Praxis: MariaDB:
Kommandozeilenclient „mysql“

```
mysql -p -u <username> -h <host>

z.B. mysql -p -u e5000160 -h dbserv.ba-nitsch.de
```

Den Kommandozeilenclient gibt es z.B. für Linux und für Windows. Unter Windows muss man zuvor ein Terminalfenster wie ‚cmd‘ oder ‚powershell‘ öffnen, um ihn aufzurufen.

MySQL / MariaDB:
Kommandozeilenclient

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1191
Server version: 5.5.62-0+deb8u1 (Debian)

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Begrüßungsbildschirm im Client

MariaDB Kommandozeilenclient
Datenbanken auflisten

```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| owncloud           |
| performance_schema |
| uebung             |
+--------------------+
```

show databases;

Databases oder auch Schemas (mariadb unterscheidet da nicht) kann man sich wie „Unterverzeichnisse“ vorstellen. So kann man in Database ‚owncloud‘ bzw. ‚uebung‘ je eine Tabelle „foo“ anlegen und beide unabhängig voneinander ändern.

MariaDB Kommandozeilenclient

Datenbank auswählen

use <database>;

MariaDB Kommandozeilenclient

Tabellen auflisten / erklären

```
MariaDB [uebung]> show tables;
+------------------+
| Tables_in_uebung |
+------------------+
| bar              |
| foo              |
+------------------+
2 rows in set (0,001 sec)

MariaDB [uebung]> explain foo;
+-------+---------+------+-----+---------+-------+
| Field | Type    | Null | Key | Default | Extra |
+-------+---------+------+-----+---------+-------+
| ID    | int(11) | YES  |     | NULL    |       |
+-------+---------+------+-----+---------+-------+
1 row in set (0,005 sec)

MariaDB [uebung]>
```

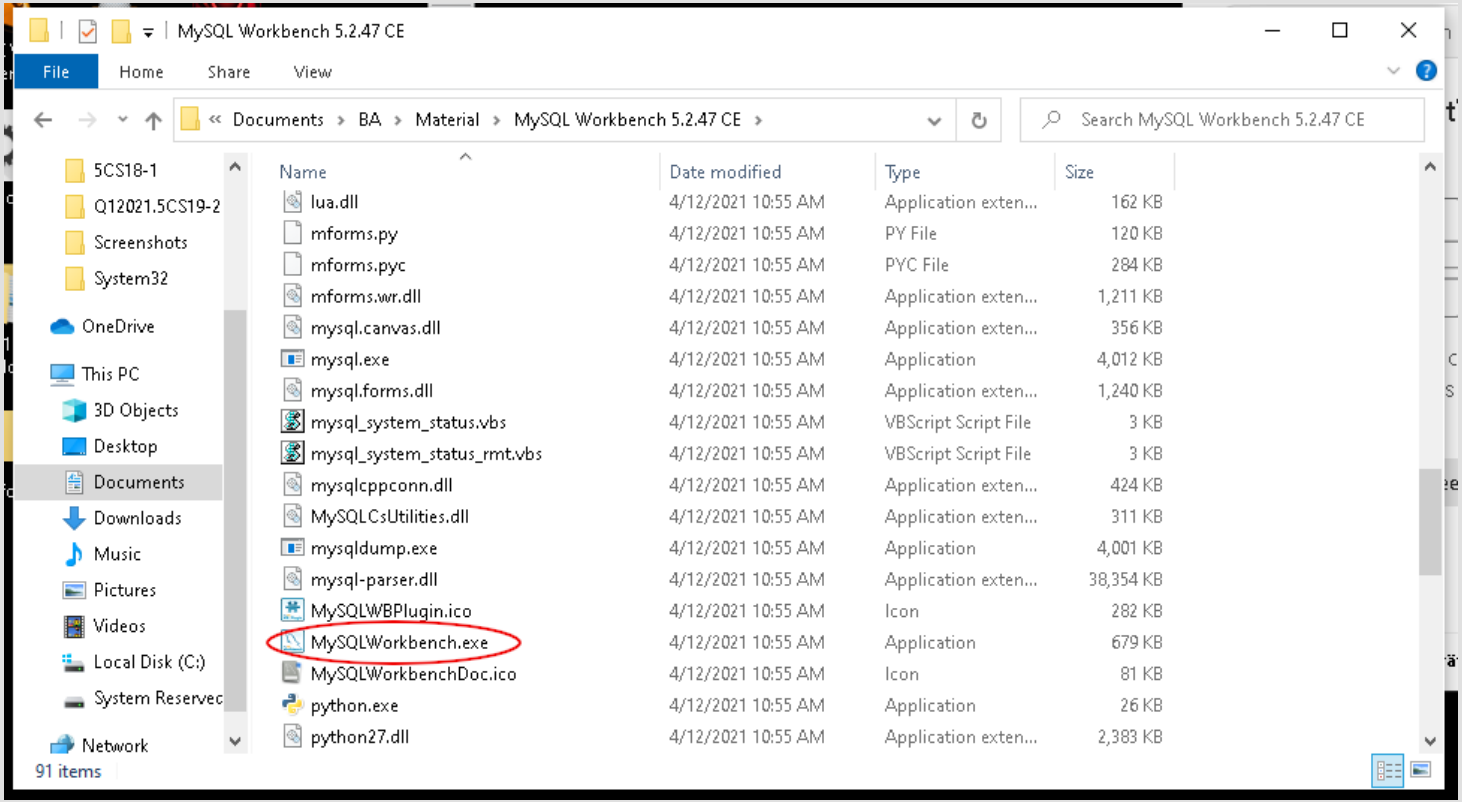
show tables;
explain <tabellenname>;

MariaDB:
MySQL Workbench - Client

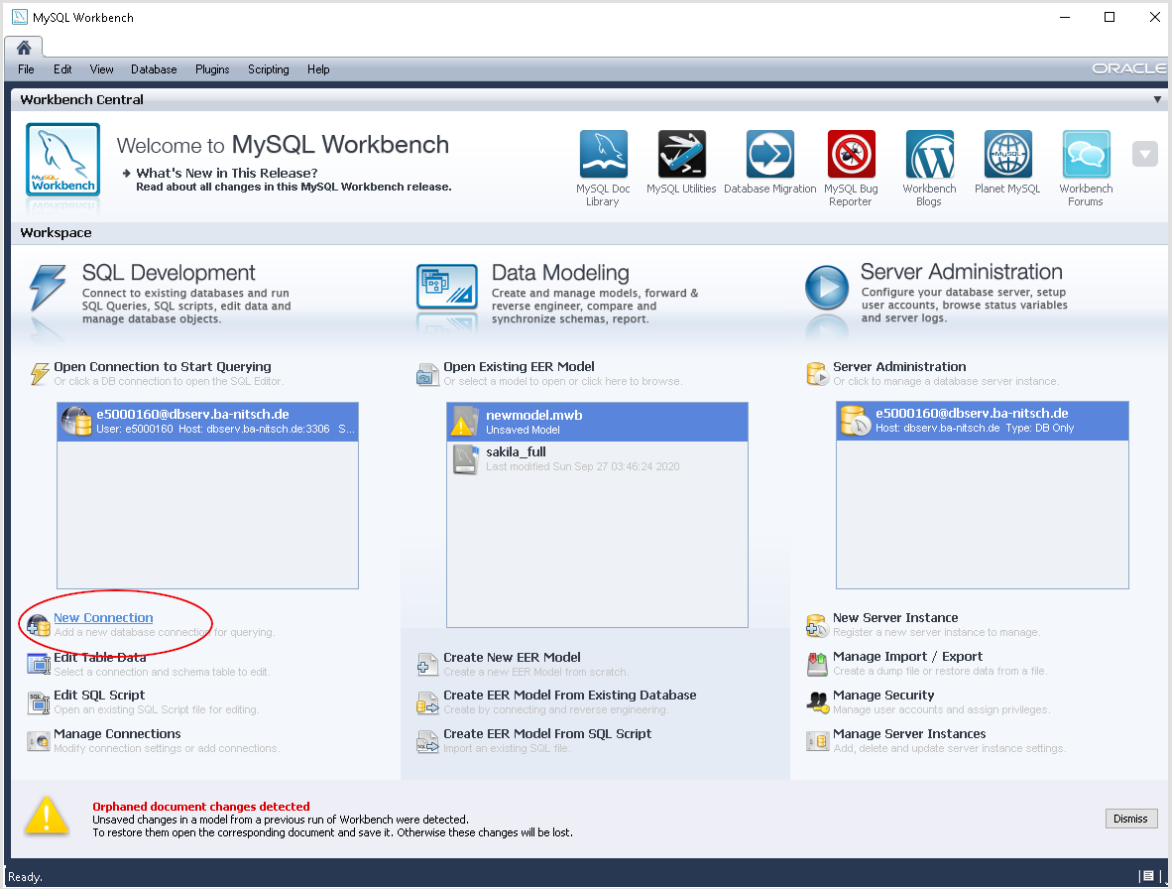
Auf dem Fileserver

\info-klausur\MySQL Workbench 5.2.47 CE

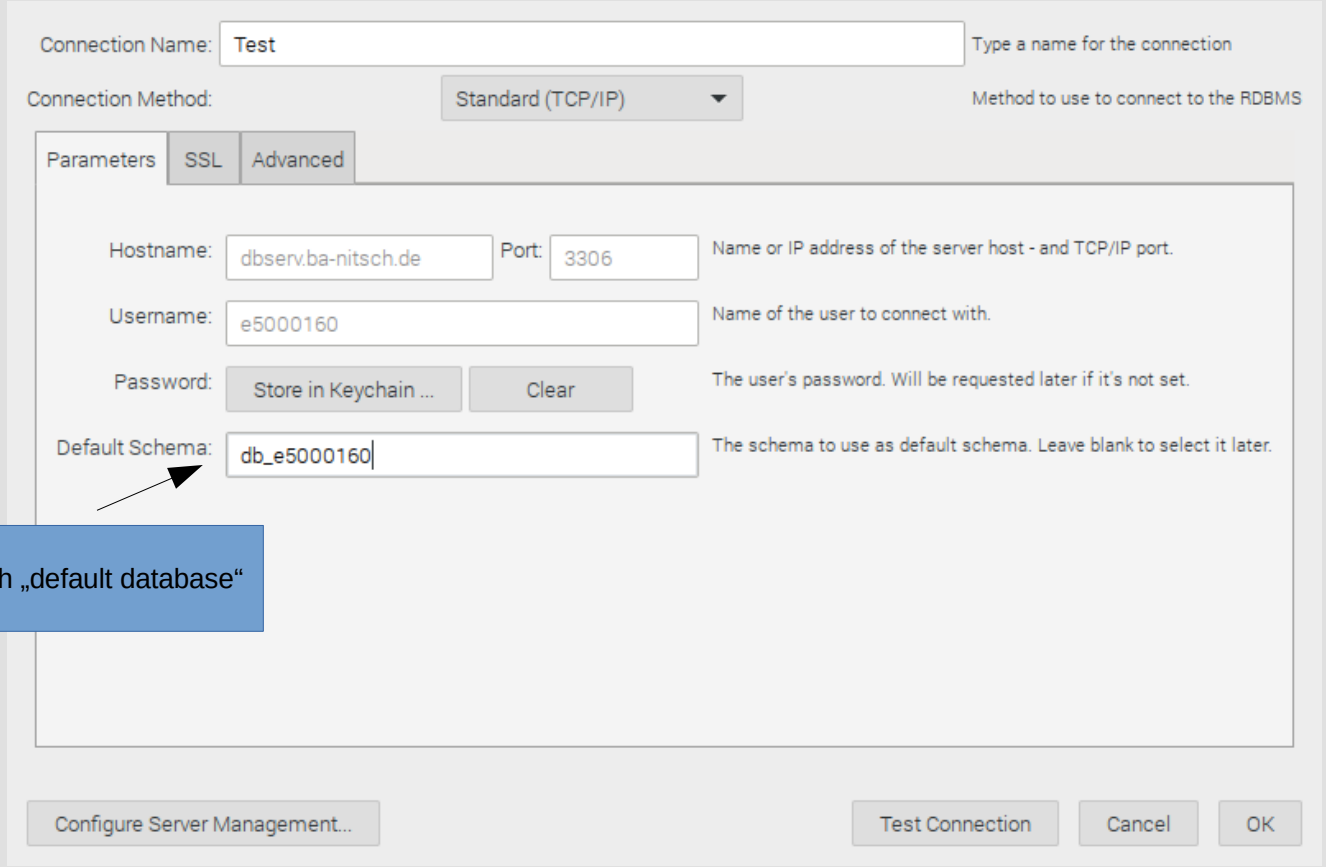
MySQL / MariaDB:
MySQL Workbench: Ordnerstruktur



MySQL Workbench:
Verbindung konfigurieren

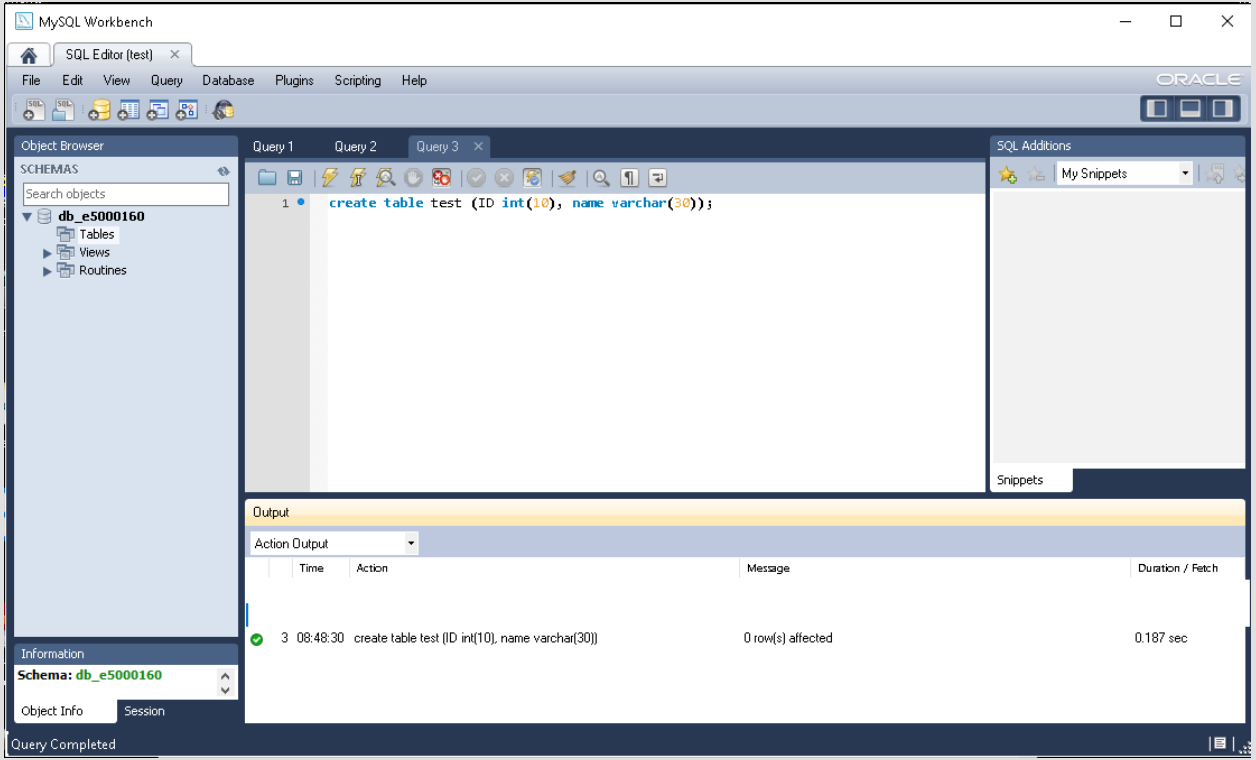


MySQL Workbench: Verbindung konfigurieren



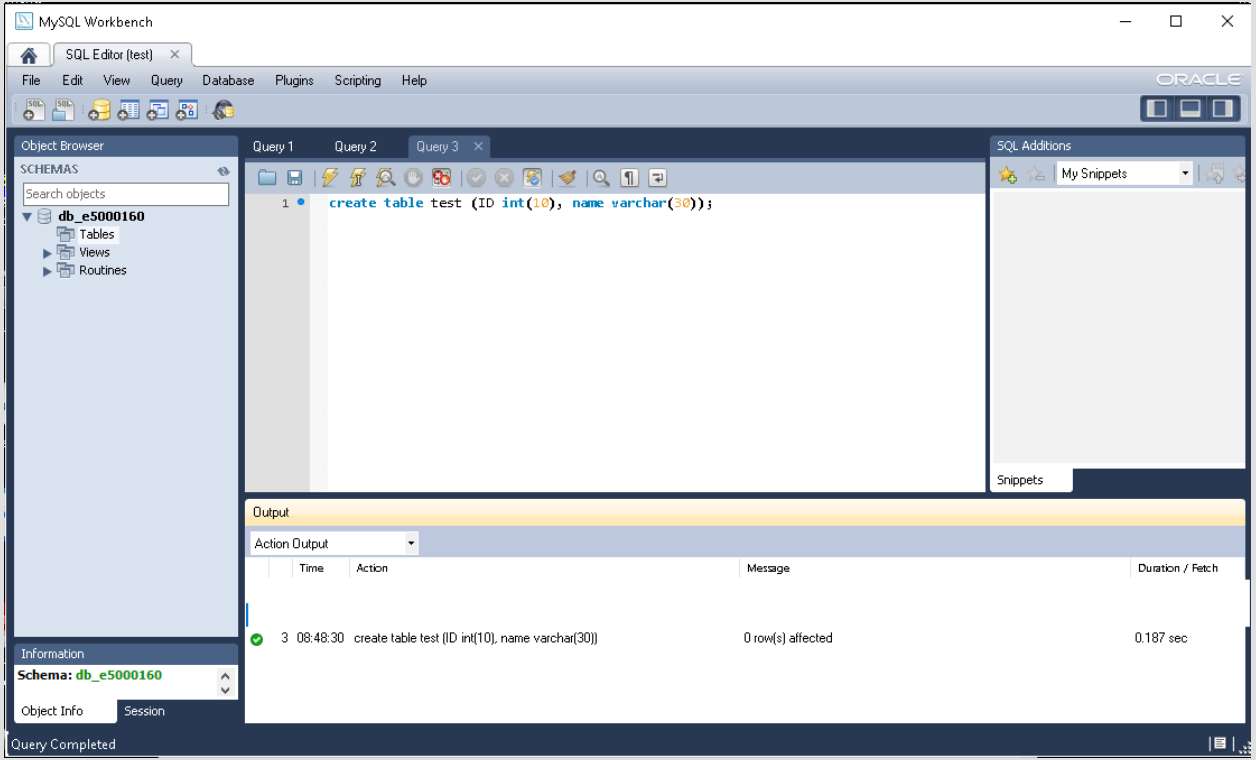
oder auch „default database“

MySQL Workbench:
MySQL Worbbench benutzen



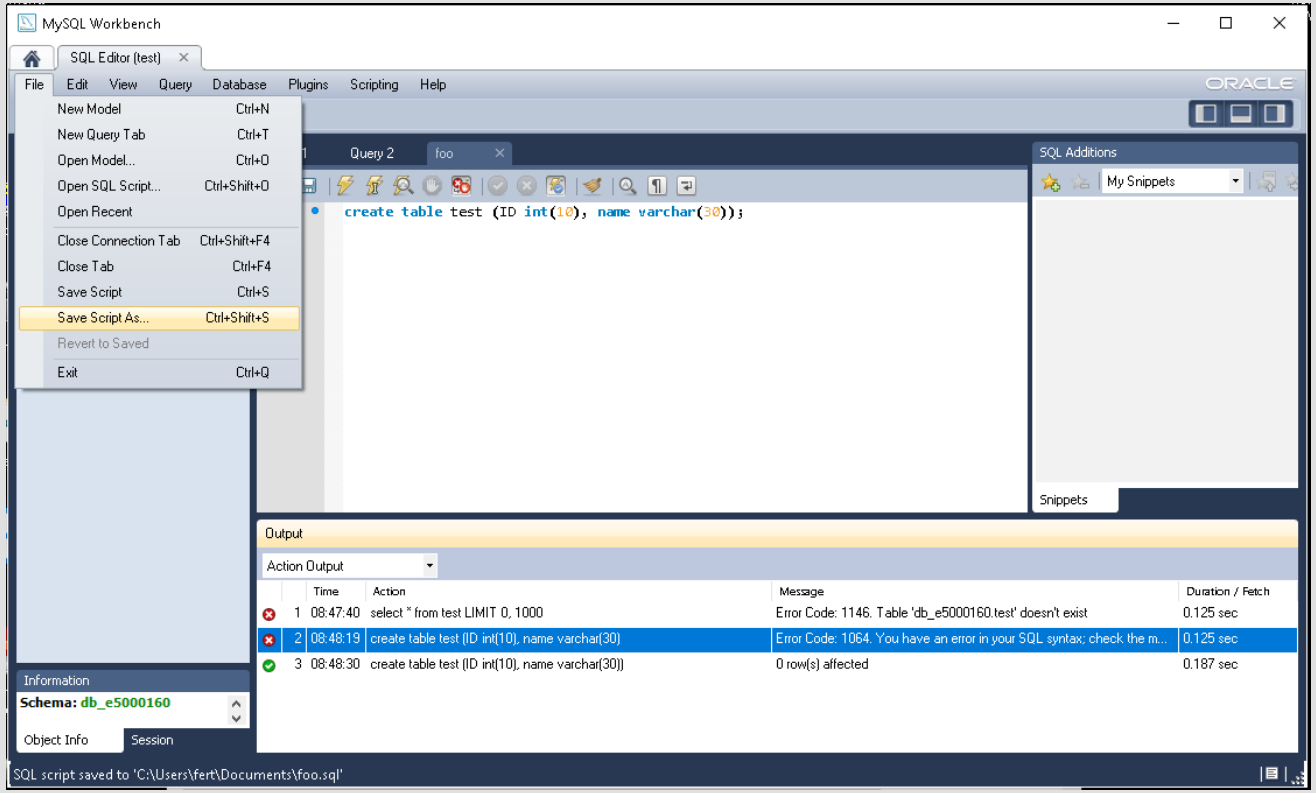
SQL-Konsole

MySQL Workbench:
MySQL Worbbench benutzen



SQL-Konsole
Kommandos ausführen
mit Ctrl-Shift-Enter
oder über Menü „Query“

MySQL Workbench:
MySQL Workbench benutzen



SQL-Kommandos speichern

MariaDB:
MySQL Workbench - Client

Der Datenbank-Server dbserv.ba-nitsch.de ist aus dem Netzwerk der BA ggf. nicht erreichbar (Firewall).

Daher: Alternative Konfiguration von MySQL Workbench mit SSH-Tunnel

MariaDB:
MySQL Workbench - Client

Nochmal dazu, was so ein SSH-Tunnel tut:

Bekannt ist evtl. dies: Man loggt sich per SSH auf einem anderen Rechner ein, hier als Nutzer e5000160 auf dem Rechner dbserv.ba-nitsch.de

ssh e5000160@dbserv.ba-nitsch.de

Der Vollständigkeit halber: Das @ hat nichts mit Emailadressen zu tun, sondern ist nur ein Trennzeichen zur Angabe des Nutzernamens.

MariaDB:
MySQL Workbench - Client

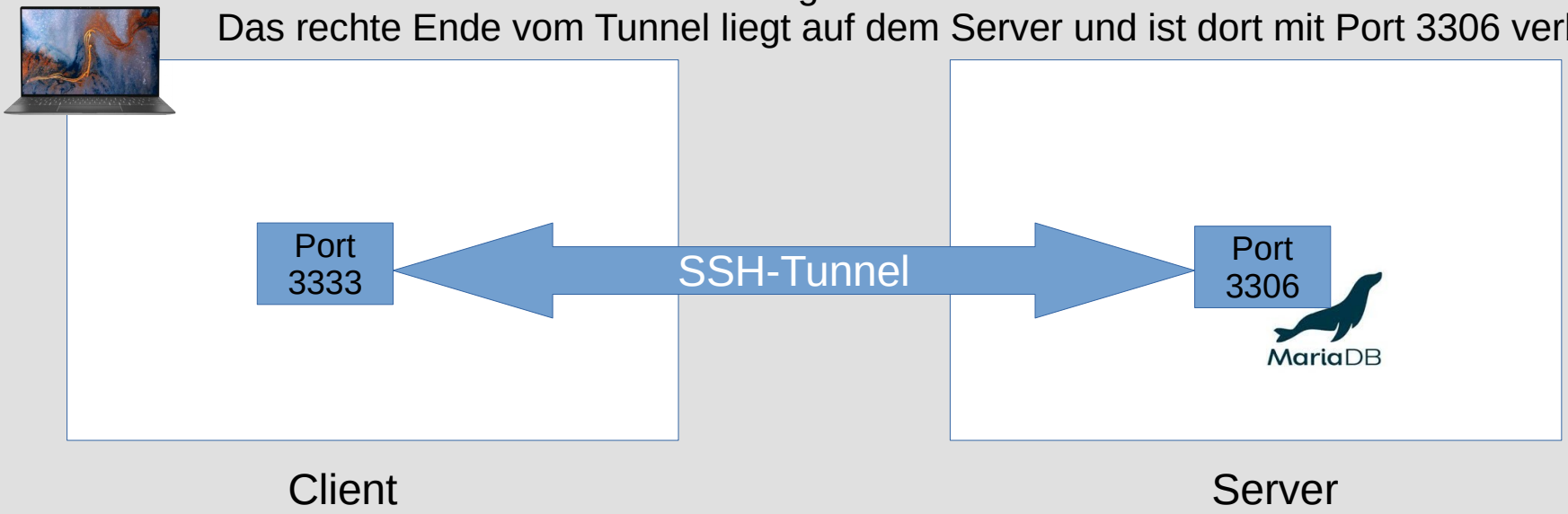
Aber SSH kann mehr: Wenn man sein Kommando noch etwas erweitert, wird sich eingeloggt und extra noch der SSH-Tunnel gebaut.

ssh -L 3333:127.0.0.1:3306 e5000160@ba-nitsch.de

MariaDB:
MySQL Workbench - Client

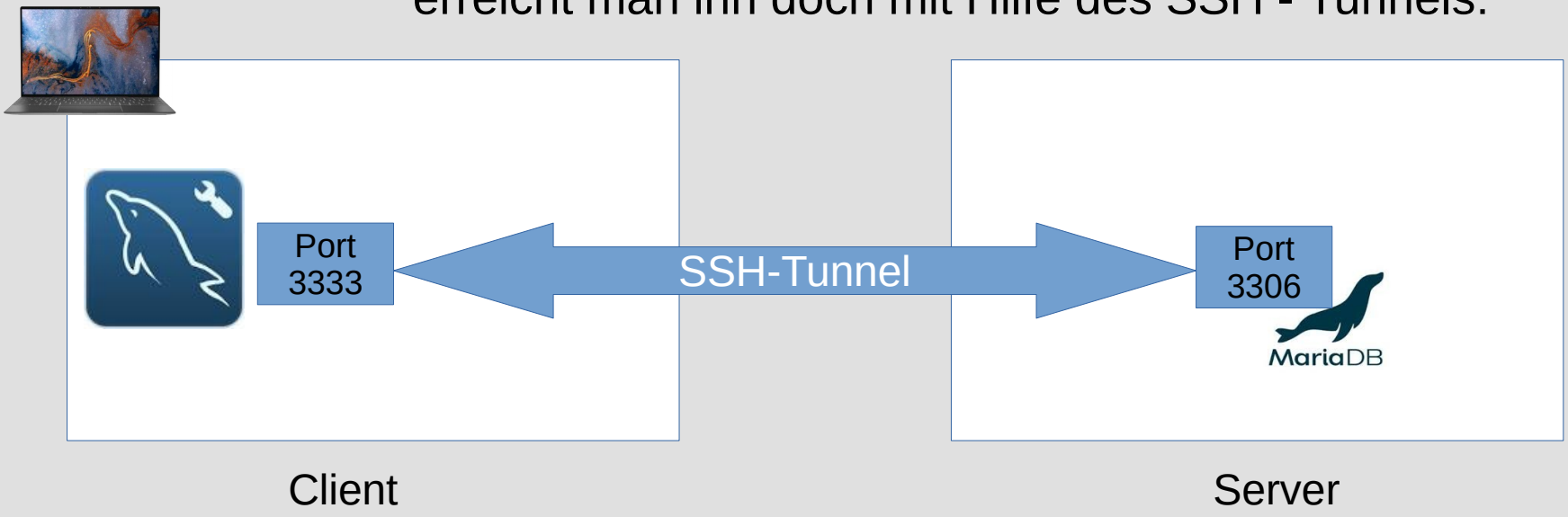
ssh -L 3333:127.0.0.1:3306 e5000160@ba-nitsch.de

Das linke Ende vom Tunnel liegt auf dem Client und lauscht dort auf Port 3333
Das rechte Ende vom Tunnel liegt auf dem Server und ist dort mit Port 3306 verbunden.



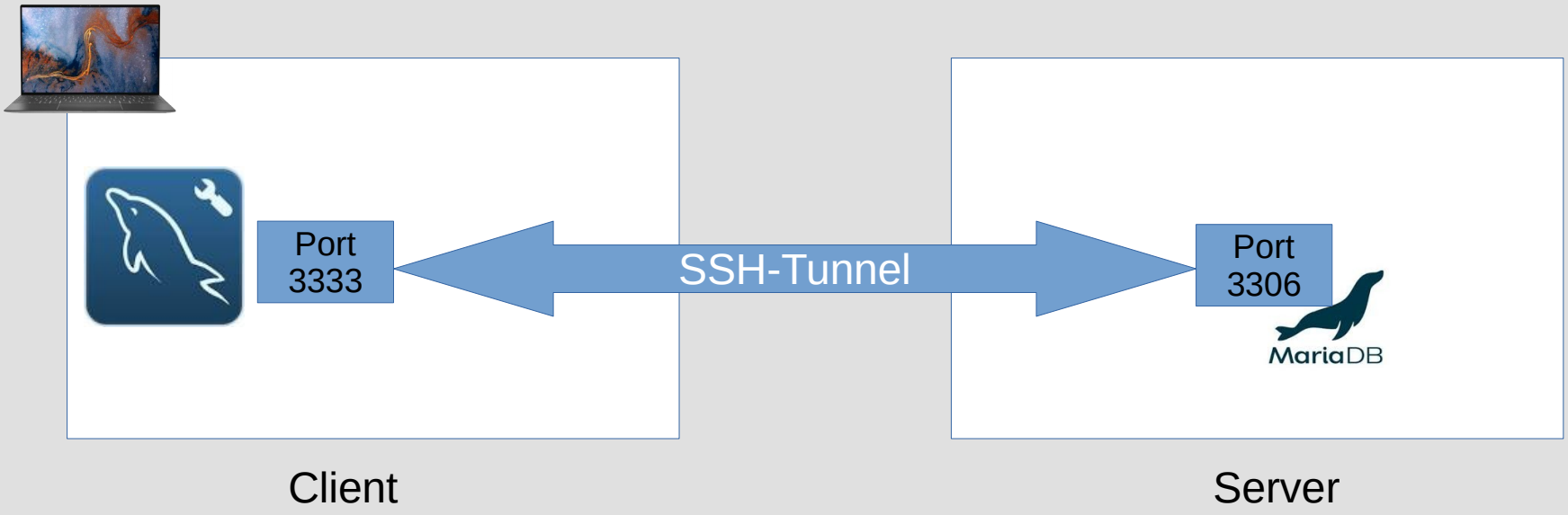
MariaDB:
MySQL Workbench - Client

Nun konfiguriert man sein MySQL Workbench so:
Datenbankserveradresse: 127.0.0.1, Datenbankserver-Port: 3333
Und obwohl auf dem Client gar kein Datenbankserver läuft,
erreicht man ihn doch mit Hilfe des SSH - Tunnels:



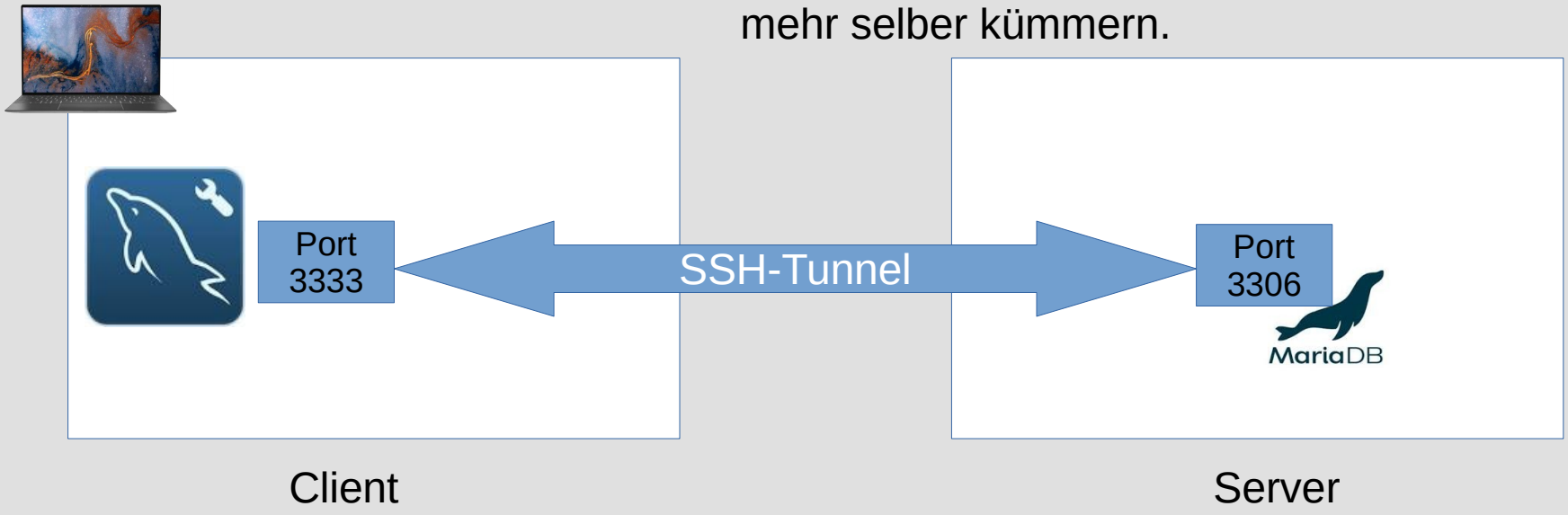
MariaDB:
MySQL Workbench - Client

Die Daten vom MySQL-Workbench gehen via Port 3333 auf
dem eigenen Rechner in den Tunnel und kommen auf dem Server wieder raus,
wo sie dann direkt in den Datenbankserver gelangen.

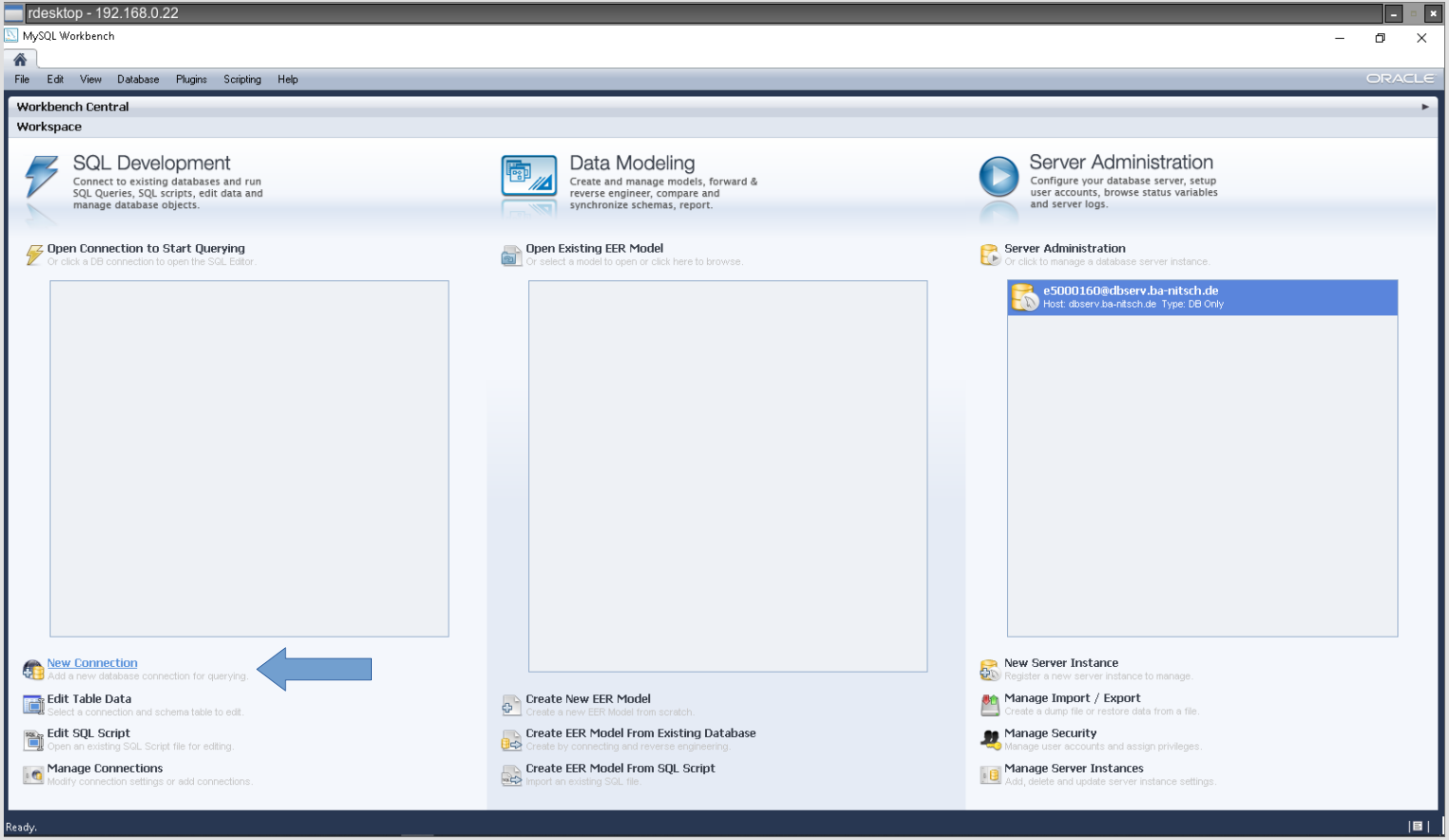


MariaDB:
MySQL Workbench - Client

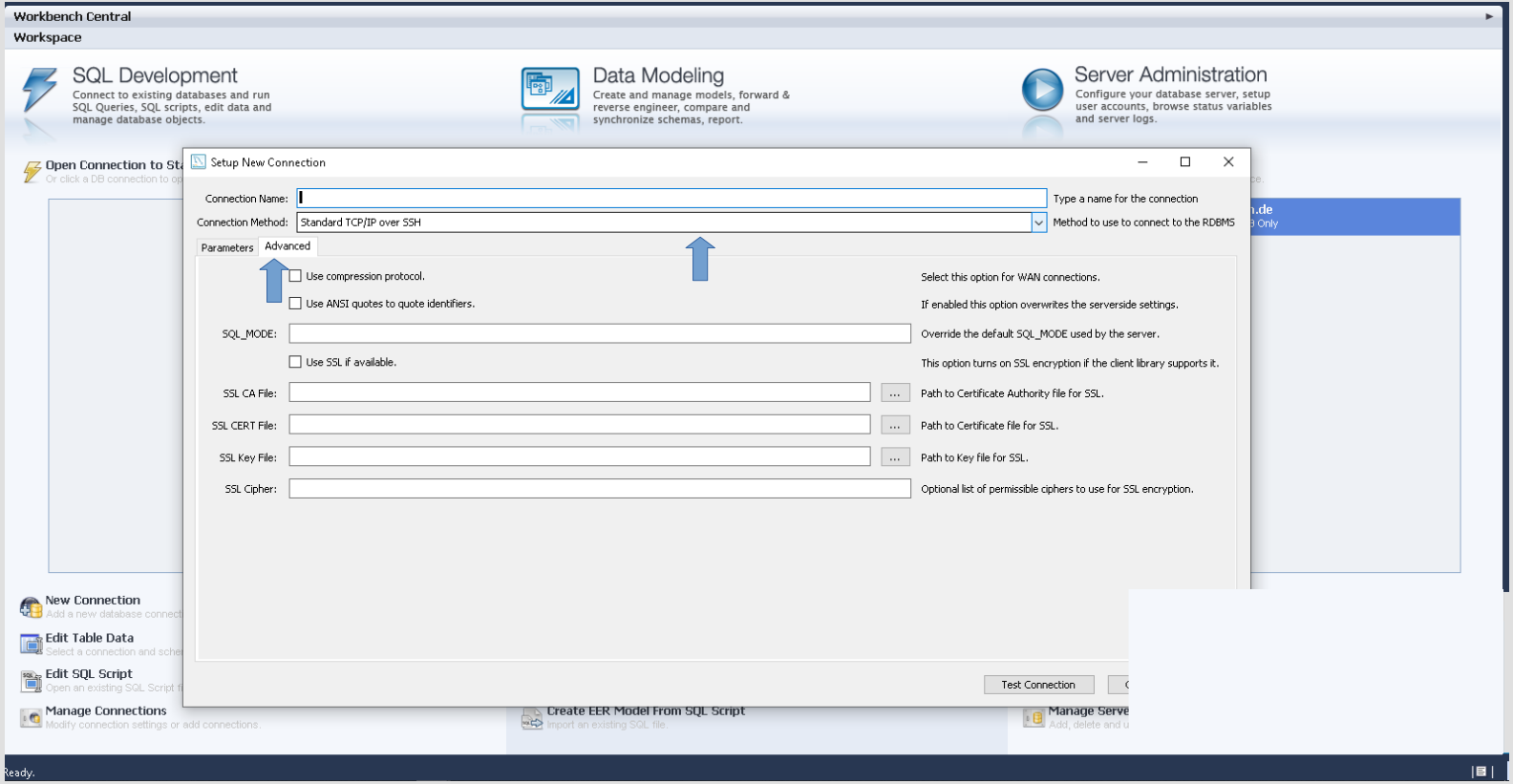
Dies hatten wir beim letzten Mal gemacht.
Zum Glück kann MySQL Workbench solche SSH-Tunnel selber aufbauen,
man muss nur noch Benutzernamen und Passwort konfigurieren.
Der Rest geht dann automatisch, auch um den lokalen Port 3333 muss man sich nun nicht mehr selber kümmern.



MariaDB:
MySQL Workbench - Client

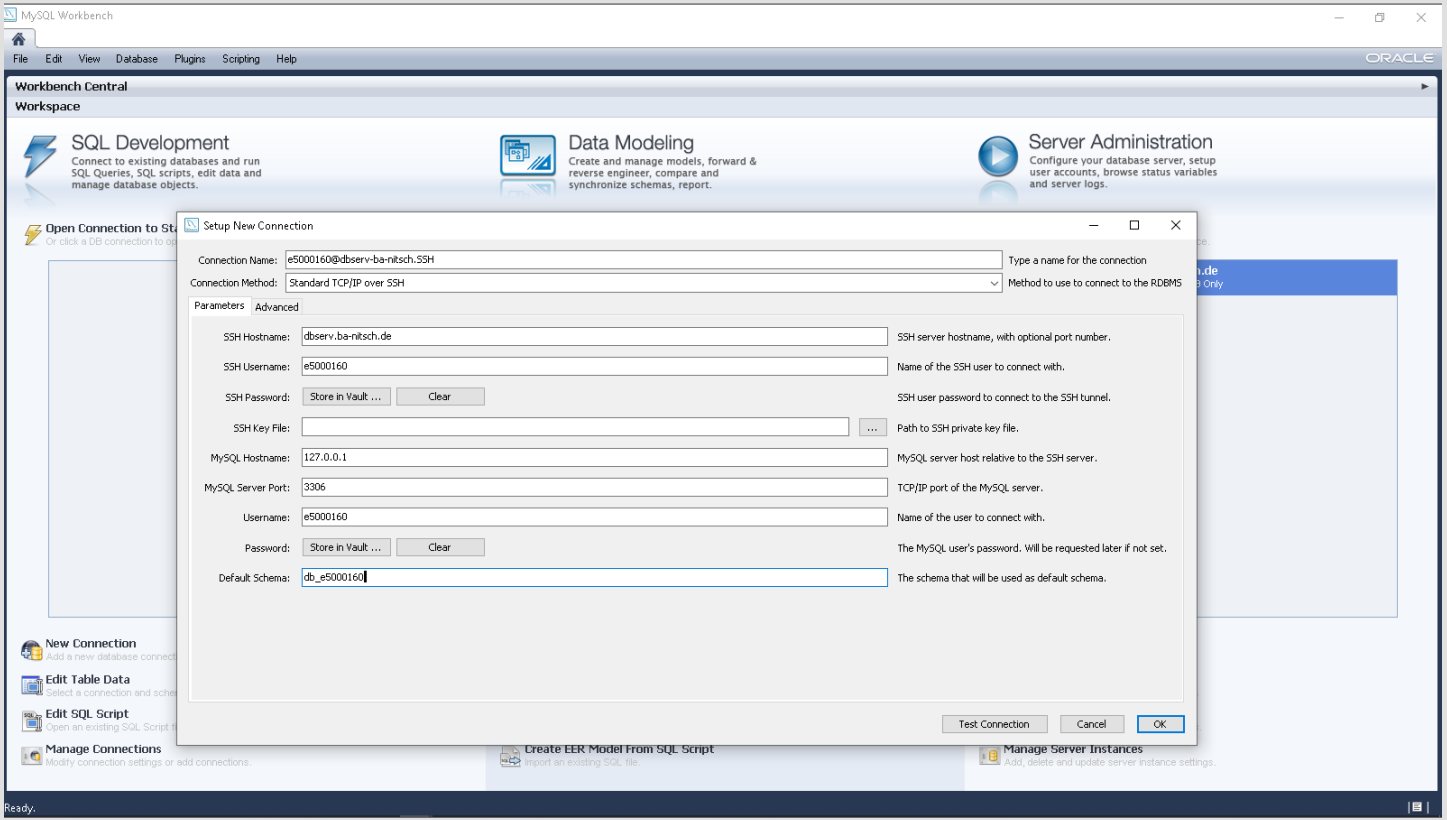


MariaDB:
MySQL Workbench - Client



Tab „Advanced“ und dort „Standard TCP/IP over SSH“ wählen

MariaDB:
MySQL Workbench - Client

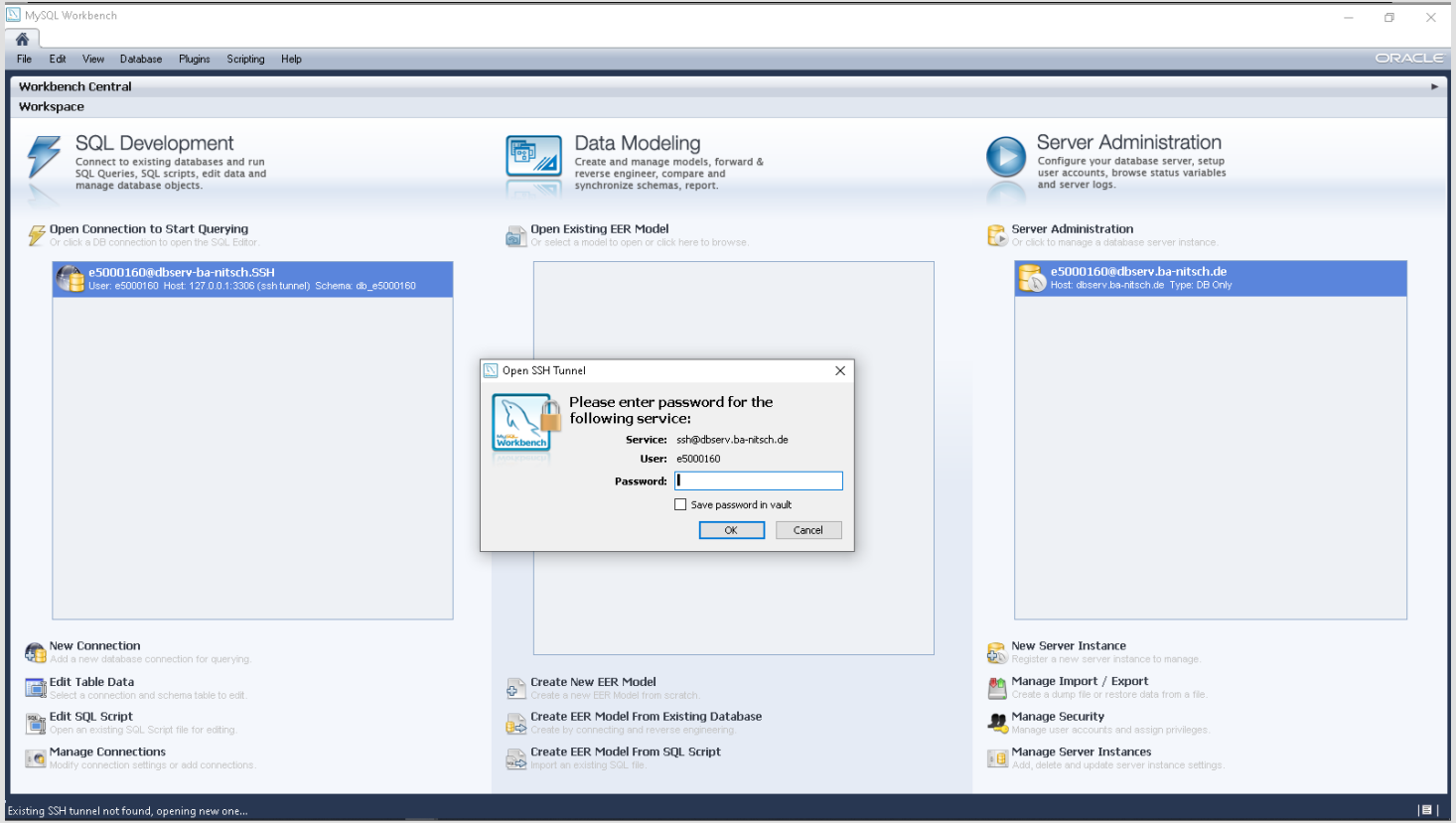


Es gibt nun 2 Logins und 2 Passwörter: Einmal SSH-Tunnel-Username und einmal Username für MariaDB.

In unserem Fall sind Login und Passwort für beide Fälle gleich.

Wieder Tab „Parameters“ wählen und Formular ausfüllen

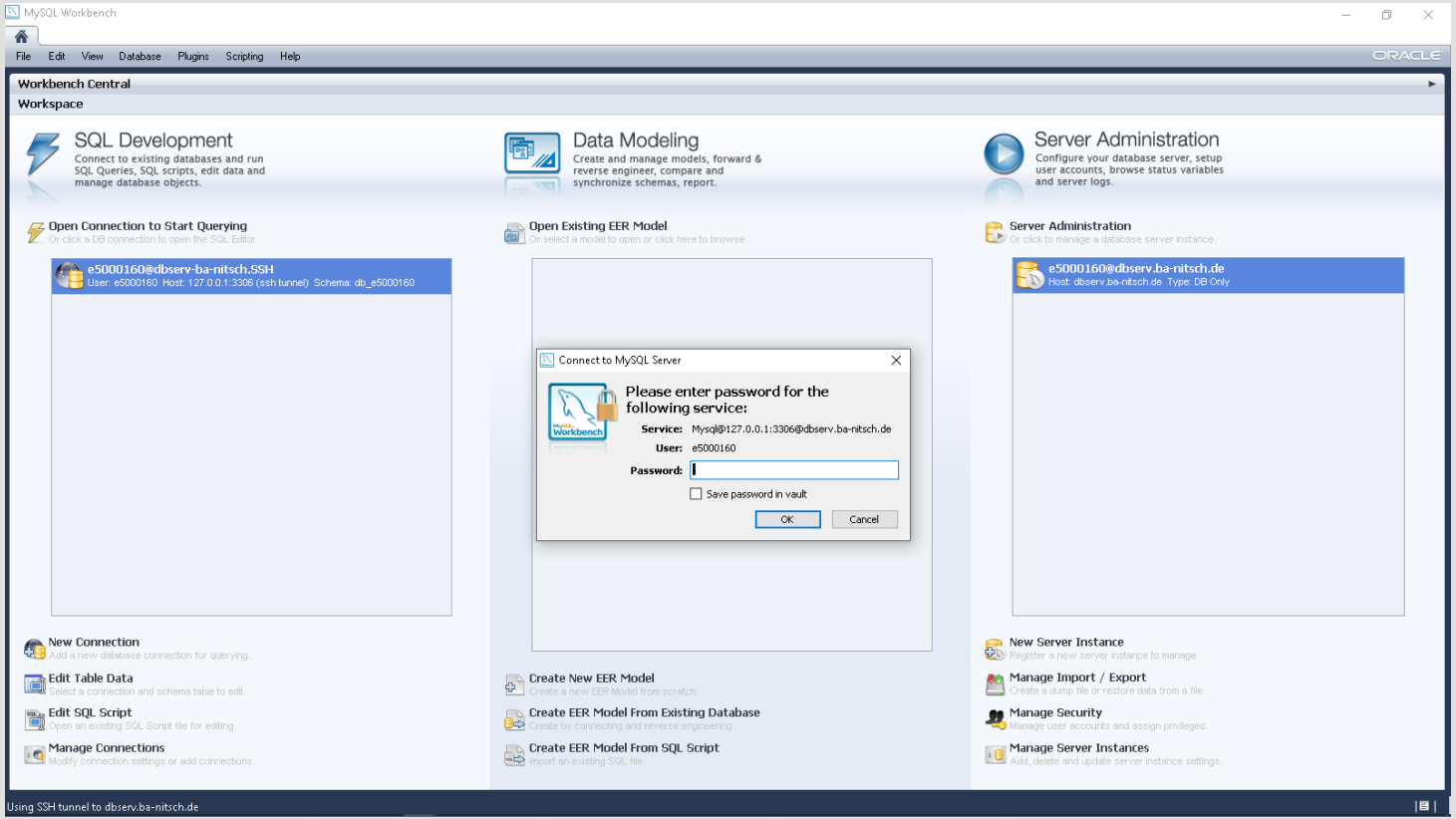
MariaDB:
MySQL Workbench - Client



Hier Abfrage des ersten Pass-wortes für den SSH-Tunnel

Zum Verbinden auf den Verbindungsnamen doppelklicken

MariaDB:
MySQL Workbench - Client



… und hier Abfrage des Passwortes für für die Datenbank selbst