**Задание 1**

**Раздел “Management”**

1. **Раздел “Server Status”:** В разделе отображается общая информация о сервере и подключении к нему. Включает в себя просмотр рабочего состояния сервера (остановлен / работает), доступных функций, основных каталогов сервера, состояния репликации и настроек безопасности для аутентификации и SSL. Отчеты также включают информацию и графики для отслеживания использования памяти, подключений, показателей успешности
2. **Раздел “Client Connection”:** Браузер клиентских подключений отображает список активных и спящих клиентских подключений MySQL, а также добавляет возможность уничтожать операторы и подключения, а также просматривать дополнительные сведения и атрибуты подключения.
3. **Раздел “Users and Privileges”:** вкладка содержит список всех пользователей и привилегий, которые относятся к экземпляру активного MySQL сервера. На этой вкладке вы можете добавлять учетные записи пользователей и управлять ими, настраивать права и срок действия паролей.
4. **Раздел “Status and System Variables”:** Представлен полный набор серверных переменных для активного соединения MySQL. Вы также можете скопировать все или выбранные переменные в буфер обмена.
5. **Раздел “Data Export”:** Эта вкладка позволяет экспортировать данные MySQL. Выберите каждую схему, которую вы хотите экспортировать, при необходимости выберите конкретные объекты / таблицы схемы из каждой схемы и сгенерируйте экспорт. Опции конфигурации включают экспорт в папку проекта или автономный файл SQL, опциональный дамп сохраненных подпрограмм и событий или пропуск данных таблицы.
6. **Раздел “Data Import/Restore”:** Вкладка позволяет выбрать схему, в которую вы хотите импортировать данные.

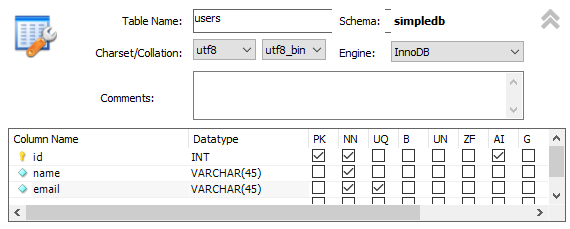
**Раздел “Instance”**

1. **Раздел “Startup/Shutdown”:** вкладка позволяет выполнять следующие действия сервиса управления: просмотрите журнал сообщений запуска, запустите и выключите экземпляр MySQL, просмотрите текущий статус экземпляра MySQL.
2. **Раздел “Server Logs”:** Отображается информация журнала для сервера MySQL, представленная каждой вкладкой подключения. Для каждого соединения вкладка включает дополнительные вкладки для общих журналов ошибок и журналов медленных логов (если они есть).
3. **Раздел “Options File”:** Редактор используется для просмотра и редактирования файлов конфигурации MySQL (my.ini на Windows, или my.cnf на Linux и MacOS) с помощью флажков и других элементов управления с графическим интерфейсом, а затем сделать изменения. MySQL Workbench делит файл параметров на отдельные группы в виде набора вкладок.

**Раздел “Performance”**

1. **Раздел “Dashboard”:** Просматривайте статистику производительности сервера на графической панели управления. Чтобы отобразить панель мониторинга, откройте вкладку запроса и затем щелкните Панель мониторинга в области производительности боковой панели навигатора с выбранной вкладкой «Управление». На следующем рисунке показано расположение информации на вкладке «Администрирование» - «Панель мониторинга».
2. **Раздел “Performance Reports”:** Отчеты на основе схемы производительности обеспечивают понимание операций сервера MySQL с помощью полезных отчетов высокого уровня. MySQL Workbench использует представления SYS в схеме производительности для создания более 20 отчетов, помогающих анализировать производительность ваших баз данных MySQL. Отчеты помогают анализировать точки доступа ввода-вывода, обнаруживать дорогостоящие операторы SQL и просматривать статистику ожидания и показатели механизма InnoDB.
3. **Раздел “Performance Schema Setup”:** В разделе можно удалить все данные схемы.

**Задание 2**



**Задание 3**

CREATE TABLE `users` (

`id` int NOT NULL AUTO\_INCREMENT,

`name` varchar(45) NOT NULL,

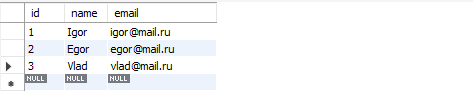
`email` varchar(45) NOT NULL,

PRIMARY KEY (`id`),

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

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

**Задание 4**



**SQL запросы при добавлении 3 строк с данными:**

INSERT INTO `simpledb`.`users` (`name`, `email`) VALUES ('Igor', 'igor@mail.ru');

INSERT INTO `simpledb`.`users` (`name`, `email`) VALUES ('Egor', 'egor@mail.ru');

INSERT INTO `simpledb`.`users` (`name`, `email`) VALUES ('Vlad', 'vlad@mail.ru');

**Задание 5**

Timestamp относится к временному типу данных.



**SQL запросы при добавлении новых столбцов:**

ALTER TABLE `simpledb`.`users`

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

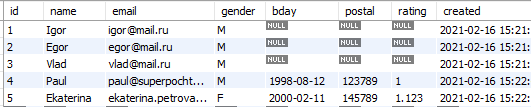
ADD COLUMN `bday` DATE NULL AFTER `gender`,

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

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

ADD COLUMN `created` TIMESTAMP NULL DEFAULT CURRENT\_TIMESTAMP() AFTER `rating`;

**Задание 6**



**Задание 7**

/\*

-- Query: SELECT \* FROM simpledb.users

LIMIT 0, 1000

-- Date: 2021-02-16 15:24

\*/

INSERT INTO `` (`id`,`name`,`email`,`gender`,`bday`,`postal`,`rating`,`created`) VALUES (1,'Igor','igor@mail.ru','M',NULL,NULL,NULL,'2021-02-16 15:21:35');

INSERT INTO `` (`id`,`name`,`email`,`gender`,`bday`,`postal`,`rating`,`created`) VALUES (2,'Egor','egor@mail.ru','M',NULL,NULL,NULL,'2021-02-16 15:21:35');

INSERT INTO `` (`id`,`name`,`email`,`gender`,`bday`,`postal`,`rating`,`created`) VALUES (3,'Vlad','vlad@mail.ru','M',NULL,NULL,NULL,'2021-02-16 15:21:35');

INSERT INTO `` (`id`,`name`,`email`,`gender`,`bday`,`postal`,`rating`,`created`) VALUES (4,'Paul','paul@superpochta.ru','M','1998-08-12','123789',1,'2021-02-16 15:22:41');

INSERT INTO `` (`id`,`name`,`email`,`gender`,`bday`,`postal`,`rating`,`created`) VALUES (5,'Ekaterina','ekaterina.petrova@outlook.com','F','2000-02-11','145789',1.123,'2021-02-16 15:22:56');

**Задание 8**

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 DEFAULT CURRENT\_TIMESTAMP(),

PRIMARY KEY (`resumeid`),

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

CONSTRAINT `userid`

FOREIGN KEY (`userid`)

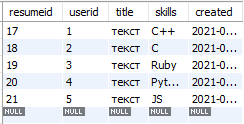
REFERENCES `simpledb`.`users` (`id`)

ON DELETE CASCADE

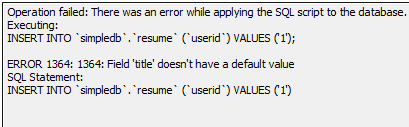
ON UPDATE CASCADE);

Так как resume связана с users по id, то при удалении строк из users в resume не будет появляться строк для данных того или иного пользователя.

**Задание 9**



У пользователя может быть максимум одно резюме, а может и не быть



**SQL-запросы:**

/\*

-- Query: SELECT \* FROM simpledb.resume

LIMIT 0, 1000

-- Date: 2021-02-16 15:45

\*/

INSERT INTO `` (`resumeid`,`userid`,`title`,`skills`,`created`) VALUES (17,1,'текст','C++','2021-02-16 15:42:02');

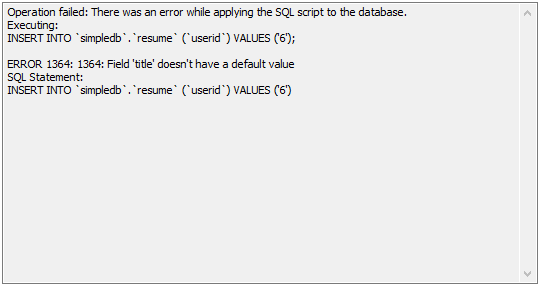
INSERT INTO `` (`resumeid`,`userid`,`title`,`skills`,`created`) VALUES (18,2,'текст','C','2021-02-16 15:42:02');

INSERT INTO `` (`resumeid`,`userid`,`title`,`skills`,`created`) VALUES (19,3,'текст','Ruby','2021-02-16 15:42:02');

INSERT INTO `` (`resumeid`,`userid`,`title`,`skills`,`created`) VALUES (20,4,'текст','Python','2021-02-16 15:42:02');

INSERT INTO `` (`resumeid`,`userid`,`title`,`skills`,`created`) VALUES (21,5,'текст','JS','2021-02-16 15:42:02');

При добавлении несуществующего id выдает ошибку о том, что нет таких значений



**Задание 10**

DELETE FROM `simpledb`.`users` WHERE (`id` = '5');

При удалении пользователя, в таблице resume автоматически удаляются все данные связанные с номером id, который удалили.

При изменении id, он также изменяется в таблице resume.

