

# MODUL 164

## TEIL 2: DATENBANKPROJEKT UND AUSWAHL RDBMS

Michael Abplanalp

Version 2022

# Modul 164: Datenbanken erstellen und Daten einfügen

## Teil 2: Datenbankprojekt und Auswahl RDBMS

### Inhaltsverzeichnis

Arbeitsblatt Übersicht.....	2
Themen .....	2
Ziele.....	2
Kompetenzen .....	2
Unterlagen / Quellen .....	2
Projektauftrag .....	3
Anforderungen.....	3
Projektablauf .....	3
Projektvorschläge.....	4
Lernumgebung einrichten .....	6
Betriebssystem .....	6
RDBMS.....	6
Installation von MariaDB (nur smartlearn-SSD) .....	6
Installation von MySQL (nur smartlearn-SSD).....	7
Installation von PostgreSQL (nur smartlearn-SSD) .....	8
Installation des Tools SQLyog Community .....	8
Testdatenbank .....	9
Datenbank/Tabellen erstellen und Daten einfügen.....	9
Datenbank bzw. Tool kennenlernen.....	10

# Arbeitsblatt Übersicht

Ab diesem Arbeitsblatt arbeiten Sie neben den Aufträgen parallel an Ihrem Projekt. Deshalb nehmen die Arbeitsblätter nicht mehr 4 Lektionen in Anspruch.

## Themen

- › Anforderungen an das Projekt
- › Projektablauf
- › Projektvorschläge
- › Lernumgebung einrichten
- › Datenbanken testen:
  - › SQL Server: Datenbank mit Skript erstellen und Daten importieren
  - › MariaDB: Datenbank mit Skript erstellen und Daten importieren
  - › evtl. MySQL und PostgreSQL testen

## Ziele

- › Sie kennen die Anforderungen an das Projekt
- › Sie kennen den Projektablauf
- › Sie entscheiden sich für eine Tandempartnerin / einen Tandempartner
- › Sie entscheiden sich für ein Projektthema
- › Sie beginnen mit dem konzeptionellen Datenmodell
- › Sie testen mind. 2 verschiedene Datenbanksysteme RDBMS
- › Sie entscheiden sich für ein RDBMS

## Kompetenzen



1.1 Erstellt aus Anforderungen konzeptionelles Datenmodell

1.2. Erstellt aus konzeptionellem Datenmodell logisches Datenmodell

Das Wissen für diese beiden Kompetenzen haben Sie im Modul 162 erlangt. Letztes Mal haben wir zudem in einer Repetition die beiden Datenmodelle nochmals angeschaut.

## Unterlagen / Quellen

- › Dieses Arbeitsblatt
- › Arbeitsblätter des Moduls 162:
  - › ab162-03Datenbankentwurf1
  - › ab162-04Datenbankentwurf2
  - › ab162-07Normalisierung
- › Buch *SQL Grundlagen und Datenbankdesign* im Ordner 06\_Diverses
- › Youtube-Videos: [Datenbanken und SQL für Anfänger](#)

# Projektauftrag

Sie erstellen eine Datenbank von Grund auf und durchlaufen alle Phasen, bis sich die Daten in der Datenbank befinden.

## Anforderungen

- › Es soll sich um eine **realistische** Datenbank handeln, also eine, die in der Realität vorkommt bzw. vorkommen könnte.
- › Die Datenbank enthält mind. **6 Tabellen**.
- › Die **Tabellen sind korrekt definiert** und enthalten die **notwendigen Attribute**.
- › Die Attribute sind mit dem **korrekten Datentyp** definiert. Folgende Datentypen sind **mind. 1x** vorhanden:
  - › Text/Zeichenkette
  - › Ganzzahl
  - › Kommazahl
  - › Datum
  - › Wahr/Falsch (Datentyp **tinyint**, da es in RDBMS den Typ boolean nicht gibt)
- › **Mind. zwei m:n-Beziehungen** sind vorhanden. Eine davon soll wenn möglich ein **Ereignis** sein.

Beispiel Bibliothek: Jeder Kunde kann viele Bücher ausleihen. Jedes Buch kann von vielen Kunden ausgeliehen werden. Die Ausleihe ist ein Ereignis (Datum von, Datum bis und Preis).
- › Mind. **zwei 1:m-** bzw. **m:1-Beziehungen** sind vorhanden.

Beispiel Bibliothek: Jedes Buch hat einen Verlag, jeder Verlag publiziert viele Bücher.
- › **Die Primärschlüsselwerte werden von der Datenbank verwaltet.**
- › Die referentielle Integrität muss sichergestellt sein (Primärschlüssel-Fremdschlüssel-Beziehung).
- › Weitere Constraints:
  - › Die wichtigen Attribute werden als obligatorisch definiert (NOT NULL).
  - › Der Wertebereich von mind. 1 Attribut wird eingeschränkt (CHECK).
  - › Mind. ein weiterer Constraint (UNIQUE bzw. DEFAULT) ist vorhanden.
- › Jede Tabelle enthält mind. 10 Datensätze.
- › Die Daten sind realistisch und sinnvoll.
- › Wichtige Abfragen liegen vor, d.h. Abfragen, die der Endkunde durchführen würde (SELECT-Befehl).
- › Ihre gesamte Datenbank muss auf einem anderen System (identisches Produkt) reproduziert werden können:
  - › Datenbankschema mit CREATE DATABASE, CREATE TABLE und CONSTRAINT
  - › Daten zur Hälfte mit INSERT-Befehlen und zur Hälfte als CSV-Import

## Projektablauf

1. Sie bilden ein Lerntandem mit einer Kollegin/einem Kollegen
2. Sie entscheiden sich für ein Thema/Projekt, siehe nächstes Kapitel *Projektvorschläge*.
3. Sie erstellen das konzeptionelle Datenmodell entweder direkt oder aufgrund von Ihren eigenen Anforderungen.
4. Aufgrund des konzeptionellen Datenmodells erstellen Sie das logische Datenmodell.

5. Sie führen die Normalisierung durch bzw. überprüfen, ob sich das konzeptionelle Datenmodell in der dritten Normalform befindet.
6. Sie führen das logische Datenmodell in das physische über: Datentypen und Constraints definieren, Zeichensatz festlegen, evtl. Indizes definieren.
7. Sie erstellen die Datenbank und Tabellen.
8. Sie stellen die referentielle Integrität mit Beziehungen (Primär- und Fremdschlüssel) sicher.
9. Sie definieren weitere Constraints und Regeln beim Löschen von Datensätzen.
10. Sie fügen Daten in die Tabellen ein und führen einen Datenimport durch. Es sollten mind. 10 Datensätze pro Tabelle vorhanden sein.
11. Sie überprüfen die Daten und führen Anpassungen durch, falls notwendig.
12. Am Ende erstellen Sie Skripts, um die Datenbank auf einem anderen System zu reproduzieren und die Daten einzufügen.



Es ist ratsam, jeweils nach den einzelnen Schritten Ihre Arbeit der Lehrperson vorführen: Die Lehrperson schaut den Projektstand an, weist Sie auf allfällige Fehler hin und bespricht mit Ihnen Korrekturmöglichkeiten.

Im Falle der Datenmodelle ist es notwendig, diese mit der Lehrperson zu besprechen, bevor Sie weiterfahren. So vermeiden Sie Folgefehler, die zu einem späteren Zeitpunkt nur mit grossem Aufwand korrigiert werden können.

## Projektvorschläge

Am besten arbeiten Sie mit einer eigenen Projektidee. Einem Thema, das Sie interessiert. Falls Ihnen nichts Passendes einfällt, dürfen Sie gerne mit einem der Projektvorschläge arbeiten.

### Web-Börse

Sie erstellen eine Datenbank für eine Web-Börse, ähnlich wie [eBay](#) oder [Ricardo](#). Sowohl die Anbieter als auch die Bieter müssen sich im System registrieren. Die Anbieter erfassen zuerst ihre Artikel, bevor sie Angebote erstellen können.

Die Angebote haben ein Startdatum, danach sind sie für die Bieter sichtbar. Bis zum Enddatum haben die Bieter Gelegenheit, mit einem Betrag für den/die Artikel zu bieten (= Gebot). Derjenige, der am Enddatum das höchste Angebot gemacht hat, erhält den Zuschlag für den/die Artikel.

Minimale Anforderungen für die Datenspeicherung:

- › Personenangaben (Anbieter und Bieter): Name, Adresse, Kontaktangaben, usw.
- › Artikel:
  - › können in Kategorien eingeteilt werden, z.B. "Computer", "TV", "Hi-Fi", usw.
  - › haben einen Zustand, z.B. "neu", "wie neu", "sehr gut", usw.
- › Angebot: Startdatum, Enddatum, Anfangspreis, usw.
- › Jedes Gebot ist ein Ereignis und wird festgehalten mit: Datum, Preis, Bieter, usw.

Mögliche 1:m-Beziehungen: Orte-Personen, Zustand-Artikel, Artikel-Angebote.

Mögliche m:n-Beziehungen: Artikel-Kategorien, Personen-Angebote (wobei die Zwischentabelle *Gebote* ein Ereignis ist mit Datum und Preis).

## Verwaltung von Ferienwohnungen

Sie erstellen für einen Tourismusanbieter ein relationales Datenbankmodell für eine Ferienwohnungsverwaltung. Auf der einen Seite gibt es Eigentümer, die Wohnungen anbieten und vermieten. Auf der anderen Seite stehen die Kunden, die Wohnungen mieten können.

Minimale Anforderungen für die Datenspeicherung:

- › Eigentümerangaben, Mieterangaben: Name, Adresse, Kontaktangaben, usw.
- › Liegenschaften:
  - › haben eine Adresse
  - › es befinden sich eine oder mehrere Wohnungen darin
- › Wohnungen:
  - › haben Eigenschaften, z.B. Vermieter, Preis pro Tag oder Woche, usw.
  - › haben eine Ausstattung, z.B. 2 Schlafzimmer, Waschmaschine, Balkon, usw.
  - › werden in Kategorien eingeteilt: 1-5 Sterne
- › Jede Vermietung ist ein Ereignis und wird festgehalten mit: Datum von, Datum bis, Preis, Mieter, usw.

## Musikverwaltung

Sie erstellen für einen Anbieter von Dienstleistungen im Bereich Musik eine Applikation zur Verwaltung von Songs und Alben. Es werden Interpreten, Songs und Alben erfasst. Die Songs und Alben können auf CD, LP oder digital erscheinen. Zudem gibt es Benutzer, die sich registrieren und Listen mit Alben erstellen können. Benutzer können auch ihre eigenen Alben zum Verkauf anbieten.

Die Funktionen sollen etwa denjenigen von [Discogs](#) entsprechen, mit reduziertem Umfang (siehe Anforderungen).

Minimale Anforderungen für die Datenspeicherung:

- › Interpreten werden mit den notwendigen Informationen erfasst.
- › Es gibt Singles, EPs und Alben. Diese bestehen jeweils aus einem oder mehreren Songs und werden mit Ausgabejahr, Genre und Interpret gespeichert.
- › Jeder Song wird mit Titel und Dauer in Sekunden gespeichert.
- › Interpreten, Songs und Medien werden von einem Redaktionsteam erfasst.
- › Benutzer können sich im System registrieren.
- › Benutzer können Listen mit Singles, EPs und Alben anlegen. Dabei können Sie keine eigenen Medien erfassen, sondern nur aus den bestehenden auswählen.
- › Benutzer können Ihre Singles, EPs und Alben zu einem Fixpreis zum Verkauf anbieten (neu oder gebraucht).
- › Den Kauf eines Artikels kann weggelassen werden.

## Web-Shop Computerartikel

Sie erstellen eine Datenbank für einen Onlineshop im Bereich IT und Multimedia, ähnlich wie [Brack.ch](#) oder [digitec.ch](#). Im Shop werden Artikel in verschiedenen Kategorien angeboten. Der Kunde kann Artikel in den Warenkorb legen und kaufen.

Minimale Anforderungen für die Datenspeicherung:

- › Artikel:
  - › haben eine Beschreibung
  - › haben 1 oder mehrere Bilder hinterlegt (nur Dateipfad)
  - › haben Preise in unterschiedlichen Preisgruppen (Endkunde, Händler, Schüler usw.)

- › sind in Kategorien und Unterkategorien eingeteilt
- › können sich in mehreren Kategorien befinden
- › Kunden:
  - › müssen sich vor dem Kauf registrieren (Name, Adresse, Kontaktangaben)
  - › können alle getätigten Käufe einsehen
- › Jede Bestellung ist ein Ereignis und wird festgehalten mit: Kunde, Datum, Preis, usw.
  - › Bestellungen können aus mehreren Artikeln bestehen
  - › Bestellungen haben einen Status (bestellt, in Bearbeitung, geliefert, usw.)

# Lernumgebung einrichten

## Betriebssystem

Sie haben die Wahl, ob Sie mit *Windows 10* oder *Ubuntu 20.04* arbeiten. Falls Sie sich für SQL Server entscheiden, wird Windows 10 empfohlen, auch wenn es eine Version für Linux gibt. Bei den anderen RDBMS spielt das Betriebssystem keine Rolle.

Alle Datenbanken und Skripts wurden mit Windows 10 getestet, die Unterlagen basieren alle auf diesen Installationen.

## RDBMS

Folgende **Relationale Datenbankmanagementsysteme** können im Modul 164 eingesetzt werden:

- › **SQL Server** ist zusammen mit dem **SQL Server Management Studio** auf smartlearn.Online und auf der smartlearn-SSD installiert. Alle Skripte sind in der Version SQL Server vorhanden.
- › **MariaDB** ist zusammen mit **HeidiSQL** (grafisches GUI) auf smartlearn.Online installiert. Das RDBMS muss auf der smartlearn-SSD installiert werden. Alle Skripte sind in der Version MariaDB vorhanden.
- › Äquivalent zu MariaDB ist **MySQL**, d.h. beide Systeme sind weitgehend identisch. Falls Sie MySQL bevorzugen: Auf smartlearn.Online ist das System bereits installiert, auf der smartlearn-SSD muss es installiert werden. Die MariaDB-Skripte sollten alle auch mit MySQL laufen.
- › Falls Sie mit **PostgreSQL** arbeiten wollen: Auf smartlearn.Online ist das System bereits installiert, auf der smartlearn-SSD muss es installiert werden. Damit die Skripte laufen, müssen diese zuerst angepasst werden.

Bevor Sie sich für ein RDBMS entscheiden, haben Sie die Gelegenheit, die verschiedenen Systeme zu testen.

## Installation von MariaDB (nur smartlearn-SSD)

Sie finden die Installationsdatei auf dem Modulshare:

`iet-164/07_Software/mariadb-10.6.5-win64.msi`

1. Starten Sie die Installation.
2. Verändern Sie die Standardwerte auf den Masken nicht, bis Sie auf die Maske *Default Instance Properties* gelangen.
3. Geben Sie als *New root password* ein: `sml12345`
4. Markieren Sie *Use UTF8 as default server's character set*.

5. Lassen Sie auf der nächsten Maske die Standardwerte sein:
  - › Install as service markieren
  - › Service Name: MariaDB
  - › Enable networking markieren
  - › TCP port: 3306
6. Schliessen Sie die Installation ab.
7. Starten Sie HeidiSQL.
8. Klicken Sie auf den Button *Neu* links unten.
9. Lassen Sie die Standardeinstellungen sein und ergänzen das Passwort:
  - › IP: 127.0.0.1
  - › Benutzername: root
  - › Passwort: sml12345
  - › Port: 3306
  - › Datenbanken: Lassen wir vorläufig leer. Später kann hier die Datenbank eingetragen werden, mit der Sie sich verbinden wollen.
10. Nach dem Klicken auf den Button *Öffnen* können Sie mit dem RDBMS arbeiten.

## Installation von MySQL (nur smartlearn-SSD)

Sie finden die Installationsdatei auf dem Modulshare:

`iet-164/07_Software/mysql-installer-web-community-8.0.27.0.msi`

1. Starten Sie die Installation.
2. Wählen Sie den Installationstyp *Developer Default*.
3. Falls die Fehlermeldung erscheint, dass Python 3.9 nicht installiert ist, dann führen Sie die Installation durch mit der Datei  
`iet-164/07_Software/python-3.9.9-amd64.exe`
4. Falls die Fehlermeldung erscheint, dass Visual Studio 2019 nicht installiert ist, dann führen Sie die Installation durch mit der Datei  
`iet-164/07_Software/vs_community_2019.exe`
5. Im Falle von 3 und/oder 4: Starten Sie die Installation erneut und wählen Sie den Installationstyp *Developer Default*.
6. Wählen Sie auf der folgenden Maske die folgenden Produkte aus:
  - › MySQL Server
  - › MySQL Workbench
  - › MySQL Shell
  - › MySQL Documentation
  - › Alle anderen Optionen können Sie abwählen
7. Wählen Sie auf der Maske *Type and Networking* aus:
  - › Config Type: Development Computer
  - › TCP/IP: ausgewählt
  - › **Port: 3307 (3306 ist durch MariaDB belegt!)**
  - › X Protocol Port: 33070
  - › Open Windows Firewall ports for network access: ausgewählt
8. Wählen Sie auf der Maske *Authentication Method* aus:  
Use Strong Password Encryption for Authentication
9. Geben Sie auf der nächsten Maske das Passwort ein: sml12345
10. Lassen Sie auf der Maske *Windows Service* alle Standardwerte sein.
11. Schliessen Sie die Installation ab.



12. Starten Sie MySQL Workbench und wählen *Local instance MySQL80*.
13. Geben Sie das Passwort *sml12345* ein, danach können Sie mit dem RDBMS arbeiten.

## Installation von PostgreSQL (nur smartlearn-SSD)

Sie finden die Installationsdatei auf dem Modulshare:

`iet-164/07_Software/postgresql-14.1-1-windows-x64.exe`

1. Starten Sie die Installation.
2. Klicken Sie sich durch alle Standardwerte durch, bis Sie zum Formular *Password* gelangen.
3. Geben Sie als Passwort ein: *sml12345*
4. Auf der nächsten Maske lassen Sie den Port auf: 5432
5. Auf der nächsten Maske können Sie *Locale* auf [Default locale] sein lassen.
6. Schliessen Sie die Installation ab.
7. Die nachfolgende (automatisch startende) Installation von Stack Builder können Sie abbrechen.
8. Starten Sie *pgAdmin 4* und geben das Passwort *sml12345* ein, danach können Sie mit dem RDBMS arbeiten.

## Installation des Tools SQLyog Community

Mit diesem Tool kriegen Sie ein tolles GUI für MySQL und MariaDB. Dieses ist eine Verbesserung gegenüber *HeidiSQL* und *MySQL Workbench*!

Sie finden die Installationsdatei auf dem Modulshare:

`iet-164/07_Software/SQLyog-13.1.8-0.x64Community.exe`

1. Starten Sie die Installation.
2. Klicken Sie sich durch alle Standardoptionen und schliessen die Installation ab.
3. Starten Sie das Tool und klicken im Tab *MySQL* auf den Button *New...*
4. Als Namen geben Sie *MariaDB* (oder einen Namen nach Wunsch) ein.
5. MariaDB: fahren Sie mit den Punkten 6-8 weiter. MySQL: Fahren Sie mit den Punkten 9-11 weiter.
6. Geben Sie für **MariaDB** ein (bzw. lassen die Werte sein):
  - › MySQL Host Address: localhost
  - › Username: root
  - › Password: sml12345
  - › Port: 3306
  - › Database(s): Lassen wir vorläufig leer. Später kann hier die Datenbank eingetragen werden, mit der Sie sich verbinden wollen.
7. Speichern Sie die Verbindung.
8. Sie können mit MariaDB loslegen.
9. Geben Sie für **MySQL** ein (bzw. lassen die Werte sein):
  - › MySQL Host Address: localhost
  - › Username: root
  - › Password: sml12345
  - › Port: 3307
  - › Database(s): Lassen wir vorläufig leer. Später kann hier die Datenbank eingetragen werden, mit der Sie sich verbinden wollen.
10. Speichern Sie die Verbindung.

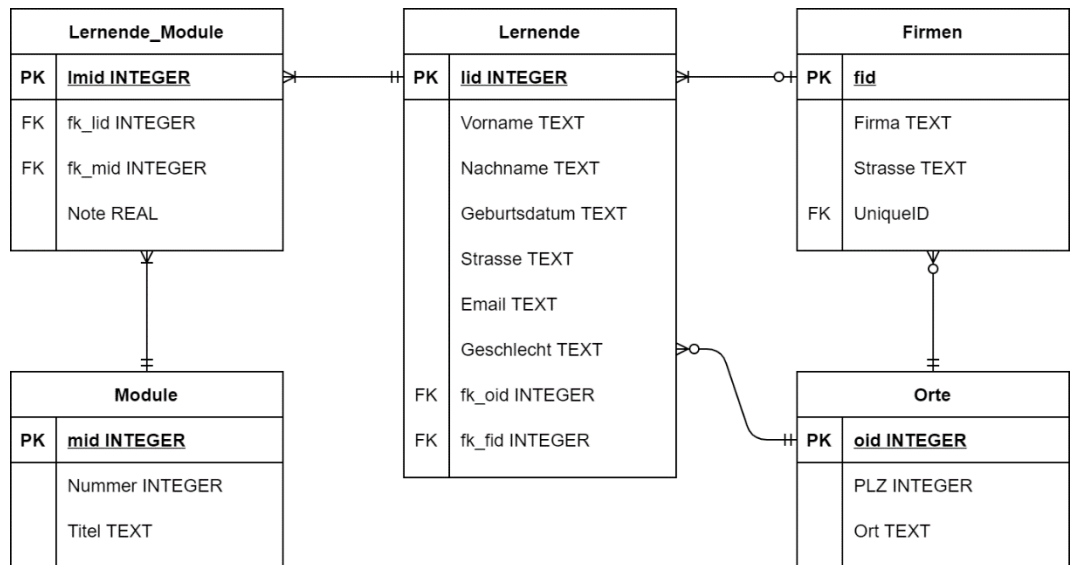
11. Sie können mit MySQL loslegen.

# Testdatenbank

## Datenbank/Tabellen erstellen und Daten einfügen

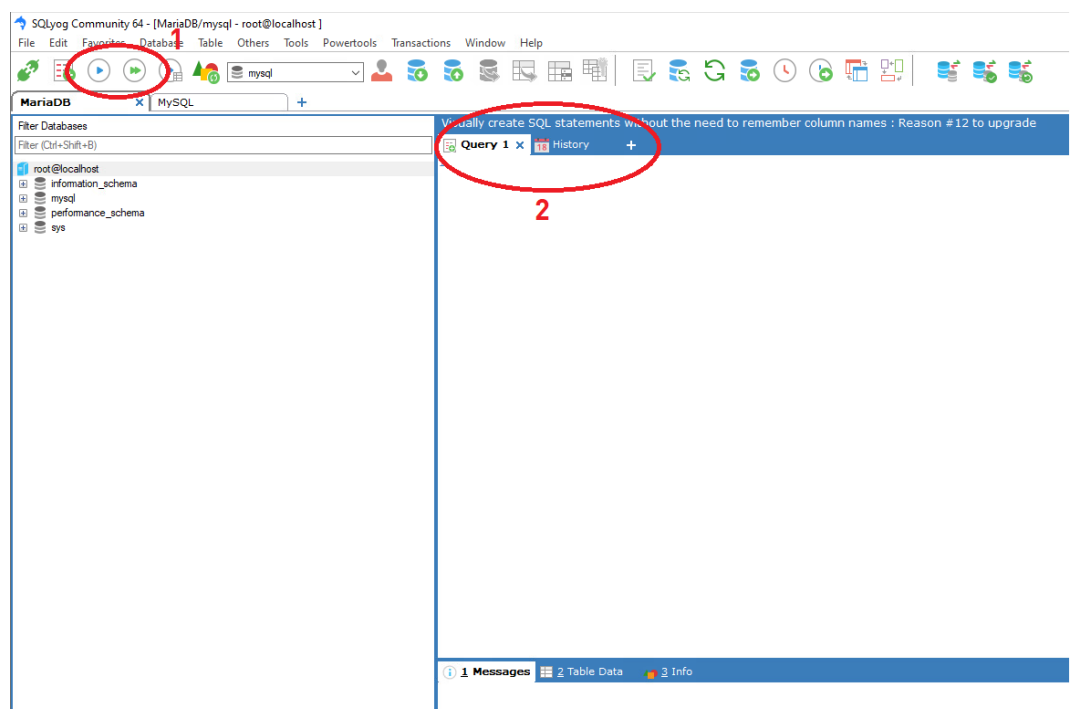
Wir wollen eine Testdatenbank erstellen, damit Sie die verschiedenen RDBMS testen können. Es existieren Skripte für SQL Server und MariaDB/MySQL.

Bei der Datenbank handelt es sich um die Datenbank *Lernende* aus dem Modul 162. Das logische Datenmodell:



Der Ablauf wird beispielhaft mit *MariaDB* und *SQLyog* gezeigt:

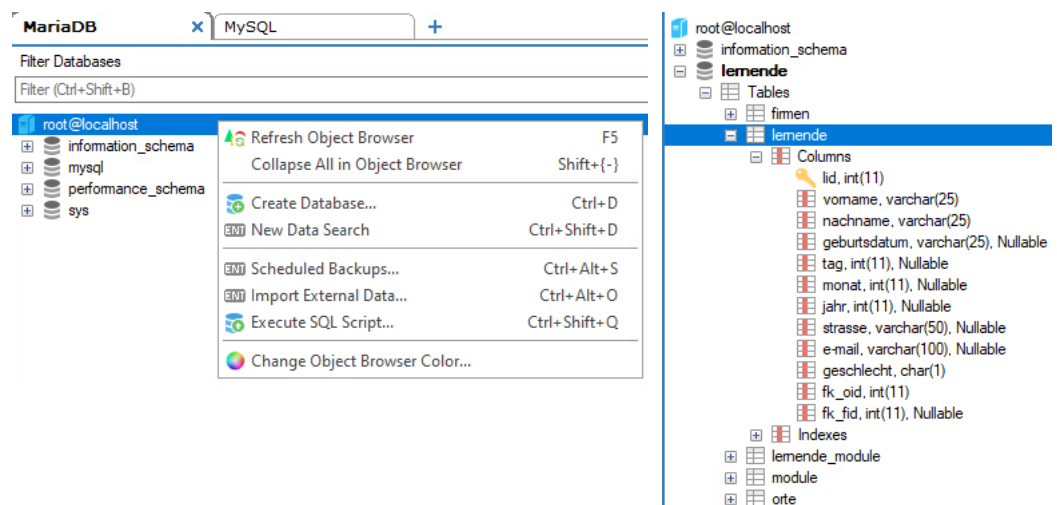
### 1. Starten Sie SQLyog



Unter 1 kann ein einzelner SQL-Befehl bzw. es können alle SQL-Befehle, die sich im Fenster rechts befinden, ausgeführt werden.

Im Fenster **2** können SQL-Befehle eingegeben werden. Mit dem + Zeichen kann ein neuer Tab geöffnet werden.

- Erstellen Sie die Datenbank *lernende*: `CREATE DATABASE lernende`
- smartlearn-SSD: Erstellen Sie ein neues Datenverzeichnis für das Modul 164, z.B. `C:\Daten\m164`
- Kopieren Sie in dieses Verzeichnis: Die 4 sql-Dateien, die sich auf dem Modulshare unter `iet-164\03_Arbeitsblaetter\ab164-02_Hilfsdateien` befinden.
- Führen Sie das Skript `MariaDB_lernende_create_table.sql` aus, entweder indem Sie *Execute SQL Script* auswählen oder indem Sie die Datei im Editor öffnen und den Inhalt mit copy&paste in das Query-Fenster einfügen.
- Klicken Sie mit der rechten Maustaste auf `root@localhost` und wählen *Refresh Object Browser*.



Danach sehen Sie die neue Datenbank *lernende* und können die Tabellen und Tabellenstruktur sehen.

7. Führen Sie das Skript `MariaDB_lernende_import_data.sql` aus, indem Sie *Execute SQL Script* auswählen.
- Achtung:** Es funktioniert nur diese Methode, weil die Datei zu lange ist, um mit copy&paste in das Query-Fenster einzufügen!

## Datenbank bzw. Tool kennenlernen

Testen Sie verschiedene Funktionen mit der Datenbank und dem Tool.

- › Führen Sie verschiedene SQL-Befehle aus, soweit Sie diese kennen (z.B. SELECT-Befehl).
- › Untersuchen Sie mit dem GUI die Tabellenstruktur, z.B. ist folgende Ansicht möglich (Tabelle erweitern → Columns → rechte Maustaste → Manage Columns):

[illegible]

- › Versuchen Sie, über das GUI die Tabellenstruktur zu verändern.
- › Versuchen Sie, über das GUI Daten einzufügen, zu ändern und zu löschen.
- › Versuchen Sie, über das GUI Daten anzuzeigen.
- › Erkunden Sie das Tool und die Datenbank so gut als möglich.
- › Ziel ist es, dass Sie verschiedene DBMS und verschiedene Tools kennenlernen und schlussendlich zu einer Entscheidung kommen, welches DBMS Sie im Verlauf des Moduls verwenden werden.