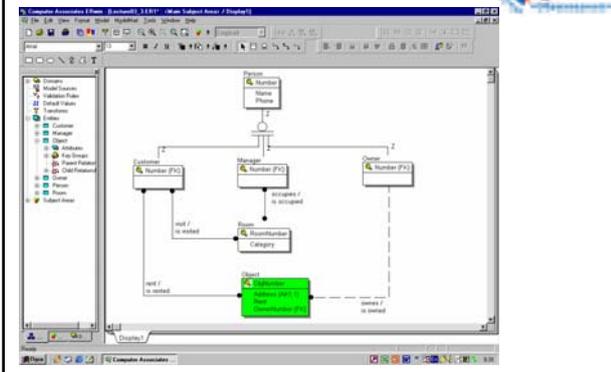
6. Специализация/генерализация



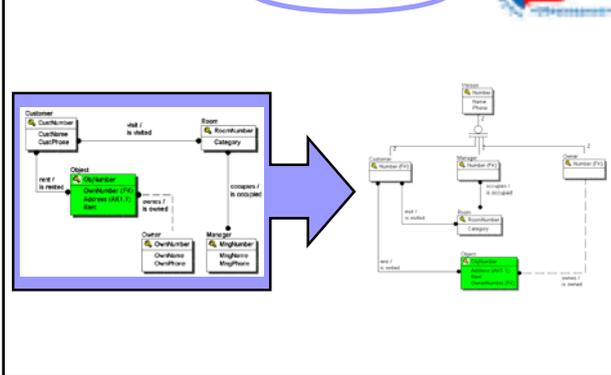
Специализация – процесс увеличения различий между отдельными членами типов сущностей за счет выделения их отличительных характеристик

Генерализация – процесс сведения различий между сущностями к минимуму путем выделения их общих характеристик

6. Специализация/генерализация



6. Специализация/генерализация



7. Обсуждение с пользователями 8. Документирование

Entity Name	Entity Attribute Name	Entity Attribute Definition	Entity Attribute Note	Entity Attribute Is PK	Entity Attribute Is FK
Customer	Number	Номер в базе данных		Yes	Yes
Manager	ObjNumber	Номер объекта недвижимости в БД		Yes	No
	Address	Адрес объекта недвижимости в БД		No	No
	Rent	Стоимость аренды в месяц		No	No
	OverNumber	Номер в базе данных		No	Yes
Over	Number	Номер в базе данных		Yes	Yes
	Name	Фамилия, Имя, Отчество	Системный атрибут	No	No
	Phone	Телефон	Множественный атрибут	No	No
Room	RoomNumber	Номер комнаты		Yes	No
	Category	Категория комнаты		No	No

Логическое проектирование



25

Процесс конструирования информационной модели (с учетом выбранного типа СУБД, но не зависимо от конкретной используемой СУБД и прочих физических условий реализации) на основе концептуальной модели.

Этапы логического проектирования



26

1. Преобразование концептуальной модели в логическую;
2. Проверка модели с помощью правил нормализации;
3. Проверка модели в отношении транзакций пользователей;
4. Определение требований поддержки целостности данных;
5. Обсуждение логических моделей данных с конечными пользователями.
6. Документирование.

1. Преобразование концептуальной модели в логическую



27

- Удаление связей типа «многие ко многим»;
- Удаление связей с атрибутами;
- Преобразование связей «суперкласс/подкласс»;
- Удаление множественных атрибутов;
- Выделение атрибутов с фиксированным набором значений;
- Перепроверка связей «один к одному»;
- Удаление избыточных связей.

2. Проверка модели с помощью правил нормализации



- Нормализация используется для улучшения модели данных [лекция 2].
- Выполнение нормализации требует полного понимания назначения каждого атрибута, присутствующего в создаваемой базе данных, что позволяет достичь существенных преимуществ.
- В результате нормализации получается весьма гибкий проект базы данных, позволяющий легко вносить в нее необходимые расширения.

Транзакция



Транзакция представляет собой действие (набор действий), выполняемых отдельным пользователем или прикладной программой с целью доступа или изменения содержимого базы данных. Транзакция является логической единицей работы, выполняемой в БД. Транзакция всегда должна переводить БД из одного согласованного состояния в другое, хотя допускается, что согласованность БД будет нарушаться в ходе выполнения транзакции.

3. Проверка модели в отношении транзакций пользователей



На данном этапе необходимо убедиться в том, что модель данных позволяет выполнить все необходимые транзакции. Если какую-либо из транзакций выполнить не удастся, значит, составленная модель является неадекватной и содержит ошибки, которые необходимо устранить.

4. Определение требований поддержки целостности данных

Ограничения целостности данных представляют собой такие ограничения, которые вводятся с целью предотвратить помещение в базу противоречивых данных.

Типы ограничений:

- Обязательные данные;
- Ограничения для доменов атрибутов;
- Целостность сущностей
- Ссылочная целостность;
- Требования бизнес-правил (правил организации)

5. Обсуждение с пользователями

6. Документирование

Entity Name	Entity Attribute Name	Entity Attribute Definition	Entity Attribute Note	Entity Attribute Is PK	Entity Attribute Is FK
Customer	Number	Номер в базе данных		Yes	Yes
Manager					
Object	ObjNumber	Номер объекта недвижимости в БД		Yes	No
	Address	Адрес объекта недвижимости в БД		No	No
	Rent	Стоимость аренды в месяц		No	No
Owner	OwnerNumber	Номер в базе данных		No	Yes
Owner Person	Number	Номер в базе данных		Yes	Yes
	Name	Фамилия, Имя, Отчество	Системный атрибут	No	No
Room	Phone	Телефон	Множественный атрибут	No	No
	RoomNumber	Номер комнаты		Yes	No
	Category	Категория комнаты		No	No

ИННОВАЦИОННО-ОБРАЗОВАТЕЛЬНАЯ ПРОГРАММА
Саратовского государственного университета им. Н.Г.Чернышевского

Принципы построения и функционирования баз данных

**Лекция четвертая.
Microsoft SQL Server**

Факультет нелинейных процессов
Физика открытых нелинейных систем
Саратов - 2008

2

О чем сегодняшняя лекция?

- Общие сведения о Microsoft SQL Server
- Объекты базы данных
- Физическая архитектура базы данных

3

Системные базы данных

- Master;
- Model;
- Tempdb;
- Msdb.

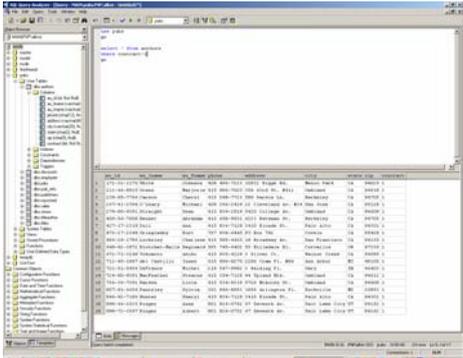
4

Администрирование может быть выполнено:

- С использованием средств Transact-SQL;
- С помощью графического интерфейса Enterprise Manager;
- С помощью мастеров (Wizards).

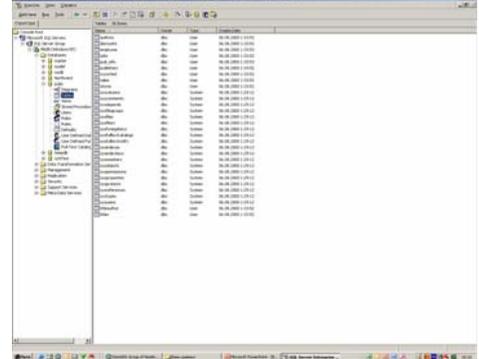
5

SQL Server Query Analyser

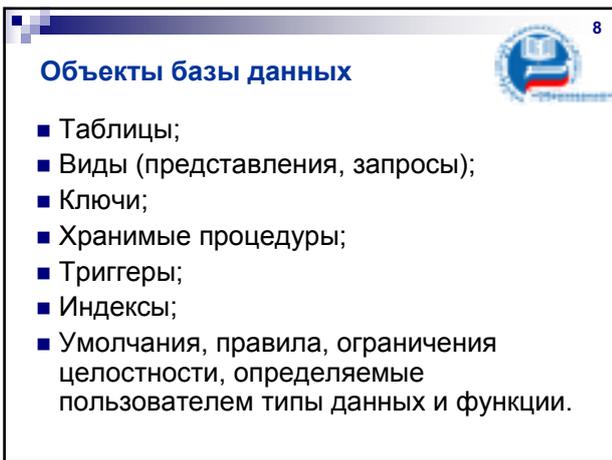


6

SQL Server Enterprise Manager









10



Представление (Вид)

- Виртуальная таблица, которая отображает данные, хранящиеся в других таблицах.
- Содержимое представления генерируется динамически на основе результата выполнения запроса.
- Основные причины использования видов:
 - Скрытие части данных от пользователей (как столбцов, так и строк)
 - Представление данных, хранящихся в разных (нормализованных) таблицах в виде одной таблицы, то есть в форме, удобной для пользователя

11



Ключи

- Используются для связывания таблиц друг с другом.
- Типы ключей:
 - Первичный ключ (Primary key)
 - Внешний ключ (Foreign key)
- Являются одним из типов ограничения целостности данных:
 - Ограничение целостности Primary key
 - Ограничение целостности Foreign key

12



Хранимая процедура

- Набор команд Transact-SQL, сохраненный специальным образом, имеющий свое имя и хранящийся на сервере.
- Основные причины использования хранимых процедур:
 - Повышение безопасности данных
 - Повышение эффективности работы системы (перенос логики обработки данных на сервер, снижение сетевого трафика)
 - Возможность более гибкого решения сложных задач

13

Триггеры



- Специальный тип хранимых процедур, автоматически запускаемых сервером при выполнении операции удаления, вставки или изменения данных в конкретной таблице.
- Использование триггеров:
 - Для проверки данных перед изменением, удалением или вставкой данных в таблицу
 - Для осуществления модификации данных в одной или нескольких таблицах, связанной с изменением, удалением или вставкой данных в таблицу

14

Триггеры - 2



Когда пользователь начинает изменение данных, сервер автоматически начинает транзакцию, в которой и выполняется триггер. В теле триггера могут быть реализованы различные алгоритмы по проверке или изменению данных. В конце работы триггера происходит либо откат, либо фиксация транзакции. При этом отменяются (фиксируются) как изменения, выполненные пользователем, так и триггером.

15

Индексы



- Объекты этого типа предназначены для повышения производительности работы сервера при поиске нужных данных в таблицах и представлениях, что достигается путем хранения в упорядоченном состоянии данных одного или более столбцов таблицы или представления. Индексы связаны с таблицами (представлениями) и не могут существовать сами по себе.

16

Умолчания, правила,



- **Умолчание** – этот тип объектов описывает значения, которые присваиваются столбцам таблицы, если при добавлении строки явно не было указано значение для соответствующего столбца. Считается морально устаревшим, рекомендуется использовать ограничение целостности **Default**.
- **Правило** – логическое условие, ограничивающее диапазон возможных значений для столбца таблицы или определяемого пользователем типа данных. Считается морально устаревшим, рекомендуется использовать ограничение целостности **Check**.

17

Ограничения целостности (constraints)



- **Специальные управляющие конструкции, ограничивающие диапазон возможных значений в столбце таблицы:**
 - Check
 - Null
 - Default
 - Unique
 - Primary Key
 - Foreign Key
 - No Action
 - Cascade

18

Определяемые пользователем типы данных и функции



- **Определяемые пользователем типы данных** – типы данных, создаваемые пользователем.
- **Определяемые пользователем функции** – функции, создаваемые пользователем, на которые можно ссылаться в теле запроса

19

Требования к именам объектов

- Уникальность;
- Максимальная длина – 128 символов;
- Не должно быть зарезервированным словом (например, Select, Insert и т.п.)
- Первый символ – символ национального или латинского алфавитов, а также символ подчеркивания «_»;
- Не допускается использование символов: пробел, (,), [,], {, }, !, %, ^, &, ~, -, .(точка), ,(запятая), \, ,

20

Ограниченные идентификаторы

Если имя объекта заключается в квадратные скобки (либо в двойные кавычки), то в имени могут использоваться любые символы, включая скобки, пробелы, специальные символы, а также зарезервированные слова. Кроме того, первым символом имени может быть любой символ.

Требование уникальности к имени объекта остается!

Select * from [3 кв. 2004 г. - отчет]

21

Доступ к объектам

Полное имя объекта записывается в виде:

[[[Server.][DataBase.][OwnerName].]ObjectName

При необходимости любой из элементов в иерархии полного имени объекта может быть указан явно:

- Server.DataBase.OwnerName.ObjectName
- Server.DataBase..ObjectName
- DataBase.OwnerName.ObjectName
- DataBase..ObjectName
- OwnerName.ObjectName
- ObjectName

22

**Физическая архитектура
базы данных**



- В SQL Server 2000 существуют два типа файлов БД:
 - Файлы данных (data files). Предназначены для хранения информации, находящейся в таблицах БД. В этих же файлах также размещены процедуры, ограничения, триггеры, индексы и другая информация.
 - Файлы журнала транзакций (transaction log files). В них содержится информация о ходе выполнения транзакций.

23

Повышение производительности



- Данные, которые требуют много места для хранения, рекомендуется размещать в отдельных файлах.
- Если сервер имеет много физических дисков, рекомендуется для каждой БД создавать как минимум один файл на каждом физическом диске.
- По возможности следует располагать файлы данных и журнала транзакций на отдельных физических дисках.

24

Файлы данных



- Основной или главный файл (Primary File). Предназначен для хранения всей системной информации о БД (структура БД, системные таблицы, созданные объекты и проч.). В нем могут также храниться пользовательские данные. Имеет по умолчанию расширение mdf (Master Data File). Может существовать только один главный файл. Если БД содержит только один файл данных, то он и будет основным.

Файлы данных – 2



- Вторичный или дополнительный файл (Secondary File). В дополнительных файлах хранится только пользовательская информация. БД может содержать множество дополнительных файлов или не содержать их совсем. По умолчанию для вторичных файлов устанавливается расширение ndf (secoNdary Data File).

Файлы данных – 3



- Файл журнала транзакций (Transaction Log File). В БД должен быть как минимум один такой файл. Для ускорения обработки транзакций можно использовать несколько журналов транзакций, расположенных на разных физических дисках. По умолчанию имеет расширение ldf (Log Data File).

Имена файлов БД



- Физическое имя (OS File Name) – имя файла на диске.
- Логическое имя (Logical File Name) – имя, которое будет применяться в SQL Server для ссылки на соответствующий файл. Это имя можно рассматривать как псевдоним.

Имена обоих типов могут совпадать, но могут быть и различными. По умолчанию SQL Server предлагает совпадающие имена. Во избежание путаницы не рекомендуется изменять эти значения.

Группы файлов



- Основная группа файлов (Primary File Group) – содержит основной файл БД и, как следствие, включает в себя все системные данные. Поскольку основной файл БД только один, то и основная группа может быть только одна. В эту группу автоматически включаются все файлы, явно не приписанные ни к какой иной группе. Основная группа может состоять даже только из одного основного файла и всегда существует в базе данных.

Группы файлов – 2



- Пользовательская группа файлов (User File Group). Служит для объединения дополнительных файлов БД. Состав файлов той или иной пользовательской группы зависит от ее назначения. Может быть одна или несколько (а может не быть ни одной) пользовательских групп.

Группы файлов – 3

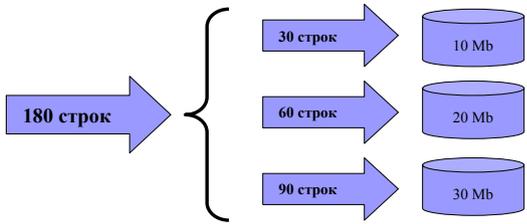


- Группа файлов по умолчанию (Default File Group). Фактически, это маркер, присваиваемый одной из существующих в БД групп. Основное назначение этой группы – хранение данных, явно не приписанных ни к одной из групп. Существует только одна группа файлов по умолчанию. Сразу после создания БД в качестве такой группы назначается основная группа файлов.

Группы файлов – 4



SQL Server 2000 равномерно распределяет данные между всеми файлами в группе.



ИННОВАЦИОННО-ОБРАЗОВАТЕЛЬНАЯ ПРОГРАММА
 Саратовского государственного университета им. Н.Г.Чернышевского

Принципы построения и функционирования баз данных

**Лекция пятая.
 Индексирование баз данных**

Факультет нелинейных процессов
 Физика открытых нелинейных систем
 Саратов - 2008

О чем сегодняшняя лекция

- Индексирование базы данных
 - Типы индексов
 - Фактор заполнения
 - Управление индексами (создание, перестроение, удаление индексов)
- Транзакции
 - Требования ACID к выполнению транзакций
 - Методы определения транзакций
 - Распределенные транзакции
 - Вложенные транзакции

Типы индексов

- Некластерный индекс (Nonclustered Index);
- Кластерный индекс (Clustered Index);
- Уникальный индекс (Unique Index);

4

Некластерный индекс



- **Некластерный индекс** представляет собой набор всех значений индексируемого столбца, упорядоченных по возрастанию или убыванию и указатель на исходную строку таблицы
- **Указатель на строку** включает следующую информацию:
 - Идентификационный номер файла (ID file)
 - Идентификационный номер страницы (ID page)
 - Номер слота (slot number) строки на странице

5

Некластерный индекс



- **Некластерный индекс** рекомендуется создавать в следующих ситуациях:
 - Столбец содержит большое количество уникальных значений, например, комбинацию имени и отчества
 - Запрос не должен возвращать большой набор данных
 - Столбец часто включается в запрос в качестве условия поиска в разделе WHERE или HAVING.
 - Длина столбцов не превышает 10 байт
 - Рекомендуется применять индексы для редко изменяемых столбцов

6

Кластерный индекс



- При создании **кластерного индекса** происходит физическое перестроение порядка строк в таблице
- Некластерные индексы обычно создаются в качестве дополнительных индексов, тогда как в качестве основного используется кластерный индекс. Рекомендуется применять кластерный индекс к данным, которые наиболее часто выступают в качестве критериев поиска.

7

Кластерный индекс



- **Кластерный индекс** рекомендуется создавать в следующих ситуациях:
 - Столбец содержит большое количество уникальных значений, например, комбинацию имени и отчества.
 - Запрос не должен возвращать большой набор данных. Если в запросе присутствуют операторы BETWEEN, >, >=, <=, или <, то кластерный индекс может заметно увеличить производительность.
 - Столбец часто применяется при группировке (раздел GROUP BY) или для слияния (раздел JOIN), особенно в качестве внешнего ключа (FOREIGN KEY).
 - Когда столбец используется системами оперативной обработки транзакций (OLTP).

8

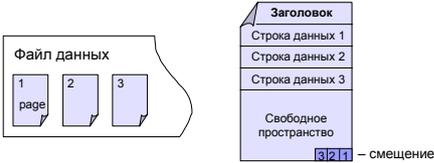
Уникальный индекс



- **Уникальный индекс** предназначен для обеспечения уникальности значений соответствующего индекса
- Уникальный индекс не существует как самостоятельный физический тип индекса. То есть нельзя создать просто уникальный индекс – этот индекс всегда будет либо кластерным, либо некластерным.
- Можно сказать, что уникальный индекс представляет собой ограничения целостности UNIQUE, примененное к данным индекса.

9

Фактор заполнения



Файл данных

1	2	3
page		

Заголовок
Строка данных 1
Строка данных 2
Строка данных 3
Свободное пространство
3 2 1 – смещение

- **Фактор заполнения (fill factor)** – это параметр, в процентах определяющий плотность записи данных на странице. Фактор заполнения указывается при создании индекса.

10

Управление индексами. Создание индексов



Ситуации, когда происходит создание индекса:

- При определении в таблице первичного ключа. В этом случае сервер автоматически создает уникальный индекс. Этот индекс будет кластерным, если в таблице уже не существует кластерного индекса. Однако пользователь может явно указать, что для первичного ключа должен быть создан некластерный индекс.
- Уникальный индекс также автоматически создается при определении в таблице ограничений целостности UNIQUE.

11

Управление индексами. Создание индексов



Ситуации, когда происходит создание индекса:

- Индекс любого типа может быть создан в процессе создания таблицы. В команде CREATE TABLE предусмотрены средства, позволяющие выполнить организацию индекса, как часть процесса создания таблицы. При этом разрешается создание индекса как для одного, так и для нескольких столбцов.
- Индекс может быть создан уже после создания таблицы с помощью команды CREATE INDEX, специально предназначенной для этого.

12

Создание индекса при создании таблицы



```
CREATE TABLE
  [database_name.[owner] | owner] table_name
  ({ <column_definition> | column_name AS
  computed_column_expression |
  <table_constraint> } [,..n])
  [ON {filegroup | DEFAULT}]
  [TEXTIMAGE_ON {filegroup | DEFAULT}]
```

Создание индекса при создании таблицы



- Пример кода, описывающий создание таблицы с ограничением целостности Primary Key на уровне одного столбца:

```
CREATE TABLE jobs
    (job_id smallint IDENTITY(1,1)) PRIMARY
    KEY CLUSTERED, job_desc varchar(50)
    NOT NULL DEFAULT 'New Position'
```

Методы создания индексов с Transact-SQL



```
CREATE [ UNIQUE ] [ CLUSTERED | NONCLUSTERED ]
INDEX index_name ON { table | view }
    ( column [ASC | DESC] [ ,...n ] )
    [ WITH <index_option> [ ,...n ] ]
    [ ON filegroup ]
```

```
<index_option> ::= =
{ PAD_INDEX | FILLFACTOR = fillfactor |
  IGNORE_DUP_KEY | DROP_EXISTING |
  STATISTICS_NORECOMPUTE | SORT_IN_TEMPDB }
```

Методы создания индексов с Transact-SQL



- Пример кода, позволяющего организовать для таблицы *Categories* базы данных *Northwind* кластерный индекс *PK_Categories* на основе столбца *CategoryID*:

```
USE NORTHWIND
GO
CREATE UNIQUE CLUSTERED
INDEX [PK_Categories] ON
[dbo].[Categories] ([CategoryID])
WITH DROP_EXISTING
ON [PRIMARY]
```

16

Создание индексов с помощью Enterprise Manager

The screenshot shows two dialog boxes in SQL Enterprise Manager. The 'Manage Indexes' dialog is on the left, showing a table 'Categories' with an existing index 'PK_Categories'. The 'Create New Index' dialog is on the right, showing the same table and index options. The 'Index name' is 'MyIndex', and the 'Columns' list includes 'CategoryID', 'Description', and 'Picture'. The 'Index options' section is checked for 'Unique values', 'Fill factor', and 'Do not recompute statistics'.

17

Использование мастера Create Index Wizard

The screenshot shows two steps of the 'Create Index Wizard'. The first dialog, 'Select Database and Table', shows 'Northwind' as the database and 'Categories' as the object name. The second dialog, 'Current Index Information', shows a table with columns 'Index Name', 'Clustered', and 'Columns'. The row for 'PK_Categories' is highlighted, showing it is a clustered index on the 'CategoryID' column.

18

Перестроение индексов

- **Перестроение индекса** – это процесс перераспределения строк индексных страниц, чтобы установить фактор заполнения на нужный уровень и тем самым повысить производительность.
- Для выполнения перестроения индексов предназначена команда
DBCC DBREINDEX

19

Переименование индекса

- Для переименования индекса служит хранимая процедура `sp_rename`, имеющая синтаксис:

```
sp_rename [@objname =] 'object_name',  
[@newname =] 'new_name', 'INDEX'
```

20

Удаление индекса

- Для удаления индекса используется команда `DROP INDEX`, имеющая синтаксис:

```
DROP INDEX 'table.index' [,...n]
```

- При удалении индекса происходит освобождение всех страниц, которые использовались для хранения данных индекса.

21

Транзакции

- Транзакция представляет собой набор из одной или более команд, обрабатываемых как единое целое. То есть будет выполнен либо весь набор, либо не выполнена ни одна из команд.
- Транзакции:
 - Стандартные
 - Распределенные

22

Транзакции



- Требования к выполнению транзакций СУБД
 - Атомарность (Atomicity)
 - Согласованность (Consistency)
 - Изолированность (Isolation)
 - Устойчивость или долговечность (Durability)

23

Определение транзакций



- В SQL Server 2000 предусмотрено несколько методов определения транзакций:
 - автоматический
 - явный
 - подразумеваемый или неявный
- По умолчанию SQL Server 2000 работает в режиме автоматического определения транзакции. Каждая команда обрабатывается как отдельная транзакция.

24

Определение транзакций

Явное определение транзакции



- Команды Transact-SQL, с помощью которых можно воздействовать на поведение транзакций:
 - BEGIN TRAN – для обозначения начала транзакции
 - SAVE TRAN – для создания точки сохранения
 - ROLLBACK TRAN, ROLLBACK WORK – откат транзакции или восстановление точки сохранения
 - COMMIT TRAN – выполняет фиксирование транзакции

25

Определение транзакций



Неявное определение транзакции

- Пользователь не должен явно указывать начало транзакции с помощью команды BEGIN TRAN, система автоматически начинает новую транзакцию после того, как будет завершена предыдущая. При этом после выполнения очередной команды не происходит автоматического фиксирования транзакции (в отличие от режима автоматического определения). Пользователь должен явно выполнить откат или фиксирование транзакции с помощью команд ROLLBACK TRAN или COMMIT TRAN.

26

Определение транзакций



Неявное определение транзакции

- Для переключения SQL Server 2000 в режим неявного начала транзакции существует команда
SET IMPLICIT_TRANSACTION ON
- Для переключения SQL Server 2000 обратно в режим автоматического определения транзакции необходимо использовать команду
SET IMPLICIT_TRANSACTION OFF

27

Распределенные транзакции



- **Распределенные транзакции** представляют собой совокупность двух или более локальных транзакций, выполняемых одновременно в различных базах данных.
- *Координатор распределенных транзакций* (MSDTC, Microsoft Distributed Transaction Coordinator) контролирует всю работу по инициализации, откату и фиксированию локальных транзакций.

Вложенные транзакции



```
BEGIN TRAN TRAN_1
  BEGIN TRAN TRAN_2
    .....
    BEGIN TRAN TRAN_3
      .....
      COMMIT TRAN TRAN_3
    .....
  COMMIT TRAN TRAN_2
  .....
  BEGIN TRAN TRAN_4
    .....
  COMMIT TRAN TRAN_4
COMMIT TRAN TRAN_1
```

ИННОВАЦИОННО-ОБРАЗОВАТЕЛЬНАЯ ПРОГРАММА
 Саратовского государственного университета им. Н.Г.Чернышевского

Принципы построения и функционирования баз данных

Лекция шестая.
 Возможные неприятности и пути их преодоления

Факультет нелинейных процессов
 Физика открытых нелинейных систем
 Саратов - 2008

2

О чем сегодняшняя лекция?

- О возможных проблемах при работе с базами данных и путях их решения
- Работа с пользователями и правами доступа
- Некомпьютерные методы защиты
- Создание отказоустойчивой системы

3

«Какие такие неприятности? И зачем они меня ждут?»



«Котенок по имени Гав»

4

Возможные неприятности



- Неумышленное удаление или изменение данных пользователем, приведшее к утере, искажению информации или нарушению целостности данных
- Злонамеренные действия (кража, удаление, искажение информации и др.)
- Сбои в аппаратной части компьютера
- Чрезвычайные происшествия (пожар, взрыв, и т.д.)
- Стихийные бедствия (землетрясения, ураганы, и проч.)

5

Система безопасности



Необходимо грамотно спланировать права доступа к данным. Пользователь должен иметь доступ только к той информации, которая ему необходима в соответствии с его обязанностями.

Система безопасности условно может быть разделена на два уровня:

- Уровень сервера;
- Уровень базы данных.

6

Идентификация и аутентификация



- **Идентификация** – сообщение системе, кем является пользователь, пытающийся получить доступ.
- **Аутентификация** – это проверка подлинности пользователя.

7



Аутентификация

- SQL Server 2000 поддерживает два метода аутентификации:
 - Средствами Windows NT
 - Средствами SQL Server 2000
- Система безопасности SQL Server может работать в одном из двух режимов:
 - Режим смешанной аутентификации (Mixed Mode)
 - Режим аутентификации Windows (Windows Authentication Mode)

8



Роли

Роли предназначены для группировки пользователей и предоставления им необходимых прав, как по администрированию сервера, так и по доступу к данным.

- Роли сервера;
- Роли базы данных:
 - Фиксированные роли базы данных;
 - Пользовательские роли базы данных;
 - Роли приложений;

9



Роли сервера

- Набор ролей сервера (fixed server roles) строго ограничен, невозможно создать или удалить существующую роль
- Права управления сервером можно распределить между разными пользователями. Каждый такой пользователь имеет возможность выполнять только определенный круг задач

10



Роли сервера – 2

- **System Administrators** – абсолютные права
- **Setup Administrators** – права управления связанными серверами, конфигурирование хранимых процедур, запускаемых при старте SQL Server 2000
- **Server Administrators** – администрирование сервера: остановка, изменение параметров работы служб и т.п.
- **Security Administrators** – создание новых учетных записей, в том числе с правами на создание БД и ее объектов
- **Process Administrators** – управление процессами
- **Disk Administrators** – для совместимости с версией SQL Server 7
- **Database Creators** – могут создавать (удалять, переименовывать, восстанавливать) БД
- **Bulk Insert Administrators** – права на вставку данных с помощью средств массовой загрузки

11



Фиксированные роли БД

Фиксированные роли делают возможным предоставление пользователям набора прав, которые нельзя организовать никаким другим образом. Количество и назначение таких ролей стандартно и не может быть изменено.

12



Фиксированные роли БД – 2

- **db_securityadmin** – управление правами доступа и членством в группах
- **db_owner** – права владельца, могут выполнять любые действия
- **db_denydatawriter** – запрещено изменение данных
- **db_denydatareader** – запрещен просмотр данных
- **db_ddladmin** – права создавать, изменять, удалять объекты БД
- **db_datawriter** – права на изменение данных в любой таблице
- **db_datareader** – права на чтение из любой таблицы
- **db_backupoperator** – права на создание резервной копии
- **db_accessadmin** – права управления пользователями БД

13

Пользовательские роли БД



Пользовательские роли базы данных служат для группировки пользователей с целью облегчения управления их правами доступа к объектам базы данных. Когда в базе данных существуют пользователи, которым необходимы одинаковые права доступа, то их лучше объединить в одну административную единицу.

14

Роли приложения



Роли приложения позволяют выдавать права доступа не конкретному пользователю (или их группе), а приложению в целом. При этом не важно, какой именно пользователь работает с приложением, и какие у него права доступа. Получив доступ к приложению, пользователь может выполнять все действия, разрешенные роли приложения.

15

Права доступа



- Права доступа к данным
- Права на выполнение хранимых процедур
- Права на выполнение команд Transact-SQL, предназначенных для создания новых объектов базы данных

Права доступа подразделяют на явные и неявные. Неявные права доступа представляются пользователю через членство в фиксированной роли сервера или базы данных. Еще одним примером неявных прав доступа является получение владельцем объекта абсолютных прав на управление этим объектом.

16

Права доступа – 2

- Для доступа к тому или иному объекту данных пользователю необходимо предоставить доступ (GRANT)
- Запрещение доступа (DENY) имеет более высокий приоритет, чем предоставление
- Неявное отклонение доступа (REVOKE) является отменой ранее выданных разрешений (как GRANT, так и DENY).

17

Иерархия прав доступа

При обращении пользователя к объектам БД, ссылающихся на другие объекты, выполняется проверка прав доступа не только для объекта, к которому непосредственно обращается пользователь, но и к объектам, на которые этот объект ссылается. Проверка не выполняется, если владельцем зависимых объектов является тот же пользователь, владеющий основным объектом, к которому происходит обращение

18

Иерархия прав доступа: пример

Пользователь John предоставляет пользователю David права доступа к виду View1, при этом пользователь David не имеет прав доступа к Table1

```

graph TD
    A[John.View1] --> B[John.Table1]
    
```

✓

```

graph TD
    A[John.View1] --> B[Martin.Table1]
    
```

✗

19

**Система безопасности –
некомпьютерные средства
контроля**



- Разработка мер безопасности и защиты от непредвиденных обстоятельств;
- Контроль за персоналом и физическим доступом;
- Защита помещений и хранилищ;

20

Меры обеспечения безопасности



Должны быть регламентированы:

- Область деловых процессов, для которых устанавливаются меры безопасности;
- Ответственность и обязанности отдельных работников;
- Дисциплинарные меры, принимаемые в случае обнаружения нарушения установленных ограничений;
- Процедуры, которые должны обязательно выполняться

21

**Типичный план защиты от
непредвиденных обстоятельств**



- Сведения о главном ответственном лице и как с ним можно связаться;
- Кто и на каком основании принимает решение о том, что возникла необычная ситуация;
- Технические требования к передаче управления резервным службам
- Сведения о внешних структурах, в которых можно получить помощь

Создание отказоустойчивой системы



22

В SQL Server существует несколько технологий, позволяющих обеспечить работоспособность системы в случае отказа:

- Резервный сервер
- Кластер
- Технология RAID
- Резервное копирование

Резервный сервер



23

Резервный сервер представляет собой специальный выделенный сервер, который дублирует функции основного сервера. Он работает отдельно от основного сервера, имеет свое собственное сетевое имя и сетевой адрес. Резервный сервер содержит те же БД, что и основной. С основного сервера периодически копируются все изменения. При выходе из строя основного сервера пользователи должны вручную переключиться на использование резервного, который должен быть переведен в нормальный режим.

Недостатки резервного сервера



24

- Администратор должен вручную синхронизовать основной и резервный сервера, причем, чтобы свести ущерб от выхода из строя первичного сервера, это необходимо делать как можно чаще.
- При выходе из строя основного сервера нужно вручную переключить клиентов на использование резервного сервера и запустить заново все задания, выполнявшиеся на основном сервере.

25

Кластер

Главный сервер Вторичный сервер

SQL сервер на общем диске

Виртуальный сервер

Клиент

Серверы кластера (узлы), использующие общий диск, объединяются в виртуальный сервер, который и видят пользователи.

26

RAID массивы

RAID 0 – чередующийся набор дисков, представляющий собой совокупность множества физических дисков (подключенных к разным контроллерам), объединенных в один виртуальный том. Данные распределяются между всеми физическими дисками, что дает значительное повышение производительности. Нет возможности восстановления данных в случае повреждения одного из дисков.

27

RAID массивы – 2

RAID 1 – два жестких диска, подключенных к одному контроллеру (зеркальное отображение дисков) или к разным контроллерам (дублирование дисков). Обеспечивают наиболее высокую степень защиты данных среди всех уровней массивов RAID. Недостаток – двойной расход дискового пространства.

28



RAID массивы – 3

RAID 5 – чередующийся набор дисков с контролем четности (используется операция XOR). Помимо самих данных сохраняют информацию, с помощью которой можно восстановить данные в случае полного или частичного повреждения одного из дисков. Наряду с высокой производительностью обеспечивают также высокую степень надежности.

	0	1
0	0	1
1	1	0

Потери дискового пространства:
3 диска – 33%
10 дисков – 10%

29



Резервное копирование

Один из самых надежных способов защиты данных от потери. При потере данных резервная копия переносится в систему, и пользователи могут снова работать с информацией. Система при этом будет восстановлена в том состоянии, в котором она была во время создания резервной копии.

30



Типы резервного копирования

- Полная копия
- Разностная копия
- Копия журнала транзакций
- Резервное копирование файлов и групп

31



Полная копия

- Полное копирование всей информации базы данных
- Для восстановления работоспособности системы необходимо восстановить лишь один архив
- Длительное время создания архива
- Отслеживание изменений в уже скопированных данных

32



Разностное (дифференциальное) копирование

- Состоит из двух этапов:
 1. Создание полной копии
 2. Создание дифференциальной копии
- Разностное копирование требует меньше времени по сравнению с полным.
- При последовательном создании нескольких дифференциальных копий каждая следующая будет полностью включать ВСЕ измененные страницы.

33



Копия журнала транзакций

- Позволяет восстановить систему в состоянии, в котором она была в любой момент времени.
- При копировании автоматически выполняется усечение журнала транзакций. Ни пользователь, ни SQL Server не должны выполнять усечение журнала транзакций.
- При восстановлении необходимо последовательно применять ВСЕ копии журнала транзакций

34

Возможная стратегия для резервного копирования больших баз данных



1. Создается полная резервная копия, обычно раз в неделю, на выходных.
2. Создается разностная копия, например, каждую ночь.
3. Создается копия журнала транзакций, например, раз в несколько часов.

35

Резервное копирование файлов и групп файлов.



Позволяет выполнить резервное копирование части данных, вплоть до конкретного столбца таблицы. В основе этого подхода лежит возможность привязывания таблицы или отдельного столбца к конкретному файлу или группе файлов.

Структура итогового теста

№	Тема	Раздел	Количество вопросов
1	Введение. Основные сведения	Основные сведения и терминология	1
		Архитектура многопользовательских СУБД	1
2	Реляционная модель данных.	Основные сведения и терминология	2
		1NF и 2NF	1
		3NF, NFBK, 4NF и 5NF	1
		Нормализация	1
3	Основы проектирования реляционных баз данных.	Основные сведения и терминология	1
		Концептуальное проектирование	1
		Логическое проектирование	1
4	Microsoft SQL Server	Объекты Microsoft SQL Server	1
		Физическая архитектура базы данных	1
5	Работа с объектами SQL Server	CREATE TABLE, DROP TABLE, ALTER TABLE	1
		INSERT, DELETE, UPDATE	1
		системные хранимые процедуры	1
6	Основы T-SQL	Простая выборка данных	2
		SELECT с функциями агрегирования, группировки	1
		Объединение таблиц и подзапросы	2
		Представления	1
		Хранимые процедуры	1
7	Индексирование баз данных		1
8	Возможные неприятности и пути их преодоления.	Способы защиты от возможных неприятностей	1
		Резервное копирование	1

ОБРАЗЦЫ ВОПРОСОВ ИТОГОВОГО ТЕСТА

Данный материал содержит образцы вопросов итогового теста и призван (наряду со структурой итогового теста, приведенной в настоящих учебно-методических материалах) помочь студентам составить представление о форме итогового контроля по данному учебному курсу, осуществляемого в конце семестра. Этот материал может рассматриваться студентом как пробный тест, пройти который ему будет предоставлена возможность во время консультации перед экзаменом.

Пробный тест по дисциплине «Высокоуровневые методы программирования и информационные системы. Принципы построения и функционирования баз данных» включает в себя 5 вопросов (в отличие от 25 вопросов экзаменационного теста) из различных разделов курса. На прохождение теста отводится 6 минут (для сравнения: на прохождение экзаменационного теста отводится 30 минут). Среди вариантов ответов всегда содержится один полный и правильный, который оценивается в 2 балла, может содержаться (или не содержаться) один или два отчасти правильных (или правильных, но не полных), которые оцениваются в 1 балл, остальные - неправильные.

1. Отношение, в котором ни одно поле строки не содержит более одного значения, любое ключевое поле не пусто, а первичный ключ состоит только из одного столбца, является ...
 - а) универсальным отношением;
 - б) ненормализованной формой;
 - в) первой нормальной формой;
 - г) второй нормальной формой;
 - д) третьей нормальной формой;
 - е) четвертой нормальной формой;
 - ж) пятой нормальной формой;
 - з) нормальной формой Бойса-Кодда.

2. Имеется выражение для создания таблицы **Books** в базе данных Microsoft SQL Server.

```
CREATE TABLE Books  
(BookID numeric(6,0) PRIMARY KEY,  
Title varchar(100) NOT NULL,  
Author varchar(200) NULL,  
Publisher varchar(200),  
ISBN varchar(40) NULL,  
Pub_year numeric(4,0) CHECK (Pub_year>1900) NOT NULL)
```

Что будет выведено в результате выполнения этого выражения?

- а) запрос не выполнится, т.к. не указано, что первичный ключ должен принимать ненулевые значения;
- б) таблица Books будет успешно создана;
- в) запрос выполнится, но первичный ключ создан не будет;
- г) запрос не выполнится, т.к. ключевое слово CHECK не допустимо в Microsoft SQL Server.

3. Какого из нижеперечисленных объектов НЕ существует в Microsoft SQL Server?
- таблица;
 - вид;
 - храняемая процедура;
 - триггер;
 - конвертер;
 - индекс;
 - умолчание.
4. Процедура конструирования информационной модели, не зависящей от каких-либо физических условий реализации, таких как выбранный тип СУБД, состав программ приложения, используемый язык программирования, операционная система и другие физические особенности, называется ...
- специализацией базы данных;
 - генерализацией базы данных;
 - реализацией базы данных;
 - созданием прототипа;
 - планированием разработки базы данных;
 - физическим проектированием;
 - концептуальным проектированием;
 - логическим проектированием.
5. В базе данных **world** имеется таблица **city**, содержащая информацию о городах мира. Таблица **city** включает в себя следующие атрибуты:
- ID - идентификатор города, первичный ключ;
 - Name - название города;
 - CountryCode - код города;
 - DistrictID - идентификатор района;
 - Population - население города.
- Чтобы выбрать названия 100 наименьших городов мира в порядке возрастания численности населения, нужно подать следующий запрос:
- SELECT TOP 100 name FROM city ORDER BY population DESC;
 - SELECT TOP 100 name FROM city ORDER BY population ASC;
 - SELECT TOP 100 name FROM city ORDER BY population;
 - SELECT TOP 100 name, population FROM city ORDER BY population DESC;
 - SELECT TOP 100 name, population FROM city ORDER BY population;
 - среди представленных вариантов ответа имеется несколько правильных.

Рекомендуемая литература

1. *Гарсиа М.Ф. и др.* Microsoft SQL Server. Справочник администратора. ЭКОМ: Москва, 2002.
2. *Хоторн Р.* Разработка баз данных Microsoft SQL Server 2000 на примерах. М.: Издат. дом «Вильямс», 2001.
3. *Баженова И.Ю.* Основы проектирования приложений баз данных. Изд-во: Интернет-университет информационных технологий. 2006.
4. *Кузнецов С.Д.* Основы баз данных. Изд-во: Интернет-университет информационных технологий. 2005.
5. *Уилтон П., Колби Дж.* Язык запросов SQL для начинающих. М.: Диалектика, 2006.
6. *Шпенник М., Слендж О.* Руководство администратора баз данных Microsoft SQL Server 2000. М.: Издат. дом «Вильямс», 2001.
7. *Мейер М.* Теория реляционных баз данных. - М.: Мир, 1987. - 608 с.
8. *Мартин Дж.* Организация баз данных в вычислительных системах. М.: Мир, 1980. - 662 с.
9. *Мартин Дж.* Планирование развития автоматизированных систем. М.: Финансы и статистика, 1984. - 196 с.
10. *Ульман Дж.* Основы систем баз данных. М.: Финансы и статистика, 1983. - 334 с.
11. *Тиори Т., Фрай Дж.* Проектирование структур баз данных: В 2-х кн. М.: Мир, 1985. - Кн. 1. - 287 с.; Кн. 2. - 320 с.
12. *Хаббард Дж.* Автоматизированное проектирование баз данных. М.: Мир, 1984. - 294 с.
13. *Дейт К.Дж.* Введение в системы баз данных. Киев: Диалектика, 1998. - 784 с.
14. *Бойко В.В., Савинков В.М.* Проектирование баз данных информационных систем. - М.: Финансы и статистика, 1989. - 351 с.
15. *Кириллов В.В.* Основы проектирования реляционных баз данных: Учеб. пособие / Санкт-Петербургский государственный институт точной механики и оптики - СПб., 1994 - 88 с.
16. *Гринь А.М.* Логическое проектирование и реализация баз данных: Учеб. пособие. - Новосибирск: Изд-во НГТУ, 1994.
17. *Грабер М.* Введение в SQL. - М.: ЛОРИ, 1996. - 380 с.
18. *Стасьшин В.М.* Язык структурных запросов SQL: Учеб. пособие. - Новосибирск: Изд-во НГТУ, 1996. - 33 с.
19. *Конноли Т., Бегг К., Страчан А.* Базы данных. Проектирование, реализация и сопровождение. Теория и практика. М.: Издательский дом «Вильямс», 2000. - 1120 с.