

Задание 1.

Раздел "Management":

1. Server Status:

- Отображает общую информацию о сервере и его текущем состоянии.
- Группирует информацию, такую как название хоста, номер порта, статус подключений, версия сервера и другие системные переменные.

2. Client Connections:

- Предоставляет информацию о текущих подключениях клиентов к серверу базы данных.
- Включает в себя сведения о соединениях, их статусе, адресах клиентов и другие параметры.

3. Users and Privileges:

- Позволяет управлять пользователями, их правами доступа и привилегиями.
- Включает функции создания, редактирования и удаления пользователей, а также настройки их привилегий.

4. Status and System Variables:

- Отображает текущий статус и значения системных переменных сервера базы данных.
- Включает в себя информацию о нагрузке, использовании ресурсов и других параметрах конфигурации.

5. Data Export:

- Предоставляет инструменты для экспорта данных из базы данных в различные форматы файлов.
- Позволяет настроить параметры экспорта, такие как формат файла, типы данных и т. д.

6. Data Import/Restore:

- Позволяет импортировать данные в базу данных из внешних источников или восстановить данные из резервных копий.

- Включает в себя функции настройки параметров импорта и восстановления.

Раздел "Instance":

1. Startup/Shutdown:

- Предоставляет средства для запуска и остановки конкретного экземпляра базы данных.
- Включает в себя параметры запуска и контроль статуса экземпляра.

2. Server Logs:

- Позволяет просматривать журналы событий и ошибок, связанных с конкретным экземпляром базы данных.

3. Options File:

- Предоставляет доступ и возможность редактирования файла настроек (например, my.cnf) для более тонкой настройки параметров экземпляра.

Раздел "Performance":

1. Dashboard:

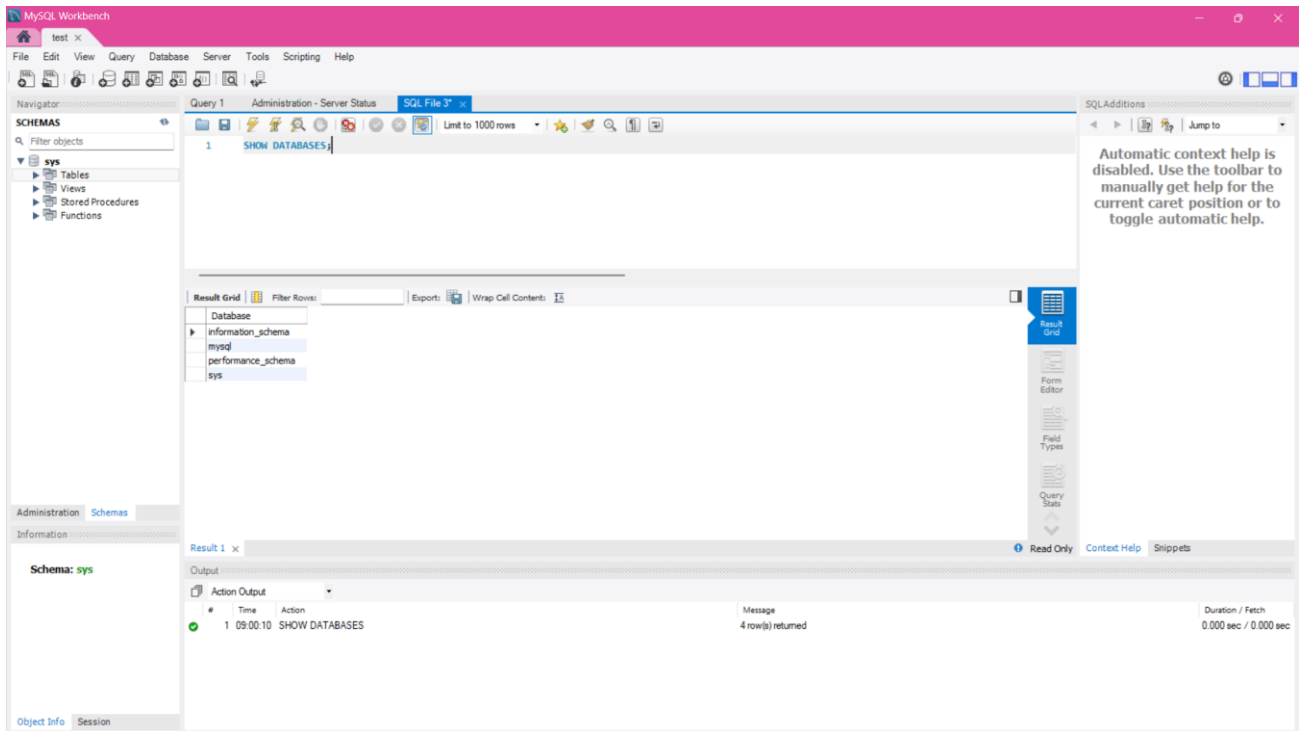
- Включает графики и диаграммы, предоставляя обзор общей производительности сервера и экземпляра базы данных.

2. Performance Reports:

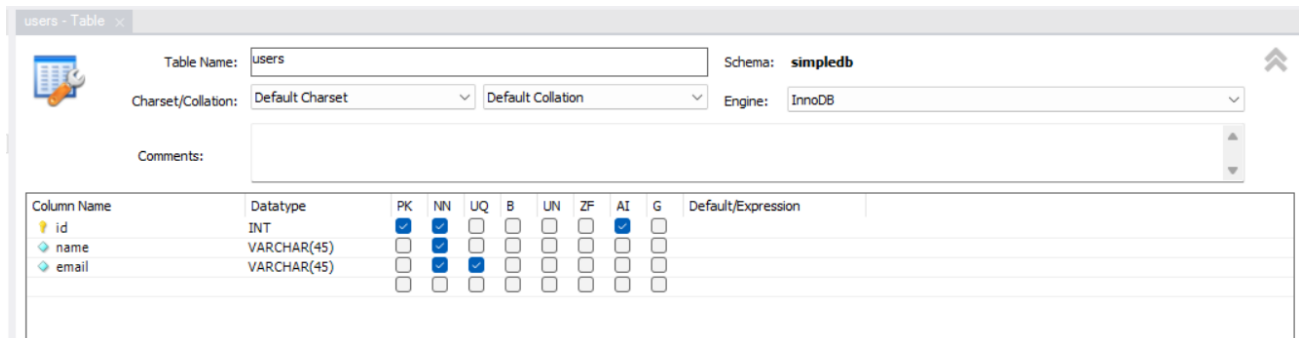
- Предоставляет детальные отчеты о производительности базы данных, включая статистику по запросам и использованию ресурсов.

3. Performance Schema Setup:

- Позволяет настраивать схему производительности, которая предоставляет дополнительную информацию о производительности и использовании ресурсов.



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=utf8mb3;
```

Задание 4.

Созданная таблица:

	id ▲	name	email
	1	Pedro	Pedro@gmail.com
	2	Padro	Padro@gmail.com
	3	Pudro	Pudro@gmail.com

```
INSERT INTO simplifiedb.new_table (`id`, name, `email`) VALUES ('1', Pedro, 'Pedro@gmail.com');
```

```
INSERT INTO simplifiedb.new_table (`id`, name, `email`) VALUES ('2', Padro, 'Padro@gmail.com');
```

```
INSERT INTO simplifiedb.new_table (`id`, name, `email`) VALUES ('3', Pudro, 'Pudro@gmail.com');
```

Изменения:

```
UPDATE `simplifiedb`.`users` SET `name` = 'Pabro', `email` = 'Pabro@gmail.com' WHERE (`id` = '2');
```

Задание 5.

```
ALTER TABLE `simplifiedb`.`users`  
ADD COLUMN `gender` ENUM('M', 'F') NULL AFTER `email`,  
ADD COLUMN `bday` DATE NULL AFTER `gender`,  
ADD COLUMN `postal_code` VARCHAR(10) NULL AFTER `bday`,  
ADD COLUMN `rating` FLOAT NULL AFTER `postal_code`,  
ADD COLUMN `created` DATETIME NULL AFTER `rating`,  
CHANGE COLUMN `name` `name` VARCHAR(50) NOT NULL ;
```

Задание 6.

```
UPDATE `simplifiedb`.`users` SET `gender` = 'm', `bday` = '1643-01-01',  
`postal_code` = '1234567', `rating` = '1', `created` = '2024-02-14' WHERE (`id` = '1');
```

```
UPDATE `simplifiedb`.`users` SET `gender` = 'm', `bday` = '2023.12.31',  
`postal_code` = '2131221', `rating` = '1.21', `created` = '2024-02-15' WHERE (`id` =  
'2');
```

```
UPDATE `simplifiedb`.`users` SET `gender` = 'm', `bday` = '2045.01.02',  
`postal_code` = '1234121', `rating` = '1.213', `created` = '2024-02-16' WHERE (`id` =  
'3');
```

```
INSERT INTO `simplifiedb`.`users` (`name`, `email`, `gender`, `bday`,  
`postal_code`, `rating`, `created`) VALUES ('Ekaterina',  
'ekaterina.petrova@outlook.com', 'f', '2000-02-11', '145789', '1.123', '2024-02-20');
```

```
INSERT INTO `simplifiedb`.`users` (`name`, `email`, `gender`, `bday`,  
`postal_code`, `rating`, `created`) VALUES ('Paul', 'paul@superpochta.ru', 'm', '1998-  
08-12', '123789', '1', '2024-02-20');
```

Задание 7.

id,name,email,gender,bday,postal_code,rating,created

1,Pedro,Pedro@gmail.com,M,1643-01-01,1234567,1,"2024-02-14 00:00:00"

2,Pabro,Pabro@gmail.com,M,2023-12-31,2131221,1.21,"2024-02-15
00:00:00"

3,Pudro,Pudro@gmail.com,M,2045-01-02,1234121,1.213,"2024-02-16
00:00:00"

4,Ekaterina,ekaterina.petrova@outlook.com,F,2000-02-
11,145789,1.123,"2024-02-20 00:00:00"

5,Paul,paul@superpochta.ru,M,1998-08-12,123789,1,"2024-02-20 00:00:00"

Задание 8.

```
CREATE TABLE `simplifiedb`.`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 `simplifiedb`.`users` (`id`)  
ON DELETE CASCADE  
ON UPDATE CASCADE);
```

Задание 9.

resumeid,userid,title,skills,created

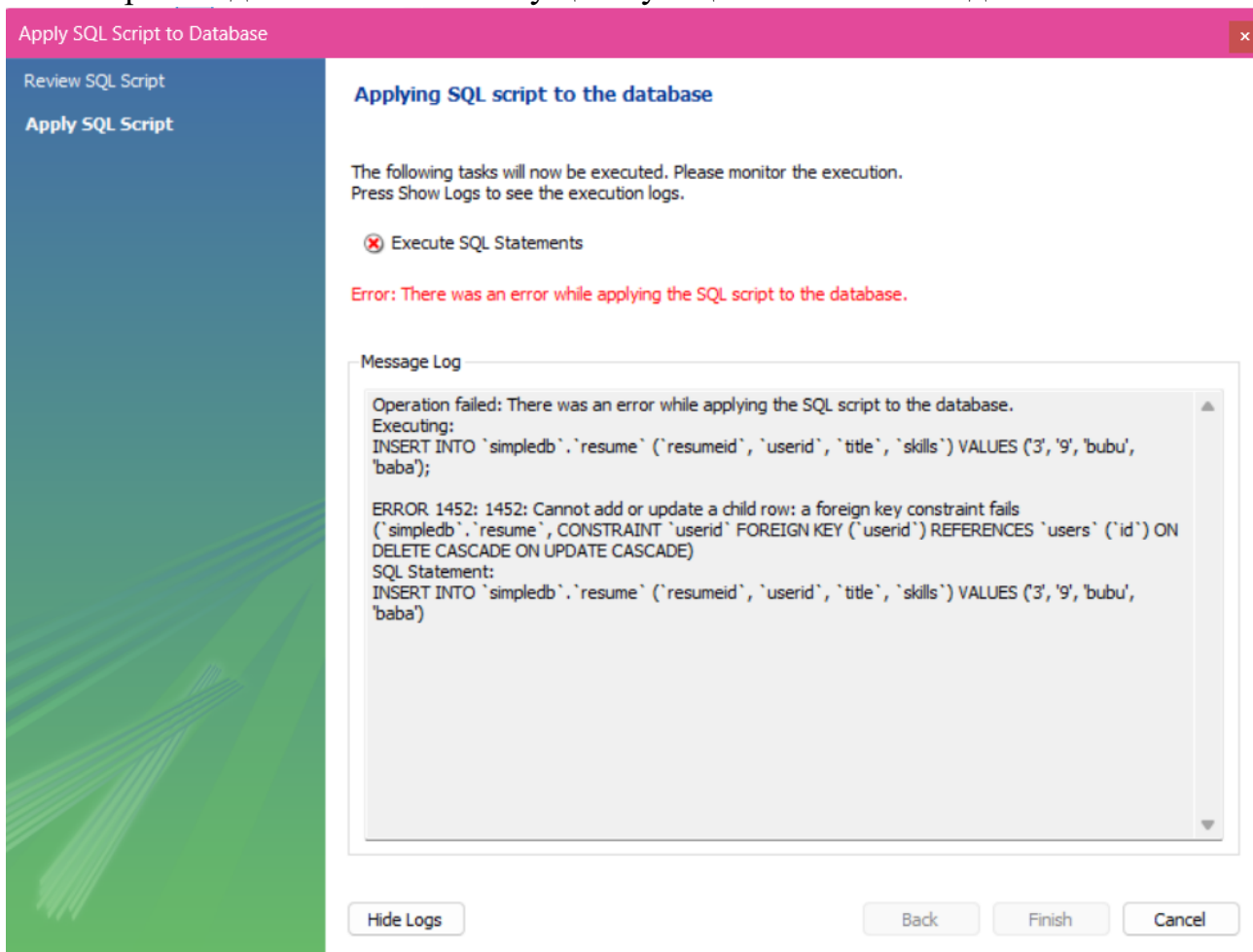
1,3,professor,"speed reading","2024-02-21 07:24:40"

2,4,builder,builder,"2024-02-21 07:24:40"

Минимум одно резюме.

Максимальное количество неограниченно.

При добавлении несуществующего id выдаётся ошибка:



Задание 10.

```
DELETE FROM `simplifiedb`.`users` WHERE (`id` = '3');
```

```
DELETE FROM `simplifiedb`.`users` WHERE (`id` = '4');
```

	resumeid	userid	title	skills	created
▶*	NULL	NULL	NULL		NULL

Если произойдет удаление пользователя, связанного с каким-либо резюме, то вместе с удалением пользователя также будут удалены все его резюме.

Также, если произойдет изменение идентификатора (id) пользователя, связанного с резюме, то вместе с изменением id пользователя также изменится соответствующий id в его резюме.