Wiederholung

Wir hatten nur wenige einfache „Problemchen“:

→ Persistente Datenhaltung

→ Performancesicherung

→ Einfache Komplexität

→ Atomarität

→ Schnittstellen

Wiederholung

Aber jedes Mal, je komplexer es wird:
→ Wunsch nach einer Art → „Middleware“ (Datenbank),
die bei den Verwaltungsaufgaben unterstützt.

Allerdings: Nicht immer erlauben die Ressourcen den Einsatz einer solchen (hier eine kleine 64 MHz ARM-CPU)

Heute: SQL - Grundlagen

- Tabellen in SQL
- Attributnamen, Datentypen für Attribute
- Tabellen erstellen
- Tabellen mit Daten befüllen
- Daten löschen
- Tabellen auslesen

- Praxis: Zugriff auf MariaDB - Datenbankserver

Grundlagen

- In relationalen Datenbanken wie MySQL ist „alles“ eine Tabelle
- Grob mit Excel vergleichbar
- Wie haben Tabellen mit n Spalten und m Spalten

Grundlagen:
Tabellen in Excel

	A	B	C	**D**
1	**Nachname**	**Vorname**	**Land**	**Geburtsjahr**
2	Gurion	Ben	Israel	1886
3	Nasser	Gamal, Abdel	Ägypten	1918
4	Curie	Marie	Polen	1867
5	Gagarin	Juri	UdSSR	1934

Grundlagen: Tabellen in Excel

	A	B	C	**D**
1	**Nachname**	**Vorname**	**Land**	**Geburtsjahr**
2	Gurion	Ben	Israel	1886
3	Nasser	Gamal, Abdel	Ägypten	1918
4	Curie	Albert	Deutschland	1867
5	Gagarin	Juri	UdSSR	1934

A2 = „Gurion“

Grundlagen:
Tabellen in SQL

	A	B	C	D
1	**Nachname**	**Vorname**	**Land**	**Geburtsjahr**
2	Gurion	Ben	Israel	1886
3	Nasser	Gamal, Abdel	Ägypten	1918
4	Curie	Marie	Polen	1867
5	Gagarin	Juri	UdSSR	1934

Die automatischen Spalten- / Zeilenbezeichner alá Excel entfallen, dafür bekommen Spalten nun Namen zugewiesen, z.B. „Nachname“ statt „A“.

Grundlagen:
Tabellen in SQL

Nachname	Vorname	Land	Geburtsjahr
Gurion	Ben	Israel	1886
Nasser	Gamal, Abdel	Ägypten	1918
Einstein	Albert	Deutschland	1879

- In SQL sind die **Tabellenköpfe** relevant.
- Deren Einträge heißen hier „Attribute“.
- Wichtige Attributeigenschaften sind **Name** und **Typ**.
- Diese gelten im Gegensatz zu Excel aber für die **gesamte Spalte**.
- Im Beispiel haben wir als Typen nur „Zeichenkette“ und „numerisch“.

Grundlagen:
Tabellen in SQL

Name	Vorname	Land	Geburtsjahr
Gurion	Ben	Israel	1886
Nasser	Gamal, Abdel	Ägypten	1918
Eins			1879
Gaga			1934

```
MariaDB [BA]> explain personen;

+-------------+--------------+------+-----+---------+-------+
| Field       | Type         | Null | Key | Default | Extra |
+-------------+--------------+------+-----+---------+-------+
| Nachname    | varchar(255) | YES  |     | NULL    |       |
| Vorname     | varchar(255) | YES  |     | NULL    |       |
| Land        | varchar(255) | YES  |     | NULL    |       |
| Geburtsjahr | int(4)       | YES  |     | NULL    |       |
+-------------+--------------+------+-----+---------+-------+
```

So kann man sich z.B. im mysql-Client eine Tabelle beschreiben lassen.

Grundlagen:
Tabellen in SQL erstellen

Name	Vorname	Vorname	Geburtsjahr
Gurion	Ben	Israel	1886
Nasser	Gamal, Abdel	Ägypten	1918
Einstein	Albert	Deutschland	1879
Gagarin	Juri	UdSSR	1934

SQL - Befehl zum Erzeugen von Tabellen:

CREATE TABLE *TABELLENNAME* (NAME TYP, NAME TYP, ...);

Grundlagen: Tabellen in SQL

Nachname	Vorname	Land	Geburtsjahr
Gurion	Ben	Israel	1886
Nasser	Gamal, Abdel	Ägypten	1918
Eins			
Gag			

Beispiel:

```
CREATE TABLE personen (
  Nachname varchar(255),
  Vorname varchar(255),
  Land varchar(255),
  Geburtsjahr int(4) );
```

Grundlagen: Tabellen in SQL

Nachname	Vorname	Land	Geburtsjahr
Gurion	Ben	Israel	1886
Nasser	Gamal, Abdel	Ägypten	1918
Einstein	Albert	Deutschland	1879

Die wichtigsten Datentypen

varchar(255) entspricht einem String mit maximaler Länge von 255 Zeichen

int(4) stellt eine 4-stellige(!) Integerzahl dar.

Grundlagen: Tabellen in SQL

Nachname	Vorname	Land	Geburtsjahr
Gurion	Ben	Israel	1886
Nasser	Gamal, Abdel	Ägypten	1918
Curie	Marie	Polen	1867
Gagarin	Juri	UdSSR	1934

Create table erzeugt nur leere Tabellen.
Nun: Tabellen mit Daten füllen

Grundlagen:
Tabellen in SQL mit Daten befüllen

Nachname	Vorname	Land	Geburtsjahr
Gurion	Ben	Israel	1886
Nasser	Gamal, Abdel	Ägypten	1918
Einstein	Albert	Deutschland	1879
Gagarin	Juri	UdSSR	1934

INSERT INTO *TABELLENNAME* (NAME1, NAME2,..) values (VALUE1, VALUE2,..)

Grundlagen:
Tabellen in SQL mit Daten befüllen

Nachname	Vorname	Land	Geburtsjahr
Gurion	Ben	Israel	1886
Nasser	Gamal, Abdel	Ägypten	1918
Einstein	Albert	Deutschland	1879

Bsp:

```
INSERT INTO personen
   (Nachname, Vorname, Land, Geburtsjahr)
values
  ('Gurion', 'Ben', 'Israel', 1886)
```

Grundlagen:
Tabellen in SQL mit Daten befüllen

Nachname	Vorname	Land	Geburtsjahr
Gurion	Ben	Israel	1886

Eine Anmerkung zum NULL - Wert:
NULL ist in der Tat UNDEF, nicht etwa numerisch 0 und auch kein leerer String.

INSERT INTO personen
(Nachname, Vorname, Land, Geburtsjahr)
values
('Gurion', 'Ben', **NULL**, 1886)

Grundlagen: Tabellen in SQL

Nachname	Vorname	Land	Geburtsjahr
Gurion	Ben	Israel	1886
Nasser	Gamal, Abdel	Ägypten	1918
Einstein	Albert	Deutschland	1879

Bemerkungen zur Syntax:

→ Strings bitte immer in Hochkommata

→ Integers auch, auch wenn es da kein Zwang ist

→ Das abschließende **;** ist kein SQL, aber u.U. auf der mySQL-Konsole notwendig

Bsp:

```
INSERT INTO personen (Nachname, Vorname, Land, Geburtsjahr)
values ('Gurion', 'Ben', 'Israel', ‘1886‘);
```

Grundlagen:
Tabelleninhalte in SQL ändern

Nachname	Vorname	Land	Geburtsjahr
Gurion	Ben	Israel	1886
Nasser	Gamal, Abdel	Ägypten	1918
Einstein	Albert	Deutschland	1879
Gagarin	Juri	UdSSR	1934

```
UPDATE TABELLENNAME SET NAME = VALUE WHERE BEDINGUNG
```

Grundlagen:
Tabelleninhalte in SQL ändern

Nachname	Vorname	Land	Geburtsjahr
Gurion	Ben	Israel	1886

Bsp.:

UPDATE personen set Geburtsjahr = 1886 where Nachname='Gurion';

Die where - Bedingung selektiert die Zeilen, die geändert werdensollen. Hier ist es nur die erste Zeile mit Nachnamen ‚Gurion‘.

Würde man ‚where‘ weglassen,würde das Geburtsjahr in allen Zeilen geändert werden.

Grundlagen:
Tabellen in SQL löschen

Nachname	Vorname	Land	Geburtsjahr
Gurion	Ben	Israel	1886
Nasser	Gamal, Abdel	Ägypten	1918
Einstein	Albert	Deutschland	1879

Löschen von Datensätzen (kompletten Zeilen):
DELETE FROM TABELLENNAME where BEDINGUNG

Löschen von Tabellen:
DROP table personen

Grundlagen:
Tabellen in SQL löschen

Nachname	Vorname	Land	Geburtsjahr
Gurion	Ben	Israel	1886
Nasser	Gamal, Abdel	Ägypten	1918
Einstein	Albert	Deutschland	1879

Beispiel:

DELETE FROM PERSONEN where Nachname='Gurion';
Auch hier selektiert wieder ‚where' die Zeile, die gelöscht werden soll.

DROP table PERSONEN;

Grundlagen: Tabellen in SQL

Nachname	Vorname	Land	Geburtsjahr
Gurion	Ben	Israel	1886
Nasser	Gamal, Abdel	Ägypten	1918
Curie	Marie	Marie	1867
Gagari			

Zwischenfazit: Das war schon fast das wichtigste in SQL:

Tabellen anlegen mit CREATE TABLE
Tabellen befüllen mit INSERT INTO
Daten aktualisieren mit UPDATE
Daten löschen mit DELETE / DROP

Grundlagen:
Tabellen auslesen

Nachname	Vorname	Land	Geburtsjahr
Gurion	Ben	Israel	1886
Nasser	Gamal, Abdel	Ägypten	1918
Curie	Marie	Marie	1867
Gagar			

SELECT *SPALTENNAME* FROM *TABELLENNAME* WHERE *BEDINGUNG*

Grundlagen:
Tabellen auslesen

Nachname	Vorname	Land	Geburtsjahr
Gurion	Ben	Israel	1886
Nasser	Gamal, Abdel	Ägypten	1918
Curie	Marie	Polen	1867
Gagarin	Juri	UdSSR	1934

SELECT *SPALTENNAME* FROM *TABELLENNAME* WHERE *BEDINGUNG*

SPALTENNAME wählt Spalten
BEDINGUNG wählt Zeilen

Grundlagen:
Tabellen auslesen

Nachname	Vorname	Land	Geburtsjahr
Gurion	Ben	Israel	1886
Nasser	Gamal, Abdel	Ägypten	1918
Curie	Marie	Polen	1867
Gagarin	Juri	UdSSR	1934

```
SELECT Nachname, Geburtsjahr  FROM PERSONEN WHERE Geburtsjahr<1900;
```

Zeigt die Spalten ‚Nachname' und ‚Geburtsjahr' von allen Zeilen an, in denen das Geburtsjahr kleiner als 1900 ist.

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]>
```

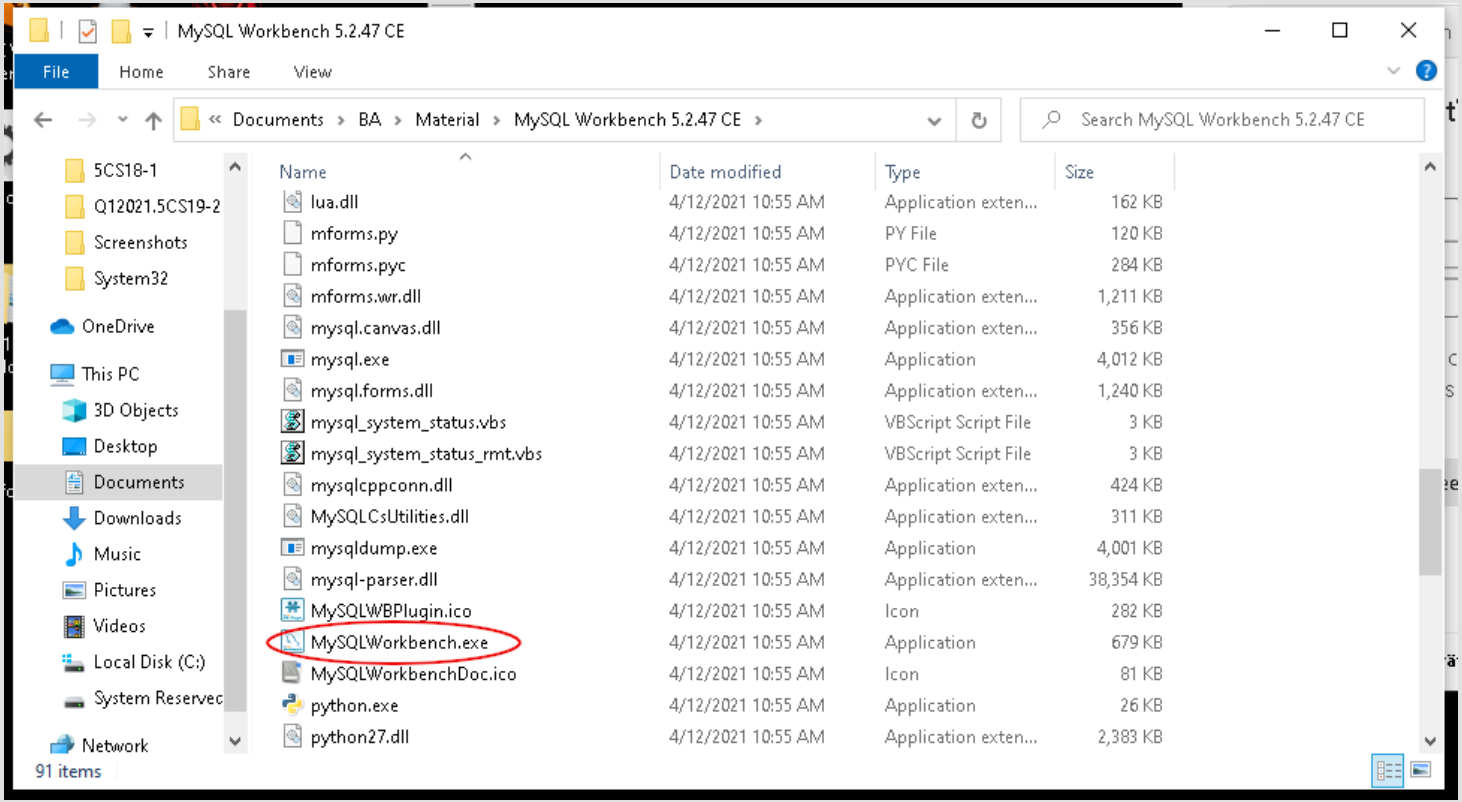
show tables;
explain <tabellenname>;

MariaDB:
MySQL Workbench - Client

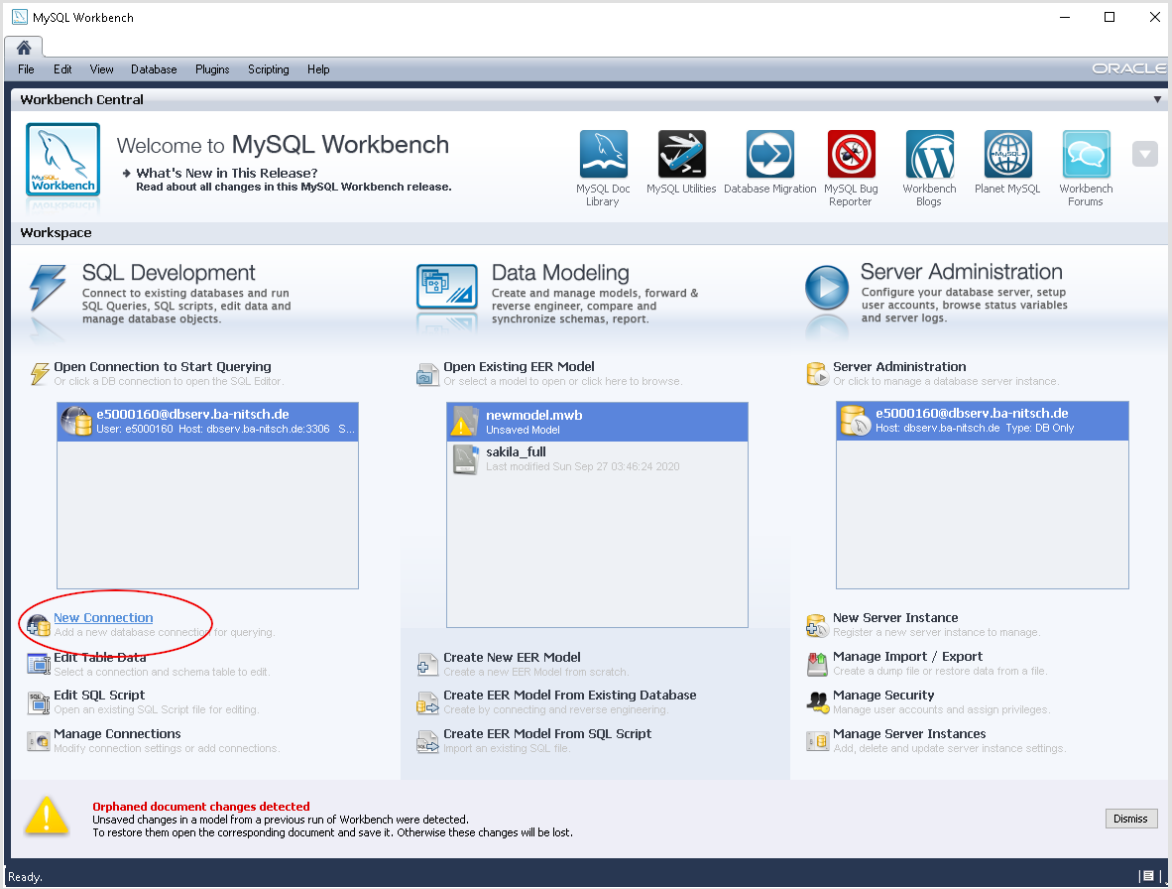
Auf dem Fileserver

\info-klausur\MySQL Workbench 5.2.47 CE

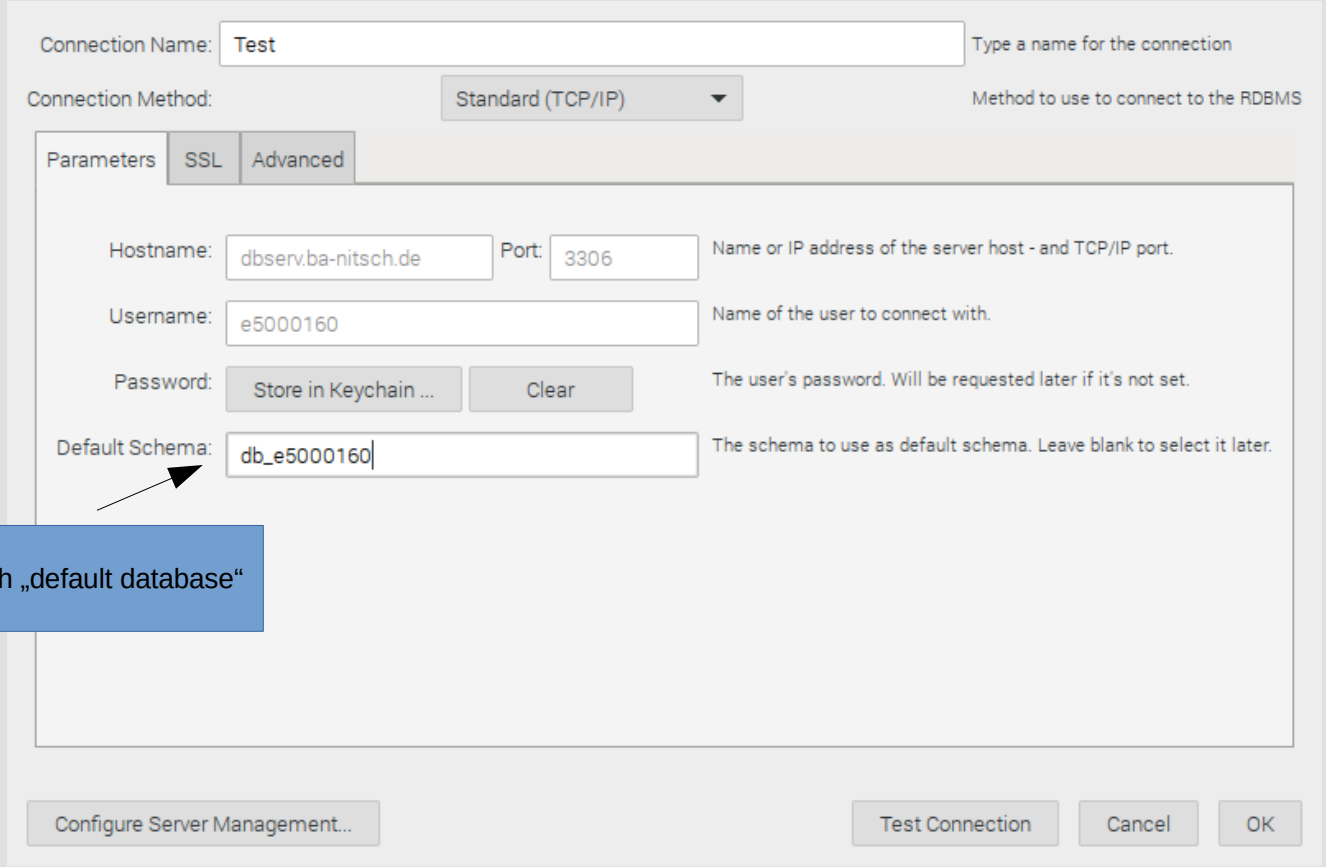
MySQL / MariaDB:
MySQL Workbench: Ordnerstruktur



MySQL Workbench:
Verbindung konfigurieren

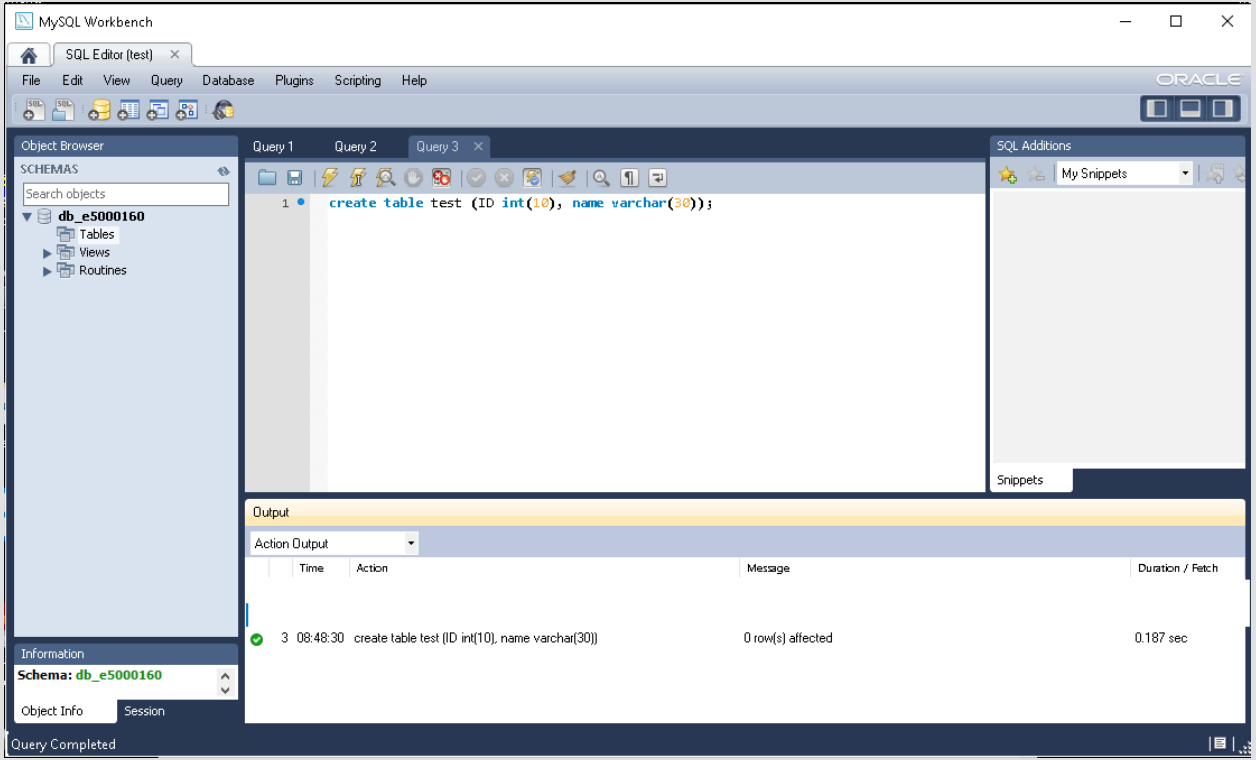


MySQL Workbench: Verbindung konfigurieren



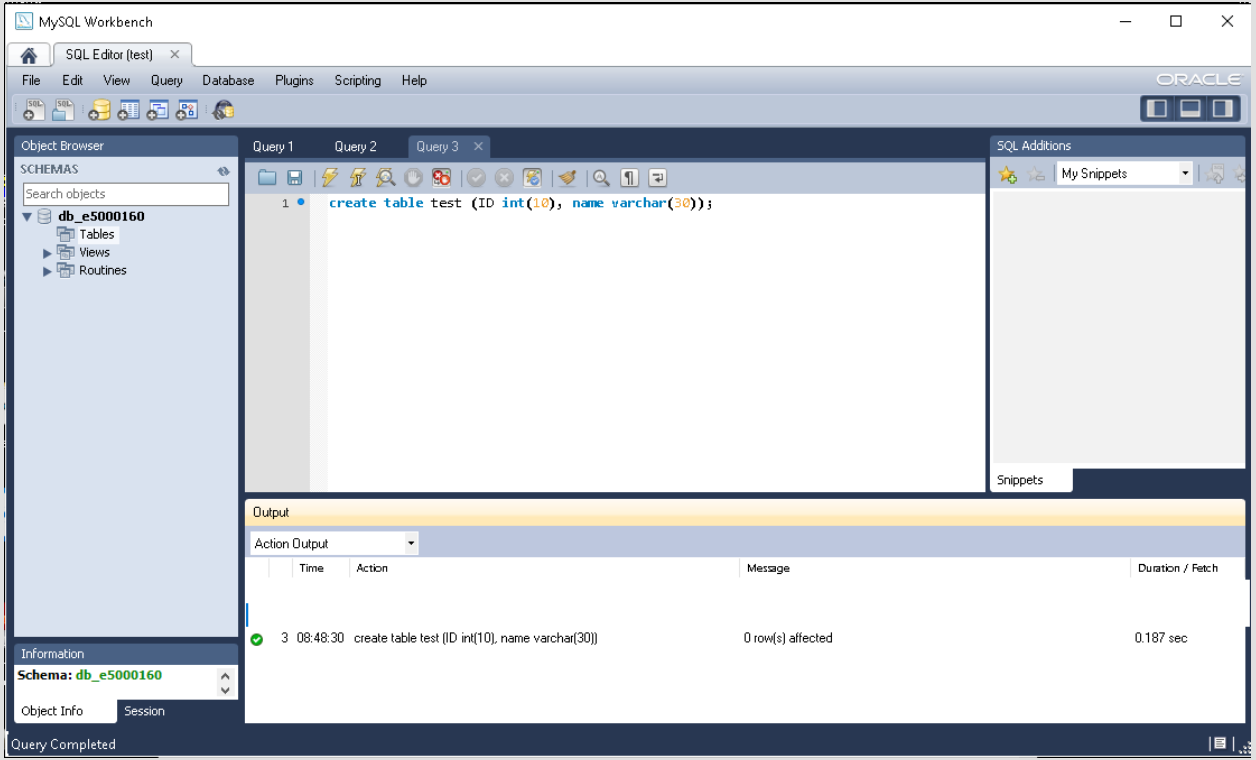
oder auch „default database“

MySQL Workbench:
MySQL Worbbench benutzen



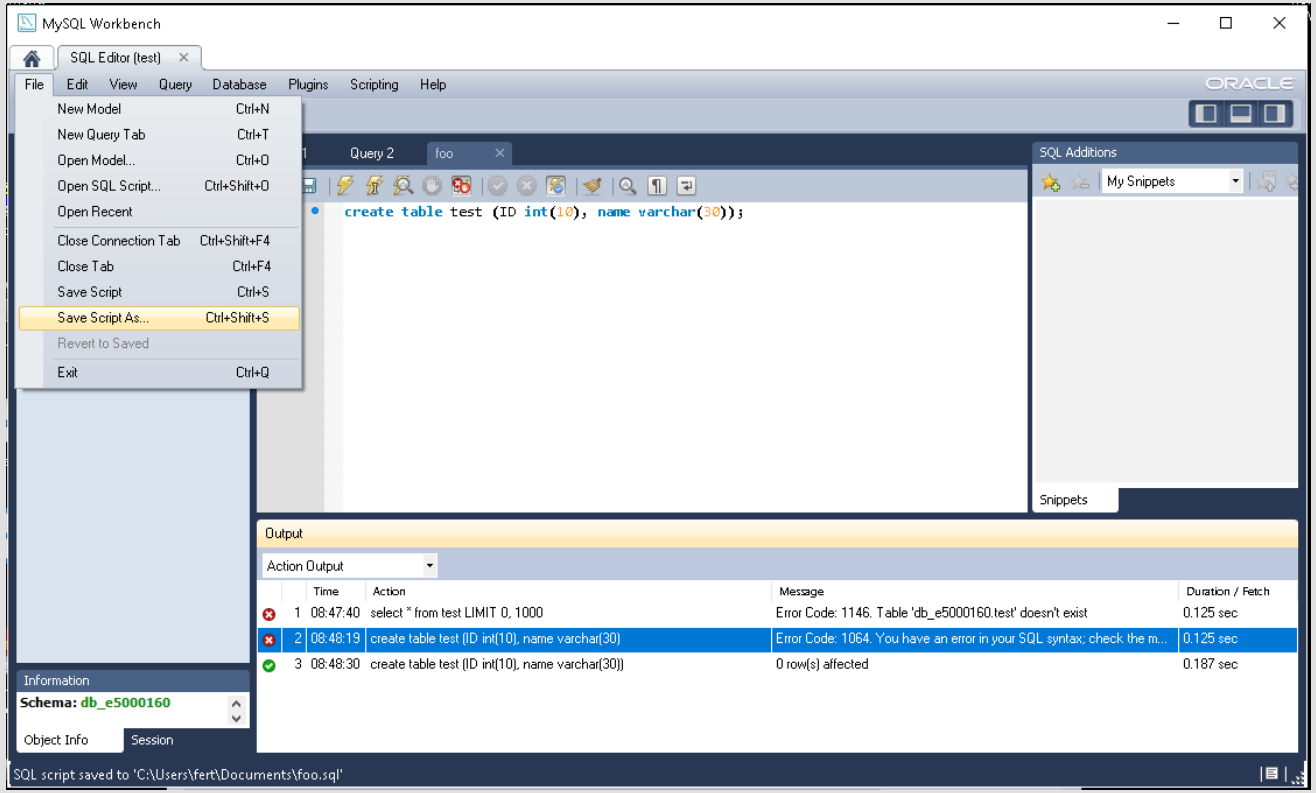
SQL-Konsole

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

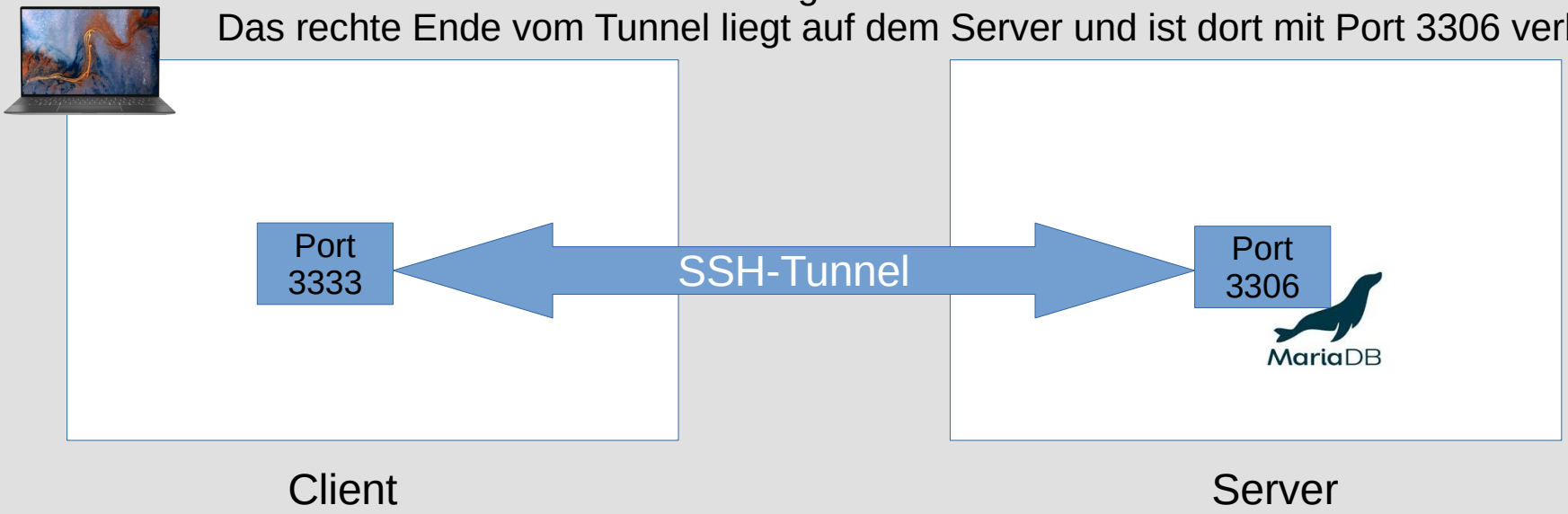
Aber SSH kann mehr: Wenn man sein Kommando noch etwas erweitert, wird sich eingeloggt und extra noch der SSH-Tunnel gebaut.

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

MariaDB:
MySQL Workbench - Client

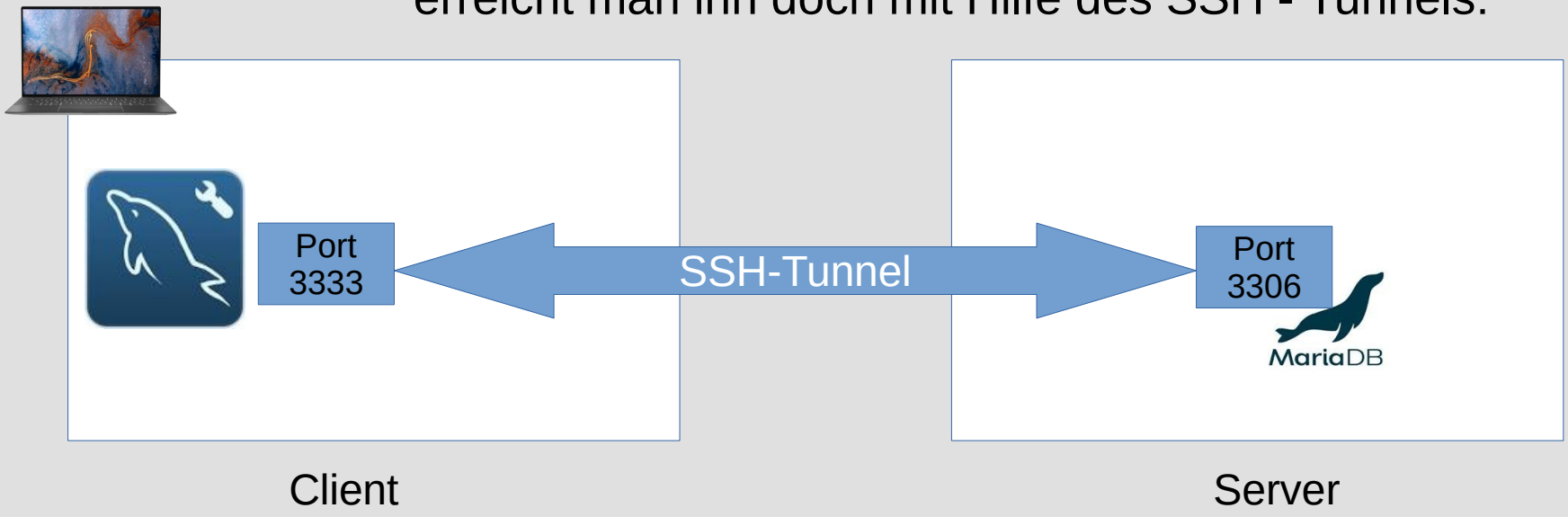
ssh -L 3333:127.0.0.1:3306 e5000160@ba-nitsch.de

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.



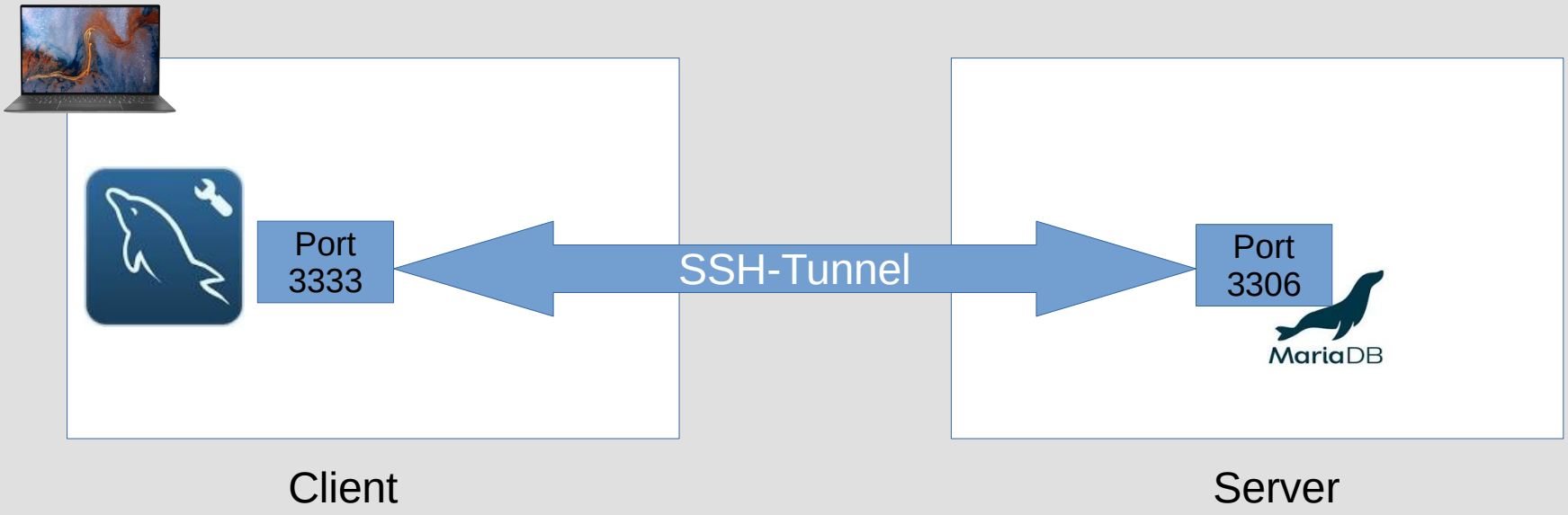
MariaDB:
MySQL Workbench - Client

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:



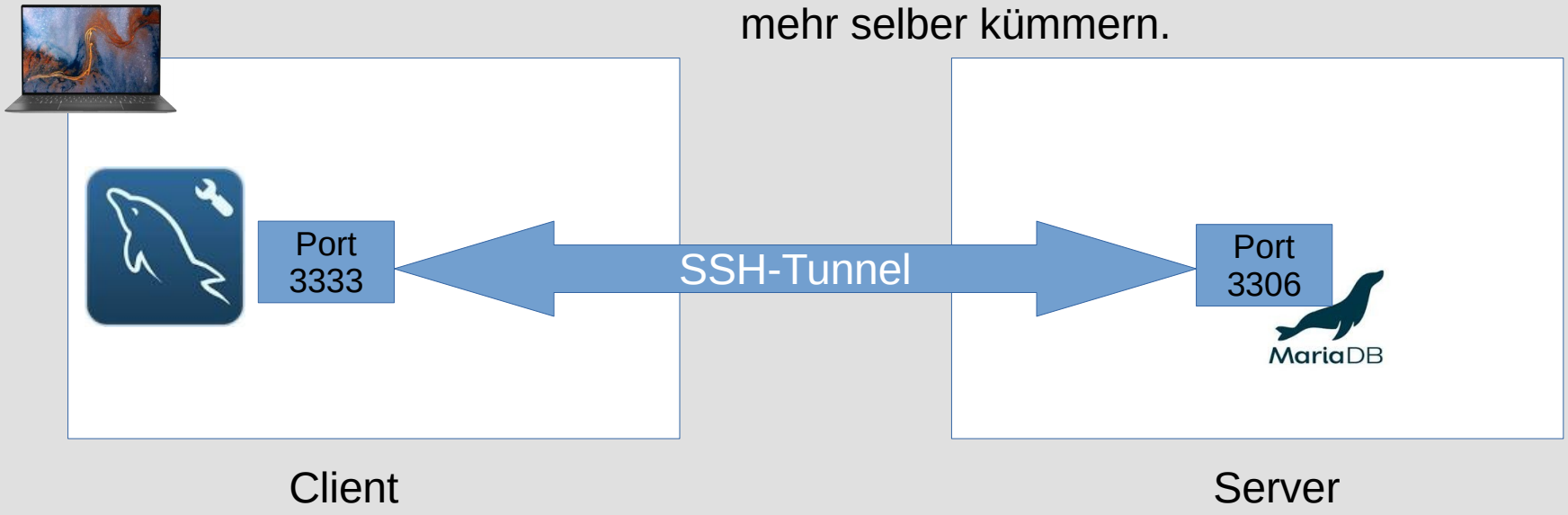
MariaDB:
MySQL Workbench - Client

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.

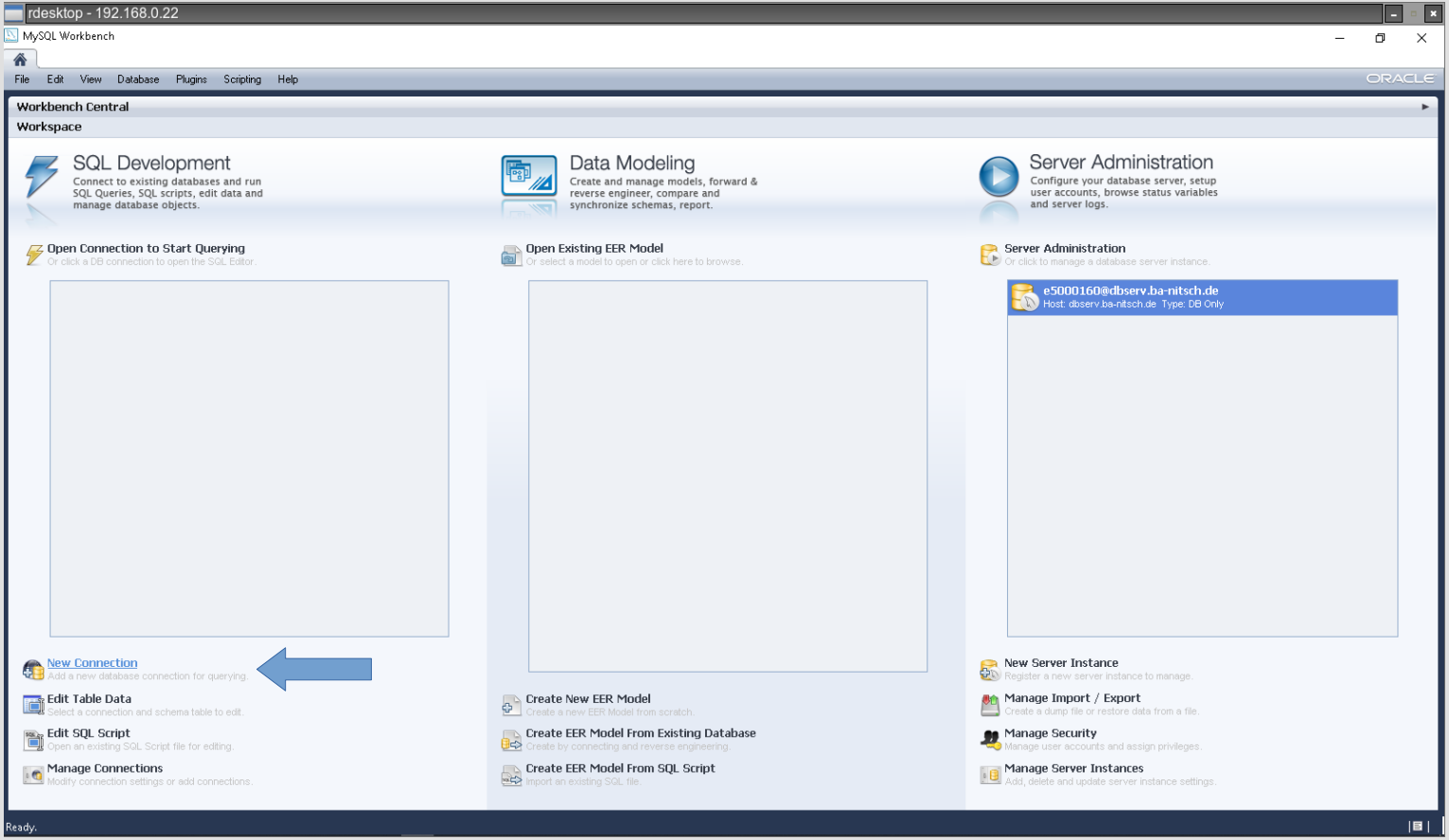


MariaDB:
MySQL Workbench - Client

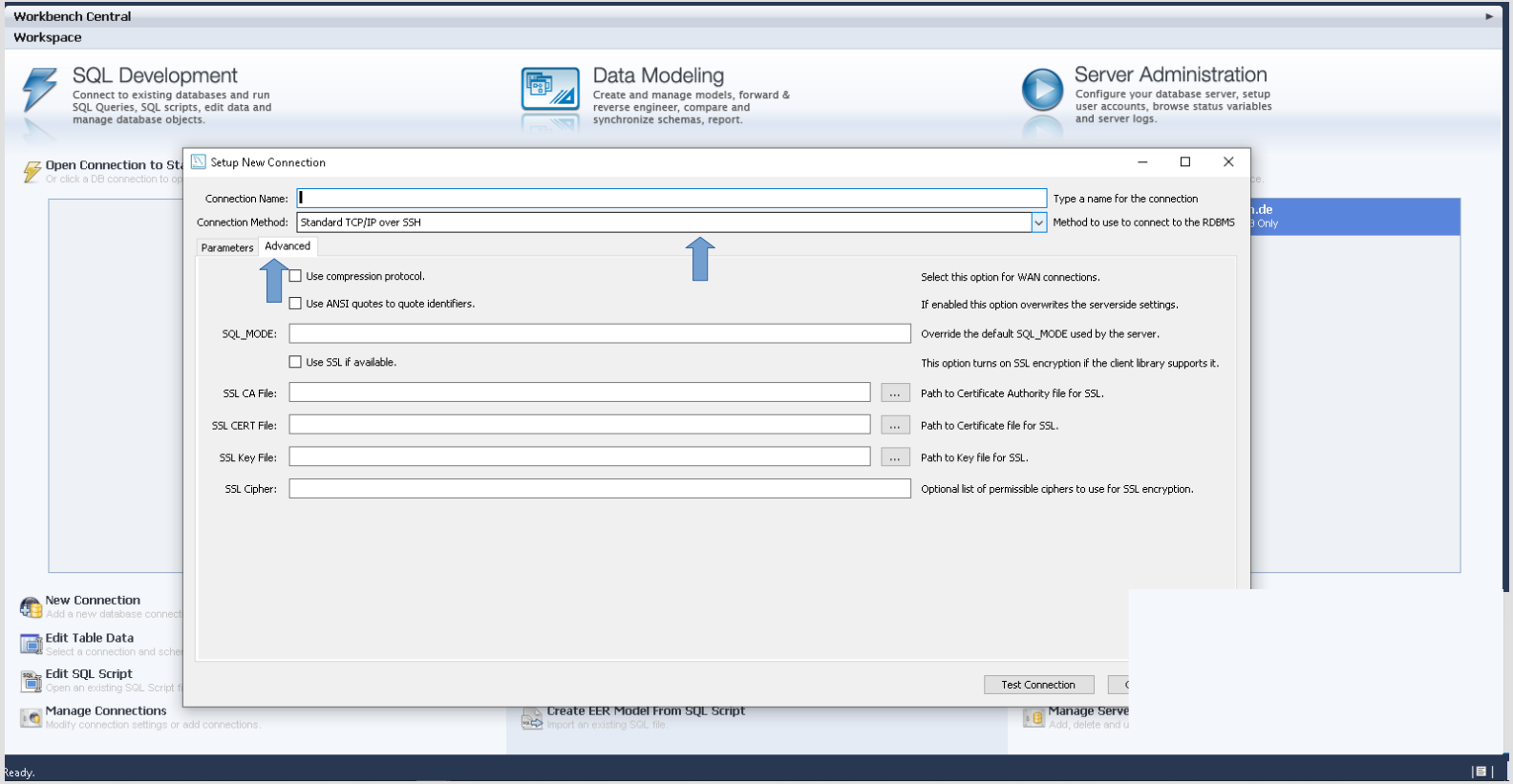
Dies hatten wir beim letzten Mal gemacht.
Zum Glück kann MySQL Workbench solche SSH-Tunnel selber aufbauen,
man muss nur noch Benutzernamen und Passwort konfigurieren.
Der Rest geht dann automatisch, auch um den lokalen Port 3333 muss man sich nun nicht mehr selber kümmern.



MariaDB:
MySQL Workbench - Client

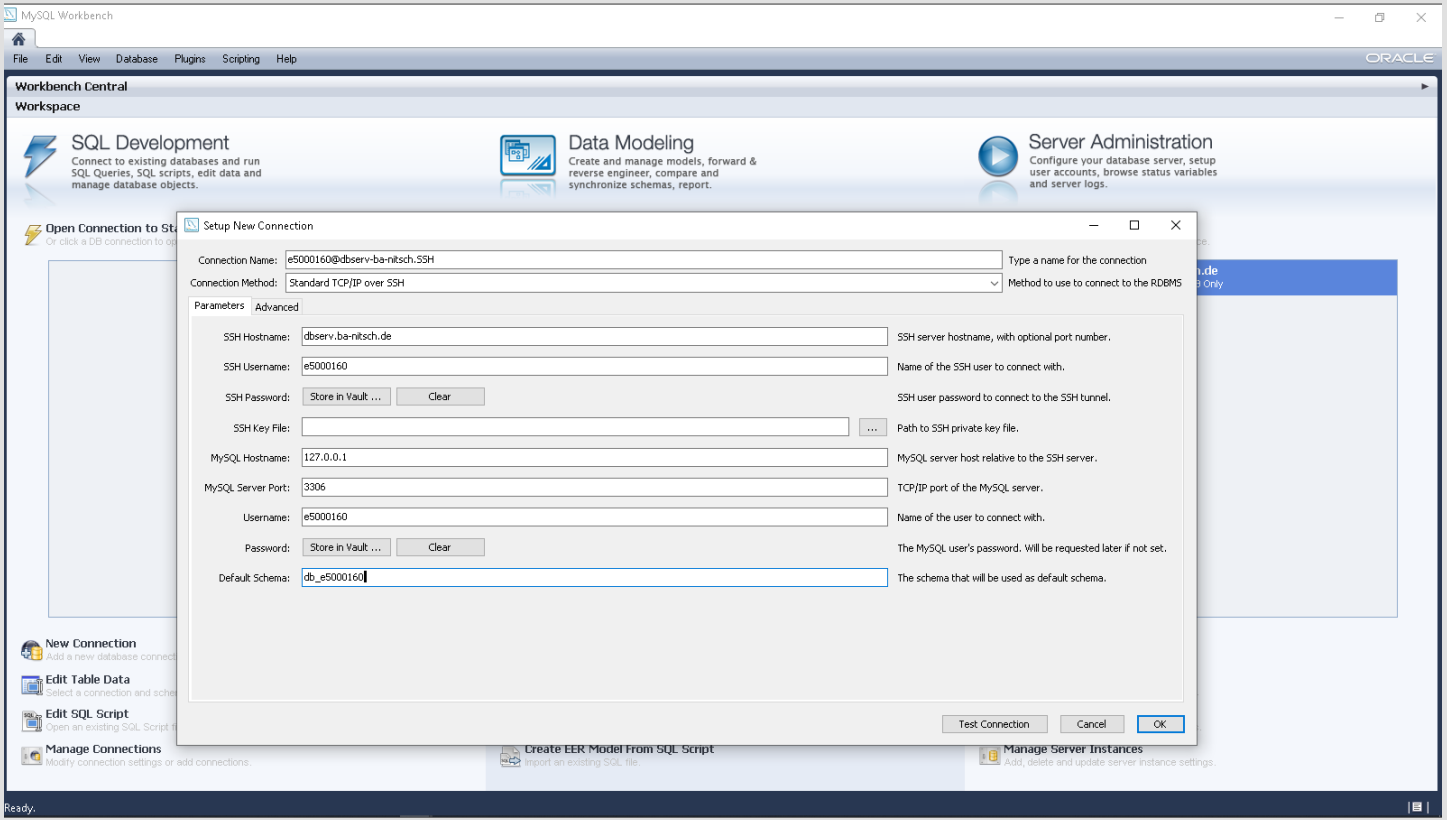


MariaDB:
MySQL Workbench - Client



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

MariaDB:
MySQL Workbench - Client

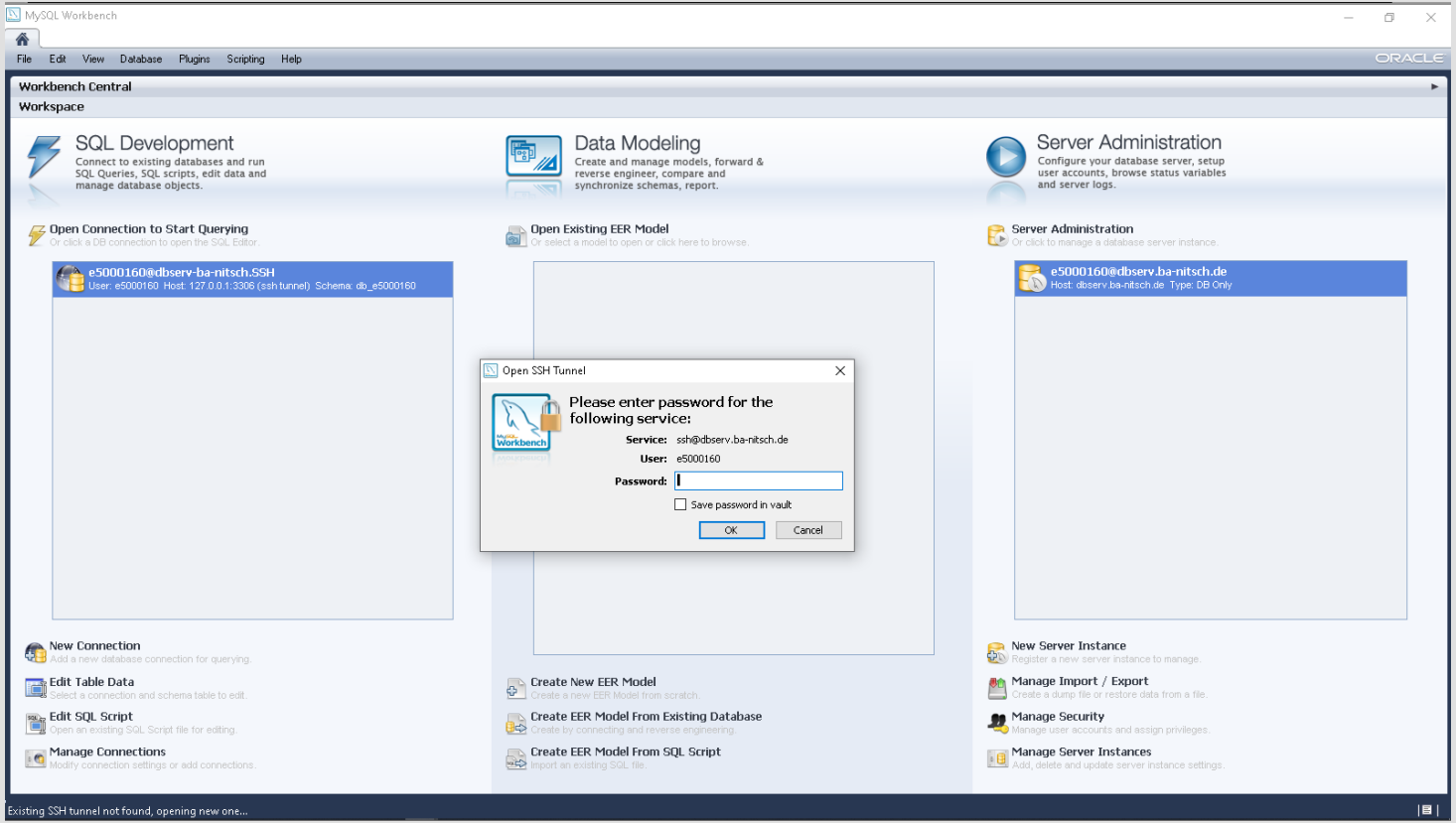


Es gibt nun 2 Logins und 2 Passwörter: Einmal SSH-Tunnel-Username und einmal Username für MariaDB.

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

Wieder Tab „Parameters“ wählen und Formular ausfüllen

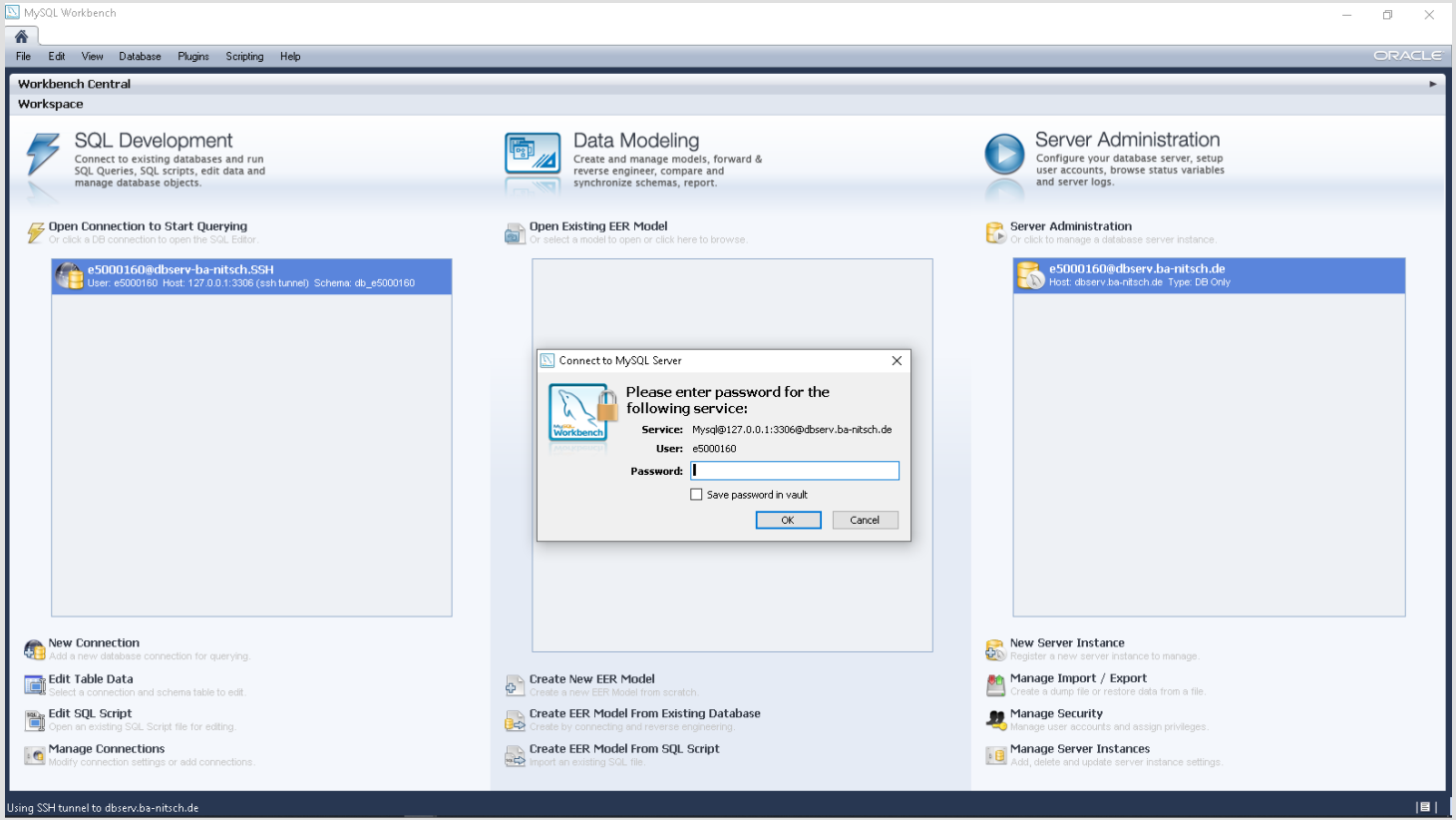
MariaDB:
MySQL Workbench - Client



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