

Министерство образования и науки РФ
Амурский государственный университет
(ГОУВПО «АмГУ»)

УТВЕРЖДАЮ
Зав. кафедрой ИиУС
_____ А.В. Бушманов
«__» _____ 2010 г.

Учебно-методический комплекс дисциплины

Проектирование баз данных

для направления подготовки 230100.68

Информатика и вычислительная техника

Составитель: Чепак Л.В.

2010 г.

*Печатается по решению
редакционно-издательского совета
факультета математики и информатики
Амурского государственного
университета*

Проектирование баз данных для направления подготовки 230100.68 «Информатика и вычислительная техника»: учебно-методический комплекс дисциплины. / Чепак Л.В. – Благовещенск. Изд-во Амурского гос. ун-та, 2010. – 52 с.

Учебно-методическое пособие содержит: рабочую программу преподавания дисциплины; методические указания и учебные задания для выполнения курса лабораторных работ.

©Амурский государственный университет, 2010

©Кафедра информационных и управляющих систем, 2010

1. РАБОЧАЯ ПРОГРАММА

По дисциплине: Проектирование баз данных
Для направления 230100.68 – Информатика и вычислительная техника
подготовки магистра:

Курс: 6 Семестр: С

Лекции: 18 (час.) Экзамен: нет

Лабораторные занятия: 34 (час.) Зачет: С семестр

Самостоятельная работа: 71 (час.)

Всего часов: 130 (час.)

Составитель: Чепак Л.В.

Факультет: «Математики и информатики»

Кафедра: «Информационные и управляющие системы»

2010 г.

1. ЦЕЛИ И ЗАДАЧИ ДИСЦИПЛИНЫ, ЕЕ МЕСТО В УЧЕБНОМ ПРОЦЕССЕ

1.1. Цели и задачи дисциплины

Дисциплина «Проектирование баз данных» занимает важное место в системе образования в области современных информационных технологий. Целью преподавания дисциплины является изучение: этапов жизненного цикла информационной системы, в состав которой входит база данных; языков определения и манипулирования данными, основных архитектур доступа к базам данных.

Целью дисциплины является изучение и практическое освоение методов проектирования баз данных.

1.2. Требования к уровню освоения содержания дисциплины

В результате изучения курса студенты должны

знать:

- последовательность и содержание этапов проектирования баз данных;
- принципы архитектуры доступа к базам данных;
- основные конструкции языка манипулирования данными SQL;
- средства управления транзакциями;
- современные методы и средства защиты баз данных;

уметь:

- применять современную методологию для анализа требований к системе;
- иметь навык описания информационных потребностей пользователей;
- применять средства разработки схем баз данных;
- применять современные методы разработки приложений баз данных;
- реализовывать проекты баз данных;
- управлять доступом к базе данных;

иметь представление:

– о тенденциях и перспективах развития современных систем управления базами данных.

1.3. Перечень дисциплин с указанием разделов (тем), усвоение которых студентами необходимо при изучении данной дисциплины

Изучение данной дисциплины требует от студентов предварительного усвоения таких дисциплин как «Информатика», «Дискретная математика», «Алгоритмические языки и программирование», «Информационные технологии», «Системное программное обеспечение», «Базы данных» в объеме государственного образовательного стандарта высшего профессионального образования.

2. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

2.1. Федеральный компонент

Дисциплина «Проектирование баз данных» является дисциплиной, входящей в блок вариативных дисциплин для магистратуры по направлению 230100.68 – «Информатика и вычислительная техника». Тематическое содержание курса государственным стандартом не ограничено.

2.2. Наименование тем, их содержание, объем в лекционных часах

ТЕМАТИЧЕСКИЙ ПЛАН ЛЕКЦИОННЫХ ЗАНЯТИЙ

Наименование темы	Кол-во часов
1. Основы проектирования баз данных и анализ требований к системе	2
2. Средства разработки схемы и приложений базы данных.	4
3. Язык запросов SQL.	6

4. Архитектура доступа к базам данных.	2
5. Управление доступом к базе данных.	2
6. Управление транзакциями и защита баз данных.	2
ИТОГО	18

Тема 1. Основы проектирования баз данных и анализ требований к системе

База данных – основной компонент автоматизированной информационной системы. Модели проектирования систем на основе баз данных: общая схема проектирования, каскадная модель проектирования, спиральная модель проектирования. Описание информационных потребностей пользователей. Диаграммы потоков данных. Методологии IDEF0 и IDEF3.

Тема 2. Средства разработки схемы и приложений базы данных.

Схема базы данных. Проектирование логической и физической схемы базы данных с помощью пакета ERWin. Современные средства разработки приложений баз данных. Создание приложений. Создание подключения к базе данных.

Тема 3. Язык запросов SQL.

Язык структурированных запросов SQL. Основные конструкции языка. Оператор SELECT. Типы данных. Операторы определения данных. Операторы SQL манипулирования данными.

Тема 4. Архитектура доступа к базам данных.

Архитектура файлового сервера. Двухуровневая архитектура клиент-сервер. Трехуровневая архитектура клиент-сервер. Архитектура клиент-сервер в Intranet/Internet.

Тема 5. Управление доступом к базе данных.

Операторы SQL управления доступом к данным. Идентификаторы поль-

зователей и права владения. Предоставление и отмена привилегий.

Тема 6. Управление транзакциями и защита баз данных.

Поддержка транзакций. Управление параллельностью. Монитор транзакций. Защита баз данных. Типы опасностей. Компьютерные и некомпьютерные средства контроля.

2.3. Лабораторные занятия, их содержание и объем в часах.

ТЕМАТИЧЕСКИЙ ПЛАН ЛАБОРАТОРНЫХ ЗАНЯТИЙ

Наименование темы	Кол-во часов
1. Создание логической схемы с помощью пакета ERWin. Определение сущностей. Выбор СУБД	2
2. Создание логической схемы с помощью пакета ERWin. Назначение описательных и ключевых атрибутов. Спецификация связей.	2
3. Создание физической схемы с помощью пакета ERWin. Уровни физической модели. Нормализация отношений.	2
4. Генерация кода клиентской части с помощью ERWin. Создание отчетов в ERWin.	4
5. Выборка данных из базы с использованием конструкции SELECT языка SQL.	6
6. Создание, модификация и удаление объектов базы данных с использованием SQL	4
7. Использование курсоров для выполнения операций обновления и удаления данных	4
8. Создание базы данных, диаграмм, хранимых процедур и триггеров.	6
9. Создание и управление транзакциями	4
ИТОГО	34

При выполнении лабораторных работ по данному курсу студенты должны продемонстрировать умение решать задачи по созданию и ведению базы данных. При прохождении практикума на ЭВМ рекомендованы Case-средство ERWin и СУБД Microsoft SQL Server 2005.

Лабораторная работа выполняется строго в соответствии с выданным преподавателем заданием и вариантом. Завершающим этапом выполнения лабораторной работы является оформление отчета. Отчет содержит: титульный лист, лист задания, раздел, содержащий теоретические основы соответствующего раздела курса, включая основные используемые операторы, раздел, содержащий описание программной реализации: листинг программного блока (описание интерфейса программы можно вынести в приложение), раздел, содержащий описание результатов, список использованной литературы.

2.4 Самостоятельная работа студентов (71 час).

В качестве самостоятельной работы по дисциплине «Проектирование баз данных» студентам предлагается выполнить индивидуальную работу по проектированию и реализации базы данных. Данная работа должна содержать подробное исследование предметной области, анализ требований пользователей, проектирование базы данных, разработку пользовательских приложений, реализацию базы данных. Работа должна отвечать требованиям, предъявляемым к оформлению курсовых работ, и содержать разделы 1-4, описанные ниже.

ПРАВИЛА ОФОРМЛЕНИЯ ЗАЧЕТНЫХ РАБОТ

Зачетная работа начинается с титульного листа стандартной формы, за которым следует лист задания, реферат, содержание работы, введение, основная часть, состоящая из нескольких разделов, заключение, библиографического списка, приложения.

Введение содержит цель проектирования базы данных, общий обзор работы, позволяющий составить общее представление об исследуемой проблеме, анализ современного состояния исследований, разработок в данной области и полученных результатах. Во введении может быть предложена краткая аннотация отдельных разделов работы.

Первый раздел касается исследования предметной области. Должен содержать: описание предметной области, описание возможных пользователей базы данных; определение круга запросов и задач, которые предполагается решать с использованием созданной базы данных; определение частоты решения задач и используемые при этом бизнес-правила.

Второй раздел содержит проектирование базы данных, включая схемы, полученные с помощью CASE-средства ERWin.

Третий раздел содержит руководство пользователя с экранными формами программного продукта: назначение и функции программы; описание категорий пользователей программы, разграничения прав пользователей; описание последовательности пользовательских интерфейсов, реализующих каждую задачу; описание входных и выходных данных.

Заключение содержит перечень основных результатов, полученных в работе, и сделанных выводов. В него могут включаться рекомендации относительно перспектив продолжения данной работы.

2.5. Вопросы к зачету

1. Базы данных в составе автоматизированных систем.
2. Ввод и обработка данных. Базы данных в контуре управления.
3. Модели проектирования систем на основе баз данных.
4. Описание информационных потребностей пользователей.
5. Диаграммы потоков данных. Нотация Йордона – Де Марко.
6. Диаграммы потоков данных. Нотация Гейна – Сарсона.
7. Описание схемы базы данных. Проектирование логической схемы с помощью пакета ERWin.
8. Проектирование физической схемы базы данных с помощью пакета ERWin. Нормализация отношений.
9. Современные средства разработки приложений баз данных.

10. Создание приложений. Создание подключения к базе данных.
11. Архитектура файлового сервера.
12. Архитектура клиент-сервер.
13. Архитектура клиент-сервер в Intranet/Internet.
14. Архитектура на основе объектов.
15. Язык структурированных запросов SQL.
16. Основные конструкции языка.
17. Оператор SELECT.
18. Подзапросы SELECT.
19. Использование кванторов в инструкции SELECT.
20. Объединение, пересечение и разность в запросах SELECT.
21. Группировка и функции агрегирования в SELECT.
22. Типы данных SQL.
23. Операторы определения данных.
24. Операторы SQL обновления данных.
25. Понятие и типы курсоров.
26. Операторы SQL управления курсором.
27. Операторы ведения транзакций.
28. Идентификаторы пользователей и права владения
29. Операторы управления доступом к базе данных.
30. Поддержка транзакций.
31. Управление параллельностью.
32. Монитор транзакций.
33. Защита баз данных.
34. Типы опасностей. Компьютерные и некомпьютерные средства контроля.

2.6 Виды контроля.

Текущий контроль за аудиторной и самостоятельной работой обучаемых

осуществляется во время проведения аудиторных занятий посредством устного опроса, проведения контрольных работ или осуществления лекции в форме диалога. Промежуточный контроль осуществляется два раза в семестр в виде анализа выполнения логического и физического проектирования базы данных. Итоговый контроль осуществляется после успешного прохождения студентами текущего, промежуточного контроля, проверки и защиты индивидуальной работы в виде зачета.

2.7 Требования к знаниям студентов, предъявляемые на зачете

Зачет сдается в конце семестра. Форма сдачи зачета – устная. Необходимым условием допуска на зачет является сдача всех лабораторных работ и индивидуальной контрольной работы. В предлагаемый билет входят два вопроса и задача. Студент должен дать развернутый ответ на один из вопросов, и краткий – на другой. Развернутый ответ предполагает полное знание теории по данной части курса, свободную ориентацию в материале, краткий ответ – основных теоретических моментов: понятий и терминологии. В качестве задачи предлагается составление SQL запроса. При выполнении указанных требований ставится отметка «зачтено».

3. УЧЕБНО-МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ

3.1. Перечень обязательной (основной) литературы

1. Илюшечкин В.М. Основы использования и проектирования баз данных : учеб. пособие: рек. УМО/ В.М. Илюшечкин. – М.: Юрайт, 2009. – 224 с.
2. Сеннов А.С. Access 2007: учеб. курс СПб.: Питер, 2007. – 267.
3. СУБД: язык SQL в примерах и задачах: учеб. пособие. – М.: Физматлит, 2007. – 166 с.

3.2. Перечень дополнительной литературы

1. Грофф Д. Р., Вайнберг П.Н. SQL. Полное руководство. – Киев: «Ирина», 2001. – 800 с.
2. Дэвидсон Л. Проектирование баз данных на SQL SERVER 2000. – М.: БИНОМ. Лаборатория знаний, 2003. – 680 с.
3. Кумскова И.А. Базы данных. – М.: КноРус, 2010. – 496 с.
4. Маклаков С.В. Создание информационных систем с ALLFusion Modeling Suite. – М.: Диалог МИФИ, 2003. – 432 с.
5. Саукап Р. Основы Microsoft SQL Server 6.5. – М.: «Русская редакция», 1999. – 704 с.
6. Ульман Дж., Уидом Дж. Введение в системы баз данных. - М.: Издательство "Лори". - 2000, 374 с.
7. Чекалов А. П. Базы данных: от проектирования до разработки приложений. – СПб.: БВХ-Петербург, 2003. – 384 с.

4. НЕОБХОДИМОЕ ТЕХНИЧЕСКОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Лекции проводятся в стандартной аудитории, оснащенной в соответствии с требованиями преподавания теоретических дисциплин.

Для проведения лабораторных работ необходим компьютерный класс на 12-14 посадочных рабочих мест пользователей. В классе должен быть установлена СУБД Microsoft SQL Server, пакет ERWin.

5. УЧЕБНО-МЕТОДИЧЕСКАЯ (ТЕХНОЛОГИЧЕСКАЯ) КАРТА ДИСЦИПЛИНЫ

Номер недели	Номер темы	Вопросы, изучаемые на лекции	Занятия (номера)		Используемые нагляд. и метод пособия	Самостоятельная работа студентов		Форма контроля
			Практич. (семина.)	Лабораторные		Содержание	часы	
1	2	3	4	5	6	7	8	9
1	1	1-5		1, 2	3, 5 – 7, 9, 12 – осн. 3, 6, 7 – доп.	подготовка к защите лаб. работы	4	отчет по лабораторной работе №1
3	2	1-2		3, 4	8 – осн. 3, 4 – доп.	подготовка к защите лаб. работы	4	отчет по лабораторной работе №2
5	2	3-5		4	1, 5, 11 – осн. 3, 7 – доп.	подготовка к защите лаб. работы	2	отчет по лабораторной работе №3
7	3	1-3		5	2, 4, 6, 10, 13, 15, 16 – осн. 1, 2, 6 – доп.	подготовка к защите лаб. работы	2	отчет по лабораторной работе №4
9	3	3-4		5	2, 4, 6, 10, 13, 15, 16 – осн. 1, 2, 6 – доп.	подготовка к защите лаб. работы	2	отчет по лабораторной работе №5
11	3	5-6		6	2, 4, 6, 10, 13, 15, 16 – осн. 1, 2, 6 – доп.	подготовка к защите лаб. работы	2	отчет по лабораторной работе №6
13	4	1-4		7	1, 3, 6 – осн. 3, 7 – доп.	подготовка к защите лаб. работы	2	отчет по лабораторной работе №7
15	5	1-3		8	1, 5, 11 – осн. 3, 7 – доп.	подготовка к защите лаб. работы	2	отчет по лабораторной работе №8
17	6	1-6		9	2, 4, 10, 14 – осн. 1, 2, 5 – доп.	подготовка к защите лаб. работы	2	отчет по лабораторной работе №9

Условные обозначения:
к.р. – контрольная работа
сб. – собеседование

2. ГРАФИК САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

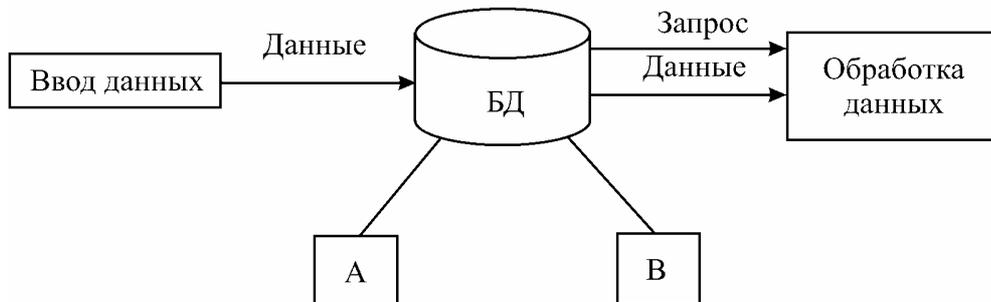
Содержание	Объем в часах	Сроки и форма контроля
3.1. Анализ требований к системе. Определение границ предметной области. Анализ информационных потребностей пользователей	15 час.	Собеседование (3 неделя)
3.2 Разработка концептуально инфологической модели базы данных	16 час.	Собеседование (5 неделя)
3.2. Разработка реляционной логической моделей базы данных	20 час.	Собеседование (9 неделя)
3.3. Разработка физической модели базы данных, основанной на логической модели	20 час.	Собеседование (13 неделя)

3. КРАТКИЙ КОНСПЕКТ ЛЕКЦИЙ

Лекция 1. Основы проектирования баз данных и анализ требований к системе

1. БД в составе автоматизированных систем.

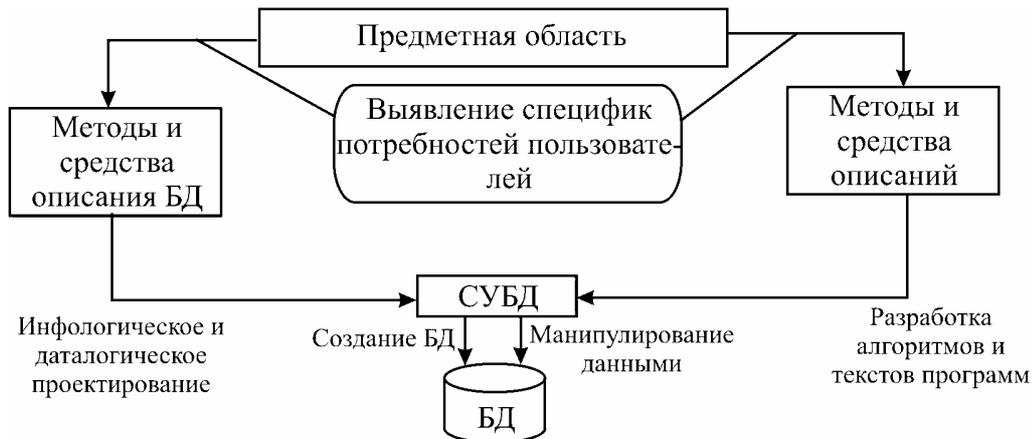
Если в запросе обрабатывается несколько таблиц хранящихся на нескольких узлах, то исходный запрос автоматически декомпозируется на несколько подзапросов. Подзапросы обрабатываются в узлах; результаты поиска передаются на сервер и там соединяются, полученные данные возвращаются в программу и в ней обрабатываются.



2. Модели проектирования систем на основе баз данных

В теории проектирования систем, в состав, которых входят базы данных, выделяют следующие модели: общая, каскадная и спиральная.

Общая модель проектирования систем на основе БД.



Каскадная модель проектирования систем на основе БД.

Включает следующие этапы:

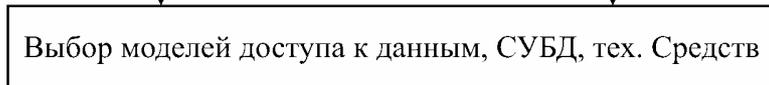
1) Выявление информационных потребностей:



2) Концептуальное проектирование:



3) Выбор архитектуры:



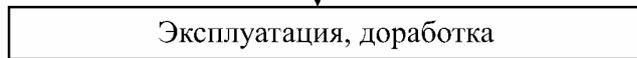
4) Логическое проектирование:



5) Отладка:

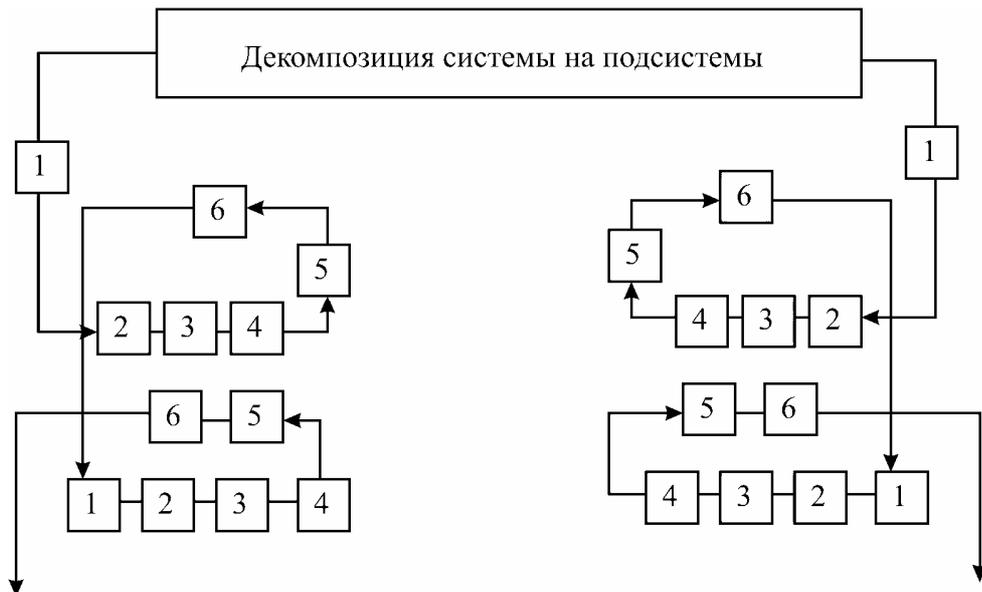


6) Сопровождение системы:



Спиральная модель проектирования систем на основе БД.

Такая модель является развитием каскадной.



[1] - Выбор архитектуры, [2] – введение информационных потребностей, [3] – концептуальное проектирование, [4] – логическое проектирование, [5] – отладка, тестирование, анализ, [6] – использование, сопровождение.

В отличие от каскадной модели, этапы [1] - [6] можно реализовать с помощью CASE – средств.

Анализ требований является наиболее важным этапом среди всех этапов жизненного цикла. На нем необходимо понять:

- 1) что предполагается сделать.
- 2) задокументировать требования.

Лекция 2. Средства разработки схемы и приложений базы данных.

Средство ERWin поддерживает методологию IDEF1X и стандарт IE (Information engineering). Методология IDEF1X подразделяется на уровни, соответствующие проектируемой модели данных системы. Каждый такой уровень соответствует определенной фазе проекта.

Верхний уровень состоит из Entity Relation Diagram (диаграмма сущность-связь) Key-Based model (модель данных, основанная на ключах). Диаграмма сущность-связь определяет сущности и их отношения. Модель данных, основанная на ключах, дает более полное представление данных. Оно включает описание всех сущностей и первичных ключей, которые соответствуют предметной области.

Нижний уровень состоит из Transformation Model (трансформационная модель) и Fully Attributed (полная атрибутивная модель). Трансформационная модель содержит всю информацию для реализации проекта, который может быть частью общей информационной системы и описывать предметную область. Трансформационная модель позволяет проектировщикам и администраторам БД представлять, какие объекты БД хранятся в словаре данных, и проверить, насколько физическая модель данных удовлетворяет требованиям информационной системы. Фактически из трансформационной модели можно автома-

тически получить модель СУБД, которая является полным отображением системного каталога СУБД.

ERwin имеет два уровня представления модели - логический и физический.

Логический уровень - это абстрактный взгляд на данные, на нем данные представляются так, как выглядят в реальном мире, и могут называться так, как они называются в реальном мире, например "Постоянный клиент", "Отдел" или "Фамилия сотрудника". Объекты модели, представляемые на логическом уровне, называются сущностями и атрибутами. Логическая модель данных является универсальной и никак не связана с конкретной реализацией СУБД.

Различают три уровня логической модели, отличающихся по глубине представления информации о данных:

- диаграмма сущность-связь (Entity Relationship Diagram, ERD);
- модель данных, основанная на ключах (Key Based model, KB);
- полная атрибутивная модель (Fully Attributed model, FA).

Диаграмма сущность-связь представляет собой модель данных верхнего уровня. Она включает сущности и взаимосвязи, отражающие основные бизнес-правила предметной области. Такая диаграмма не слишком детализирована, в нее включаются основные сущности и связи между ними, которые удовлетворяют основным требованиям, предъявляемым к ИС. Диаграмма сущность-связь может включать связи многие-ко-многим и не включать описание ключей. Как правило, ERD используется для презентаций и обсуждения структуры данных с экспертами предметной области.

Модель данных, основанная на ключах, - более подробное представление данных. Она включает описание всех сущностей и первичных ключей и предназначена для представления структуры данных и ключей, которые соответствуют предметной области.

Полная атрибутивная модель - наиболее детальное представление струк-

туры данных: представляет данные в третьей

Физическая модель данных, напротив, зависит от конкретной СУБД, фактически являясь отображением системного каталога. В физической модели содержится информация обо всех объектах БД. Поскольку стандартов на объекты БД не существует (например, нет стандарта на типы данных), физическая модель зависит от конкретной реализации СУБД. Следовательно, одной и той же логической модели могут соответствовать несколько разных физических моделей. Если в логической модели не имеет значения, какой конкретно тип данных имеет атрибут, то в физической модели важно описать всю информацию о конкретных физических объектах - таблицах, колонках, индексах, процедурах и т. д. Разделение модели данных на логические и физические позволяет решить несколько важных задач.

Различают два уровня физической модели:

- трансформационная модель (Transformation Model);
- модель СУБД (DBMS Model).

Физическая модель содержит всю информацию, необходимую для реализации конкретной БД. Трансформационная модель содержит информацию для реализации отдельного проекта, который может быть частью общей ИС и описывать подмножество предметной области. ERwin поддерживает ведение отдельных проектов, позволяя проектировщику выделять подмножество модели в виде предметных областей (Subject Area). Трансформационная модель позволяет проектировщикам и администраторам БД лучше представлять, какие объекты БД хранятся в словаре данных, и проверить, насколько физическая модель данных удовлетворяет требованиям к ИС.

Модель СУБД автоматически генерируется из трансформационной модели и является точным отображением системного каталога СУБД. ERwin непосредственно поддерживает эту модель путем генерации схемы БД.

Разработка приложений – проектирование пользовательского интерфей-

са и прикладных программ, предназначенных для работы с базой данных.

Лекция 3. Язык запросов SQL.

Язык SQL имеет общесистемный характер, поскольку:

- 1) является языком манипулирования и описания данных;
- 2) выступает как стандартное средство взаимосвязи серверов в распределенной системе;
- 3) используется практически во всех СУБД с архитектурой клиент-сервер.

Команды SQL делятся на 6 категорий:

1) Команды DQL – **языка запросов**, предназначены для извлечения данных из таблиц и описания внешнего вида полученных данных.

Основной инструкцией данной категории является инструкция SELECT.

2) Команды DML – **языка манипулирования данными**. Включает инструкции: INSERT, DELETE, UPDATE, которые позволяют добавлять, удалять и обновлять данные.

3) Команды DTL – **языка обработки транзакций**. Включает конструкции: SET TRANSACTION, COMMIT, ROLLBACK. Эти инструкции позволяют объединить команды в DML-группы, которые являются транзакциями. Если одна из команд, входящая в транзакцию, не может быть выполнена, то отменяются все предшествующие ей операции – откат транзакции.

4) Команды DDL – **языка определения данных**. Включают инструкции: CREATE, ALTER, DROP, которые определяют внешний вид объектов базы данных, изменение структуры, удаление.

5) Команды DCL – **языка управления данными**. Включают инструкции: GRANT, REVOKE, которые отвечают за предоставление и отмену привилегий.

6) Команды DCC – **языка управления курсором**, которые позволяют выбрать для обработки одну строку результирующего множества запроса. Включают инструкции: DECLARE, FETCH.

SELECT – инструкция на создание запроса на выборку без изменения данных (используется для поиска и чтения данных из таблиц БД).

Синтаксис оператора SELECT:

SELECT [**DISTINCT**] атрибут(ы)

FROM имя таблиц(ы)

[**WHERE** условие поиска]

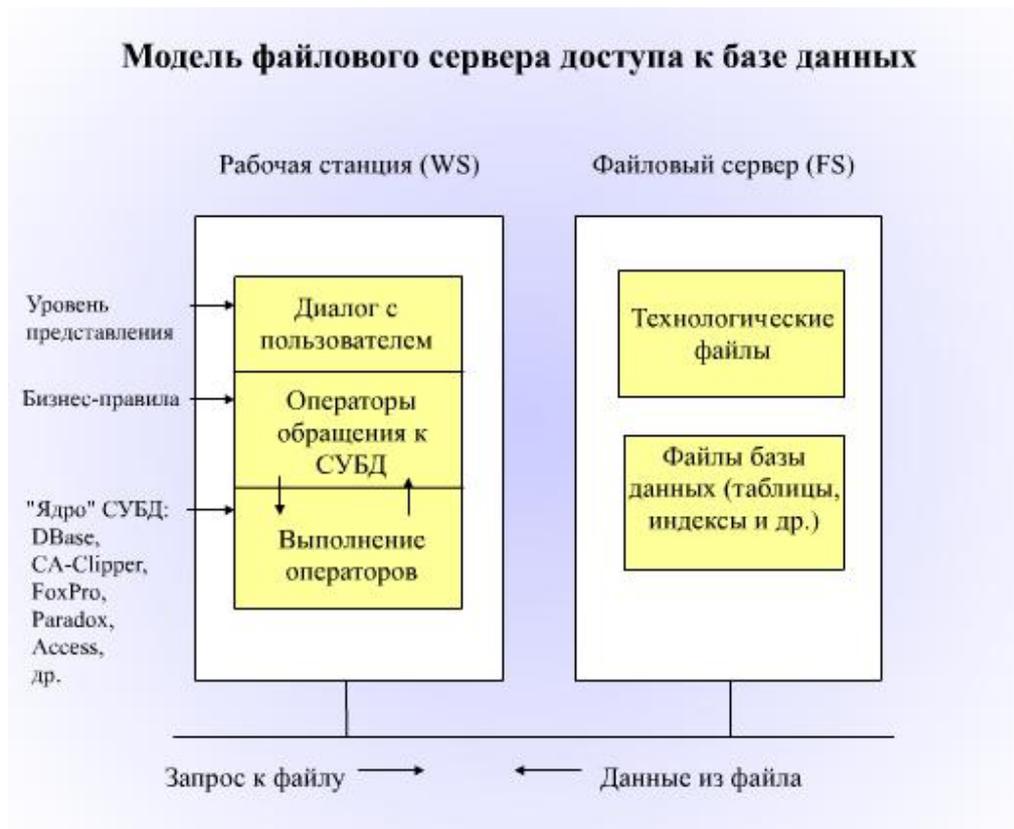
[**GROUP BY** атрибут(ы) [**HAVING** условие поиска]]

[**ORDER BY** атрибут(ы)]

Лекция 5. Архитектура доступа к базам данных

1. Архитектура файлового сервера

Рассматривается организация доступа к БД в случае, если она хранится на файловом сервере (см. рис. 1).



Рассмотренная модель доступа применяется в настоящее время редко. Используется в основном в небольших ЛВС с небольшой нагрузкой.

2. Архитектура клиент-сервер

2.1 Модель сервера БД

Приложения выполняются на рабочих станциях. Приложения включают: модули для организации диалога с пользователем и бизнес-правила, включающие ПП, описывающие бизнес процессы. Ядро СУБД является общим для всех рабочих станций и находится на сервере (рис. 2).

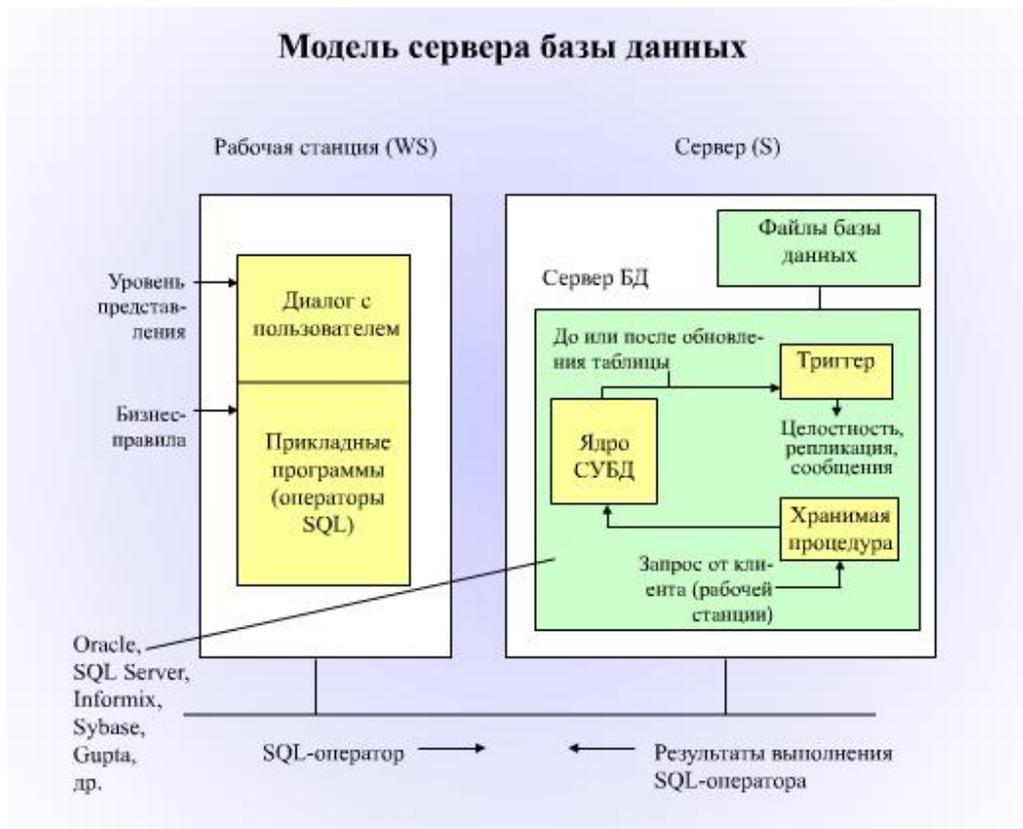


Рис. 2

Данная модель поддерживается следующими СУБД: Oracle, SQL Server, Informix, Sybase, Gupta и др.

Лекция 6. Управление доступом к базе данных.

Каждая СУБД должна предоставлять механизм, гарантирующий, что доступ к базе данных смогут получить только те пользователи, которые имеют соответствующее разрешение. Язык SQL включает операторы **GRANT** и **REVOKE**, предназначенные для организации защиты таблиц в базе данных.

Применяемый механизм защиты построен на использовании идентификаторов пользователей, предоставляемых им прав владения и привилегий.

1. Идентификаторы пользователей и права владения

Идентификатором пользователя называется обычный идентификатор языка SQL, используемый для обозначения некоторого пользователя базы данных. Каждому пользователю базы данных должен быть назначен собственный идентификатор, присваиваемый администратором базы данных (АБД). По очевидным соображениям защиты данных *идентификатор пользователя, как правило, защищается паролем*. Каждый выполняемый СУБД оператор SQL выполняется от имени какого-либо пользователя.

Идентификатор пользователя применяется для определения того, на какие объекты базы данных может ссылаться пользователь и какие операции с этими объектами он имеет право выполнять.

Лекция 7. Управление транзакциями и защита баз данных

Транзакцией называется последовательность операций, производимых над базой данных и переводящих базу данных из одного непротиворечивого (согласованного) состояния в другое непротиворечивое (согласованное) состояние.

Разработчик приложений или хранимых процедур определяет это исходя из смысла обработки данных, именно семантика совокупности операций над базой данных, которая моделирует с точки зрения разработчика некоторую одну неразрывную работу, и составляет транзакцию.

Рассмотрим пример, который связан с принятием заказа в фирме на изготовление компьютера. Компьютер состоит из комплектующих, которые сразу резервируются за данным заказом в момент его формирования. Тогда транзакцией будет вся последовательность операций, включающая следующие операции:

- ввод нового заказа со всеми реквизитами заказчика;
- изменения состояния для всех выбранных комплектующих на складе на

«занято» с привязкой их к определенному заказу;

– подсчет стоимости заказа с формированием платежного документа типа выставляемого счета к оплате;

– включение нового заказа в производство.

С точки зрения работника, это единая последовательность операций; если она будет прервана, то база данных потеряет свое целостное состояние.

2. Свойства транзакций. Способы завершения транзакций

Существуют различные модели транзакций, которые могут быть классифицированы на основании различных свойств, включающих структуру транзакции, параллельность внутри транзакции, продолжительность и т. д.

В настоящий момент выделяют следующие типы транзакций: плоские или классические транзакции, цепочечные транзакции и вложенные транзакции.

Плоские, или традиционные, транзакции, характеризуются четырьмя классическими свойствами: *свойство атомарности* (Atomicity) выражается в том, что транзакция должна быть выполнена в целом или не выполнена вовсе; *свойство согласованности* гарантирует, что по мере выполнения транзакций данные переходят из одного согласованного состояния в другое — транзакция не разрушает взаимной согласованности данных; *свойство изолированности* означает, что конкурирующие за доступ к базе данных транзакции физически обрабатываются последовательно, изолированно друг от друга, но для пользователей это выглядит так, как будто они выполняются параллельно; *свойство долговечности* трактуется следующим образом: если транзакция завершена успешно, то те изменения в данных, которые были ею произведены, не могут быть потеряны ни при каких обстоятельствах (даже в случае последующих ошибок).

4. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ПРОВЕДЕНИЮ ЛАБОРАТОРНЫХ РАБОТ

Лабораторный практикум, состоящий из 6 лабораторных работ затрагивает основные разделы дисциплины «Проектирование баз данных», позволяет студентам получить достаточно полное представление о структуре баз данных, способах создания схемы базы данных, а также приобрести практические навыки разработки запросов, необходимые для решения различных задач создания и использования баз данных.

Лабораторные работы имеют различный уровень сложности и выполняются два и четыре часа. Каждая предполагает самостоятельную работу студентов по освоению лекций. Текущий контроль знаний осуществляется путем опроса студентов после выполнения работы по вопросам, перечень которых приведен в каждой лабораторной работе.

ЛАБОРАТОРНАЯ РАБОТА № 1 (2 часа) ОТОБРАЖЕНИЕ МОДЕЛИ ДАННЫХ В ERWIN

Цель работы: изучить методологию IDEF1X; изучить уровни методологии IDEF1X; освоить инструментарий ERWin.

1. Физическая и логическая модель данных

ERwin имеет два уровня представления модели - логический и физический. *Логический уровень* - это абстрактный взгляд на данные, на нем данные представляются так, как выглядят в реальном мире, и могут называться так, как они называются в реальном мире, например "Постоянный клиент", "Отдел" или "Фамилия сотрудника". Объекты модели, представляемые на логическом уровне, называются сущностями и атрибутами. Логическая модель данных является универсальной и никак не связана с конкретной реализацией СУБД.

Физическая модель данных, напротив, зависит от конкретной СУБД, фактически являясь отображением системного каталога. В физической модели содержится информация о всех объектах БД. Поскольку стандартов на объекты БД не существует (например, нет стандарта на типы данных), физическая модель зависит от конкретной реализации СУБД. Следовательно, одной и той же логической модели могут соответствовать несколько разных физических моделей. Если в логической модели не имеет значения, какой конкретно тип данных

имеет атрибут, то в физической модели важно описать всю информацию о конкретных физических объектах - таблицах, колонках, индексах, процедурах и т. д. Разделение модели данных на логические и физические позволяет решить несколько важных задач.

Документирование модели. Многие СУБД имеют ограничение на именование объектов (например, ограничение на длину имени таблицы или запрет использования специальных символов - пробела и т. п.). Зачастую разработчики ИС имеют дело с нелокализованными версиями СУБД. Это означает, что объекты БД могут называться короткими словами, только латинскими символами и без использования специальных символов (т. е. нельзя назвать таблицу предложением - только одним словом). Кроме того, проектировщики БД нередко злоупотребляют "техническими" наименованиями, в результате таблица и колонки получают наименования типа *RTD_324* или *CUST_A12* и т. д. Полученную в результате структуру могут понять только специалисты (а чаще всего только авторы модели), ее невозможно обсуждать с экспертами предметной области. Разделение модели на логическую и физическую позволяет решить эту проблему. На физическом уровне объекты БД могут называться так, как того требуют ограничения СУБД. На логическом уровне можно этим объектам дать синонимы - имена более понятные неспециалистам, в том числе на кириллице и с использованием специальных символов.

Масштабирование. Создание модели данных, как правило, начинается с создания логической модели. После описания логической модели, проектировщик может выбрать необходимую СУБД и ERwin автоматически создаст соответствующую физическую модель. На основе физической модели ERwin может сгенерировать системный каталог СУБД или соответствующий SQL-скрипт. Этот процесс называется прямым проектированием (Forward Engineering). Тем самым достигается масштабируемость - создав одну логическую модель данных, можно сгенерировать физические модели под любую поддерживаемую ERwin СУБД. С другой стороны, ERwin способен по содержимому системного каталога или SQL-скрипту воссоздать физическую и логическую модель данных (Reverse Engineering). На основе полученной логической модели данных можно сгенерировать физическую модель для другой СУБД и затем сгенерировать ее системный каталог. Следовательно, ERwin позволяет решить задачу по переносу структуры данных с одного сервера на другой. Например, можно перенести структуру данных с Oracle на Informix (или наоборот) или перенести структуру dbf-файлов в реляционную СУБД, тем самым облегчив решение по переходу от файл-серверной к клиент-серверной ИС

Для переключения между логической и физической моделью данных служит список выбора в левой части панели инструментов Erwin (рис. 1).

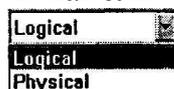


Рис. 1. Переключение между логической и физической моделью

При переключении, если физической модели еще не существует, она будет создана автоматически.

ЛАБОРАТОРНАЯ РАБОТА №2 (2 часа) СОЗДАНИЕ ЛОГИЧЕСКОЙ МОДЕЛИ ДАННЫХ

Цель работы: ознакомиться с технологией построения логической модели в ERWin, изучить методы определения ключевых атрибутов сущности, освоить метод проверки адекватности логической модели, изучить типы связей между сущностями.

Первым шагом при создании логической модели БД является построение диаграммы ERD (Entity Relationship Diagram). ERD-диаграммы состоят из трех частей: сущностей, атрибутов и взаимосвязей. ERD-диаграмма позволяет рассмотреть систему целиком и выяснить требования, необходимые для ее разработки, касающиеся хранения информации. ERD-диаграммы можно подразделить на отдельные фрагменты, соответствующие отдельным задачам, решаемым проектируемой системой. Это позволяет рассматривать систему с точки зрения функциональных возможностей, делая процесс проектирования управляемым.

1. Сущности и атрибуты

Основные компоненты диаграммы Erwin - это сущности, атрибуты и связи. Каждая сущность является множеством подобных индивидуальных объектов, называемых экземплярами. Каждый экземпляр индивидуален и должен отличаться от всех остальных экземпляров. Атрибут выражает определенное свойство объекта.

Построение модели данных предполагает определение сущностей и атрибутов, т. е. необходимо определить, какая информация будет храниться в конкретной сущности или атрибуте. Сущность можно определить как объект, событие или концепцию, информация о котором должна сохраняться. Примером может быть сущность *Заказчик* с атрибутами *Номер заказчика*, *Фамилия заказчика* и *Адрес заказчика*. На уровне физической модели ей может соответствовать таблица *Customer* с колонками *Customer_number*, *Customer_name* и *Customer_address*.

Для внесения сущности в модель необходимо "кликнуть" по кнопке сущности на панели инструментов (ERwin Toolbox) , затем "кликнуть" по тому месту на диаграмме, где необходимо расположить новую сущность. Щелкнув правой кнопкой мыши по сущности и выбрав из всплывающего меню пункт Entity Editor, можно вызвать диалог Entity Editor, в котором определяются имя, описание и комментарии сущности (рис. 2).

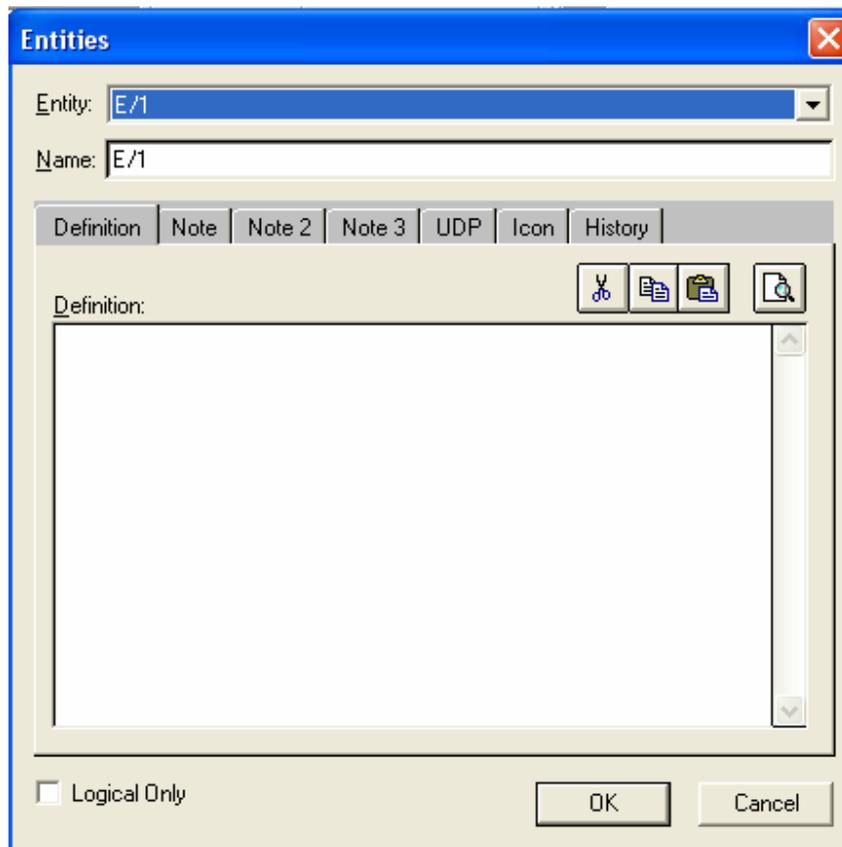


Рис. 2. Диалог Entity Editor

Каждая сущность должна быть полностью определена с помощью текстового описания в закладке Definition. Закладки Note, Note 2, Note 3, UDP (User Defined Properties - Свойства, определенные пользователем) служат для внесения дополнительных комментариев и определений к сущности. В прежних версиях ERwin закладкам Note2 и Note3 соответствовали окна Query и Sample.

Закладка Definition используется для ввода определения сущности. Эти определения полезны как на логическом уровне, поскольку позволяют понять, что это за объект, так и на физическом уровне, поскольку их можно экспортировать как часть схемы и использовать в реальной БД (CREATE COMMENT on entity_name).

Закладка Note позволяет добавлять дополнительные замечания о сущности, которые не были отражены в определении, введенном в закладке Definition. Здесь можно ввести полезное замечание, описывающее какое-либо бизнес-правило или соглашение по организации диаграммы.

В закладке Note 2 можно задокументировать некоторые возможные запросы, которые, как ожидается, будут использоваться по отношению к сущности в БД. При переходе к физическому проектированию, записанные запросы помогут принимать такие решения в отношении проектирования, которые делают БД более эффективной.

Закладка Note 3 позволяет вводить примеры данных для сущности (в произвольной форме).

В закладке Icon каждой сущности можно поставить в соответствие изображение, которое будет отображаться в режиме просмотра модели на уровне иконок. В этой закладке можно задать как большую иконку, которая будет отображаться на уровне Icon, так и малую иконку, которая будет отображаться на всех уровнях просмотра модели. Для связывания изображения с сущностью необходимо щелкнуть по кнопке , в появившемся диалоге ERwin Icon Editor щелкнуть по кнопке Import и выбрать соответствующий файл формата bmp.

ЛАБОРАТОРНАЯ РАБОТА № 3 (2 часа)

НОРМАЛИЗАЦИЯ ДАННЫХ. СОЗДАНИЕ ФИЗИЧЕСКОЙ МОДЕЛИ ДАННЫХ

Цель работы: изучить виды нормальных форм, освоить роль CASE-средства ERWin при нормализации и денормализации БД, построить физическую модель, изучить алгоритмы перевода БД в первую, вторую третью нормальную форму.

1. Нормализация

Нормализация - процесс проверки и реорганизации сущностей и атрибутов с целью удовлетворения требований к реляционной модели данных. Нормализация позволяет быть уверенным, что каждый атрибут определен для своей сущности, значительно сократить объем памяти для хранения информации и устранить аномалии в организации хранения данных. В результате проведения нормализации должна быть создана структура данных, при которой информация о каждом факте хранится только в одном месте. Процесс нормализации сводится к последовательному приведению структуры данных к нормальным формам - формализованным требованиям к организации данных. Известны шесть нормальных форм:

1. первая нормальная форма (1NF);
2. вторая нормальная форма (2NF);
3. третья нормальная форма (3NF);
4. нормальная форма Бойса - Кодда (усиленная 3NF);
5. четвертая нормальная форма (4NF);
6. пятая нормальная форма (5NF).

Функциональная зависимость (FD). Атрибут В сущности Е функционально зависит от атрибута А сущности Е тогда и только тогда, когда каждое значение А в Е связало с ним точно одно значение В в Е, т. е. А однозначно определяет В.

Полная функциональная зависимость. Атрибут В сущности Е полностью функционально зависит от ряда атрибутов А сущности Е тогда и только тогда, когда В функционально зависит от А и не зависит ни от какого подряда А.

Поддержка нормализации в ERwin. ERwin не содержит полного алгоритма нормализации и не может проводить нормализацию автоматически, однако его возможности облегчают создание нормализованной модели данных. Запрет на присвоение неуникальных имен атрибутов в рамках модели (при соответствующей установке опции Unique Name) облегчает соблюдение правила "один факт - в одном месте".

Денормализация. В результате нормализации все взаимосвязи данных становятся правильно определены, исключаются аномалии при оперировании с данными, модель данных становится легче поддерживать. Однако часто нормализация данных не ведет к повышению производительности ИС в целом. Время выполнения запроса с объединением может во много раз превосходить время выполнения запроса к одной таблице. В целях повышения производительности при переходе на физический уровень приходится сознательно отходить от нормальных форм для того, чтобы использовать возможности конкретного сервера или ИС в целом.

В отличие от процесса нормализации денормализация не может быть представлена в виде четко сформулированных правил. В каждом конкретном случае приходится искать конкретные решения, которые используют специфику ИС и предметной области и не могут быть тиражированы.

Поддержка денормализации в ERwin. Денормализация, как правило, проводится на уровне физической модели. ERwin позволяет сохранить на уровне логической модели нормализованную структуру, при этом построить на уровне физической модели структуру (возможно, денормализованную), которая обеспечивает лучшую производительность, используя особенности конкретной СУБД и бизнес-правил предметной области.

ERwin имеет следующую функциональность для поддержки денормализации:

– Сущности, атрибуты, ключи и домены можно создавать только на уровне логической модели, включив в соответствующих редакторах опцию Logical Only. Такие объекты не будут отображаться на уровне физической модели и не будут создаваться при генерации БД.

– Таблицы, колонки, домены и индексы можно создавать только на уровне физической модели (опция Physical Only).

– При автоматическом разрешении связи многие-ко-многим в физической модели создается новая таблица и структура данных может быть дополнена только на уровне физической модели.

ЛАБОРАТОРНАЯ РАБОТА № 4 (4 часа)

ГЕНЕРАЦИЯ КОДА КЛИЕНТСКОЙ ЧАСТИ С ПОМОЩЬЮ ERWIN.

СОЗДАНИЕ ОТЧЕТОВ В ERWIN.

Цель работы: научиться генерировать программный код пользовательского приложения, формировать отчеты.

1. Генерация кода в Visual Basic

ERwin поддерживает генерацию кода для MS Visual Basic В качестве источника информации при генерации форм служит модель ERwin. Использование ERwin позволяет одновременно описывать как клиентскую часть (объекты, отображающие данные на экране), так и сервер БД (процедуры и триггеры), тем самым оптимально распределяя функциональность ИС между клиентской и серверной частью. Компонент ERwin Form Wizard автоматически проектирует формы с дочерними объектами -кнопками, списками, полями, радиокнопками и т. д., используя расширенные атрибуты.

Совместное использование ERwin и Visual Basic может значительно сократить жизненный цикл разработки ИС, поскольку для каждой задачи используется наиболее эффективный инструмент. Visual Basic может быть использован для проектирования визуального интерфейса, а ERwin - для разработки логической и физической модели данных с последующей генерацией системного каталога сервера. Если БД уже существует, то с помощью ERwin можно провести обратное проектирование (reverse engineering), полученную модель дополнить расширенными атрибутами и сгенерировать клиентское приложение.

Для генерации клиентской части в диалоге Target Client (меню Client/Target Client) необходимо выбрать среду программирования - Visual Basic либо Power Builder. В среде Visual Basic 5.0 необходимо создать новый проект Visual Basic и подключить к нему два внешних файла CONST40.BAS и ERWIN40.BAS, расположенных в каталоге ERwin. Затем следует выбрать в меню Add-Ins/ERwin/Form Wizard. В появляющихся затем диалогах (для перехода к последующему служит кнопка Next, к предыдущему - Back) нужно последовательно указать имя файла модели, родительской и дочерних таблиц (возможно построение формы по родительской и нескольким дочерним таблицам) и колонок, которые будут отображены в сгенерированной форме. Затем нужно установить стиль отображения таблиц - свободный (freeform), в виде полей или табличный (grid). После нажатия на кнопку ОК будет сгенерировано приложение, которое может быть откомпилировано и выполнено без дополнительного редактирования.

Сгенерированная экранная форма помимо элементов отображения информации БД будет содержать управляющие кнопки New, Update, Delete, Close и навигатор для перемещения по строкам. Если полученную форму сделать главной формой проекта, то при запуске приложения возникнет диалог связи с БД. При успешном соединении (через ODBC) будет загружена информация из

БД.

ЛАБОРАТОРНАЯ РАБОТА № 5 (6 часов)
ВЫБОРКА ДАННЫХ ИЗ БАЗЫ ДАННЫХ
С ИСПОЛЬЗОВАНИЕМ ЯЗЫКА SQL

Цель работы: изучение спецификации запроса SQL, приобретение практических навыков составления и содержательной интерпретации запросов выборки данных.

База данных книготорговой компании

Рассмотрим предметную область, связанную с книгоизданием. В рамках данной предметной области существуют издатели, которые публикуют книги, авторы, которые книги пишут и сами книги. Диаграмма сущность-связь представлена на рисунке 3.



Рис. 3. Схема базы данных

Так как между сущностями Автор и Книга существует связь «многие ко многим», то создается дополнительная сущность Книги авторов.

Структура таблицы авторы (authors)

Имя атрибута	Тип данных	Размер	Возможность значений NULL	Описание
Au_id	varchar	11	Нет	Номер автора
Au_lname	varchar	40	Нет	Фамилия автора
Au_fname	varchar	20	Нет	Имя автора
Phone	char	12	Нет	Номер телефона
Address	varchar	40	Да	Адрес
City	varchar	20	Да	Город
State	char	2	Да	Штат

Имя атрибута	Тип данных	Размер	Возможность значений NULL	Описание
Zip	char	5	Да	Энергичность
Contract	bit	1	Нет	Наличие контракта

Структура таблицы издательство (publishers)

Имя атрибута	Тип данных	Размер	Возможность значений NULL	Описание
Pub_id	char	4	Нет	Номер издательства
Pub_name	varchar	40	Да	Название издательства
City	varchar	20	Да	Город
State	char	2	Да	Штат
Country	varchar	30	Да	Страна

Структура таблицы книги (titles)

Имя атрибута	Тип данных	Размер	Возможность значений NULL	Описание
Title_id	varchar	6	Нет	Номер книги
Title	varchar	80	Нет	Название книги
Type	char	12	Нет	Тип книги
Pub_id	char	4	Да	Номер издательства
Price	money	8	Да	Цена
Advance	money	8	Да	Аванс (стоимость предварительной продажи)
Royalty	int	4	Да	Гонорар
Ytd_sales	int	4	Да	Число книг, проданных в текущем году
Notes	varchar	200	Да	Замечания
Pubdate	datetime	8	Нет	Дата опубликования

Структура таблицы Книги авторов (titleauthor)

Имя атрибута	Тип данных	Размер	Возможность значений NULL	Описание
Au_id	varchar	11	Нет	Номер автора книги
Title_id	varchar	6	Нет	Номер книги
Au_ord	tinyint	1	Да	Порядок автора в названии книги
Royaltyper	int	4	Да	Авторский гонорар

ЛАБОРАТОРНАЯ РАБОТА № 6 (4 часа)

СОЗДАНИЕ, МОДИФИКАЦИЯ И УДАЛЕНИЕ ОБЪЕКТОВ БАЗЫ ДАННЫХ С ИСПОЛЬЗОВАНИЕМ SQL

Цель работы – изучить языки определения и манипулирования данными

SQL, получить практические навыки составления SQL-запросов для работы с таблицами, индексами, представлениями, а также их выполнение на SQL-сервере с использованием клиентских утилит.

Выполнение работы

1. Изучение языка баз данных

1.1. Изучить язык определения данных SQL, включая операторы создания и удаления. Изучить типы ограничений и способы их представления на языке SQL.

1.2. Изучить операторы манипулирования данными, включая операторы связанные с курсором.

1.3. Изучить интегрированную утилиту SQL Enterprise Manager, а также системные процедуры с точки зрения их использования для просмотра объектов базы данных, создания и удаления таблиц. В частности, изучить работу с системной процедурой `sp_help`, используемой для получения информации о базе данных и объектах базы данных.

2. Работа с таблицей

2.1. В соответствии с вариантом задания разработать точную структуру строк таблицы, включая выбор типов данных для каждого поля строки.

2.2. Составить оператор создания таблицы с учетом приведенных в задании ограничений, выполнить данный оператор на SQL-сервере с использованием клиентской утилиты. Просмотреть результат выполнения данного оператора с помощью системной хранимой процедуры `sp_help` или клиентской утилиты SQL Enterprise Manager.

2.3. Заполнить созданную таблицу данными с использованием оператора включения (не менее 10 строк). Просмотреть заполненную таблицу.

2.4. Изменить две строки таблицы с использованием оператора поисковой модификации. Просмотреть измененную таблицу.

2.5. Удалить одну строку таблицы с использованием оператора поисковой модификации. Просмотреть измененную таблицу.

2.6. Добавить столбец в таблицу с использованием оператора модификации таблицы. Просмотреть измененную таблицу.

2.7. Удалить ограничения из таблицы с использованием оператора модификации таблицы. Просмотреть измененную таблицу.

4. Работа с индексами

4.1. Создать индекс, который бы позволял быстрый поиск по первичному ключу, содержащему столбцы уникальности.

4.2. Составить и выполнить запрос к индексированной таблице.

4.3. Удалить индекс.

ЛАБОРАТОРНАЯ РАБОТА № 7 (4 ЧАСА)

ИСПОЛЬЗОВАНИЕ КУРСОРОВ ДЛЯ ВЫПОЛНЕНИЯ ОПЕРАЦИЙ ОБНОВЛЕНИЯ И УДАЛЕНИЯ ДАННЫХ

1. Работа с курсором

1.1. Объявить скроллируемый курсор в соответствии со спецификацией курсора из лабораторного задания.

1.2. Открыть курсор.

1.3. Просмотреть первую, i -ю, и последнюю строки результирующей таблицы, а также последовательно всю результирующую таблицу от начала до конца и от конца до начала (число i задается преподавателем).

1.4. Удалить i -ю строку результирующей таблицы с использованием оператора позиционного удаления (число i задается преподавателем). Просмотреть базовую таблицу и сравнить ее с предыдущим вариантом.

1.5. Модифицировать i -ю строку результирующей таблицы, изменив значение одного или нескольких полей, с использованием оператора позиционной модификации (число i задается преподавателем). Просмотреть базовую таблицу и сравнить ее с предыдущим вариантом.

1.6. Закрыть курсор.

2. Работа с курсором

2.1. В соответствии с заданием предыдущей лабораторной работы составить оператор создания представления, выполнить данный оператор на SQL-сервере с использованием клиентской утилиты. Просмотреть результат выполнения данного оператора с помощью системной хранимой процедуры `sp_help` или клиентской утилиты SQL Enterprise Manager. Просмотреть представляемую таблицу.

2.2. Включить несколько записей в представляемую таблицу с использованием оператора включения. Просмотреть представляемую и базовую таблицы и сравнить их с предыдущими вариантами этих таблиц.

2.3. Изменить несколько строк представляемой таблицы с использованием оператора поисковой модификацией. Просмотреть представляемую и базовую таблицы и сравнить их с предыдущими вариантами этих таблиц.

2.4. Удалить несколько строк из представляемой таблицы с использованием оператора поискового удаления. Просмотреть представляемую и базовую

таблицы и сравнить их с предыдущими вариантами этих таблиц.

2.5. Удалить представление с использованием оператора удаления представления.

Вариант 1

Схема таблицы СТУДЕНТ:

- идентификатор зачетки
- фамилия и инициалы студента
- специальность
- группа
- дата рождения
- наличие стипендии (имеется/не имеется)
- адрес проживания
- средний балл зачетки

Спецификация курсора: результирующая таблица включает фамилию и инициалы студента, группу и адрес проживания для студентов, имеющих средний балл зачетки более 4.5.

ЛАБОРАТОРНАЯ РАБОТА № 8 (6 часов)

СОЗДАНИЕ БАЗЫ ДАННЫХ, ДИАГРАММ, ХРАНИМЫХ ПРОЦЕДУР И ТРИГГЕРОВ.

Цель работы: изучить оператор создания базы данных, диаграммы базы данных, хранимых процедур, триггеров. Получить практические навыки реализации базы данных и составления SQL-запросов, удовлетворяющих информационных запросов пользователей.

1. Создание базы данных books Microsoft SQL Server

1. Для создания базы данных воспользуемся клиентской утилитой SQL Server Management Studio.

2. Перейдите в папку «Базы данных» на сервере, в котором работаете. Щелкните правой кнопкой мыши по папке «Базы данных» и в контекстном меню выберите пункт «Создать базу данных ...».

3. В появившемся диалоговом окне «Создание базы данных» задаем имя базы данных. Имена баз данных должны быть уникальны внутри экземпляра SQL Server и должны соответствовать правилам для идентификаторов. Имя может иметь максимальную длину 128 символов. Вводим имя базы данных **books**. Остальные параметры оставляем без изменения. Нажмите ОК.

4. База данных books и файл журнала базы данных будут сохранены в

папке C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Data.

В окне «Создание базы данных» формируется список файлов данных. Набор файлов формируется с помощью таблицы, которая имеет несколько столбцов. В первом из них определяется имя файла, в четвертом – его начальный размер.

1. Указанное имя базы данных станет использоваться и в качестве логического, и в качестве физического имени файла базы данных.

2. Создание диаграммы базы данных books Microsoft SQL Server

После того как база данных создана, в ней определены таблицы, можно построить диаграмму базы данных. Для создания диаграммы в контекстном меню папки «Диаграмма базы данных» требуемой базы данных необходимо выбрать пункт контекстного меню «Создать диаграмму базы данных».

В появившемся диалоговом окне «Добавление таблицы» последовательно добавляем в окно диаграммы все необходимые таблицы. Если вами правильно определены внешние и первичные ключи, то получите диаграмму схемы базы данных.

ЛАБОРАТОРНАЯ РАБОТА № 9 (4 часа)

СОЗДАНИЕ И УПРАВЛЕНИЕ ТРАНЗАКЦИЯМИ

Цель работы: изучение способов обеспечения надежной работы SQL Server с помощью механизма транзакций и контрольных точек, приобретение навыков управления локальными и распределенными транзакциями различных видов, а также ознакомление с физической и логической архитектурой журнала транзакций и способами восстановления баз данных.

Транзакции должны как можно меньше включать команд и изменять минимум данных. Соблюдение этого требования позволит наиболее эффективным образом обеспечить одновременную работу с данными множества пользователей системы SQL Server. Они определяются на уровне соединения с сервером. Поэтому при закрытии соединения происходит откат невыполненной транзакции, и ее нельзя выполнить позже после восстановления соединения.

Для управления распределенными транзакциями в MS SQL Server 2000 используется координатор DTC (Distribution Transaction Coordinator), удовлетворяющий спецификации “X/ OPUN XA for Distributed Transaction Processing”. Координатор MS DTC начинает и заканчивает локальные транзакции, а также откатывает их назад, если одна из них закончилась с ошибкой. При выполнении распределенных транзакций пользователь может обращаться не только к серверам SQL Server 2000, но и к другим источникам данных: Oracle, Access, источники ODBC и другие.

Распределенная транзакция может быть начата несколькими способами.

1. Если приложение в локальной транзакции использует распределенный запрос, то сервер автоматически начинает выполнение распределенной транзакции.

2. Если приложение начинает локальную транзакцию и из нее вызывает удаленную хранимую процедуру при установленном параметре `REMOTE_PROC_TRANSACTION`, то эта транзакция автоматически расширяется до распределенной транзакции (см. `sp_configure`).

3. Приложение может начать распределенную транзакцию, используя методы `OLE DB` или `ODBC`.

4. Сервер начинает выполнение распределенной транзакции, если встречает команду

`BEGIN DISTRIBUTED TRANSACTION` Имя транзакции.

В этом случае для завершения и отката транзакций используются команды `COMMIT TRANSACTION` и `ROLLBACK TRANSACTION`.

При работе с явными транзакциями можно использовать вложенные транзакции, выполнение которых инициируется из тела уже активной транзакции. Для управления вложенными транзакциями используются те же команды, при этом каждая команда `COMMIT` работает только с последней начатой транзакцией.

Если в команде `ROLLBACK` не задано имя транзакции, то откатываются все вложенные транзакции и транзакция самого высокого уровня.

Если же имеется необходимость откатить лишь часть транзакций, то предварительно надо создать точку сохранения с помощью команды `SAVE TRANSACTION`, которую следует указывать при откате.

Функция `@@TRANCOUNT` предназначена для определения количества активных транзакций, начатых в активном соединении.

Во всех транзакциях нельзя использовать следующие команды: `ALTER DATABASE`, `BACKUP LOG`, `CREATE DATABASE`, `DISK INIT`, `DROP DATABASE`, `DUMP TRANSACTION`, `LOAD DATABASE`, `LOAD TRANSACTION`, `RECONFIGURE`, `RESTORE DATABASE`, `RESTORE LOG`, `UPDATE STATISTICS`, а также системной хранимой процедуры `sp_droption` и любой другой хранимой процедуры, изменяющей значения в системной базе `master`.

Для отката таких действий необходимо использовать архивирование базы данных и последующее ее восстановление.

`MS SQL Server 2000` для обеспечения целостности данных использует журнал транзакции, который имеет сложную физическую и логическую архитектуру и в котором производятся все промежуточные модификации до завершения транзакции, после которой происходит фиксация изменений в самой базе данных. Контрольные точки минимизируют данные в журнале транзакций.

5. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ПРИМЕНЕНИЮ СОВРЕМЕННЫХ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ ДЛЯ ПРЕПОДАВАНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ

Для выполнения лабораторных работ используются ERwin и Microsoft SQL Server.

Для построения модели данных PLATINUM technology предлагает мощный и удобный инструмент - ERwin. ERwin имеет два уровня представления модели - логический и физический. ERwin позволяет проводить процессы прямого и обратного проектирования БД. Кроме того, ERwin позволяет выравнивать модель и содержимое системного каталога после редактирования того либо другого. ERwin интегрируется с популярными средствами разработки клиентской части - PowerBuilder, Visual Basic, Delphi, что позволяет автоматически генерировать код приложения, который полностью готов к компиляции и выполнению.

Microsoft SQL Server представляет собой СУБД, обеспечивающую создание систем, в состав которых входят базы данных с архитектурой «клиент-сервер». Эта СУБД поддерживает: тиражирование данных, параллельную обработку, создание и обработку больших баз данных, обеспечивает интеграцию баз данных SQL Server в Web.

Системы, построенные с использованием этой СУБД, традиционно характеризуются невысокой суммарной стоимостью владения, что для специалистов вкупе с возможностями системы является одним из самых важных критериев при выборе программного продукта, который будет использован при построении баз данных предприятия.

Поэтому СУБД Microsoft SQL Server и пакет ERwin используются при выполнении лабораторных работ по дисциплине «Проектирование баз данных» может быть использована в дальнейшей деятельности выпускников.

6. ЗАДАНИЯ ДЛЯ ЛАБОРАТОРНЫХ РАБОТ

Лабораторная работа № 1

Задание

1. Создать новый проект в ERWin.
2. Сформировать модель БД системы согласно разработанной вами инфологической модели. Включить в модель только имена сущностей, не определяя атрибуты.
3. Сохранить модель в файл.
4. Выбрать сервер БД.
5. Сгенерировать схему БД для выбранного сервера.

Контрольные вопросы.

1. Назовите уровни методологии IDEF1X.
2. Из каких моделей состоит логический уровень?
3. Из каких моделей состоит физический уровень?
4. Что включает в себя диаграмма сущность-связь?
5. Что включает в себя модель данных, основанная на ключах?
6. Какую информацию содержит трансформационная модель?
7. Что включает в себя полная атрибутивная модель?
8. Что называют моделью СУБД?
9. Какие кнопки панели инструментов позволяют изменить уровень просмотра модели?
10. Как сгенерировать схему БД?
11. Каким образом осуществляется выбор сервера для генерации схемы БД?
12. Как добавить сущность на диаграмму?

Лабораторная работа № 2

Задание

1. Задать атрибуты сущностей, созданных в лабораторной работе № 1.
2. Определить первичные ключи в сущностях.
3. Определить состав альтернативных ключей.
4. Связать сущности между собой используя описанные типы связей.
5. После установления связей определить состав внешних ключей.
6. Проверить модель на соответствие предметной области.
7. Сохранить полученную диаграмму.

Контрольные вопросы.

1. Назовите основные части ERD-диаграммы.
2. Какова цель создания ERD-диаграммы?
3. Что является основным компонентом реляционных БД?
4. Что называется сущностью?
5. Какие типы связей Вам известны?
6. Чем отличается идентифицирующая связь от неидентифицирующей?
7. Что такое функциональное имя?
8. Что называют первичным ключом?
9. Назовите принципы, согласно которым формируется первичный ключ?
10. Что называют альтернативным ключом?
11. В каком случае образуются внешние ключи?
12. Что называют иерархической рекурсией?
13. Что называют сетевой рекурсией?
14. Что такое «правила ссылочной целостности»?
15. Как в ERWin реализована связь «многие-ко-многим»?
16. Что называется инверсионным входом?

Лабораторная работа № 3

Задание

1. Нормализовать БД до 3НФ.
2. Построить физическую модель БД, рассматриваемой вами в 1 и 2 лабораторных работах.
3. Для колонок физической модели установите правила валидации и значения по умолчанию.

Контрольные вопросы.

1. Что такое нормализация?
2. Какова цель денормализации?
3. Что называется функциональной зависимостью?
4. Что называется полной функциональной зависимостью?
5. 1НФ.
6. 2НФ.
7. 3НФ.
8. НФБК.
9. 4НФ.
10. 5НФ.
11. Что такое домен?
12. С какой целью в ERWin определяются домены?

13. Какова цель создания физической модели?
14. Что такое представление?
15. С какой целью в физической модели применяются представления?
16. Что такое «правила валидации»?
17. Как реализовываются правила валидации?

Лабораторная работа № 5

Задание

1. Изучить утилиту Management Studio СУБД SQL Server.
2. Изучить состав базы данных книготорговой компании (pubs). Структуру и семантику ее таблиц.
3. Для указанных заданий создать SQL-запросы по их содержательному описанию и выполнить на SQL-сервере.
4. Сохранить результаты выполнения запросов.

Вариант 1

- 1) Получить список фамилий и имен всех авторов.
- 2) Получить список всех изданных книг.
- 3) Определить число книг с названием, начинающимся на "Со" и заканчивающимся на "s".
- 4) Определить тип и среднюю цену всех книг стоимостью не ниже 11\$, если средняя цена типа больше 19.7\$.
- 5) Определить фамилию и имя авторов, название книг по современным компьютерам, изданных в США или по психологии, изданным во Франции.
- 6) Определить название, тип и цену книг, цена которых больше всех цен книг по психологии.
- 7) Определить города и штаты проживания авторов, в которых не находятся издательства, и города и штаты издательств, в которых не проживают авторы. В запросе использовать оператор UNION и квантор NOT EXISTS
- 8) Для каждого штата определить минимальную, максимальную и среднюю цену книг, написанных авторами не из США.
- 9) Выбрать информацию о книгах, идентификаторы которых начинаются буквой "B", заканчиваются строкой "1342", а вторым символом идентификатора являются буквы "A", "U" или "N". Информация о книгах должна включать тип, идентификатор и цену книги.
- 10) Произвести проекцию на столбцы title и pub_name декартова произведения таблиц titles и publishers.
- 11) Определить, кто из авторов написал какую книгу по психологии. В выбираемые данные включить имя и фамилию автора, а также название книги. В разделе FROM запроса использовать операцию соединению JOIN.

- 12) Выбрать все столбцы таблицы titles и столбец pub_name таблицы publishers результата эквисоединения данных таблиц по идентификатору издателя.
- 13) Определить все штаты, в которых проживают авторы. Названия штатов в результирующей таблице не должны повторяться. Вывести названия штатов в порядке возрастания.
- 14) Определить минимальную и максимальную цену для каждого из типов книг. В результирующую таблицу не включать те типы книг, для которых разность между максимальной и средней ценой меньше 7 долларов.
- 15) Определить названия и цену самых дешевых книг, вышедших в США. (Самые дешевые книги имеют минимальную цену).
- 16) Найти издательства, среди изданных книг которых найдется хоть одна книга по компьютерам стоимостью более двух долларов. В запросе использовать подзапрос и предикат с квантором.

Лабораторная работа № 6

Задание

1. Вариант задания получить у преподавателя.
2. Операторы создания и удаления таблицы.
3. Операторы создания и удаления представления.
4. Операторы создания и удаления индекса.
5. Операторы манипулирования данными, относящиеся к базовой, представляемой и результирующей таблицам.
6. Исходная базовая таблица.
7. Исходная представляемая таблица.
8. Измененные таблицы (базовая, представляемая, результирующая) и ссылки на соответствующие им операторы изменения таблиц (для каждого акта изменения.)

Вариант 1

Схема таблицы СТУДЕНТ:

- идентификатор зачетки
- фамилия и инициалы студента
- специальность
- группа
- дата рождения
- наличие стипендии (имеется/не имеется)
- адрес проживания
- средний балл зачетки

Ограничение уникальности: идентификатор зачетки.

Проверочные ограничения: а) Код группы должен иметь следующую структуру: <цифра><цифра><буква><буква><цифра>; б) средний балл зачетки должен быть в интервале [2,5].

Спецификация представления: представляемая таблица содержит идентификатор зачетки, фамилию и инициалы студента, а также средний балл зачетки для студентов, получающих стипендию.

Лабораторная работа № 7

Задание

1. Вариант задания получить у преподавателя.
2. Операторы объявления, открытия и закрытия курсора.
3. Исходная результирующая таблица курсора.
4. Операторы манипулирования данными, относящиеся к базовой, представляемой и результирующей таблицам.
5. Исходная базовая таблица.
6. Исходная представляемая таблица.
7. Измененные таблицы (базовая, представляемая, результирующая) и ссылки на соответствующие им операторы изменения таблиц (для каждого акта изменения.)

Вариант 1

Схема таблицы СТУДЕНТ:

- идентификатор зачетки
- фамилия и инициалы студента
- специальность
- группа
- дата рождения
- наличие стипендии (имеется/не имеется)
- адрес проживания
- средний балл зачетки

Спецификация курсора: результирующая таблица включает фамилию и инициалы студента, группу и адрес проживания для студентов, имеющих средний балл зачетки более 4.5.

Лабораторная работа № 8

Задание

1. Согласно полученному варианту задания создать базу данных.
2. Создать и заполнить таблицы (не менее 5 строк).
3. Создать диаграмму.
4. Создать запросы на выборку.
5. Создать хранимые процедуры.
6. Создать триггеры.

Содержание отчета

1. Операторы создания объектов базы данных.

2. Исходные таблицы.
3. Диаграмма схемы базы данных.
4. SQL-запросы, соответствующие задачам пользователей.
5. Результаты выполнения SQL-запросов.
6. Операторы создания хранимых процедур.
7. Результаты работы хранимых процедур
8. Операторы создания триггеров.
9. Результаты работы триггеров.

Лабораторная работа № 9

Задание

1. Проверить режимы явного и неявного начала транзакций.
2. Создать вложенные транзакции, выполнив следующие команды:

```
CREATE TABLE #aaa (cola int) -- 0-й уровень
BEGIN TRAN -- 1-й уровень
INSERT INTO #aaa VALUES (111)
BEGIN TRAN -- 2-й уровень
INSERT INTO #aaa VALUES (222)
BEGIN TRAN -- 3-й уровень
INSERT INTO #aaa VALUES (333)
SELECT * FROM #aaa
SELECT 'Вложенность транзакций', @@TRANCOUNT
ROLLBACK TRAN
SELECT * FROM #aaa -- откат на 0-й уровень
SELECT 'Вложенность транзакций', @@TRANCOUNT
```

 Проанализировать полученные результаты.
3. Написать пример пакета запросов с использованием команд COMMIT и ROLLBACK для автоматических, неявных и явных транзакций.
4. Написать пример пакета команд, иллюстрирующих использование средств оптимизации при откате транзакций.
5. Используя средства MS SQL Server 2000, изучить физическую и логическую архитектуру журнала транзакций.
6. С помощью системной хранимой процедуры sp_configure изменить интервал контрольных точек для базы данных Pubs.
7. Уточнить синтаксис команд управления транзакциями и написать пример пакета с использованием всех вариантов этих команд.

7. ФОНД ТЕСТОВЫХ И КОНТРОЛЬНЫХ ЗАДАНИЙ ДЛЯ ОЦЕНКИ КАЧЕСТВА ЗНАНИЙ ПО ДИСЦИПЛИНЕ

Вариант 1

1. Укажите верный запрос: выдать номера поставщиков, которые живут в Москве и имеют состояние не менее 25 у.е.

Поставщики S

<u>№ поставщика</u>	Имя	Состояние	Город
S1	Иванов	12	Москва
S2	Петров	20	Тула
S3	Сидоров	35	Москва

1) SELECT №_поставщика
FROM S
WHERE Город = 'Москва'
AND Состояние >= 25;

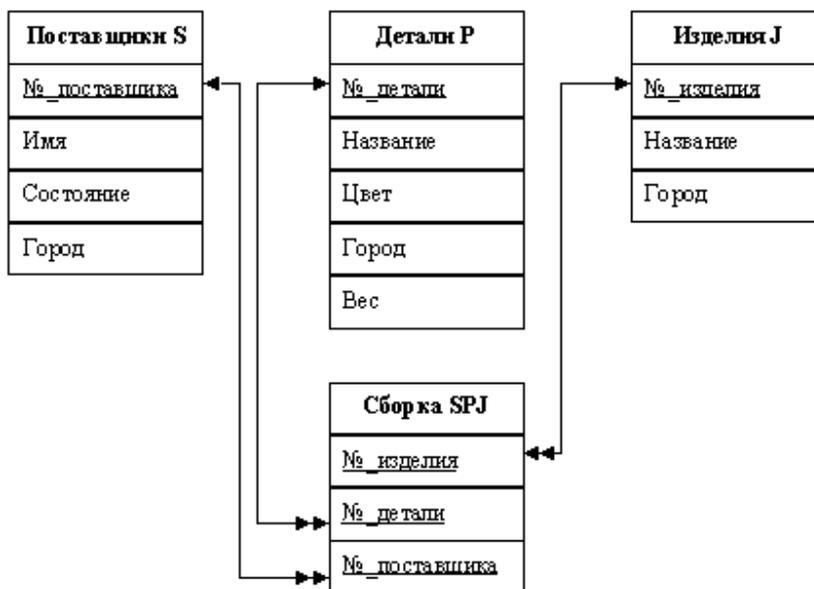
3) SELECT №_поставщика
FROM S
WHERE Город = Москва
AND Состояние >= 25;

2) SELECT №_поставщика, Состояние, Город
FROM S
WHERE Город = 'Москва' AND Состояние >= 25;

4) SELECT №_поставщика
FROM S
WHERE Город = 'Москва' OR Состояние >= 25;

2. Укажите неверный оператор, соответствующий запросу: выдать имена поставщиков, которые поставляют гайки.

Схема БД имеет вид



Поставщики S

<u>№ поставщика</u>	Имя	Состояние	Город
S1	Иванов	12	Москва
S2	Петров	20	Тула
S3	Сидоров	35	Москва
S4	Орлов	5	Курск

Сборка SPJ

<u>№ поставщика</u>	<u>№ детали</u>	<u>№ изделия</u>	<u>Количество</u>
S1	P1	J1	10 ⁷
S2	P2	J2	10 ⁵
S3	P3	J3	10 ⁴
S4	P2	J3	10 ⁵

Детали P

<u>№ детали</u>	<u>Название</u>	<u>Цвет</u>	<u>Вес</u>	<u>Город</u>
P1	болт	черный	0,01	Москва
P2	гайка	черный	0,02	Москва
P3	винт	красный	0,03	Тула

1) SELECT Имя
FROM S, SPJ, P
WHERE S.№_поставщика = SPJ.№_поставщика
AND SPJ. №_детали = P. №_детали
AND Название = 'гайка';

3) SELECT Имя
FROM S JOIN SPJ ON
S.№_поставщика = SPJ.№_поставщика,
SPJ JOIN P ON SPJ. №_детали = P. №_детали
WHERE Название = 'гайка';

2) SELECT Имя
FROM S NATURAL JOIN SPJ,
SPJ NATURAL JOIN P
WHERE Название = 'гайка';

4) SELECT Имя
FROM S, P
WHERE P.Название = 'гайка';

3. Укажите выражение, которое возвращает истинное значение, если таблица, соответствующая подзапросу не является пустой

1) $a > \text{ANY}(\text{SELECT } b \text{ FROM } \dots \text{ WHERE } \dots)$

2) EXISTS

(SELECT ... FROM ... WHERE ...)

3) NOT EXISTS

(SELECT ... FROM ... WHERE ...)

4) $a > \text{ALL}(\text{SELECT } b \text{ FROM } \dots \text{ WHERE } \dots)$

4. Укажите верный оператор определения таблицы Поставщика S.

Поставщики S

<u>№ поставщика</u>	<u>имя</u>	<u>состояние</u>	<u>город</u>
S1	Иванов	12	Москва
S2	Петров	20	Тула
S3	Сидоров	35	Москва
S4	Орлов	5	Курск

1) CREATE TABLE S
(№_поставщика VARCHAR (9) NOT NULL,
имя VARCHAR (25),
состояние FLOAT,
город VARCHAR (20));

3) CREATE TABLE S
(№_поставщика VARCHAR (9) NOT NULL,
имя VARCHAR (25),
состояние FLOAT,
город VARCHAR (20)
PRIMARY KEY (№_поставщика));

2) CREATE TABLE S
(№_поставщика VARCHAR (9) NULL,
имя VARCHAR (25),
состояние FLOAT,
город VARCHAR (20)
PRIMARY KEY (№_поставщика));

4) CREATE TABLE S
(№_поставщика VARCHAR (9),
имя VARCHAR (25),
состояние FLOAT,
город VARCHAR (20)
FOREIGN KEY (№_поставщика));

5. Дана схема транзакции:

SET TRANSACTION READ WRITE

UPDATE 1

SAVEPOINT SA1

UPDATE 2

UPDATE 3

Если ошибка 1, то ROLLBACK TO SAVEPOINT SA1

UPDATE 4

Если ошибка 2, то ROLLBACK

COMMIT

Укажите изменения, которые будут иметь место при наличии первой ошибки

1) UPDATE 1, UPDATE 4

2) UPDATE 1, UPDATE 2, UPDATE 3, UPDATE 4

3) UPDATE 1, UPDATE 2, UPDATE 3

4) нет изменений

6. Укажите опасность, которой не может подвергаться база данных

1) похищение и фальсификация данных

2) утрата конфиденциальности

3) ограничение целостности

4) потеря доступности данных

Вариант 2

1. Укажите верный запрос: выдать имена поставщиков, которые имеют состояние от 5 до 25 у.е.

Поставщики S

№ поставщика	Имя	Состояние	Город
S1	Иванов	12	Москва
S2	Петров	20	Тула
S3	Сидоров	35	Москва

1) SELECT Имя

FROM S

WHERE Состояние = 5 AND Состояние = 25;

3) SELECT Имя

FROM S

WHERE Состояние >= 5 AND Состояние <= 25;

2) SELECT Имя

FROM S

WHERE Состояние >= 5 OR Состояние <= 25;

4) SELECT №_поставщика, Имя

FROM S

WHERE Состояние >= 5

AND Состояние <= 25;

2. Укажите неверный оператор, соответствующий запросу: выдать имена поставщиков, которые поставляют хотя бы одну черную деталь.

Поставщики S

№ поставщика	Имя	Состояние	Город
S1	Иванов	12	Москва
S2	Петров	20	Тула
S3	Сидоров	35	Москва
S4	Орлов	5	Курск

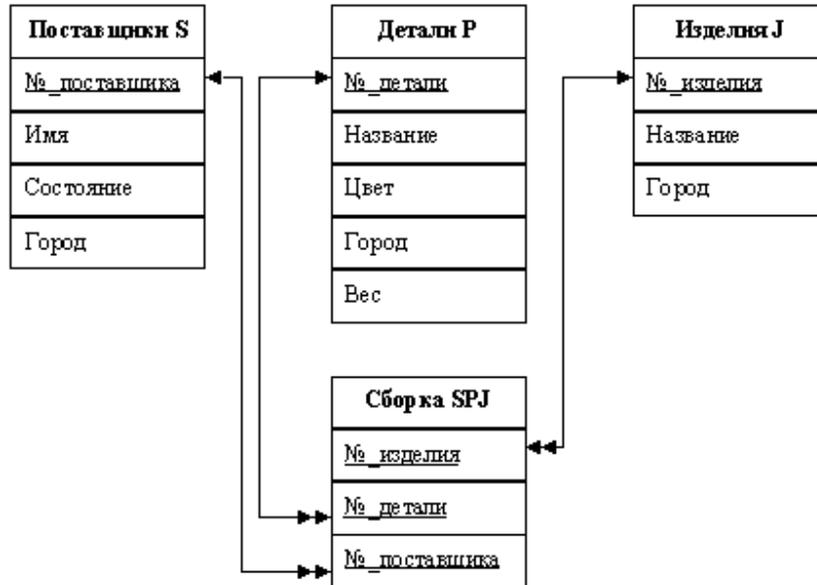
Сборка SPJ

№ поставщика	№ детали	№ изделия	Количество
S1	P1	J1	10 ⁷
S2	P2	J2	10 ⁵
S3	P3	J3	10 ³
S4	P2	J3	10 ⁵

Детали Р

<u>№ детали</u>	Название	Цвет	Вес	Город
P1	болт	черный	0,01	Москва
P2	гайка	черный	0,02	Москва
P3	винт	красный	0,03	Тула

Схема БД имеет вид



1) SELECT Имя
FROM S JOIN SPJ ON
S.№_поставщика = SPJ.№_поставщика,
SPJ JOIN P ON SPJ. №_детали = P. №_детали
WHERE Цвет = 'черный';

2) SELECT Имя
FROM S
WHERE №_поставщика IN
(SELECT №_поставщика
FROM SPJ
WHERE №_детали IN
(SELECT №_детали
FROM P
WHERE Цвет = 'черный'));

3) SELECT Имя
FROM S, (SELECT №_поставщика
FROM SPJ, P
WHERE Цвет = 'черный') M
WHERE S.№_поставщика = M.№_поставщика;

4) SELECT Имя
FROM S
WHERE EXISTS
(SELECT №_поставщика
FROM SPJ
WHERE №_детали IN
(SELECT №_детали
FROM P
WHERE Цвет = 'черный'));

3. Укажите выражение, которое возвращает истинное значение, если значение выражения *a* превосходит все значения в множестве *b*, возвращаемом после выполнения подзапроса

1) $a > \text{ANY} (\text{SELECT } b \text{ FROM } \dots \text{ WHERE } \dots)$

2) EXISTS
(SELECT ... FROM ... WHERE ...)

3) NOT EXISTS
(SELECT ... FROM ... WHERE ...)

4) $a > \text{ALL} (\text{SELECT } b \text{ FROM } \dots \text{ WHERE } \dots)$

4. Укажите верный оператор SQL определения таблицы Детали P:

Детали Р

<u>№ детали</u>	<u>название</u>	<u>цвет</u>	<u>вес</u>	<u>город</u>
P1	болт	черный	0,01	Москва
P2	гайка	черный	0,02	Москва
P3	винт	красный	0,03	Тула

1) CREATE TABLE P
(№_детали VARCHAR (9) PRIMARY KEY,
название VARCHAR (25),
цвет VARCHAR (15),
вес FLOAT,
город VARCHAR (20));

3) CREATE TABLE P
(№_детали VARCHAR (9) NOT NULL,
название VARCHAR (25) NOT NULL,
цвет VARCHAR (15),
вес FLOAT,
город VARCHAR (20) NOT NULL);

2) CREATE TABLE P
(№_детали VARCHAR (9) NULL,
название VARCHAR (25),
цвет VARCHAR (15),
вес FLOAT,
город VARCHAR (20),
PRIMARY KEY (№_детали));

4) CREATE TABLE P
(№_детали VARCHAR (9) DEFAULT NULL,
название VARCHAR (25) NOT NULL,
цвет VARCHAR (15),
вес FLOAT,
город VARCHAR (20)
PRIMARY KEY (№_детали));

5. Дана схема транзакции:

SET TRANSACTION READ WRITE

UPDATE 1

UPDATE 2

SAVEPOINT SA1

UPDATE 3

Если ошибка 1, то ROLLBACK TO SAVEPOINT SA1

UPDATE 4

Если ошибка 2, то ROLLBACK

COMMIT

Укажите изменения, которые будут иметь место при наличии первой ошибки

1) UPDATE 1, UPDATE 4

2) UPDATE 1, UPDATE 2, UPDATE 4

3) UPDATE 1, UPDATE 2, UPDATE 3

4) UPDATE 1, UPDATE 2, UPDATE 3, UPDATE 4

6. Укажите угрозу для защиты базы данных, которая направлена со стороны СУБД

1) незаконная модификация или копирование данных

2) использование средств доступа другого лица

3) распространение вирусов

4) отказ механизмов защиты

СОДЕРЖАНИЕ

1. Рабочая программа	3
2. График самостоятельной работы студентов	15
3. Краткий конспект лекций	16
4. Методические рекомендации по выполнению лабораторных работ	26
5. Методические указания по применению современных ИТ для преподавания учебной дисциплины	40
6. Задания для лабораторных работ	41
7. Фонд тестовых и контрольных заданий для оценки качества знаний по дисциплине	47

Лариса Владимировна ЧЕПАК
*доцент кафедры Информационных и управляющих систем АмГУ,
кандидат технических наук, доцент*

Проектирование баз данных для направления подготовки магистров
230100.68 – «Информатика и вычислительная техника»:
Учебно-методический комплекс дисциплины.

Издательство АмГУ.