

УО «Полоцкий государственный университет»

МЕТОДИЧЕСКИЕ УКАЗАНИЯ №7

**к выполнению лабораторной работы
по курсу «Базы данных»
для специальности Программное обеспечение информационных
технологий 1-40 01 01**

ТЕМА: Основы работы с базами данных в MySQL.

Полоцк 2016

Методические указания разработали:

Преподаватель-стажер технологий программирования Войтехович Агния Витольдовна

Старший преподаватель кафедры технологий программирования Бураченко Ирина
Брониславовна

ТЕМА: Основы работы с базами данных в MySQL.

ЦЕЛЬ: Научиться работать с базами данных, используя утилиту dbForge Studio for MySQL.

Результат обучения:

После успешного завершения занятия пользователь должен:

- Уметь создавать и редактировать базы данных в MySQL.
- Уметь импортировать в MySQL базы данных Microsoft Access и их объекты.
- Уметь создавать, редактировать и изменять связи между таблицами базы данных MySQL.

Используемая программа: dbForge Studio for MySQL

План занятия:

1. Общие сведения о MySQL.
2. Клиент/серверные системы.
3. Базы данных в MySQL.
4. Работа с утилитой dbForge Studio for MySQL.
5. Основные этапы работы с MySQL.

1. ОБЩИЕ СВЕДЕНИЯ ОБ SQL SERVER

MySQL — свободная реляционная система управления базами данных. Разработку и поддержку MySQL осуществляет корпорация Oracle.

MySQL является решением для малых и средних приложений. Входит в состав серверов WAMP, AppServ, LAMP и в портативные сборки серверов Денвер, ХАМРР, VertrigoServ. Обычно MySQL используется в качестве сервера, к которому обращаются локальные или удалённые клиенты, однако в дистрибутив входит библиотека внутреннего сервера, позволяющая включать MySQL в автономные программы.

Гибкость СУБД MySQL обеспечивается поддержкой большого количества типов таблиц: пользователи могут выбрать как таблицы типа MyISAM, поддерживающие полнотекстовый поиск, так и таблицы InnoDB, поддерживающие транзакции на уровне отдельных записей. Более того, СУБД MySQL поставляется со специальным типом таблиц EXAMPLE, демонстрирующим принципы создания новых типов таблиц. Благодаря открытой архитектуре и GPL-лицензированию, в СУБД MySQL постоянно появляются новые типы таблиц.

MySQL портирована на большое количество платформ: AIX, BSDi, FreeBSD, HP-UX, Linux, Mac OS X, NetBSD, OpenBSD, OS/2 Warp, SGI IRIX, Solaris, SunOS, SCO OpenServer, UnixWare, Tru64, Windows 95, Windows 98, Windows NT, Windows 2000, Windows XP, Windows Server 2003, WinCE, Windows Vista, Windows 7 и Windows 10. Существует также порт MySQL к OpenVMS. Важно отметить, что на официальном сайте СУБД для свободной загрузки предоставляются не только исходные коды, но и откомпилированные и оптимизированные под конкретные операционные системы готовые исполняемые модули СУБД MySQL.

MySQL имеет API[источник не указан 1358 дней] для языков Delphi, C, C++, Эйфель, Java, Лисп, Perl, PHP, Python, Ruby, Smalltalk, Компонентный Паскаль и Tcl, библиотеки для языков платформы .NET, а также обеспечивает поддержку для ODBC посредством ODBC-драйвера MySQL.

MySQL представляет собой драйвер ODBC (2.50) уровня 0 (с некоторыми возможностями уровней 1 и 2) для подключения совместимого с ODBC приложения к MySQL. MySQL работает на всех системах Microsoft Windows и на большинстве платформ Unix.

Максимальный размер таблиц в MySQL 3.22 до 4 гигабайт, в последующих версиях максимальный размер ограничивается максимальным размером файла используемой операционной системы.

Размер таблицы ограничен её типом. В общем случае тип MySQL ограничен предельным размером файла в файловой системе операционной системы. Например, в NTFS этот размер теоретически может быть до 32 эксабайт. В случае InnoDB одна таблица может храниться в нескольких файлах, представляющих единое табличное пространство. Размер последнего может достигать 64 терабайт.

В отличие от MySQL, в InnoDB имеется значительное ограничение на количество столбцов, которое можно добавить в одну таблицу. Размер страницы памяти по умолчанию составляет 16 килобайт, из которых под данные отведено 8123 байта. Размер указателя на динамические поля составляет 20 байт. Таким образом, в случае использования динамического формата строки (ROW_FORMAT=DYNAMIC), одна таблица может вместить максимум 409 столбцов типа blob или text.

dbForge Studio является универсальным инструментом для работы с MySQL сервером, который позволяет разработчикам MySQL и администраторам баз данных MySQL создавать и выполнять запросы, разрабатывать и отлаживать процедуры и функции, а также автоматизировать управление объектами баз данных MySQL с помощью удобного пользовательского интерфейса.

Это приложение для работы с MySQL дополнительно содержит инструменты для сравнения, синхронизации, создания резервных копий баз данных по графику, а также для анализа и создания отчетов по данным таблиц MySQL.

2. КЛИЕНТ/СЕРВЕРНЫЕ СИСТЕМЫ

Программное обеспечение сервера базы данных обрабатывает запросы, инициируемые программным обеспечением клиента, посылая результат обработки запросов клиенту. Программное обеспечение клиента «подсоединяется» к программному обеспечению сервера, делает запрос, получает результат и обрабатывает его, возможно, выводя на экран. dbForge Studio также является клиентским приложением относительно сервера MySQL, несмотря даже на то, что он может быть установлен на том же компьютере.

На рисунке 1 показано взаимодействие программного обеспечения сервера и клиентов. Компьютер клиента, работающий под управлением какой-либо операционной системы (Windows XP/Vista/7/8, Unix-системы), запускает клиентское приложение MySQL. Это приложение посылает запросы на сервер MySQL, принадлежащий компьютеру сервера, который работает под управлением ОС Windows XP/Vista/7/8 или Unix-системы и запускает все типы приложений, включая приложение клиента и сервера.

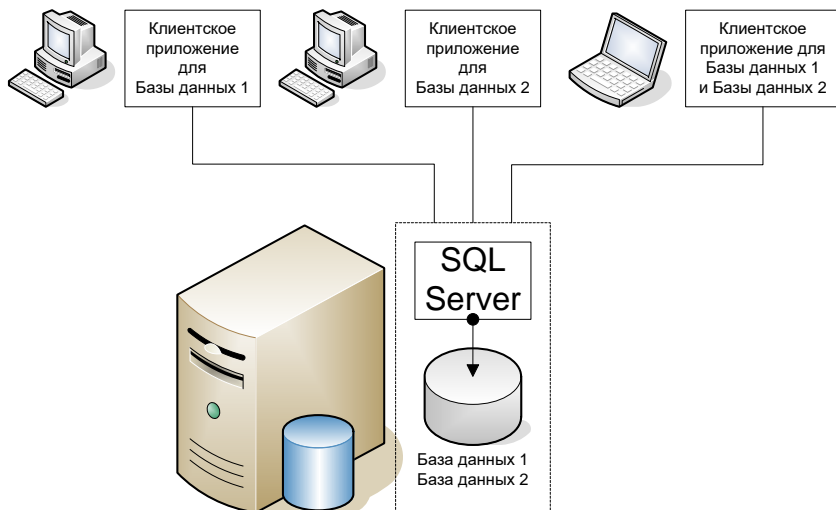


Рис. 1. *Взаимодействие программного обеспечения сервера и клиентов*

Пользователей базы данных, которая обслуживается сервером, может быть много. Данные в базе в целях сохранности и безопасности могут быть закрыты частично или полностью (по крайней мере, для обновления) от различных категорий пользователей. Для этого в MySQL имеются соответствующие механизмы, в том числе так называемые *представления* и *хранимые процедуры*, которые, как и таблицы базы данных, могут быть доступны или недоступны определенным группам пользователей. Представления — это своеобразные фильтры, отображающие данные базы. Клиенту можно вообще закрыть всякий доступ к самим таблицам базы и разрешить доступ к ним только посредством представлений. С помощью таких фильтров можно скрыть некоторые данные от клиента. Хранимые процедуры являются программами, которые работают на сервере и вызываются либо приложением клиента, либо правилами, поддерживающими целостность данных (триггерами). Хранимые процедуры применяют в работе мощные аппаратные решения MySQL и способствуют высокой производительности при обработке данных. Как представления, так и хранимые процедуры пишутся на одном из диалектов языка SQL.

Менеджер, работающий с клиентским приложением, должен знать параметры доступа к данным (логическое имя и пароль, зарегистрированные на сервере администратором), которые доступны ему, согласно его функциональным обязанностям (о которых, конечно, должен быть осведомлен администратор, определяющий права доступа к объектам баз данных). Если менеджеру необходимы дополнительные способы обработки данных, он может обратиться либо к разработчику клиентской, либо к разработчику серверной части СУБД.

3. БАЗЫ ДАННЫХ В MYSQL

Данные в MySQL хранятся в базах данных. Физическая реализация базы представляет собой несколько файлов на диске. В SQL Server, кроме пользовательских, имеются несколько встроенных (системных) баз, одними из которых являются **information_schema**, **mysql** и автоматически созданная при первом подключении

клиента к серверу MySQL база данных **test**. На рис. 2 (с помощью утилиты dbForge Studio) показан состав баз данных MySQL до того, как пользователь создал свою первую базу.

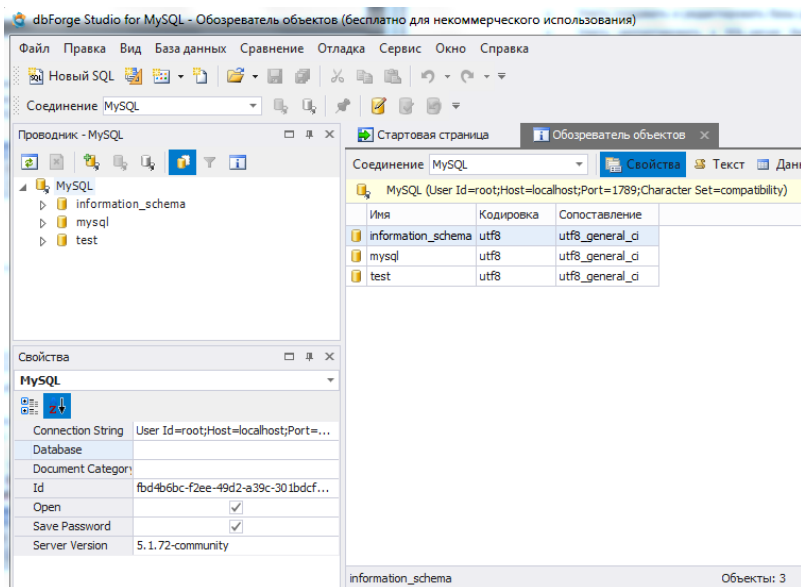


Рис. 2. Системные объекты базы данных MySQL

Как системные, так и пользовательские базы данных в MySQL располагаются в различных объектах. К основным из них относятся (рис. 3): **таблицы (Tables)**, **представления (Views)**, **процедуры (Procedures)**, **функции (Functions)**, **триггеры (Triggers)**, **события (Events)**, **UDF'ы (UDFs)**.

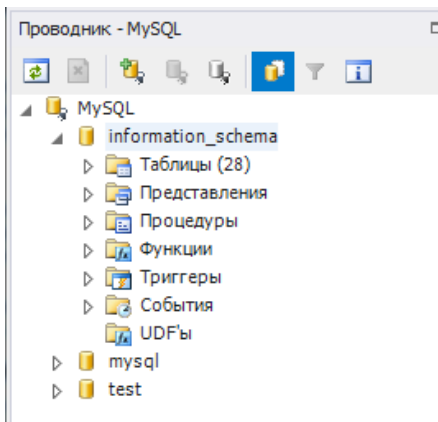


Рис. 3. Основные объекты базы данных в MySQL

Таблицы (Tables) базы данных являются местом хранения бизнес-данных, организованных на этапе проектирования базы в логически связанные между собой двумерные таблицы.

Представления (Views) — ОБЪЕКТ ДАННЫХ КОТОРЫЙ не содержит никаких данных его владельца. Это - тип таблицы, чье содержание выбирается из других таблиц с помощью выполнения запроса. Поскольку значения в этих таблицах меняются, то автоматически, их значения могут быть показаны представлением.

Хранимые процедуры (Stored Procedures) представляют собой набор команд SQL, которые могут компилироваться и храниться на сервере. Таким образом, вместо того, чтобы хранить часто используемый запрос, клиенты могут ссылаться на соответствующую хранимую процедуру. Это обеспечивает лучшую производительность, поскольку данный запрос должен анализироваться только однажды и уменьшается трафик между сервером и клиентом. Концептуальный уровень можно также повысить за счет создания на сервере библиотеки функций. Процедура вызывается, используя инструкцию CALL, и может только передавать обратные значения, используя переменные вывода.

Функции (Functions) — это механизм, аналогичный хранимым процедурам. Часто эти две категории объединяют одним понятием – хранимые подпрограммы. Так же, как и процедуры, пользовательские функции хранятся в той базе данных, в которой были созданы. Разница состоит лишь в том, что она возвращает скалярное значение. Функция может быть названа точно так же, как и любая другая функция языка (то есть, вызывая имя функции).

Триггеры (Triggers) — это хранимая процедура, которая не вызывается непосредственно, а исполняется при наступлении определенного события (вставка, удаление, обновление строки).

События (Events) — это извещения системы о том, что пользователь выполнил какое-либо действие, либо внутри самой системы возникло некоторое условие.

Определяемые пользователем функции (UDF's) - это средство, позволяющее расширить MySQL за счет новых функций, которые работают подобно "родным" (встроенным) функциям MySQL, таким как ABS() и CONCAT().

Пользователи SQL Server могут на основе существующих определять собственные типы данных, называемые *типы данных, определенные пользователем*.

4. РАБОТА С УТИЛИТОЙ DBFORGE STUDIO FOR MYSQL

Одной из наиболее многофункциональных и удобных утилит для работы с базами данных является dbForge Studio for MySQL. Рассмотрим основные положения, которые могут пригодиться при работе с утилитой dbForge Studio. Она устанавливается независимо от сервера MySQL. После этого ее можно запустить либо нажав на ярлык на рабочем столе, либо из меню **Пуск/Все программы** (рис. 4).

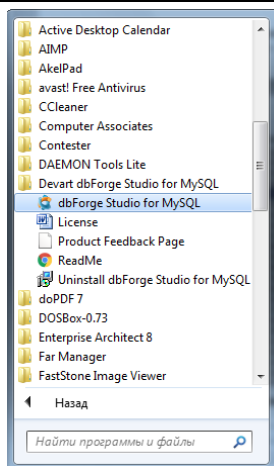


Рис. 4. Утилита **dbForge Studio for MySQL** доступна из меню **Пуск**

На рис. 5 отмечены основные элементы диалогового окна утилиты SQL Server Enterprise Manager.

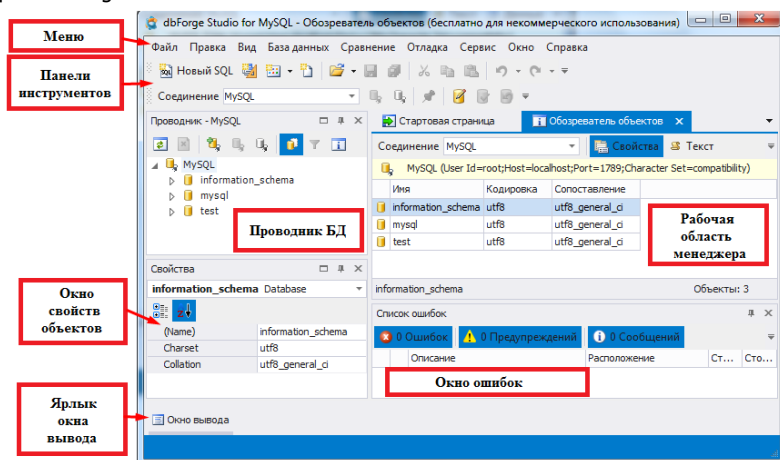


Рис. 5. Основные элементы диалогового окна утилиты **dbForge Studio for MySQL**

Рис. 6.

Перечень возможностей dbForge Studio for MySQL:

1. Интеллектуальная разработка SQL кода

Разумное дополнение кода, красивое форматирование и шаблоны кода делают написание SQL кода более эффективным и удобным. Навигация по коду, получение основной информации об объектах и проверка синтаксиса производится моментально.

2. Сравнение и синхронизация БД

При внесении изменений в структуру баз данных MySQL, перемещении данных между серверами, анализе различий между базами Вы также можете:

- Сравнить и синхронизировать данные и схемы
- Планировать стандартные задачи по синхронизации БД
- Генерировать отчеты о сравнении

3. Визуальный дизайнер запросов

Визуальное создание запросов на диаграмме, используя удобный редактор выражений. Вы можете создавать запросы любой сложности за считанные минуты. Приложение автоматически соединяет таблицы и позволяет работать с выражениями INSERT, UPDATE, и DELETE.

4. Дизайнер баз данных

Используйте диаграмму базы данных для создания или реверсивного проектирования, настройки, анализа и печати баз данных MySQL, а также для:

- Просмотра связей по внешним ключам
- Отображения объектов БД со свойствами
- Выполнение хранимых процедур

5. Импорт/экспорт данных

Добавляйте данные из внешних источников в базы MySQL с помощью инструментов для экспорта и импорта. Поддержка 10 популярных форматов и использование шаблонов позволяет автоматизировать весь процесс экспорта и импорта данных с помощью командной строки.

6. Резервные копии БД

Создавайте резервные копии баз данных MySQL, используя 5 уровней сжатия файлов и большой набор опций для настройки резервного копирования. Сохраняйте Ваши настройки в виде проекта для дальнейшего использования или для планировщика задач, вызывая проект из командной строки.

7. Администрирование БД

Инструменты для администрирования и управления базами данных MySQL включают средства для:

- Управления ролями и привилегиями пользователей
- Контроля сервисов MySQL
- Управления переменными сервера
- Обслуживания таблиц
- Управления сессиями

8. Отладчик MySQL

Первый отладчик для MySQL, который предоставляет пошаговое выполнение кода, точки останова, переменные, стек вызовы. Отладчик сохраняет логику выполнения процедур, а также позволяет производить отладку функций и триггеров MySQL.

9. Дизайнер таблиц

Визуальный дизайнер для изменения или создания структуры таблиц и индексов. Предоставляет полный контроль над вносимыми изменениями, а эвристика предложит тип столбца при его создании, основываясь на вводимом имени.

10. Рефакторинг базы данных

Совершенствуйте дизайн баз данных с помощью небольших изменений. Такие изменения будут абсолютно прозрачными и приложение позаботится о всех зависимостях в базе данных:

- Переименование объектов баз данных из Проводника БД
- Переименование столбцов таблицы из дизайнера
- Просмотр скрипта рефакторинга

11. Профилировщик запросов

Совершенствуйте запросы, выполнение которых занимает много времени, с помощью профилировщика запросов. Используя этот инструмент, Вы найдете проблемные места в запросе и сможете оптимизировать его, используя интерфейс приложения. Профилировщик:

- Предлагает визуальное профилирование запросов
- Сравнивает результаты профилирования

12. Отчеты и анализ данных

Отчеты по данным можно создавать с помощью мастера с большим набором возможностей или вручную, на пустом бланке отчетов. Готовые отчеты можно экспортировать в 8+ форматов и доставлять получателям, используя командную строку.

Используйте сводные таблицы для облегчения чтения данных и их понимания.

Более подробную информацию об утилите dbForge Studio for MySQL вы можете прочитать на сайте разработчика: <https://www.devart.com/ru/dbforge/mysql/studio/>

5. ОСНОВНЫЕ ЭТАПЫ РАБОТЫ С MYSQL

Работа с MySQL состоит из непосредственной работы на сервере и написании программного обеспечения так называемой *клиентской части*, которая будет создавать запросы (обычно посредством пользовательского интерфейса) для сервера и принимать и представлять полученную информацию.

Основные этапы работы непосредственно с сервером можно выполнить при помощи утилиты dbForge Studio или непосредственно языка MySQL. Они состоят, как минимум, из:

- создания базы (или нескольких) данных — набора таблиц;
- написания представлений и хранимых процедур для обработки данных на сервере;
- установления прав доступа для каждого элемента базы данных.

Основная работа, связанная с построением клиентской части, состоит из построения интерфейса и написания программ для взаимодействия с серверной частью.

5.1. Создание базы данных при помощи dbForge Studio for MySQL

Конечно, в больших коллективных системах управления базами данных созданием новых баз занимается только администратор, но начинающий разработчик баз данных на своем персональном компьютере может делать все, что душе угодно, т.е. во много раз больше, чем может позволить себе многоопытный и ответственный администратор баз данных.

Создание базы — обычный процесс (конечно, без учета этапа проектирования) описания таблиц и их взаимных связей — в данном случае выполняется, как и во многих других СУБД. При этом можно воспользоваться теми таблицами, которые вы, быть может, ранее применяли в других системах, например, Microsoft Visual FoxPro, Paradox

(начиная с версии III), dBase (версии III-IV), Microsoft Access и Excel. Для этих целей подойдут и просто текстовые файлы. С вновь созданной базой при помощи утилиты dbForge Studio for MySQL можно работать таким же образом, как во многих других оболочках систем управления базами данных. Эта утилита может быть использована почти для всех операций администрирования локальных или удаленных баз данных.

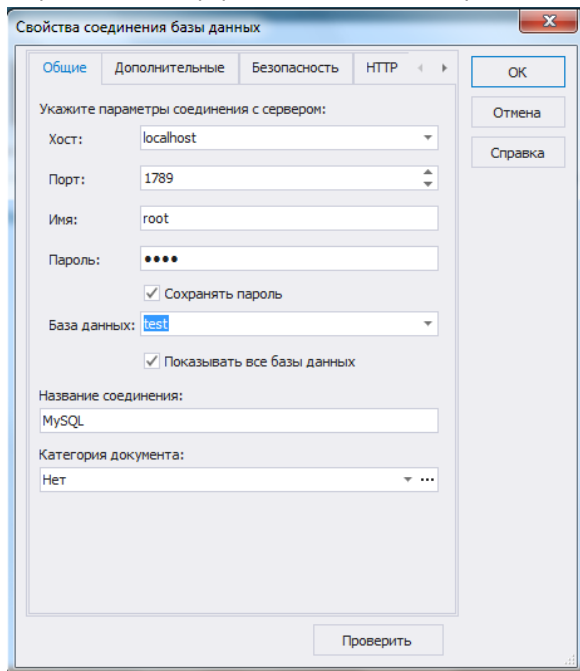


Рис. 7. Для работы с dbForge Studio for MySQL сначала следует создать подключение утилиты к серверу MySQL

Для работы с dbForge Studio for MySQL сначала следует создать подключение утилиты к серверу MySQL (если эта операция не выполняется при загрузке операционной системы) и щелкнуть на кнопке **Обновить** (рис. 6). Теперь можно работать с сервером баз данных.

Далее щелкните правой кнопкой мыши на компоненте  с названием подключения в окне **проводника MySQL** и из контекстного меню выберите **Новая база данных...** (рис. 8).

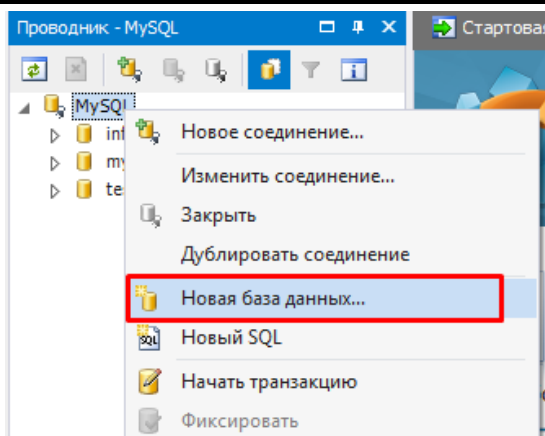


Рис. 8. Для создания новой базы данных щелкните правой кнопкой мыши на компоненте **Подключение** в окне проводника **MySQL** и из контекстного меню выберите **Новая база данных...**

В текстовом окне **Название** диалогового окна создания базы данных введите новое имя базы данных, например, **Фирма**, и выберите кодировку символов для новой базы данных, как на рис. 9.



Рис. 9. Диалоговое окно для создания новой базы данных

Щелкните на кнопке **Обновить базу**. Новая база данных появится в окне **проводника MySQL** (рис. 10). Как вы могли уже догадаться (по наименованию базы данных), далее рассматривается процесс создания таблиц очень простой базы данных, которая подробно описана в главах 1 и 2.

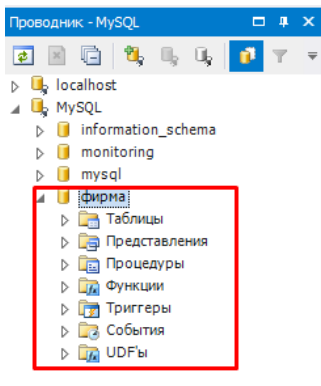


Рис. 10. Новая база данных в окне *проводника MySQL*

Создать в базе данных таблицу не сложнее, чем саму базу данных. Щелкните правой кнопкой мыши на имени базы в окне **Проводника** и в контекстном меню выберите сначала **Новый объект**, а затем — **Таблица** (рис. 11), либо разверните дерево папок указанной базы данных, щелкните правой клавишей мыши на папке **Таблицы** и в контекстном меню выберите команду **Новая таблица**.

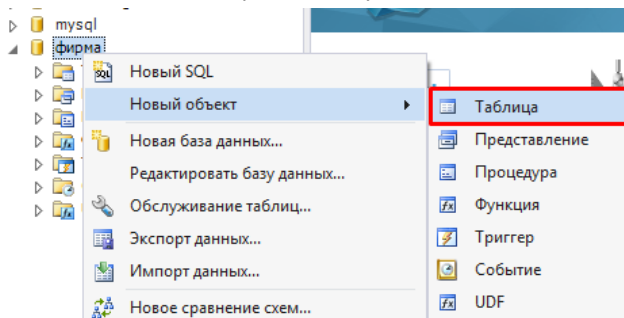


Рис. 11. Создать в базе данных таблицу не сложнее, чем саму базу данных

В окне **dbForge Studio -[фирма.table1]**, которое очень похоже на окно приложения Microsoft Access, введите описание таблицы **Подразделения** (рис. 12). Как и в Microsoft Access, для указания первичного ключа достаточно поставить галочку в нужной строке. Кроме того, параллельно с созданием таблицы в конструкторе утилита автоматически генерирует SQL-скрипт для создания этой таблицы.

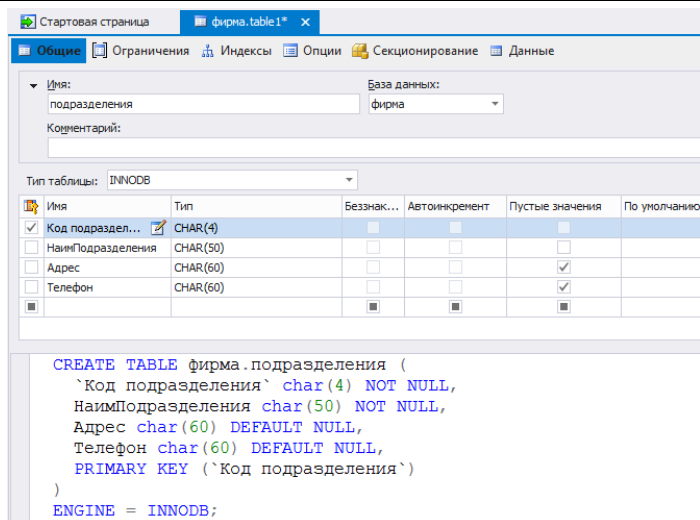


Рис. 12. Окно *dbForge Studio* -[фирма.table1], очень похожее на окно приложения *Microsoft Access*, позволяющее ввести описание таблицы в базе данных

Для выхода из режима создания таблицы сначала нажмите на кнопку **Обновить базу**, а затем нажмите на крестик в верхнем правом углу рабочей области менеджера. В дальнейшем вы всегда можете редактировать таблицу, выбрав ее из иерархического меню, дважды щелкнув левой кнопкой мыши на значке нужной таблицы.

Подобным образом можно определить и две другие таблицы (рассмотренные ранее) — **Товары** и **Запасы**. На рис. 13 и 14 эти таблицы приведены в режиме редактирования.

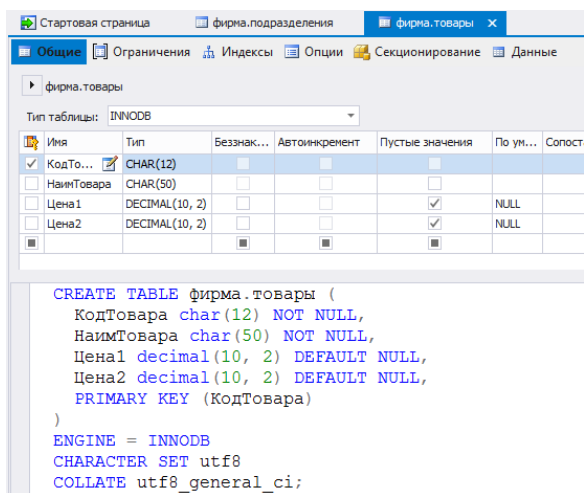


Рис. 13. Таблица **Товары** в режиме редактирования

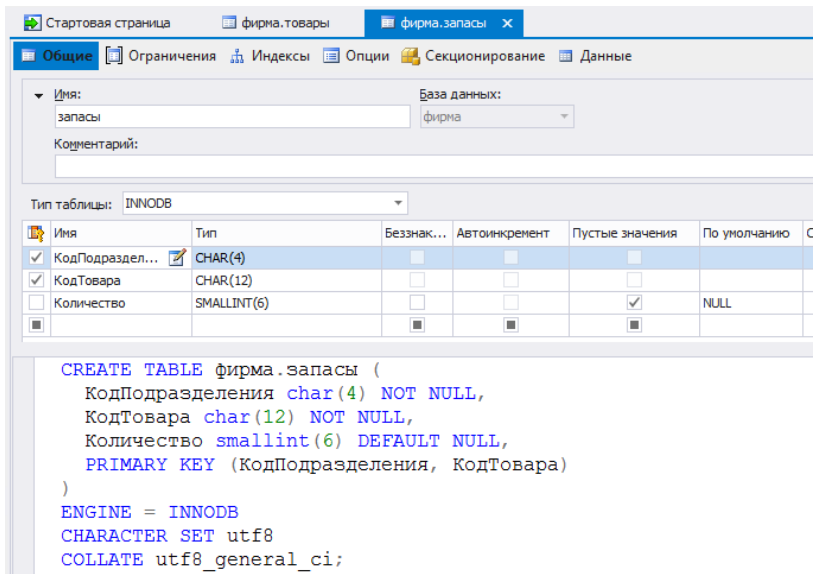


Рис. 14. Таблица **Запасы** в режиме редактирования

5.2. Связи между таблицами

Кроме создания таблиц, в окне **dbForge Studio** можно указать связи между таблицами. Для этого нужно войти в режим редактирования одной из таблиц базы данных, в строке меню щелкнуть кнопку **Таблица** выбрать пункт **Новый внешний ключ...**

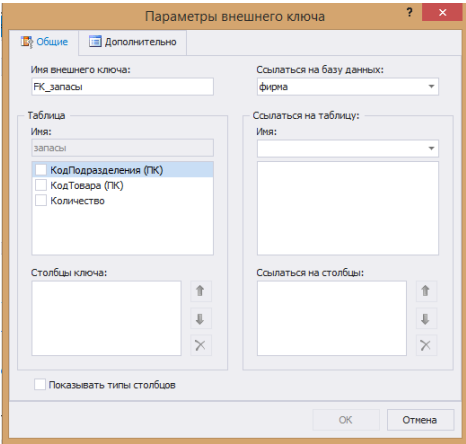


Рис. 15. Окно **Параметры внешнего ключа** для таблицы **запасы**

В окне комбинированного списка **Таблица** укажите поле с вторичным ключом, а в окне **Ссылаться на таблицу** — таблицу с внешним ключом, Сразу под этими

списками находятся списки для указания связываемых полей. Выберите для таблиц **Подразделения** и **Запасы** поле **КодПодразделения**, как на рис. 16, и для таблиц **Товары** и **Запасы** — поле **КодТовара**, как на рис. 17.

Для создания связи щелкните кнопку **ОК**. Если указанные связи допустимы, окно закроется и связь установится.

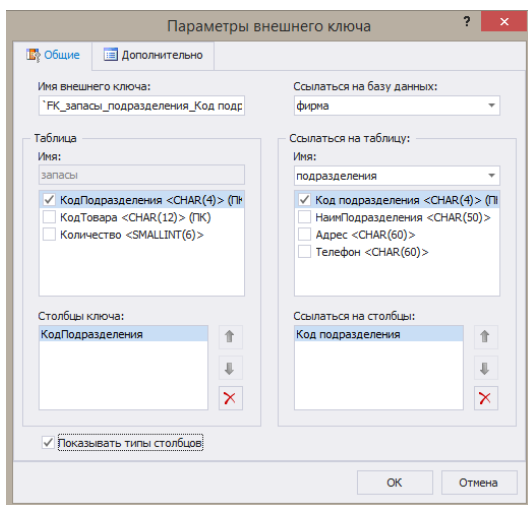


Рис. 16. Выберите для таблиц **Подразделения** и **Запасы** поле **КодПодразделения**

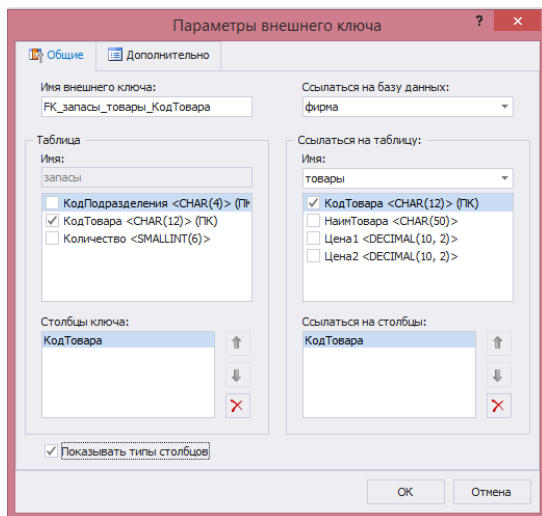


Рис. 17. Выберите для таблиц **Товары** и **Запасы** поле **КодТовара**

5.3. Импортирование таблиц в MySQL

Создать таблицу в базе на MySQL можно и по-другому: просто импортировав ее из любой ранее созданной базы данных. Это может быть основным способом создания таблиц, если вы, например, решите все свои разработки по базам данных в других СУБД перенести на MySQL.

На рис. 18 показано, как выбрать **(База данных | Импорт данных)** операцию импорта таблицы.

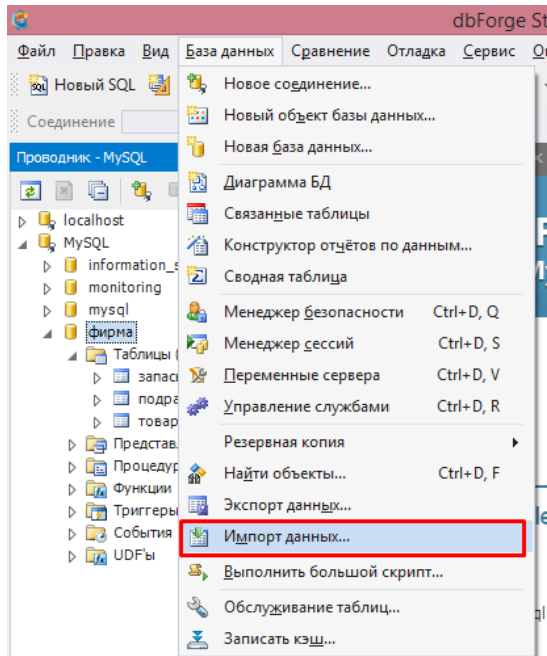


Рис. 18. Выбор операции импорта таблицы

После выбора вами **Импорта данных** вас «окружит своей заботой» соответствующий «помощник» и перепись таблицы из баз многих форматов покажется вам приятным занятием. На рис. 19 показано окно импорта данных. Для начала необходимо выбрать формат импорта и файл, из которого будут выгружаться данные и нажать на кнопку **Далее>**.

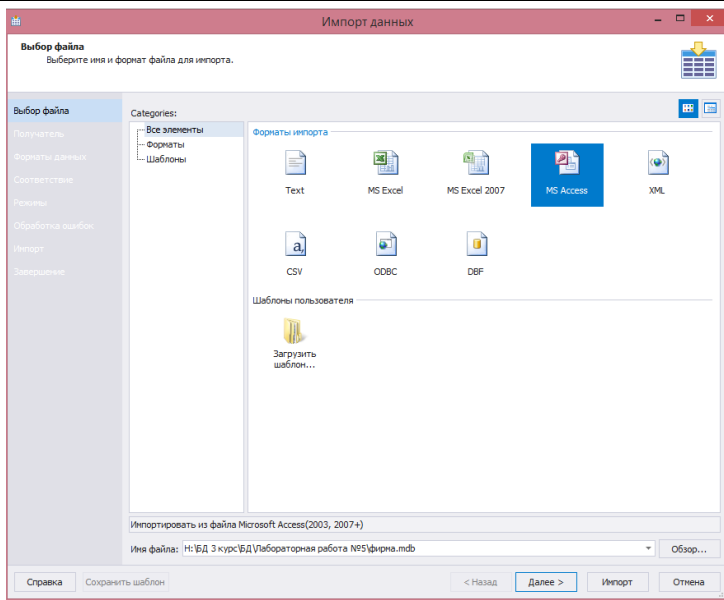


Рис. 19. Для начала необходимо выбрать **формат импорта и файл**, из которого будут выгружаться данные

ПРИМЕЧАНИЕ: Для выполнения перевода у вас должен быть установлен Microsoft Access Database Engine. Он отвечает за установку компонентов, которые могут быть использованы для облегчения передачи данных между файлами Microsoft Access и не-Microsoft приложениями. В противном случае, мастер импорта покажет сообщение об ошибке (рисунок 20).

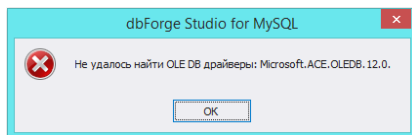


Рис. 20. Сообщение об ошибке при отсутствии компонента **Microsoft Access Database Engine**

Обратите внимание, что разрядные версии вашей ОС Windows и Microsoft Access Database Engine должны совпадать, то есть, если у вас есть 64-битная система, вы должны использовать 64-битную версию программы установки. Тем не менее, бывают случаи, когда установлена 32-разрядная версия Microsoft Access на 64-битной ОС Windows. В этом случае выполните следующие действия перед установкой.

- 1) Нажмите кнопку **Пуск**, выберите **Программы/Служебные**.
- 2) Щелкните правой кнопкой мыши **Командная строка**, а затем выберите **Запуск от имени администратора**.
- 3) Укажите путь к файлу установщика, а также **/passive** в конце. Запрос должен выглядеть следующим образом:

```
C:\Users\Агния\Downloads>AccessDatabaseEngine_x64.exe /passive
```

В случае выше ОС Windows является 64-разрядной, но установленная версия Microsoft Access является 32-разрядной. Именно поэтому требуется 64-разрядная программа установки.

Для дальнейшей работы необходимо понять, что и куда следует импортировать. В качестве примера далее приведен процесс импорта таблиц в базу данных **Фирма**.

Далее (рис. 20) мы имеем возможность указать таблицы, которые следует импортировать. Выделите таблицу для импорта и укажите, куда именно следует импортировать таблицы и нажмите на кнопку **Далее>**.

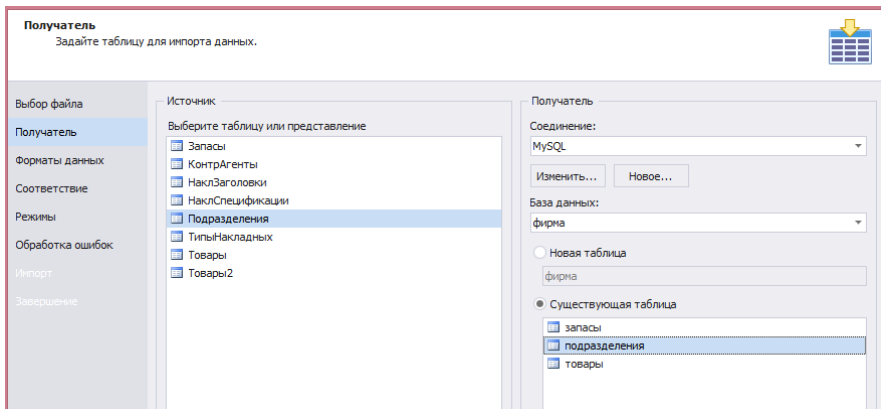


Рис. 21. Выделите таблицу для импорта и укажите, куда именно следует импортировать таблицу.

В следующем окне (рис. 21) мы имеем возможность настроить формат импортируемых данных, в случае, если он автоматически определился неверно. и вновь нажмите на кнопку **Далее>**.

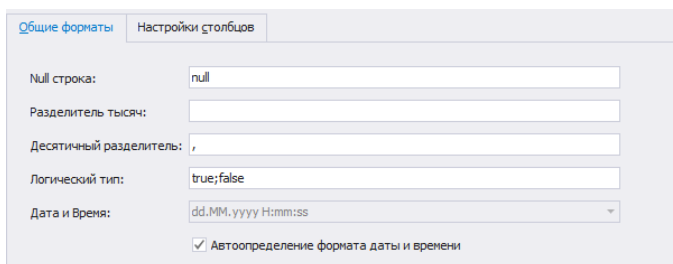


Рис. 22. В этом окне мы имеем возможность настроить формат импортируемых данных

На рис. 22 представлено окно соответствия импортируемых данных и уже существующей таблицы. На данном этапе можно устранить ошибки, касающиеся неверного распределения данных, которые мы получаем из исходной таблицы. Также здесь можно просмотреть содержимое импортируемой таблицы. Нажмите на кнопку **Далее>**.

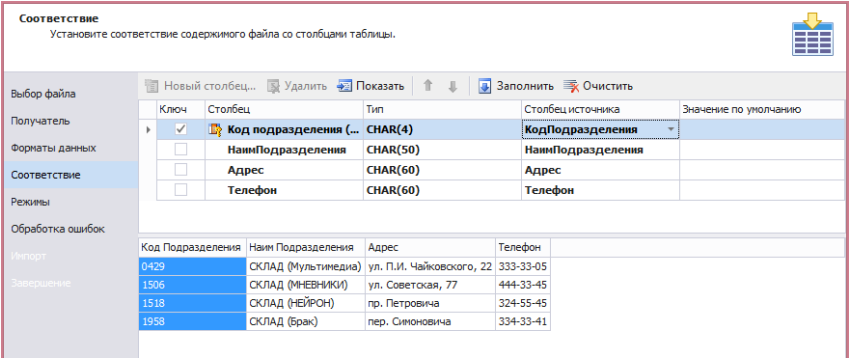
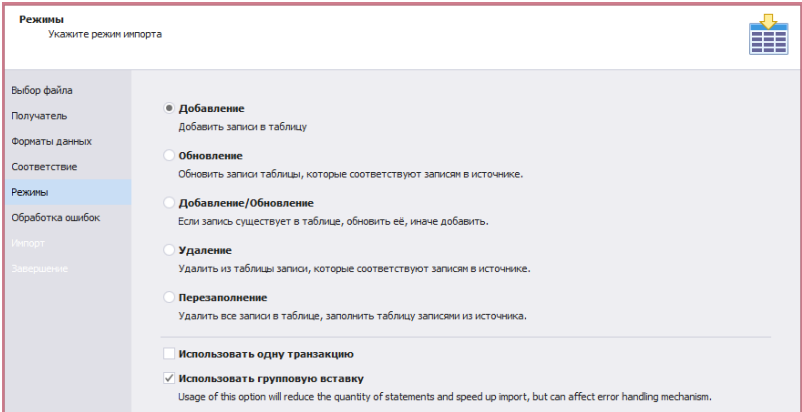


Рис. 23. Окно соответствия импортируемых данных и уже существующей таблицы.

Далее необходимо выбрать, в каком режиме происходит импорт данных. Подробное описание каждого режима есть в меню выбора (рисунок 23).



После этого Вам останется лишь настроить обработчик ошибок при импорте (рисунок 23) и нажать кнопку **Импорт**.

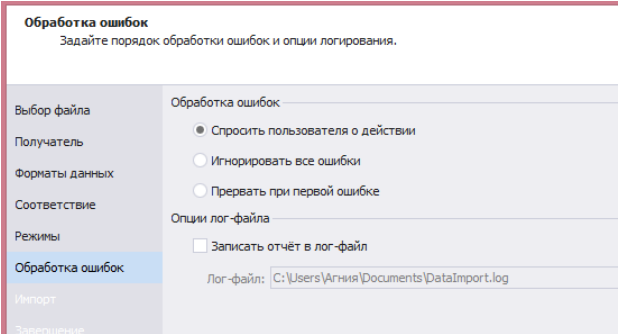


Рис. 24. Вам останется лишь настроить обработчик ошибок при импорте и нажать кнопку **Импорт**

После передачи информации между базами данных «помощник» обязательно сообщает о результатах. В том числе, и ошибках (рисунк 24). Кроме того, Вы и сами можете посмотреть, что теперь хранится в той или иной таблице после выполнения операции импорта. Если необходимо импортировать еще таблицы, то нужно нажать на кнопку **Импортировать еще**, а если нет – то нажать кнопку **Завершить**.

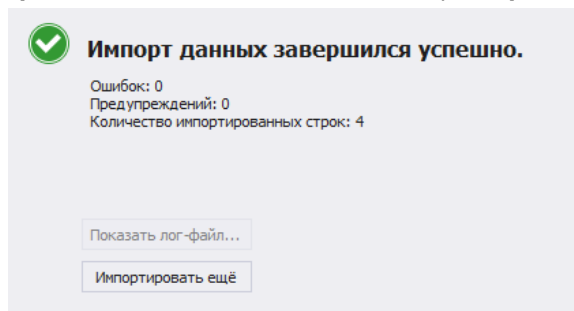


Рис. 25. После передачи информации между базами данных «помощник» обязательно сообщает о результатах

Код подразделения	НаимПодразделения	Адрес	Телефон
10429	СКЛАД (Мультимедиа)	ул. П.И. Чайковского, 22	333-33-05
1506	СКЛАД (МНЕВНИКИ)	ул. Советская, 77	444-33-45
1518	СКЛАД (НЕЙРОН)	пр. Петровича	324-55-45
1958	СКЛАД (Брак)	пер. Симоновича	334-33-41

Рис. 26. Так может выглядеть таблица **Подразделения**, импортированная из mdb-файла

ЗАДАНИЕ. Для закрепления навыков импорта данных из базы данных MS Access импортируйте ранее созданную базу данных «Фирма» в MySQL в следующем порядке:

- 1) подразделения;
- 2) товары;
- 3) запасы (сначала удалите предпоследнюю запись из БД – источника, иначе при импорте возникнет ошибка и данные не будут переданы, т.к. в таблице будут продублированы значения двух первых(ключевых) полей)
- 4) КонтрАгенты (получатель: фирма, новая таблица);
- 5) ТипыНакладных;
- 6) НаклЗаголовки;
- 7) НаклСпецификации.

5.4. Создание диаграммы базы данных

Чтобы построить диаграмму базы данных нужно в пункте меню **База данных** выбрать пункт **Диаграмма БД**, как показано на рисунке 27.

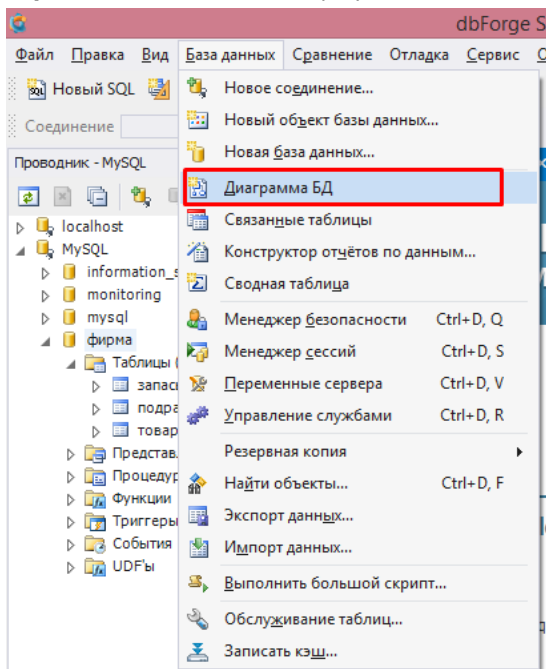


Рис. 27. В пункте меню **База данных** выберите пункт **Диаграмма БД**.

После этого в рабочей области менеджера откроется окно под названием «Диаграмма1.dbd» (рисунок 28).

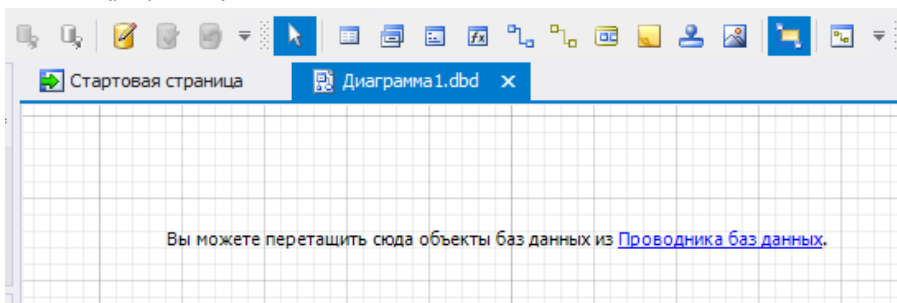


Рис. 28. Окно для работы с диаграммами БД

В это окно необходимо перетащить из **Проводника** все таблицы, из которых следует создать диаграмму базы данных и она построится автоматически (рисунок 29).

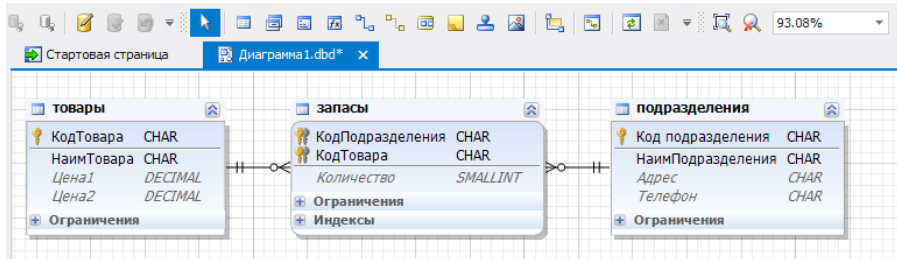


Рис. 29. Диаграмма БД, созданной в разделах 5.1 и 5.2

ЗАДАНИЕ. Создайте диаграмму базы данных «фирма», включив в нее только таблицы «товары», «запасы» и «подразделения». Сравните полученный вами результат с рисунком 29.

Добавьте на диаграмму остальные таблицы.

Обратите внимание, что таблицы, импортированные из MS Access напрямую, без предварительного создания таблиц-получателей, никак не связаны с остальными таблицами (рисунок 30). Причиной этого является тот факт, что при импорте данных в новую таблицу информация о первичных и вторичных ключах не сохраняется.

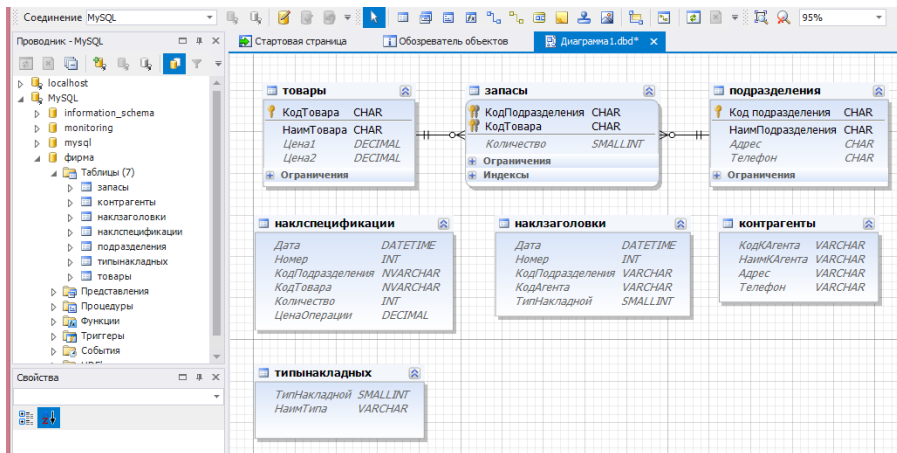


Рис. 30. Таблицы базы данных **Фирма**

Для начала, нам необходимо установить первичные ключи (прямо на диаграмме), а затем заняться связями между таблицами. Для определения первичного ключа выделите соответствующее поле в таблице, например, **КодАгента** в таблице **контрагенты**, и дважды щелкните на нем левой кнопкой мыши. На экране появится окно свойств столбца (рисунок 31). Установите флажок напротив характеристики «Первичный ключ», удалите значение по умолчанию и сохраните изменения нажав кнопку «Обновить базу».

Рис. 31. Окно свойств столбца **КодАгента** в таблице **контрагенты**

ЗАДАНИЕ. Установите первичные ключи для таблиц **контрагенты** и **типынакладных**.

Теперь можно установить связи между таблицами. Эта операция выполняется так же, как и в Microsoft Access: выделите первичный ключ на объектной таблице и, не отпуская кнопку мыши, переместите курсор мыши на соответствующее поле связанной таблицы, а затем отпустите кнопку мыши.

Обратите внимание, что невозможно таким образом установить связь между таблицами, у которых отсутствует первичный ключ. Поэтому надежнее будет использовать добавление вторичного ключа вручную. Для этого необходимо щелкнуть правой клавишей мыши на названии таблицы и в пункте **«Создать...»** выбрать подпункт **«Новый внешний ключ...»** и добавить связи этой таблицы с другими в текущей базе данных по аналогии с тем, как описано в пункте 5.3. Таким образом, мы получим набор ограничений для таблицы **наклаголовки**, отображенный на рисунке 32.

Имя	Столбцы	Тип
FK_наклаголовки_подразделения_Код подразделения	КодПодразделе...	Внешний ключ
FK_наклаголовки_контрагенты_КодКАгента	КодАгента	Внешний ключ
FK_наклаголовки_типынакладных_ТипНакладной	ТипНакладной	Внешний ключ

Рис. 32. Набор ограничений для таблицы **наклаголовки**

ЗАДАНИЕ. Установите все необходимые связи между таблицами БД «Фирма», ориентируясь на БД – источник.

На рисунке 33 показана вся диаграмма, которую необходимо было получить. Пока она не имеет только собственного наименования.

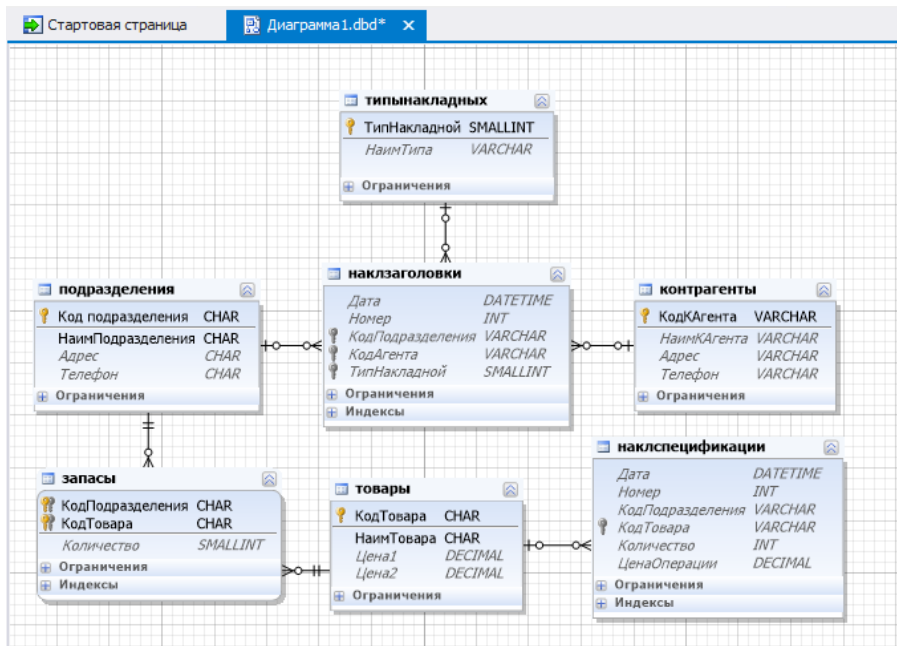


Рис. 33. Диаграмма, которую необходимо было получить

Для присвоения построенной диаграмме наименования следует щелкнуть кнопку **Сохранить** и в окне **Сохранить файл как** (рис. 34) ввести это наименование или оставить то, которое предлагается по умолчанию.

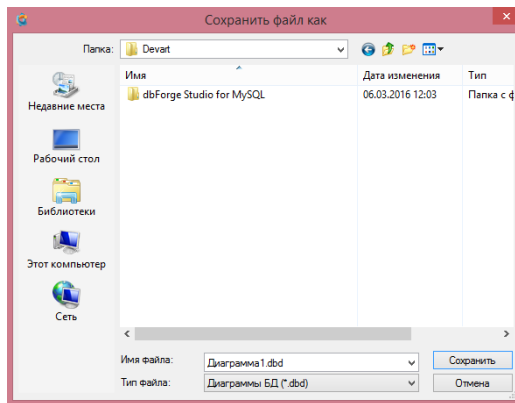


Рис. 34. Введите наименование диаграммы или оставьте то, которое предлагается по умолчанию

Остается добавить, что при работе с диаграммой можно не только изменять структуру таблиц, но и создавать новые таблицы, устанавливать связи между этими таблицами, а также создавать запросы и пользовательские функции. Кроме того, есть весьма полезная функция экспорта диаграммы в картинку. Для этого необходимо щелкнуть правой клавишей на рабочую область диаграммы (не на таблицу или связь) и выбрать пункт «Экспортировать в картинку...». Затем выбрать место и формат сохранения картинки. Результатом экспорта будет изображение, представленное на рисунке 35.

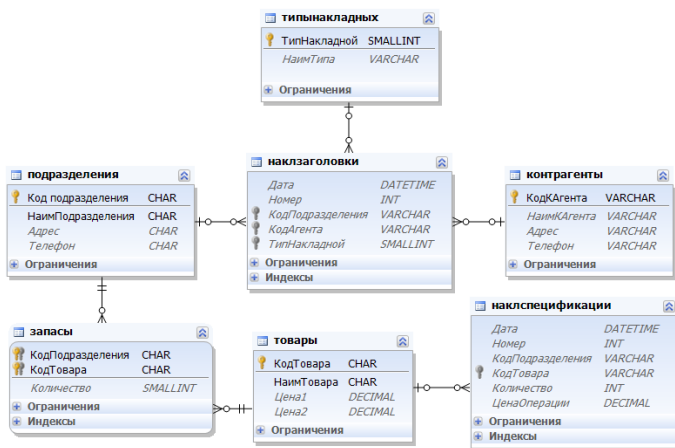


Рис. 35. Результат экспорта диаграммы БД в картинку

Продемонстрируйте Вашу работу преподавателю!

КОНТРОЛЬНЫЕ ВОПРОСЫ:

1. Расскажите, что вы знаете о СУБД типа клиент/сервер MySQL?
2. Какие возможности предоставляет СУБД типа клиент/сервер MySQL?
3. Что представляет собой физическая реализация базы данных в MySQL?
4. Перечислите системные (встроенные) базы MySQL.
5. Назовите объекты, в которых располагаются системные и пользовательские базы данных в MySQL. Охарактеризуйте эти объекты.
6. Какие возможности предоставляет для работы с базами данных утилита dbForge Studio for MySQL?
7. Перечислите основные этапы работы с MySQL.
8. Как создать базу данных при помощи dbForge Studio for MySQL?
9. Как осуществить импортирование таблиц в MySQL из Microsoft Access?
10. Расскажите о процессе создания диаграмм баз данных в dbForge Studio for MySQL.