

**УВАЖАЕМЫЕ СТУДЕНТЫ!** Изучите повторите теоретический материал по созданию и наполнению таблиц информацией в среде MS SQL SERVER, выполните задание лабораторной работы, при выполнении готовьте отчет к лабораторной работе – сохраняйте скриншоты пошагового выполнения работы.

*Работа рассчитана на 4 часа.*

Результаты работы – отчет, предоставить преподавателю на e-mail: [xvsviv@rambler.ru](mailto:xvsviv@rambler.ru) **в трехдневный срок с момента получения задания.**

*При возникновении вопросов по приведенному материалу  
обращаться по следующим номерам телефонов:072-138-93-11.*

***ВНИМАНИЕ!!! При отправке работы, не забывайте указывать  
ФИО студента, наименование дисциплины, дата проведения занятия  
(по расписанию).***

## **Лабораторная работа**

### **Создание нового пользователя в MySQL. Распределение привилегий пользователей.**

**Цель работы:** знакомство с системой безопасности в MySQL.

#### **Система привилегий и безопасность в MySQL**

- User
- Db
- Host
- Пользовательские привилегии

#### **База данных mysql и таблицы привилегий.**

Итак, вы успешно вошли в базу данных mysql, которая используется для администрирования сервера. Что же здесь находится? А находятся здесь 5 таблиц, которые ничем не отличаются от других таблиц баз данных, за исключением того, что эти таблицы используются для предоставления доступа к базам данных и таблицам в них пользователям. Рассмотрим каждую из них.

Введите следующую команду, **show tables**, которая покажет таблицы в базе данных mysql.

Кратко рассмотрим функции каждой из таблиц:

### Таблица User

Определяет, разрешено ли пользователю, пытающемуся подключиться к серверу делать это. Содержит имя пользователя, пароль а также привилегии. Если ввести команду `show columns from user;` то получим следующее:

#### Структура таблицы User

Field	Type	Null	Key	Default	Extra
Host	char(60)		PRI		
User	char(16)		PRI		
Password	char(41)				
Select_priv	enum('N','Y')			N	
Insert_priv	enum('N','Y')			N	
Update_priv	enum('N','Y')			N	
Delete_priv	enum('N','Y')			N	
Create_priv	enum('N','Y')			N	
Drop_priv	enum('N','Y')			N	
Reload_priv	enum('N','Y')			N	
Shutdown_priv	enum('N','Y')			N	
Process_priv	enum('N','Y')			N	
File_priv	enum('N','Y')			N	
Grant_priv <sup>1</sup>	enum('N','Y')			N	
References_priv	enum('N','Y')			N	
Index_priv	enum('N','Y')			N	
Alter_priv	enum('N','Y')			N	
Show_db_priv	enum('N','Y')			N	
Super_priv	enum('N','Y')			N	
Create_tmp_table_priv	enum('N','Y')			N	
Lock_tables_priv	enum('N','Y')			N	
Execute_priv	enum('N','Y')			N	
Repl_slave_priv	enum('N','Y')			N	
Repl_client_priv	enum('N','Y')			N	
Create_view_priv	enum('N','Y')			N	
Show_view_priv	enum('N','Y')			N	
Create_routine_priv	enum('N','Y')			N	
Alter_routine_priv	enum('N','Y')			N	
Create_user_priv	enum('N','Y')			N	
Event_priv	enum('N','Y')			N	
Trigger_priv	enum('N','Y')			N	
ssl_type	enum(", 'ANY', 'X509', 'SPECIFIED')				
ssl_cipher	blob			NULL	
x509_issuer	blob			NULL	
x509_subject	blob			NULL	
max_questions	int(11) unsigned			0	
max_updates	int(11) unsigned			0	
max_connections	int(11) unsigned			0	
max_user_connections	int(11) unsigned			0	

<sup>1</sup>Эта и все, описанные ниже команды добавлены начиная с версии 5.12

Изначально эта таблица содержит пользователя root без пароля. По умолчанию root может входить с любого хоста, имеет все привилегии и доступ ко всем базам данных. Также в таблице содержится запись для пользователя '%'.  
В БД MYSQL содержатся таблицы, называемых таблицами привилегий. Система привилегий будет подробно рассмотрена в следующих работах, а пока вы можете выполнить команды на добавления своего пользователя:

Для добавления нового пользователя **your\_name**, можно выполнить следующие операторы языка (Insert):

```
Insert into user (host, user, password, ssl_cipher2, x509_issuer, x509_subject )  
values ('localhost', 'your_name', password('your_pass'), '', '', '');
```

Выполнением команды

```
Select host, user, password from user;
```

Мы выводим перечисленные поля в виде таблицы

Host	User	Password
%	root	456g879k34df9

Если необходимо выделить все столбцы таблицы, то необходимо набрать \* в качестве аргумента команды *select*.

Чтобы изменения вступили в силу нужно перезагрузить сервер, предварительно закончив текущий сеанс работы командой *quit*.

```
mysqladmin -u root reload (эта команда перезагружает сервер)
```

После установки пароля для пользователя нужно перезагрузить сервер командой *mysqladmin reload*, чтобы изменения вступили в силу. После этого можно попробовать войти снова:

```
Mysql/bin/mysql -u your_name -p mysql  
Enter password:*****
```

Если же после этой операции вы не получите приглашение ко входу, то необходимо будет повторить вход в сервер под учетной записью **ROOT** и

---

<sup>2</sup> Атрибуты *ssl\_cipher<sup>2</sup>, x509\_issuer, x509\_subject* обязательны для заполнения для версии сервера 5.12

назначить необходимые права. Т.о., недостаточно добавить сведения о пользователе в системную БД, дополнительно необходимо назначить права пользователю, после чего можно начинать настраивать таблицы привилегий, вводить новых пользователей, создавать базы данных и таблицы, то есть делать все то, что называется администрированием. Назначить права можно указанием инструкцией **INSERT** для заполнения соответствующие привилегии

```
Мysql/bin/mysql -u root
```

И выполнить следующий запрос к БД:

```
Мysql>USE MYSQL;
```

```
Мysql>GRANT ALL PRIVILEGES ON *.* TO  
'your_name'@'localhost3' IDENTIFIED BY 'your_pass' WITH GRANT  
OPTION;
```

```
Мysql>FLUSH PRIVILEGES;
```

Если пароль был случайно забыт, чтобы его задать по новой, придется стереть файлы mysql.frm mysql.MYI и mysql.MYD из папки с базами данных, затем запустить скрипт mysql\_install\_db и повторить все по новой. Можно воспользоваться ключом MYSQL и ввести **--skip-grant-tables**, при этом все пароли будут имеет пустое поле.

Команда имеет вид **mysqld --skip-grant-tables**.

### **Пояснения:**

1. Команда insert вставляет данные в таблицу, не забывайте завершать команды ';'.  
2. При вводе пароля используйте функцию password(), иначе пароль работать не будет!

3. Все пароли шифруются mysql, поэтому в поле Password вы видите абракадабры. Это делается в целях безопасности.

4. Не есть хорошей практикой назначать привилегии пользователям в таблице user, так как в этом случае они являются глобальными и распространяются на все базы данных. Предоставляйте привилегии каждому пользователю к конкретной базе данных в таблице db, которая будет рассмотрена далее.

---

<sup>3</sup> Для случая, если работаете на том же компьютере где запущен сервер БД

5. При задании имени хоста для входа через сеть рекомендуется явно указывать полное имя хоста, а не '%'. В приведенном выше примере пользователю mary разрешается вход на сервер со всех машин домена tomsk.ru. Можно также указывать IP-адреса машин и маски подсетей для большей безопасности.

## Таблица Db

Определяет к каким базам данных каким пользователям и с каких хостов разрешен доступ. В этой таблице можно предоставлять каждому пользователю доступ к базам данных и назначать привилегии. Если выполнить команду *show columns from db*; получим следующее:

Таблица 1 - Структура таблицы Db

Field	Type	Null	Key	Default	Extra
Host	char(60)		PRI		
Db	char(32)		PRI		
User	char(16)		PRI		
Select_priv	char(1)			N	
Insert_priv	char(1)			N	
Update_priv	char(1)			N	
Delete_priv	char(1)			N	
Create_priv	char(1)			N	
Drop_priv	char(1)			N	

По умолчанию, все привилегии установлены в 'N'. Например, предоставим юзеру mary доступ к базе данных mysql и дадим ему привилегии **select**, **insert** и **update** (описание основных команд mysql будет дано в следующих лабораторных работах, сейчас ваша цель увидеть, как работают таблицы привилегий).

Для справки:

```
Insert into db (host, user, db, select_priv, insert_priv, update_priv)
Values ('localhost', 'your_name', mysql, 'Y', 'Y', 'Y');
```

Привилегии, устанавливаемые в таблице db, распространяются только на базу данных library. Если же установить эти привилегии в таблице user, то они будут распространяться и на другие базы данных, даже если доступ к ним и не установлен явно.

## Таблица Host

Таблица host используется для расширения диапазона доступа в таблице db. К примеру, если доступ к какой-либо базе данных должен быть предоставлен более чем одному хосту, тогда следует оставить пустой колонку host в таблице db, и внести в таблицу host необходимые имена хостов. Выполним команду

```
show columns from host;
```

**Таблица 2 - Структура таблиц Host**

Field	Type	Null	Key	Default	Extra
Host	char(60)		PRI		
Db	char(32)		PRI		
Select_priv	char(1)			N	
Insert_priv	char(1)			N	
Update_priv	char(1)			N	
Delete_priv	char(1)			N	
Create_priv	char(1)			N	
Drop_priv	char(1)			N	

Как видно из таблицы, здесь также можно задавать привилегии для доступа к базе данных. Они обычно редко используются без необходимости. Все привилегии доступа нужно задавать в таблице db для каждого пользователя, а в таблице host только перечислить имена хостов. Сервер читает все таблицы, проверяет имя пользователя, пароль, имя хоста, имя базы данных, привилегии. Если в таблице db привилегии select, insert установлены в 'Y', а в таблице host в 'N', то в итоге юзер все равно получит 'Y'. Чтобы не вносить путаницы, лучше назначать привилегии в таблице db.

Эти 3 таблицы являются основными. В новых версиях MySQL, начиная с 3.22 добавлены еще 2 таблицы- tables\_priv и columns\_priv, которые позволяют задать права доступа к определенной таблице в базе данных и даже к определенной колонке. Они работают подобно таблице db, только ссылаются на таблицы и колонки. Также, начиная с версии 3.22 можно использовать команду GRANT для предоставления доступа к базам данных, таблицам и колонкам таблиц, что избавляет от необходимости вручную модифицировать таблицы db, tables\_priv и columns\_priv. Команда GRANT будет подробно рассмотрена в следующих разделах.

## Привилегии, предоставляемые MySQL

Таблица 3 - Привилегии пользователя<sup>4</sup>

Привилегия	Колонка	Где используется
<b>select</b>	Select_priv	таблицы
<b>insert</b>	Insert_priv	таблицы
<b>Update</b>	Update_priv	таблицы
<b>delete</b>	Delete_priv	таблицы
<b>index</b>	Index_priv	таблицы
<b>alter</b>	Alter_priv	таблицы
<b>create</b>	Create_priv	БД, таблицы, индексы
<b>drop</b>	Drop_priv	БД или таблицы
<b>grant</b>	Grant_priv	БД или таблицы
<b>References</b>	References_priv	БД или таблицы
<b>reload</b>	Reload_priv	администрирование сервера
<b>Shutdown</b>	Shutdown_priv	администрирование сервера
<b>Process</b>	Process_priv	администрирование сервера
<b>file</b>	File_priv	доступ к файлам на сервере

### Основные утилиты MySQL.

В состав дистрибутива MySQL входят следующие утилиты:

- mysql
- mysql
- Mysqladmin
- mysqlaccess
- mysqlshow
- mysqldump
- isamchk

### Mysqladmin

Утилита для администрирования сервера. Может использоваться администратором, а также некоторыми пользователями, которым предоставлены определенные привилегии, например – **Reload\_priv**, **Shutdown\_priv**, **Process\_priv** и **File\_priv**. Данная команда может использоваться для создания баз данных, изменения пароля пользователя(администратор может изменить пароль любому пользователю, а рядовой пользователь – только свой собственный), перезагрузки и остановки сервера, просмотра списка процессов, запущенных на сервере. Mysqladmin поддерживает следующие команды:

---

<sup>4</sup> Даны для справки, для текущей версии сервера может быть существенно расширены

**Таблица 4 - Опции команды MySQLadmin**

Create [database_name]	Создает базу данных
Drop [database_name]	Удаляет базу данных и все таблицы в ней
Reload	Перезагружает сервер
Shutdown	Останавливает работу сервера MySQL
Processlist	Выводит список процессов на сервере
Status	Выводит сообщение о статусе сервера

Пример использования mysqladmin для изменения пароля:

*mysqladmin -u your\_name password your\_pass*

Следует заметить, что в случае использования mysqladmin для установки пароля, не требуется использование функции password(). MySQLadmin сам заботится о шифровании пароля.

### **MySQLaccess**

Используется для проверки привилегий пользователя для доступа к конкретной базе данных. Общий синтаксис:

*mysqlaccess [host] [user] [db] опции*

Полезная утилита для проверки прав доступа пользователя, если он получает сообщение Access denied, при попытке соединиться с базой данных.

**Опции:**

**Таблица 5 - Опции команды MySQLAccess**

-?, --help	Справка
-u, --user=[username]	Имя пользователя
-p, --password=[password]	Пароль пользователя
-h, --host=[hostname]	Имя хоста для проверки прав доступа
-d, --db=[dbname]	Имя базы данных для проверки прав доступа
-U, --superuser=[susername]	Имя суперпользователя(root)
-P, --spassword=[spassword]	Пароль администратора
-b, --brief	Выводит краткие сведения о таблице

### **MySQLshow**

Используется, чтобы показать, с какими базами данных работает сервер, какие таблицы содержит каждая БД и какие колонки есть в каждой таблице. Синтаксис:

*mysqlshow [опции] [database [table [field]]]*

MySQLshow может использовать следующие параметры:

**Таблица 6 - Параметры команды MySQLshow**

-?, --help	Справка
-h, --host=[hostname]	Имя сервера
-k, --key	Показать ключи для таблицы
-p, --password=[password]	Пароль пользователя
-u, --user=[username]	Имя пользователя
-p, --port=[port]	Порт для связи
-V, --version	Вывести информацию о версии



Если ввести `mysqlshow` без аргументов, будут показаны все базы данных, если указать имя БД, будут показаны все таблицы в ней.

Команды

*`mysqlshow`*

*`mysqlshow mysql`*

## **MySqlDump**

Программа `mysqldump` используется для создания дампа содержания базы данных MySQL. Она пишет инструкции SQL в стандартный вывод. Эти инструкции SQL могут быть переназначены в файл. Можно резервировать базу данных MySQL, используя `mysqldump`, но при этом Вы должны убедиться, что в этот момент с базой данных не выполняется никаких других действий. А то `mysqldump` Вам такого нарезервирует...

Программа `mysqldump` поддерживает следующие параметры (Вы можете использовать короткую или подробную версию):

**Таблица 7 - Опции команды MySQLdump**

<code>-#, --debug=[options]</code>	<b>Вывести в протокол отладочную информацию. В общем виде 'd:t:o, filename'.</b>
<code>-, --help</code>	Справка.
<code>-c, --complete-insert</code>	Генерируйте полные инструкции insert (не исключая значений, которые соответствуют значениям столбца по умолчанию).
<code>-h, --host=[hostname]</code>	Соединиться с сервером hostname.
<code>-d, --no-data</code>	Экспорт только схемы информации (исключая данные).
<code>-t, --no-create-info</code>	Экспорт только данных, исключая информацию для создания таблицы. Противоположность <code>-d</code> .
<code>-p, --password=[password]</code>	Пароль пользователя, для соединения с сервером MySQL. Обратите внимание, что не должно быть пробела между <code>-p</code> и паролем.
<code>-q, --quick</code>	Не буферизовать результаты запроса, дампы выдать непосредственно к STDOUT.
<code>-u, --user=[username]</code>	Имя пользователя. Если не задано, используется текущий логин.
<code>-v, --verbose</code>	Вывести подробную информацию относительно различных стадий выполнения <code>mysqldump</code> .
<code>-P, --port=[port]</code>	Порт для связи.
<code>-V, --version</code>	Информация о версии.

Вы можете направить вывод `mysqldump` в клиентскую программу MySQL, чтобы копировать базу данных. ПРИМЕЧАНИЕ: Вы должны убедиться, что база данных не изменяется в это время, иначе Вы получите противоречивую копию!

Для справки:

```
mysqldump -u root -p mysql user>mysql-1.sql
```

```
mysqldump -u root mysql>mysql-2.sql
```

**Примечание** флаг `-p` используется в случае, если пользователь наделен паролем.

После выполнения этой команды у нас появился файл `mysql-1.sql` и `mysql-2.sql`. Загрузим их в текстовый редактор, чтобы поподробнее изучить, и, возможно, немного поправить.

### **Задание**

Запустите сервер MySQL. Зарегистрируйте своего пользователя в консольном приложении, задайте ему права.

С помощью утилиты `Mysqlshow` выполните команду на просмотр структуры и состав таблиц базы `Mysql`. Приведите в отчете её схему. С помощью утилиты `Mysqldump` получите полный дамп базы `Mysql` (данные и таблицы), а также отдельные дампы таблиц и данных.

### **Требования к оформлению отчета**

1. Титульный лист.
2. Название работы.
3. Тему, цель и задание к работе.
4. Снимки экрана (скриншоты) процесса разработки.
5. Выводы (что узнали, где можно применить полученные знания).