Министерство образования и науки Российской Федерации
Российский государственный профессионально-педагогический университет
Уральское отделение Российской академии образования

В.В. Вьюхин, С.В. Супрун, Т.А. Кочнева

Базы данных

Лабораторный практикум

Часть 1

Допущено Учебно-методическим объединением по профессионально-педагогическому образованию в качестве учебного пособия для студентов высших учебных заведений, обучающихся по специальности 030500.06 Профессиональное обучение (информатика, вычислительная техника и компьютерные технологии)

Екатеринбург
2005

УДК 004.65 (076.5)

ББК 3973.2–018.2я73-5
В 96

Вьюхин В.В., Супрун С.В., Кочнева Т.А. Базы данных: Лаб. практикум:В 2 ч. Екатеринбург: Изд-во Рос. гос. проф.-пед. ун-та, 2005. Ч. 1. 67 с.

Лабораторный практикум содержит описание лабораторных работ по дисциплинам "Базы данных" и "Базы данных и управление ими", которые выполняются с использованием СУБД MySQL.

Предназначен для студентов специальностей 351400 Прикладная информатика (по отраслям) и 030500.06 Профессиональное обучение (информатика, вычислительная техника и компьютерные технологии).

Рецензенты: кандидат технических наук, доцент В.П. Подогов (Российский государственный профессионально-педагогический университет); доктор физико-математических наук А.В. Ким (Институт математики и механики УрО РАН)

Введение

Информационные системы, применяющие базы данных, представляют собой одну из важнейших областей современных компьютерных технологий. Знакомство с основами разработки и функционирования систем управления базами данных становится необходимым атрибутом подготовки специалистов в этой области.

Целью лабораторного практикума является ознакомление обучающихся с основными возможностями СУБД на персональных компьютерах, а также формирование практических навыков эксплуатации баз данных в условиях использования модели удаленного доступа к данным.

Знакомство с основами функционирования реляционных СУБД на персональных компьютерах проводится в первых трех лабораторных работах на примере использования пакета Foxpro2.

Большинство лабораторных работ первой части практикума посвящены изучению основ использования языка SQL в СУБД MySQL для модели «клиент–сервер», а также организации взаимодействия клиента с сервером. Однако все работы практикума могут быть выполнены при индивидуальном изучении дисциплин, если клиент и сервер MySQL будут установлены на одном компьютере.

Планируется создание второй части практикума, в которую войдут лабораторные работы, связанные с администрированием баз данных, обеспечением их безопасности и оптимизацией эксплуатации.

Лабораторные работы рекомендуется выполнять в том порядке, в котором они приведены в практикуме.

Прежде чем приступить к выполнению лабораторной работы, необходимо полностью ознакомиться с ее содержанием.

В некоторых лабораторных работах в рамках приводятся форматы команд (текстов запросов), которые могут использоваться при выполнении этих работ. Они исполняют роль справочных данных и должны быть тщательно изучены перед выполнением лабораторной работы. Рамками также выделены некоторые положения, на которые студентам следует обратить особое внимание. Часть справочных данных вынесена в приложения.

Выполнение каждой работы сопровождается созданием отчетного файла (отчета), в котором приводятся команды на создание запросов и результаты их выполнения. Все отчетные файлы собираются обучаемыми на дискетах по мере их выполнения. Имя отчетного файла (файла результатов) составляется из фамилии автора и номера работы.

Работа считается выполненной и зачтенной после ее защиты.

Лабораторная работа 1

ЗНАКОМСТВО С СУБД FOXPRO2. СОЗДАНИЕ БАЗ ДАННЫХ. ВЫБОРКИ

1. Ознакомьтесь с системой Foxpro2. Для этого:

- создайте папку для работы с Foxpro2;
- создайте внутри этой папки запускающий файл (например, fox.bat), содержащий единственную команду – команду запуска Foxpro2 из соответствующей папки;
- запустите файл fox.bat;
- познакомьтесь со структурой окна Foxpro2;
- используя справку (F1), познакомьтесь с типовой структурой команды Foxpro2, с компонентами команды Foxpro2 (командное слово, границы, список выражений, FOR <условие>, WHILE <условие>);
- изучите команды Create, List, Replace, Browse, Use, Select, Index, Set relation, Set skip, Modify Command;
- разберите примеры использования команд (с различными вариантами условий и математических и логических операций).

2. Создайте и заполните таблицу базы данных **Дом**. Для этого:

- создайте структуру для таблицы из полей:

– *Номер квартиры,*
– *Количество комнат,*
– *Метраж квартиры,*
– *Фамилия квартиросъемщика,*
– *Количество жильцов,*
– *Зарплата папы,*
– *Зарплата мамы,*
– *Пенсия* (суммарная),
– *Этаж,*
– *Кошка в квартире* (числовое),
– *Собака в квартире* (логическое),
– *Девочек в квартире* (числовое),
– *Мальчиков в квартире* (логическое),
– *Количество пенсионеров в квартире;*

- заполните таблицу базы данных как минимум на 10 строк, имея в виду, что для каждой квартиры предполагается обязательным наличие мамы и папы, а поле *Пенсия* заполняется только при наличии пенсионеров в квартире;

- дополните таблицу базы данных следующими полями:

– *Удельный метраж* (количество квадратных метров жилой площади на одного жильца),

– *Суммарный доход жильцов в квартире*,

– *Доход на одного жильца*,

– *Сумма зарплат мамы и папы*,

– *Количество детей в квартире* (количество жильцов минус количество пенсионеров минус два).

3. С помощью встроенного редактора (Modify Command <имя_файла>) создайте в командном файле команды выборок и получите следующие списки-выборки (каждая выборка должна содержать поле *Номер квартиры* и все поля, используемые при отборе записей):

- перечень квартир, в которых есть мальчики;
- перечень квартир, в которых имеются пенсионеры;
- перечень квартир, в которых есть дети;
- перечень квартир, в которых имеются животные;
- перечень квартир на 4-м этаже и выше, в которых проживают пенсионеры;
- перечень квартир, в которых на одного жильца приходится меньше 18 кв. м жилья;
- перечень квартир, в которых есть девочки и кошки;
- перечень квартир, в которых на каждого жильца приходится не менее чем по одной комнате;
- перечень квартир, в которых проживает не менее 5 жильцов;
- перечень квартир, в которых доход на одного жильца не превышает 5000 р.;
- перечень квартир, в которых имеются дети и пенсионеры, а доход на одного жильца не превышает 4000 р.

4. Для просмотра результатов создайте отчетный файл, в который командами "?" и "list" выведите соответственно тексты запросов на выполнение выборок и результаты их выполнения. Отчетный файл назначается в начале исполняемого командного файла с помощью команд

set printer to <имя_файла>
set print on

и отменяется с помощью команд

set print off
set printer to,

расположенных в конце командного файла.

Лабораторная работа 2

СВЯЗЫВАНИЕ ТАБЛИЦ "ОДНА С ОДНОЙ". СВЯЗЫВАНИЕ ТАБЛИЦ "ОДНА СО МНОГИМИ"

Правила выполнения

1. Каждый новый пункт задания выполняется и оформляется ниже в том же командном файле, как модернизация уже имеющихся выше вариантов команды.

2. В заголовках окон BROWSE (опция TITLE) обязательно указывать отличительный признак команды в соответствии с выполняемым пунктом.

Порядок работы

1. Создайте таблицу Список базы данных со следующими полями:

- *Табельный номер,*
- *Фамилия,*
- *Имя,*
- *Отчество,*
- *Дата рождения,*
- *Пол,*
- *Семейное положение,*
- *Количество детей,*
- *Код подразделения* (в котором работает человек),
- *Тарифный коэффициент* (по сетке ЕТС – от 2 до 10),
- *Надбавка.*

2. Введите минимум 5 записей.

3. Напишите и запустите команду BROWSE.

4. Напишите и отладьте команду BROWSE с опциями TITLE и FIELDS, перечислив в опции FIELDS все поля таблицы Список.

5. Напишите и отладьте команду BROWSE, обеспечивающую:

- русские названия имен полей;
- ограничение видимого размера для двух полей;
- контроль выхода из двух полей (заполненных ранее или заполненных вновь);
- режим чтения для двух полей;
- создание не менее 4 вычисляемых полей:

Ставка=600**Тарифный_коэффициент*+*Надбавка*,
Льгота (по подоходному налогу) =(1+*Количество детей*)*830.49,
Дотация = *Количество детей* *0.6*830.49,
Уральские=15 % от ставки.

6. Создайте таблицу **Нормативы** базы данных, разместите в ней следующие данные:

- *Минимальная ставка* (она же льгота по подоходному налогу, которая составляет 830.49 р.),
- *Дотация на одного ребенка* (процент от минимальной ставки, ориентировочно – 60%),
- *Районный коэффициент* (процент от ставки, ориентировочно – 15%).

7. Откройте таблицу **Нормативы** в любой свободной области и в соответствии с п. 6 отредактируйте команду BROWSE так, чтобы в окне появились вычисляемые поля, использующие при вычислении значений не константы, а имена полей из таблицы **Нормативы** (т.е. из другой области).

8. Создайте таблицу **Образование** базы данных со структурой:

- *Табельный номер*,
- *Что закончил* (наименование учреждения образования),
- *Когда закончил*,
- *Специальность*.

9. Свяжите таблицу **Образование** базы данных с таблицей **Список** по типу "одна с одной" и выведите все поля обеих таблиц (родительская таблица – **Список**).

10. Создайте таблицу **Дети** базы данных с полями:

- *Табельный номер родителя*,
- *Имя ребенка*,
- *Возраст ребенка*,
- *Пол ребенка*.

11. Заполните ее таким образом, чтобы у некоторых людей из таблицы **Список** базы данных было более одного ребенка.

12. Свяжите дополнительно таблицу **Дети** базы данных с таблицей **Список** по типу "одна со многими" (родительская таблица – **Список**) и выведите поля *Табельный номер, Фамилия, Дата рождения, Что закончил, Когда закончил, Имя ребенка, Возраст ребенка*.

Лабораторная работа 3

ПРОГРАММИРОВАНИЕ POPUP-МЕНЮ

1. Проверьте наличие или создайте вновь таблицу Список базы данных (см. лабораторную работу 2).

2. Проверьте наличие или создайте вновь таблицу Образование базы данных (см. лабораторную работу 2).

3. Проверьте наличие или создайте вновь таблицу Дети базы данных (см. лабораторную работу 2).

4. Создайте POPUP-меню со следующими пунктами:

- активизировать в первой области таблицу (файл) Список и показать его содержимое с помощью команды BROWSE;
- активизировать во второй области таблицу (файл) Образование и показать его содержимое с помощью команды CHANGE;
- активизировать в третьей области таблицу (файл) Дети, связать все три таблицы базы данных (в качестве родительской выбрать таблицу Список) и показать с помощью команды BROWSE поля *Табельный номер, Фамилия, Что закончил, Имя ребенка, Возраст ребенка*. Использовать, где это необходимо, множественную связь;
- активизировать подменю вывода следующих результатов:

– подпункт 1 – подсчитать количество мальчиков в таблице Дети;

– подпункт 2 – подсчитать сумму всех ставок сотрудников из таблицы Список;

– подпункт 3 – определить среднее значение ставки для всех сотрудников из таблицы Список;

- выход в командное окно Foxpro2.

5. Создайте POPUP-меню, в котором в качестве пунктов меню используются имена dbf-файлов в вашем рабочем каталоге (опция prompt files). Меню должно обеспечивать открытие в указанной области выбранного файла и показ его с помощью команды CHANGE.

6. Создайте POPUP-меню, в котором в качестве пунктов меню используются имена полей таблицы (файла) Список (опция prompt structure). При выборе пункта меню должна выполняться распечатка двух полей: поля *Табельный номер* и выбранного в меню поля для всех записей таблицы Список.

7. Создайте POPUP-меню, в котором в качестве пунктов меню используется содержимое поля *Фамилия* (опция prompt fields). Для выбранного с помощью меню человека необходимо распечатать всю запись из таблицы Список.

8. Результаты работы предъявите преподавателю.

Лабораторная работа 4

ДЕКОМПОЗИЦИЯ ПЛОСКОЙ ТАБЛИЦЫ С РАЗРАБОТКОЙ РЕЛЯЦИОННОЙ БАЗЫ ДАННЫХ

1. Выполните декомпозицию приведенной ниже плоской таблицы (или иной таблицы, предложенной преподавателем):

ФИО студента	Группа	Дата экзамена	ФИО преподавателя	Предмет
Иванов И.И.	Кт–102	12.12.98	Пушкин А.С.	Математика
Иванов И.И.	Кт–102	16.12.98	Ломов А.П.	Физика
Кочкин И.И.	Кт–102	20.12.98	Саахов Ш.С.	Химия
Петров П.П.	Ио–101	12.12.98	Пушкин А.С.	Математика
Павлов С.П.	Кт–104	16.12.98	Ломов А.П.	Физика
Жуков А.П.	Кт–104	20.12.98	Саахов Ш.С.	Химия
Черняк Д.И.	Ио–101	16.12.98	Пушкин А.С.	Математика
Кочев Р.В.	Кт–102	12.10.99	Ломов А.П.	Физика
Антонов В.И.	Кт–104	12.10.99	Саахов Ш.С.	Химия

Для этого необходимо:

1) разделить таблицу на части, соответствующие самостоятельным объектам;
2) разбить поля, содержащие группы значений;
3) ввести справочники для повторяющихся данных.

2. Нарисуйте шапку таблицы с полями:

- *Таб_номер,*
- *Фамилия,*
- *Имя,*
- *Отчество,*
- *Дата_рождения,*
- *Должность,*
- *Оклад,*
- *Дата_назначения_оклада,*
- *Имя_ребенка,*
- *Дата_рождения_ребенка,*
- *Дата_начала_больничного,*
- *Дата_окончания_больничного,*
- *Воинское_звание,*
- *Номер_страхового_свидетельства,*
- *ИНН.*

3. Заполните таблицу на 8 человек. При заполнении таблицы следует иметь в виду, что:

- люди из списка могут иметь одну из 4 специальностей (номенклатура специальностей по усмотрению студента);
- может быть использована следующая шкала воинских званий: младший сержант, сержант, старший сержант, старшина;
- поля *Номер_страхового_свидетельства* и *ИНН* – это 12-разрядные последовательности цифр.

4. Добавьте записи, с помощью которых вводится дополнительно следующая информация:

- для одного-двух человек отражается изменение оклада (с учетом дат);
- у двух человек имеется по два ребенка, у одного – три ребенка (учесть даты рождения детей и возраст родителя);
- трое человек из списка были на больничном листе за отчетный период по одному разу, один-два раза в разные сроки (дата возвращения с больничного должна быть позднее даты ухода на больничный). Больничные листы у одного и того же человека не должны перекрываться по срокам;
- один человек, находящийся в запасе, должен изменить воинское звание с сержанта на старшего сержанта, а затем на старшину (с учетом дат), два человека должны получить очередное воинское звание в пределах указанной номенклатуры званий.

Обратите внимание на существенное увеличение размеров таблицы.

5. В получившейся плоской таблице выявите группы полей (столбцов), которые могут быть выделены в отдельные таблицы, поскольку относятся к отдельным объектам, и данные, которые повторяются от записи к записи.

6. Составьте заголовки ("шапки") таблиц, на которые можно разбить созданную плоскую таблицу.

7. Заполните таблицы с этими заголовками, используя данные плоской таблицы.

8. Дополните полученный набор таблиц справочниками для должностей и воинских званий.

9. Выполните декомпозицию плоской таблицы по варианту, выданному преподавателем или методистом[1], не допуская повторения одних и тех же данных.

10. Определите количество элементов данных для плоских таблиц и их реляционных моделей для баз данных по пп. 1 и 9. Сравните результаты и сделайте выводы, помня о том, что количество записей в таблицах возрастает до значительных размеров (сотен – тысяч записей).

11. Предъявите преподавателю отчет, который должен содержать все плоские исходные таблицы и таблицы реляционных баз данных, а также результаты расчетов по элементам данных и выводы о сравнении исходных таблиц и реляционных баз данных.

[1] Можно использовать также "Методические указания по выполнению контрольных работ по дисциплине “Базы данных и управление ими”" (Екатеринбург, 2002. 21 с.).

Лабораторная работа 5

ОПЕРАЦИИ С ОТНОШЕНИЯМИ

Задание 1

1. Используйте в качестве исходных следующие отношения (таблицы) для знакомства с операциями:

ООО "Титан" (в формулах обозначается R1)

Номер зачетной книжки	Фамилия	Начислено
1	Иванов	2000
2	Петров	3000
3	Сидоров	2500
5	Коршунов	2000
7	Пашкевич	3000
8	Марков	2000

ООО "Гигант" (в формулах обозначается R2)

Номер зачетной книжки	Фамилия	Начислено
1	Иванов	2000
2	Петров	6000
2	Петров	3000
4	Кузнецов	3200
6	Антонов	4000

Предметы (в формулах обозначается R3)

Код предмета	Наименование предмета
1	Математика
2	Информатика
3	Русский язык

2. Выполните следующие операции, обратив внимание на смысловую сторону этих операций:

- объединение (R3=R1$\cup$R2);
- пересечение (R3=R1$\cap$R2);
- разность (R3=R1\R2).

3. Для отношений (таблиц) "ООО “Гигант”" и "Предметы" выполните операцию расширенного декартова произведения (R4=R2⊗R3).

4. Для отношения "ООО “Титан”" выполните:

- операцию выбора (фильтрации), отобрав записи со значением в поле *Начислено*, равным 2000;
- операцию выбора (фильтрации), отобрав записи со значениями в поле *Начислено*, соответствующими интервалу от 2300 до 3300, предварительно написав команду на выполнение операции;
- операцию проектирования, отобрав атрибут *Начислено*.

Внимание!

1. Формат операции выбора: $R_{рез}=R_{исх}$[условие отбора записей].
2. Формат операции проектирования: $R_{рез1}=R_{исх1}$[список полей].
3. Если при выполнении операции проектирования получаются одинаковые записи, то в результирующем отношении остается по одному экземпляру таких записей.

Задание 2

1. Создайте таблицу **Список** с полями *Табельный номер, Фамилия* и таблицу **Образование** с *полями Табельный номер, Учреждение, Специальность*.
2. Заполните таблицы **Список** и **Образование** (по 5 записей в каждой).
3. Выполните операцию расширенного декартова произведения.
4. Выполните условное соединение записей таблиц **Список** и **Образование** по условию совпадения табельных номеров. Сравните две последние операции и сделайте выводы.

Формат операции условного соединения: $R_{рез}=R_{исх1}$[условие соединения] $R_{исх2}$.

5. Создайте таблицу **Дети** с полями *Фамилия, Имя ребенка, Возраст ребенка* и заполните ее тремя-четырьмя записями (для одного человека – один ребенок, еще для одного – двое детей).
6. Выполните условное соединение записей таблиц **Список** и **Дети** по условию совпадения фамилий.
7. Выполните условное соединение записей таблиц **Список** и **Дети** (см. п. 6 задания) с таблицей **Образование** (по условию совпадения поля *Табельный номер*) и спроектируйте результат на атрибуты (поля) *Фамилия, Специальность, Возраст ребенка*.

Задание 3

1. Заполните отношения R7–R9 со схемами
R7=(*ФИО, Дисциплина, Оценка*),
R8=(*ФИО, Группа*),
R9=(*Группа, Дисциплина*)
не менее чем десятью кортежами (записями) каждое.

2. Составьте следующие запросы:

- список студентов, которые сдали на отлично экзамен по предмету (указывает преподаватель);
- список тех, кто сдавал экзамен по предмету (указывает преподаватель);
- список тех, кто имеет несколько одинаковых оценок (указывает преподаватель);
- список групп, студенты которых не имеют неудовлетворительных оценок;
- список круглых отличников.

Лабораторная работа 6

ЗАПУСК КЛИЕНТА MYSQL В РЕЖИМЕ КОМАНДНОЙ СТРОКИ

1. Запустите программу FAR. Перейдите в папку, содержащую файлы клиентов MySQL (папка sql или mysql или другая указанная преподавателем папка). Ознакомьтесь со списком клиентов MySQL.

Система управления базами данных MySQL включает:
• *сервер:*
mysqld – работает в фоновом режиме, принимает запросы от клиентов;
• *группу основных клиентов:*
mysqlaccess – Perl-сценарий, проверяет привилегии для заданных: узла, пользователя и базы данных
mysqladmin – выполняет ряд административных задач. С помощью утилиты на сервер посылаются команды, обычно разрешенные лишь администратору базы данных
mysqldump – извлекает информацию из указанной базы данных
mysqlimport – импортирует записи в таблицы из указанного текстового файла
mysqlshow – возвращает информацию о базах данных и таблицах
perror – возвращает описание числового кода ошибки

2. Проверьте подключение сервера MySQL следующей командой (если такая команда содержится в указанной папке):

mysqladmin.exe --host=194.226.238.70 --port=3306 --user=<имя> ping

(где <имя> – имя пользователя на сервере).

Факт функционирования сервера подтверждается сообщением mysqld is alive.

При запуске клиента именем пользователя является сочетание символов user и числового номера, выданного студенту преподавателем, например: user15, user3, user99.

3. Если сервер не работает, следует обратиться к преподавателю или администратору сети.

4. Уточните у преподавателя пароль и имя пользователя для работы с сервером (клиентом) и запустите клиента MySQL командой командной строки

mysql.exe --host=194.226.238.70 --port=3306 --user=<имя>

Формат запуска клиентов MySQL (прил. 1)

mysql

--host=узел (-h узел) – задает узел, к которому будет подключаться клиент

--database=имя (-D имя) – задает базу данных, с которой будет вестись работа

--defaults-file=файл – переопределяет стандартный конфигурационный файл

--execute=команда (-e команда) – заставляет клиента выполнить указанную команду (команда заключается в одинарные кавычки)

--force (-f) – заставляет клиента продолжать обработку данных даже в случае обнаружения ошибки

--password (-p) – запрашивает пароль на подключение утилиты

--port=порт (-P порт) – переопределяет стандартный номер порта, с которым работает клиент (по умолчанию – 3306),

--skip-column-names (-N) – означает, что названия столбцов не следует включать в результаты запроса

--table (-t) – результаты выдаются в виде таблицы

--user=имя (-u имя) – задает имя пользователя при регистрации на сервере

--verbose (-v) – заставляет клиента выдавать более подробные сообщения

--vertical (-E) – заставляет клиента выдавать результаты в вертикальном виде

Внимание!

Полные слова опции вводятся после двух тире без пробела, а короткие (в виде букв) – после одиночного тире. Буквы коротких команд следует набирать с учетом регистра.

При коротком формате приведенная в п. 4 команда запуска клиента должна выглядеть следующим образом:

mysql.exe –h 194.226.238.70 –P 3306 –u <имя>

Если запуск клиента затруднен, необходимо уточнить у преподавателя <имя>.

Для того чтобы получить подробную справку о клиенте, необходимо запустить клиента mysql с дополнительной опцией --help. Однако в связи с большой длиной справки на экране останется только ее конечная часть. Можно переадресовать вывод справки в файл. Тогда формат команды будет выглядеть следующим образом:

mysql.exe –h 194.226.238.70 –P 3306 –u <имя> --help > file_name

(где file_name – имя файла результатов).

Внимание!

1. Аналогичным образом можно получить справку о других клиентах MySQL и даже о сервере mysqld.

2. Опция help должна быть набрана строчными (маленькими) буквами.

5. Запустите клиента mysql и выведите справку в указанный вами файл. Проанализируйте информацию в файле.

6. Запустите клиента mysql без указания файла результатов (и вы попадете в интерактивную программу, допускающую запуск отдельных инструкций на языке sql). Далее выполните следующее:

- введите команду help и проанализируйте полученную информацию;
- выйдите из программы-клиента mysql, набрав команду quit.

Внимание!

1. Если необходимо, чтобы при запуске выполнялись инструкции текстового файла, надо после имени клиента с опциями поместить имя текстового файла с предваряющим символом "<", например:

mysql.exe --host=194.226.238.70 --port=3306 --user=<имя> < file_instr

2. Возможно одновременное использование файлов-инструкций и файлов результатов (file_out):

mysql.exe --host=194.226.238.70 --port=3306 --user=<имя> < file_instr > file_out

7. Используя инструкции из текстовых файлов (< file_instr) и перенаправление вывода информации (> file_out) после соответствующей команды, получите информацию по опциям help и status (эти опции можно давать как инструкции в текстовом файле).

8. Введите несколько инструкций из числа полученных в п. 6. данной работы (по запросу help). Выполните запуск клиента и проанализируйте результаты.

9. Введите в листинг комментарии перед каждой строкой. Проверьте возможность комментирования программного кода (символами или группами символов "#", "/*", "-- "). Сделайте выводы по способам комментирования.

10. Если сервер (в связи с проблемами организации сетевого доступа) запускался на период выполнения лабораторной работы, то завершите его работу командой (если такая команда доступна)

mysqladmin.exe --host=192.168.6.48 --port=3306 --user=<имя> shutdown

Внимание!

Последовательность инструкций командной строки можно оформлять в виде bat-файла, который может содержать как сами команды, так и комментарии по каждой отдельной команде или по наборам команд.

11. Предъявите преподавателю в качестве отчета конспект с информацией о использованных командах и файл с именем file_out. Конспект обязательно должен содержать описание всех опций утилиты mysql.

Внимание!

1. Начиная со следующей лабораторной работы, при запуске клиента mysql обязательно использование опций --table и --verbose.

2. Никаких дополнительных комментариев в отчетный файл передавать не следует.

Лабораторная работа 7

РАБОТА В ПАКЕТНОМ РЕЖИМЕ. ПРОСТЕЙШИЕ SELECT-ЗАПРОСЫ

1. Конфигурируйте рабочее место для выполнения лабораторного практикума в MySQL, для чего:

- создайте свою папку (каталог) внутри папки Temp (или Рабочая);
- создайте bat-файл, запускающий из вашей папки программу-клиент mysql, расположенную в папке SQL (или в иной указанной преподавателем папке);

Пример bat-файла

```
PATH c:\mysql\bin;d:\Temp\Ivanov
mysql.exc –h 194.226.238.70 –u Ivanov  < file_instr > file_out
```

- создайте в своей папке файл запросов (Shift+F4). Файл будет заполняться инструкциями SQL по мере выполнения очередной лабораторной работы. Имя файла должно включать в себя символы "IN", номер лабораторной работы, фамилию студента и номер группы (например, "IN6_Ivanov_KT–401"). Файлы вывода должны иметь аналогичные названия, но вместо "IN" должно стоять "OUT" (например, "OUT6_Ivanov_KT–401").

Внимание!

1. Имена папок и файлов не должны содержать пробелов.
2. Желательно, чтобы имена файлов записывались латинскими буквами.

2. Выберите учебную базу данных (USE имя_базы_данных). Имя базы данных уточните у преподавателя (если не указано другое, то это имя EDUCATION). Список таблиц и перечень полей в таблицах для базы данных EDUCATION приведен в прил. 2.

3. Перенаправьте вывод в файл (обеспечивая перезапись (">"), или дописывание (">>") в конец файла). Подготовьте и покажите преподавателю файлы, полученные отдельно перезаписью и дописыванием, отобразив во втором случае последовательные этапы применения одной и той же команды (например, SELECT) с постепенным наращиванием списка параметров (например, количества выводимых столбцов).

4. Предъявите преподавателю файл программного кода для получения нижеперечисленных выборок, а также файл с результатами всех запросов. Вы

должны уметь объяснить назначение каждого компонента во всех командах программного кода.

Упрощенный формат команды выбора SELECT[2]

```
SELECT [ALL | DISTINCT] <Список полей>|*
    FROM <Список таблиц>
    [WHERE <Предикат–условие выборки или соединения>]
```

Список выборок

1. Вывести список всех студентов, занесенных в базу данных.
2. Вывести фамилию, курс и дату рождения для всех студентов базы данных.
3. Вывести идентификатор (т.е. код), фамилию, имя и отчество для всех преподавателей.
4. Вывести список всех предметов обучения.
5. Вывести список всех институтов (факультетов).
6. Вывести список студентов, а также кодов институтов, в которых эти студенты обучаются.
7. Вывести список всех студентов, получающих стипендию.
8. Вывести список студентов первого курса.
9. Вывести список студентов второго и третьего курсов.
10. Вывести список студентов, которые обучаются на первом и третьем курсах (студентов второго курса в список включать не надо).
11. Вывести список идентификаторов и названий предметов, по которым получены отличные оценки (без повторов).
12. Вывести список всех возможных значений стипендии (без повторов).
13. Вывести список номеров всех курсов, на которых учатся студенты, занесенные в базу данных (без повторов).
14. Вывести без повторов список, включающий наименования предметов и количество часов на их изучение по тем дисциплинам, по которым студентами получена хотя бы одна двойка.
15. Вывести список студентов с указанием наименований предметов и экзаменационных оценок.
16. Вывести список студентов, получивших неудовлетворительную оценку по химии. Список должен включать фамилии студентов, название предмета и оценку.
17. Вывести список студентов-двоечников с указанием наименований несданных предметов и дат неудачных сдач.

[2] Более полный формат команды Select приведен в прил. 3.

18. Вывести список всех студентов, которые получили неудовлетворительные оценки 24 мая 2003 г. или 13 июня 2003 г.

19. Создать запрос на получение списка фамилий студентов, сдавших экзамены на "отлично", с указанием дат сдачи и наименований предметов, а также фамилий преподавателей, принимавших экзамены.

20. Составить список институтов (факультетов), преподаватели которых имеют по дисциплинам семестровую нагрузку выше 80 ч, но тем не менее у студентов имеются неудовлетворительные оценки по дисциплинам этих преподавателей. Список должен содержать также названия дисциплин, фамилии студентов, данные о семестровой нагрузке и экзаменационных оценках.

Лабораторная работа 8

ЗАПРОСЫ С ИСПОЛЬЗОВАНИЕМ ОПЦИЙ IN, BETWEEN, LIKE, IS NULL

1. Выполните следующие запросы, используя опции IN и BETWEEN:

- список сданных экзаменов для трех студентов (на выбор). Список должен включать фамилии студентов, названия предметов и оценки (использовать опцию IN);
- список студентов, имеющих одно из двух указанных вами имен (использовать опцию IN). Список должен включать фамилии и имена студентов;
- список студентов, обучающихся на любых двух курсах (использовать опции IN и BETWEEN);
- список студентов двух любых курсов вуза за исключением студентов трех выбранных вами институтов (факультетов). Список должен включать фамилии и имена студентов, номера курсов и наименования институтов (факультетов);
- список фамилий студентов, начинающихся с букв из указанного вами интервала (использовать опцию BETWEEN);
- список студентов, родившихся в указанный интервал месяцев (границы интервала вы устанавливаете сами). Рекомендуется использование функции MONTH(date), возвращающей номер месяца года (IN и BETWEEN);
- список студентов, получающих одну из двух стипендий фиксированного уровня (например, 400 или 600 р).

2. Выполните следующие запросы, используя опцию LIKE:

- список фамилий студентов, начинающихся на две любые указанные вами буквы;
- список фамилий студентов, оканчивающихся на любое заданное вами сочетание букв (длина сочетания 3 – 4 символа);
- список фамилий студентов, у которых третья и четвертая с начала фамилии буквы имеют определенное сочетание (задается обучаемым);
- список фамилий студентов, у которых третья и четвертая с конца фамилии буквы имеют определенное сочетание (задается обучаемым);
- список фамилий, имен и отчеств преподавателей, у которых первая буква фамилии и вторая буква имени соответствуют буквам, заданным обучаемым;
- список фамилий и имен студентов, у которых имена состоят из определенного количества букв (это количество в разумных пределах задается вами).

3. Определите, в каких полях таблицы Exam содержится значение NULL.

4. Обеспечьте предъявление команд и вывод результатов запросов в одном отчетном файле.

5. Предъявите отчетный файл преподавателю. Объясните результаты.

Лабораторная работа 9

ПРЕОБРАЗОВАНИЕ ВЫВОДА И ВСТРОЕННЫЕ ФУНКЦИИ

1. Ознакомьтесь с назначением и форматами функций (прил. 4).

2. Выполните преобразование числовых данных с использованием функций SIN(), FLOOR(), CEILING(), EXP(), SQRT(), ROUND(), MOD(), LOG().

Аргументами функций могут быть числовые константы, числовые поля таблиц (но не поля идентификаторов записей) или выражения, содержащие эти константы и поля. При этом обязателен вывод аргумента, выражения и результата применения функции.

Тексты запросов и соответствующие им результаты выведите в одном отчетном файле. Аргументы используемых функций определяются студентом.

Для функций FLOOR(), CEILING(), ROUND(), MOD() аргументы должны иметь по две-три цифры слева и справа от десятичной точки. Если такие аргументы в базе данных отсутствуют, рекомендуется использование вложенных функций, при этом любые промежуточные результаты, применяемые в качестве аргументов, также должны распечатываться.

3. Преобразуйте текстовые данные с использованием функций объединения CONCAT(), вхождения INSTR(), LEFT(), дополнения LPAD(), удаления начальных пробелов LTRIM(), замены подстроки REPLACE(), удаления начальных (LTRIM()) и конечных (RTRIM()) пробелов, возвращения строки пробелов SPACE(), а именно:

- выведите фамилии с инициалами всех студентов, у которых фамилии начинаются с той же буквы, что и у вас (если таких фамилий не обнаружится, то начинающихся с ближайшей следующей буквы алфавита):

– используя только LEFT(),

– используя CONCAT() и LEFT(),

– используя CONCAT() и LEFT() с вставкой точек после инициалов и пробела после фамилии;

- определите вхождение первой буквы вашей фамилии (строчной) в названия предметов обучения;
- создайте строку вида ('################Иванов############'), используя в качестве центрального элемента фамилию преподавателя (из одноименной таблицы), а также функции LPAD(), RPAD(), LENDTH(). Общая длина строки – 6-кратная длина фамилии обучаемого, фамилия должна быть в середине этой строки;
- замените любую выбранную вами подстроку в поле *Subj_name* на вашу фамилию.

4. Преобразуйте данные типа даты с помощью функций возвращения текущей даты CURDATE(), CURRENT_DATE(), CURRENT_TIME(), NOW() (используя дополнительные команды SELECT), добавления времени к значению даты DATE_ADD(), DAYNAME(), DAYOFMONTH(), DAYOFWEEK(), DAYOFFYEAR(), MONTH(), используя в качестве основы для преобразований любое поле типа даты (например, *Дата рождения студента* или *Дата сдачи экзамена*). Количество записей в любой выборке не должно превышать двух.

5. Определите количество месяцев между датами рождения первого и последнего в списке студентов (в таблице Student). Даты определить дополнительными запросами, используя поле *Student_id* (функция TO_DAYS() и встроенное присваивание вида @var:=выражение).

6. Используя функцию IF(), выведите данные о размере стипендии для студентов первого и второго курсов и дату рождения студентов третьего курса.

7. Используя функцию CASE(), помимо фамилии студента выведите в зависимости от номера курса размер стипендии, город проживания, дату рождения, идентификационный номер студента или код института (факультета).

Лабораторная работа 10

АГРЕГИРОВАНИЕ И ГРУППОВЫЕ ФУНКЦИИ

1. Подсчитайте количество записей в таблице Student.

2. Подсчитайте количество записей в таблице Student, не содержащих значений NULL.

3. Подсчитайте количество записей в выборке (COUNT(имя столбца)) с использованием и без использования условий отбора (условие выбирается обучаемым, но в отчетном файле должно быть приведено не менее трех условий разного типа).

4. Определите с помощью функций MIN(), MAX() или AVG() по двум-трем полям (поля выбираются вами или указываются преподавателем):

- длину самой короткой и самой длинной фамилии студентов, начинающуюся на ту же букву, что и ваша фамилия;
- среднюю длину наименования предметов, округленную до целого числа;
- наибольшее суммарное количество символов в фамилии, имени и отчестве студентов;
- средний возраст студентов первого курса (с точностью до десятой доли года).

5. Выведите результаты применения групповых функций по всем числовым полям учебной базы данных. Идентификационные коды к таковым не относятся.

6. Используя группировку (GROUP BY ...), получите данные о количестве студентов всех курсов. Выведите номера курсов и количество студентов на каждом курсе. Выполните сортировку выводимых данных сначала по возрастанию, а затем по убыванию, используя следующие способы задания столбцов сортировки:

- по имени столбца,
- по псевдониму,
- по номеру столбца.

7. Получите сведения о среднем балле успеваемости для студентов первого-третьего курсов. Список упорядочить по убыванию номера курса. Выведите номера курсов и значения среднего балла.

8. Выполните действия, указанные в п. 7. Список упорядочите по убыванию значения среднего балла.

9. Выведите сведения о среднем балле успеваемости студентов, получающих стипендию, любого выбранного вами курса.

10. Получите сведения о величине среднего балла оценок, выставляемых преподавателями студентам первого курса. Выведите фамилии преподавателей и значения среднего балла.

11. Выведите фамилии преподавателей, у которых среднее значение выставляемых оценок превышает 3.

12. Выведите данные о суммарной величине стипендии по годам рождения студентов.

13. Определите количество преподавателей для различных институтов. Выведите количество преподавателей и названия институтов.

14. Определите количество месяцев между датами рождения самого младшего и самого старшего из студентов (в таблице Student). Даты определить дополнительными запросами, используя поле *Student_id* (функция TO_DAYS() и встроенное присваивание вида @var:=выражение).

15. Отчетный файл (с обязательным выводом инструкций и результатов) предъявите для проверки преподавателю.

Лабораторная работа 11

СОЗДАНИЕ И УДАЛЕНИЕ ОБЪЕКТОВ БАЗЫ ДАННЫХ

Внимание!
При любом изменении структуры таблицы или добавлении/удалении записей необходимо выводить содержание всей таблицы в отчетный файл.

1. По конспекту лекций изучите синтаксис команд удаления и создания (DROP и CREATE) баз данных, таблиц, индексов и функций, а также наиболее широко используемые типы данных (целые и действительные числовые, строковые, даты), а также атрибуты полей и рекомендации по их использованию.

2. Установите командой USE текущую базу данных. Имя этой базы данных совпадает с именем пользователя (при выполнении практикума в аудиториях РГППУ).

3. Изучите синтаксис команд удаления и вставки записей (DELETE, INSERT).

4. Создайте в базе данных таблицу Avto с полями: *Номер записи, Марка автомобиля, Год выпуска, Дата покупки, Фамилия покупателя* (CREATE TABLE).

5. Введите 5 записей (INSERT). Дважды отсортируйте таблицу по какому-либо полю (на ваш выбор), задавая это поле сначала именем, а затем псевдонимом.

6. Удалите первые три записи таблицы Avto (DELETE).

7. Удалите остальные записи таблицы Avto (DELETE).

8. Создайте таблицу Tovar с полями: *Номер записи, Наименование изделия, Марка изделия, Дата выпуска, Цена, Дата покупки, Фамилия покупателя*. Поле *Номер записи* – первичный ключ, заполняемый автоматически. Поле *Дата покупки* заполняется по умолчанию (соответствует дате проведения занятия). Все поля (за исключением поля *Дата выпуска*) не допускают значения NULL.

9. Введите в таблицу Tovar 8 записей. Сформулируйте запрос и выведите все записи из таблицы. Вывод выполните по двум вариантам: по возрастанию и убыванию, задавая столбец сортировки по его номеру.

10. Изучите синтаксис и варианты использования команды SHOW (см. прил. 3). С помощью этой команды предъявите следующие данные:

- о двух-трех колонках двух последних таблиц;
- о запросе, создающем таблицу;
- о индексах и таблицах;
- о статусах таблиц.

11. Создайте виртуально базу данных HOSPITAL. (Это делается внутри базы данных ИМЯ_ПОЛЬЗОВАТЕЛЯ). Для этого:

- Создайте таблицу Pacient с полями соответствующих типов, предусмотрев поля, которые должны заполняться автоматически:

– *Код пациента,*
– *Фамилия пациента,*
– *Пол пациента,*
– *Дата рождения пациента,*
– *Домашний адрес пациента,*
– *Код лечащего врача,*
– *Диагноз,*
Номер палаты.

- Создайте таблицу Doctor с полями:

– *Код лечащего врача,*
– *Фамилия лечащего врача,*
– *Адрес лечащего врача,*
– *Телефон лечащего врача.*

- Создайте таблицу Palata с полями:

– *Номер палаты,*
– *Этаж,*
– *Отделение,*
– *Количество мест,*
– *Вид палаты* (мужская, женская).

12. Введите записи в таблицы базы данных HOSPITAL с помощью INSERT:

- во все поля записей всех трех созданных баз (10 записей в первую таблицу и по 5–6 записей во вторую и третью таблицы),
- в отдельные поля записей (3–5 записей).

13. Проанализируйте содержание полей различного типа, допускающих (NULL) и не допускающих (NOT NULL) наличие неопределенных значений для случаев, когда в эти поля вводятся или не вводятся какие-то данные.

14. Создайте индексы для таблиц базы данных:

- по одному полю (например, по полю *Фамилия пациента*);
- по двум полям (например, по полям *Фамилия пациента* и *Дата рождения пациента*).

15. Посмотрите информацию об индексах (SHOW INDEX). Проанализируйте эту информацию.

16. Выполните запросы, обеспечивающие вывод данных из полей связанных таблиц:

- фамилии пациентов и их лечащих врачей;
- фамилии врачей, поставивших указанный вами диагноз;
- фамилии лечащих врачей, поставивших этот диагноз пациентам мужского (или женского) пола старше (или младше) заданного вами возраста. В список кроме фамилий врачей включить фамилии пациентов, их пол и возраст;
- фамилии всех пациентов какого-либо отделения;
- фамилии пациентов, которым поставлен диагноз «пневмония», а также фамилии врачей, поставивших этот диагноз, и номера палат, в которых лежат больные с данным диагнозом;
- количество пациентов мужского и женского пола, приходящихся на каждого лечащего врача;
- количество больных, приходящихся на каждого лечащего врача, и их средний возраст.

17. Предъявите файл результатов преподавателю.

Лабораторная работа 12

МАНИПУЛИРОВАНИЕ ДАННЫМИ

1. Добавьте записи в таблицу Tovar с помощью серии команд INSERT, внося информацию:
   - во все поля записей (5 записей);
   - в отдельные поля записей (3–5 записей);
   - в те же поля, что и в предыдущем пункте, но в другом порядке;
   - в одну конкретную запись (по выбору преподавателя).

2. Проанализируйте содержание записей (для каждого поля), в которых данные введены не во все поля.

3. Добавьте в таблицу Tovar с помощью одной команды INSERT серию из 8 записей.

4. Удалите 5 строк таблицы по одной пятью отдельными командами DELETE с разными условиями.

5. Восстановите все записи таблицы.

6. Удалите все введенные записи из таблицы с помощью команды DELETE.

7. Повторно введите данные во все поля таблицы таким образом, чтобы нумерация записей начиналась с 500.

8. Удалите все записи.

9. Повторно введите 10 записей, половина которых пронумерована числами из интервала от 100 до 200, а половина – из интервала от 200 до 300.

10. Удалите записи по условию, используя предложение WHERE, двумя-тремя отдельными командами с разными условиями.

11. Создайте таблицу Tovar1 со структурой, аналогичной структуре таблицы Tovar.

12. Используя вложенный запрос, введите в таблицу Tovar1 все записи из таблицы Tovar, относящиеся к товару с выбранным вами наименованием изделия (INSERT … SELECT …).

13. Обновите содержимое двух-трех полей по вашему выбору инструкцией UPDATE с использованием предложения SET.

14. Обновите содержимое двух-трех записей инструкцией UPDATE. Записи отбирайте с помощью предложения WHERE <условие>. Смысл условия определите сами.

15. Обновите значения 5 различных полей в пяти различных записях (по одному полю на запись). Условия отбора должны быть различными для каждой записи и для каждого поля.

16. Обновите одно поле в трех записях, используя опцию LIMIT.

17. Тексты инструкций и состояние базы данных после выполнения каждой инструкции выводите в файл результатов.

18. Файл результатов предъявите преподавателю.

Лабораторная работа 13

ОГРАНИЧЕНИЯ НА МНОЖЕСТВО ДОПУСТИМЫХ ЗНАЧЕНИЙ ДАННЫХ

1. Создайте и заполните 5 записями таблицу Diagnoz, второй и четвертый столбцы которой (см. список полей по порядку) допускают неопределенные значения. Поля таблицы:
   - *Код пациента,*
   - *Диагноз,*
   - *Температура тела,*
   - *Артериальное давление,*
   - *Пульс,*
   - *Частота дыхания.*

2. Сосчитайте количество записей в таблице с помощью:
   - COUNT(*),
   - COUNT(имя поля), используя поля *Диагноз* или *Артериальное давление*,
   - COUNT(имя поля), используя любое другое поле.

3. Создайте и заполните 10 записями таблицу Worker, предусматривающую ввод значений по умолчанию (DEFAULT) для последних четырех полей приведенного ниже списка (значения по усмотрению студента):
   - *Табельный номер* (должно обеспечиваться автоматическое увеличение значения поля),
   - *Фамилия, И.О.* (заполнение обязательно),
   - *Пол,*
   - *Место работы,*
   - *Должность,*
   - *Дата приема на работу.*

4. Измените команду CREATE, обеспечив для полей *Должность* и *Пол* задание значений с помощью ENUM. Заполните таблицу пятью записями, вводя поля *Пол* и *Должность* по умолчанию. Сформулируйте правила действия системы при вводе данных по умолчанию с использованием ENUM.

5. Модифицируйте структуру таблицы Pacient (назовите ее Pacient1) и заполните последнюю пятью записями, предусматривая заполнение только уникальных комбинаций полей (фамилии пациента и даты его рождения). Проанализируйте, что будет при повторном вводе уникальной комбинации полей.

6. Файл результатов должен содержать все команды и результаты выборок. Предъявите этот файл преподавателю и объясните результаты.

Лабораторная работа 14

ИЗМЕНЕНИЕ И УДАЛЕНИЕ ТАБЛИЦЫ

1. Создайте и заполните:

- таблицу Car с полями *Марка, Год выпуска, Пробег, Цена*;
- таблицу School с полями *Код* (начальная, средняя школа), *Город, Улица, Номер дома*.

2. Измените таблицу Car с помощью команды ALTER, обеспечив:

- добавление одного столбца в таблицу (ADD) с установкой столбца первым;
- добавление одного столбца в таблицу (ADD) с установкой столбца в нужное место (по усмотрению обучаемого);
- добавление трех столбцов в таблицу (ADD);
- изменение спецификации (типа) для двух столбцов (CHANGE);
- изменение названий для двух столбцов (CHANGE);
- добавление и удаление первичного ключа по выбранным столбцам (ADD):

– по одному столбцу;
– по двум столбцам;
– добавление первичного ключа при уже имеющемся первичном ключе. Посмотрите и осмыслите поведение системы в последнем случае;

- удаление одного-двух столбцов из имеющейся таблицы (DROP);
- добавление и удаление индексов (ADD и DROP);
- изменение для одного поля значения по умолчанию (...ALTER ...SET...) и отмену значения по умолчанию для другого поля (...ALTER ...DROP DEFAULT);
- переименование таблицы (новые имена таблиц по выбору вашему);
- ввод данных в указанные вами поля в ограниченное количество записей (UPDATE).

Внимание!

После каждого изменения состояния таблицы не забывайте выводить в отчетный файл структуру и содержание таблицы.

3. Выполните аналогичные операции с таблицей School.

4. Предъявите результаты преподавателю в отчетном файле. Объясните результаты.

Лабораторная работа 15

ПОДДЕРЖКА ЦЕЛОСТНОСТИ ДАННЫХ. ИСПОЛЬЗОВАНИЕ КЛЮЧЕЙ

1. Создайте и заполните записями на пять человек таблицы Worker (*Табельный номер, Фамилия, Имя, Отчество, Дата рождения, Улица, Дом, Квартира*) и Slugba (*Табельный номер, Отдел, Должность, Разряд, Оклад, Дата назначения*).

2. Выполните запрос о движении человека (по вашему выбору) по служебной лестнице (данные выбираются из таблицы Slugba для одного человека и должны быть отсортированы по датам).

3. Выберите все данные о людях с окладом из установленного интервала и с указанной должностью (интервал окладов и должность вы устанавливаете сами).

4. Выполните запрос о сотрудниках какого-либо отдела, проживающих на указанной вами улице.

5. Создайте обычный индекс (INDEX, KEY) для обеих таблиц и выполните запрос с выбором проиндексированных полей из связанных таблиц.

6. Добавьте в таблицу Worker поле *Корпус*.

7. Увеличьте количество записей таблицы Slugba до 30 (тридцати), введя несколько дополнительных записей для каждого сотрудника.

8. Создайте уникальные ключи (UNIQUE) для таблиц. Попытайтесь добавить в таблицу Worker запись сначала о человеке с новым табельным номером, а затем с уже имеющимся в таблице табельным номером. Введите изменения, допускающие ввод данных с повторяющимся значением ключа в таблицу Slugba.

9. Переопределите таблицу Worker, введя первичный ключ (PRIMARY KEY). Выведите данные таблицы, упорядоченные по первичному ключу, в порядке возрастания, а затем убывания (ORDER BY <имя_поля> ASC/DESC). В качестве первичного ключа используйте по очереди поля *Фамилия, Имя, Дата рождения*.

Внимание!

1. При каждом изменении структуры или данных в таблицах следует выполнять распечатку данных таблицы в отчетный файл.

2. После каждой распечатки таблицы командой SHOW (см. прил. 3) выведите информацию о индексах и статусе таблицы.

10. С помощью команды SHOW GRANTS покажите привилегии, предоставленные пользователю.

11. Предъявите результаты для проверки преподавателю.

Лабораторная работа 16

СПОСОБЫ ЗАГРУЗКИ ДАННЫХ

1. Познакомьтесь с инструкцией чтения данных из текстового файла LOAD DATA INFILE (см. прил. 3).

2. Создайте таблицу Vvod (*Номер, Фамилия, Дата рождения, Пол*).

3. Создайте текстовый файл (имя файла на ваше усмотрение) для ввода данных в таблицу Vvod. При создании текстового файла по умолчанию значения данных внутри записи отделяются друг от друга символом табуляции, а записи друг от друга – переводом строки (ENTER). Никаких дополнительных сведений внутри файла данных быть не должно.

4. Составьте инструкцию для прочтения файла и загрузки данных в таблицу Vvod (LOAD DATA INFILE).

5. Введите в таблицу не менее 7 записей.

6. Создайте таблицу Vvod1 с полями *Номер, Марка компьютера, Фирма-изготовитель, Год выпуска, Объем оперативной памяти, Объем жесткого диска, Быстродействие*.

7. Создайте текстовый файл для ввода 8 записей в таблицу Vvod1.

8. Создайте инструкцию для прочтения файла и загрузки данных в таблицу Vvod1.

9. Заполните таблицу Vvod1 8 записями с помощью инструкции LOAD DATA INFILE.

10. Создайте таблицу Vvod2 с полями *Автор, Название произведения, Год издания, Издательство, Количество страниц, Тираж.*

11. Создайте текстовый файл для ввода 8 записей в таблицу Vvod2.

12. Заполните ее 8 записями с помощью инструкции LOAD DATA INFILE.

13. Создайте таблицу Vvod2_1 со структурой, аналогичной структуре Vvod2.

14. Используя вложенный запрос, введите в таблицу Vvod2_1 все записи из таблицы Vvod2, относящиеся к товару с выбранным вами интервалом значений из любого поля таблицы Vvod2 (INSERT … SELECT …).

15. Создайте инструкцию для прочтения файла и загрузки данных в таблицу Vvod2.

16. Инструкции для создания структуры таблицы, чтения данных из текстового файла и вывода всех данных (командой SELECT) в виде одного файла, а также текстовые файлы с данными предъявите преподавателю для проверки.

Внимание!

По умолчанию программа MySQL ожидает, что поля файла разделяются символами табуляции, а строки заканчиваются символом новой строки (перевода строки). Опции FIELDS/LINES TERMINATED BY позволяют изменить эти установки.

Лабораторная работа 17

АНАЛИЗ ВЛИЯНИЯ КОНФИГУРАЦИОННЫХ ФАЙЛОВ НА ПАРАМЕТРЫ ЭКСПЛУАТАЦИИ. ПРОСМОТР ПЕРЕМЕННЫХ СРЕДЫ

1. Проверьте (если окажется возможным) наличие файла конфигурации:

- в системном каталоге (windows) – файл my.ini;
- в корневом каталоге диска C – файл my.cnf;
- в каталоге пользователя – файл my.cnf.

Внимание!
Каждый последующий файл перекрывает установки предыдущего и переопределяет значения переменных среды.

2. Откройте в текстовом редакторе пример файла конфигурации my-example.cnf (прил. 5) или файл, размещенный в папке sql. Изучите его структуру и правила записи опций конфигурационных файлов. Проанализируйте размещенный в папке sql файл client_r.txt, в котором приведены опции клиента mysql на русском или английском (файл client_e.txt) языках, изучите описанные в нем опции.

3. Создайте в своей рабочей папке файл my.cnf, который будет содержать настройки клиента MySQL. При запуске клиента MySQL используйте только опции командной строки вывода результатов запросов в файл, выполнение команд из файла и опцию --defaults-file=<путь к созданному вами файлу my.cnf>, обеспечивающую считывание значений параметров команды из конфигурационного файла. Это позволит задавать все остальные параметры и опции в файле my.cnf.

4. Создайте в файле my.cnf группу опций для клиента MySQL (раздел опций [client]), используя различные формы задания в командной строке и в конфигурационном файле (см. файлы client_r.txt, client_e.txt, my-example.cnf) следующих опций:

- host. Установите в файле конфигурации клиента IP-адрес компьютера, на котором установлен сервер MySQL (значение 194.226.238.70);
- user. Установите в файле конфигурации клиента имя пользователя, под которым происходит соединение;
- port. Установите в файле конфигурации клиента значение для порта 3306;

- table. Установите в файле конфигурации клиента вывод результатов запроса в виде таблицы. Выполните два любых запроса к базе данных EDUCATION, убедитесь, что результаты запросов выводятся в виде таблицы;
- vertical. Установите в файле конфигурации клиента вывод строк запроса в вертикальном формате. Выполните запрос к базе данных EDUCATION, убедитесь, что результаты запросов выводятся в вертикальном формате.

5. Выполните несколько раз запрос (например, вывести все записи из таблицы Lecturer базы данных EDUCATION), изменяя отдельные параметры, и просмотрите соответствующие результаты (SHOW VARIABLES). Информацию об изменении параметров и результаты просмотров (значения параметров) запишите в отчетный файл.

6. Просмотрите список баз данных (SHOW DATABASES).

7. Просмотрите список таблиц в базе данных (SHOW TABLES).

8. Просмотрите описания столбцов таблицы (SHOW COLUMNS). Проанализируйте выведенную информацию.

9. Просмотрите статистическую информацию о сервере (SHOW STATUS).

10. Инструкции и результаты их выполнения предъявите преподавателю для проверки.

11. Результаты просмотров добавьте в отчетный файл лабораторной работы.

12. Предъявите отчетный файл преподавателю.

Внимание!

При необходимости можно использовать дописывание в отчетный файл, применяя вместо опции > out опцию >>out, где out – имя отчетного файла.

Лабораторная работа 18

СРЕДСТВА ЗАЩИТЫ СЕТЕВОГО ДОСТУПА. ЗНАКОМСТВО С ТАБЛИЦАМИ РАЗРЕШЕНИЙ. ВИДЫ ПРИВИЛЕГИЙ. ДЕЛЕГИРОВАНИЕ ПОЛНОМОЧИЙ

Примечание. При выполнении данной работы по умолчанию предполагается, что обучаемым присвоены порядковые номера, определенные физическим размещением компьютеров в аудитории (или установленные преподавателем). Считается, что сосед слева имеет меньший предыдущий номер, а сосед справа – следующий больший номер. Первый и последний номера также являются соседними. Иной порядок может быть установлен преподавателем.

1. Изучите структуры таблиц User, Db, Host, Tables-priv, Columns_priv базы данных MYSQL. Структуры таблиц приведены в файле Priveleg в папке sql (прил. 6). Проанализируйте характер информации в таблицах.

2. Ознакомьтесь со списком привилегий (SHOW GRANTS) для пользователя.

3. Создайте в своей базе данных таблицу Dostup_r (*Владелец таблицы, Номер машины справа, Фамилия пользователя справа, Имя пользователя справа*). Заполните таблицу одной записью, указав в ней свою фамилию, а также данные соседа справа (по кругу).

4. Передайте соседу справа привилегии SELECT, INSERT.

5. Убедитесь в получении привилегий от левого соседа, просмотрев список привилегий (SHOW GRANTS).

6. Дополните таблицу Dostup_r соседа слева двумя записями, заполнив два правых поля его таблицы. Просмотрите эту таблицу и выведите результаты в отчетный файл.

7. Просмотрите свою таблицу Dostup_r после добавления в нее двух записей соседом справа. Отмените привилегии для соседа справа.

8. Удалите таблицу Dostup_r.

9. Создайте в своей базе данных таблицу Dostup_l (*Владелец таблицы, Номер машины слева, Фамилия пользователя слева, Имя*

пользователя слева). Заполните таблицу одной записью, указав в ней свою фамилию, а также данные соседа слева (по кругу).

10. Передайте соседу слева (или по указанию преподавателя) привилегии SELECT, INSERT.

11. Убедитесь в получении привилегий от соседа справа, просмотрев список привилегий (SHOW GRANTS).

12. Дополните таблицу Dostup_I соседа, находящегося справа, двумя записями, заполнив два правых поля его таблицы. Просмотрите эту таблицу и выведите результаты в отчетный файл.

13. Просмотрите свою таблицу Dostup_I после добавления в нее двух записей соседом слева. Отмените привилегии для соседа слева.

14. Удалите таблицу Dostup_I.

15. Предъявите преподавателю отчетный файл.

Библиографический список

Астахова И.Ф., Толстобров А.П., Мельников В.М. SQL в примерах и задачах: Учеб. Пособие. – Минск: Новое знание, 2002. – 176 с.

Аткинсон Л. MySQL. Библиотека профессионала: Пер. с англ. – М.: Вильямс, 2002. – 624 с.: ил.

Дейт К. Введение в системы баз данных: Пер. с англ. –Киев; -М.; СПб.: Вильямс, 2000. – 848 с.

Дюбуа П. MySQL. – М.: Вильямс, 2001. – 816 с.: ил.

Карпова Т.С. Базы данных: модели, разработка, реализация. – СПб.: Питер, 2001. – 304 с.: ил.

Методические указания по выполнению контрольных работ по дисциплине "Базы данных и управление ими"/Сост. В.В.Вьюхин; Рос. гос. проф.-пед. ун-т.– Екатеринбург, 2002. – 21 с.

Попов А.А. FoxPro 2.5/2.6. Создание приложений для FoxPro 2.5/2.6 в DOS и Windows. – М.: Март, 1996. – 660 с.

Опции клиента mysql

Формат использования клиента mysql:

mysql.exe [ОПЦИИ] [база данных]

-B, --batch Печать результатов со знаком табуляции в качестве разделителя
-C, --compress Использование сжатия в протоколе клиент/сервер
-#, --debug Вывод отладочной информации
-T, --debug-info Вывод некоторой отладочной информации при выходе
-e, --execute=... Выполнение команды и выход из клиента
-f, --force Продолжение выполнения, даже если произошла ошибка
-?, --help Вывод справки и выход из клиента
-h, --host=... Установка соединения с узлом, имеющим адрес ...
-n, --unbuffered Очистка буфера после каждого запроса
-O, --set-variable var=option
Установка значения переменной
-o, --one-database
Обновление только базы данных по умолчанию
-p[password], --password[=...]
Установка пароля, используемого при соединении с сервером. Если пароль не задан, то он запрашивается

-P --port=. .. Задание порта, используемого при установлении соединения
-q, --quick Отмена кэширования результатов запросов
-L, --skip-line-numbers
Отмена вывода номеров строк пр выводе ошибок
-t --table Вывод результатов в формате таблицы
-u, --user=# Использование имени при установлении соединения
-v, --verbose Вывод дополнительной информации
-V, --version Вывод информации о версии программы и выход
-E, --vertical Вывод строк результата запроса в верикальном формате
-w, --wait Установка режима ожидания и повторения попыток соединения в случае неудачи предыдущей попытки

Разрешенные переменные для установки:

max_allowed_packet – текущее значение: 25165824l
net_buffer_length – текущее значение: 16384l

Учебная база данных EDUCATION

Таблица Student ("Студент")

Student_id	Код студента
Family	Фамилия студента
Name	Имя студента
Last_name	Отчество студента
Course	Курс, на котором учится студент
Stipendia	Размер стипендии студента
Birthday	Дата рождения студента
Inst_id	Код института (факультета), где учится студент

Таблица Lecturer ("Преподаватель")

Lecturer_id	Код преподавателя
Family	Фамилия преподавателя
Name	Имя преподавателя
Last_name	Отчество преподавателя
Inst_id	Код института (факультета), где работает преподаватель

Таблица Subject ("Предмет обучения")

Subj_id	Код предмета
Subj_name	Наименование предмета
Hour	Количество часов на изучение предмета
Semestr	Семестр, в котором изучается данный предмет

Таблица Institute ("Институт/Факультет")

Inst_id	Код института (факультета)
Inst_name	Наименование института (факультета)
Rating	Рейтинг института (факультета)

Таблица Subject_Lect ("Учебные дисциплины преподавателей")

Lecturer_id	Код преподавателя
Subj_id	Код предмета

Таблица Exam ("Экзамен")

Exam_id	Код экзамена
Student_id	Код студента
Subj_id	Код предмета
Mark	Экзаменационная оценка
Exam_date	Дата экзамена

Инструкции языка SQL для MySQL

При описании инструкций языка SQL используется общепринятая нотация. Элементы в квадратных скобках считаются необязательными. Если имеются альтернативные варианты, они разделяются символами вертикальной черты.

Инструкции языка даются в алфавитном порядке.

ALTER TABLE

Инструкция ALTER TABLE позволяет менять определение таблицы. Для выполнения команды необходимо наличие привилегий ALTER, CREATE, INSERT. Общий формат команды:

ALTER [IGNORE] TABLE имя_таблицы
спецификация [,спецификация ...]

где спецификация:

ADD [COLUMN] определение [FIRST | AFTER столбец] – добавляет столбец к таблице в конец, начало или после указанного столбца;

ADD [COLUMN] (определение, определение,...) – добавляет группу новых столбцов. Определения столбцов разделяются запятыми;

ALTER [COLUMN] столбец { SET DEFAULT значение | DROP DEFAULT } – устанавливает новое значение по умолчанию или отменяет старое;

CHANGE [COLUMN] столбец определение – меняет определение столбца;

DROP {[COLUMN] столбец | PRIMARY KEY | INDEX индекс} – удаляет столбец, первичный ключ или индекс из таблицы.

Флаг IGNORE заставляет программу MySQL игнорировать изменения, вносимые в таблицу. Если IGNORE не определен, то копирование будет прервано и процесс отработан назад в случае наличия любых уникальных ключей, дублированных в новой таблице.

Внимание!

1. ADD и CHANGE используют те же определения, что и CREATE TABLE (см. инструкцию CREATE TABLE)
2. CHANGE столбец, DROP столбец и DROP INDEX являются MySQL–расширениями ANSI SQL
3. [COLUMN] – факультативный параметр и может быть опущен
4. DROP PRIMARY KEY удаляет первый уникальный ключ в таблице

CREATE DATABASE

Инструкция CREATE DATABASE создает базу данных (БД). Синтаксис инструкции:

CREATE DATABASE [IF NOT EXISTS] имя_БД

В MySQL каждая база данных хранится в отдельном подкаталоге, поэтому при выполнении инструкции создается пустой каталог. Просмотреть список существующих баз данных можно с помощью инструкции SHOW DATABASES.

CREATE TABLE

Инструкция CREATE TABLE предназначена для создания таблиц. Это одна из наиболее сложных SQL–инструкций. Общий ее формат таков:

CREATE [TEMPORARY] TABLE [IF NOT EXISTS] имя
[(спецификация, ...)]
[опция...]
[[IGNORE | REPLACE] запрос_на_выборку]

TEMPORARY задает создание временной таблицы, существующей в течение текущего сеанса. По завершении сеанса таблица удаляется.

IF NOT EXIST подавляет вывод сообщений об ошибках в случаях, если таблица с указанным именем уже существует. Имени таблицы может предшествовать имя базы данных, отделенное точкой (например, Сессия.ФИО). Если это не сделано, таблица будет создана в базе данных, которая установлена по умолчанию.

Спецификации столбцов и индексов приводятся в круглых скобках и разделяются запятыми. Спецификация имеет следующий формат:

Имя_столбца тип_столбца
[NOT NULL | NULL]
[DEFAULT значение]
[AUTO_INCREMENT]
[PRIMARY KEY]
[ссылка]

Спецификация типа включает название типа и его размерность.

У любого столбца есть значение по умолчанию, которое задается явно с помощью предложения DEFAULT или выбирается MySQL автоматически. Поля–счетчики, создаваемые с помощью AUTO_INCREMENT, игнорируют значения по умолчанию, т.к. в них записываются порядковые номера. Тип счетчика должен быть беззнаковым целым (например, INTEGER(10) UNSIGNED).

Спецификация PRIMARY KEY позволяет назначить столбец первичным ключом. При этом для столбца будет создан индекс.

В круглых скобках задаются спецификации не только столбцов, но также ограничений и индексов.

В конце инструкции CREATE TABLE может находиться запрос на выборку (в виде инструкции SELECT...). Результаты запроса на выборку будут занесены в создаваемую таблицу.

Если в самой инструкции CREATE TABLE отсутствуют спецификации столбцов, то вид создаваемой таблицы будет соответствовать таблице результатов запроса. В противном случае столбцы результатов запроса будут добавлены к определенным ранее столбцам.

Для создания таблицы вы должны иметь права доступа *create*.

Примечания:

1. Если столбец имеет дополнительное ключевое слово AUTO_INCREMENT (его можно использовать только в одном поле таблицы), то при вставке значение нуля в этот столбец вы получите номер столбца, который на 1 больше, чем самый большой предварительно использованный номер.
2. Если вы желаете начать отсчет не с нуля, просто вставьте желательное стартовое значение в первой записи, которую вы вставляете в данную таблицу.

CREATE INDEX

Инструкция CREATE INDEX добавляет индекс с указанным именем к заданной таблице. Формат инструкции:

CREATE [UNIQUE | FULLTEXT] INDEX *имя*
ON *таблица* (столбец [(длина)], ...)

UNIQUE обеспечивает создание индекса с уникальным значением по таблице *таблица*.

DELETE

Инструкция DELETE удаляет записи из таблицы. Ее синтаксис:

DELETE [LOW_PRIORITY] FROM таблица
[WHERE *условие*]
[LIMIT число_записей]

Здесь *условие* имеет формат:
условие1 | условие2 [AND|OR] условие3,
а любое из составляющих условий (1,2,3) имеет формат:
колонка [>|>=|=|<>|<=|<] колонка|константа or
колонка LIKE колонка|константа or
колонка IS NULL | колонка IS NOT NULL

В качестве условия в предложении WHERE допускается использование булевых выражений, составленных из произвольного числа выражений сравнения, объединяемых с помощью логических операторов.

LOW_PRIORITY откладывает операцию удаления до того момента, пока не завершатся все операции чтения таблицы.

LIMIT ограничивает число удаляемых записей.

Программа MySQL лишь помечает удаляемую строку как освобожденную. Пока она не будет затерта другой новой строкой, ее данные остаются на диске.

Инструкция OPTIMIZE TABLE удаляет неиспользуемые строки и восстанавливает правильный формат таблицы, т.е. уничтожает неиспользуемые записи, соединяет разделенные строки и выполняет сортировку индексов.

Инструкция DELETE:

- Возвращает количество обработанных записей.
- Если она (инструкция) используется без WHERE, то таблица будет очищена. В этом случае DELETE вернет 0 для числа обработанных записей.
- Сравнение с явным NULL (столбец == NULL) эквивалентно условию IS NULL, т. е. использованию (столбец IS NULL). Это было сделано для совместимости с mSQL.
- Необходимо иметь права доступа *delete* для удаления записей.

DESCRIBE

Инструкция DESCRIBE возвращает таблицу, содержащую описание одного или нескольких столбцов заданной таблицы. Формат инструкции:

{DESCRIBE|DESC} таблица [столбец|условие отбора]

Инструкция описывает таблицу или столбец. Эта команда подобна команде SHOW. Факультативный параметр может быть именем столбца или строкой с условием отбора.

Если параметр – строка, то он может содержать символы подстановки.

DROP DATABASE

Инструкция DROP DATABASE удаляет базу данных из системы. Формат инструкции:

DROP DATABASE [IF EXISTS] имя_таблицы

Вместе с базой данных удаляются все таблицы и вся хранимая информация (в том числе индексы). Если каталог базы данных пуст, он также удаляется. Если в каталоге присутствуют посторонние файлы, то сама база данных останется. Удаление каталога файловой системы тоже приводит к удалению базы данных.

Вы должны иметь права доступа *delete*, чтобы использовать DROP.

DROP INDEX

Инструкция DROP INDEX удаляет индекс таблицы с указанным именем. Формат инструкции:

DROP INDEX имя ON таблица

DROP TABLE

Инструкция DROP TABLE удаляет все файлы, относящиеся к таблице. Она имеет следующий синтаксис:

DROP TABLE [IF EXISTS] таблица [, таблица ...]

Если вы хотите только удалить все данные в таблице и сохранить ее структуру для будущего повторного заполнения, вы должны использовать команду DELETE.

Внимание!

DROP TABLE полностью удалит именованную таблицу(ы) из вашей системы. Не предусмотрено никакого UNDO или UNERASE (если вы не имеете резервной копии, конечно).

Можно указывать несколько имен таблиц, разделяя их запятыми.
Вы должны иметь права доступа *delete*, чтобы использовать DROP.

GRANT

Инструкция GRANT предоставляет указанным пользователям требуемые привилегии, создавая учетную запись пользователя в случае необходимости. Общий формат инструкции:

```
GRANT (ALL [PRIVILEGES])|(ALTER, CREATE, DELETE, DROP,
        FILE,INDEX, INSERT, PROCESS, RELOAD, SELECT,
        SHUTDOWN, UPDATE,USAGE (список_столбцов))
        ON таблица
        TO пользователь[@узел] [IDENTIFIED BY пароль]
        [, пользователь [IDENTIFIED BY пароль]...]
        [WITH GRANT OPTION]
```

Привилегии выдаются в зависимости от контекста:

- глобально,
- для конкретной базы,
- для таблицы или столбца.

В предложении ON возможно использование метасимвола *, соответствующего всем объектам диапазона.

Имя пользователя user[@узел] – это строка длиной не более 16 символов. Если вслед за именем стоят символ @ и сетевое имя, то привилегия применима только тогда, когда пользователь регистрируется на сервере с указанного компьютера. Если в именах пользователя и компьютера содержатся метасимволы, имена надо заключать в кавычки.

Вместо сетевого имени может стоять и IP–адрес. Если же узел не указан, то информация об узле вообще не будет учитываться.

Предложение IDENTIFIED BY задает пароль пользователя. Если учетная запись пользователя уже существует, его пароль меняется.

Спецификация WITH GRANT OPTION разрешает пользователю предоставлять имеющиеся привилегии другим пользователям.

Список привилегий для пользователя выводится инструкцией SHOW GRANTS.

INSERT

Инструкция INSERT добавляет записи в таблицу. Ее синтаксис:

```
INSERT [LOW_PRIORITY | DELAYED] [IGNORE]
    [INTO] таблица [(столбец, ...)]
    {VALUES (значение, ...),(...), ... | запрос | SET столбец=значение,
    ...}
```

LOW_PRIORITY означает, что операция вставки должна быть отложена до окончания всех операций чтения. Во время вставки на таблицу накладывается жесткая блокировка, разрешающая доступ к таблице для чтения и записи только потоку-владельцу.

После ключевого слова VALUES в скобках приводится набор значений, разделенных запятыми, соответствующих одной записи. Разрешается в одном предложении VALUES вводить значения сразу для нескольких записей (набор значений для каждой записи в отдельных скобках), что запрещено в стандарте.

Флаг IGNORE подавляет вывод сообщений об ошибках, выдаваемых в случае обнаружения дубликатов. При наличии флага дублирующая запись будет отброшена и операция продолжится.

KILL

Инструкция KILL разрывает соединение с базой данных. Ей передается идентификатор потока:

KILL поток

Все потоки пронумерованы. Инструкция SHOW PROCESSLIST отображает список активных потоков. Обычно пользователь имеет право уничтожать только свои собственные потоки, но с помощью привилегии PROCESS ему можно удалять любые потоки.

LOAD DATA INFILE

Инструкция LOAD DATA INFILE читает данные из текстового файла и вставляет их в указанную таблицу. Неполный формат инструкции:

```
LOAD DATA [LOW PRIORITY] [LOCAL] INFILE файл
        INTO TABLE таблица
        [FIELDS [TERMINATED BY разделитель]...
            [ENCLOSED by ' " ']
            [ESCAPED BY '\\']]
        [LINES TERMINATED BY признак_конца_строки]
        [IGNORE число_строк LINES]
        [(список_колонок)]
```

LOW PRIORITY означает, что процесс импорта данных должен быть отложен, пока не завершатся все операции чтения таблицы.

При отсутствии LOCAL файл открывается на сервере, в противном случае – в клиентской файловой системе. Для открытия файла на сервере следует иметь привилегию FILE.

Файл предполагает задание полного путевого имени файла. Неполные путевые имена вычисляются относительно каталога данных MySQL.

TERMINATED BY разделитель – указывает разделитель полей,

LINES TERMINATED BY признак_конца_строки – задает конечный символ строки.

ENCLOSED by – задает символ (обычно кавычку), который служит для дополнительного обособления значений.

ESCAPED BY '\\' – задает символ, который будет отменять специальное назначение управляющих символов. Ограждающие символы разрешается просто удваивать. Например, если все поля файла заключены в двойные кавычки, то для вставки такой кавычки в поле можно записать "", \".

С помощью предложения IGNORE...LINES можно пропустить требуемое число строк файла (например, заголовки в импортируемых файлах).

LOCK TABLES

Инструкция LOCK TABLES запрещает другим потокам осуществлять запись в ту или иную таблицу. Синтаксис инструкции:

```
LOCK TABLES таблица [AS псевдоним]
        {READ [LOCAL] | [LOW_PRIORITY] WRITE}
        [, таблица [AS псевдоним] {READ [LOCAL] |
        [LOW_PRIORITY] WRITE}
```

Инструкция ожидает бесконечно долго, пока требуемые блокировки не будут получены. В ней указывается список имен таблиц, разделенных запятыми. После каждого имени задается тип блокировки:

- READ (нежесткая) – запрещает запись в таблицу всем потокам, включая тот, кому она принадлежит.
- WRITE (жесткая) – разрешает доступ к таблице для чтения и записи только потоку–владельцу.
- LOCAL – разрешает выполнять инструкции INSERT, не приводящие к конфликтам. Но тогда файлы с табличными данными становятся доступными для записи и появляется угроза нарушения целостности данных.
- LOW_PRIORITY – жесткая блокировка будет представлена только тогда, когда нет отложенных нежестких блокировок. Обычно же приоритет жестких блокировок выше, чем у нежестких.

REPLACE

Инструкция REPLACE аналогична инструкции INSERT, за исключением того, что дублирующиеся записи заменяют собой существующие.

```
REPLACE [LOW_PRIORITY | DELAYED]...[INTO] таблица [(столбец,
...)]
        {VALUES (значение, ...),(...), ... | запрос | SET
        столбец=значение, ...}
```

REVOKE

Инструкция REVOKE отменяет привилегии, выданные ранее инструкцией GRANT. Ее формат такой же, как и у инструкции GRANT:

```
REVOKE тип [(столбец), ...)] [, тип [(столбец, ...)] ...]
        ON таблица
        FROM пользователь [, пользователь ...]
```

Типы привилегий те же, что и для инструкции GRANT. Инструкция REVOKE должна соответствовать определенному контексту и распространяет свое действие глобально, на базу данных, таблицу или столбцы.

Внимание!

Привилегии с более узким или более широким контекстом не затрагиваются.

SET

Инструкция SET меняет значения опций и переменных. Синтаксис инструкции:

SET [OPTION] параметр=значение [, параметр=значение]...

Устанавливаемые значения действительны лишь в течение текущего сеанса и по его завершении сбрасываются.

SHOW

Инструкция SHOW отображает различную информацию о базе данных MySQL. Форматы инструкции для различных вариантов использования:

SHOW [COLUMNS|FIELDS] FROM таблица [FROM база_данных] [LIKE шаблон] – возвращает описание столбцов таблицы. Описание каждого столбца представляет собой таблицу со столбцами, содержащими: имя столбца таблицы; тип столбца; данные о том, может или нет содержаться в столбце значение NULL; ответ на вопрос, является ли столбец индексируемым; значение по умолчанию; дополнительная информация о столбце; привилегии, выделенные для столбца.

SHOW CREATE TABLE таблица – возвращает описание запроса, который необходим для создания указанной таблицы.

SHOW DATABASES [LIKE шаблон] – возвращает список баз данных на сервере.

SHOW GRANTS FOR пользователь – выводит список привилегий для пользователя.

SHOW {INDEX | KEYS} FROM таблица [FROM база_данных] – возвращает информацию об индексах таблицы.

SHOW [MASTER | SLAVE] STATUS [LIKE шаблон] – возвращает статистическую информацию о сервере. Флаги MASTER или SLAVE

указывают на то, что информация выдается соответственно о главном или подчиненном сервере.

SHOW TABLES [FROM база_данных] [LIKE шаблон] – возвращает список таблиц в базе;

SHOW TABLE STATUS [FROM база_данных] [LIKE шаблон] – выводит статусную информацию о таблицах: имя таблицы, тип таблицы, формат хранения строки; число строк в таблице; среднее число байт на строку; размер файла в байтах; время создания таблицы; время последнего изменения таблицы и т.д.

UPDATE

Инструкция изменяет содержимое строк таблицы. Изменению подлежат строки, выбранные с помощью определенного в предложении WHERE выражения. Формат команды:

```
UPDATE [LOW_PRIORITY] таблица
        SET колонка=выражение [, колонка1=выражение1] ...
        [WHERE условие] [LIMIT n]
```

LOW_PRIORITY означает, что инструкция не выполнятся до тех пор, пока все клиенты не перестанут читать таблицу.

n в предложении LIMIT определяет максимальное число строк, которые будут обновлены.

USE

Инструкция USE задает базу данных, используемую по умолчанию. Формат команды:

USE база_данных

В последующих инструкциях все таблицы, имена которых приведены в коротком формате, будут считаться принадлежащими к этой базе данных.

Приложение 4

Формат и содержание некоторых функций SQL для MySQL

Примечание. Полный набор функций существенно шире. Для их изучения необходимо обратиться к руководству пользователя или к специальной литературе.

CASE() – вычисляет проверяемое выражение, сравнивает его со списком значений и в зависимости от результатов проверки возвращает один из результатов. Используется в двух вариантах:

Первый вариант:

```
CASE проверяемое значение
    WHEN значение1 THEN возвращаемое_значение1
    WHEN значение2 THEN возвращаемое_значение2
    ...
    [ELSE возвращаемое_значение для всех прочих проверяемых значений]
END
```

Второй вариант:

```
CASE
    WHEN условие1 THEN возвращаемое_значение1
    WHEN условие2 THEN возвращаемое_значение2
    ...
    [ELSE возвращаемое_значение для всех прочих условий]
END
```

CEILING(x) – возвращает наименьшее целое значение, не превышающее значения аргумента x

CONCAT(str1,str2,...) – возвращает строку, созданную путем объединения аргументов

CURDATE(), CURRENT_DATE – каждая из функций возвращает текущую дату. Вторая функция в скобках не нуждается

CURRENT_TIME – возвращает текущее время

DATE_ADD(date, INTERVAL expr interval) – возвращает новую дату после добавления интервала времени expr к значению даты date (возможные значения параметра interval – YEAR, MONTH, DAY, HOUR, MINUTE, SECOND, ...)

DAYNAME(date) – возвращает название дня недели по дате date

DAYOFMONTH(date) – возвращает числовое значение дня месяца по дате date

DAYOFWEEK(date) – возвращает номер дня недели по дате date из диапазона от 1 (воскресенье) до 7 (суббота)

DAYOFYEAR(date) – возвращает числовое значение дня года по дате date

EXP(x) – возвращает значение экспоненциальной функции (e^x)

FLOOR(x) – возвращает наибольшее целое значение, не превышающее значение аргумента x

INSTR(str, substr, pos) – возвращает номер позиции вхождения подстроки substr в строку str для pos экземпляра подстроки substr

LEFT(str, len)/RIGHT(str, len) – возвращает len левых (правых) крайних символов из строки str

LENGTH(str) – возвращает количество байтов (символов) в строке str

LOG(x) – возвращает значение натурального логарифма аргумента x

LPAD(str, len, pad_str), RPAD(str, len, pad_str) – дополняет слева (справа) строку str до длины len строкой pad_str

LTRIM(str) – возвращает подстроку после удаления из нее начальных пробелов

RTRIM(str) – возвращает подстроку после удаления из нее конечных пробелов

MOD(x,y) – возвращает остаток от деления нацело x на y. Аргументы x и y предварительно округляются до целых значений

MONTH(date) – возвращает номер месяца года по дате date

NOW() – возвращает текущие дату и время

REPLACE(str,str1,str2) – возвращает строку str после замены подстроки str1 подстрокой str2 в строке str

ROUND(x[,d]) – возвращает значение аргумента x, округленное до d десятичных разрядов. Округление выполняется до d знаков после десятичной точки, если d>0, и до d знаков до десятичной точки, если d<0

SIN(x) – возвращает тригонометрический синус аргумента x. Аргумент x измеряется в радианах

SPACE(n) – возвращает строку из n пробелов

SQRT(x) – возвращает положительный квадратный корень из неотрицательного числа x

TO_DAYS(дата) – вычисляет количество дней, прошедших с начала летоисчисления до указанной даты

YEAR(date) – возвращает номер года по дате

Пример конфигурационного файла

```
# Пример конфигурационного файла MySQL

# комментарии – закомментированная строка
# [<группа>] – имя программы или группы, для которой вы желаете установить
# опции
# После этой строки все установленные опции применяются именно для этой
# программы или группы

# <опция> – эквивалент --<опция> для командной строки

# <опция>=<опция>– эквивалент  <опция>=<значение> для командной строки

# set-variable=<переменная>=<значение переменной> – установка значения
# указанной переменной. Эквивалент set variable <переменная>=<значение>
# для командной строки

#####################################################################
#####################################################################

# Следующие опции влияют на клиента MySQL
[client]

# Пароль, используемый при подключении к серверу MySQL
password=d3453467

# Порт, через который происходит соединение
port=3306

# Подключение к серверу MySQL под указанным именем
user=administrator

# IP–адрес или имя компьютера, на котором расположен сервер MySQL
# и на который будут посылаться запросы
host=192.268.64.42

# Использовать сжатие в протоколе клиент/сервер
compress

# Выполнить команды из файла и завершить работу клиента
execute=c:/command.txt
```

```
#####################################################################
#####################################################################

# Следующие опции влияют на сервер MySQL
[mysqld]

# Порт, через который происходит соединение
port=3306

# Путь к каталогу MySQL
basedir=c:/mysql

# Путь к каталогу, содержащему базы данных MySQL
datadir=c:/mysql/data

# Максимальное число одновременных соединений с сервером MySQL
set-variable=max_connections=100

# Запрещает использование команды SHOW DATABASES
skip-show-database

# Запускает сервер и выполняет SQL команды из заданного файла
init-file=c:/command.txt

# Запись информации о соединениях и запросах в указанный файл
log=c:/log.txt

# Запуск сервера без пользовательских привилегий. Такой режим дает
# ПОЛНЫЙ ДОСТУП всем пользователям
skip-grant-tables

# Загрузить демон MySQL как сервис NT
install

# Загрузить демон MySQL как программу NT
standalone
```

Таблицы привилегий MySQL

```
CREATE TABLE columns_priv
 (
 Host char(60)              binary       NOT NULL default '',
 Db char(64)                binary       NOT NULL default '',
 User char(16)              binary       NOT NULL default '',
 Table_name char(64)        binary       NOT NULL default '',
 Column_name char(64) binary             NOT NULL default '',
 Timestamp          timestamp(14)        NOT NULL,
 Column_priv set('Select', 'Insert', 'Update', 'References') NOT NULL default '',
 PRIMARY KEY  (Host, Db, User, Table_name, Column_name)
 )
TYPE=MyISAM
COMMENT='Column privileges';


CREATE TABLE db
 (
 Host char(60) binary                       NOT NULL default '',
 Db   char(64) binary                       NOT NULL default '',
 User char(16) binary                       NOT NULL default '',
 Select_priv enum('N','Y')                  NOT NULL default 'N',
 Insert_priv enum('N','Y')                  NOT NULL default 'N',
 Update_priv enum('N','Y')                  NOT NULL default 'N',
 Delete_priv enum('N','Y')                  NOT NULL default 'N',
 Create_priv enum('N','Y')                  NOT NULL default 'N',
 Drop_priv   enum('N','Y')                  NOT NULL default 'N',
 Grant_priv  enum('N','Y')                  NOT NULL default 'N',
 References_priv enum('N','Y')              NOT NULL default 'N',
 Index_priv  enum('N','Y')                  NOT NULL default 'N',
 Alter_priv  enum('N','Y')                  NOT NULL default 'N',
 Create_tmp_table_priv enum('N','Y')        NOT NULL default 'N',
 Lock_tables_priv enum('N','Y')             NOT NULL default 'N',
 PRIMARY KEY  (Host, Db, User),
 KEY User (User)
 )
TYPE=MyISAM
COMMENT='Database privileges';
```

```
CREATE TABLE host
 (
 Host       char(60) binary                 NOT NULL default '',
 Db         char(64) binary                 NOT NULL default '',
 Select_priv enum('N','Y')                  NOT NULL default 'N',
 Insert_priv enum('N','Y')                  NOT NULL default 'N',
 Update_priv enum('N','Y')                  NOT NULL default 'N',
 Delete_priv enum('N','Y')                  NOT NULL default 'N',
 Create_priv enum('N','Y')                  NOT NULL default 'N',
 Drop_priv  enum('N','Y')                   NOT NULL default 'N',
 Grant_priv  enum('N','Y')                  NOT NULL default 'N',
 References_priv enum('N','Y')              NOT NULL default 'N',
 Index_priv  enum('N','Y')                  NOT NULL default 'N',
 Alter_priv  enum('N','Y')                  NOT NULL default 'N',
 Create_tmp_table_priv enum('N','Y')        NOT NULL default 'N',
 Lock_tables_priv enum('N','Y')             NOT NULL default 'N',
 PRIMARY KEY  (Host, Db)
 )
TYPE=MyISAM
COMMENT='Host privileges; Merged with database privileges';


CREATE  TABLE tables_priv
 (
 Host          char(60) binary       NOT NULL default '',
 Db            char(64) binary       NOT NULL default '',
 User          char(16) binary       NOT NULL default '',
 Table_name char(60) binary          NOT NULL default '',
 Grantor       char(77)              NOT NULL default '',
 Timestamp timestamp(14)             NOT NULL,
 Table_priv set('Select', 'Insert', 'Update', 'Delete', 'Create', 'Drop', 'Grant',
     'References', 'Index', 'Alter')   NOT NULL default '',
 Column_priv set('Select', 'Insert', 'Update', 'References') NOT NULL default '',
 PRIMARY KEY  (Host, Db, User, Table_name),
 KEY Grantor (Grantor)
 )
TYPE=MyISAM
COMMENT='Table privileges';
```

```
CREATE TABLE user
 (
 Host varchar(60)       binary   NOT NULL default '',
 User varchar(16)       binary   NOT NULL default '',
 Password varchar(45) binary   NOT NULL default '',
 Select_priv       enum('N','Y')  NOT NULL default 'N',
 Insert_priv       enum('N','Y')  NOT NULL default 'N',
 Update_priv       enum('N','Y')  NOT NULL default 'N',
 Delete_priv       enum('N','Y')  NOT NULL default 'N',
 Create_priv       enum('N','Y')  NOT NULL default 'N',
 Drop_priv         enum('N','Y')  NOT NULL default 'N',
 Reload_priv       enum('N','Y')  NOT NULL default 'N',
 Shutdown_priv  enum('N','Y')  NOT NULL default 'N',
 Process_priv      enum('N','Y')  NOT NULL default 'N',
 File_priv         enum('N','Y')  NOT NULL default 'N',
 Grant_priv        enum('N','Y')  NOT NULL default 'N',
 References_priv enum('N','Y')  NOT NULL default 'N',
 Index_priv        enum('N','Y')  NOT NULL default 'N',
 Alter_priv        enum('N','Y')  NOT NULL default 'N',
 PRIMARY KEY  (Host, User)
 )
TYPE=MyISAM
COMMENT='Users and global privileges';
```

Содержание

Вьюхин Виктор Викторович,
Супрун Светлана Владимировна,
Кочнева Татьяна Анатольевна

БАЗЫ ДАННЫХ

Лабораторный практикум

Часть 1

Редактор Л.И. Кузнецова

Компьютерная верстка В.В. Вьюхина

Печатается по постановлению
редакционно-издательского совета университета

Подписано в печать 18.06.05. Формат 60x84/16. Бумага для множ. аппаратов. Печать плоская. Усл. печ. л. 5,0. Уч.-изд. л. 5,2. Тираж 50 экз. Заказ № 80
Издательство Российского государственного профессионально-педагогического университета. Екатеринбург, ул. Машиностроителей, 11.

Ризограф РГППУ. Екатеринбург, ул. Машиностроителей, 11.