Granular Database Documentation

In this document I will be explaining how to connect to my MYSQL database, and check the access-privileges of 3 different users.

1.1 User List

The following table provides an overview of my server's users, who all have a different type of access to the main table, **'test_table'**. Underneath this, you will also see the details regarding IP and port.

Username	**Password**	**Permissions**
write_user	supersecretpass1	Can only write, not read (ALTER, DELETE, INSERT)
read_user	1ssaptercesrepus	Can only read, not write (SELECT)
read_write_user	sts1erspucseera	Can read and write (ALTER, SELECT, DELETE, INSERT)

Description	**Value**
IP Address	165.22.70.135
Port	3306

2.1 How to connect

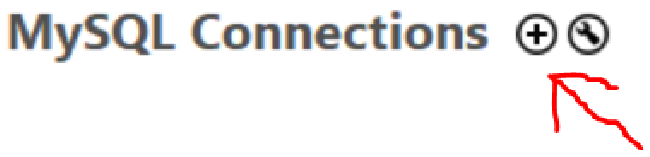
Next we will be covering how to access these users and test their privileges on our main table.

2.2 Install the MySQL Workbench

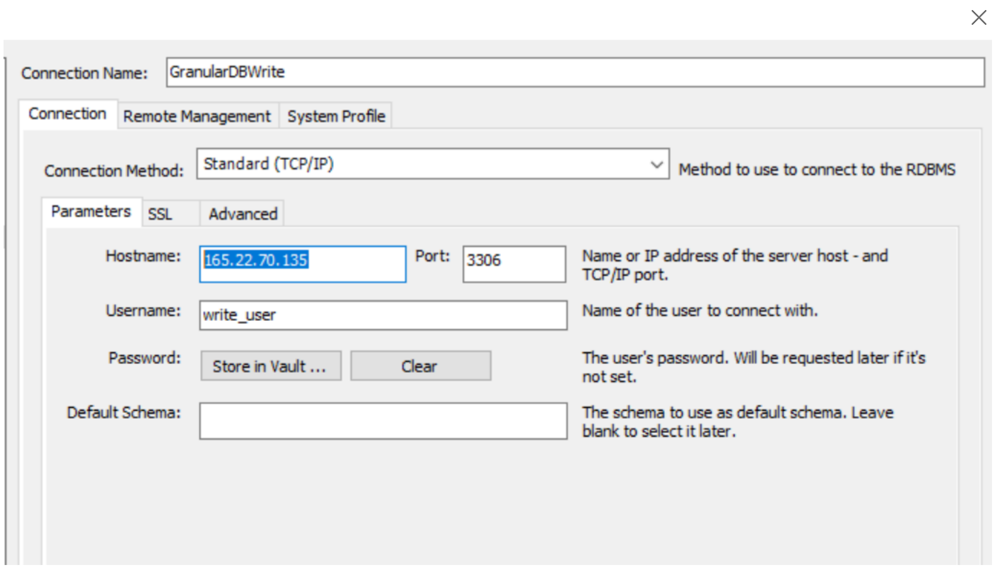
The easiest method to access our MySQL Database would be to download MySQL and its associated workbench. The installer can be found via this link.

2.3 Connect to the server from the workbench

Once you're in the workbench, click the small plus icon to create a new server-connection



From there, you should get a window that lets you type in the details of your connection. Make sure to set Hostname and port correctly as displayed in Cloud Configuration, and then type in the appropriate username from the User list. You should give it a Connection Name as well, but its content isn't important. Once you press "OK" in the bottom-right corner, the connection will be added to your list of server-connections. If you click it now, you will be prompted to give a password. Type in the correct password for the user.



Connection Name: GranularDBWrite

Connection Method: Standard (TCP/IP) — Method to use to connect to the RDBMS

Hostname: 165.22.70.135 Port: 3306 — Name or IP address of the server host - and TCP/IP port.

Username: write_user — Name of the user to connect with.

Password: — The user's password. Will be requested later if it's not set.

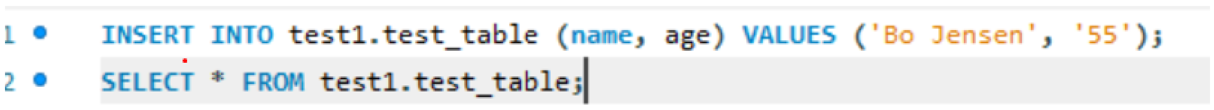
Default Schema: — The schema to use as default schema. Leave blank to select it later.

2.4 Execute SQL-statements to test user-privileges

Assuming you have now logged in to one of the three users and accessed the database, it is time to test out what an user can, and cannot do.

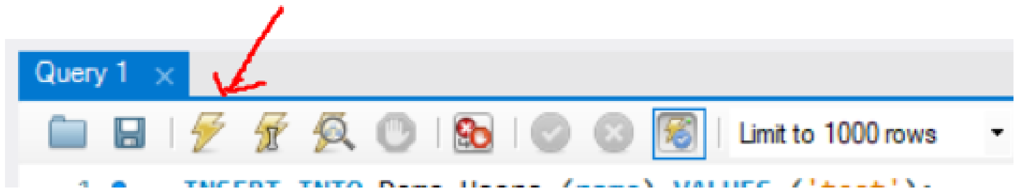
- **Username**: write_user; Can only write data, not read it - *Select-statements* will fail, while *Insert-statements* will succeed.
- **Username**: read_user; Can only read data, but never write it. *Select-statements* will succeed, *Insert-statements* will fail.
- **Username**: read_write_user; Can both read, and write data. Both *Select-statements* and *Insert-statements* will succeed.

Inside the database 'test1', I have a table named 'table_test' with some dummy data. Feel free to play around with it using some SQL-queries such as this:

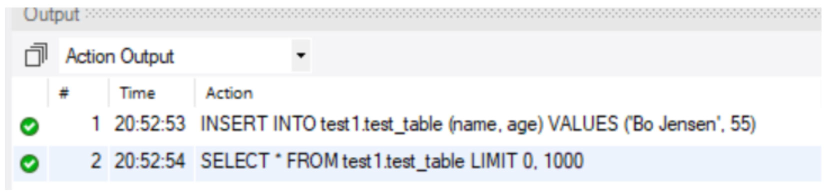


```
INSERT INTO test1.test_table (name, age) VALUES ('Bo Jensen', '55');
SELECT * FROM test1.test_table;
```

Make sure to hit the yellow arrow to execute the queries:



Watch the bottom to check for results:

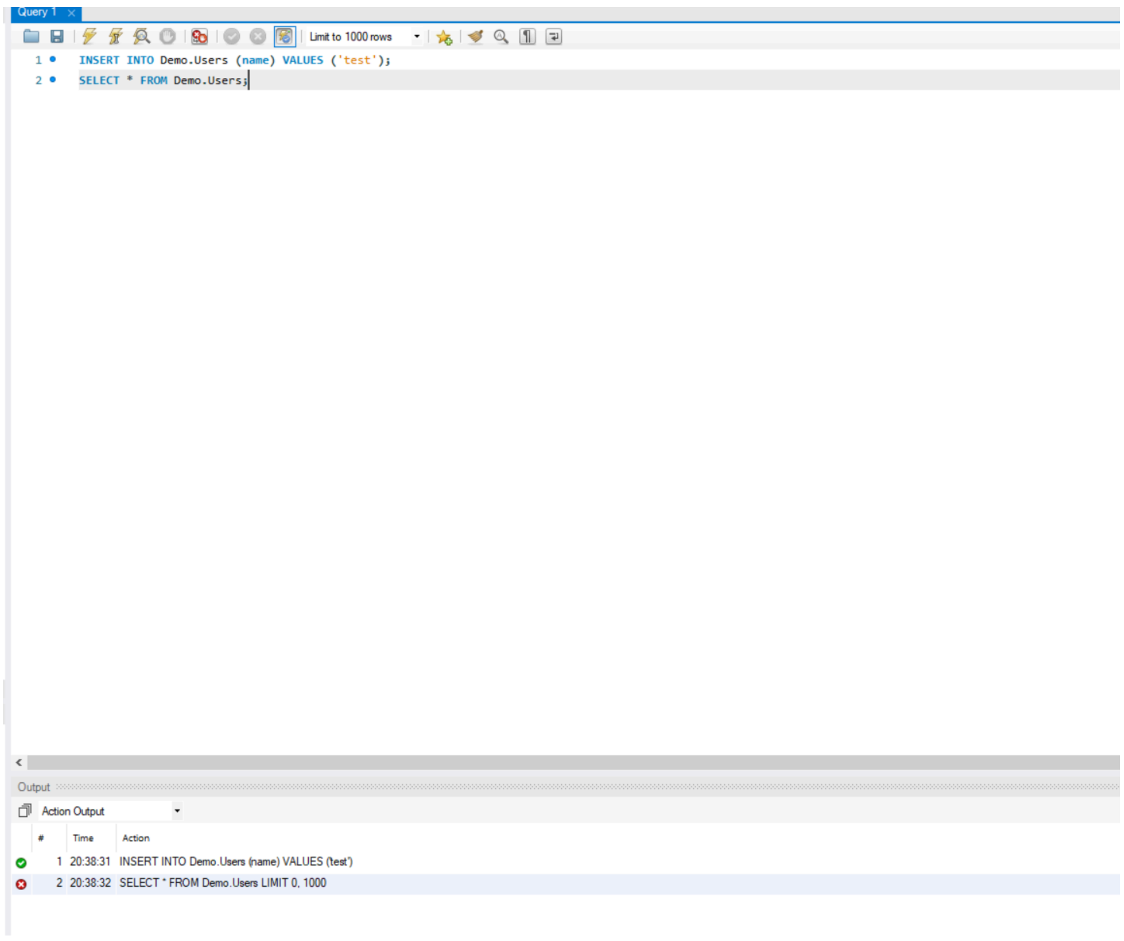


```
Output
Action Output
  #  Time      Action
  1  20:52:53  INSERT INTO test1.test_table (name, age) VALUES ('Bo Jensen', 55)
  2  20:52:54  SELECT * FROM test1.test_table LIMIT 0, 1000
```

Repeat the steps from 2.3 to 2.4 for the two remaining users.

Integration Screenshots

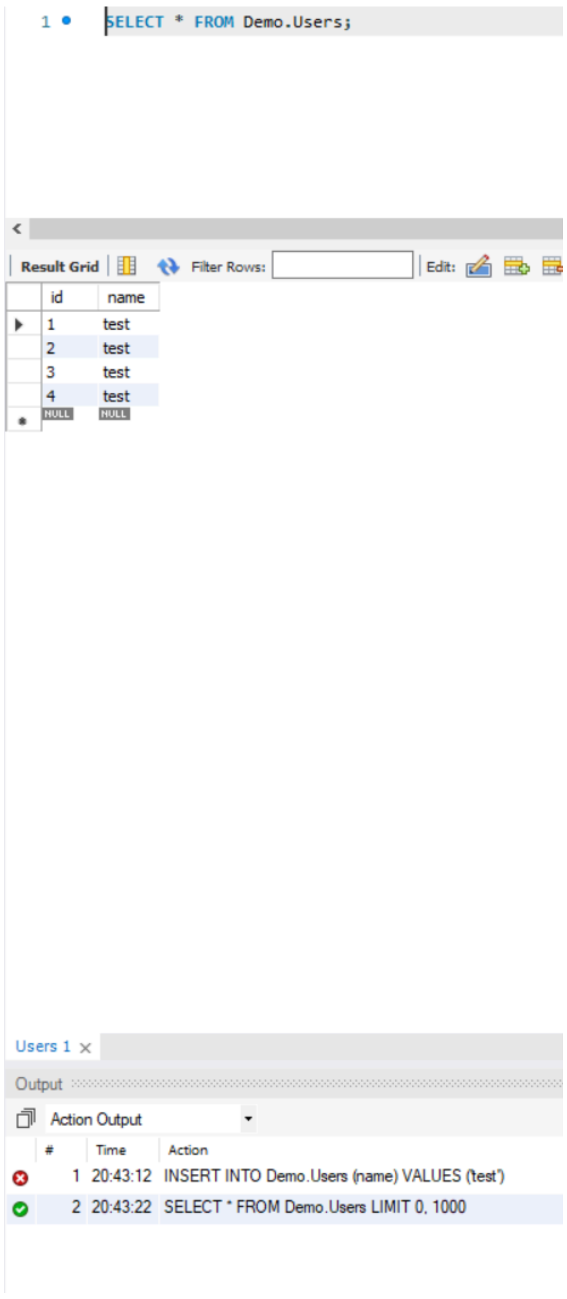
These are screenshots from when I tested my partner’s database.



```
INSERT INTO Demo.Users (name) VALUES ('test');
SELECT * FROM Demo.Users;
```

```
   #  Time      Action
   1  20:38:31  INSERT INTO Demo.Users (name) VALUES ('test')
   2  20:38:32  SELECT * FROM Demo.Users LIMIT 0, 1000
```

From the user with write-permissions.^

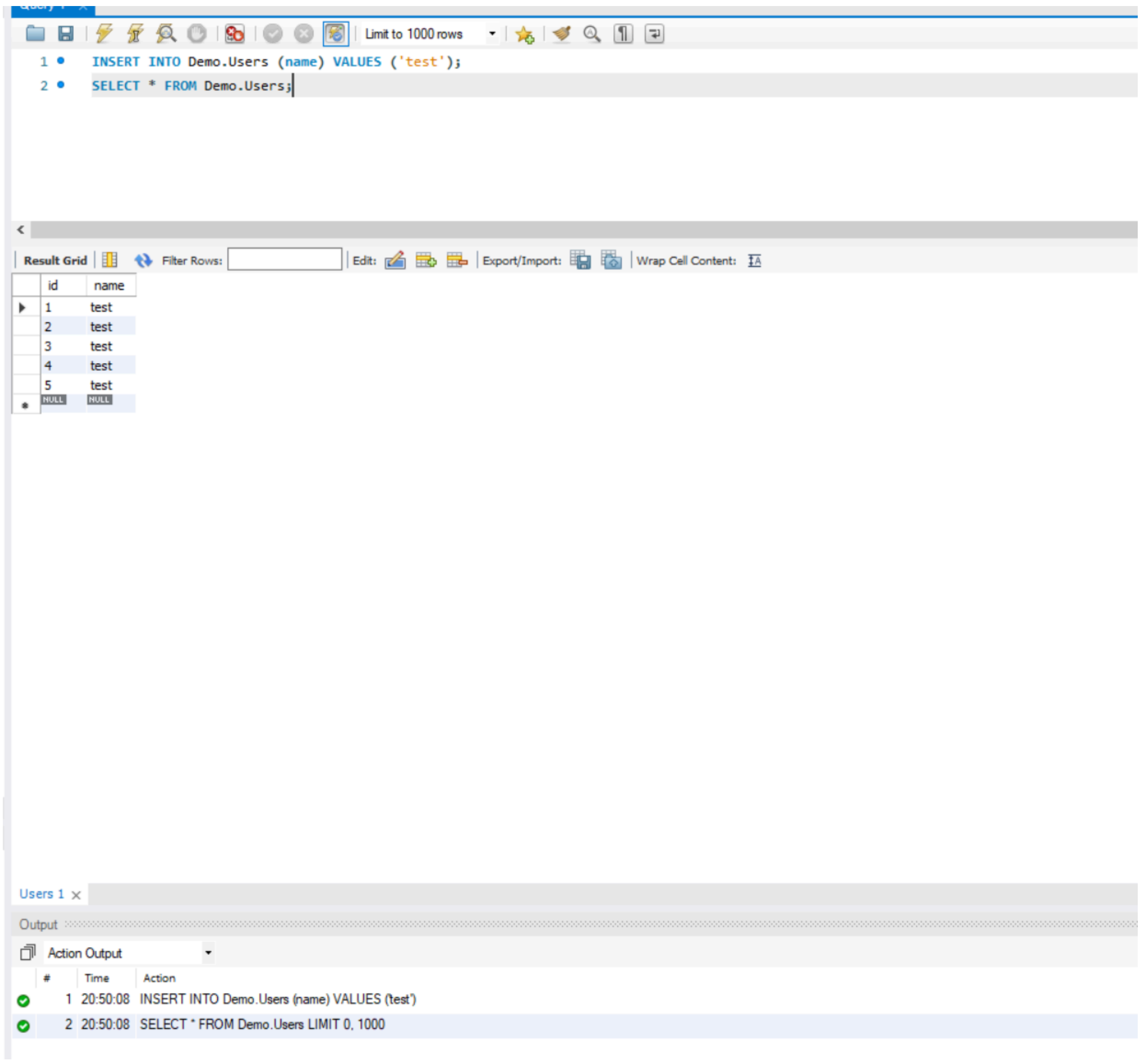


```
SELECT * FROM Demo.Users;
```

id	name
1	test
2	test
3	test
4	test
NULL	NULL

#	Time	Action
1	20:43:12	INSERT INTO Demo.Users (name) VALUES ('test')
2	20:43:22	SELECT * FROM Demo.Users LIMIT 0, 1000

From the user with read-permissions.^



```sql
INSERT INTO Demo.Users (name) VALUES ('test');
SELECT * FROM Demo.Users;
```

id	name
1	test
2	test
3	test
4	test
5	test
NULL	NULL

Output

Action Output

```
#  Time      Action
1  20:50:08  INSERT INTO Demo.Users (name) VALUES ('test')
2  20:50:08  SELECT * FROM Demo.Users LIMIT 0, 1000
```

From the user with read- and write permissions.^