

Лабораторная работа 1

Задание 1: Используя справочные материалы по MySQL Workbench, опишите назначение пунктов “Instance” (“Экземпляр БД”) и “Performance” (“Производительность”).

Раздел Instance – служит для управления экземплярами сервера:

1. Раздел «Startup/Shutdown» предоставляет доступ к действиям сервисного контроля
Во вкладке Startup/Shutdown можно выполнить следующие действия:
 - посмотреть лог сообщений при запуске;
 - запустить и завершить экземпляр MySQL;
 - посмотреть текущий статус экземпляра MySQL
2. Раздел «Server Logs» позволяет перейти к информации логов для сервера MySQL, представленной на каждой вкладке соединения.
Для каждого соединения на вкладке Server Logs есть дополнительные вкладки для общих логов ошибок и логов медленных запросов (если они доступны)
3. Раздел «Options File» позволяет просматривать и редактировать файл конфигурации MySQL (my.ini в Windows или my.cnf в Linux и macOS)

Раздел Performance - мониторинг производительности сервера:

1. Раздел «Dashboard» дает статистическое представление о производительности сервера. На панели мониторинга можно посмотреть ключевые показатели производительности сервера, сети и InnoDB.
Кроме того, в этом разделе есть возможность:
 - Получить информацию об операциях сервера MySQL с помощью отчётов Performance Schema Reports.
 - Посмотреть ключевую статистику выполненных запросов с помощью функции Query Statistics.
2. Раздел «Performance Reports» дает анализ производительности баз данных MySQL.
В этом разделе представлены более 20 отчётов, которые помогают анализировать горячие точки ввода-вывода, дорогие SQL-запросы, статистику ожидания и метрики двигателя InnoDB.
3. Раздел «Performance Schema Setup» - формирование и точная настройка инструментровки Performance Schema

По умолчанию открывается страница «Easy Setup». Чтобы включить все доступные инструменты Performance Schema, нужно передвинуть слайдер «Performance Schema Full Enabled» в положение «YES».

Задание 2-3:

```
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:

Запросы, которые будут сделаны в результате добавления записей в таблицу:

```
DELETE FROM `simplifiedb`.`users` WHERE (`id` = '1');  
DELETE FROM `simplifiedb`.`users` WHERE (`id` = '2');  
DELETE FROM `simplifiedb`.`users` WHERE (`id` = '3');  
INSERT INTO `simplifiedb`.`users` (`name`, `email`) VALUES ('Kristina',  
'kristi.cool@ya.ru');  
INSERT INTO `simplifiedb`.`users` (`name`, `email`) VALUES ('Marina',  
'marinka.malinka@gygl.com');  
INSERT INTO `simplifiedb`.`users` (`name`, `email`) VALUES ('Oleg',  
'Olejka.master@ya.ru');
```

SQL-запрос после обновления одного из полей:

```
UPDATE `simplifiedb`.`users` SET `email` = 'kristina.vladivistok@gygl.com' WHERE  
(`id` = '4');
```

Задание 5:

-Определение столбца created как TIMESTAMP с значением по умолчанию CURRENT_TIMESTAMP означает, что при добавлении новой записи в таблицу столбец будет автоматически заполняться текущей датой и временем.

- Все поля могут быть NULL в зависимости от того, будет ли пользователь вводить свои данные

```
ALTER TABLE `simplifiedb`.`users`  
ADD COLUMN `gender` ENUM('M', 'F') NULL AFTER `email`,  
ADD COLUMN `bday` DATE NULL AFTER `gender`,  
ADD COLUMN `postal` VARCHAR(10) NULL AFTER `bday`,  
ADD COLUMN `rating` FLOAT NULL AFTER `postal`,  
ADD COLUMN `created` TIMESTAMP NULL AFTER `rating`,  
CHANGE COLUMN `name` `name` VARCHAR(50) NOT NULL ;
```

Задание 7:

```
/*  
-- Query: SELECT * FROM simplifiedb.users  
LIMIT 0, 1000  
  
-- Date: 2025-02-13 14:33  
*/  
  
INSERT INTO `` (`id`,`name`,`email`,`gender`,`bday`,`postal`,`rating`,`created`)  
VALUES (4,'Kristina','kristina.vladivistok@gygl.com','F','2000-12-31',  
'576421',1,'2025-02-13 13:45:00');  
  
INSERT INTO `` (`id`,`name`,`email`,`gender`,`bday`,`postal`,`rating`,`created`)  
VALUES (5,'Marina','marinka.malinka@gygl.com','F','1990-04-12',  
'907356',1.123,'2025-02-13 13:50:00');  
  
INSERT INTO `` (`id`,`name`,`email`,`gender`,`bday`,`postal`,`rating`,`created`)  
VALUES (6,'Oleg','Olejka.master@ya.ru','M','2009-09-14','415787',1.123,'2025-02-13 14:00:00');
```

Задание 8:

```
CREATE TABLE `simplifiedb`.`resume` (
```

```
`resumeid` INT NOT NULL AUTO_INCREMENT,  
`userid` INT NOT NULL,  
`title` VARCHAR(100) NOT NULL,  
`skills` TEXT NULL,  
`created` VARCHAR(45) NULL,  
`resumecol` TIMESTAMP NULL DEFAULT CURRENT_TIMESTAMP(),  
PRIMARY KEY (`resumeid`),  
INDEX `userid_idx` (`userid` ASC) VISIBLE,  
CONSTRAINT `userid`  
FOREIGN KEY (`userid`)  
REFERENCES `simplifiedb`.`users` (`id`)  
ON DELETE CASCADE  
ON UPDATE CASCADE);
```

СУБД при удалении связанных записей из таблиц users и resume будет удалять данные ото всюду. Сначала из родительской, а потом из дочерней

Задание 9:

По моему мнению у одного пользователя может быть минимум одно резюме, а максимум столько, сколько навыков у пользователя


resumeid,userid,title,skills,created

1,6,Oleg,"top programist","2024-02-13 15:51:00"

2,5,Marina,"master manikur","2024-02-13 16:00:00"

При добавлении не существующего пользователя высвечивается ошибка

The following tasks will now be executed. Please monitor the execution.
Press Show Logs to see the execution logs.

 Execute SQL Statements

Error: There was an error while applying the SQL script to the database.

Message Log

Operation failed: There was an error while applying the SQL script to the database.

Executing:
INSERT INTO `simplifiedb`.`resume` (`resumeid`, `userid`, `title`, `skills`, `created`) VALUES ('3', '1', 'Kirill', 'go injener', '2024-02-13 16:05:00');

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

SQL Statement:
INSERT INTO `simplifiedb`.`resume` (`resumeid`, `userid`, `title`, `skills`, `created`) VALUES ('3', '1', 'Kirill', 'go injener', '2024-02-13 16:05:00')

Hide Logs

Back

Finish

Cancel

Задание 10:

```
DELETE FROM `simplifiedb`.`users` WHERE (`id` = '3');
```

Связанная сущность удалилась в дочерней таблице, как и в родительской

```
UPDATE `simplifiedb`.`users` SET `id` = '6' WHERE (`id` = '5');
```

Id пользователя изменился как в родительском файле, так и в дочернем