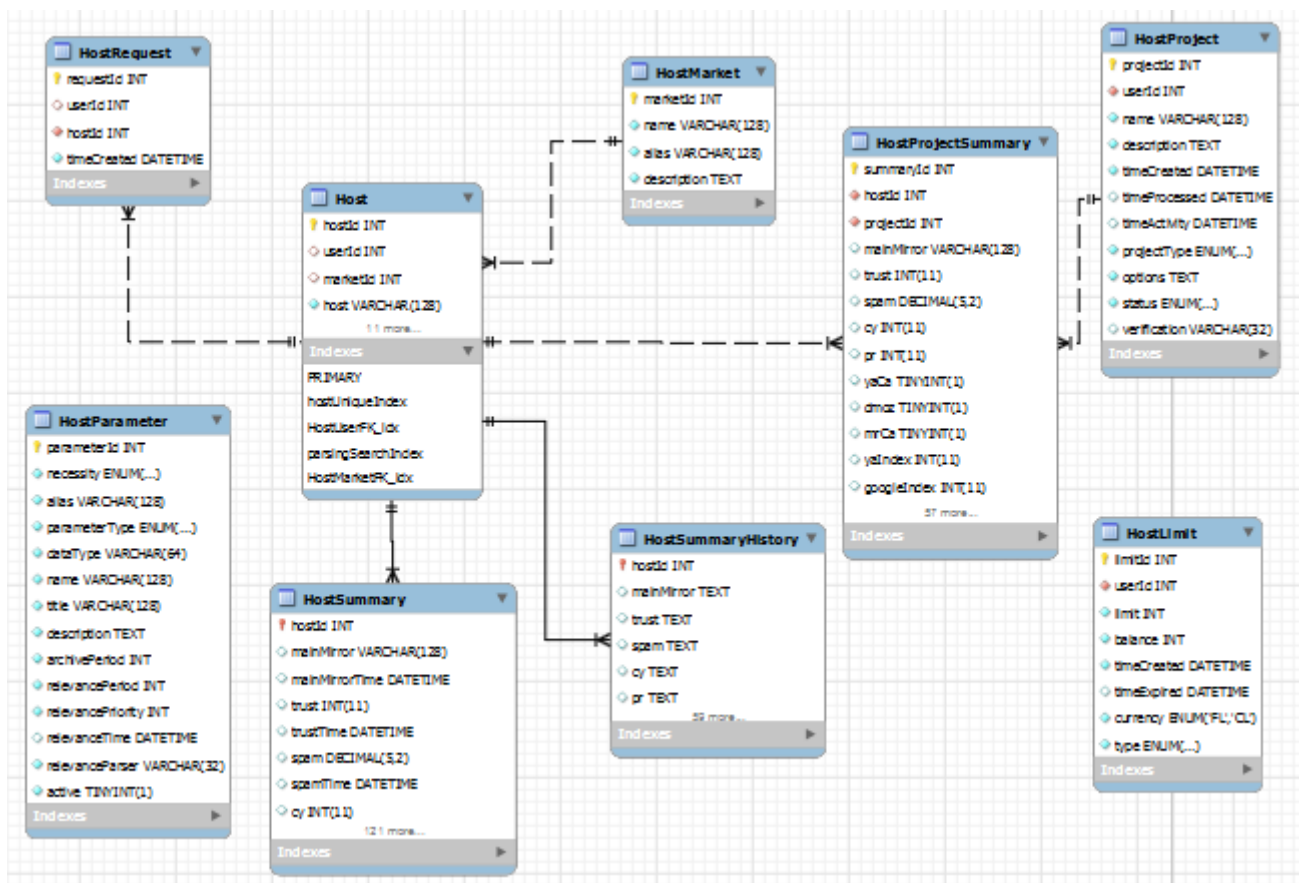


Основы работы с MySQL Workbench: быстрый старт, управление схемой данных

Веб-разработчик растёт вместе с проектами, которые он создаёт и развивает. С ростом проектов увеличивается сложность программной части, неизбежно возрастает количество обрабатываемых ею данных, а так же **сложность схемы данных**. Общение с другими вебщиками показывает, что огромной популярностью среди нас пользуются базы данных MySQL, а для управления ими - небезызвестный **PHPMyAdmin**. Переходя от маленьких проектов к большим, от cms к фреймворкам, многие, как и я, остаются верны MySQL. Однако для проектирования сложной базы данных с большим количеством таблиц и связей, возможностей PHPMyAdmin катастрофически не хватает. Поэтому я решил написать обзор **MySQL Workbench - замечательной бесплатной десктопной программы для работы с MySQL**.

В первой части обзора я расскажу о самых основах работы с программой, так что, можете использовать эту статью как **руководство начинающего пользователя**. [Вторая часть](#) будет посвящена использованию Workbench в бою при работе с удалённым сервером. В ней я дам базовые **инструкции** и рекомендации по настройке подключения сервера и синхронизации с ним.

MySQL Workbench — инструмент для визуального проектирования баз данных, интегрирующий проектирование, моделирование, создание и эксплуатацию БД в единое бесшовное окружение для системы баз данных MySQL.



Должен сказать, что программа действительно великолепная. Она позволяет быстро и с удовольствием накидывать **схемы данных проекта**, проектировать **сущности и связи** между ними, безболезненно **внедрять изменения** в схему и так же быстро и безболезненно **синхронизировать** её с удалённым сервером. А графический редактор **ЕER-диаграмм**, напоминающих забавных таракашек, позволяет увидеть общую картину модели данных и насладиться её лёгкостью и элегантностью :) После первой же пробы этот инструмент становится незаменимым помощником в боевом арсенале веб-программиста.

Скачать MySQL Workbench

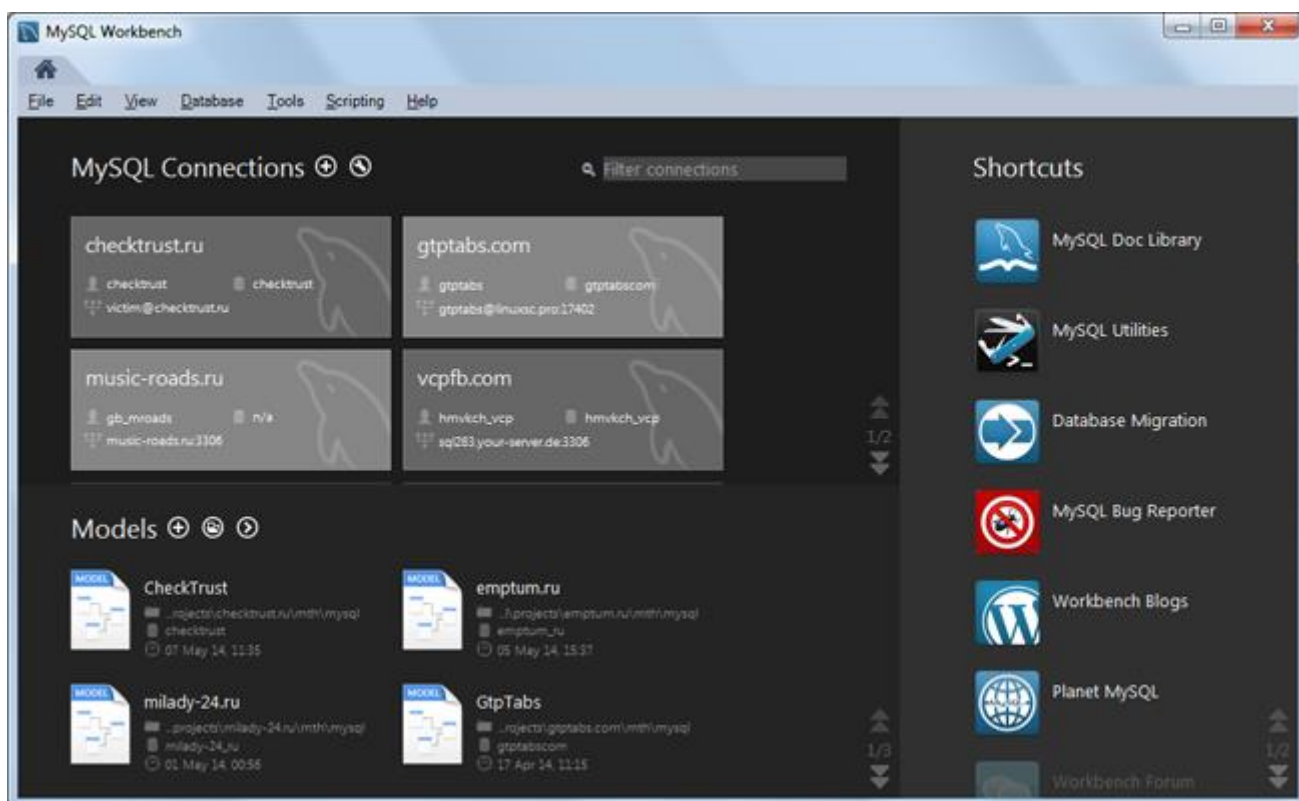
Дистрибутив MySQL Workbench доступен на [этой странице](#). Самая свежая версия программы на момент написания статьи - *Version 6.1*. Перед скачиванием требуется выбрать одну из следующих платформ:

- Microsoft Windows (*доступны MSI Installer и ZIP архив*)
- Ubuntu Linux
- Fedora
- Red Hat Enterprise Linux / Oracle Linux
- Mac OS X

После выбора платформы вам предлагают зарегистрироваться или авторизоваться в Oracle. Если не хотите, внизу есть ссылка **"No thanks, just start my download"** - жмите на неё ;)

Начало работы

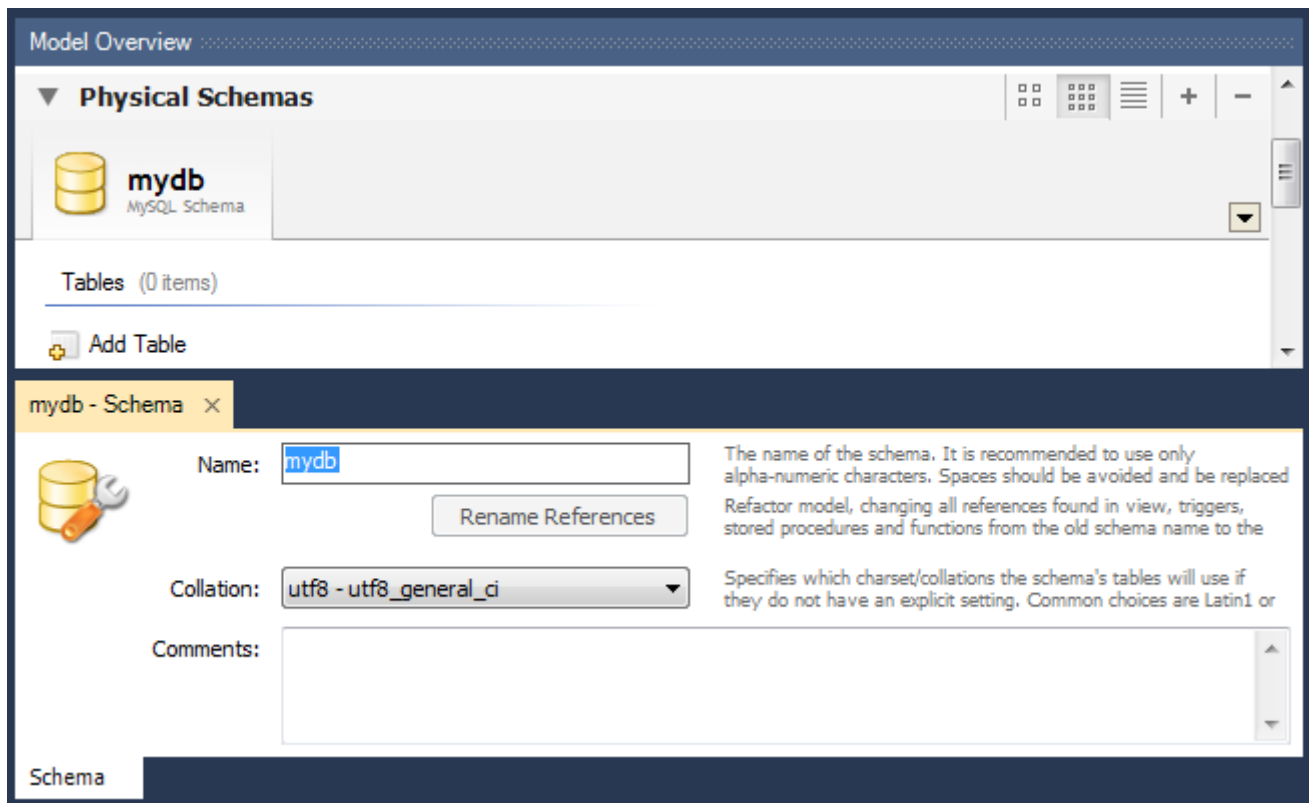
Стартовый экран программы отражает основные направления её функциональности - проектирование моделей баз данных и их администрирование:



В верхней части экрана находится список подключений к MySQL серверам ваших проектов, а список последних открытых моделей данных - в нижней части экрана. Работа обычно начинается с **создания схемы данных** или **загрузки существующей структуры в MySQL Workbench**. Приступим к работе!

Создание и редактирование модели данных

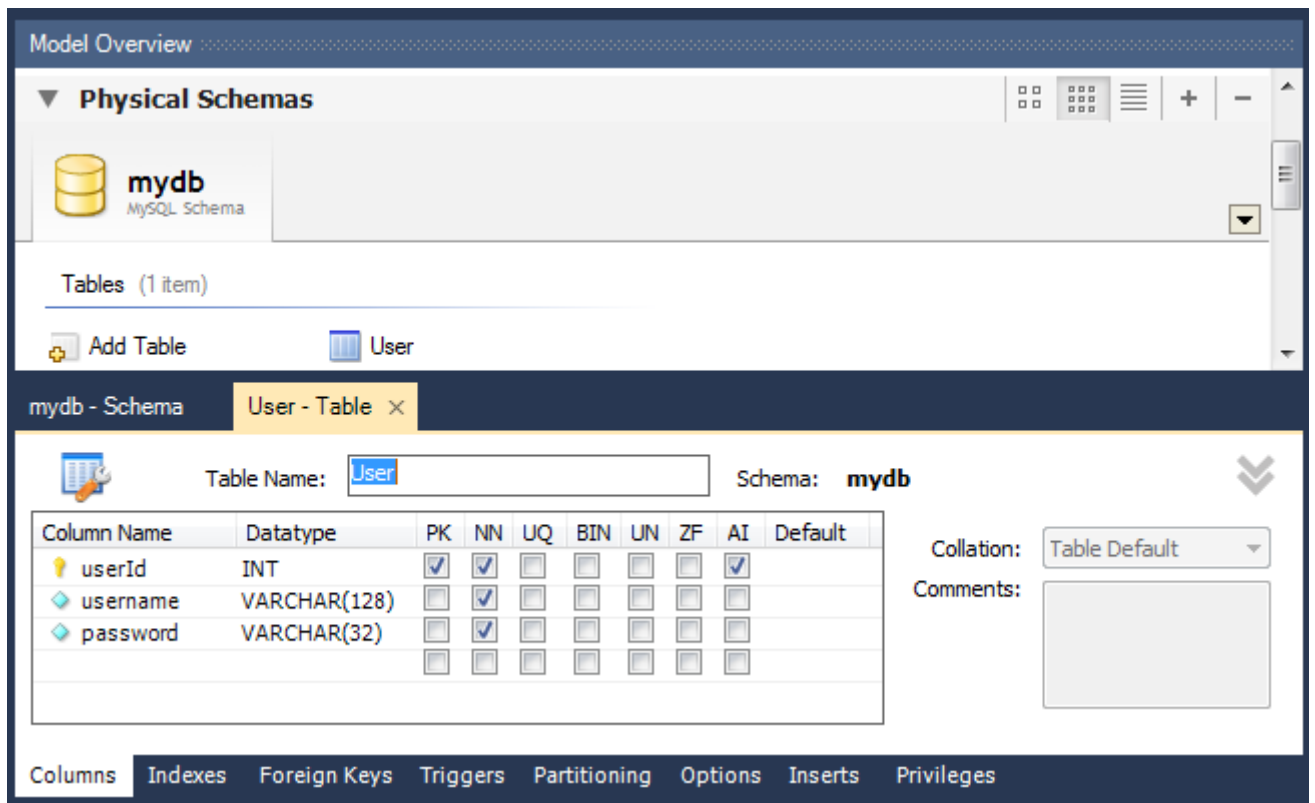
Для добавления модели нажимаем плюсик рядом с заголовком "Models" или выбираем *"File → New Model"* (*Ctrl + N*):



На этом экране вводим имя базы данных, выбираем кодировку по умолчанию и, если нужно, заполняем поле комментария. Можно приступить к созданию таблиц.

Добавление и редактирование таблицы

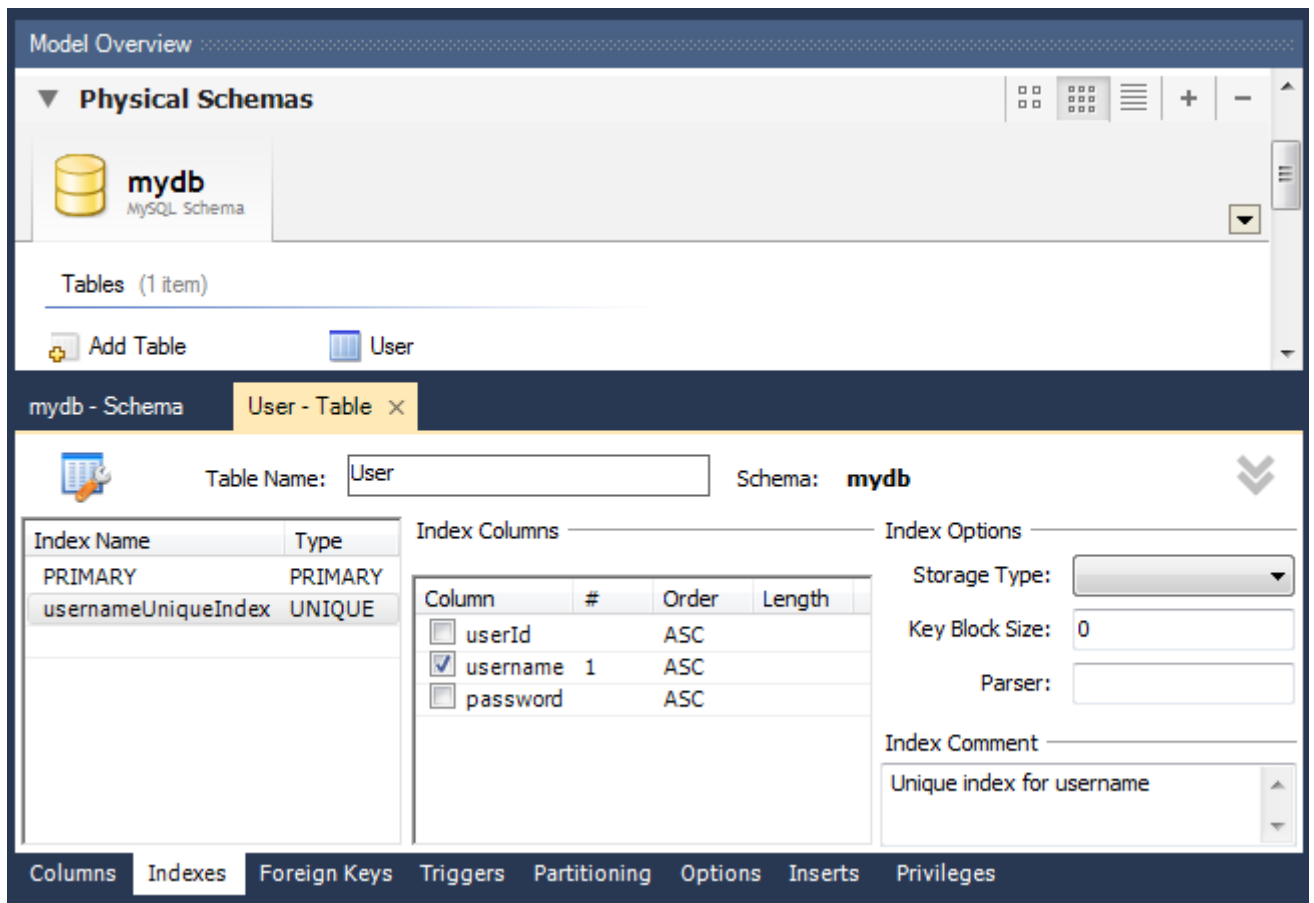
Список баз данных проекта и список таблиц в пределах базы данных будет располагаться во вкладке "*Physical Schemas*". Чтобы создать таблицу, дважды кликаем на "+Add Table":



Откроется удобный интерфейс для редактирования списка полей и их свойств. Здесь мы можем задать название поля, тип данных, а так же установить для полей различные атрибуты: назначить поле *первичным ключом (PK)*, пометить его *Not Null (NN)*, *бинарным (BIN)*, *уникальным (UQ)* и другие, установить для поля *авто-инкрементирование (AI)* и *значение по умолчанию (Default)*.

Управление индексами

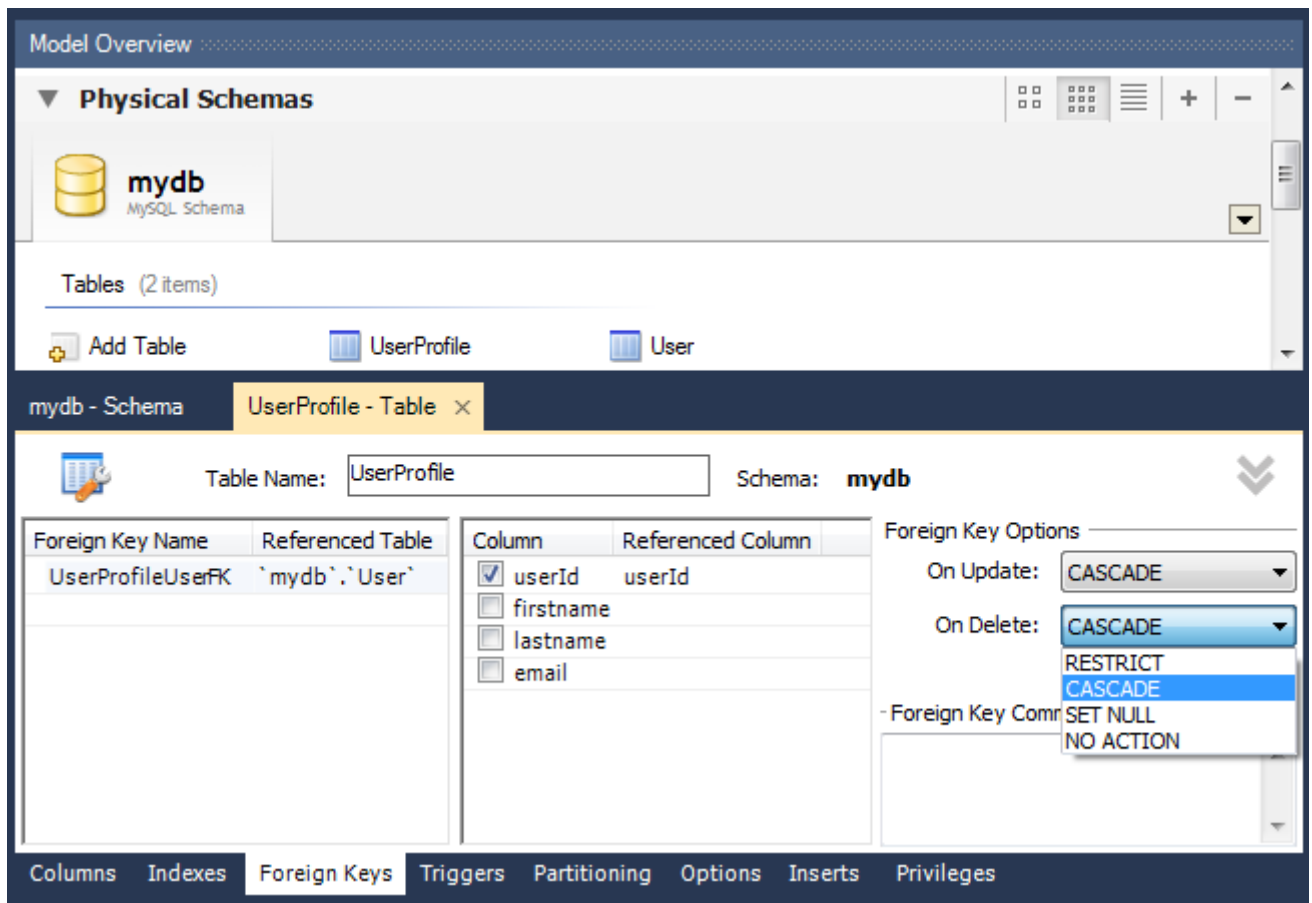
Добавлять, удалять и редактировать индексы таблиц можно во вкладке "*Indexes*" интерфейса управления таблицей:



Вводим название индекса, выбираем его тип, затем галочками помечаем в нужном порядке список полей, участвующих в данном индексе. Порядок полей будет соответствовать порядку, в котором были проставлены галочки. В данном примере я добавил уникальный индекс к полю *username*.

Связи между таблицами

Установка внешних ключей и связывание таблиц возможно только для таблиц *InnoDB* (эта система хранения данных выбирается по умолчанию). Для управления связями в каждой таблице находится вкладка "*Foreign Keys*":



Для добавления связи открываем вкладку *"Foreign Keys"* дочерней таблицы, вводим имя внешнего ключа и выбираем **таблицу-родителя**. Далее в средней части вкладки в графе *Column* выбираем поле-ключ из дочерней таблицы, а в графе *Referenced Column* - соответствующее поле из родительской таблицы (тип полей должен совпадать). При создании внешних ключей в дочерней таблице **автоматически создаются соответствующие индексы**.

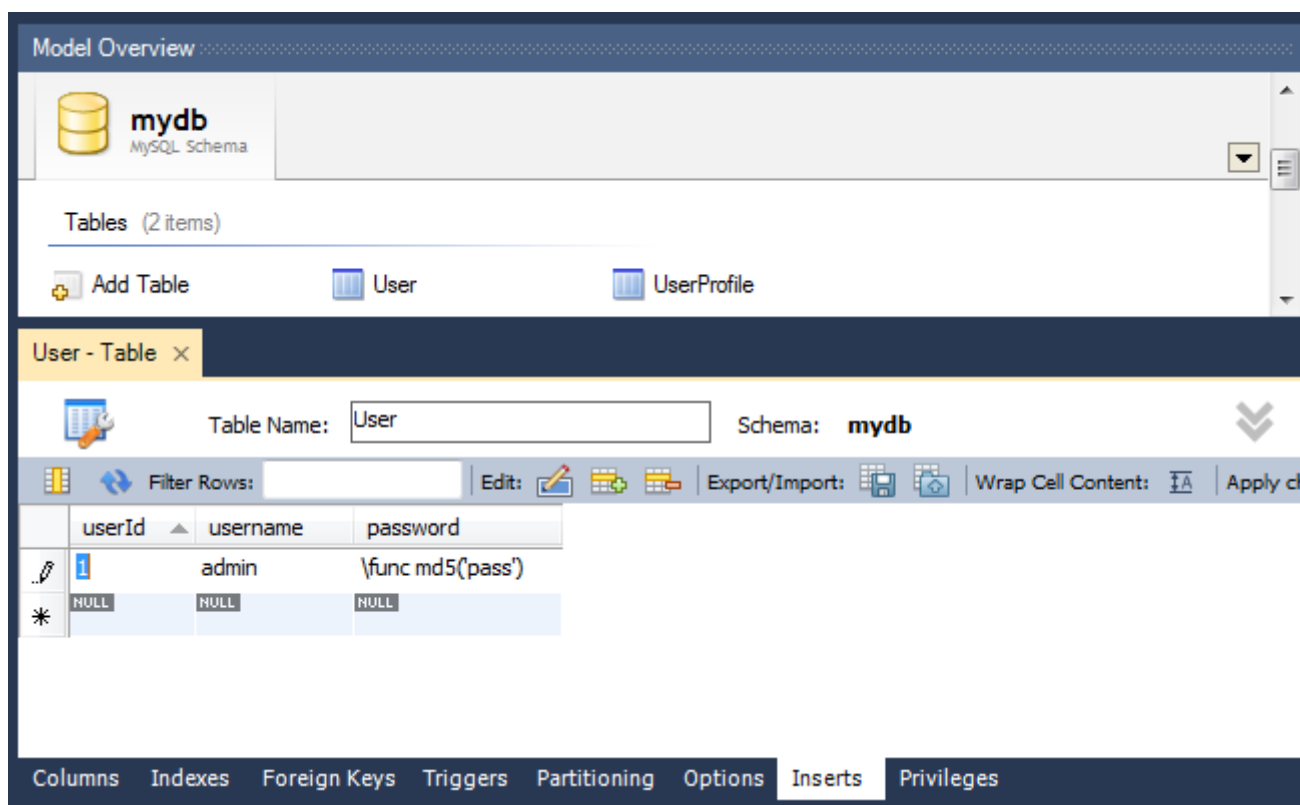
В разделе *"Foreign Key Options"* настраиваем поведение внешнего ключа при изменении соответствующего поля (*ON UPDATE*) и удалении (*ON DELETE*) родительской записи:

- *RESTRICT* - выдавать ошибку при изменении / удалении родительской записи
- *CASCADE* - обновлять внешний ключ при изменении родительской записи, удалять дочернюю запись при удалении родителя
- *SET NULL* - устанавливать значение внешнего ключа *NULL* при изменении / удалении родителя (**неприемлемо для полей, у которых установлен флаг *NOT NULL!***)
- *NO ACTION* - не делать ничего, однако по факту эффект аналогичен *RESTRICT*

В приведённом примере я добавил к дочерней таблице *UserProfile* внешний ключ для связи с родительской таблицей *User*. При редактировании поля *userId* и удалении позиций из таблицы *User* аналогичные изменения будут **автоматически** происходить и со связанными записями из таблицы *UserProfile*.

Наполнение таблицы базовыми данными

При создании проекта в базу данных часто нужно добавлять стартовые данные. Это могут быть корневые категории, пользователи-администраторы и т.д. В управлении таблицами MySQL Workbench для этого существует вкладка "Inserts":

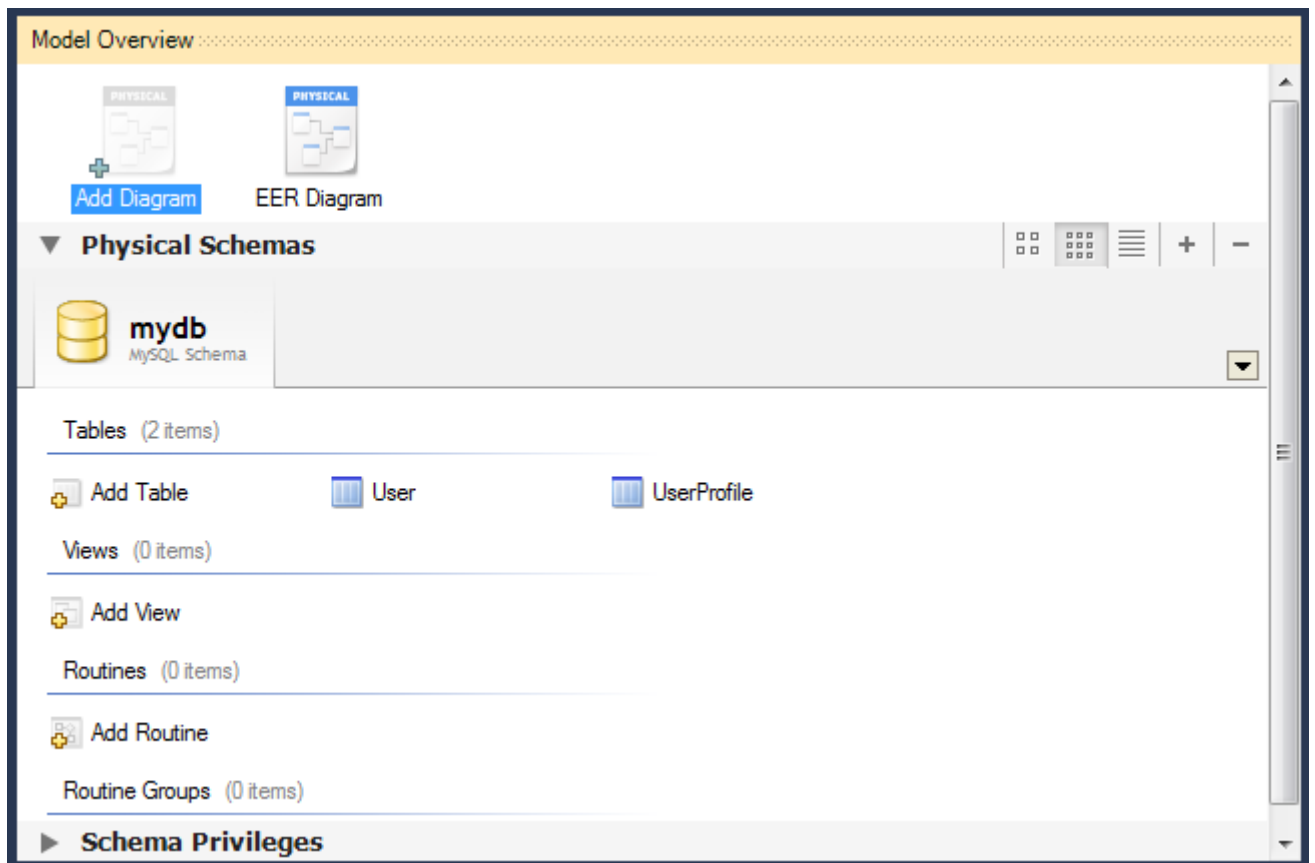


Как видно из примера, в случае, если перед записью в базу данных к данным нужно применить какую-то функцию MySQL, это делается с помощью синтаксиса `\func functionName('data')`, например, `\func md5('password')`.

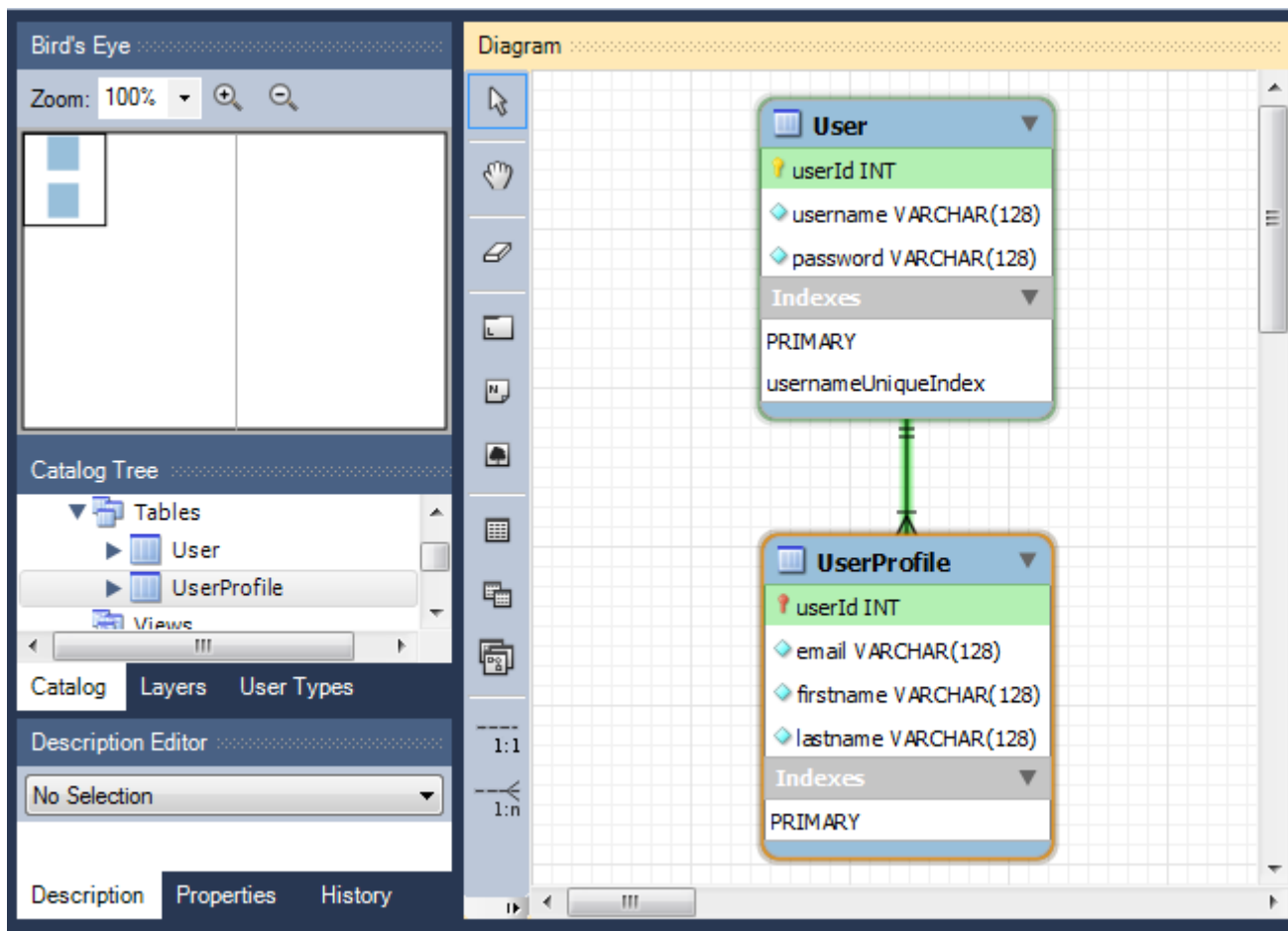
После ввода данных необходимо сохранить их в локальную базу данных нажатием на кнопку "Apply Changes".

Создание EER диаграммы (диаграммы "сущность-связь")

Для представления схемы данных, сущностей и их связей в графическом виде в MySQL Workbench существует редактор EER-диаграмм. Для создания диаграммы в верхней части экрана управления базой данных дважды кликаем на иконку "+Add Diagram":



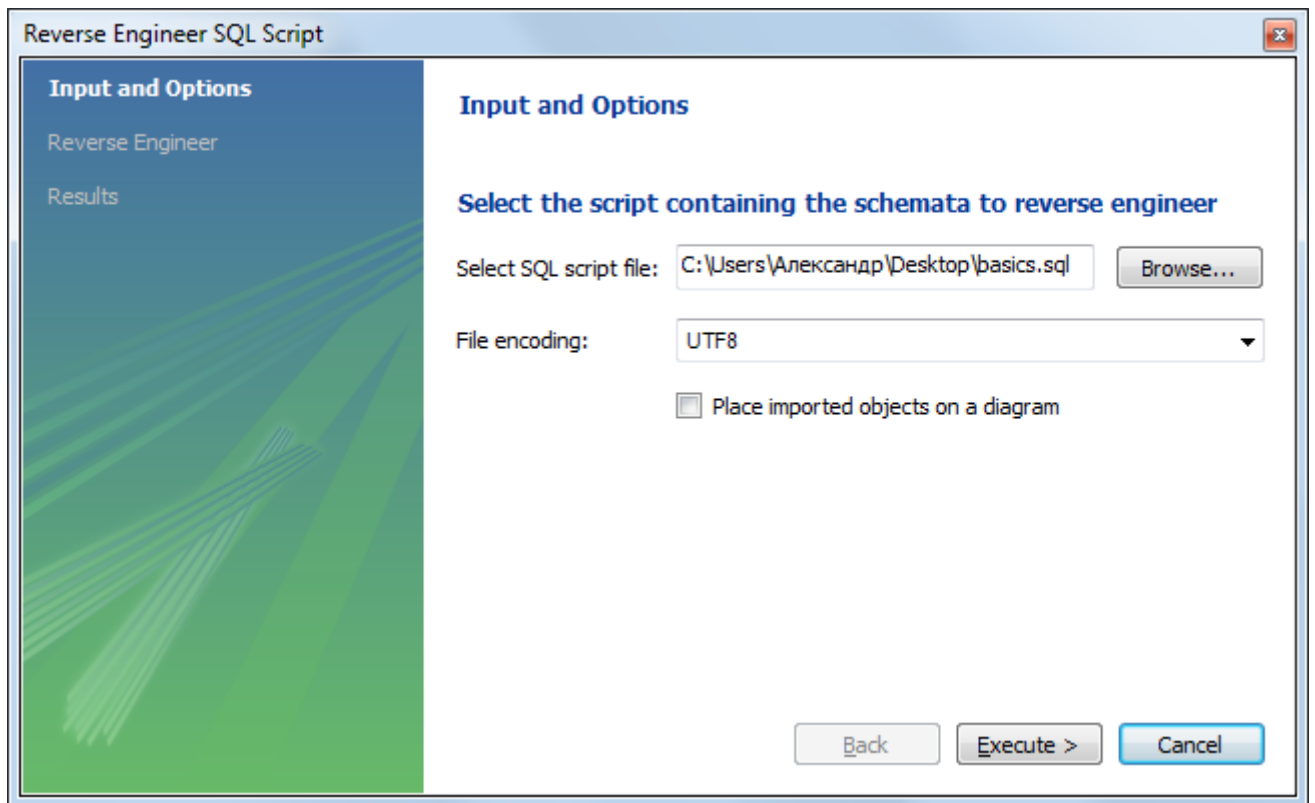
В его интерфейсе можно создавать и редактировать таблицы, добавлять между ними связи различных типов. Чтобы добавить уже существующую в схеме таблицу на диаграмму, просто перетащите её из панели "Catalog Tree".



Для экспорта схемы данных в графический файл выберите *"File → Export"*, а затем один из вариантов (*PNG, SVG, PDF, PostScript File*).

Импорт существующей схемы данных (из SQL дампа)

Если у нас уже есть схема данных, её можно без труда импортировать в MySQL Workbench для дальнейшей работы. Для импорта модели из SQL файла выбираем *"File → Import → Reverse Engineer MySQL Create Script..."*, после чего выбираем нужный SQL файл и жмём *"Execute >"*



В MySQL Workbench так же предусмотрен импорт и синхронизация модели данных напрямую с удалённым сервером. Для этого потребуется создать **подключение удалённого доступа к MySQL**, о которых я расскажу в [продолжении данного обзора](#).

Демо-проект из статьи доступен для скачивания по [этой ссылке](#). Желаю успехов и красивых **тараканек** схем!