

**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 “Management”:**

1. Раздел “Server Status”. В разделе отображается общая информация о сервере и подключении к нему. Информация логически сгруппирована. Можно выделить следующие группы:
  - a. *Общая информация* (например, название хоста, номер порта, версия БД).
  - b. *Настройки сервера* (например, включен ли брандмауэр, используется ли SSL)
  - c. *Каталоги сервера*
  - d. *Сводка по используемым ресурсам компьютера* (ОЗУ, процессор и т. д.)
  - e. *Настройки соединения SSL* (если SSL включена).
2. Раздел “Client Connections”. В разделе клиентских соединений перечисляются активные и спящие клиентские соединения MySQL. Имеется возможность уничтожать операторы и соединения, а также просматривать дополнительные сведения и атрибуты соединения. Выделяют три вкладки, помимо основного окна:
  - a. *Подробности* (сведения о соединении: идентификатор процесса, тип, пользователь, хост, инструмент и другая дополнительная информация)
  - b. *Блокировки* (MySQL использует блокировку метаданных для управления доступом. Иногда запрос может быть заблокирован, когда им управляет другое соединение от другого пользователя. Функция блокировки использует эти блокировки метаданных MySQL (MDL), чтобы показать заблокированные соединения, а также показывает информацию о блокировках, что они ожидают и что они удерживают)
  - c. *Атрибуты* (атрибуты подключения, такие как ОС, имя клиента, версия клиента и платформа)
3. Раздел “Users and Privileges”. Раздел содержит список всех пользователей и привилегий. На этой вкладке вы можете добавлять учетные записи пользователей и управлять ими, настраивать привилегии и просрочивать пароли. Выделяют следующие группы:

- a. *Учетные записи пользователей* состоят из вертикального поля, в котором перечислены все учетные записи пользователей, связанные с активным соединением MySQL.
  - b. *Вход* (представлена следующая информация, связанная с выбранной учетной записью пользователя: имя для входа, тип аутентификации, ограничение на совпадение хостов, пароль и подтверждение пароля)
  - c. *Ограничения учетной записи* (определяет следующие ограничения для выбранной учетной записи пользователя: максимальное количество запросов, обновлений и подключений, а также количество одновременных подключений)
  - d. *Административные роли* (роли — это быстрый способ предоставления пользователю набора привилегий в зависимости от работы, которую он должен выполнять на сервере. Также возможно назначить несколько ролей учетной записи или назначить привилегии без предварительного назначения ролей)
  - e. *Схемы привилегий* (назначение прав доступа к одной или нескольким схемам по учетной записи пользователя)
4. Раздел “Status and System Variables”. Раздел содержит полный набор серверных переменных для активного соединения MySQL.
  5. Раздел “Data Export”. Этот раздел позволяет вам экспортировать данные MySQL.
  6. Раздел “Data Import/Restore”. В данном разделе возможно восстановить экспортированные данные.

#### **Раздел “Instance”:**

1. Раздел “Startup / Shutdown”. Раздел позволяет выполнять следующие действия по управлению службами: просмотр журнала сообщений о запуске, запуск и выключение экземпляра MySQL, просмотр текущего состояния экземпляра MySQL. На вкладке представлены две кнопки: “Остановить сервер” и “Перевести в автономный режим”.
2. Раздел “Server Logs”. В разделе отображается информация журнала для сервера, также содержатся дополнительные вкладки для общих журналов ошибок и медленных журналов (если они доступны).
3. Раздел “Options File”. Раздел используется для просмотра и редактирования файла конфигурации MySQL.

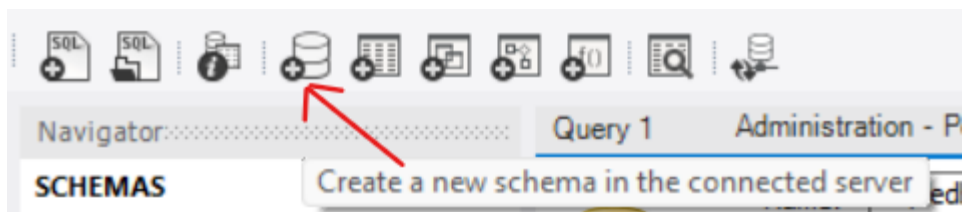
#### **Раздел “Performance”:**

1. Раздел “Dashboard”. В разделе можно посмотреть статистику производительности сервера на графической панели.
  - a. *Статус сети* (показывает статистику сетевого трафика, отправленного и полученного через клиентские соединения. Сюда включают входящий сетевой трафик, исходящий сетевой трафик и клиентские подключения)
  - b. *Статус MySQL* (показывает основную активность сервера MySQL и статистику производительности. Сюда включают эффективность открытого кэша таблиц, количество выполненных операторов SQL и количество (в секунду) для операторов SELECT, INSERT, UPDATE, DELETE, CREATE, ALTER и DROP)
  - c. *Статус InnoDB* (показывает обзор буферного пула InnoDB и дисковой активности. В вкладке выделяют три группы: применение, запись, чтение)
2. Раздел “Performance Reports”. Раздел представляет данные отчета о производительности, а также дает возможность экспортировать их. Имеются следующие элементы управления:
  - a. *Экспорт* (экспорт всех записей и связанных данных из текущего отчета о производительности. Открывает диалоговое окно файла для экспорта)
  - b. *Копировать выбранное* (копирует одну запись и связанные данные (и заголовки столбцов) из текущего отчета о производительности. Сохраняет в буфер обмена системы)
  - c. *Копировать запрос* (копирует SQL-запрос, сгенерировавший отчет о производительности. Сохраняет в системный буфер обмена)
  - d. *Обновить* (обновляет (перезагружает) отчет о производительности)
3. Раздел “Performance Schemas Setup”. Данный раздел позволяет использовать инструмент для сбора статистики и данных о производительности, а также регистрировать собранные события в таблицы для дальнейшего анализа. Сначала загружается вкладка “Easy Setup”, которой достаточно для большинства пользователей. Чтобы включить все доступные инструменты Performance Schema, наведите указатель мыши на пункт “Fully Enabled” и щелкните кружок на ползунке.

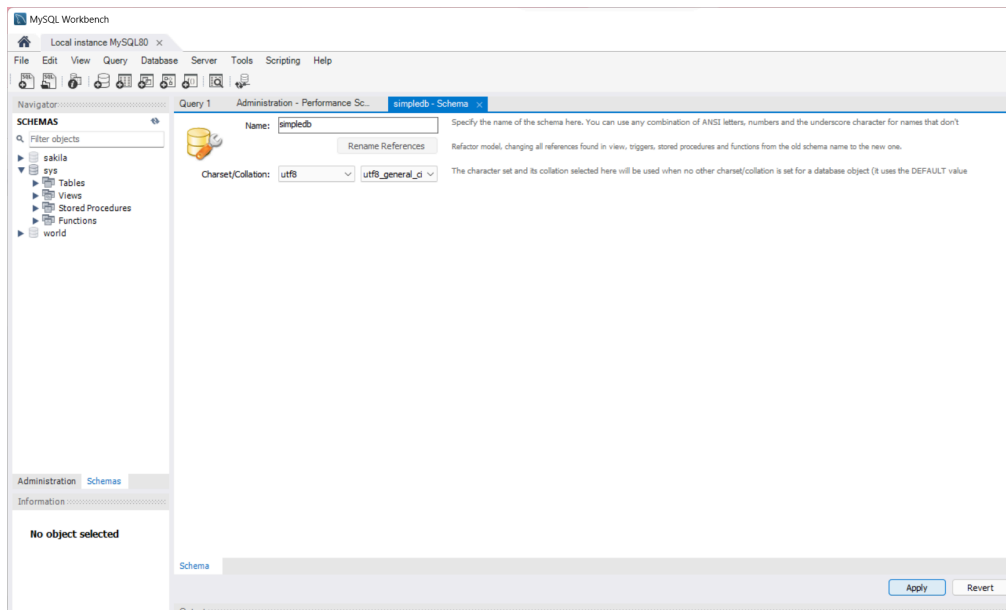
### **Задание 2. Создать и настроить новую базу данных *simpledb*.**

1. Нажимаем кнопку “Создание новой базы данных в рамках данного

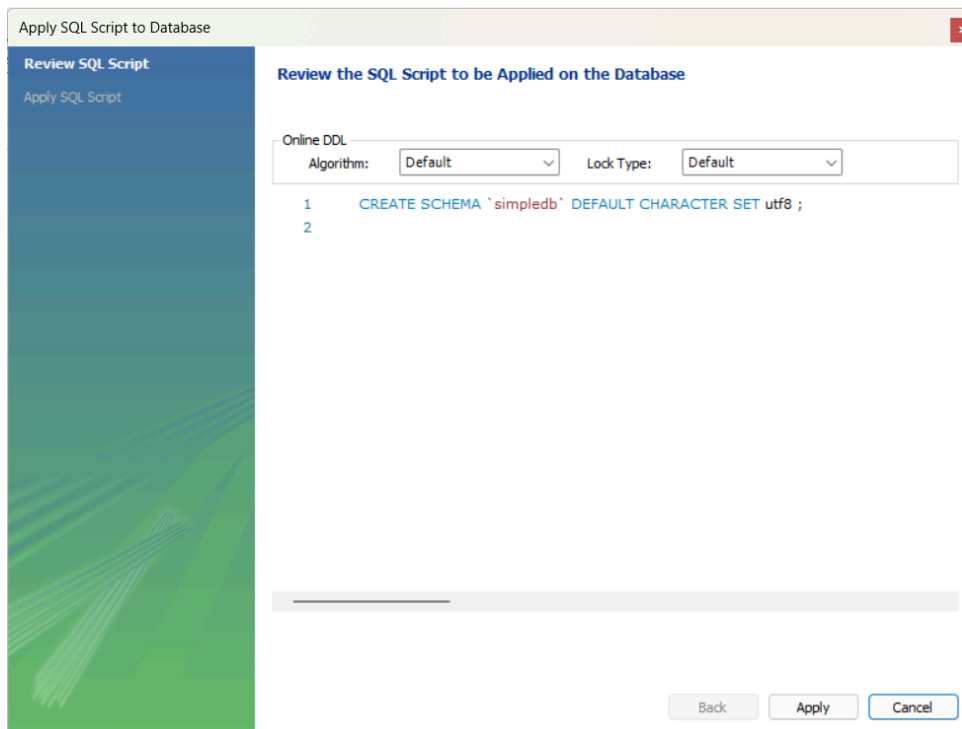
подключения к серверу”



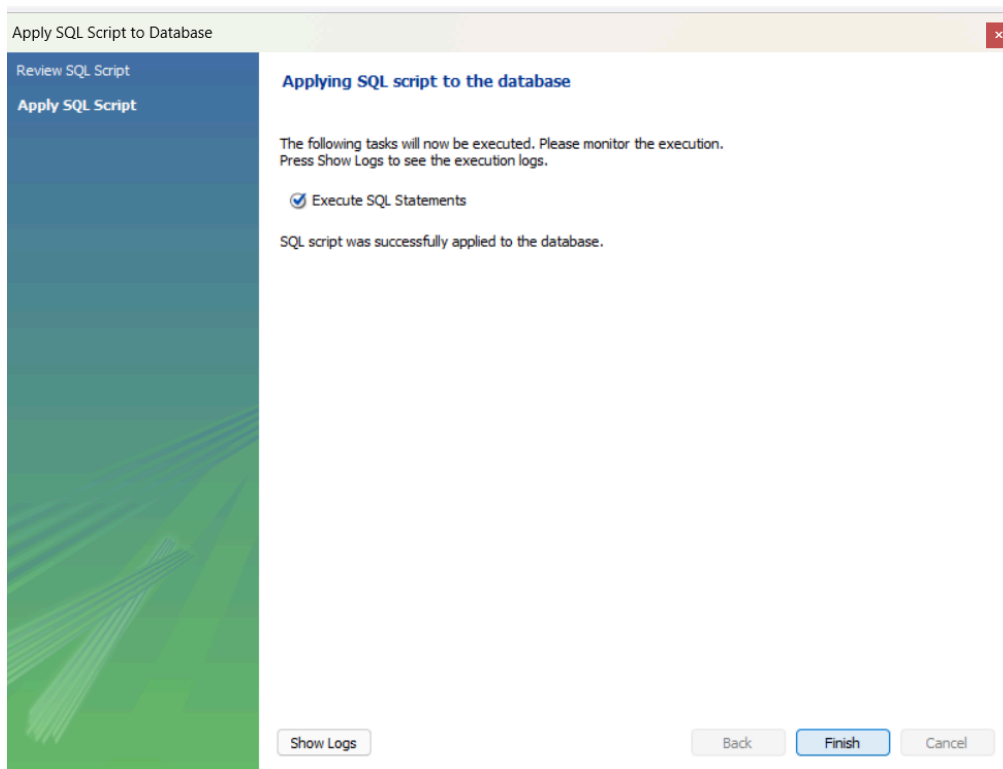
## 2. Задаем настройки и нажимаем кнопку Apply



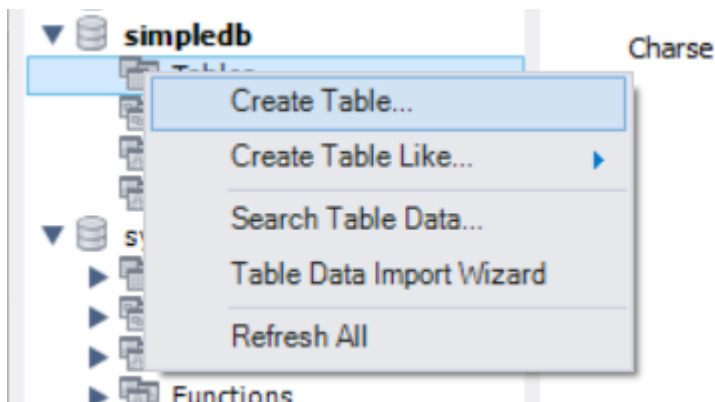
## 3. Нажимаем кнопку Apply



## 4. Нажимаем кнопку Finish. Теперь у нас создана новая база данных.



5. Далее, создадим несколько таблиц в этой базе данных. Для этого раскроем с помощью треугольника содержимое этой БД. В контекстном меню по пункту Tables, выберем пункт "Create Table".



6. Создадим таблицу users, заполнив данные, нажмем кнопку Apply

Query 1 Administration - Performance Sc... simpledb - Schema users - Table

Table Name:  Schema: **simpledb**

Charset/Collation:   Engine:

Comments:

Column Name	Datatype	PK	NN	UQ	B	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 checked="" 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"/>	

Column Name:  Data Type:

Charset/Collation:   Default:

Comments:

Storage: ☐ Virtual ☐ Stored

☐ Primary Key ☒ Not Null ☒ Unique

☐ Binary ☐ Unsigned ☐ Zero Fill

☐ Auto Increment ☐ Generated

Columns Indexes Foreign Keys Triggers Partitioning Options

Apply Revert

## 7. Нажимаем кнопку Apply

Apply SQL Script to Database

Review SQL Script

Apply SQL Script

Review the SQL Script to be Applied on the Database

Online DDL

Algorithm:  Lock Type:

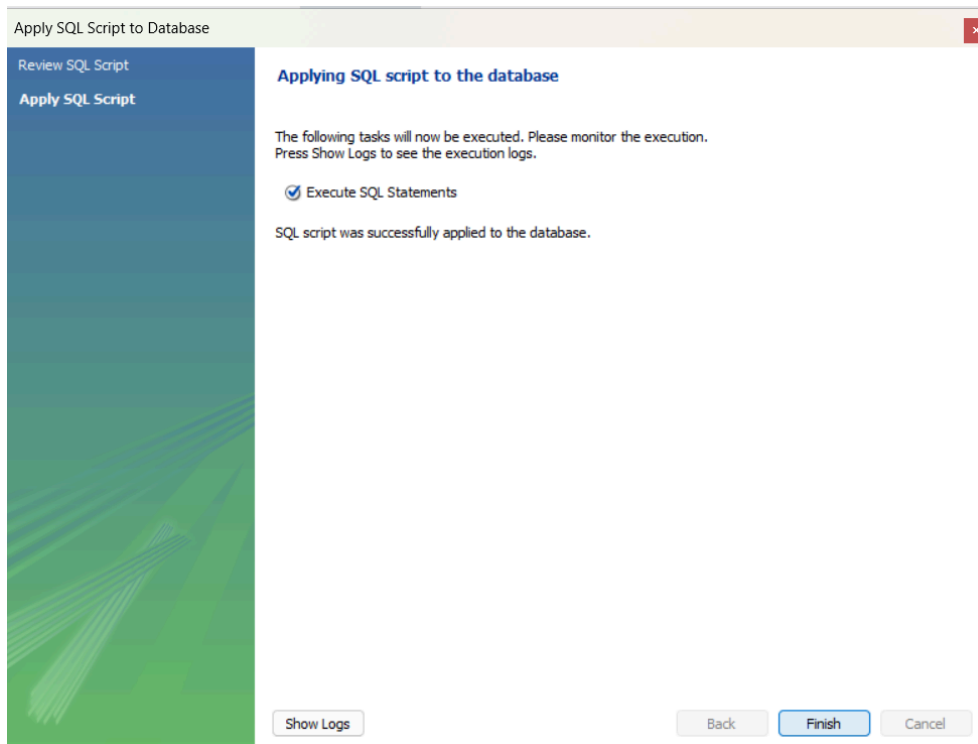
```

1 CREATE TABLE `simpledb`.`users` (
2   `id` INT NOT NULL AUTO_INCREMENT,
3   `name` VARCHAR(45) NOT NULL,
4   `email` VARCHAR(45) NOT NULL,
5   PRIMARY KEY (`id`),
6   UNIQUE INDEX `email_UNIQUE` (`email` ASC) VISIBLE);
7

```

Back Apply Cancel

## 8. Нажимаем кнопку Finish. Теперь у нас создана новая таблица.



### **Задание 3**

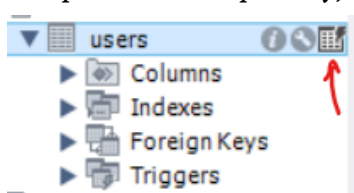
*Скопируйте запрос, соответствующий созданию этой таблицы и вставьте его в отчет по выполнению этой лабораторной работы.*

```
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;
```

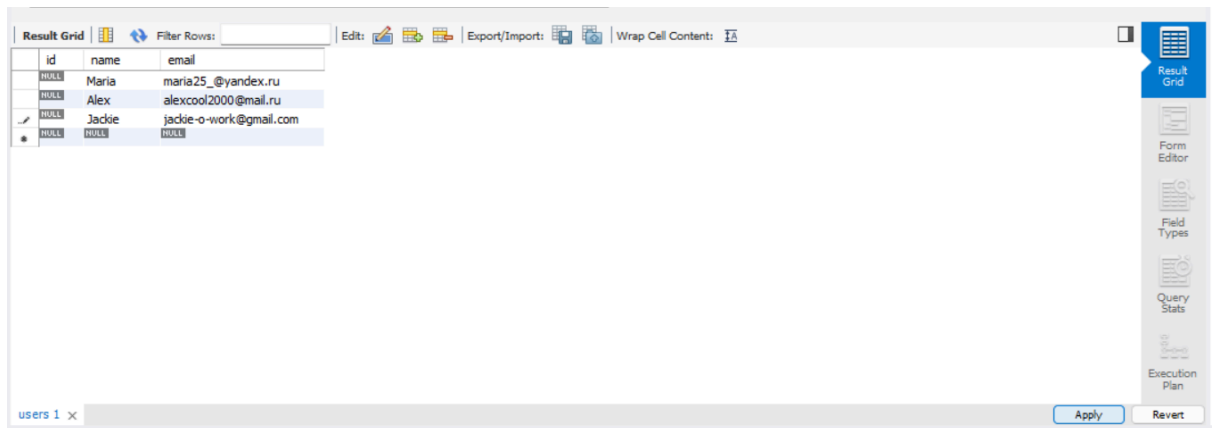
### **Задание 4**

*Добавьте несколько примеров-записей в созданную таблицу.*

1. Необходимо кликнуть на название таблицы левой кнопкой мыши и выбрать пиктограмму, где изображена таблица и молния



## 2. Введем данные и нажмем Apply



```
INSERT INTO `simplifiedb`.`users` (`name`, `email`) VALUES ('Maria', 'maria25_@yandex.ru');
```

```
INSERT INTO `simplifiedb`.`users` (`name`, `email`) VALUES ('Alex', 'alexcool2000@mail.ru');
```

```
INSERT INTO `simplifiedb`.`users` (`name`, `email`) VALUES ('Jackie', 'jackie-o-work@gmail.com');
```

После этого обновите одно или несколько полей (например, *name* и/или *email*), нажмите *Apply* и сохраните полученный SQL-запрос в отчете.

Какой SQL-запрос при этом выполнится? Отразите его в отчете.

```
UPDATE `simplifiedb`.`users` SET `name` = 'Mari', `email` = 'mariaivanova25@yandex.ru' 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()*

```
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` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP()
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.*

Присваивает текущую метку времени как значение по умолчанию, значение может обновляться автоматически

### **Замечание 3**

*Определите самостоятельно какие поля могут быть NULL и будут иметь это значение в качестве значения по умолчанию, представив себе пользователя, который не очень хочет делиться персонализированной информацией. Отрадите это в отчёте.*

gender, bday Date, postal\_code

### **Задание 6**

*Дополните таблицу, добавив данные двумя способами:*

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

Екатерину я ввела вручную, копируя данные значения

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content:

	id	name	email	gender	bday	postal_code	rating	created
	1	Mari	mariaivanova25@yandex.ru					2023-02-16 21:09:54
	2	Alex	alexcool2000@mail.ru					2023-02-16 21:09:54
	3	Jackie	jackie-o-work@gmail.com					2023-02-16 21:09:54
▶	4	Ekaterina	ekaterina.petrova@outloo...		2000-02-11	145789	1.123	2023-02-16 23:40:39
	5	Paul	paul@superpochta.ru	M	1998-08-12	123789	1	2023-02-16 23:43:20
*								

Для Паула в свою очередь нажала кнопку Apply

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content:

id	name	email	gender	bday	postal_code	rating	created
1	Mari	mariaivanova25@yandex.ru	NULL	NULL	NULL	NULL	2023-02-16 21:09:54
2	Alex	alexcool2000@mail.ru	NULL	NULL	NULL	NULL	2023-02-16 21:09:54
3	Jackie	jackie-o-work@gmail.com	NULL	NULL	NULL	NULL	2023-02-16 21:09:54
4	Ekaterina	ekaterina.petrova@outoo...	F	2000-02-11	145789	1.123	2023-02-16 23:40:39
5	Paul	paul@superpochta.ru	M	1998-08-12	123789	1	2023-02-16 23:43:20
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

users 1

Apply

Revert

Result Grid

Form Editor

Field Types

Query Stats

В появившемся окне я ввела: `INSERT INTO `simplifiedb`.`users` (`name`, `email`, `postal_code`, `gender`, `bday`, `rating`) VALUES ('Paul', 'paul@superpochta.ru', '123789', 'm', '1998-08-12', '1');` и нажала Apply и Finish

Apply SQL Script to Database

Review SQL Script

Apply SQL Script

Review the SQL Script to be Applied on the Database

1 `INSERT INTO `simplifiedb`.`users` (`name`, `email`, `postal_code`, `gender`, `bdi`  
2

Back

Apply

Cancel

### **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.

```
/*  
-- Query: SELECT * FROM simplifiedb.users  
LIMIT 0, 1000  
  
-- Date: 2023-02-16 23:51  
*/  
INSERT INTO `` (`id`,`name`,`email`,`gender`,`bday`,`postal_code`,`rating`,`created`)  
VALUES (1,'Mari','mariaivanova25@yandex.ru','F','2003-01-27','123468',1,'2023-02-16  
21:09:54');  
INSERT INTO `` (`id`,`name`,`email`,`gender`,`bday`,`postal_code`,`rating`,`created`)  
VALUES (2,'Alex','alexcool2000@mail.ru','M',NULL,'735921',0.6,'2023-02-16  
21:09:54');  
INSERT INTO `` (`id`,`name`,`email`,`gender`,`bday`,`postal_code`,`rating`,`created`)  
VALUES (3,'Jackie','jackie-o-work@gmail.com','M',NULL,NULL,1.5,'2023-02-16  
21:09:54');  
INSERT INTO `` (`id`,`name`,`email`,`gender`,`bday`,`postal_code`,`rating`,`created`)  
VALUES  
(4,'Ekaterina','ekaterina.petrova@outlook.com','F','2000-02-11','145789',1.123,'2023-  
02-16 23:40:39');  
INSERT INTO `` (`id`,`name`,`email`,`gender`,`bday`,`postal_code`,`rating`,`created`)  
VALUES (5,'Paul','paul@superpochta.ru','M','1998-08-12','123789',1,'2023-02-16  
23:43:20');
```

### **Задание 8**

Создайте еще одну таблицу с названием *resumе*

Создание новой таблицы:

simpledb - Schema users - Table resume - Table

Table Name:  Schema: **simpledb**

Charset/Collation:   Engine:

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
resumeid	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
userid	INT	<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"/>	
title	VARCHAR(100)	<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"/>	
skills	TEXT	<input type="checkbox"/>	<input 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 type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	CURRENT_TIMESTAMP()

Column Name:  Data Type:

Charset/Collation:   Default:

Comments:

Storage: ☐ Virtual ☐ Stored

☐ Primary Key ☐ Not Null ☐ Unique

☐ Binary ☐ Unsigned ☐ Zero Fill

☐ Auto Increment ☐ Generated

Columns Indexes Foreign Keys Triggers Partitioning Options

Apply Revert

Output

## Определение внешнего ключа:

simpledb - Schema users - Table resume - Table

Table Name:  Schema: **simpledb**

Charset/Collation:   Engine:

Comments:

Foreign Key Name	Referenced Table	Column	Referenced Column
userid	'simpledb'. 'users'	<input checked="" type="checkbox"/> resumeid	id
		<input checked="" type="checkbox"/> userid	
		<input type="checkbox"/> title	
		<input type="checkbox"/> skills	
		<input type="checkbox"/> created	

Foreign Key Options

On Update:

On Delete:

☐ Skip in SQL generation

Foreign Key Comment:

Columns Indexes Foreign Keys Triggers Partitioning Options

Apply Revert

```
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);

### **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.

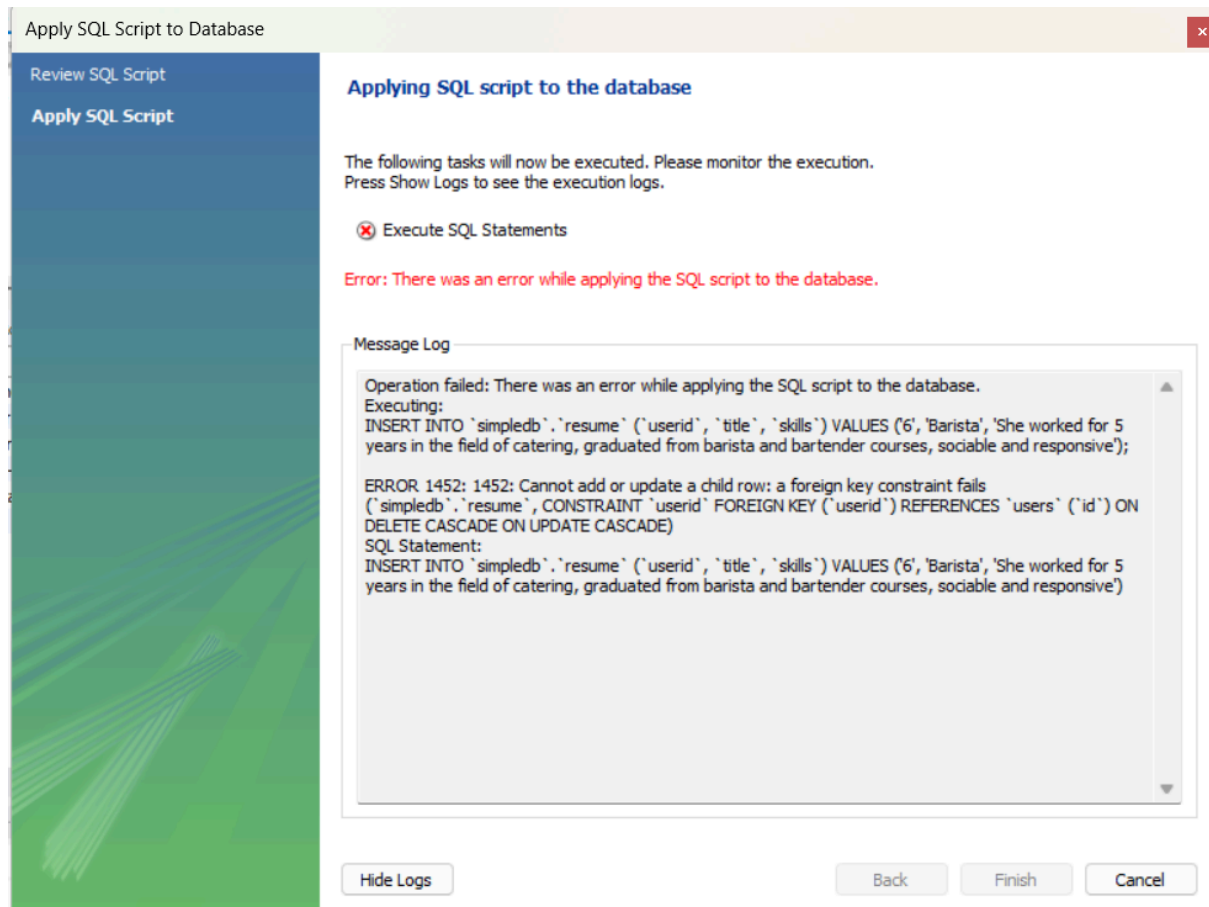
Подумайте и напишите в отчете, сколько резюме может быть у одного пользователя (минимум и максимум)?

Я думаю, что минимум - это отсутствие резюме, а максимум - бесконечно много.

```
/*  
-- Query: SELECT * FROM simpledb.resume  
LIMIT 0, 1000  
  
-- Date: 2023-02-17 00:14  
*/  
INSERT INTO `` (`resumeid`,`userid`,`title`,`skills`,`created`) VALUES  
(1,1,'Teacher','Has a pedagogical education, developed soft skills','2023-02-17  
00:12:22');  
INSERT INTO `` (`resumeid`,`userid`,`title`,`skills`,`created`) VALUES  
(2,2,'Reporter','Has his own blog with a large audience, delivered a  
speech','2023-02-17 00:12:22');  
INSERT INTO `` (`resumeid`,`userid`,`title`,`skills`,`created`) VALUES  
(3,3,'Singer','Graduated from a music school, an observatory, has been performing  
for 10 years','2023-02-17 00:12:22');  
INSERT INTO `` (`resumeid`,`userid`,`title`,`skills`,`created`) VALUES  
(4,4,'Seller-cashier','Worked for 5 years in different stores, sociable and  
responsive','2023-02-17 00:12:22');  
INSERT INTO `` (`resumeid`,`userid`,`title`,`skills`,`created`) VALUES  
(5,4,'Barista','She worked for 5 years in the field of catering, graduated from barista  
and bartender courses, sociable and responsive','2023-02-17 00:12:22');
```

Попробуйте добавить в таблицу resume строчку с userid несуществующего пользователя (такого пользователя, id которого нет в таблице users). Будет ли это возможно? Каков будет результат?

Результат добавления строчки с userid несуществующего пользователя. Это невозможно, программа будет выдавать ошибку



### Задание 10

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

```
DELETE FROM `simplifiedb`.`users` WHERE (`id` = '1');
```

К первому пользователю было привязано резюме с должностью “учитель”. После удаления пользователя в таблице “users”, в таблице “resume” резюме автоматически удалилось.

Попробуем в таблице users изменить id какого-то существующего пользователя. Я поменяла айди Екатерины с 4 на 6 в таблице “users”, автоматически поменялось и в таблице “resume”:

Было:

Result Grid								
		Filter Rows:		Edit:		Export/Import:		Wrap Cell Content: <a href="#">I A</a>
	id	name	email	gender	bday	postal_code	rating	created
	2	Alex	alexcool2000@mail.ru	M	NULL	735921	0.6	2023-02-16 21:09:54
	3	Jackie	jackie-o-work@gmail.com	M	NULL	NULL	1.5	2023-02-16 21:09:54
	4	Ekaterina	ekaterina.petrova@outloo...	F	2000-02-11	145789	1.123	2023-02-16 23:40:39
	5	Paul	paul@superpochta.ru	M	1998-08-12	123789	1	2023-02-16 23:43:20
▶*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Стало:

Result Grid								
		Filter Rows:		Edit:		Export/Import:		Wrap Cell Content: <a href="#">I A</a>
	id	name	email	gender	bday	postal_code	rating	created
	2	Alex	alexcool2000@mail.ru	M	NULL	735921	0.6	2023-02-16 21:09:54
	3	Jackie	jackie-o-work@gmail.com	M	NULL	NULL	1.5	2023-02-16 21:09:54
▶	5	Paul	paul@superpochta.ru	M	1998-08-12	123789	1	2023-02-16 23:43:20
	6	Ekaterina	ekaterina.petrova@outloo...	F	2000-02-11	145789	1.123	2023-02-16 23:40:39
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Было:

Result Grid					
		Filter Rows:		Edit:	
	resumeid	userid	title	skills	created
	2	2	Reporter	Has his own blog with a la...	2023-02-17 00:12:22
	3	3	Singer	Graduated from a music s...	2023-02-17 00:12:22
	4	4	Seller-cashier	Worked for 5 years in diff...	2023-02-17 00:12:22
▶	5	4	Barista	She worked for 5 years in ...	2023-02-17 00:12:22
*	NULL	NULL	NULL	NULL	NULL

Стало:

Result Grid					
		Filter Rows:		Edit:	
	resumeid	userid	title	skills	created
	2	2	Reporter	Has his own blog with a la...	2023-02-17 00:12:22
	3	3	Singer	Graduated from a music s...	2023-02-17 00:12:22
	4	6	Seller-cashier	Worked for 5 years in diff...	2023-02-17 00:12:22
▶	5	6	Barista	She worked for 5 years in ...	2023-02-17 00:12:22
*	NULL	NULL	NULL	NULL	NULL