

Лабораторная работа №1

Задание 1.

Раздел “Instance” («Экземпляр БД»)

1. Раздел “Startup / Shutdown”. Позволяет управлять состоянием сервера MySQL. В данном разделе можно запустить или остановить сервер, а также выполнить его перезапуск.
2. Раздел “Server Status”. Отображает текущее состояние сервера, включая основную информацию о его работе:
 - a. Время работы сервера (uptime).
 - b. Количество активных соединений.
 - c. Нагрузка на процессор и использование памяти.
 - d. Число выполняемых запросов.
3. Раздел “Options File”. Предоставляет интерфейс для редактирования конфигурационного файла MySQL (my.cnf или my.ini). Позволяет изменять параметры сервера, такие как:
 - a. Настройки кэширования и памяти.
 - b. Логи сервера.
 - c. Параметры соединений.
4. Раздел “Server Logs”. В этом разделе можно просматривать журналы работы сервера, которые помогают диагностировать проблемы. Включает:
 - a. Журнал ошибок MySQL.
 - b. Журнал запросов.
 - c. Журнал общих событий.
5. Раздел “Users and Privileges”. Позволяет управлять пользователями MySQL, их привилегиями и ролями. Основные возможности:
 - a. Создание, удаление и изменение учетных записей.
 - b. Настройка привилегий пользователей.
 - c. Управление паролями.

Раздел “Performance” («Производительность»)

1. Раздел “Performance Dashboard”. Графический интерфейс для мониторинга ключевых показателей производительности MySQL. Включает:
 - a. Количество соединений.
 - b. Среднее время выполнения запросов.
 - c. Использование процессора и памяти.
 - d. Количество заблокированных запросов.
2. Раздел “Performance Reports”. Включает готовые отчеты по основным метрикам MySQL, которые помогают выявлять проблемы с производительностью. Примеры отчетов:

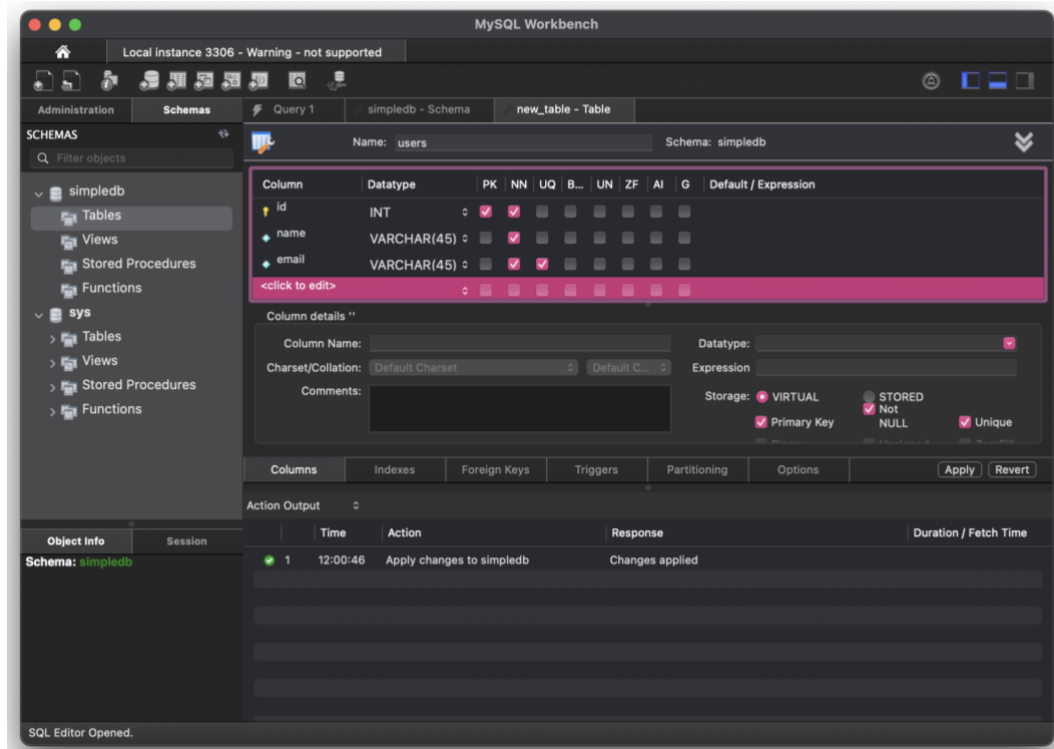
- a. Анализ медленных запросов.
- b. Частота блокировок таблиц.
- c. Использование индексов.
- 3. Раздел “Performance Schema Setup”. Позволяет включить и настроить Performance Schema – встроенный механизм MySQL для сбора детализированной информации о выполнении запросов и работе сервера.
- 4. Раздел “Table I/O and Index Statistics”. Предоставляет статистику по использованию таблиц и индексов, включая:
 - a. Частоту чтения и записи в таблицы.
 - b. Частоту использования индексов.
 - c. Запросы, не использующие индексы.
- 5. Раздел “Memory Usage”. Позволяет отслеживать распределение памяти между различными компонентами MySQL, такими как:
 - a. Кэши запросов.
 - b. Буферы соединений.
 - c. Используемые пуллы памяти.

Раздел “Management” («Управление»)

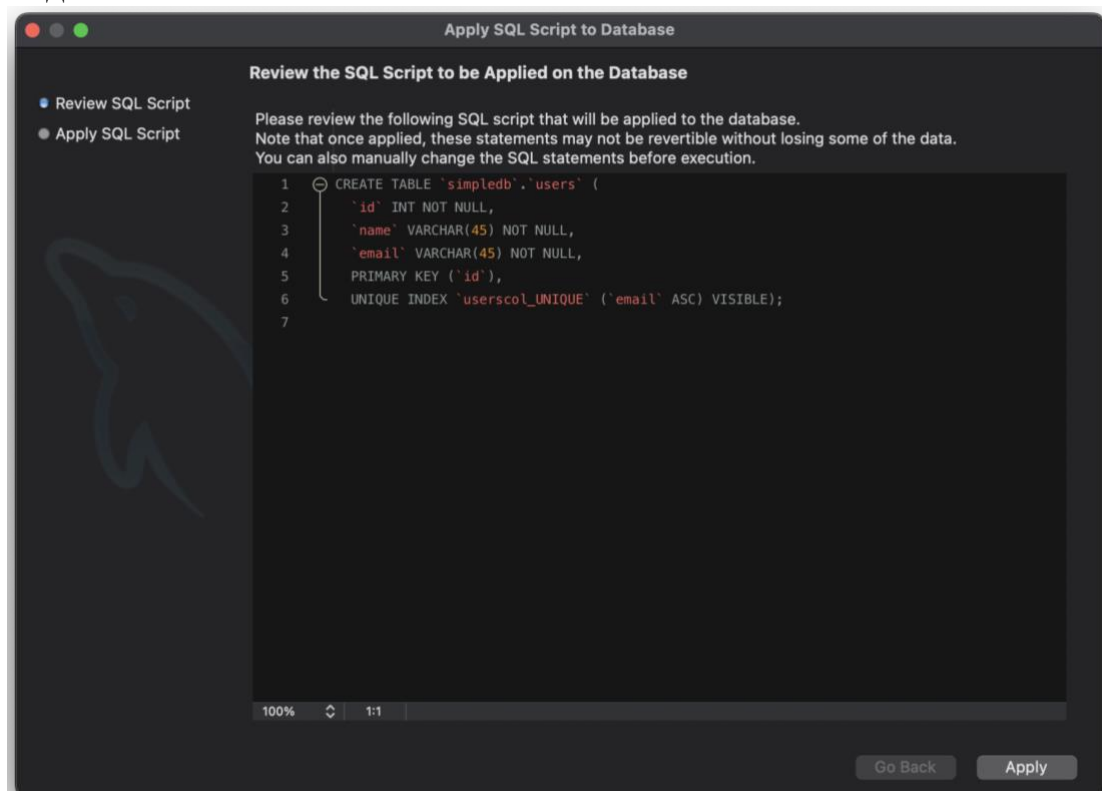
- 1. Client Connections – позволяет просматривать и управлять активными подключениями к серверу. Включает:
 - a. Список текущих пользователей и их соединений.
 - b. Выполняемые запросы и их статус.
 - c. Возможность завершения соединений или блокировки пользователей.
- 2. Users and Privileges – инструмент для управления учетными записями MySQL. Позволяет:
 - a. Создавать, изменять и удалять пользователей.
 - b. Назначать права доступа к базам данных.
 - c. Управлять паролями и методами аутентификации.
- 3. Schemas – предоставляет список всех баз данных (схем), доступных на сервере. В этом разделе можно:
 - a. Просматривать структуру таблиц, представлений и процедур.
 - b. Выполнять SQL-запросы для управления данными.
- 4. Data Export – инструмент для резервного копирования данных. Позволяет:
 - a. Экспортировать базы данных или отдельные таблицы.
 - b. Создавать дампы SQL-файлов для последующего восстановления.
 - c. Настраивать параметры экспорта (например, структура, данные, триггеры).
- 5. Data Import/Restore – используется для восстановления данных из резервных копий. Включает:
 - a. Импорт данных из SQL-файлов.
 - b. Восстановление структуры и содержимого баз данных.

6. Server Logs – предоставляет доступ к журналам сервера MySQL, включая:
- Ошибки, запросы и общие события.
 - Возможность анализа проблем и производительности.

Задание 2.



Задание 3.



Задание 4.

16

13:22:30

SELECT * FROM simplifiedb.users LIMIT...

3 row(s) returned

0.00077 sec / 0.0000...

	id	name	email	gender	
	1	Malcolm	malcolm@gmail.com	m	
	2	Scott	scott@outlook.com	m	
	3	Jermaine	jermaine.lamar@yahoo.com	m	

Задание 5.

Тип столбца **TIMESTAMP** обеспечивает тип представления данных, который можно использовать для автоматической записи текущих даты и времени при выполнении операций **INSERT** или **UPDATE**. При наличии нескольких столбцов типа **TIMESTAMP** только первый из них обновляется автоматически.

Если пользователь, минимально делится персональными данными, то некоторые поля в базе данных могут быть **NULL** по умолчанию.

```
CREATE TABLE `users` (  
  `id` int NOT NULL,  
  `name` varchar(50) NOT NULL,  
  `email` varchar(45) NOT NULL,  
  `gender` enum('M','F') DEFAULT NULL,  
  `bday` date DEFAULT NULL,  
  `postal_code` varchar(10) DEFAULT NULL,  
  `rating` float DEFAULT NULL,  
  `created` varchar(45) NOT NULL,  
  PRIMARY KEY (`id`),  
  UNIQUE KEY `userscol_UNIQUE` (`email`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb3;
```

Задание 6.

	id	name	email	postal_code	gender	bday	created	rating
	1	Malcolm	malcolm@gmail.com	123789	M	1992-01-19	2025-02-19	1
	2	Scott	scott@outlook.com	124759	M	1984-01-30	2025-02-19	1.123
	3	Jermaine	jermaine.lamar@yahoo.com	145789	M	1985-01-28	2025-02-19	1.123
	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Задание 7.

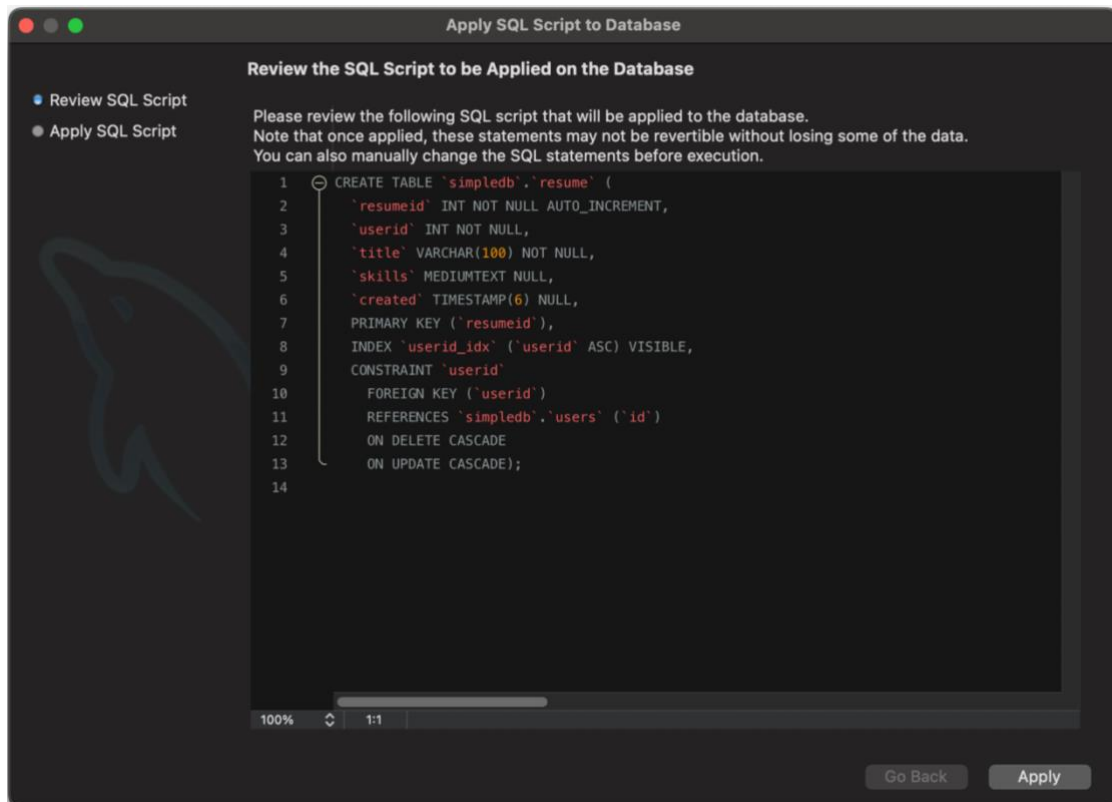
```
/*  
-- Query: SELECT * FROM simplifiedb.users  
LIMIT 0, 1000
```

-- Date: 2025-02-19 12:09

*/

```
INSERT INTO `` (`id`,`name`,`email`,`
postal_code`,`gender`,`bday`,`created`,`rating`) VALUES
(1,'Malcolm','malcolm@gmail.com','123789','M','1992-01-19','2025-02-19',1);
INSERT INTO `` (`id`,`name`,`email`,`
postal_code`,`gender`,`bday`,`created`,`rating`) VALUES
(2,'Scott','scott@outlook.com','124759','M','1984-01-30','2025-02-19',1.123);
INSERT INTO `` (`id`,`name`,`email`,`
postal_code`,`gender`,`bday`,`created`,`rating`) VALUES
(3,'Jermaine','jermaine.lamar@yahoo.com','145789','M','1985-01-28','2025-02-
19',1.123);
```

Задание 8.



Задание 9.

Я думаю у одного пользователя может быть несколько резюме. Проверил в таблице введя на один id несколько заголовков и навыков.

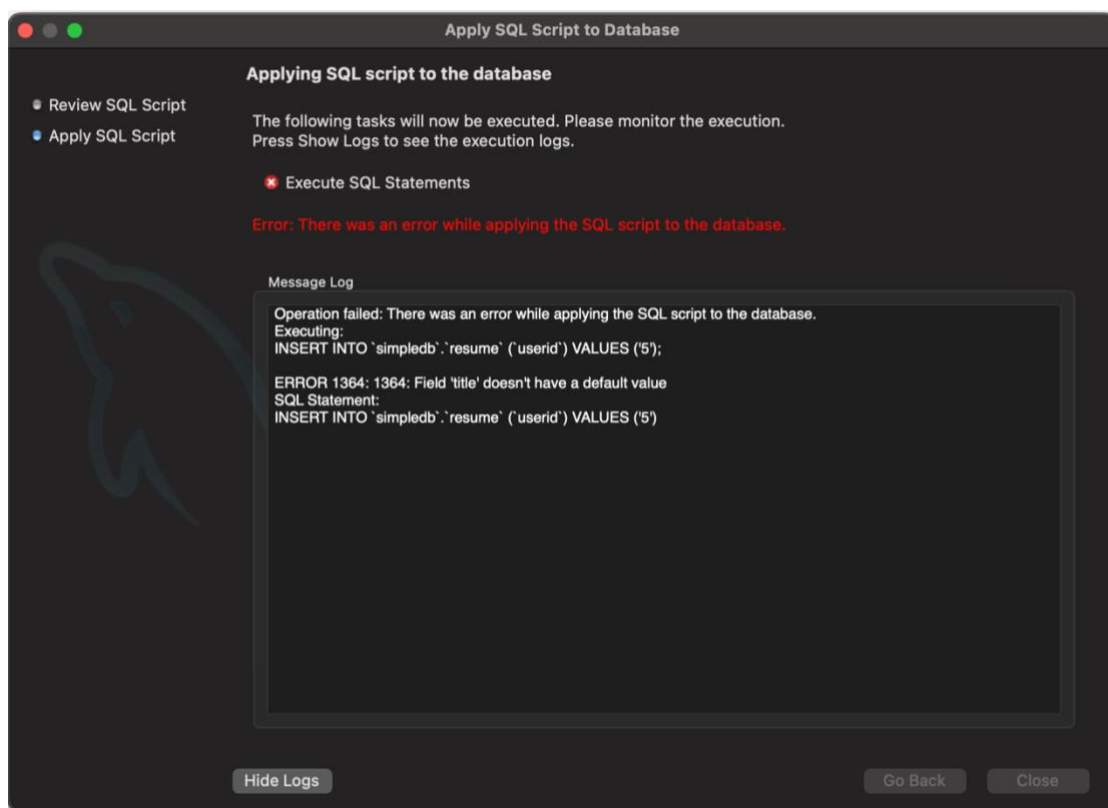
```
-- Query: SELECT * FROM simpledb.resume
LIMIT 0, 1000
```

-- Date: 2025-02-19 15:32

*/

```
INSERT INTO `` (`resumeid`,`userid`,`title`,`skills`,`created`) VALUES
(3,1,'composer','Multi-instrumentalist, Hard word','2025-02-19 00:00:00.000000');
INSERT INTO `` (`resumeid`,`userid`,`title`,`skills`,`created`) VALUES
(4,2,'basketball player','Athleticism, stamina','2025-02-19 00:00:00.000000');
INSERT INTO `` (`resumeid`,`userid`,`title`,`skills`,`created`) VALUES
(5,3,'music producer','Sensitive, relevance','2025-02-19 00:00:00.000000');
INSERT INTO `` (`resumeid`,`userid`,`title`,`skills`,`created`) VALUES
(6,2,'composer','Lyricism','2025-02-19 00:00:00.000000');
```

При добавлении в таблицу строки с несуществующим пользователем выдается ошибка.



Задание 10.

