**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ им. А. И. ГЕРЦЕНА**

Институт информационных технологий и технологического образования Кафедра компьютерные технологии и электронного обучения

Основная профессиональная образовательная программа

Направление подготовки 09.03.01 Информатика и вычислительная техника Направленность (профиль) «Технологии разработки программного обеспечения»

форма обучения - очная

ЛАБОРАТОРНАЯ РАБОТА №1

по дисциплине: «Базы данных»

Начало работы с MySQL. MySQL Workbench

Руководитель: доцент, кандидат физ.-мат. наук

Жуков Николай Николаевич

Автор работы

студентка 2 курса 1 группы 1 подгруппы

Мельникова Алена Сергеевна

Санкт-Петербург 2023

Задание 1. Используя справочные материалы по MySQL Workbench, опишите назначение пунктов меню Management (“Управление”), “Instance” (“Экземпляр БД”) и “Performance” (“Производительность”). В качестве примера описания используйте следующий пример для раздела Management (его описывать не надо).

В разделе “Navigator” есть три панели для управления базой данных:

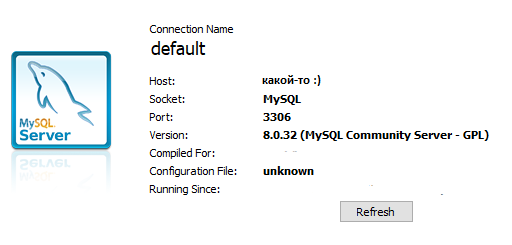
1. Management
2. Instance
3. Performance

Рассмотрим каждую из панелей и их функции.

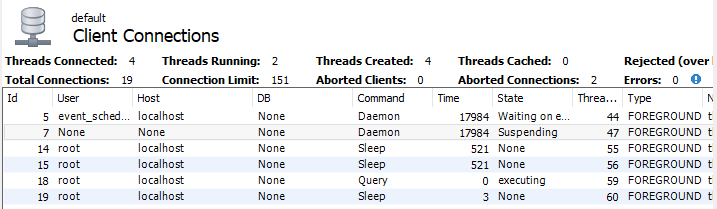
1. Панель “Management”.

Содержание панели:

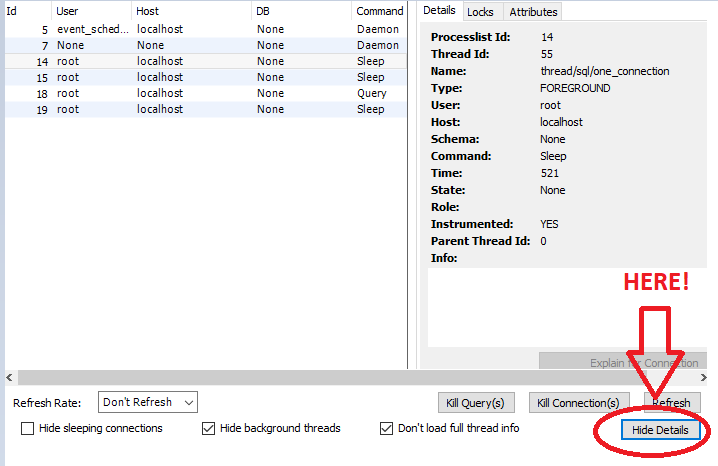
1) “Server Status” или “Состояние сервера”. Здесь можно проверить подключение. Можно узнать, например, название хоста, номер порта, версию MySQL:



2) “Client Connections” или “Клиентские соединения”. Здесь есть возможность просмотреть активные и спящие клиентские соединения MySQL и добавляет возможность уничтожать операторы и соединения, а также просматривать дополнительные сведения и атрибуты соединения.

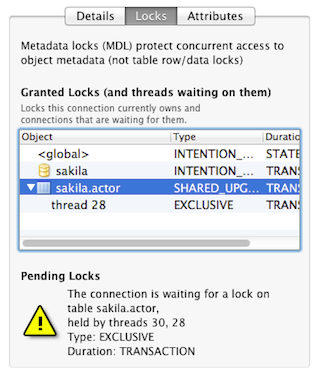


Также здесь можно посмотреть некоторые подробности при нажатии кнопки “Show Details”. Например, сведения о соединении, такие как идентификатор процесса, тип, пользователь, хост, инструмент и дополнительная информация. И соответственно закрыть эту вкладку можно с помощью кнопки “Hide Details”:

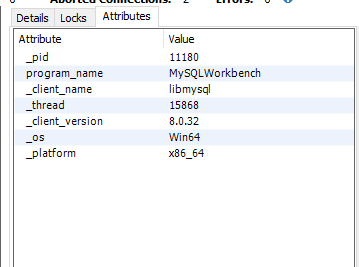


MySQL использует блокировку метаданных для управления доступом к таким объектам, как таблицы и триггеры. Иногда запрос может быть заблокирован, когда им управляет другое соединение от другого пользователя.

(Картинка из интернета)



Последнее – атрибуты. Это атрибуты подключения, такие как ОС, имя клиента, версия клиента и платформа.



3) “Users and Privileges” или “Пользователи привилегии”.

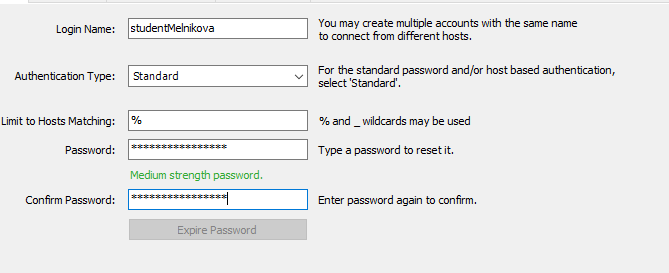
Здесь расположен список всех пользователей и их привилегии, относящиеся к активному экземпляру сервера MySQL. Здесь можно добавлять учетные записи, удалять, управлять ими.

В данном отделе есть кнопки:



(“Добавить аккаунт”, “Удалить”, “Обновить” соответственно).

Для более наглядного обзора создадим пользователя:

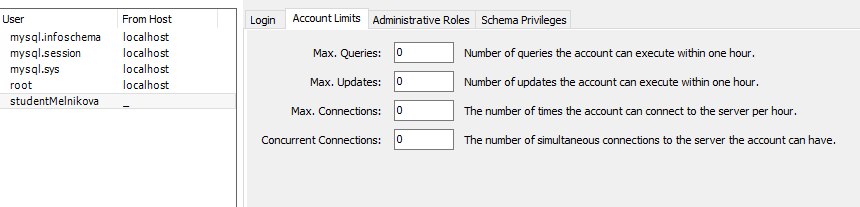


Был выбран логин для входа, тип аутентификации(выбран Standard, но есть возможность выбрать caching\_sha2\_password или SHA256\_Password для более надежного шифрования пароля), ограничение по совпадению хостов(знак процента ( %) соответствует нулю или более символов, а символ подчеркивания ( \_) соответствует одному символу) и придуман пароль для пользователя.

Нажмем кнопку для подтверждения:

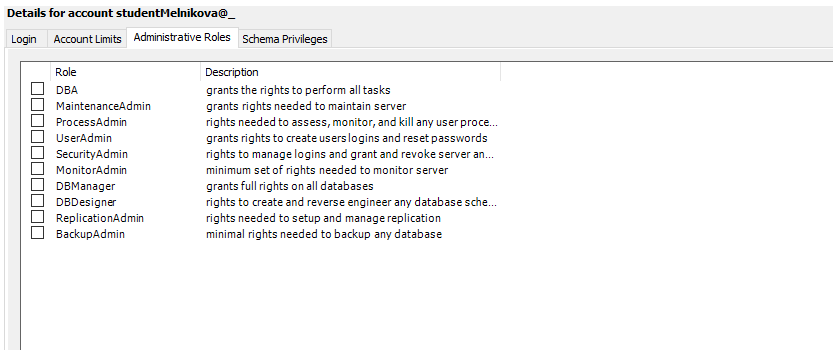


Следующий раздел “Account Limits” (“Ограничения учетной записи”):



1. Максимальные запросы: количество запросов, которые учетная запись может выполнить в течение одного часа.
2. Максимальные обновления: количество обновлений, которые учетная запись может выполнить в течение одного часа.
3. Максимальные подключения: сколько раз учетная запись может подключиться к серверу в течение часа.
4. Одновременные подключения: количество одновременных подключений к серверу, которое может иметь учетная запись.

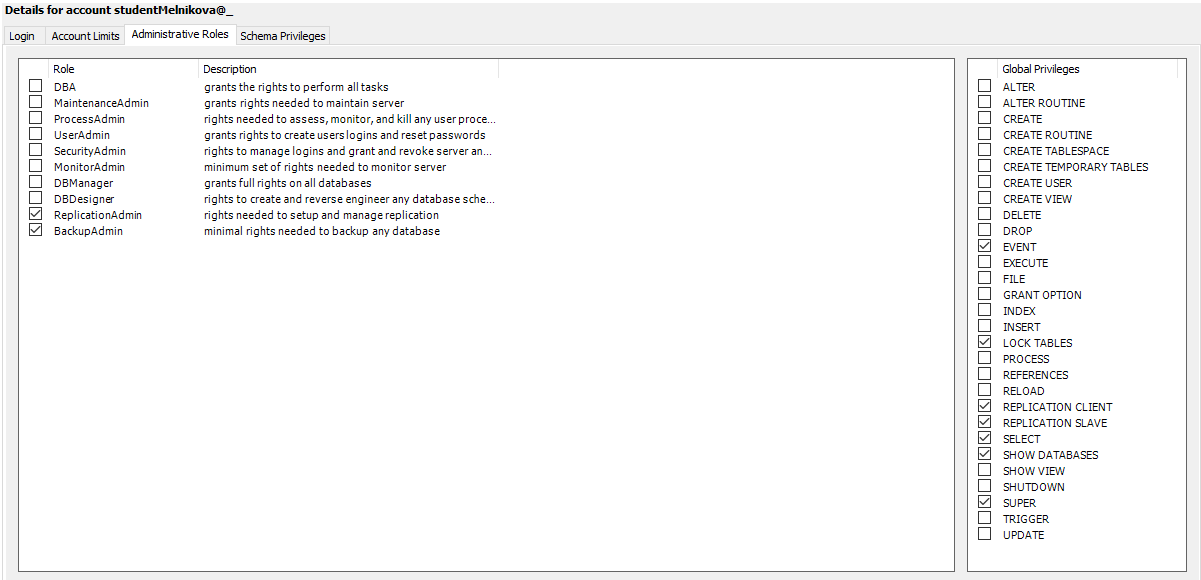
Раздел “Administrative Roles”(“Административные роли”):



Данная вкладка отвечает за привилегии пользователя. В зависимости от того, какую роль выполняет пользователь на сервере, можно выбрать ему следующие роли:

1. DBA: Предоставляет права на выполнение всех задач.
2. MaintenanceAdmin: Предоставляет права на обслуживание сервера.
3. ProcessAdmin: Предоставляет права на оценку, мониторинг и уничтожение пользовательских процессов.
4. UserAdmin: Предоставляет права на создание пользовательских логинов и сброс паролей.
5. SecurityAdmin: Предоставляет права на управление входом в систему, а также предоставляет и отзывает привилегии сервера.
6. MonitorAdmin: Предоставляет минимальные права для наблюдения за сервером.
7. DBManager: Предоставляет полные права на все базы данных.
8. DBDesigner: Предоставляет права на создание и реинжиниринг любой схемы базы данных.
9. ReplicationAdmin: предоставляет права, необходимые для настройки репликации и управления ею.
10. BackupAdmin: Предоставляет минимальные права, необходимые для резервного копирования любой базы данных.

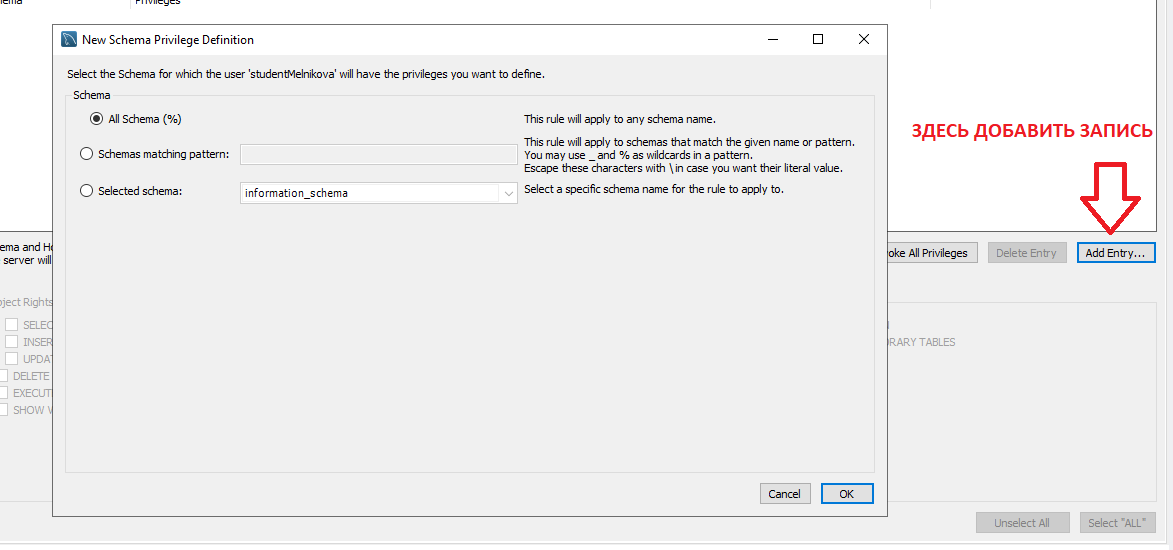
Исходя из выбранных ролей, отмечаются “Global Privileges”(“Глобальные привилегии”):



(представлен пример при выборе ролей “ReplicationAdmin” и “BackupAdmin”).

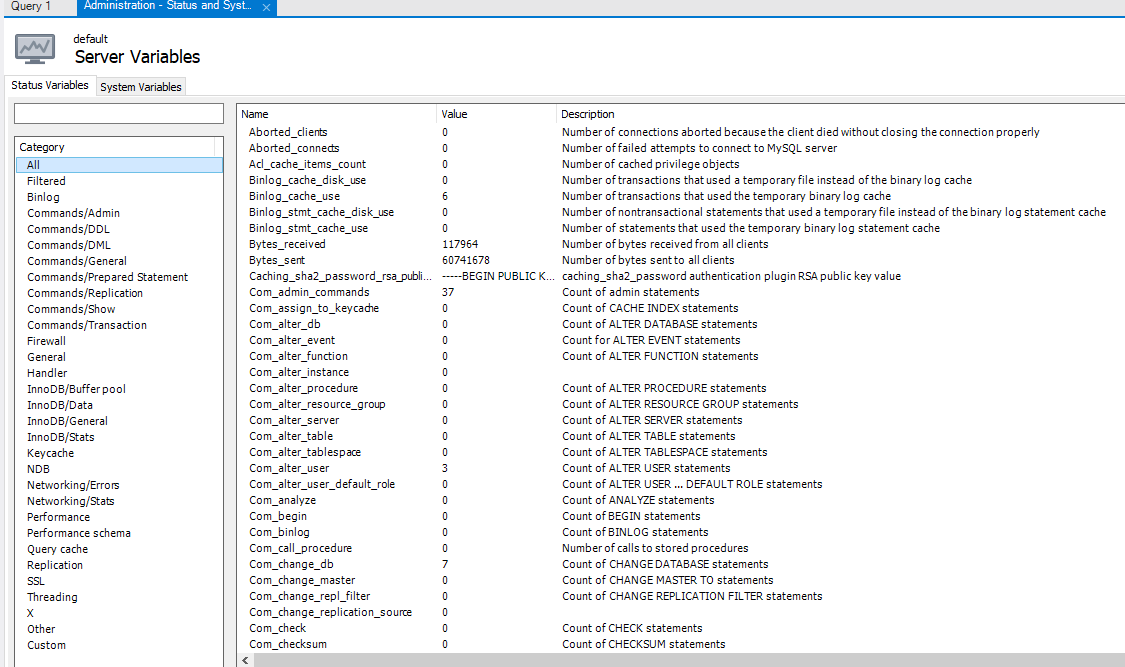
Раздел “Schema Privileges”:

На этой вкладке уточняется, как назначаются права доступа к одной или нескольким схемам по учетной записи пользователя.



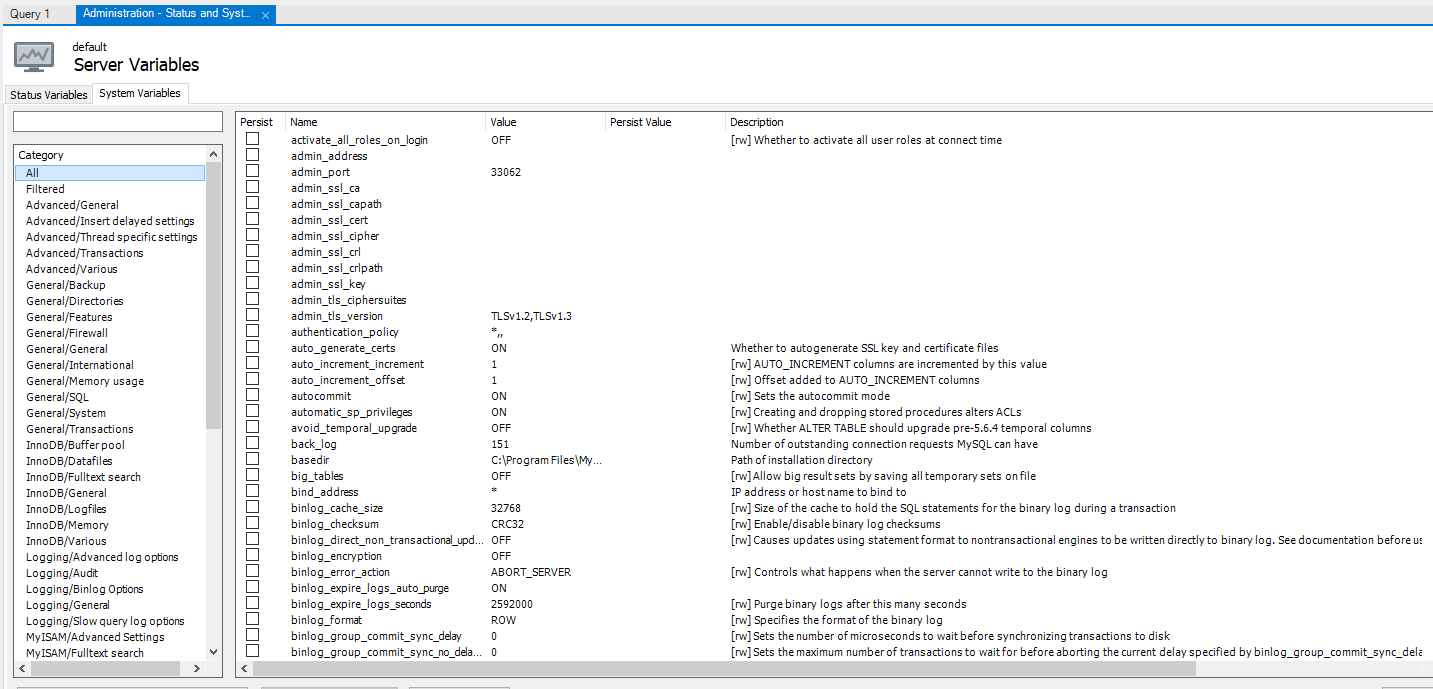
4) “Status and System Variables” или “Состояние и системные переменные”.

Данный раздел содержит полный набор серверных переменных для активного соединения MySQL.



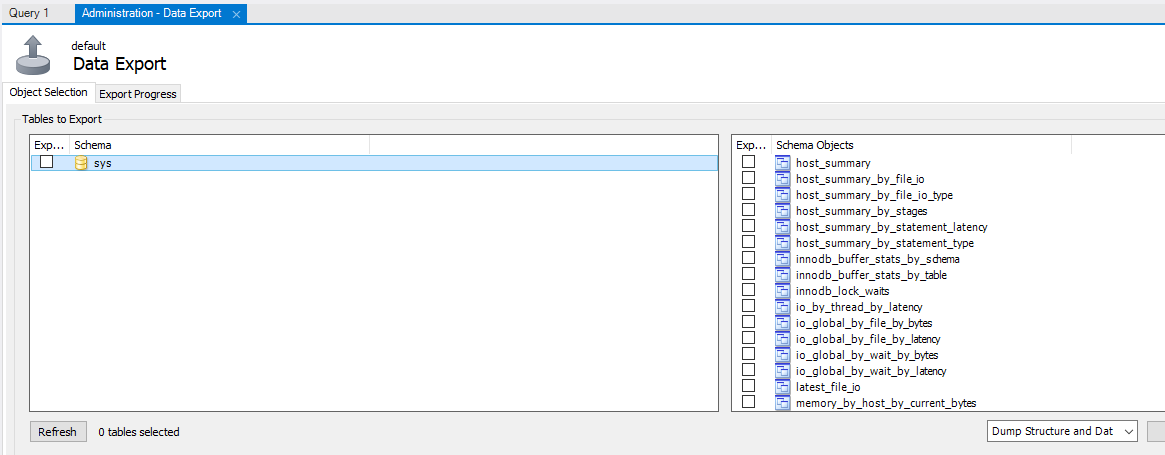
На рисунке указана вкладка “Status variables”(“Переменные состояния”). Каждая переменная имеет значение, если применимо, и описание.

На следующем рисунке показана вложенная вкладка “System Variables”(“Системные переменные”) , выбранная со всеми глобальными системными переменными для активного сервера, перечисленными по имени.



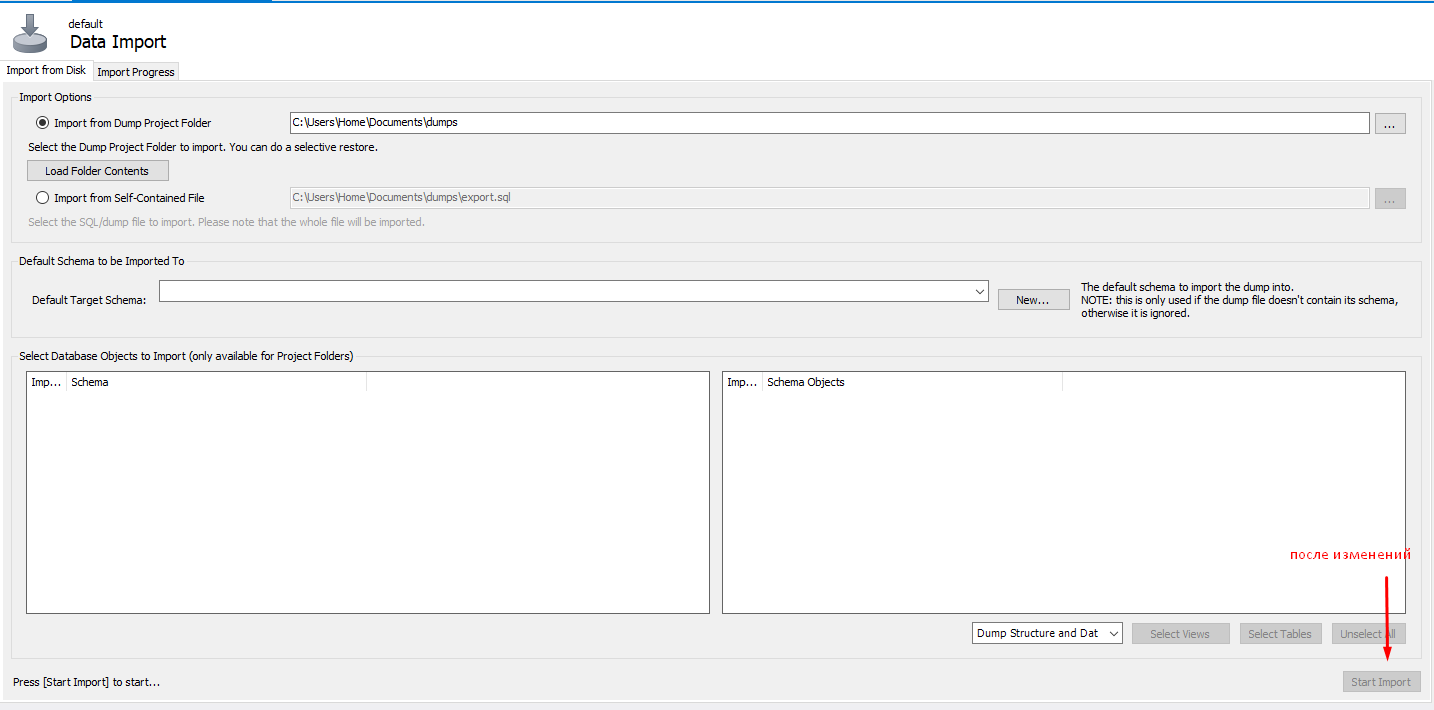
5) “Data Export” или “Экспорт данных”.

Эта вкладка позволяет экспортировать данные MySQL.



Надо выбрать схему, которую нужно экспортировать(при необходимости и другие объекты, например, объекты/таблицы схемы из каждой схемы).

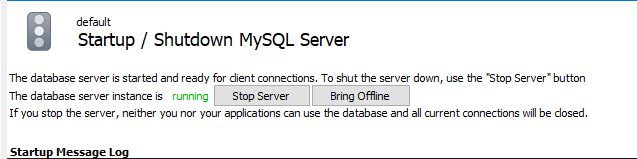
6) “Data Import/Restore” или “Импорт/Восстановление данных”  
  
Data Import/Restore (Импорт/восстановление данных): В этой вкладке можно выполнить операции импорта данных в базу данных или восстановления резервной копии. Здесь вы можете выбрать источник данных (например, файл CSV, SQL-скрипт или дамп базы данных), настроить параметры импорта и выполнить процесс импорта или восстановления.



2) Панель “Instance”.

Instance (Экземпляр):

В этой вкладке отображаются информация и настройки, связанные с текущим экземпляром сервера базы данных. Вы можете видеть основные сведения о сервере, такие как версия, хост и порт, а также настроить параметры сервера, такие как размер буферов и кэшей.  
Содержание панели:   
  
 1) “Startup/Shutdown” или “Запуск/Остановка”

В этой подкатегории вкладки "Instance" вы можете управлять процессами запуска и остановки сервера базы данных. Здесь вы можете выбрать опции запуска, указать параметры командной строки или переменные среды, а также выполнить запуск или остановку сервера.  


2) “Server Logs” или “Логи сервера”

В этой подкатегории вы можете просматривать различные логи, связанные с работой сервера базы данных. Это может включать журнал ошибок (error log), содержащий информацию об ошибках, предупреждениях и других событиях, произошедших на сервере. Здесь вы также можете анализировать логи запросов (query log) или логи обмена данными с клиентами (general log).

3) “Options File” или “Файл настроек”

В этой подкатегории вы можете просматривать и редактировать файл настроек сервера базы данных, такой как my.cnf или my.ini. Файл настроек содержит параметры и конфигурационные опции, которые влияют на работу сервера. Здесь вы можете настраивать различные параметры, такие как размеры буферов, настройки безопасности, опции сети и другие.

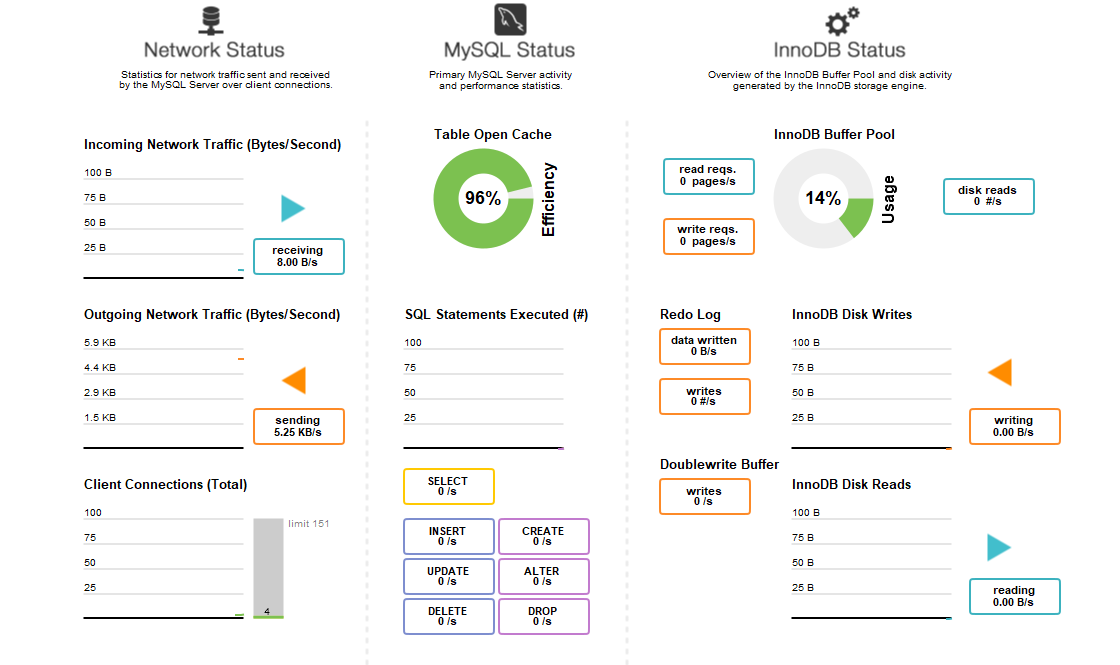
3) Панель “Perfomance”

Эта вкладка предоставляет инструменты для анализа и оптимизации производительности сервера базы данных.

Содержание панели:

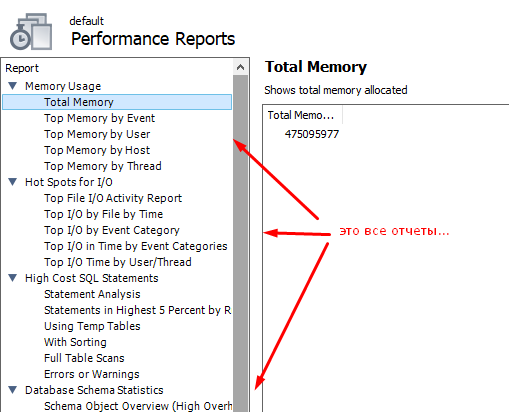
1) “Dashboard” или “Панель мониторинга”

Здесь отображается общая информация о производительности сервера базы данных, такая как нагрузка на CPU, память, количество активных соединений и т.д. Вы можете получить обзорную информацию о состоянии сервера и его ресурсах.



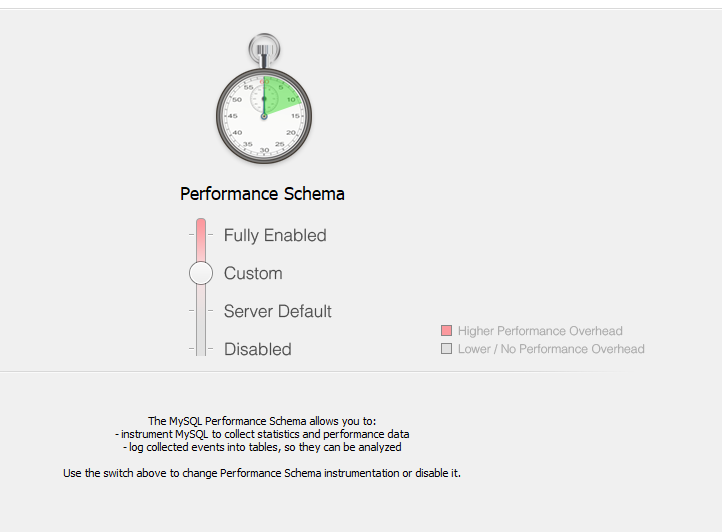
2) “Performance Reports” или “Отчеты о производительности”

В этой подвкладке вы можете собирать и анализировать данные о производительности сервера базы данных. Здесь предоставляются различные отчеты, такие как отчеты о нагрузке, выполнении запросов, использовании ресурсов и другие, которые помогут вам оптимизировать производительность базы данных.



3) “Performance Schema Setup” “Настройка схемы производительности”

Здесь вы можете настраивать Performance Schema, который является инструментом для сбора детальной информации о производительности сервера. Вы можете выбирать, какие типы событий и данные собирать, настраивать параметры и просматривать результаты анализа.



**Задание 2.** Создать и настроить новую базу данных simpledb.

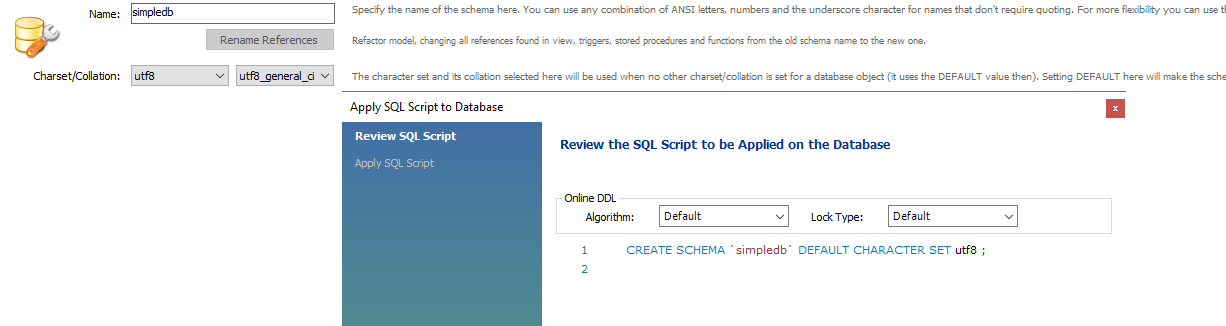
Для этого необходимо в верхнем меню нажать кнопку “Создание новой базы данных в рамках данного подключения к серверу” (4), находящуюся в панели главного окна с настройками показанными на изображении ниже:

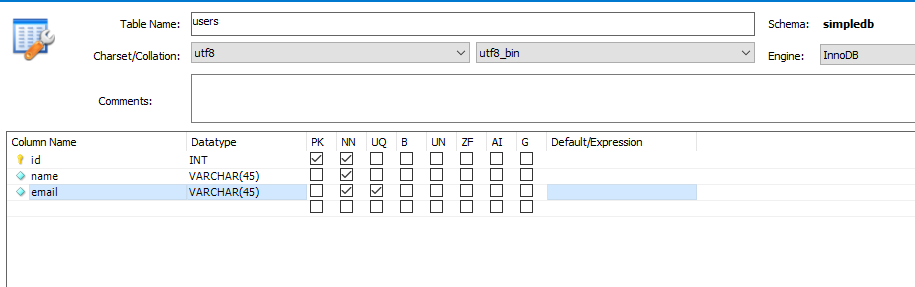
Schema Name: simpledb

Character Set: utf8

Collation: utf8\_general\_ci

Выполнение:





**Задание 3.** Скопируйте запрос, соответствующий созданию этой таблицы и вставьте его в отчет по выполнению этой лабораторной работы.

Если таблица уже создана, то вы можете щелкнуть правой кнопкой мыши по названию таблицы и выбрать пункт меню Copy to Clipboard - Create Statement.

Выполнение:

CREATE TABLE `simpledb`.`users` (

`id` INT NOT NULL,

`name` VARCHAR(45) NOT NULL,

`email` VARCHAR(45) NOT NULL,

PRIMARY KEY (`id`),

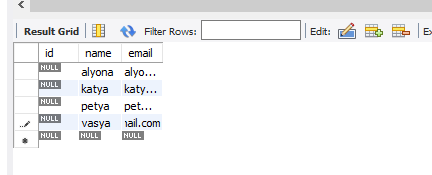
UNIQUE INDEX `email\_UNIQUE` (`email` ASC) VISIBLE)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

**Задание 4.** Добавьте несколько примеров-записей в созданную таблицу.

Выполнение:



Созданный SQL-запрос:

INSERT INTO `simpledb`.`users` (`id`, `name`, `email`) VALUES ('1', 'alyona', 'alyona@gmail.com');

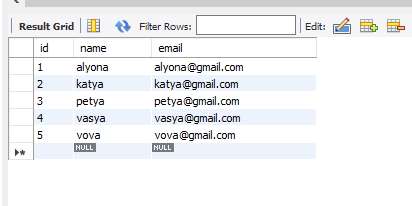
INSERT INTO `simpledb`.`users` (`id`, `name`, `email`) VALUES ('2', 'katya', 'katya@gmail.com');

INSERT INTO `simpledb`.`users` (`id`, `name`, `email`) VALUES ('3', 'petya', 'petya@gmail.com');

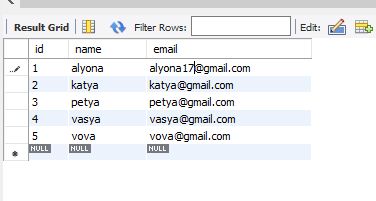
INSERT INTO `simpledb`.`users` (`id`, `name`, `email`) VALUES ('4', 'vasya', 'vasya@gmail.com');

Добавление данных:

INSERT INTO `simpledb`.`users` (`id`, `name`, `email`) VALUES ('5', 'vova', 'vova@gmail.com');



Изменение существующих данных:



SQL-запрос:

UPDATE `simpledb`.`users` SET `email` = 'alyona17@gmail.com' WHERE (`id` = '1');

**Задание 5.** Дополните таблицу users так, чтобы получилась таблица со следующими полями и параметрами:

1. id int pk, not null

2. name varchar(50)

3. email varchar(45)

4. gender ENUM(‘M’, ‘F’)

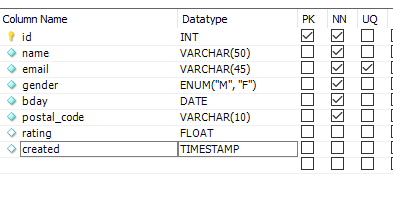
5. bday Date

6. postal\_code varchar(10)

7. rating float

8. created TIMESTAMP CURRENT\_TIMESTAMP()

Выполнение:

****

ALTER TABLE `simpledb`.`users`

ADD COLUMN `gender` ENUM("M", "F") NOT NULL AFTER `email`,

ADD COLUMN `bday` DATETIME NULL AFTER `gender`,

ADD COLUMN `postal\_code` VARCHAR(10) NOT NULL AFTER `bday`,

ADD COLUMN `rating` FLOAT NULL AFTER `postal\_code`,

ADD COLUMN `created` TIMESTAMP NULL AFTER `rating`,

CHANGE COLUMN `name` `name` VARCHAR(50) NOT NULL ;

*Замечание 2*

Тип данных у поля created - TIMESTAMP, значение по умолчанию CURRENT\_TIMESTAMP().

В отчете по ЛР опишите, что это означает, прочитав в документации или экспериментально определив.

CURRENT\_TIMESTAMP(). Вот что это означает:

1. Тип данных TIMESTAMP:

Тип данных TIMESTAMP в базе данных MySQL используется для хранения даты и времени. Он представляет собой комбинацию даты и времени в формате "ГГГГ-ММ-ДД ЧЧ:ММ:СС". В данном случае, поле "created" будет хранить информацию о дате и времени создания записи.

1. Значение по умолчанию CURRENT\_TIMESTAMP():

Значение по умолчанию CURRENT\_TIMESTAMP() означает, что если при вставке новой записи в таблицу не указано значение для поля "created", то автоматически будет использовано текущее значение даты и времени на момент вставки. То есть, при создании новой записи в поле "created" будет автоматически записана текущая дата и время.

Таким образом, при каждой вставке новой записи в таблицу, поле "created" будет автоматически заполняться текущим значением даты и времени, если не указано другое значение. Это позволяет автоматизировать процесс сохранения информации о времени создания записей и упрощает работу с данными в базе данных.

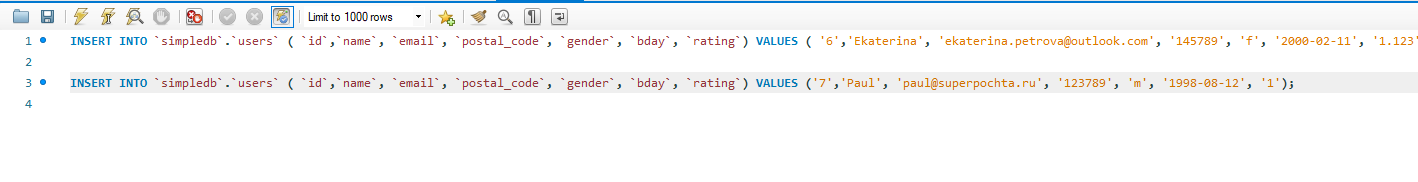
**Задание 6.** Дополните таблицу, добавив данные двумя способами:

* с помощью внесения данных вручную (как это было сделано ранее);
* с помощью выполнения SQL-запросов ниже;

INSERT INTO `simpledb`.`users` (`name`, `email`, `postal\_code`, `gender`, `bday`, `rating`) VALUES ('Ekaterina', 'ekaterina.petrova@outlook.com', '145789', 'f', '2000-02-11', '1.123');

INSERT INTO `simpledb`.`users` (`name`, `email`, `postal\_code`, `gender`, `bday`, `rating`) VALUES ('Paul', 'paul@superpochta.ru', '123789', 'm', '1998-08-12', '1');

Выполнение:





**Задание 7.** С помощью кнопки “Export recordset to external file” и получите файл с SQL-запросами (он будет экспортирован в формате .sql). Сохраните файл с SQL-запросами (в нем должно происходить добавление в таблицу данных), проанализируйте синтаксис этих запросов и вставьте их в отчет.

Выполнение:

INSERT INTO `simpledb`.`users` ( `id`,`name`, `email`, `postal\_code`, `gender`, `bday`, `rating`) VALUES ( '6','Ekaterina', 'ekaterina.petrova@outlook.com', '145789', 'f', '2000-02-11', '1.123');

INSERT INTO `simpledb`.`users` ( `id`,`name`, `email`, `postal\_code`, `gender`, `bday`, `rating`) VALUES ('7','Paul', 'paul@superpochta.ru', '123789', 'm', '1998-08-12', '1');

SELECT \* FROM `simpledb`.`users`;

**Задание 8.** Создайте еще одну таблицу с названием resume со следующей структурой:

resumeid, INT, PK, NN, AI

userid, INT, NN

title, VARCHAR(100), NN

skills, TEXT

created, TIMESTAMP, Default / Expression: CURRENT\_TIMESTAMP()

При конструировании внизу во вкладке Foreign Keys определите так называемый внешний ключ (foreign key), который будет определять связь между текущей таблицей resume и уже созданной таблицей user.

Выполнение:

CREATE TABLE `simpledb`.`resume` (

`resumeid` INT NOT NULL AUTO\_INCREMENT,

`userid` INT NOT NULL,

`title` VARCHAR(100) NOT NULL,

`skills` TEXT NULL,

`created` TIMESTAMP NULL,

PRIMARY KEY (`resumeid`),

INDEX `userid\_idx` (`userid` ASC) VISIBLE,

CONSTRAINT `userid`

FOREIGN KEY (`userid`)

REFERENCES `simpledb`.`users` (`id`)

ON DELETE CASCADE

ON UPDATE CASCADE)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

**Опишите как будет вести себя СУБД при удалении связанных записей из таблиц users и resume.**

Выполнение:

"ON UPDATE CASCADE" и "ON DELETE CASCADE" - это фразы, используемые при определении внешних ключей (foreign keys) в базе данных. Они определяют, какие действия будут выполнены в связанных таблицах при обновлении или удалении родительской записи.

ON UPDATE CASCADE (При обновлении каскадом):

Когда в родительской таблице обновляется значение первичного ключа, и если установлено условие "ON UPDATE CASCADE" для связанного внешнего ключа в дочерней таблице, то обновление будет применено не только к родительской записи, но и ко всем связанным дочерним записям. Другими словами, изменение значения первичного ключа в родительской таблице автоматически обновит соответствующие значения в дочерних таблицах.

ON DELETE CASCADE (При удалении каскадом):

Когда в родительской таблице происходит удаление записи, и если установлено условие "ON DELETE CASCADE" для связанного внешнего ключа в дочерней таблице, то удаление будет применено не только к родительской записи, но и ко всем связанным дочерним записям. Другими словами, удаление записи из родительской таблицы автоматически удалит все связанные записи в дочерних таблицах.

**Задание 9.** Наполните вторую таблицу данными так, чтобы в ней была информация хотя бы о нескольких резюме, связанных с уже существующими пользователями из таблицы users. Подумайте и напишите в отчете, сколько резюме может быть у одного пользователя (минимум и максимум)?

Выполнение:

INSERT INTO `simpledb`.`resume` (`userid`, `title`, `skills`, `created`)

VALUES

(1, 'Software Developer', 'Java, Python, SQL', CURRENT\_TIMESTAMP),

(2, 'Graphic Designer', 'Adobe Photoshop, Illustrator', CURRENT\_TIMESTAMP),

(3, 'Marketing Specialist', 'Digital marketing, SEO', CURRENT\_TIMESTAMP);

Минимальное количество резюме: 0. Возможно, пользователь не создает ни одного резюме, и в таблице resume не будет записей, связанных с данным пользователем.

Максимальное количество резюме: Неограниченно. Пользователь может создать сколько угодно резюме, и в таблице resume будет храниться соответствующее количество записей, связанных с данным пользователем.