Отчёт по лабораторной работе №1

Задание 1

Раздел "Performance"

Предназначен для отслеживания и анализа производительности базы данных, а также оптимизации работы MySQL-серверов.

- Инструменты мониторинга производительности (отображение загрузки CPU, памяти, сети, а также данных о текущих подключениях, запросах и простое сервера).

- Анализ выполнения отдельных запросов для выявления медленных или ресурсоёмких операций.

- Возможность изменения конфигурации сервера для повышения производительности (настройка буферов, кэшей и других параметров через "Performance Schema setup").

- Широкий набор инструментов для сбора статистики по затратам ресурсов на выполнение задач.

Раздел "Instance"

Позволяет пользователям настраивать параметры подключения, конфигурацию сервера, управлять учётными записями и другими аспектами взаимодействия с базами данных.

- Настройка метода подключения к серверу (например, указание адреса, порта, логина и пароля для доступа к БД).

- Управление пользователями и удалённым доступом.

- Настройка сервера вручную или через конфигурационный файл.

- Включён серверный логгер для записи событий.

Задания 2 и 3

SQL-запрос для создания таблицы (задание 2):

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 AUTO\_INCREMENT=4 DEFAULT CHARSET=utf8mb3;

Задание 4

Добавление записей в таблицу `users`:

INSERT INTO `simpledb`.`users` (`id`, `name`, `email`) VALUES ('1', 'Иван', 'ivanov@mail.com');

INSERT INTO `simpledb`.`users` (`id`, `name`, `email`) VALUES ('2', 'Петр', 'petrov@gmail.com');

INSERT INTO `simpledb`.`users` (`id`, `name`, `email`) VALUES ('3', 'Сергей', 'sidorov@yandex.ru');

INSERT INTO `simpledb`.`users` (`id`, `name`, `email`) VALUES ('4', 'Анна', 'smirnova@outlook.com');

Задание 5

Добавление новых столбцов в таблицу `users`:

ALTER TABLE `simpledb`.`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 NOT NULL AFTER `postal\_code`,

ADD COLUMN `created` TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP AFTER `rating`;

---

Задания 6 и 7

Добавление новых пользователей и обновление данных:

INSERT INTO `simpledb`.`users` (`name`, `email`, `postal\_code`, `gender`, `bday`, `rating`)

VALUES ('Ольга', 'olegova@mail.ru', '125456', 'f', '1995-07-15', '1.5');

INSERT INTO `simpledb`.`users` (`name`, `email`, `postal\_code`, `gender`, `bday`, `rating`)

VALUES ('Михаил', 'mihailov@gmail.com', '143789', 'm', '1990-11-22', '2.0');

UPDATE `simpledb`.`users` SET `bday` = '1998-05-10' WHERE (`id` = '1');

UPDATE `simpledb`.`users` SET `bday` = '1997-12-03' WHERE (`id` = '2');

UPDATE `simpledb`.`users` SET `bday` = '1999-08-17' WHERE (`id` = '3');

UPDATE `simpledb`.`users` SET `bday` = '2000-02-28' WHERE (`id` = '4');

Задание 8

Создание связанной таблицы `resume`:

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`));

ALTER TABLE `simpledb`.`resume`

ADD CONSTRAINT `userid`

FOREIGN KEY (`userid`)

REFERENCES `simpledb`.`users` (`id`)

ON DELETE CASCADE

ON UPDATE CASCADE;

Задание 9

Добавление резюме:

INSERT INTO `simpledb`.`resume` (`resumeid`, `userid`, `title`, `skills`)

VALUES ('1', '2', 'Резюме Петрова', 'программирование, базы данных');

INSERT INTO `simpledb`.`resume` (`resumeid`, `userid`, `title`, `skills`)

VALUES ('2', '1', 'Резюме Иванова', 'менеджмент, маркетинг');

Задание 10

Примеры операций:

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

UPDATE `simpledb`.`users` SET `id` = '5' WHERE (`id` = '2');