**Базы данных**

**Лабораторная работа 1**

**Начало работы с MySQL. MySQL Workbench**

**Отчет**

Выполнила

Иванова Ольга

**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

**Раздел “Management”:**

1. Раздел “Server Status”. В разделе отображается общая информация о сервере и подключении к нему. Информация логически сгруппирована. Можно выделить следующие группы:
   1. Общая информация (например, название хоста, номер порта, версия БД).
   2. Настройки сервера (например, включен ли брандмауэр, используется ли SSL)
   3. Каталоги сервера
   4. Сводка по используемым ресурсам компьютера (ОЗУ, процессор и т. д.)
   5. Настройки соединения SSL (если SSL включена).
2. Раздел “Client connections”. В разделе отображаются текущие активные и «спящие» подключения клиентов к серверу MySQL. Здесь также доступны функции завершения отдельных запросов или всего подключения, а также просмотр расширенной информации и атрибутов подключения. Раздел предназначен для MySQL версии 5.6 и выше — в более ранних версиях отображается только базовая информация (например, хост, база данных, состояние подключения). Информация в разделе сгруппирована по вкладкам:
3. Details (Подробности подключения) — содержит детальную информацию о соединении, включая идентификатор процесса, тип подключения, имя пользователя, хост, статус трассировки и дополнительные параметры.
4. Locks (Блокировки) — отображает блокировки метаданных (MDL), используемые MySQL для управления доступом к объектам, таким как таблицы и триггеры. Здесь можно увидеть, какие подключения заблокированы или находятся в ожидании, и получить информацию о том, какие ресурсы они удерживают и чего ожидают. Данные о блокировках доступны начиная с MySQL 5.7.3 через Performance Schema.
5. Attributes (Атрибуты) — содержит сведения об операционной системе клиента, имени клиента, версии клиента, платформе и других характеристиках подключения.
6. Раздел “Users and Privileges”. Раздел предоставляет доступ к управлению пользователями и их правами на подключённом экземпляре сервера MySQL. Здесь можно создавать и удалять учетные записи, настраивать права доступа, ограничивать ресурсы, назначать роли и устанавливать срок действия паролей. Основные области раздела:
7. User Accounts — список всех пользователей, зарегистрированных на сервере. Здесь отображаются имена пользователей и хосты, с которых разрешено подключение. С помощью кнопок можно добавлять, удалять или обновлять список учетных записей. При выборе пользователя отображается детальная информация о нём во вкладках ниже.
8. Login — вкладка с данными для входа: имя пользователя, способ аутентификации (в том числе стандартный и более защищённые методы, такие как caching\_sha2\_password), разрешённые хосты, пароль и возможность его сброса. Также можно установить требование обязательной смены пароля при следующем входе.
9. Account Limits — ограничения для учетной записи, такие как максимальное число запросов, обновлений и подключений в час, а также количество одновременных подключений.
10. Administrative Roles — вкладка для назначения предопределённых ролей (например, DBA, BackupAdmin, UserAdmin и др.), каждая из которых включает набор привилегий. Можно комбинировать роли и настраивать доступ индивидуально. Если выбраны нестандартные привилегии, автоматически добавляется роль Custom.
11. Schema Privileges — позволяет настраивать доступ к конкретным схемам для выбранного пользователя. Можно указать конкретную схему, шаблон (например, все схемы, начинающиеся с определённого префикса), или применить права ко всем схемам. Привилегии делятся на категории: права на объекты, DDL-права и прочие. Есть возможность удалить конкретную настройку или все привилегии пользователя.
12. Раздел “Status and System Variables”. Отображает все переменные сервера и статусные переменные для активного подключения к MySQL. Здесь можно просматривать текущие значения, описания и копировать отдельные или все переменные. В разделе есть две вкладки:
13. Status Variables — показывает текущие значения статусных переменных, таких как количество запросов, буферы и т. д.
14. System Variables — отображает глобальные системные переменные, значения которых можно изменить (если это разрешено), и сохранить для будущих запусков сервера.
15. Раздел “Data Export”. Позволяет экспортировать базы данных и таблицы MySQL в SQL-формате. Возможности:

* Выбор одной или нескольких схем для экспорта.
* Выбор объектов внутри схем (таблицы, процедуры, события и т. д.).
* Настройка формата экспорта — в отдельные SQL-файлы или проектную папку.
* Расширенные параметры (например, использование REPLACE вместо INSERT, блокировка таблиц на время экспорта, экранирование символов и др.)
* Запуск экспорта с отображением статуса выполнения.

1. Раздел “Data Import/Restore”. Позволяет восстановить данные, экспортированные ранее через mysqldump или из MySQL Workbench. Используется для переноса данных между серверами или восстановления из резервных копий. Возможности:

* Выбор файла SQL или проектной папки для импорта.
* Импорт в существующую схему или создание новой.
* При использовании проектной папки можно выбрать конкретные объекты (таблицы) для восстановления.

**Раздел “Instance”:**

1. Раздел “Startup / Shutdown”. Позволяет вам выполнить следующие действия сервисного контроля:
   1. Смотреть Startup Message Log.
   2. Запустить и завершить сервер MySQL.
   3. Смотреть текущий статус сервера.

Можно получить доступ к действиям контроля из панели Navigator или при нажатии Server и Startup/Shutdown из меню. Следующие данные показывают пример вкладки Administration - Startup / Shutdown с управлением сервером базы данных и показанными кнопками Stop Server и Bring Offline.

1. Раздел “Server Logos”. Позволяет просматривать журналы сервера MySQL для выбранного подключения. Включает два дополнительных таба:
   1. Error Log File — журнал ошибок сервера. Отображает список ошибок и сообщений о сбоях.
   2. Slow Log File — журнал медленных запросов (если включён). Позволяет отслеживать долгие SQL-запросы, влияющие на производительность.
2. Раздел “Options File”. Редактор Options File используется, чтобы рассмотреть и отредактировать MySQL configuration file (my.ini в Windows или my.cnf в Linux и macOS) выбирая опции и другие средства управления GUI и затем внося правки. MySQL Workbench делит файл на свои собственные группы как ряд вкладок (General, Logging, InnoDB и т.д.). Сделайте правки и нажмите Apply, чтобы передать изменения. Можно открыть Administration - Options File из области Navigator или при нажатии Server и Options File из меню. Редактор файлов опций включает следующие компоненты:
   1. Группы файла опций, разделенные на удобные вкладки MySQL Workbench.
   2. Поле поиска Locate option, чтобы искать в вашем конфигурационном файле.
   3. Путь Configuration File, таким образом, вы знаете, какой конфигурационный файл вы редактируете.
   4. Селектор группы опций в файле, чтобы выбрать группу опций, чтобы отредактировать. Поскольку та же самая опция может быть определена в нескольких группах, важно выбрать правильную группу, когда редактируете.

**Раздел “Performance”:**

1. Раздел “Dashboard”. Графическая панель мониторинга, отображающая ключевые параметры работы MySQL-сервера в реальном времени. Включает несколько блоков:
2. Network Status — показывает сетевую активность: входящий/исходящий трафик и количество клиентских подключений.
3. MySQL Status — основные метрики сервера: эффективность кэша открытых таблиц, количество выполненных SQL-команд (SELECT, INSERT и др.) в секунду.
4. InnoDB Status — состояние хранилища InnoDB:

* Usage — логические чтения/записи, использование буфера.
* Writes — объёмы записанных данных и операции записи на диск.
* Reads — операции чтения с диска и через doublewrite-буфер.

1. Раздел “Performance Reports”. Этот раздел содержит более 20 готовых отчётов на основе Performance Schema и представлений SYS-схемы, которые позволяют анализировать работу сервера MySQL на высоком уровне. С помощью отчётов можно выявлять проблемные участки, “тяжёлые” SQL-запросы, анализировать статистику ожиданий и метрики движка InnoDB. Отчёты сгруппированы по категориям:
   1. Memory Usage — отчёты по потреблению памяти (по событиям, пользователям, хостам, потокам).
   2. Hot Spots for I/O — отчёты по активной работе с файлами и I/O-операциями.
   3. High Cost SQL Statements — отчёты по дорогостоящим SQL-запросам (долгое выполнение, использование сортировок, временных таблиц, полные сканирования и др.).
   4. Database Schema Statistics — метрики по таблицам, индексам, неиспользуемым индексам, полным сканированиям и др.
   5. Wait Event Times (Expert) — отчёты по событиям ожидания (по пользователям, категориям и времени).
   6. InnoDB Statistics — отчёты по использованию буфера InnoDB.
   7. User Resource Use — статистика ресурсов и выполнения запросов по пользователям.

Управление отчётами включает:

* Export — экспорт всех записей отчёта.
* Copy Selected — копирование отдельной записи.
* Copy Query — копирование SQL-запроса, сформировавшего отчёт.
* Refresh — обновление данных отчёта.

1. Раздел “Performance Schema Setup”. Предоставляет интерфейс для настройки и включения системы мониторинга Performance Schema, которая используется для сбора статистики по производительности. Настройка включает два режима:
2. Easy Setup — простой режим для большинства пользователей. Слайдер позволяет включить все инструменты (инструментация).
3. Show Advanced — расширенный режим с вкладками, позволяющий тонко настраивать, какие именно события, объекты или действия сервера отслеживать.

Особенности:

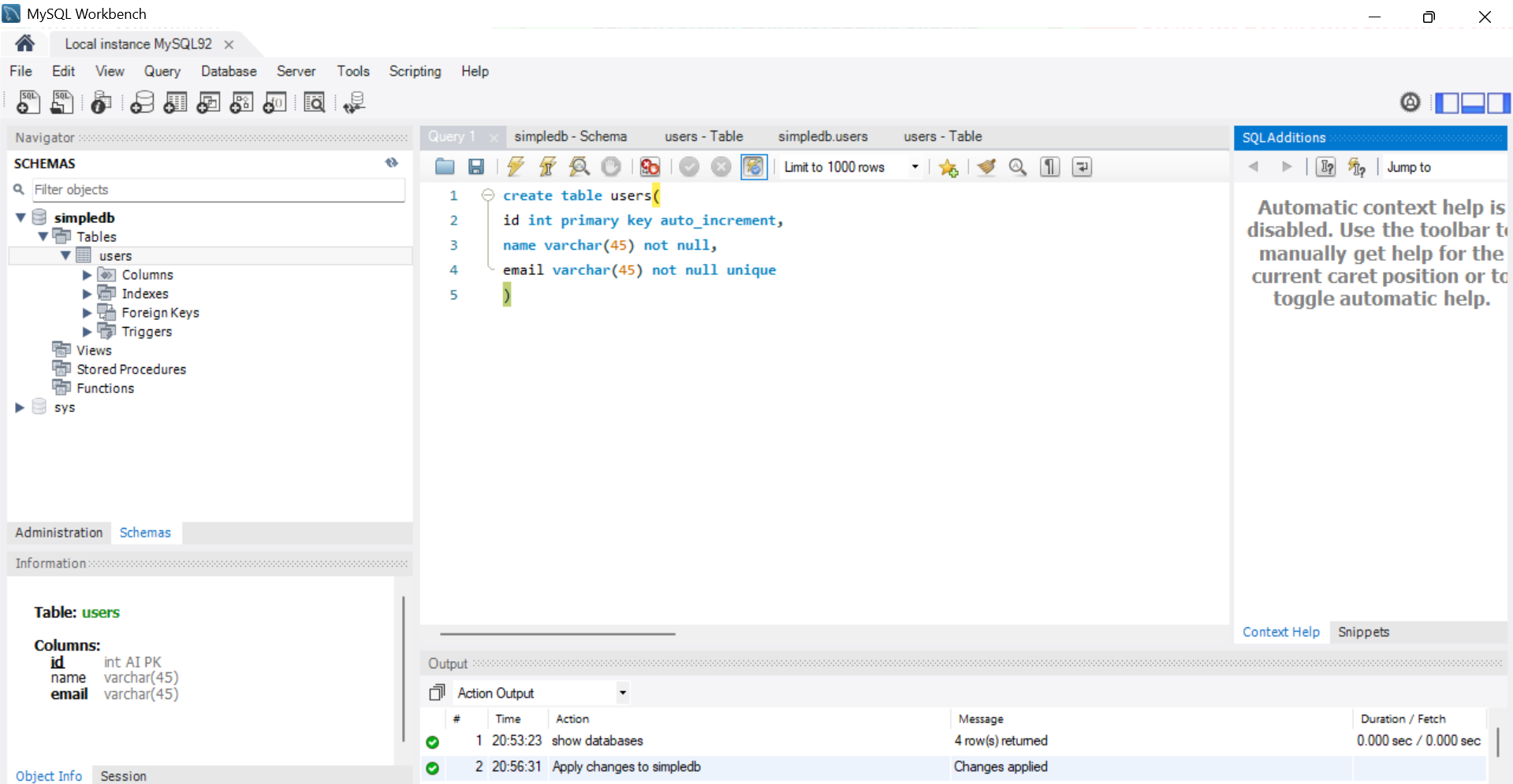
• Схема SYS входит в состав MySQL Server 5.7+ и используется MySQL Workbench. Для MySQL 5.6 Workbench может установить собственную версию схемы.

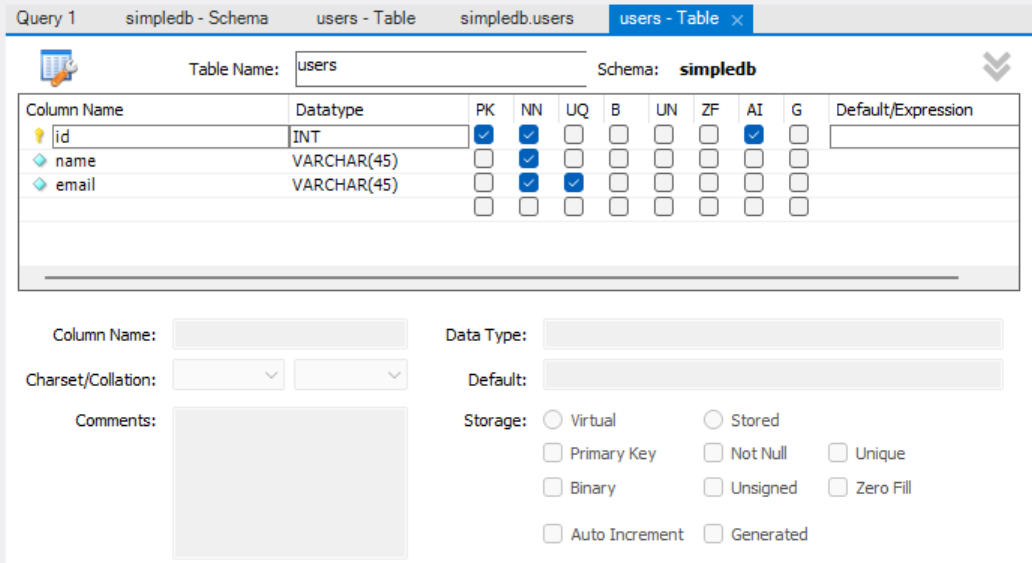
• Размер сохранённых запросов (digest) зависит от конфигурации сервера.

Эта настройка важна для работы отчётов из предыдущего пункта (Performance Reports) и для глубокого анализа производительности сервера.

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

Создадим новую базу данных Simpledb, после создадим в ней несколько таблиц:





**Задание 3**

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

Запрос:

CREATE TABLE `users` (

`id` int NOT NULL AUTO\_INCREMENT,

`name` varchar(45) NOT NULL,

`email` varchar(45) NOT NULL,

PRIMARY KEY (`id`),

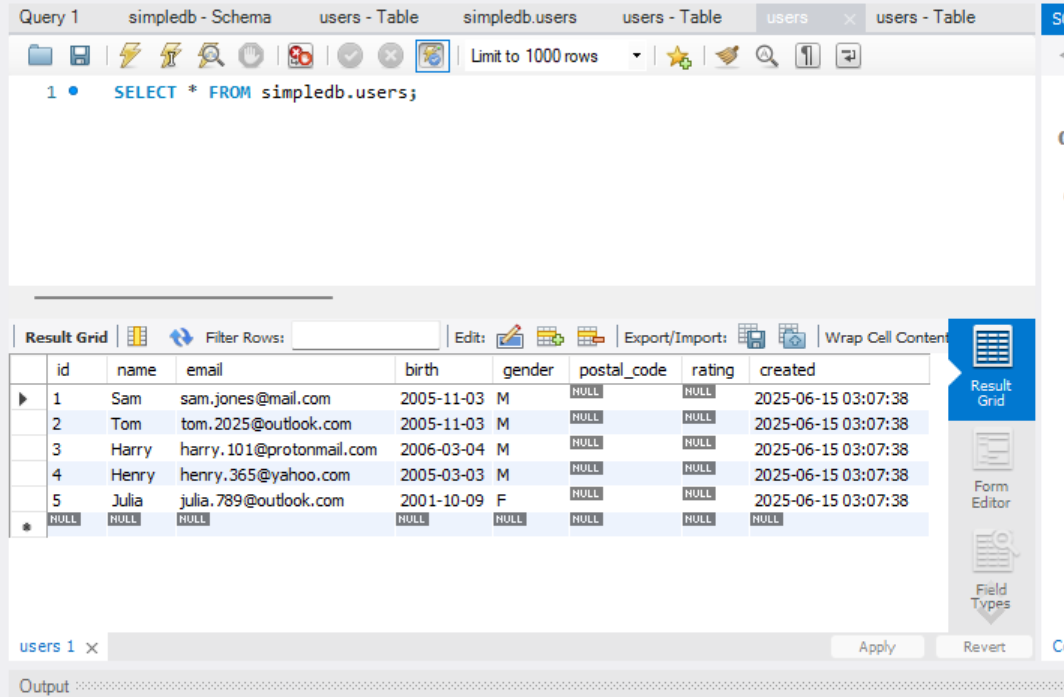
UNIQUE KEY `email` (`email`)

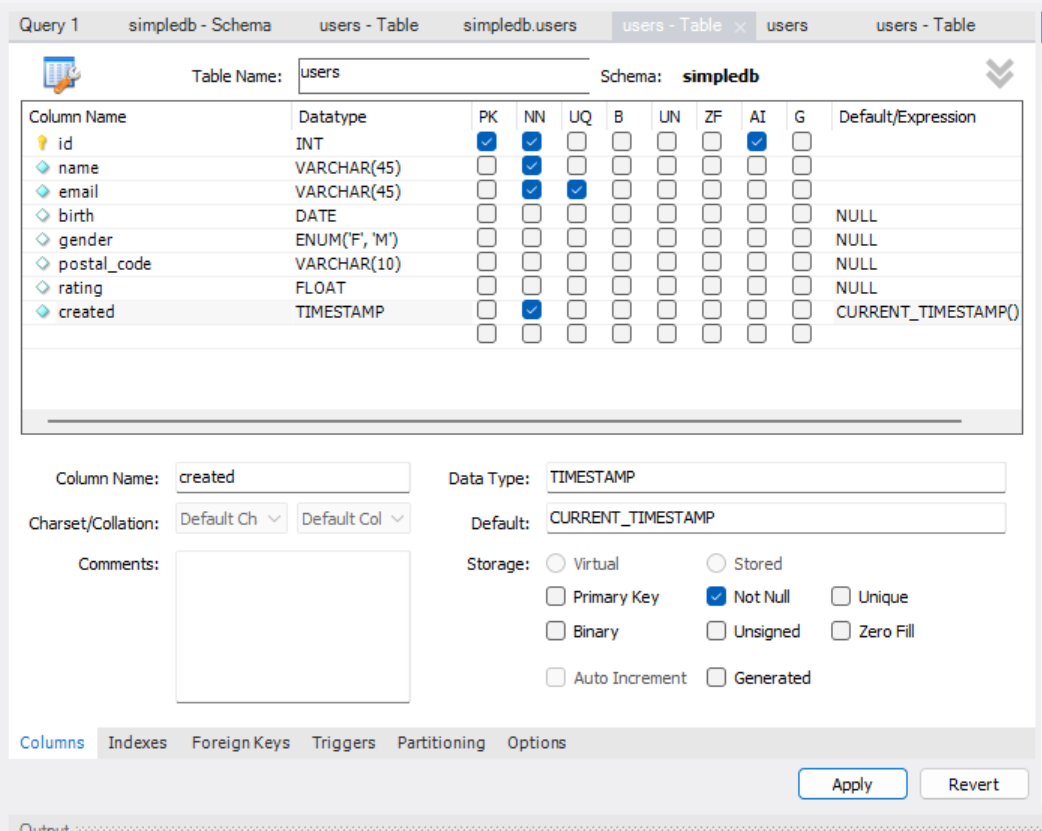
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb3;

**Задание 4**

Добавьте несколько примеров-записей в созданную таблицу.

Заполним таблицу, после добавим новые атрибуты, снова заполним таблицу.





Запрос:

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)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

**Задание 5**

Дополните таблицу users; 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; Определите самостоятельно какие поля могут быть NULL и будут иметь это значение в качестве значения по умолчанию.

Дополнили таблицу в задании 4.

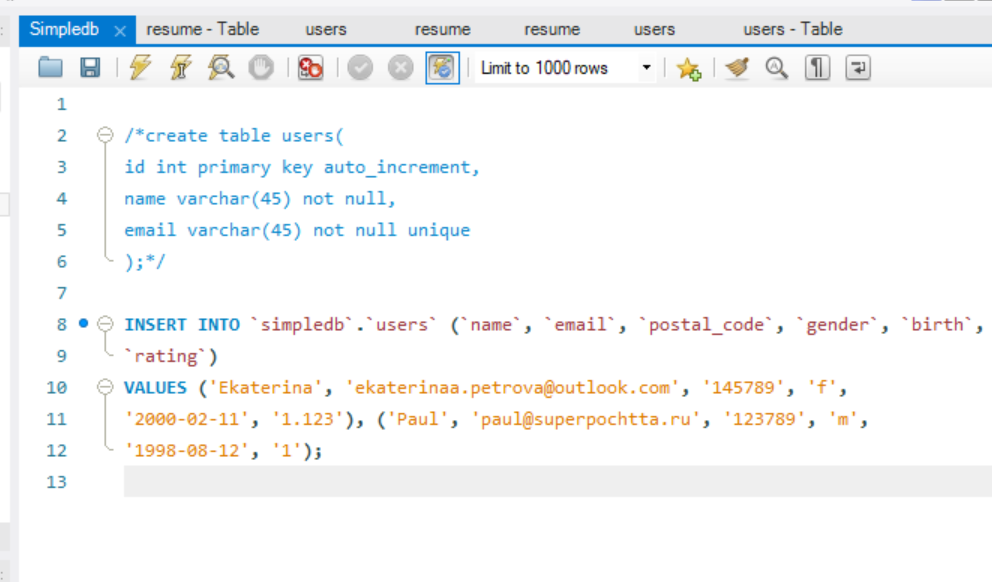
TIMESTAMP — тип данных для хранения даты и времени. CURRENT\_TIMESTAMP() — функция, которая возвращает текущее время сервера. Запись created TIMESTAMP DEFAULT CURRENT\_TIMESTAMP() означает, что при добавлении новой строки в таблицу в поле created автоматически будет записано текущее время.

Обязательными обычно являются основные поля — такие как имя, фамилия, логин и пароль, так как без них невозможно создать и идентифицировать пользователя в системе. А вот персональные данные, такие как пол, дата рождения, адрес или телефон, не каждый пользователь захочет указывать. Поэтому такие поля логично сделать допускающими значение NULL по умолчанию — это позволяет сохранить анонимность.

**Задание 6**

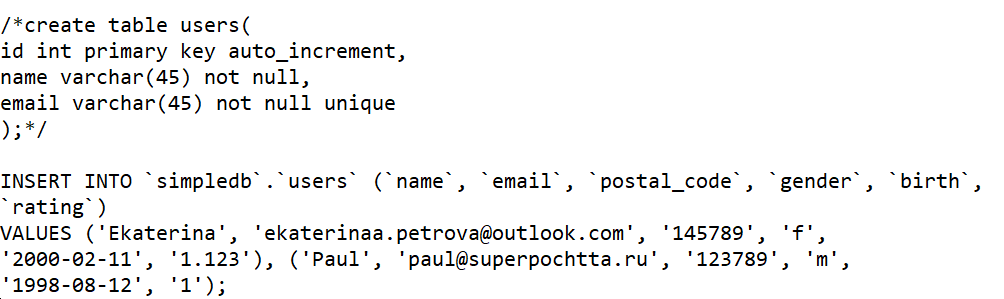
Дополните таблицу, добавив данные двумя способами:

* с помощью внесения данных вручную;
* с помощью выполнения SQL-запросов;



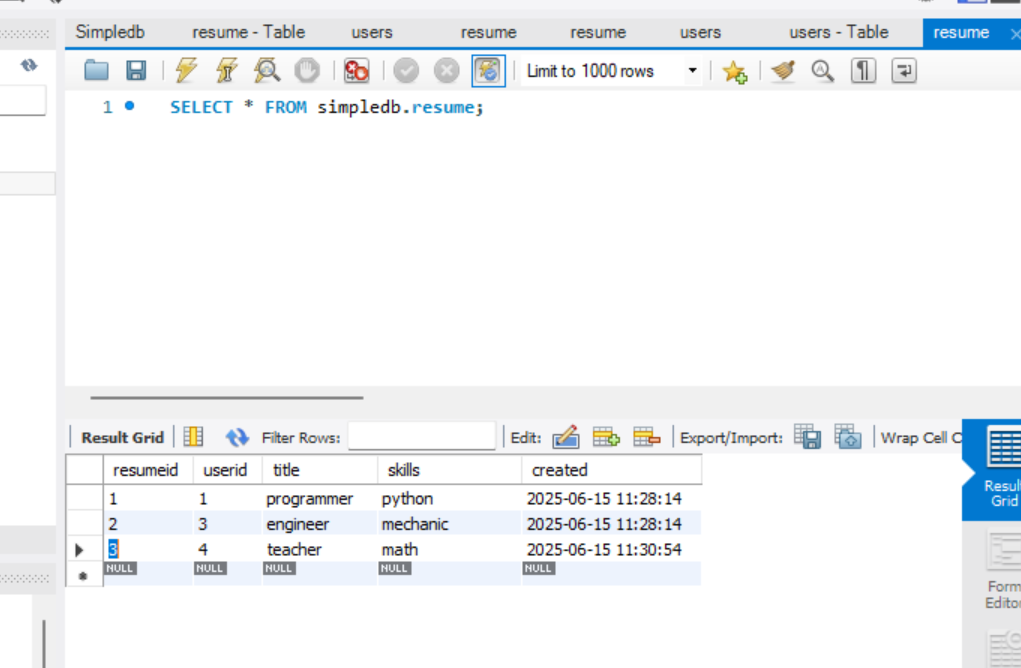
**Задание 7**

С помощью кнопки “Export recordset to external file” получите файл с SQL-запросами.



**Задание 8**

Создайте еще одну таблицу с названием resume.



Запрос:

CREATE TABLE `resume` (

`resumeid` int NOT NULL AUTO\_INCREMENT,

`userid` int NOT NULL,

`title` varchar(100) NOT NULL,

`skills` text,

`created` timestamp NULL DEFAULT CURRENT\_TIMESTAMP,

PRIMARY KEY (`resumeid`),

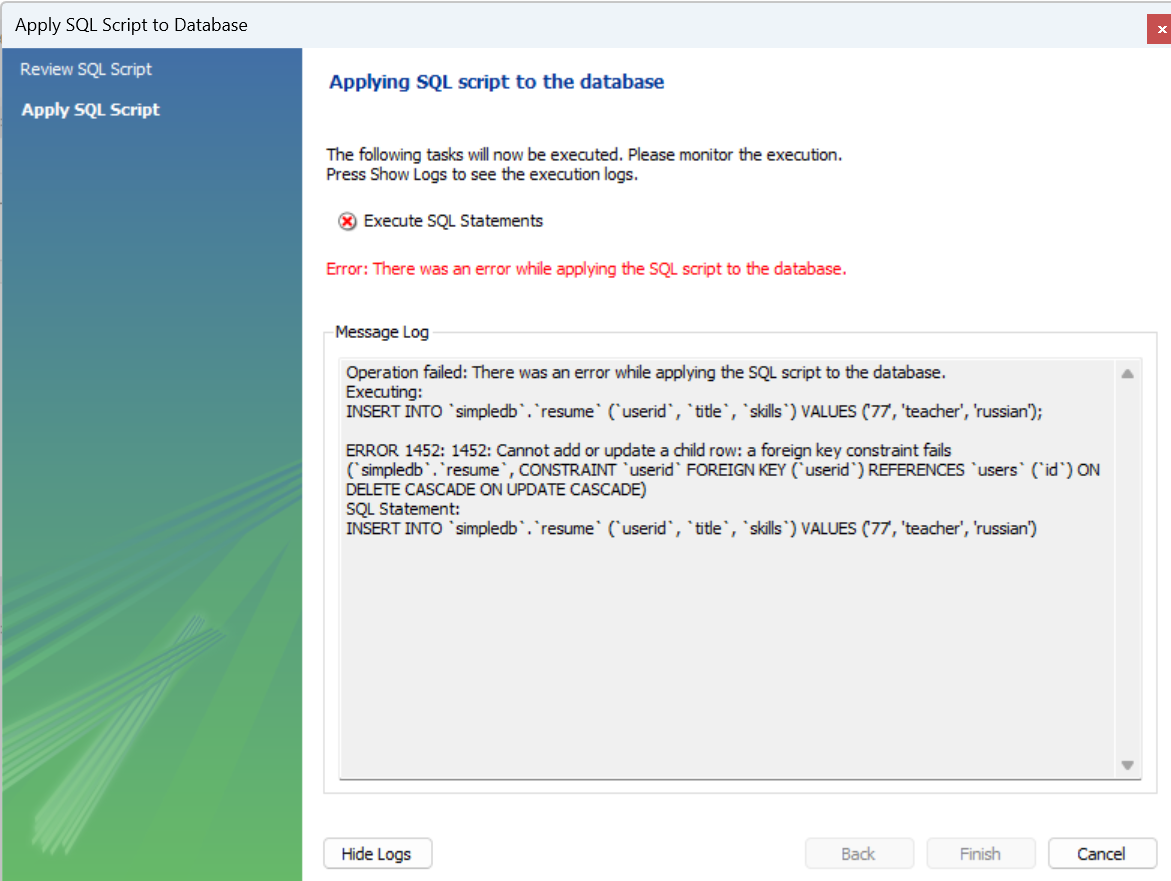
KEY `userid\_idx` (`userid`),

CONSTRAINT `userid` FOREIGN KEY (`userid`) REFERENCES `users` (`id`) ON DELETE CASCADE ON UPDATE CASCADE

) ENGINE=InnoDB AUTO\_INCREMENT=9 DEFAULT CHARSET=utf8mb3;

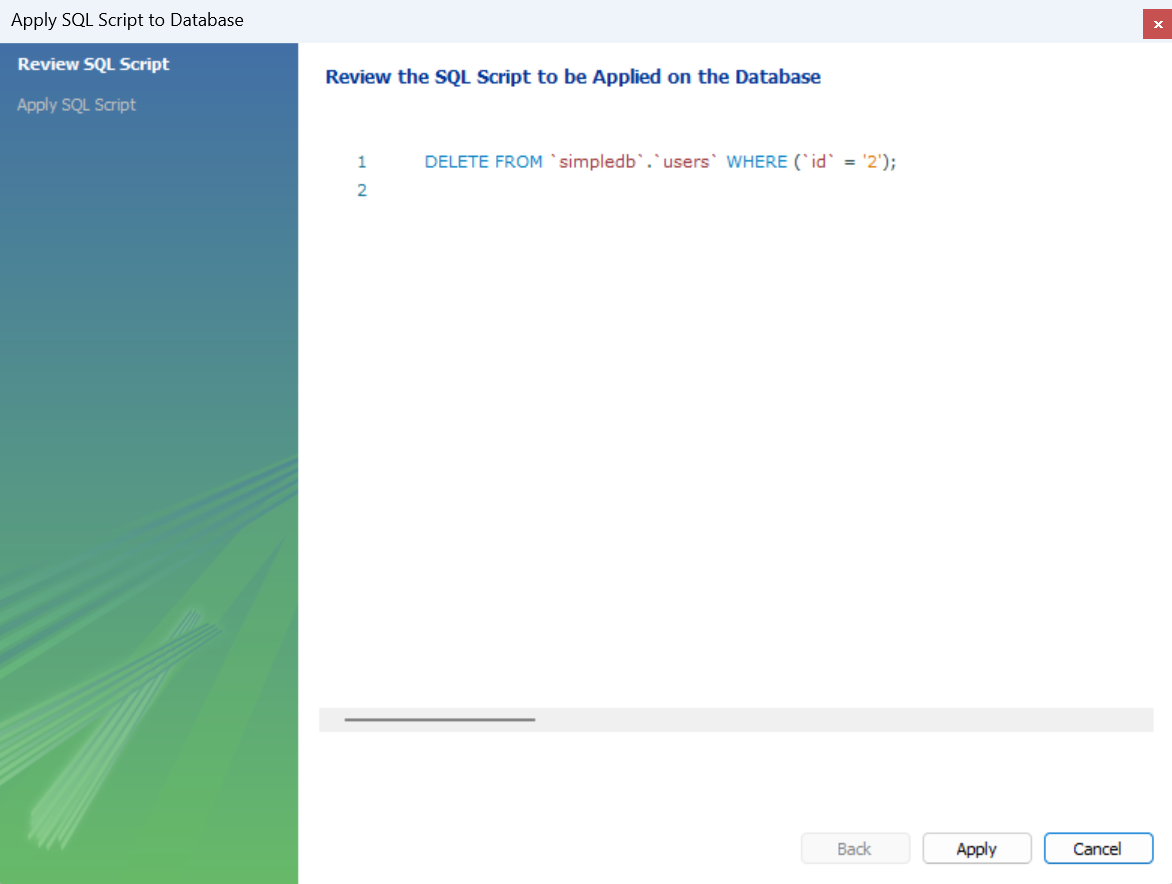
**Задание 9**

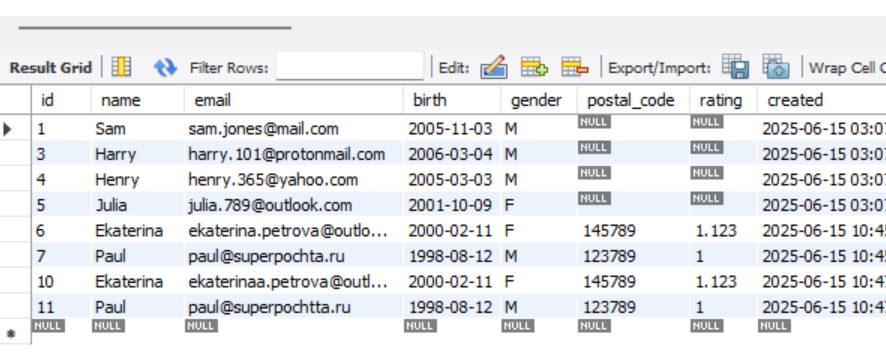
Наполните вторую таблицу данными так, чтобы в ней была информация хотя бы о нескольких резюме, связанных с уже существующими пользователями из таблицы users. Попробуйте добавить в таблицу resume строчку с userid несуществующего пользователя (такого пользователя, id которого нет в таблице users).

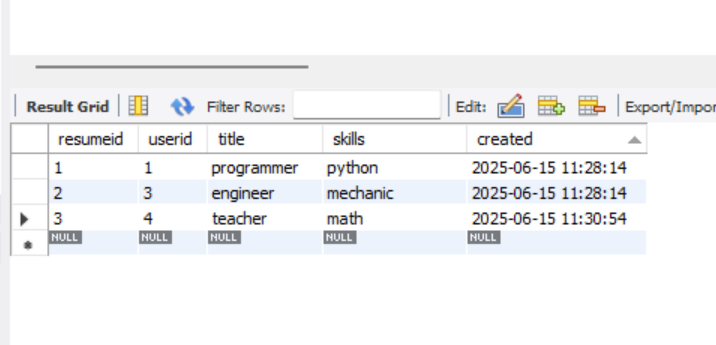


**Задание 10**

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







1. Что произойдет со связанными сущностями в таблице resume?

При удалении пользователя из таблицы users все связанные записи в таблице resume автоматически удалятся благодаря настройке ON DELETE CASCADE во внешнем ключе.

2. Что произойдет при изменении id пользователя в таблице users?

Если изменить id пользователя, то все соответствующие userid в таблице resume автоматически обновятся из-за настройки ON UPDATE CASCADE.