Федеральное государственное бюджетное образовательное учреждение высшего образования «Российский государственный педагогический университет им. А. И. Герцена

ОТЧЕТ

по дисциплине «Базы данных»

**Лабораторная работа №1**

**«Начало работы с MySQL. MySQL Workbench»**

**Выполнил:**

студент гр. 2об\_ИВТ-2/22

                                      \_\_\_\_\_\_\_\_\_\_\_\_\_\_Маковеев. Н. В.

Санкт-Петербург, 2024 г.

1. **Задание**

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 MySQL Workbench предоставляет информацию о текущих подключениях к серверу MySQL со стороны клиентов. В этом разделе отображается список активных соединений, их идентификаторы, IP-адреса, пользователи, которые подключены к серверу, а также другие параметры.
   1. Можно видеть, сколько клиентов в данный момент подключено к серверу
   2. Возможность смотреть какие запросы выполняются
   3. Есть ли какие-то проблемы с соединениями
   4. Можно также отключать ненужные соединения
   5. Устанавливать ограничения на количество одновременных соединений
   6. онтролировать доступ и безопасность сервера
3. Раздел "Users and Privileges" в MySQL Workbench предоставляет возможность администраторам баз данных управлять пользователями и их привилегиями на сервере MySQL. В этом разделе отображается список всех пользователей, имеющих доступ к серверу, их привилегии, разрешения и другие настройки.

Этот раздел позволяет администраторам баз данных создавать, удалять и изменять учетные записи пользователей, назначать различные уровни доступа к базам данных и таблицам, управлять паролями и правами доступа к данным. Администраторы также могут назначать специфические привилегии пользователям, такие как SELECT, INSERT, UPDATE, DELETE, CREATE, DROP и др.

1. Раздел "Status and System Variables" в MySQL Workbench предоставляет информацию о текущем состоянии сервера MySQL, общую статистику использования ресурсов, параметры системы и локальных переменных. Функции этого раздела можно описать следующим образом

Отображение текущего состояния сервера MySQL: раздел показывает информацию о загрузке сервера, количестве активных соединений, использовании памяти, CPU и других системных ресурсах.

* 1. Отслеживание производительности сервера: администратор может просматривать статистику по запросам, времени выполнения запросов, использованию индексов и т.д. для оценки производительности сервера MySQL.
  2. Просмотр локальных переменных и параметров системы: раздел позволяет администратору просматривать и изменять системные переменные и настройки MySQL, такие как размер буфера, максимальное количество соединений, параметры кэширования и т.д.
  3. Мониторинг активности и потребления ресурсов: с помощью раздела "Status and System Variables" можно отслеживать активность запросов, использование памяти, процессора и других ресурсов сервера для оптимизации работы и выявления проблем.
  4. Решение проблем и улучшение производительности: администратор может использовать информацию из раздела для анализа и решения проблем с производительностью, оптимизации конфигурации сервера и настройки параметров для оптимальной работы MySQL.

1. Раздел "Data Export" в MySQL Workbench предоставляет возможность администраторам баз данных экспортировать данные из базы данных MySQL в различные форматы для последующего использования или обмена информацией. Функции этого раздела можно описать следующим образом:
   1. Экспорт данных в различные форматы: раздел "Data Export" позволяет администраторам выбирать формат экспортирования данных, такие как CSV, SQL, XML, JSON, Excel и другие, в зависимости от потребностей проекта.
   2. Выбор конкретных таблиц для экспорта: администратор может выбрать определенные таблицы или множество таблиц для экспорта, что позволяет гибко управлять данными, которые необходимо экспортировать.
   3. Настройка параметров экспорта: раздел предоставляет возможность настройки различных параметров экспорта, таких как формат даты, кодировка символов, разделители и другие опции для получения данных в нужном формате.
   4. Планирование и автоматизация экспорта данных: администратор может создавать планы экспорта данных и настраивать автоматическое выполнение экспорта по расписанию, что облегчает процесс обмена информацией и делает его более эффективным.
   5. Экспорт структуры базы данных: помимо данных, раздел "Data Export" также позволяет экспортировать структуру базы данных, включая таблицы, индексы, ограничения и другие объекты, что удобно при создании резервных копий или копировании структуры на другой сервер.
2. Раздел "Data Export" в MySQL Workbench предоставляет возможность администраторам баз данных экспортировать данные из базы данных MySQL в различные форматы для последующего использования или обмена информацией. Функции этого раздела можно описать следующим образом:
   * + - 1. Экспорт данных в различные форматы: раздел "Data Export" позволяет администраторам выбирать формат экспортирования данных, такие как CSV, SQL, XML, JSON, Excel и другие, в зависимости от потребностей проекта.
         2. Выбор конкретных таблиц для экспорта: администратор может выбрать определенные таблицы или множество таблиц для экспорта, что позволяет гибко управлять данными, которые необходимо экспортировать.
         3. Настройка параметров экспорта: раздел предоставляет возможность настройки различных параметров экспорта, таких как формат даты, кодировка символов, разделители и другие опции для получения данных в нужном формате.
         4. Планирование и автоматизация экспорта данных: администратор может создавать планы экспорта данных и настраивать автоматическое выполнение экспорта по расписанию, что облегчает процесс обмена информацией и делает его более эффективным.
         5. Экспорт структуры базы данных: помимо данных, раздел "Data Export" также позволяет экспортировать структуру базы данных, включая таблицы, индексы, ограничения и другие объекты, что удобно при создании резервных копий или копировании структуры на другой сервер.
3. Раздел "Data Import" в MySQL Workbench позволяет администраторам баз данных импортировать данные в базу данных MySQL из различных источников. Функции этого раздела можно кратко описать следующим образом:
   1. Импорт данных из различных источников: раздел "Data Import" поддерживает импорт данных из различных источников, таких как файлы CSV, SQL, Excel, XML, JSON и другие форматы, что обеспечивает гибкость в импорте информации.
   2. Выбор данных для импорта: администратор может выбрать конкретные данные или таблицы для импорта, что позволяет точечно управлять процессом импорта и избегать импорта ненужных данных.
   3. Настройка параметров импорта: раздел предоставляет возможность настройки различных параметров импорта, таких как формат даты, кодировка символов, разделители и другие опции для корректного импорта данных.
   4. Мониторинг процесса импорта: администратор может мониторить процесс импорта данных и получать информацию о успешности выполнения операции, что облегчает контроль и отладку процесса импорта.
   5. Автоматизация и планирование импорта данных: раздел позволяет создавать планы импорта данных и настраивать автоматическое выполнение импорта по расписанию, что повышает эффективность и удобство работы с большим объемом данных.
4. Задание 2.

Скриншот создания таблицы.

Изображение выглядит как текст, снимок экрана, программное обеспечение, число

Автоматически созданное описание

Код создания таблицы.

CREATE TABLE `simpledb`.`users` (

`id` INT NOT NULL,

`name` VARCHAR(45) NOT NULL,

`email` VARCHAR(45) NOT NULL,

PRIMARY KEY (`id`),

UNIQUE INDEX `email\_UNIQUE` (`email` ASC) VISIBLE);

3 Задание.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Запрос для вставки:

INSERT INTO users VALUES(1,"Nikita", "@email"),(2,"Kirill", "123@mail"), (3,"Nikolay", "dbdb@dbdb")

Запрос для обновления.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Запрос для обновления данных:

UPDATE users

SET name = "foo"

WHERE id =2;

Изображение выглядит как текст, снимок экрана, программное обеспечение, число

Автоматически созданное описание

Запрос изменяющий таблицу:

ALTER TABLE `simpledb`.`users`

CHANGE COLUMN `high` `high` TINYINT(5) NOT NULL ;

Задание 5.

Изображение выглядит как текст, снимок экрана, чек, линия

Автоматически созданное описание

Код изменения таблицы:

ALTER TABLE `simpledb`.`users`

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 NULL DEFAULT CURRENT\_TIMESTAMP() AFTER `rating`,

ADD COLUMN `userscol` VARCHAR(45) NULL AFTER `created`,

CHANGE COLUMN `name` `name` VARCHAR(50) NULL ,

CHANGE COLUMN `email` `email` VARCHAR(45) NULL ,

CHANGE COLUMN `high` `gender` ENUM('M', 'F') NULL ,

DROP INDEX `email\_UNIQUE` ;

Задание 6.

Добавление пользователей с помощью SQL запроса.

Изображение выглядит как снимок экрана, текст, Мультимедийное программное обеспечение

Автоматически созданное описание

Пользователь maks был введен вручную

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Вот сгенерированный запрос

INSERT INTO `simpledb`.`users` (`name`, `email`, `gender`, `bday`, `postal\_code`, `rating`) VALUES ('maks', 'qwerty@qwerty', 'M', '2000-01-01', '111111', '1.46');

Задание 7.

Ниже переведен код из сгенерированного файла.

/\*

-- Query: SELECT \* FROM simpledb.users

LIMIT 0, 1000

-- Date: 2024-04-11 00:38

\*/

INSERT INTO `` (`id`,`name`,`email`,`gender`,`bday`,`postal\_code`,`rating`,`created`,`userscol`) VALUES (1,'Nikita','@email','',NULL,NULL,NULL,'2024-04-11 00:28:38',NULL);

INSERT INTO `` (`id`,`name`,`email`,`gender`,`bday`,`postal\_code`,`rating`,`created`,`userscol`) VALUES (2,'foo','123@mail','',NULL,NULL,NULL,'2024-04-11 00:28:38',NULL);

INSERT INTO `` (`id`,`name`,`email`,`gender`,`bday`,`postal\_code`,`rating`,`created`,`userscol`) VALUES (3,'Nikolay','dbdb@dbdb','',NULL,NULL,NULL,'2024-04-11 00:28:38',NULL);

INSERT INTO `` (`id`,`name`,`email`,`gender`,`bday`,`postal\_code`,`rating`,`created`,`userscol`) VALUES (4,'Ekaterina','ekaterina.petrova@outlook.com','F','2000-02-11','145789',1.123,'2024-04-11 00:32:12',NULL);

INSERT INTO `` (`id`,`name`,`email`,`gender`,`bday`,`postal\_code`,`rating`,`created`,`userscol`) VALUES (5,'Ekaterina','ekaterina.petrova@outlook.com','F','2000-02-11','145789',1.123,'2024-04-11 00:32:14',NULL);

INSERT INTO `` (`id`,`name`,`email`,`gender`,`bday`,`postal\_code`,`rating`,`created`,`userscol`) VALUES (6,'Pяaul','paul@superpochta.ru','M','1998-08-12','123789',1,'2024-04-11 00:32:33',NULL);

INSERT INTO `` (`id`,`name`,`email`,`gender`,`bday`,`postal\_code`,`rating`,`created`,`userscol`) VALUES (7,'maks','qwerty@qwerty','M','2000-01-01','111111',1.46,'2024-04-11 00:35:17',NULL);

Задание 8.

Созданная таблица resume

Изображение выглядит как текст, снимок экрана, линия, число

Автоматически созданное описание

Сгенерированный код:

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);

Cascade работает так- если мы удалим данные из таблицы users, данные с соответствующим FK удалятся тоже.

Задание 9.

Вставленные строки. Начинаются с 4, тк первые 3 строки были вставлены с ошибкой.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Код вставки строк:

INSERT INTO `simpledb`.`resume` (`-userid`, `title`, `skills`) VALUES ('1', '\"qwe\"', '\"good boy\"');

INSERT INTO `simpledb`.`resume` (`-userid`, `title`, `skills`) VALUES ('3', '\"man with man\"', '\"best dota player\"');

INSERT INTO `simpledb`.`resume` (`-userid`, `title`, `skills`) VALUES ('4', '\"realy cool guy\"', '\"great sleepy\"');

Если ввести несуществующи user\_id то mysql ругается

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание

Максимум можно вставить 2 147 483 647 Разных резюме, не уверен, что можно делать ID отрицательным.

Экспортируемый SQL файл:

/\*

-- Query: SELECT \* FROM simpledb.resume

LIMIT 0, 1000

-- Date: 2024-04-11 00:57

\*/

INSERT INTO `` (`resumeid`,`-userid`,`title`,`skills`,`created`) VALUES (4,1,'\"qwe\"','\"good boy\"','2024-04-11 00:51:21');

INSERT INTO `` (`resumeid`,`-userid`,`title`,`skills`,`created`) VALUES (5,3,'\"man with man\"','\"best dota player\"','2024-04-11 00:51:21');

INSERT INTO `` (`resumeid`,`-userid`,`title`,`skills`,`created`) VALUES (6,4,'\"realy cool guy\"','\"great sleepy\"','2024-04-11 00:51:21');

10 задание.

1. Удаление пользователя.

DELETE FROM users WHERE id =1;

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

Видно, что из таблицы resume запись тоже удалилась.

1. Изменение id в таблице users.

UPDATE users Set id =101 WHERE name = "Nikolay";

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

Когда меняем id в таблице users -> user\_id тоже меняется.