

Лабораторная работа 1.

Начало работы с MySQL. MySQL Workbench.

Выполнил студент 2 курса ИВТ 1гр. 2п.гр.

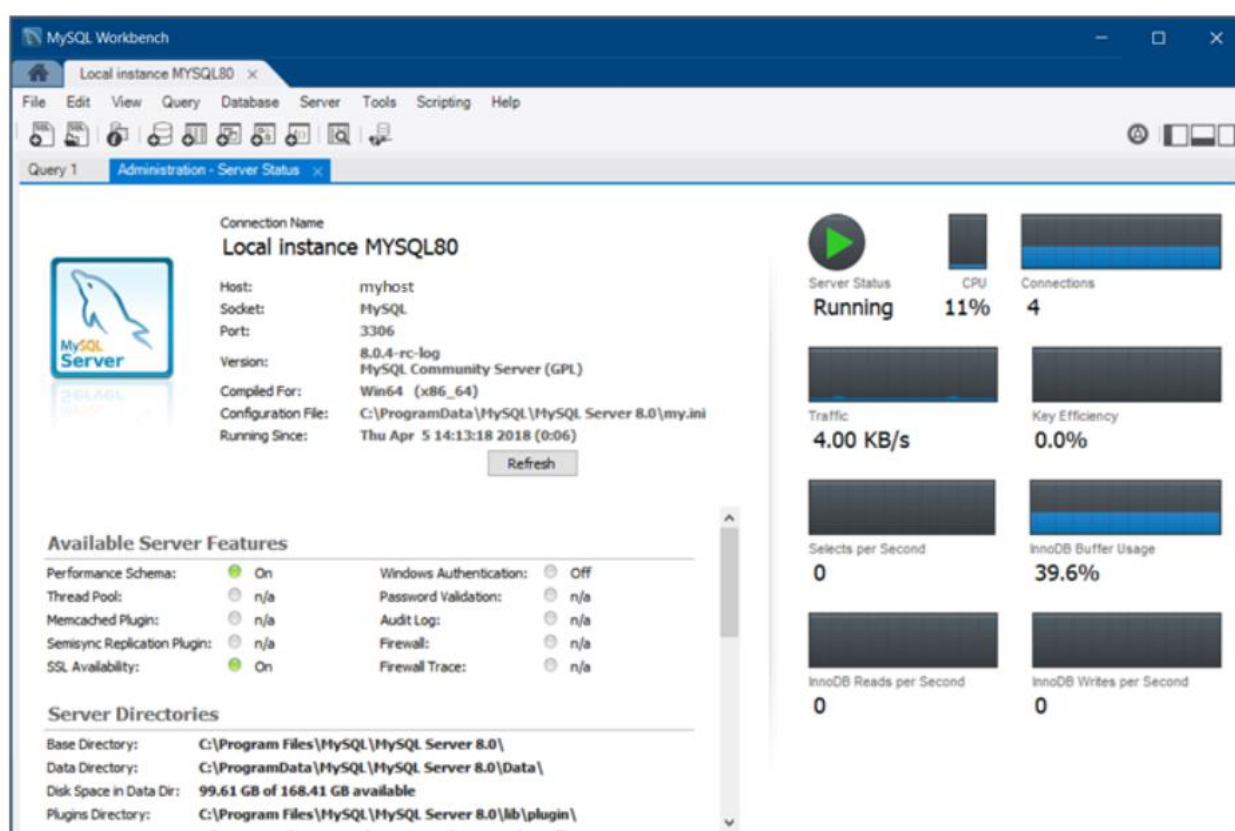
Смородин А.А.

Задание 1. Назначение пунктов меню Management, instance, performance.

Раздел Management:

1. Server Status.

Получите немедленный обзор основных индикаторов работоспособности и счетчиков для вашей среды MySQL. Как показано на следующем рисунке, сюда входит просмотр состояния работы сервера (остановлен/работает), доступных функций, основных каталогов сервера, состояния репликации и настроек безопасности для аутентификации и SSL. Отчеты также включают информацию и графики для отслеживания использования памяти, подключений, показателей попаданий и многого другого.



2. Client Connections.

Браузер клиентских подключений выводит список активных и спящих клиентских подключений MySQL, а также добавляет возможность завершать операторы и подключения, а также просматривать дополнительные сведения и атрибуты подключения. На следующем рисунке показан пример информации о клиентском подключении для локального хоста.

Рисунок 5.28 Обзор клиентского подключения

Local instance MySQL80
Client Connections

Threads Connected: 4 Threads Running: 2 Threads Created: 4 Threads Cached: 0 Rejected (over limit): 0
 Total Connections: 32 Connection Limit: 151 Aborted Clients: 0 Aborted Connections: 20 Errors: 0

ID	User	Host	DB	Command	Time	State	Thread ID	Type	Name	Parent ID	Instrumented	Info	Program
32	root	localhost	None	Sleep	1	None	61	FOREGROUND	thread/sql/one_connection	35	YES	NONE	MySQLWorkbench
31	root	localhost	None	Query	0	Sending d...	60	FOREGROUND	thread/sql/one_connection	35	YES	SELECT...	MySQLWorkbench
30	root	localhost	None	Sleep	213	None	59	FOREGROUND	thread/sql/one_connection	35	YES	NONE	MySQLWorkbench
29	root	localhost	None	Sleep	214	None	58	FOREGROUND	thread/sql/one_connection	0	YES	NONE	MySQLWorkbench
6	None	None	None	Daemon	577	Suspending	34	FOREGROUND	thread/sql/compress_gtid_table	1	YES	NONE	None
4	even...	None	None	Sleep	0	Waiting on...	33	FOREGROUND	thread/sql/event_scheduler	1	YES	NONE	None

Refresh Rate: Don't Refresh Kill Query(s) Kill Connection(s) Refresh Show Details

☐ Hide sleeping connections ☒ Hide background threads ☒ Don't load full thread info

Рисунок 5.29 Подробности клиентских подключений

Local instance MySQL57
Client Connections

Threads Connected: 4 Threads Running: 1 Threads Created: 4 Threads Cached: 0 Rejected (over limit): 0
 Total Connections: 7 Connection Limit: 151 Aborted Clients: 1 Aborted Connections: 0 Errors: 0

ID	U...	Host	DB	Command	Time	State	Thread ID	Type	Name	Parent
3	root	localhost	sakila	Sleep	7	None	25	FOREGROUND	thread/sql/one...	
4	root	localhost	sakila	Sleep	8	None	26	FOREGROUND	thread/sql/one...	
5	root	localhost	None	Query	0	Sending...	27	FOREGROUND	thread/sql/one...	
6	root	localhost	None	Sleep	0	None	28	FOREGROUND	thread/sql/one...	

Details Locks Attributes

Processlist ID: 5
Thread ID: 27
Name: thread/sql/one_connection
Type: FOREGROUND
User: root
Host: localhost
Schema: None
Command: Query
Time: 0
State: Sending data
Role:
Instrumented: YES
Parent Thread ID: 22
Info:

```
SELECT t.PROCESSLIST_ID,IF (NAME =
'thread/sql/event_scheduler','event_scheduler',t.
PROCESSLIST_USER) PROCESSLIST_USER,t.
PROCESSLIST_HOST,t.PROCESSLIST_DB,t.
PROCESSLIST_COMMAND,t.PROCESSLIST_TIME,t.
PROCESSLIST_STATE,t.THREAD_ID,t.TYPE,t.NAME,t.
PARENT_THREAD_ID,t.INSTRUMENTED,t.
PROCESSLIST_INFO,a.ATTR_VALUE FROM
performance_schema.threads t LEFT OUTER JOIN
performance_schema.session_connect_attrs a ON t.
processlist_id = a.processlist_id AND (a.attr_name IS
NULL OR a.attr_name = 'program_name') WHERE t.
TYPE <> 'BACKGROUND'
```

Explain for Connection

☐ Hide sleeping connections ☒ Hide background threads ☒ Don't load full thread info Hide Details

Рисунок 5.30 Браузер блокировок метаданных

Query 1

Administration - Client Connections

Local instance 3306

Client Connections

Threads Connected: 6

Threads Running: 2

Threads Created: 6

Threads Cached: 0

Rejected (over limit): 0

Total Connections: 11

Connection Limit: 1100

Aborted Clients: 0

Aborted Connections: 3

Errors: 0

Time	State	Thread Id	Type	Name	Pa
235	None	23	FOREGRO...	thread/sq...	
0	Sending data	25	FOREGRO...	thread/sq...	
2	None	26	FOREGRO...	thread/sq...	
17	None	27	FOREGRO...	thread/sq...	
3	Waiting for table metadata lock	28	FOREGRO...	thread/sq...	
14	None	30	FOREGRO...	thread/sq...	

Details

Locks

Attributes

Metadata locks (MDL) protect concurrent access to object metadata (not table row/data locks)

Granted Locks (and threads waiting on them)

Locks this connection currently owns and connections that are waiting for them.

Object	Type	Duration
<global>	INTENTION_...	STATE
sakila	INTENTION_...	TRAN
sakila.actor	SHARED_UPG...	TRAN
thread 28	EXCLUSIVE	TRAN

Pending Locks

The connection is waiting for a lock on table sakila.actor, held by threads 30, 28

Type: EXCLUSIVE

Duration: TRANSACTION

☐ Hide sleeping connections

☒ Hide background threads

☒ Don't load full thread info

Hide Details

Refresh Rate: Don't Refresh

Kill Query(s)

Kill Connection(s)

Refresh

значений.

Рисунок 5.31 Атрибуты клиентского подключения

MySQL Workbench

Local instance MySQL57

File Edit View Query Database Server Tools Scripting Help

Query 1 Administration - Client Connections

Local instance MySQL57

Client Connections

Threads Connected: 4

Threads Running: 1

Threads Created: 4

Threads Cached: 0

Rejected (over limit): 0

Total Connections: 7

Connection Limit: 151

Aborted Clients: 1

Aborted Connections: 0

Errors: 0

State	Thread Id	Type	Name	Parent Thr...	Instrumented	Info	Program
None	25	FOREGROUND	thread/sql/one...	0	YES	NULL	MySQLW
None	26	FOREGROUND	thread/sql/one...	0	YES	NULL	MySQLW
Sending...	27	FOREGROUND	thread/sql/one...	22	YES	SELECT t.PR...	MySQLW
None	28	FOREGROUND	thread/sql/one...	22	YES	NULL	MySQLW

Details

Locks

Attributes

Attribute

Value

_os	Win64
_client_name	libmysql
_pid	5660
_thread	4312
_platform	x86_64
program_name	MySQLWorkbench
_client_version	5.6.17

☐ Hide sleeping connections

☒ Hide background threads

☒ Don't load full thread info

Hide Details

Refresh Rate: Don't Refresh

Kill Query(s)

Kill Connection(s)

Refresh

3. Users and Privileges.

Вкладка Администрирование - Пользователи и привилегии предоставляет список всех пользователей и привилегий, которые относятся к активному экземпляру сервера MySQL. На этой вкладке вы можете добавлять и управлять учетными записями пользователей, настраивать привилегии и истекать пароли.

Вкладка «Администрирование» — «Пользователи и привилегии» содержит несколько областей задач, которые описаны в следующих разделах:

- Учетные записи пользователей
- Вкладка «Вход»
- Вкладка «Лимиты учетной записи»
- Вкладка «Административные роли»
- Вкладка «Привилегии схемы»

4. Status and System Variables.

Вторичная вкладка Администрирование - Состояние и Системные переменные содержит полный набор переменных сервера для активного соединения MySQL. Вы также можете скопировать все или выбранные переменные в буфер обмена.

5. Data Export

Эта вкладка позволяет экспортировать данные MySQL. Выберите каждую схему, которую вы хотите экспортировать, по желанию выберите конкретные объекты/таблицы схемы из каждой схемы и сгенерируйте экспорт. Параметры конфигурации включают экспорт в папку проекта или автономный файл SQL, по желанию дампы сохраненных процедур и событий или пропуск данных таблицы.

6. Data import/Restore

Восстановите экспортированные данные из операции **экспорта данных** или из других экспортированных данных из команды **mysqldump**.

Раздел INSTANCE:

1. Startup/Shutdown.

Вкладка «Администрирование» — «Запуск/Завершение работы» позволяет выполнять следующие действия по управлению службами:

- Просмотр журнала сообщений при запуске.
- Запустите и остановите экземпляр MySQL.
- Просмотр текущего состояния экземпляра MySQL.

2. Server Logs.

Вкладка Администрирование - Журналы сервера отображает информацию журнала для сервера MySQL, представленного каждой вкладкой соединения. Для каждого соединения вкладка Администрирование - Журналы сервера включает дополнительные вкладки для общих журналов ошибок и медленных журналов (если они доступны).

3. Options File.

Редактор файла параметров используется для просмотра и редактирования файла конфигурации MySQL (`my.ini` в Windows или `my.cnf` Linux и macOS) путем установки флажков и других элементов управления GUI, а затем внесения изменений. MySQL Workbench делит файл параметров на собственные группы в виде набора вкладок (например, General, Logging, InnoDB и т. д.). Внесите изменения и нажмите Apply, чтобы зафиксировать изменения.

Редактор файла параметров включает в себя следующие компоненты:

- Группировки файлов параметров, разделенные на удобные вкладки MySQL Workbench.
 - Поле поиска параметров «Найти» для поиска в файле конфигурации параметров MySQL.
 - Путь к файлу конфигурации, чтобы вы знали, какой файл конфигурации вы редактируете.
 - Селектор группы файлов опций, чтобы выбрать опцию [группу] 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. Для получения дополнительной информации о группах см. Использование файлов опций.

Раздел PERFORMANCE:

1. Dashboard.

Просмотр статистики производительности сервера на графической панели. Чтобы отобразить панель, откройте вкладку запроса, а затем щелкните Панель мониторинга в области Производительность боковой панели Navigator с выбранной вкладкой Управление. На следующем рисунке показана компоновка информации на вкладке Администрирование - Панель мониторинга.

2. Performance Reports.

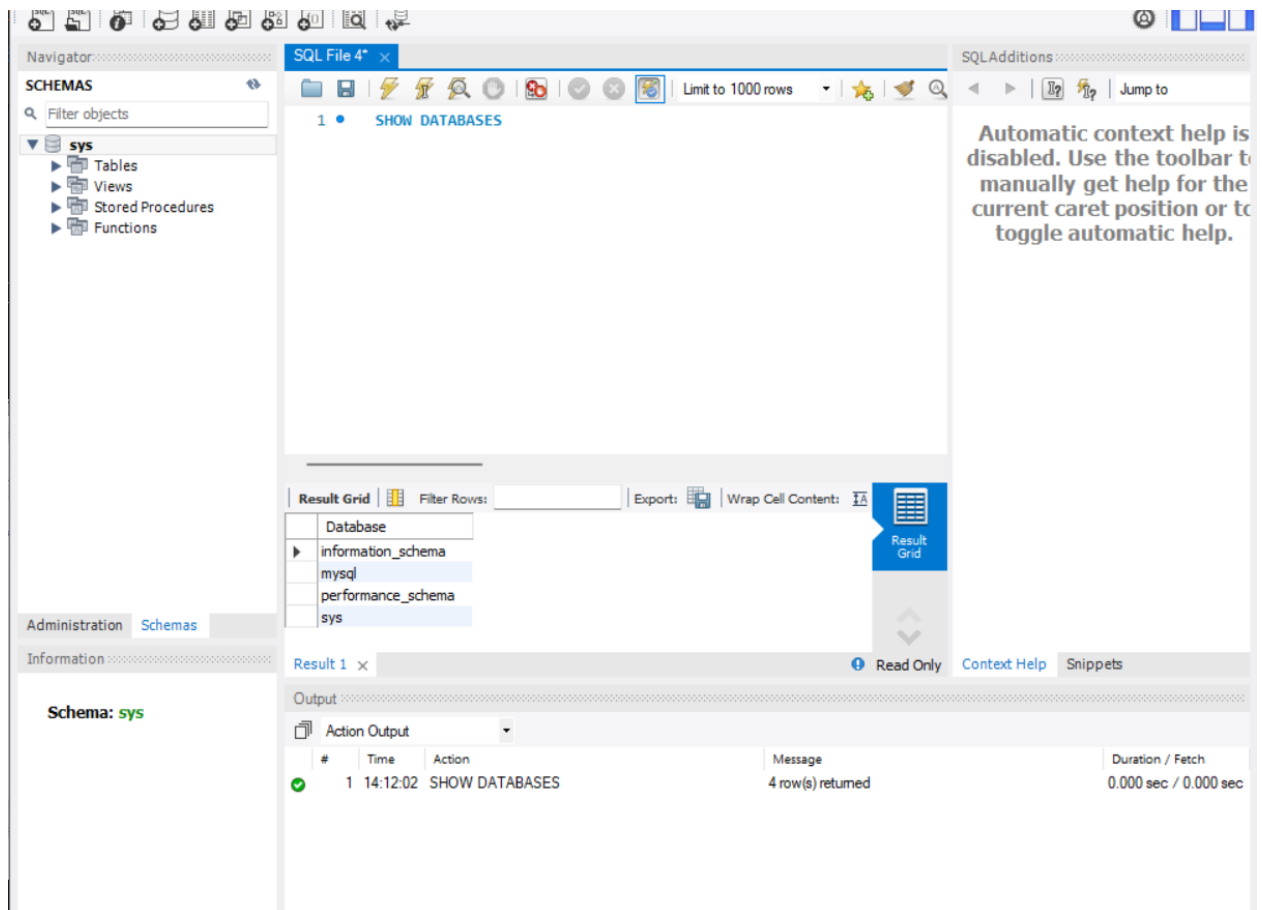
Более 20 отчетов помогают анализировать производительность ваших баз данных MySQL. Целевые отчеты позволяют анализировать горячие точки ввода-вывода, высокзатратные SQL-операторы, статистику ожидания, метрики движка InnoDB. MySQL Workbench использует представления SYS в схеме производительности.

3. Performance Schema Setup.

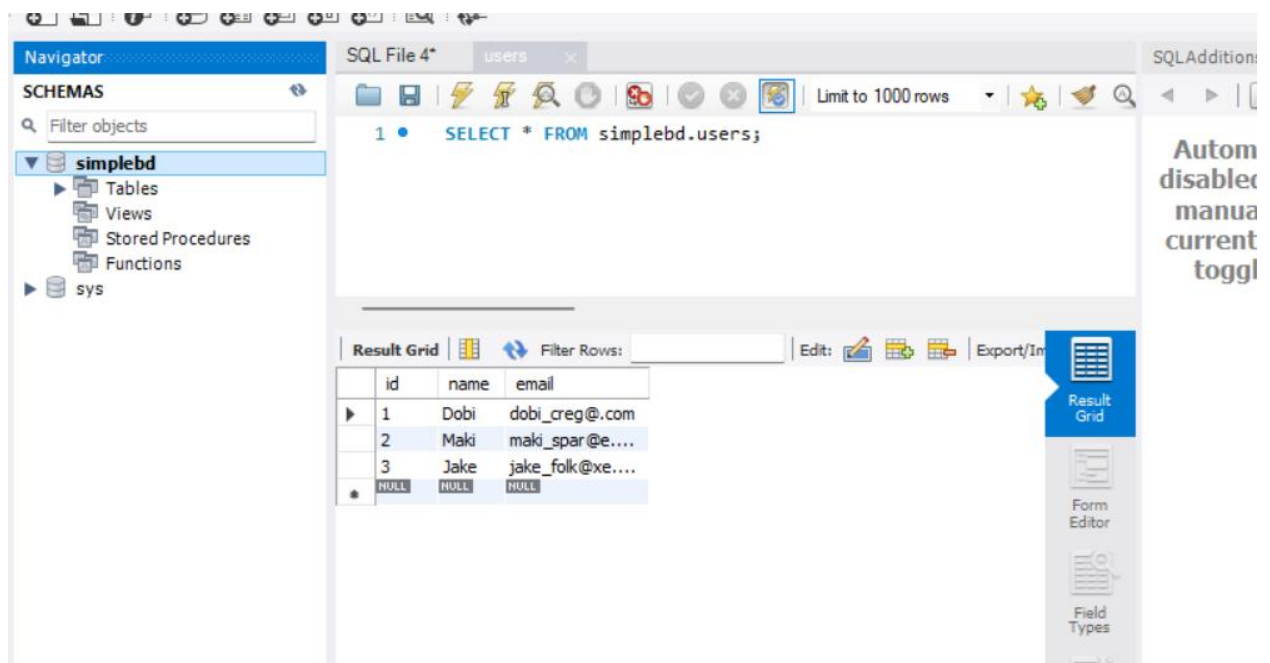
Performance Schema (Схема производительности) в MySQL – это специализированная схема хранения данных, предназначенная для сбора и анализа данных о производительности MySQL-сервера во время его работы. Это мощный инструмент, который позволяет администраторам и разработчикам понимать, что происходит внутри MySQL, выявлять узкие места и оптимизировать работу сервера.

Основные характеристики Performance Schema:

- Сбор данных во время работы сервера: Performance Schema собирает данные о производительности в режиме реального времени, не требуя перезагрузки сервера.
- Низкое влияние на производительность: Сбор данных спроектирован так, чтобы оказывать минимальное влияние на общую производительность сервера (в отличие от, например, включения общего журнала запросов). Однако, слишком активная настройка мониторинга может всё же привести к замедлению.
- Организованная структура данных: Данные организованы в виде таблиц, которые можно запросить с помощью обычного SQL.
- Гибкая настройка: Вы можете настроить, какие типы событий и данные должны собираться. Это позволяет сосредоточиться на интересующих вас аспектах производительности.
- Динамическое управление: Performance Schema управляется динамически, позволяя включать и отключать сбор данных, изменять фильтры и другие параметры без перезапуска сервера.



Задание 2:



Задание 3:

```
CREATE TABLE `users` (  
  `id` int NOT NULL AUTO_INCREMENT,  
  `name` varchar(45) NOT NULL,  
  `new_tablecol1` varchar(45) NOT NULL,  
  PRIMARY KEY (`id`),  
  UNIQUE KEY `new_tablecol1_UNIQUE` (`new_tablecol1`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb3;
```

Задание 4:

```
INSERT INTO `simplebd`.`user` (`id`, `name`, `email`) VALUES ('1', 'Dobi',  
'dobi_creg@.com');
```

```
INSERT INTO `simplebd`.`user` (`id`, `name`, `email`) VALUES ('2', 'Maki',  
'maki_spar@e.com');
```

```
INSERT INTO `simplebd`.`user` (`id`, `name`, `email`) VALUES ('3', 'Jake',  
'jake_folk@xe.com');
```

Запрос после обновления двух полей:

```
UPDATE `simplebd`.`users` SET `id` = '223' WHERE (`id` = '2');
```

```
UPDATE `simplebd`.`users` SET `email` = 'dobi_crig@.com' WHERE (`id` = '1');
```

Задание 5:

```
ALTER TABLE `simplebd`.`users`  
  
ADD COLUMN `gender` ENUM('M', 'F') NULL AFTER `email`,  
  
ADD COLUMN `bday` DATE NULL AFTER `gender`,  
  
ADD COLUMN `postal_code` VARCHAR(10) NOT NULL AFTER `bday`,  
  
ADD COLUMN `rating` FLOAT NULL AFTER `postal_code`,  
  
ADD COLUMN `created` TIMESTAMP NULL AFTER `rating`,  
  
CHANGE COLUMN `id` `id` INT NOT NULL ,  
  
CHANGE COLUMN `name` `name` VARCHAR(50) NOT NULL ,  
  
DROP PRIMARY KEY,  
  
ADD PRIMARY KEY (`id`, `postal_code`);  
  
;
```

Задание 6:

	id	name	email	gender	bday	postal_code	rating	created
▶	1	Dobi	dobi_crig@.com	M	1998-08-08	11478	1	2025-02-20 15:55:21
	3	Jake	jake_folk@xe....	M	2004-05-16	12366	2	2025-02-20 15:55:21
	223	Maki	maki_spar@e....	M	2000-01-12	14679	3	2025-02-20 15:55:21
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Задание 7:

	A	B
1	id,name,email,gender,bday,postal_code,rating,created	
2	1,Dobi,dobi_crig@.com,M,1998-08-08,11478,1,"2025-02-20 15:55:21"	
3	3,Jake,jake_folk@xe.com,M,2004-05-16,12366,2,"2025-02-20 15:55:21"	
4	223,Maki,maki_spar@e.com,M,2000-01-12,14679,3,"2025-02-20 15:55:21"	
5		
6		
7		
8		
9		
0		
1		
2		
3		
4		
5		

Задание 8:

```
CREATE TABLE `simplebd`.`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 `simplebd`.`users` (`id`)  
    ON DELETE CASCADE  
    ON UPDATE CASCADE);
```

Задание 9:

Result Grid					
Filter Rows:					
	resumeid	userid	title	skills	created
	1	3	Dev...	JS,...	2025-0...
	2	3	Resum	high	2025-0...
	3	223	Resum	high	2025-0...
	4	223	Hr	high	2025-0...
	5	3	Engi...	Nud...	2025-0...
*	NULL	NULL	NULL	NULL	NULL

Result Grid								
Filter Rows:								
	id	name	email	gender	bday	postal_code	rating	created
	1	Dobi	dobi_crig@.com	M	1998-08-08	11478	1	2025-02-20 15:55:21
	3	Jake	jake_folk@xe....	M	2004-05-16	12366	2	2025-02-20 15:55:21
	223	Maki	maki_spar@e....	M	2000-01-12	14679	3	2025-02-20 15:55:21
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

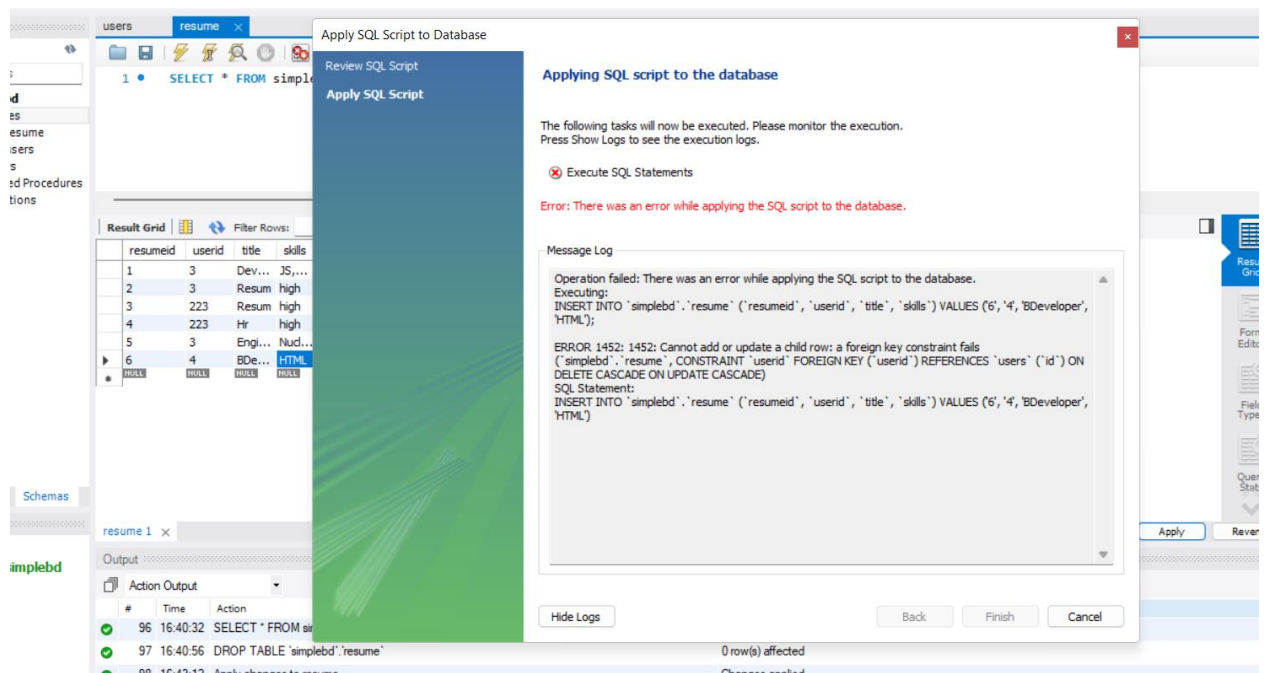
Минимум:

Теоретически, у одного пользователя может быть ноль резюме. В нашей базе данных у пользователя с id 2 нет записей в таблице resume, это означает, что у него нет резюме.

Максимум:

Теоретически у одного пользователя может быть неограниченное количество резюме. В нашей базе данных у пользователя с id 3 в таблице resume 3 записи, что означает, что у него 3 резюме.

A1 resumeid,userid,title,skills,created						
	A	B	C	D	E	
1	resumeid,userid,title,skills,created					
2	1,3,Developer,"JS,Python","2025-02-20 16:46:50"					
3	2,3,Resum,high,"2025-02-20 16:46:50"					
4	3,223,Resum,high,"2025-02-20 16:46:50"					
5	4,223,Hr,high,"2025-02-20 16:56:16"					
6	5,3,Engineer,"Nuclear Engineer","2025-02-20 16:56:16"					
7						
8						
9						
10						
11						



Если мы попытаемся в таблицу resume добавить id которого нет в таблице users, то MySQL выдаёт ошибку и не позволяет добавить данные в таблицу. Происходит это так как мы пытаемся связать резюме с несуществующим пользователем. Чтобы исправить данную ошибку нужно либо добавить пользователя с таким id в таблицу users, либо изменить userid в таблице resume на существующий в таблице users.

Задание 10:

Таблицы до удаления информации об 1 из пользователей.

WS
red Procedures
ctions

Result Grid	Filter Rows:	Edit:	Export/Import:	
resumeid	userid	title	skills	created
1	3	Dev...	JS,...	2025-0...
2	3	Resum	high	2025-0...
3	223	Resum	high	2025-0...
4	223	Hr	high	2025-0...
5	3	Engl...	Nud...	2025-0...
6	1	BDe...	HTML	2025-0...
NULL	NULL	NULL	NULL	NULL

cedures

Result Grid	Filter Rows:	Edit:	Export/Import:	Wrap Cell Contents:			
id	name	email	gender	bdays	postal_code	rating	created
1	Dobi	dobi_crig@.com	M	1998-08-08	11478	1	2025-02-20 15:55:21
3	Jake	jake_folk@xe...	M	2004-05-16	12366	2	2025-02-20 15:55:21
223	Maki	maki_spar@e...	M	2000-01-12	14679	3	2025-02-20 15:55:21
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Таблицы после удаления 1 из пользователей.

Procedures
ns

Result Grid

Filter Rows:

Edit: Export/Import: Wrap Cell Content:

	id	name	email	gender	bday	postal_code	rating	created
▶	3	Jake	jake_folk@xe....	M	2004-05-16	12366	2	2025-02-20 15:55:21
	223	Maki	maki_spar@e....	M	2000-01-12	14679	3	2025-02-20 15:55:21
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Result Grid

Filter Rows:

Edit: Export/Import: Wrap Cell Content:

	resumeid	userid	title	skills	created
▶	2	3	Resum	high	2025-02-20 16:46:50
	3	223	Resum	high	2025-02-20 16:46:50
	4	223	Hr	high	2025-02-20 16:56:16
	5	3	Engineer	Nuclear Engineer	2025-02-20 16:56:16
*	NULL	NULL	NULL	NULL	NULL

Как мы видим после удаления данных об 1 пользователе из табицы users данные об этом пользователе автоматически пропали из таблицы resume.









Запрос:

```
DELETE FROM `simplebd`.`users` WHERE (`id` = '1') and (`postal_code` = '11478');
```

Таблицы до смены id

[illegible]

- users
- AWS
- Required Procedures
- actions

Result Grid			Filter Rows: <input type="text"/>	Edit:   	Export/Import:  	Wrap Cell Content: 
	resumeid	userid	title	skills	created	
▶	2	3	Resum	high	2025-02-20 16:46:50	
	3	223	Resum	high	2025-02-20 16:46:50	
	4	223	Hr	high	2025-02-20 16:56:16	
	5	3	Engineer	Nuclear Engineer	2025-02-20 16:56:16	
✱	NULL	NULL	NULL	NULL	NULL	

После смены id в таблице users с 3 на 2 userid в таблице resume автоматически заменился с 3 на 2

The screenshot shows the DBeaver SQL editor interface. At the top, a toolbar contains various icons for file operations, search, and execution. Below the toolbar, a SQL query is entered in the editor:

```
1 • SELECT * FROM simplebd.users;
```

Below the query editor, the "Result Grid" tab is active, displaying the results of the query. The results are shown in a table with the following columns: id, name, email, gender, bday, postal_code, rating, and created. The table contains three rows of data:

	id	name	email	gender	bday	postal_code	rating	created
2	Jake	jake_folk@xe....	M	2004-05-16	12366	2	2025-02-20 15:55:21	
223	Maki	maki_spar@e....	M	2000-01-12	14679	3	2025-02-20 15:55:21	
	NULL	NULL	NULL	NULL	NULL	NULL	NULL	

The "Result Grid" tab also includes a "Filter Rows:" field and a "Wrap Cell Content:" checkbox.

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Co

	resumeid	userid	title	skills	created
▶	2	2	Resum	high	2025-02-20 16:46:50
	3	223	Resum	high	2025-02-20 16:46:50
	4	223	Hr	high	2025-02-20 16:56:16
	5	2	Engineer	Nuclear Engineer	2025-02-20 16:56:16
✱	NULL	NULL	NULL	NULL	NULL