

Лабораторная работа №1

Болгов Александр, ИВТ-2

Задание 1

Раздел “Instance” (“Экземпляр БД”):

1. Раздел “Startup / Shutdown”. Используется для запуска и остановки экземпляра сервера MySQL. Позволяет управлять состоянием сервера без использования командной строки.
2. Раздел “Server Logs”. Отображает журналы сервера MySQL. Можно просматривать:
 - a. журнал ошибок
 - b. общий журнал
 - c. журнал медленных запросов.
3. Раздел “Options File”. Предназначен для редактирования конфигурационного файла MySQL (my.cnf / my.ini). Позволяет изменять параметры сервера (память, кэш, порты и др.) через графический интерфейс.

Раздел “Performance” (“Производительность”):

1. Раздел “Dashboard”. Это визуальный центр мониторинга в реальном времени. Панель представляет ключевые метрики сервера в виде удобных графиков и диаграмм, сгруппированных по темам. Обычно включает:
2. Раздел “Performance Reports”. Инструмент для генерации детализированного, структурированного отчёта о производительности за выбранный промежуток времени. Он собирает и анализирует данные из системных таблиц и журналов.
3. Раздел “Performance Schema Setup”. Это интерфейс для управления подсистемой Performance Schema - мощным механизмом сбора низкоуровневой статистики о работе сервера, встроенным в MySQL.

Задание 3

```
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

Добавление:

```
INSERT INTO `simpledb`.`users` (`name`, `email`) VALUES ('Paul', 'Paul@gmail.com');  
INSERT INTO `simpledb`.`users` (`name`, `email`) VALUES ('Marina', 'Marina@yandex.ru');  
INSERT INTO `simpledb`.`users` (`name`, `email`) VALUES ('Ekaterina',  
'Ekaterina@mail.com');
```

Изменение:

```
UPDATE `simpledb`.`users` SET `name` = 'Sergey' WHERE (`id` = '3');
```

Задание 5

```
ALTER TABLE `simpledb`.`users`  
ADD COLUMN `gender` ENUM('M', 'F') NOT NULL AFTER `email`,  
ADD COLUMN `bdays` DATE NULL AFTER `gender`,  
ADD COLUMN `postal_code` VARCHAR(10) NULL AFTER `bdays`,  
ADD COLUMN `rating` FLOAT NULL AFTER `postal_code`,  
ADD COLUMN `created` TIMESTAMP NULL DEFAULT CURRENT_TIMESTAMP()  
AFTER `rating`,  
CHANGE COLUMN `name` `name` VARCHAR(50) NOT NULL ,  
CHANGE COLUMN `email` `email` VARCHAR(45) NULL ;  
ALTER TABLE `simpledb`.`users` ALTER INDEX `email_UNIQUE` VISIBLE;
```

created TIMESTAMP CURRENT_TIMESTAMP() означает, что поле created имеет тип TIMESTAMP, который является записью даты и времени, а CURRENT_TIMESTAMP() – это функция, которая возвращает текущие дату и время.

Если пользователь не хочет делиться информацией о себе, то я заставлю его написать только его имя и его пол. То есть поля name и gender NOT NULL, а остальные поля могут быть NULL (кроме id и created, поскольку они заполняются автоматически).

Задание 6

```
INSERT INTO simpledb.users (name, email, postal_code, gender, bday, rating)
VALUES
    ('Paul', 'paul@gmail.com', '123789', 'M', '1998-08-12', '1'),
    ('Marina', 'marina@yandex.ru', '124789', 'F', '1999-07-12', '1.1'),
    ('Ekaterina', 'ekaterina@mail.com', '145789', 'F', '2000-02-11', '1.123');
```

Задание 7

В файле с расширением .sql расположены следующие запросы с добавлением в таблицу данных:

```
INSERT INTO `users` VALUES (4,'Paul','paul@gmail.com','M','1998-08-12','123789',1,'2026-02-06 16:15:03'),(5,'Marina','marina@yandex.ru','F','1999-07-12','124789',1.1,'2026-02-06 16:15:03'),(6,'Ekaterina','ekaterina@mail.com','F','2000-02-11','145789',1.123,'2026-02-06 16:15:03');
```

То есть то же самое, что написал я, за исключением того, что здесь также есть заполненные поля id и created.

Задание 8

```
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);
```

Поскольку ON DELETE имеет значение CASCADE, при удалении записи в таблице users, удаляются также все записи в таблице resume, которые ссылаются на удаляемую в users запись.

Задание 9

```
INSERT INTO simpledb.resume (userid, title, skills)
VALUES
    ('4', 'Software Engineer', 'Python, SQL, Git'),
    ('4', 'Data Analyst', 'Excel, SQL, Python'),
    ('5', 'DevOps Engineer', 'Docker, Kubernetes, CI/CD');
```

У одного пользователя может быть минимум 0 резюме, а максимума нет, можно добавить сколько-угодно. В моем случае у Paul 2 резюме, у Marina 1, а у Ekaterina 0.

В файле .sql следующая запись данных в таблицу:

```
INSERT INTO `resume` VALUES (1,4,'Software Engineer','Python, SQL, Git','2026-02-06
17:02:23'),(2,4,'Data Analyst','Excel, SQL, Python','2026-02-06 17:02:23'),(3,5,'DevOps
Engineer','Docker, Kubernetes, CI/CD','2026-02-06 17:02:23');
```

Опять же, все также, как и у меня, + заполнены поля resumeid и created.

При добавлении в таблицу resume записи с несуществующим userid, мы получаем следующий Error:

```
Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails
(`simpledb`.`resume`, CONSTRAINT `userid` FOREIGN KEY (`userid`) REFERENCES
`users` (`id`) ON DELETE CASCADE ON UPDATE CASCADE)
```

То есть такое добавление невозможно, поскольку данному userid не на что ссыльаться в таблице users.

Задание 10

Удалим из таблицы users пользователя Paul, на которого ссылаются две записи в таблице resume:

```
DELETE FROM `simpledb`.`users` WHERE (`id` = '4');
```

Теперь зайдем в таблицу resume и увидим следующее:

100% | 1:1

Result Grid Filter Rows: Search Edit: Export/Import:

resumeid	userid	title	skills	created
3	5	DevOps Engineer	Docker, Kubernetes, CI/CD	2026-02-06 20:02:23
NULL	NULL	NULL	NULL	NULL

Мы видим, что обе записи, в которых userid было 4 (то есть записи, ссылающиеся на Paul), удалились вместе с удалением записи из таблицы users. Это подтверждает работу ON DELETE, который был выставлен в CASCADE: удалились все записи, которые ссылались на id = 4.

Далее посмотрим, что будет, если мы изменим id пользователя Marina в таблице users с 5 на 1:

```
UPDATE `simplesdb`.`users` SET `id` = '1' WHERE (`id` = '5');
```

Снова заходим в таблицу resume:

100% | 1:1

Result Grid Filter Rows: Search Edit: Export/Import:

resumeid	userid	title	skills	created
3	1	DevOps Engineer	Docker, Kubernetes, CI/CD	2026-02-06 20:02:23
NULL	NULL	NULL	NULL	NULL

В нашей единственной записи значение userid поменялось с 5 на 1. Это подтверждает работу ON UPDATE, выставленном в CASCADE: при изменении значения id в таблице users все поля userid, которые на него ссылались, также изменятся.