

# Лабораторная работа 1

## 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).

### Раздел «Instance»

- *Startup / shutdown* — позволяет включать и выключать сервер MySQL, просматривать в реальном времени логи загрузки и выключения.
- *Server logs* — просмотр обновляемого в режиме реального времени журнала сервера MySQL.
- *Options file* — графический редактор файла конфигурации сервера MySQL. Обладает возможностью управлять очень обширным набором параметров и опций.

### Раздел «Performance»

- *Dashboard* — наглядная схема (контрольная панель) для мониторинга состояния сервера MySQL.
- *Performance reports* — чрезвычайно подробные отчеты о производительности сервера, использовании памяти и других ресурсов теми или иными функциями.
- *Performance schema setup* — меню для настройки производительности сервера.

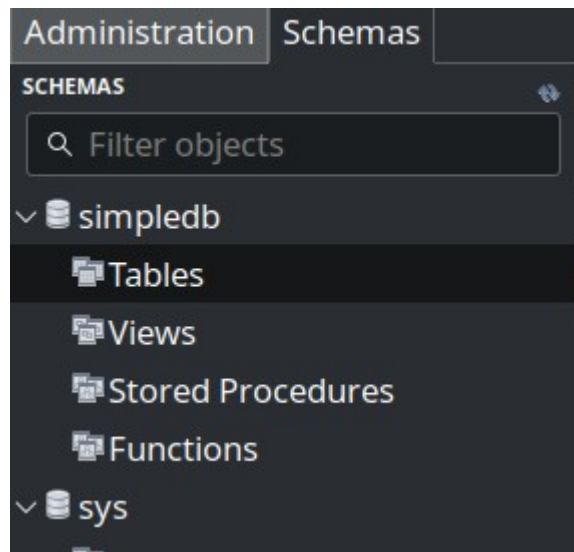
## Задание 2

Создать и настроить новую базу данных *simpledb*.

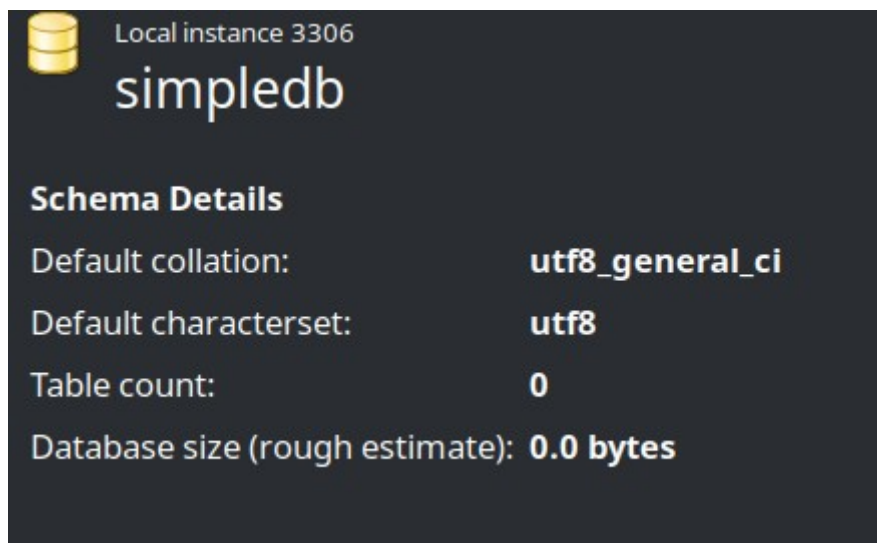
Для этого необходимо в верхнем меню нажать кнопку “Создание новой базы данных в рамках данного подключения к серверу” (4), находящуюся в панели главного окна с настройками показанными на изображении ниже:

- *Schema Name: simpledb*
- *Character Set: utf8*
- *Collation: utf8\_general\_ci*

После выполнения появилась новая схема:



Свойства схемы:



### 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*.

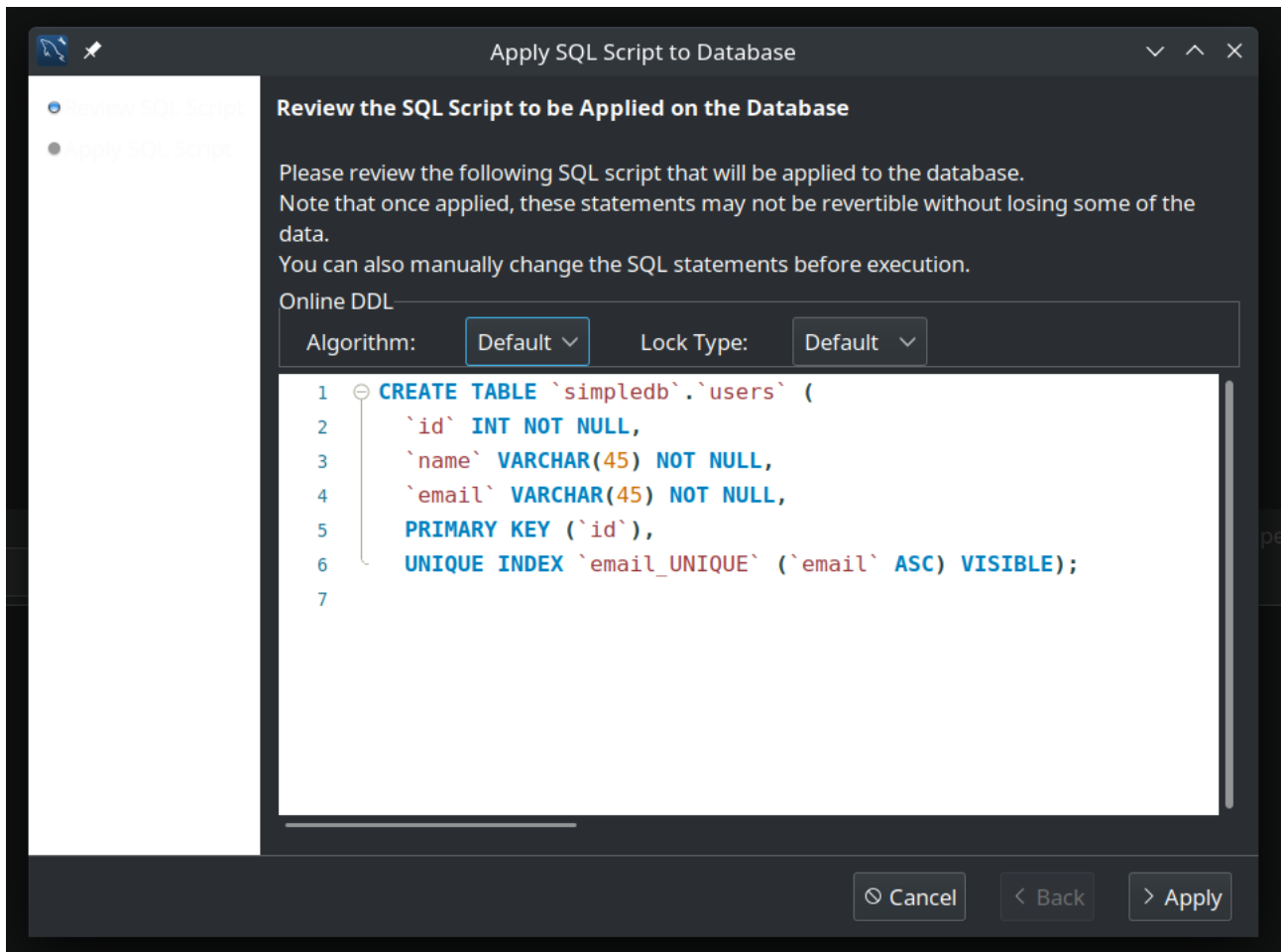
Name:

users

Schema:

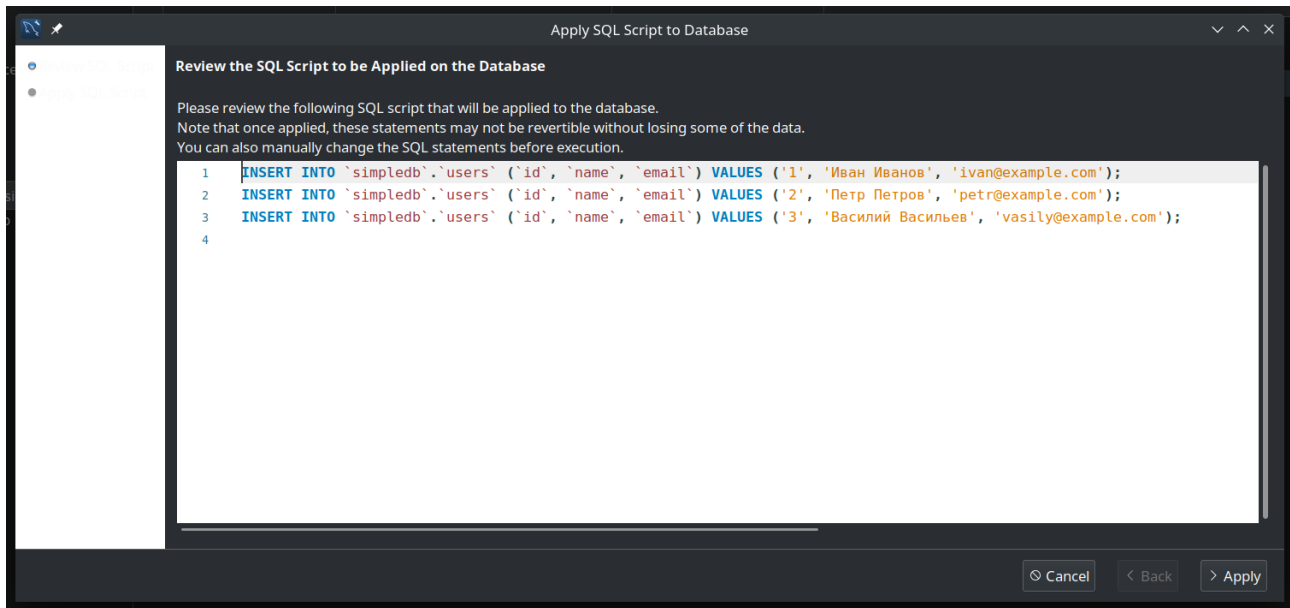
simpledb

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	G	Default / Exp
id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
name	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
email	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

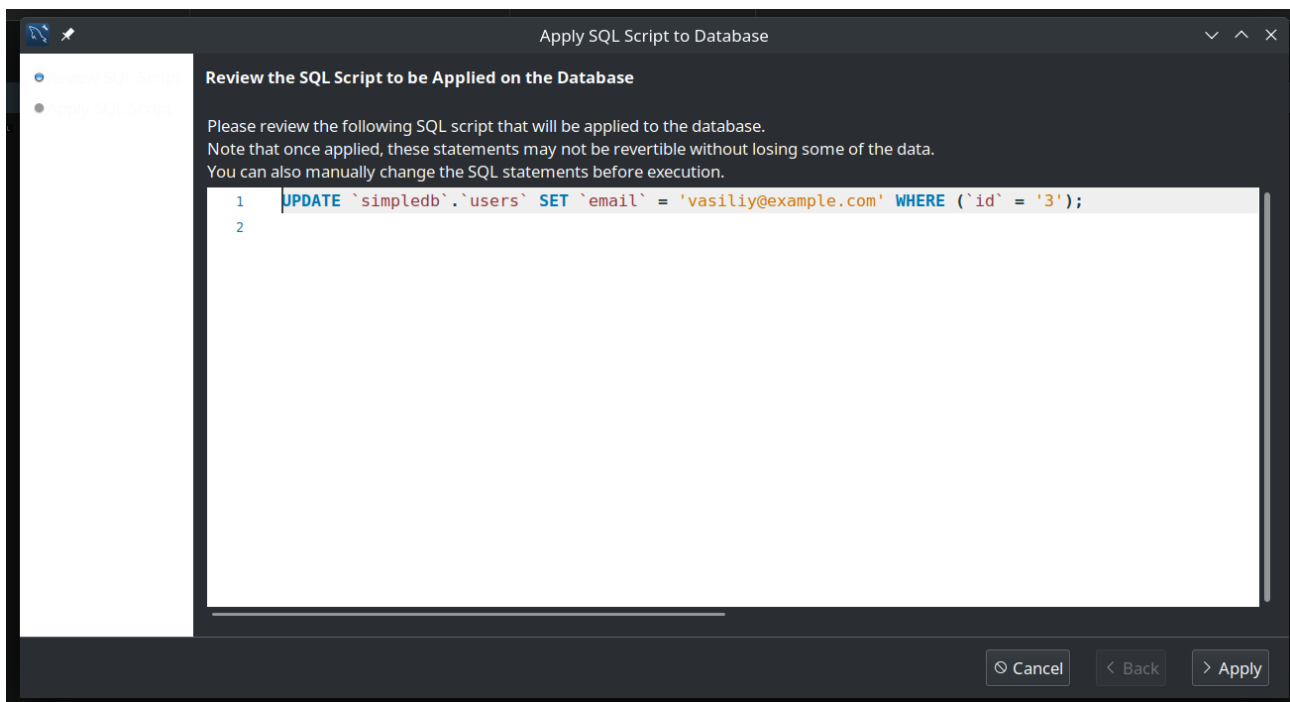


## Задание 4

Добавьте несколько примеров-записей в созданную таблицу. Для этого необходимо кликнуть на название таблицы левой кнопкой мыши и выбрать пиктограмму, где изображена таблица и молния (см. изображение ниже).



## Обновление одного и полей



## 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()*

В таблицу users были добавлены следующие поля:

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	G	Default / Expression
* id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
* name	VARCHAR(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
* email	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
* gender	ENUM('M', 'F')	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
* bday	DATE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
* postal_code	VARCHAR(10)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
* rating	FLOAT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
* created	TIMESTAMP	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

```
1 ALTER TABLE `simpledb`.`users`
2 ADD COLUMN `gender` ENUM('M', 'F') NULL AFTER `email`,
3 ADD COLUMN `bday` DATE NULL AFTER `gender`,
4 ADD COLUMN `postal_code` VARCHAR(10) NULL AFTER `bday`,
5 ADD COLUMN `rating` FLOAT NOT NULL AFTER `postal_code`,
6 ADD COLUMN `created` TIMESTAMP NOT NULL AFTER `rating`,
7 CHANGE COLUMN `name` `name` VARCHAR(50) NOT NULL ;
8
```

UPD: На скриншоте забыл указать стандартное значение CURRENT\_TIMESTAMP() для поля created, исправил.

Тип поля ENUM — строковый и позволяет выбирать значения лишь из заранее определенного набора допустимых значений.

TIMESTAMP имеет по умолчанию значение CURRENT\_TIMESTAMP, что является синонимом NOW и возвращает время своего вызова в часовом поясе системы, т. е. В данном случае там будет записана дата и время добавления записи в таблицу.

Поля gender, bday и postal\_code — опциональные (nullable), т. к. не все пользователи могут захотеть указывать эти данные без необходимости. Поля rating и created — обязательные, т. к. присваиваются самой системой (приложением) независимо от желания пользователей.

## Задание 6

*Дополните таблицу, добавив данные двумя способами:*

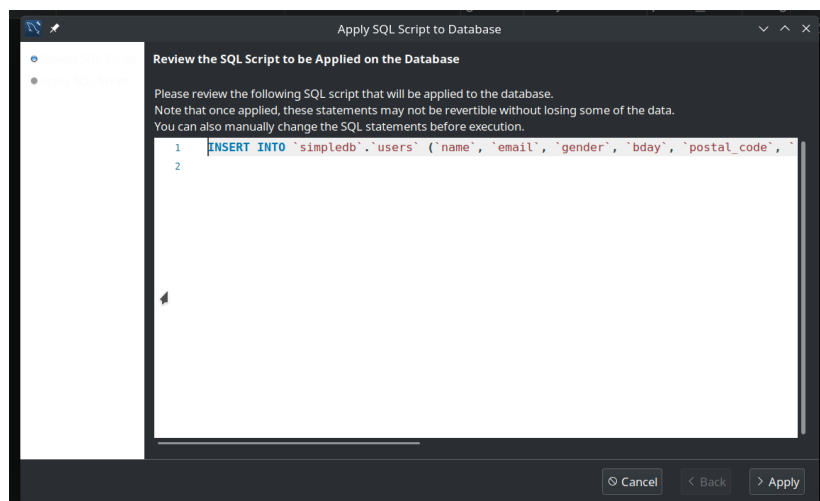
*с помощью внесения данных вручную (как это было сделано ранее);*

*с помощью выполнения SQL-запросов ниже;*

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

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

Добавим пользователя вручную через GUI:



Добавим пользователя с помощью SQL запроса:

The screenshot shows a database management tool interface. At the top, a SQL query is entered in a text area:

```
1. SELECT * FROM simpledb.users;  
2. INSERT INTO 'simpledb'.users ('name', 'email', 'postal_code', 'gender', 'bday', 'rating')  
3
```

Below the query, a table of results is displayed with the following columns: #, id, name, email, gender, bday, postal\_code, rating, created. The first row of data is:

#	id	name	email	gender	bday	postal_code	rating	created
1	1	Ekaterina	ekaterina.petrova@o...	F	2000-02-11	145789	1.123	2022-

On the right side of the interface, there is a sidebar with a 'SELECT' dropdown and a detailed explanation of the SELECT statement syntax, including information about table references, partitions, and the WHERE clause.

## 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.

Содержимое файла:

The screenshot shows a text file named 'test.sql' containing two SQL queries:

```
1. id, name, email, gender, bday, postal_code, rating, created  
2. 1, Ekaterina, ekaterina.petrova@outlook.com, F, 2000-02-11, 145789, 1.123, "2022-02-16 17:03:08"  
3. 2, Paul, paul@superpochta.ru, M, 1998-08-12, 123789, 1, "2022-02-16 17:05:35"  
4.
```

## Задание 8

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

*resumeid*, INT, PK, NN, AI

*userid*, INT, NN

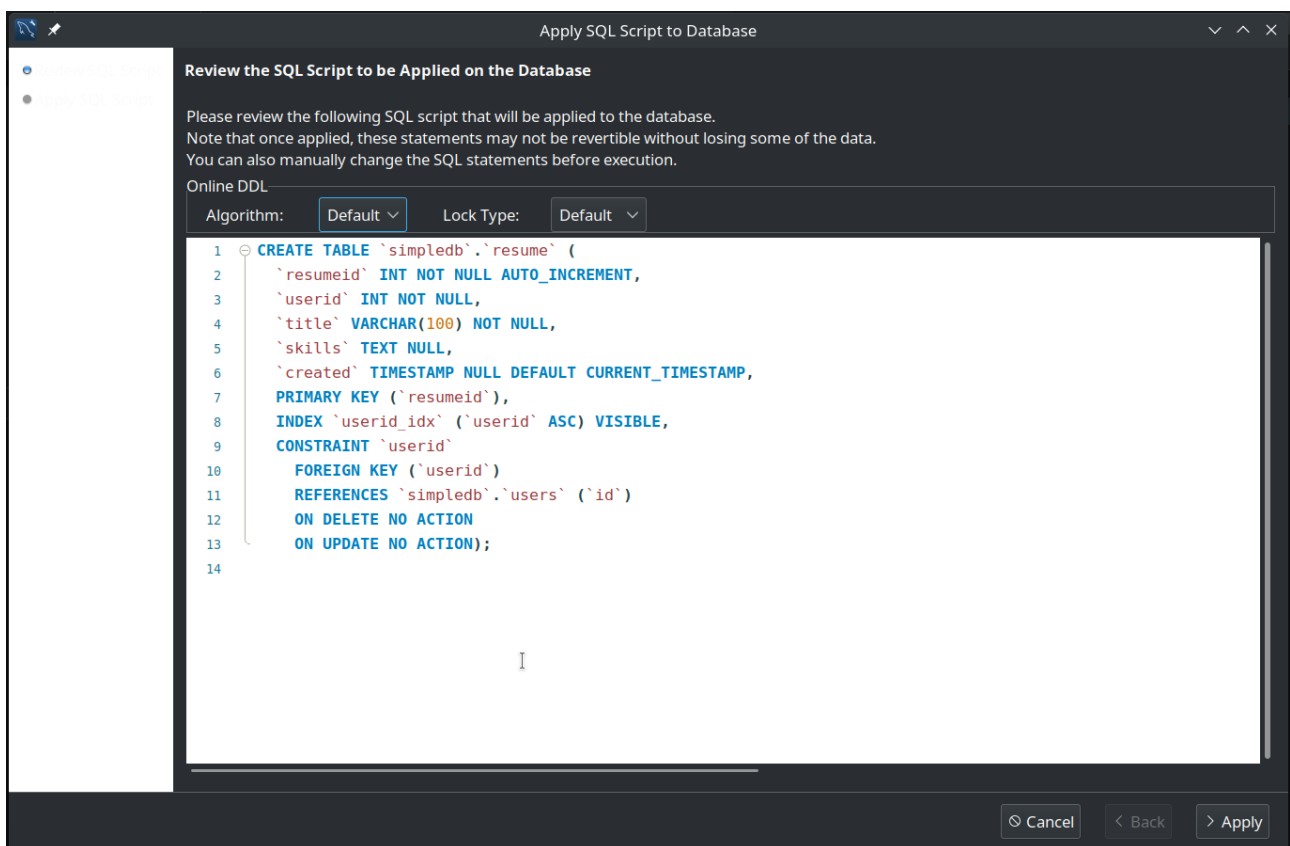
*title*, VARCHAR(100), NN

*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*.

Запрос:



Судя по 12 и 13 строке запроса, при удалении связанных сущностей, записи в другой таблице останутся без изменений.

## 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)?



#	resumeid	userid	title	skills	created
1	1	1	Мое резюме	Сварка, Фрезеровка, работа на токарном станке	2022-02-16 17:24:27
2	2	2	Работник мечты	Вязание, вышивание крестиком, аквагим детям	2022-02-16 17:25:59
*	NULL	NULL	NULL	NULL	NULL

Каждый пользователь может иметь неограниченное кол-во резюме в этой таблице, т. к. единственное условие — отсутствие повторений primary key, которым здесь является resumeid, а не userid. Соответственно, в противном случае, каждый пользователь мог бы иметь лишь одно резюме.

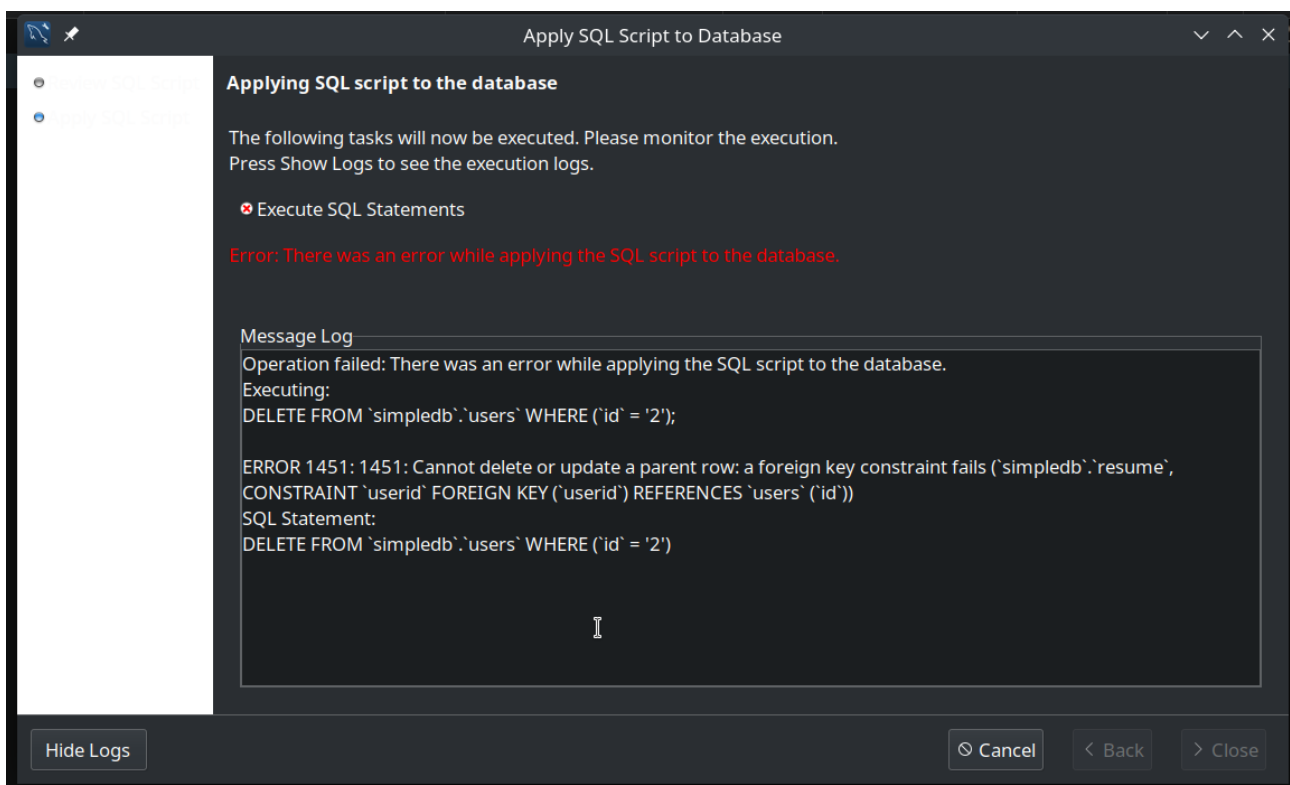
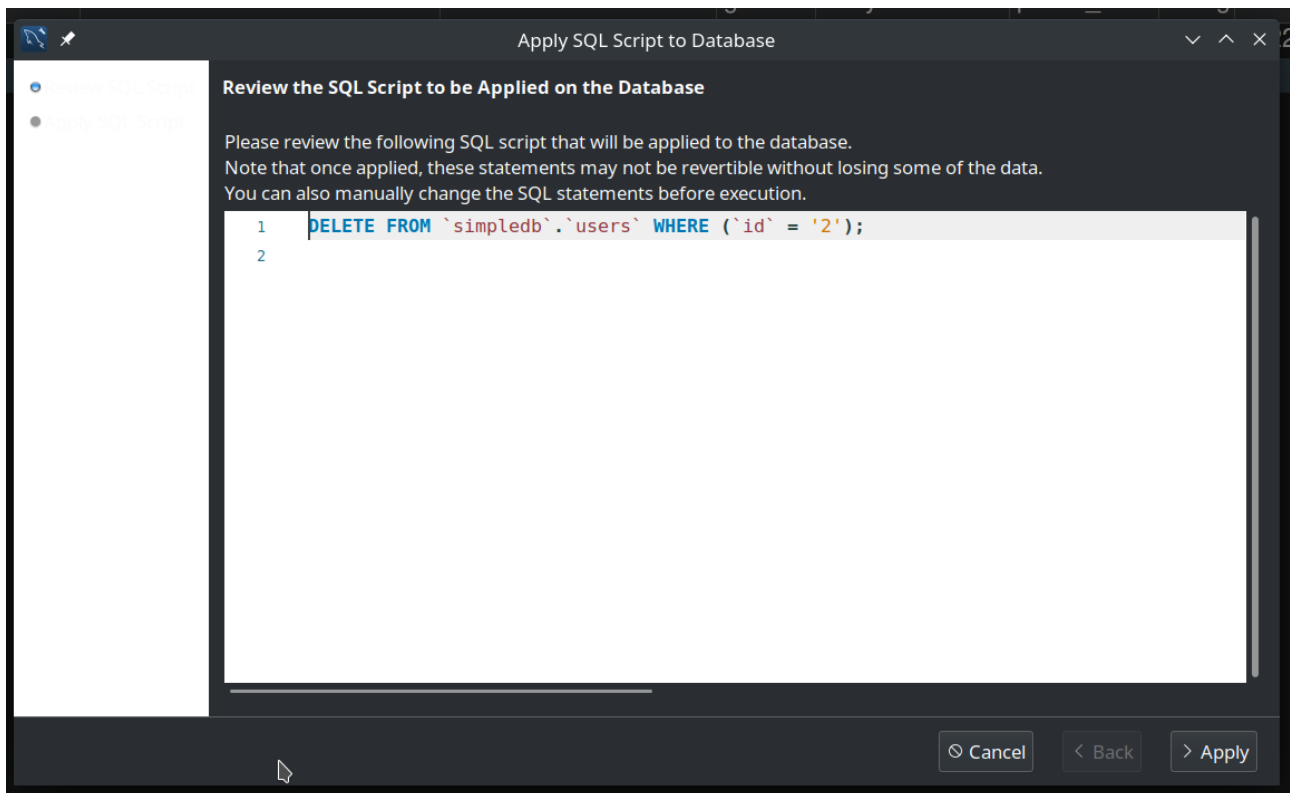
Минимум резюме на одного пользователя — 0.

## Задание 10

Удалите одного или двух таких пользователей, что для них существуют записи в таблице resume.

Это можно сделать в режиме Result Grid или Form Editor аналогично тому как вносятся изменения в записи таблицы.

Вставьте в отчет SQL-запросы, которые вы увидите после нажатия Apply.



Величко Арсений Александрович  
ИВТ 2 курс, 2 группа, 3 подгруппа  
Предмет: Базы данных

Удалить пользователя не получилось — помешали связанные данные в другой таблице.