**Лабораторная работа 1. Начало работы с MySQL. MySQL Workbench**

**Задание 1.**

*Раздел “Management”:*

1. Подраздел “Server Status”. В подразделе отображается общая информация о сервере и подключении к нему. Информация логически сгруппирована. Можно выделить следующие группы:
   * В шапке окна расположена общая информация о версии программы и о самом используемом устройстве: порт(3306),

Сокет БД(MySQL), версия сервера(8.0.23).

* + Доступные особенности сервера(включают в себя доступность

SSL, валидация пароля и т.д.).

* + Каталоги сервера.
  + Сводка по используемым ресурсам компьютера
  + Информация по Master-Slave репликации.
  + Настройки соединения SSL

1. Подраздел “Client Connections”. В подразделе отображается информация по соединениям между активной базой данных и ее клиентами.
2. Подраздел “Users and Privileges”. В подразделе описываются и редактируются права для разных пользователей БД.
3. Подраздел “Server Variables”. В подразделе расположена информация по состоянию серверных переменных и параметров.
4. Подраздел “Data Export”. В этом подразделе можно управлять экспортом различных данных из БД.
5. Подраздел “Data Import/Restore”. В этом подразделе можно управлять импортом и восстановлением различных данных из БД.

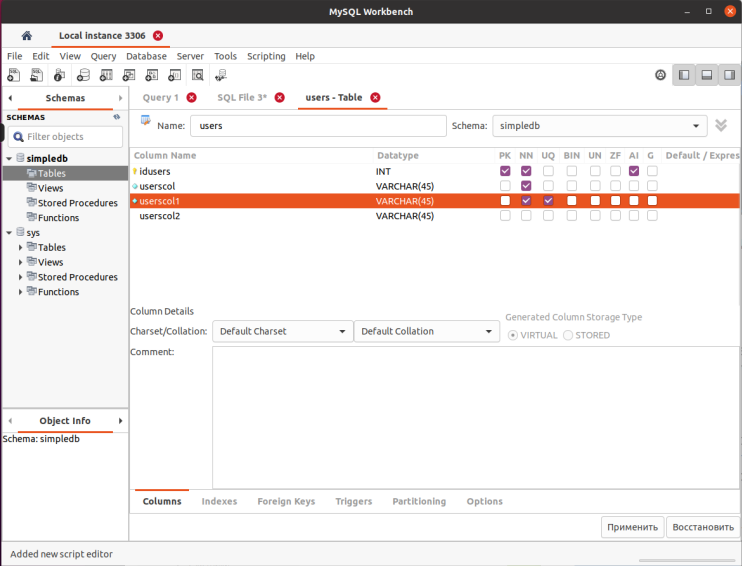
*Раздел “Instance”:*

1. Подраздел “Startup/Shutdown MySQL”. Здесь можно регулировать настройки Запуска/Остановки MySQL сервера а также просматривать лог запуска сервера.
2. Подраздел “Server Logs”. Здесь содержится лог завершенных процессов, ошибок и предупреждений сервера.
3. Подраздел “Options File”. Общий подраздел настроек(баз данных, рабочей среды Workbench, логирования и т.д.).

*Раздел “Performance”:*

1. Подраздел “Dashboard”.Это панель различных параметров MySQL сервера и базы данных, которые можно отслеживать в реальном времени.
2. Подраздел “Performance Reports”. Здесь можно запросить отчеты по быстродействию БД и сервера.
3. Подраздел “Performance Schema Setup”. Подраздел, в котором можно регулировать сбор информации о производительности БД.

**Задание 2**

****

**Задание 3**

CREATE TABLE `users` (

  `id` int NOT NULL,

  `name` varchar(45) NOT NULL,

  `email` varchar(45) NOT NULL,

  PRIMARY KEY (`id`),

  UNIQUE KEY `email\_UNIQUE` (`email`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

**Задание 4**

INSERT INTO `simpledb`.`users` (`id`, `name`, `email`) VALUES ('1', 'Igor', '[Tim@gmail.com](mailto:Tim@gmail.com)');  
INSERT INTO `simpledb`.`users` (`id`, `name`, `email`) VALUES ('2', 'Vadim', '[v4dikk@yandex.com](mailto:v4dikk@yandex.com)');  
INSERT INTO `simpledb`.`users` (`id`, `name`, `email`) VALUES ('3', 'Ivan', '[sharingan\_ivan@gmail.com](mailto:sharingan_ivan@gmail.com)');

Редактирование:

UPDATE `simpledb`.`users` SET `email` = '[sharingan@gmail.com](mailto:sharingan@gmail.com)' WHERE (`id` = '3');  
UPDATE `simpledb`.`users` SET `email` = '[v4dik@yandex.com](mailto:v4dik@yandex.com)' WHERE (`id` = '2');  
UPDATE `simpledb`.`users` SET `email` = '[Tim\_3\_@gmail.com](mailto:Tim_3_@gmail.com)' WHERE (`id` = '1');

**Задание 5**

ALTER TABLE `simpledb`.`users`

ADD COLUMN `gender` ENUM("M", "F") NULL DEFAULT NULL AFTER `email`,

ADD COLUMN `bday` DATE NULL DEFAULT '1970-01-01' AFTER `gender`,

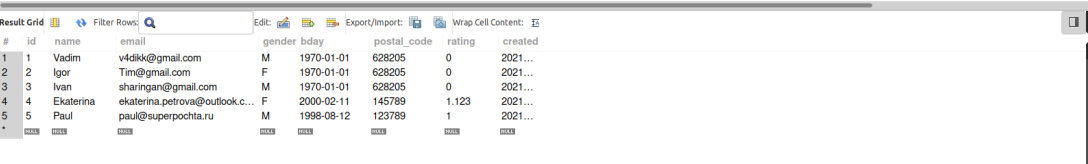
ADD COLUMN `postal\_code` VARCHAR(10) NULL DEFAULT NULL AFTER `bday`,

ADD COLUMN `rating` FLOAT NOT NULL DEFAULT 0.0 AFTER `postal\_code`,

ADD COLUMN `created` TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP AFTER `rating`;

Функция ***CURRENT\_TIMESTAMP*** вернет текущую дату в формате ‘YYYY-MM-DD HH:MM:SS’

**Задание 6**

****

**Задание 7**

/\*  
— Query: SELECT \* FROM simpledb.users  
LIMIT 0, 1000  
  
— Date: 2021-02-17 23:14  
\*/  
INSERT INTO `` (`id`,`name`,`email`,`gender`,`bday`,`postal\_code`,`rating`,`created`) VALUES (1,'Vadim','[v4dikk@gmail.com](mailto:v4dikk@gmail.com)','M','1970-01-01','628205',0,'2021-02-17 22:40:02');  
INSERT INTO `` (`id`,`name`,`email`,`gender`,`bday`,`postal\_code`,`rating`,`created`) VALUES (2,'Igor','[Tim@gmail.com](mailto:Tim@gmail.com)','F','1970-01-01','628205',0,'2021-02-17 22:40:02');  
INSERT INTO `` (`id`,`name`,`email`,`gender`,`bday`,`postal\_code`,`rating`,`created`) VALUES (3,'Ivan','[sharingan@gmail.com](mailto:sharingan@gmail.com)','M','1970-01-01','628205',0,'2021-02-17 22:40:02');  
INSERT INTO `` (`id`,`name`,`email`,`gender`,`bday`,`postal\_code`,`rating`,`created`) VALUES (4,'Ekaterina','[ekaterina.petrova@outlook.com](mailto:ekaterina.petrova@outlook.com)','F','2000-02-11','145789',1.123,'2021-02-17 22:55:44');  
INSERT INTO `` (`id`,`name`,`email`,`gender`,`bday`,`postal\_code`,`rating`,`created`) VALUES (5,'Paul','[paul@superpochta.ru](mailto:paul@superpochta.ru)','M','1998-08-12','123789',1,'2021-02-17 22:55:44');

**Задание 8**

CREATE TABLE `simpledb`.`resume` (

  `idresume` INT NOT NULL AUTO\_INCREMENT,

  `userid` INT NOT NULL,

  `title` VARCHAR(100) NOT NULL,

  `skills` TEXT NULL,

  `created` TIMESTAMP NULL DEFAULT CURRENT\_TIMESTAMP(),

  PRIMARY KEY (`idresume`),

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

  CONSTRAINT `userid`

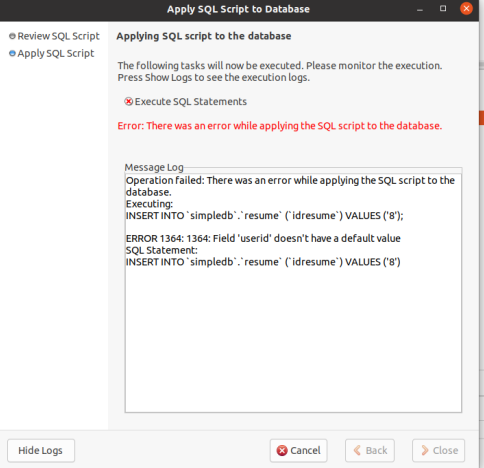
FOREIGN KEY (`userid`)

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

ON DELETE CASCADE

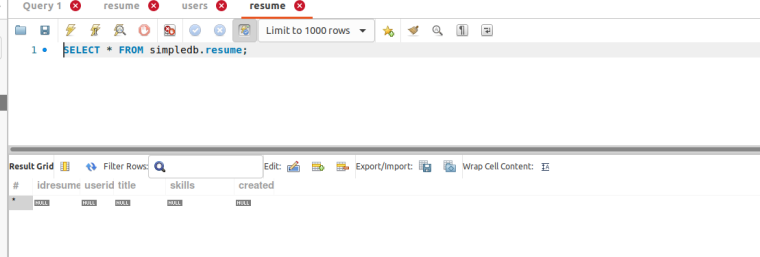
ON UPDATE CASCADE);

**Задание 9**

****

/\*  
— Query: SELECT \* FROM simpledb.resume  
LIMIT 0, 1000  
  
— Date: 2021-02-17 23:42  
\*/  
INSERT INTO `` (`idresume`,`userid`,`title`,`skills`,`created`) VALUES (1,1,'Resume','Java','2021-02-17 23:41:53');

**Задание 10**

****