

Лабораторная работа 1. Начало работы с MySQL. MySQL Workbench

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

Раздел «Instance» (Экземпляр БД):

1. Раздел Startup/Shutdown

В этом разделе можно «Остановить сервер», так же отображается статус сервера и журнал сообщений о запуске

2. Раздел Server logs

Отображается информация журнала для сервера MySQL, представленная каждой вкладкой подключения. Для каждого соединения вкладка «Администрирование — Журналы сервера» содержит дополнительные вкладки для общих журналов ошибок и медленных журналов (если они доступны).

Файл журнала ошибок

На следующем рисунке показан пример записей на вкладке Файл журнала ошибок.

Медленный файл журнала

На следующем рисунке показан пример записей на вкладке Slow Log File.

3. Раздел Options file

Редактор файла параметров используется для просмотра и редактирования файла конфигурации MySQL (my.ini в Windows или my.cnf в Linux и macOS) путем установки флажков и других элементов управления графического интерфейса, а затем внесения изменений. MySQL Workbench делит файл опций на свои собственные группы в виде набора вкладок (таких как «Общие» , «Ведение журнала» , «InnoDB » и т. д.). Внесите изменения и нажмите «Применить» , чтобы зафиксировать изменения.

Редактор файлов опций включает в себя следующие компоненты:

Группы файлов опций, разделенные на удобные вкладки MySQL Workbench.

Поле поиска параметра «Найти» для поиска в файле конфигурации параметров MySQL.

Путь к файлу конфигурации , чтобы вы знали, какой файл конфигурации вы редактируете.

Селектор группы файлов параметров, чтобы выбрать параметр [группа] для редактирования. Поскольку одна и та же опция может быть определена в нескольких группах, важно выбрать правильную группу при внесении изменений. [mysqld](сервер MySQL) является группой по умолчанию и наиболее распространенной. Дополнительные сведения о группах см. в разделе Использование файлов опций .

Раздел «Performance» (Производительность):

1. Dashboard

Просматривайте статистику производительности сервера на графической панели. Чтобы отобразить информационную панель, откройте вкладку запроса, а затем нажмите «Информационная панель» в области «Производительность» боковой панели Navigator с

выбранной вкладкой «Управление». На следующем рисунке показано расположение информации на вкладке «Администрирование — панель мониторинга».

Статус сети

Это выделяет статистику сетевого трафика, отправленного и полученного сервером MySQL через клиентские соединения. Точки данных включают входящий сетевой трафик , исходящий сетевой трафик и клиентские подключения .

Статус MySQL

Это выделяет основную активность сервера MySQL и статистику производительности. Точки данных включают эффективность открытого кэша таблиц , количество выполненных операторов SQL и количество (в секунду) для операторов SELECT, INSERT, UPDATE, DELETE, CREATE, ALTER и DROP.

Статус InnoDB

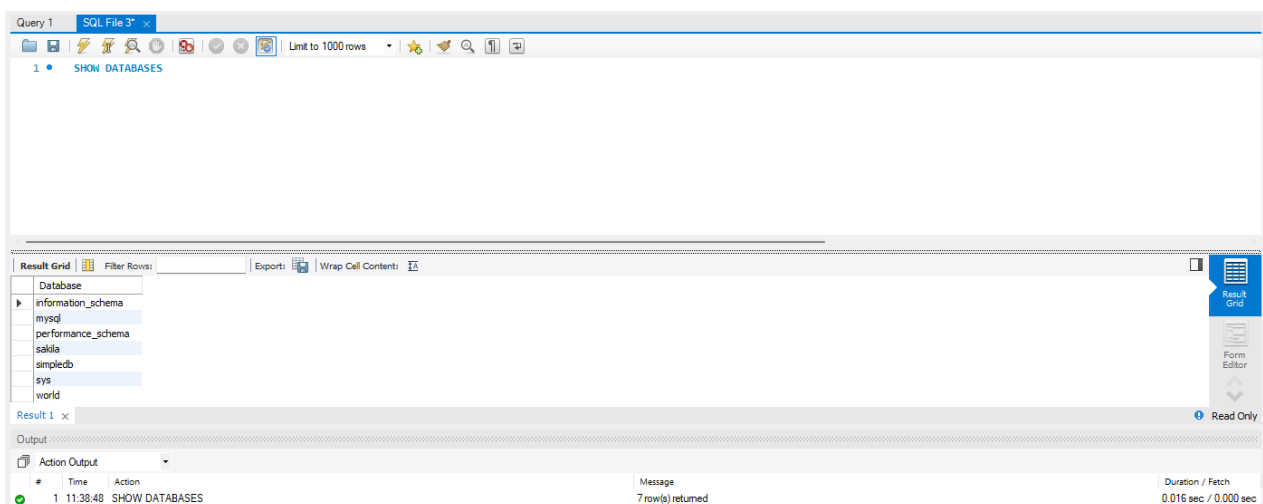
Это обеспечивает обзор буферного пула InnoDB и дисковой активности, которая генерируется механизмом хранения InnoDB. Точки данных разделены на три группы: Usage, Writes, Reads.

2. Performance reports

Более 20 отчетов помогают анализировать производительность ваших баз данных MySQL. Целевые отчеты позволяют анализировать горячие точки ввода-вывода, дорогостоящие операторы SQL, статистику ожидания, показатели механизма InnoDB. MySQL Workbench использует представления SYS в схеме производительности.

3. Performance Schema Setup

Отчеты на основе схемы производительности дают представление об операциях сервера MySQL с помощью полезных отчетов высокого уровня. MySQL Workbench использует представления SYS в схеме производительности для создания более 20 отчетов, помогающих анализировать производительность ваших баз данных MySQL. Отчеты помогают анализировать горячие точки ввода-вывода, обнаруживать дорогостоящие операторы SQL и просматривать статистику ожидания и показатели механизма InnoDB. Дополнительные сведения о схеме SYS см. в разделе Схема MySQL sys.



Задание 2: Создать и настроить новую базу данных simpledb.

Table Name: Schema: **simpledb**

Charset/Collation: Engine:

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
name	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
email	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Column Name: Data Type:

Charset/Collation:

Comments:

Storage: ☐ Virtual ☐ Stored

☐ Primary Key ☐ Not Null ☐ Unique

☐ Binary ☐ Unsigned ☐ Zero Fill

☐ Auto Increment ☐ Generated

Columns Indexes Foreign Keys Triggers Partitioning Options

Apply Revert

Output

#	Time	Action	Message	Duration / Fetch
1	11:38:48	SHOW DATABASES	7 row(s) returned	0.016 sec / 0.000 sec
2	11:40:11	DROP DATABASE 'simpledb'	1 row(s) affected	0.015 sec
3	11:40:26	Apply changes to simpledb	Changes applied	
4	11:43:02	Apply changes to users	Changes applied	

Задание 3: Скопируйте запрос, соответствующий созданию этой таблицы и вставьте его в отчет по выполнению этой лабораторной работы

```
CREATE TABLE `users` (  
  `id` int NOT NULL AUTO_INCREMENT,  
  `name` varchar(45) NOT NULL,  
  `email` varchar(45) NOT NULL,  
  PRIMARY KEY (`id`),  
  UNIQUE KEY `userscol1_UNIQUE` (`email`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb3;
```

Задание 4: Добавьте несколько примеров-записей в созданную таблицу

Query 1 test users - Table users

Limit to 1000 rows

1 • `SELECT * FROM simpledb.users;`

Result Grid

id	name	email
1	Vlad	test1
2	Oleg	test2
3	Max	test3
4	NULL	NULL

users 1

Output

#	Time	Action	Message	Duration / Fetch
1	11:38:48	SHOW DATABASES	7 row(s) returned	0.016 sec / 0.000 sec
2	11:40:11	DROP DATABASE 'simpledb'	1 row(s) affected	0.015 sec
3	11:40:26	Apply changes to simpledb	Changes applied	
4	11:43:02	Apply changes to users	Changes applied	
5	11:46:21	SELECT * FROM simpledb.users LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec

```
SELECT `users`.`id`,  
       `users`.`name`,  
       `users`.`email`  
FROM `simplifiedb`.`users`;
```

Задание 5: Дополните таблицу users так, чтобы получилась таблица со следующими полями и параметрами:

- 1. id int pk, not null**
- 2. name varchar(50)**
- 3. email varchar(45)**
- 4. gender ENUM('M', 'F')**
- 5. bday Date**
- 6. postal_code varchar(10)**
- 7. rating float**
- 8. created TIMESTAMP CURRENT_TIMESTAMP()**

Функция CURRENT_TIMESTAMP вернет текущую дату в формате 'YYYY-MM-DD HH:MM:SS', если используется в контексте строки. Функция CURRENT_TIMESTAMP вернет текущую дату в формате YYYYMMDDHHMMSS, если она используется в числовом контексте.

Значение NULL будут иметь поля: email, bday, postal_code

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

Задание 6: Дополните таблицу, добавив данные двумя способами: - с помощью внесения данных вручную (как это было сделано ранее); - с помощью выполнения SQL-запросов ниже;

The screenshot shows a database management interface. At the top, there's a query editor with a SQL query: `SELECT * FROM simpledb.users;`. Below the editor is a table view for the 'users' table. The table has 8 rows and 8 columns: id, name, email, gender, bday, postal_code, rating, created. The data is as follows:

id	name	email	gender	bday	postal_code	rating	created
1	Vlad	test1	NULL	NULL	NULL	NULL	2023-02-17 12:14:23
2	Oleg	test2	NULL	NULL	NULL	NULL	2023-02-17 12:14:23
3	Max	test3	NULL	NULL	NULL	NULL	2023-02-17 12:14:23
4	Ekaterina	ekaterina.petrova@outlook.com	F	2000-02-11	145789	1.123	2023-02-17 12:14:40
5	Paul	paul@superpochta.ru	M	1998-08-12	123789	1	2023-02-17 12:14:50

Below the table is an 'Action Output' log showing the execution of SQL queries. The log includes the following entries:

- 8 12:16:57 SELECT * FROM simpledb.users LIMIT 0, 1000 3 row(s) returned 0.000 sec / 0.000 sec
- 9 12:16:59 SELECT * FROM simpledb.users LIMIT 0, 1000 3 row(s) returned 0.000 sec / 0.000 sec
- 10 12:17:04 SELECT * FROM simpledb.users LIMIT 0, 1000 3 row(s) returned 0.000 sec / 0.000 sec
- 11 12:17:04 INSERT INTO 'simpledb`.`users` ('name', 'email', 'postal_code', 'gender', 'bday', 'rating') VALUES ('Ekaterina', 'ekate... 1 row(s) affected 0.016 sec
- 12 12:17:14 SELECT * FROM simpledb.users LIMIT 0, 1000 4 row(s) returned 0.000 sec / 0.000 sec
- 13 12:17:14 INSERT INTO 'simpledb`.`users` ('name', 'email', 'postal_code', 'gender', 'bday', 'rating') VALUES ('Ekaterina', 'ekate... Error Code: 1062. Duplicate entry 'ekaterina.petrova@outlook.com' for key 'users.userscol1_UNIQUE' 0.000 sec
- 14 12:24:52 SELECT * FROM simpledb.users LIMIT 0, 1000 4 row(s) returned 0.000 sec / 0.000 sec
- 15 12:24:52 INSERT INTO 'simpledb`.`users` ('name', 'email', 'postal_code', 'gender', 'bday', 'rating') VALUES ('Ekaterina', 'ekate... Error Code: 1062. Duplicate entry 'ekaterina.petrova@outlook.com' for key 'users.userscol1_UNIQUE' 0.000 sec
- 16 12:25:58 SELECT * FROM simpledb.users LIMIT 0, 1000 4 row(s) returned 0.000 sec / 0.000 sec
- 17 12:25:58 INSERT INTO 'simpledb`.`users` ('name', 'email', 'postal_code', 'gender', 'bday', 'rating') VALUES ('Ekaterina', 'ekate... Error Code: 1062. Duplicate entry 'ekaterina.petrova@outlook.com' for key 'users.userscol1_UNIQUE' 0.000 sec
- 18 12:26:18 SELECT * FROM simpledb.users LIMIT 0, 1000 4 row(s) returned 0.000 sec / 0.000 sec
- 19 12:26:18 INSERT INTO 'simpledb`.`users` ('name', 'email', 'postal_code', 'gender', 'bday', 'rating') VALUES ('Ekaterina', 'ekate... Error Code: 1062. Duplicate entry 'ekaterina.petrova@outlook.com' for key 'users.userscol1_UNIQUE' 0.000 sec
- 20 12:26:44 SELECT * FROM simpledb.users LIMIT 0, 1000 4 row(s) returned 0.000 sec / 0.000 sec

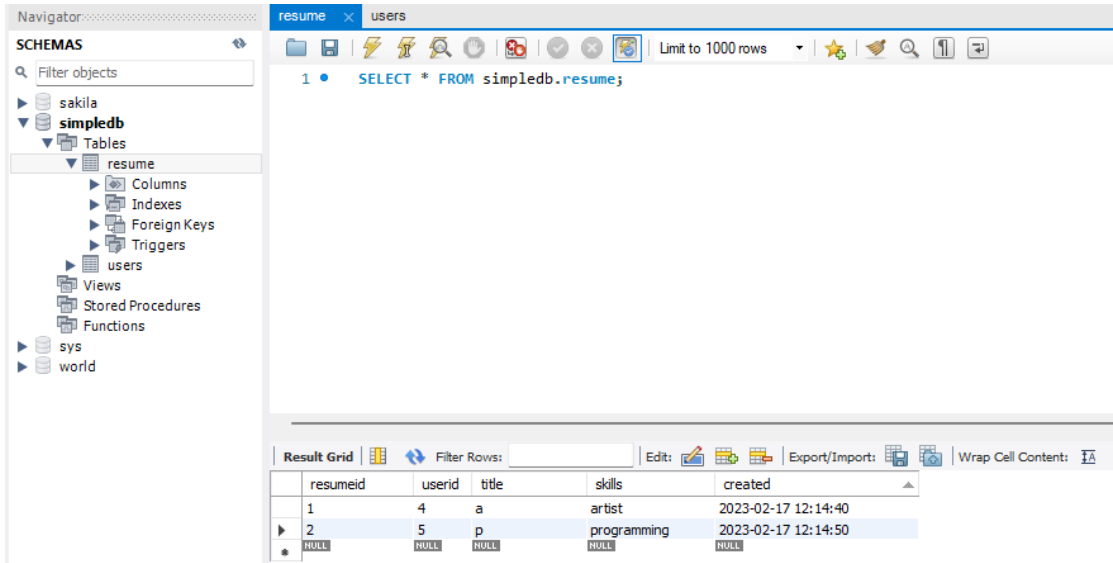
Задание 7: С помощью кнопки “Export recordset to external file” и получите файл с SQL-запросами (он будет экспортирован в формате .sql). Сохраните файл с SQL-запросами (в нем должно происходить добавление в таблицу данных), проанализируйте синтаксис этих запросов и вставьте их в отчет

	A	B	C	D	E	F	G	H	I
1	id,name,email,gender,bday,postal_code,rating,created								
2	1,Vlad,test1,NULL,NULL,NULL,NULL,"2023-02-17 12:14:23"								
3	2,Oleg,test2,NULL,NULL,NULL,NULL,"2023-02-17 12:14:23"								
4	3,Max,test3,NULL,NULL,NULL,NULL,"2023-02-17 12:14:23"								
5	4,Ekaterina,ekaterina.petrova@outlook.com,F,2000-02-11,145789,1.123,"2023-02-17 12:14:40"								
6	5,Paul,paul@superpochta.ru,M,1998-08-12,123789,1,"2023-02-17 12:14:50"								

Задание 8: Создайте еще одну таблицу с названием resume со следующей структурой: - resumeid, INT, PK, NN, AI - userid, INT, NN - title, VARCHAR(100), NN - skills, TEXT - created, TIMESTAMP, Default / Expression: CURRENT_TIMESTAMP()

```
CREATE TABLE `resume` (
  `resumeid` int NOT NULL AUTO_INCREMENT,
  `userid` int NOT NULL,
  `title` varchar(100) NOT NULL,
  `skills` text,
  `created` timestamp NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (`resumeid`),
  KEY `userid_idx` (`userid`),
  CONSTRAINT `userid` FOREIGN KEY (`userid`) REFERENCES `users` (`id`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb3;
```

Задание 9: Наполните вторую таблицу данными так, чтобы в ней была информация хотя бы о нескольких резюме, связанных с уже существующими пользователями из таблицы users.



	A	B	C	D
1	resumeid,userid,title,skills,created			
2	1,4,a,artist,"2023-02-17 12:14:40"			
3	2,5,p,programming,"2023-02-17 12:14:50"			

При попытке добавлении несуществующего userid получается ошибка

