**Aufgaben-Sammlung**

MySql / Databases

Last Changes: Walter Rothlin / June, 9th 2017

F:\\_Daten\Hochschule Zurich\Meine\_SlideSammlung\_Golden\_Source\Aufgabensammlung\_DB\_yyyy\_mm\_dd.docx

Index

[Preconditions 3](#_Toc478920922)

[Installed components 3](#_Toc478920923)

[Helpful Links 3](#_Toc478920924)

[Uebungen zu sakila DB 4](#_Toc478920925)

[ERD sakila (Example DB) 4](#_Toc478920926)

[Fragen zum ERD 5](#_Toc478920927)

[Abfragen / Selects 5](#_Toc478920928)

[Abfragen mit Function-calls 6](#_Toc478920929)

[Joins 7](#_Toc478920930)

[Meta-Daten abfragen 10](#_Toc478920931)

[Views / Functions / Stored procedures 11](#_Toc478920932)

[Views 11](#_Toc478920933)

[Functions 11](#_Toc478920934)

[Stored procedures 11](#_Toc478920935)

[Using Command Line Client 12](#_Toc478920936)

[Using Eclipse TOAD extension 13](#_Toc478920937)

[Add MySQL JDBC driver (Original in Anleitung\_Application\_Server\_Setup\_xx.docx) 13](#_Toc478920938)

[Add MySQL connection 15](#_Toc478920939)

[Create DB-Connection from TOAD Plugin 17](#_Toc478920940)

# Preconditions

## Installed components

* MySql with Workbench 6.3
* JDBC Driver mysql-connector-java-5.1.xxx-bin
* MySQL Eclipse Plugin – Toad extension

## Helpful Links

<https://zindilis.com/docs/mysql/sakila-example-queries.html>

<https://dev.mysql.com/doc/sakila/en/sakila-usage.html>

<https://dev.mysql.com/doc/refman/5.7/en/date-and-time-functions.html>

<https://dev.mysql.com/doc/refman/5.7/en/string-functions.html>

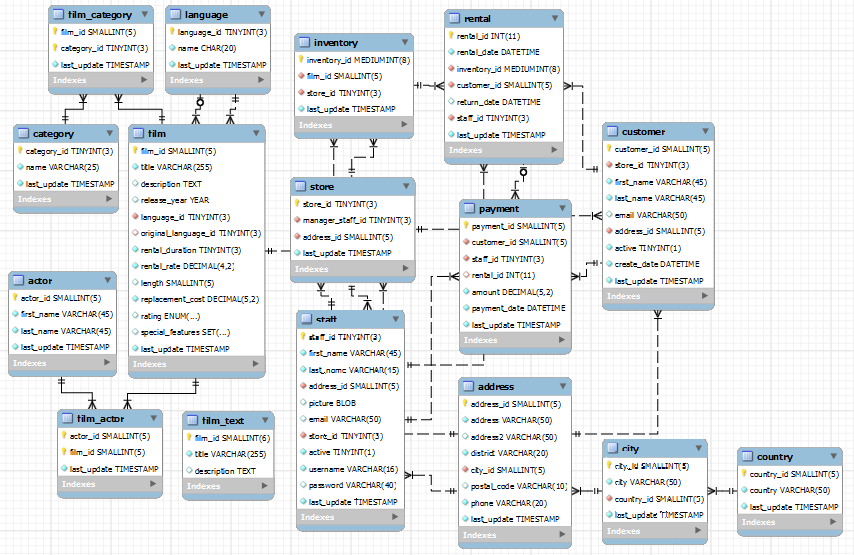
<https://dev.mysql.com/doc/refman/5.7/en/tables-table.html>

<https://www.w3schools.com/sql/sql_join.asp>

<https://dev.mysql.com/doc/sakila/en/sakila-usage.html>

# Uebungen zu sakila DB

## ERD sakila (Example DB)



## Fragen zum ERD

* Listen Sie alle Attribut-Typen auf die Sie finden. Gibt es ihnen unbekannte?
* Machen Sie eine Liste der Symbole, Farben vor den Attribute Namen und deren Bedeutung.
* Suchen Sie folgende Beziehungen:
  + 1 : 1..n
  + 1 : 0..n
  + m : n
* Was bedeuten die zwei Relationen (Beziehungen) zwischen **language** und **film**?
* Wieso steht die Tabelle film\_text in keiner Beziehung zu einer anderen Tabelle?
* Mit der Work-Bench können Sie aus einer DB ein ERD durch Reverse Engineering erzeugen. Machen Sie das mit **Database 🡪 Reverse Engineering…**

## Abfragen / Selects

Löse folgende Aufgaben mit der **MySql workbench**, dem **Toad Eclipse Plugin** und auf der **command line**:

- Von welchen Schauspielern (Vorname und Nachname) hat der Store Filme?

SELECT first\_name, last\_name FROM actor;

- Beschrifte die Resultat-Tabelle mit Vorname und Nachname

SELECT first\_name as Vorname, last\_name as Nachname FROM actor;

- sortiert nach Nachname, Vorname

SELECT first\_name, last\_name

FROM actor

ORDER BY last\_name, first\_name;

- liste alle Schauspieler-Nachnamen auf

SELECT last\_name

FROM actor

ORDER BY last\_name, first\_name;

- liste alle Schauspieler-Nachnamen auf (jeder Nachname aber nur einmal)

SELECT DISTINCT last\_name

FROM actor

ORDER BY last\_name, first\_name;

- wie viele Schauspieler hat es im Store?

SELECT count(\*) FROM actor;

- wie viele verschiedene Nachnamen gibt es bei den Schauspielern

SELECT count(distinct last\_name)

FROM actor

ORDER BY last\_name, first\_name

- liste alle Schauspielern (Vorname und Nachname) auf, welche **Kirsten** zum Vornamen heissen?

SELECT first\_name, last\_name

FROM actor

WHERE first\_name = 'Kirsten'

ORDER BY last\_name, first\_name desc;

## Abfragen mit Function-calls

- liste alle Schauspielern (Vorname, Nachname und LAST\_UPDATE [yyyy-mon-dd]), bei welchen der Vorname mit A beginnt und der zweitletzte Buchstabe im Vornamen ebenfalls ein A ist.

<https://dev.mysql.com/doc/refman/5.5/en/date-and-time-functions.html>

SELECT

first\_name as Vorname,

last\_name as Nachname,

Date\_Format(last\_update,'%Y-%M-%d') as LastUpdate

FROM actor

WHERE first\_name like 'A%A\_';

- liste alle Schauspielern (Vorname, Nachname und LAST\_UPDATE [yyyy-mon-dd]), bei welchen der **Vorname** mit **A** beginnt und der **zweitletzte Buchstabe im Vornamen** ebenfalls ein **A** ist sowie der Nachname **nicht** **ALLEN** oder **BAILEY** ist.

SELECT

first\_name as Vorname,

last\_name as Nachname,

Date\_Format(last\_update,'%Y-%M-%d') as LastUpdate

FROM actor

WHERE (not (first\_name like 'A%A\_')) and   
 (last\_name not in ('Allen','Bailey'));

- Liste film\_id, title, rating, special\_features von der Tabelle film auf.

select film\_id, title, rating, special\_features from film;

Was ist der Type der Attribute rating und special\_features?

Listen Sie nur die Raws auf, welche im rating eine PG haben und in den special\_features Trailes enthalten.

select film\_id, title, rating, special\_features

from film

where rating = 'PG' and find\_in\_set('Trailers', special\_features)

- In der staff table hat es ein Attribute picture vom Type BLOB. Googeln Sie, was das für ein Type ist und laden Sie ein Bild für eine Record (staff\_id = 1) in dieses Feld.

update staff

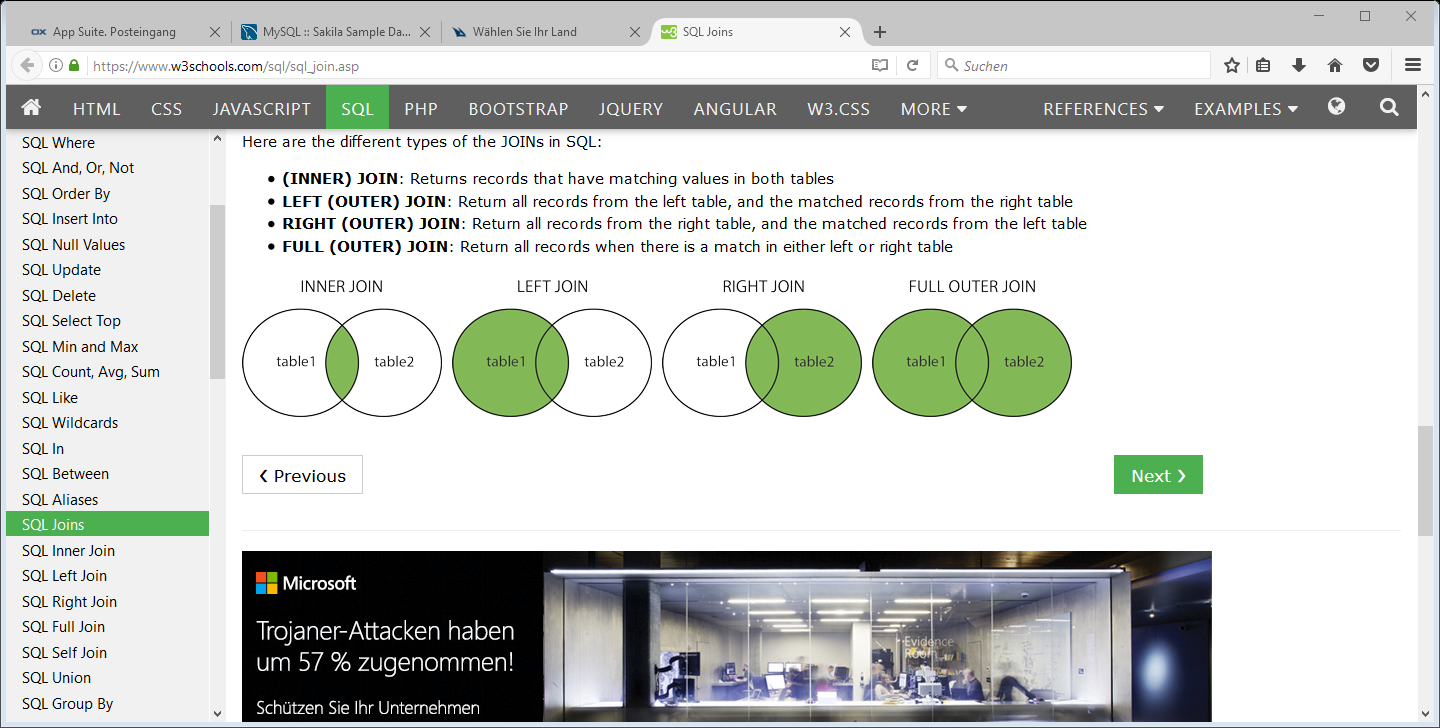
set picture=LOAD\_FILE('C:\Users\admin\Pictures\portraet.jpg')

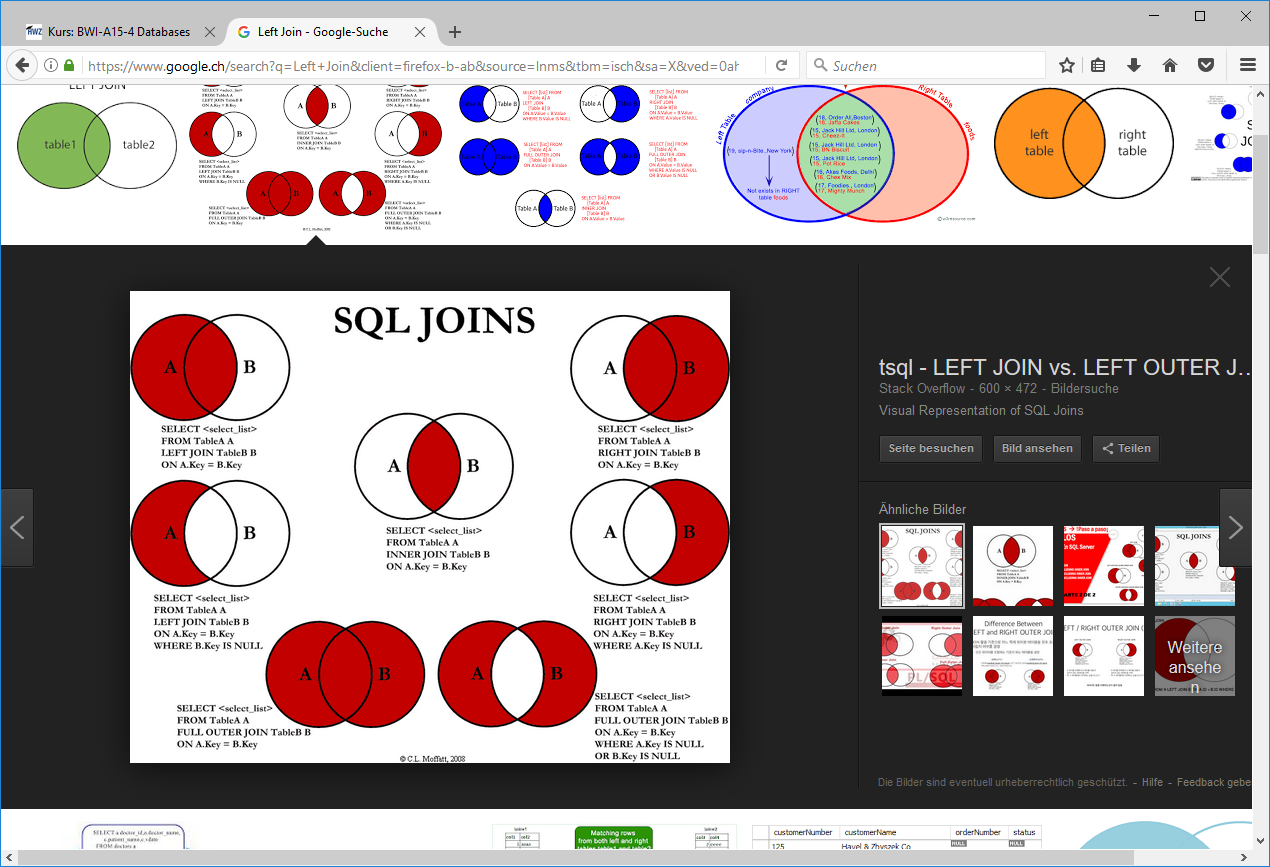
where staff\_id=2;

- Erstellen Sie eine Liste mit allen Namen und Vorname von staff, welche nur den Anfangsbuchstaben des Vornamen mit einem . getrennt vom vollständigen Nachnamen auflistet.<https://dev.mysql.com/doc/refman/5.7/en/string-functions.html>

select concat(left(first\_name,1), '. ',last\_name) as staff from staff;

## Joins

****



Full Outer Join funktioniert bei MySql nicht 🡺 muss über eine UNION gemacht werden

- Erstellen sie eine Liste mit allen DVD-Titeln und der Originalsprache.

* Wieviele Einträge hat dir Tabelle **film** und die Tabelle **language**?

select count(\*) from film; -- 1000 rows

select count(\*) from language; -- 6 rows

* Checken sie, ob es Filme gibt, bei denen die Originalsprache gesetzt ist?

select count(\*) from film where film.original\_language\_id is not null;

Falls Sie 0 bekommen, müssen Sie mit folgenden Anweisungen bei 2 Filmen die Sprache setzen:

UPDATE film SET original\_language\_id=1 WHERE film\_id=1;

UPDATE film SET original\_language\_id=6 WHERE film\_id=2;

* Listen Sie alle DVD Titel mit der Originalsprache auf. Wieviele Records bekommen Sie?

select film.title, language.name

from film, language

where language.language\_id = film.original\_language\_id; -- 2 rows

* Sie bekommen nun nur eine Liste mit den 2 DVDs, welche auch eine Originalsprache gesetzt haben (Dies entspricht einem Inner-Join.

select film.title, language.name

from film

inner join language

on language.language\_id = film.original\_language\_id;

-- 2 rows

* Mit einem OUTER join bekommen sie eine Liste mit alle 1000 DVDs

select film.title, language.name

from film

left join language

on language.language\_id = film.original\_language\_id;

-- 1000 rows

* Erstellen sie gleiche List mit einem right join.

select film.title, language.name

from language

right join film

on language.language\_id = film.original\_language\_id;

-- 1000 rows

* Nun müssen Sie die beiden eigenen Einträge wieder löschen.

UPDATE film SET original\_language\_id=null WHERE film\_id in (1,2);

- Erstellen Sie eine Liste mit allen ausgeliehenen DVDs und listen Sie die Vor-, Nachnamen und Telefonnummern der Ausleiher auf sowie den Titel der DVD welche am 27.5.2005 zurückgegeben worden sind.

-- Date\_Format(return\_date,'%Y%m%d') = '20050527'

-- return\_date = STR\_TO\_DATE('20050527','%Y%m%d')

SELECT

\*

FROM

rental

WHERE

return\_date IS NOT NULL

AND DATE\_FORMAT(return\_date, '%Y%m%d') = '20050527'

ORDER BY return\_date; --(49 rows)

SELECT

CONCAT(LEFT(customer.first\_name, 1),

'. ',

customer.last\_name) AS renter,

film.title

FROM

rental

INNER JOIN

customer ON rental.customer\_id = customer.customer\_id

INNER JOIN

inventory ON rental.inventory\_id = inventory.inventory\_id

INNER JOIN

film ON inventory.inventory\_id = film.film\_id

WHERE

return\_date IS NOT NULL

AND DATE\_FORMAT(return\_date, '%Y%m%d') = '20050527'

ORDER BY return\_date; --(12 rows)

SELECT

CONCAT(LEFT(customer.first\_name, 1),

'. ',

customer.last\_name) AS renter,

address.phone,

film.title

FROM

rental

INNER JOIN

customer ON rental.customer\_id = customer.customer\_id

INNER JOIN

address ON customer.address\_id = address.address\_id

INNER JOIN

inventory ON rental.inventory\_id = inventory.inventory\_id

INNER JOIN

film ON inventory.inventory\_id = film.film\_id

WHERE

return\_date IS NOT NULL

AND DATE\_FORMAT(return\_date, '%Y%m%d') = '20050527'

ORDER BY return\_date; --(12 rows)

Wieso hat es nun nur noch

select

concat(left(customer.first\_name,1), '. ',customer.last\_name) as renter,

address.phone

from rental

INNER JOIN customer on rental.customer\_id = customer.customer\_id

INNER JOIN address on customer.address\_id = address.address\_id

where return\_date is not NULL and

Date\_Format(return\_date,'%Y%m%d') = '20050527'

order by return\_date;

## Meta-Daten abfragen

- liste alle Tabellen in der DB (im Schema) sakila auf und zeige deren Type an.

<https://dev.mysql.com/doc/refman/5.7/en/tables-table.html>

SELECT table\_schema, table\_name, table\_type

FROM INFORMATION\_SCHEMA.TABLES

WHERE table\_name Like '%film%';

- liste alle Attribute (mit Type) aller Tabellen in der DB (im Schema) sakila auf.

<https://dev.mysql.com/doc/refman/5.7/en/columns-table.html>

SELECT table\_schema, TABLE\_NAME, column\_name, DATA\_TYPE, IS\_NULLABLE, COLUMN\_DEFAULT

FROM INFORMATION\_SCHEMA.COLUMNS

WHERE table\_schema = 'sakila'

order by Table\_NAME, column\_name;

# Views / Functions / Stored procedures

## Views

Kreieren sie eine View mit alle Städte und deren Länder.

DROP VIEW IF EXISTS test\_city\_country;

CREATE VIEW test\_city\_country AS

SELECT city.city, country.country FROM city

INNER JOIN country ON city.country\_id=country.country\_id

ORDER BY city.city ASC;

select \* from test\_city\_country;

## Functions

Schreiben sie eine eigene Function. Diese Function nimmt ein String-Parameter entgegen und gibt einen String in der Form **Hello : *inString*** zurück. Nach der Implementation testen sie die Function aus.

DROP FUNCTION IF EXISTS HelloFct;

Delimiter //

CREATE FUNCTION HelloFct(p\_input\_string CHAR(20)) RETURNS CHAR(50)

BEGIN

RETURN concat('Hallo: ', p\_input\_string);

END//

select HelloFct('Walti');

## Stored procedures

Schreiben sie eine Stored-Procedure bei welcher 2 Parameter übergeben warden können. Der erste Parameter ist ein Land und der zweite ob case-sensitve oder nicht gesucht werden soll.

DROP PROCEDURE IF EXISTS test\_searchCountry;

Delimiter //

CREATE PROCEDURE `test\_searchCountry`(IN searchQuery VARCHAR(20), IN caseSesitive BOOLEAN)

BEGIN

IF caseSesitive THEN

SELECT country FROM sakila.country WHERE country LIKE BINARY searchQuery;

ELSE

SELECT country FROM sakila.country WHERE country LIKE searchQuery;

END IF;

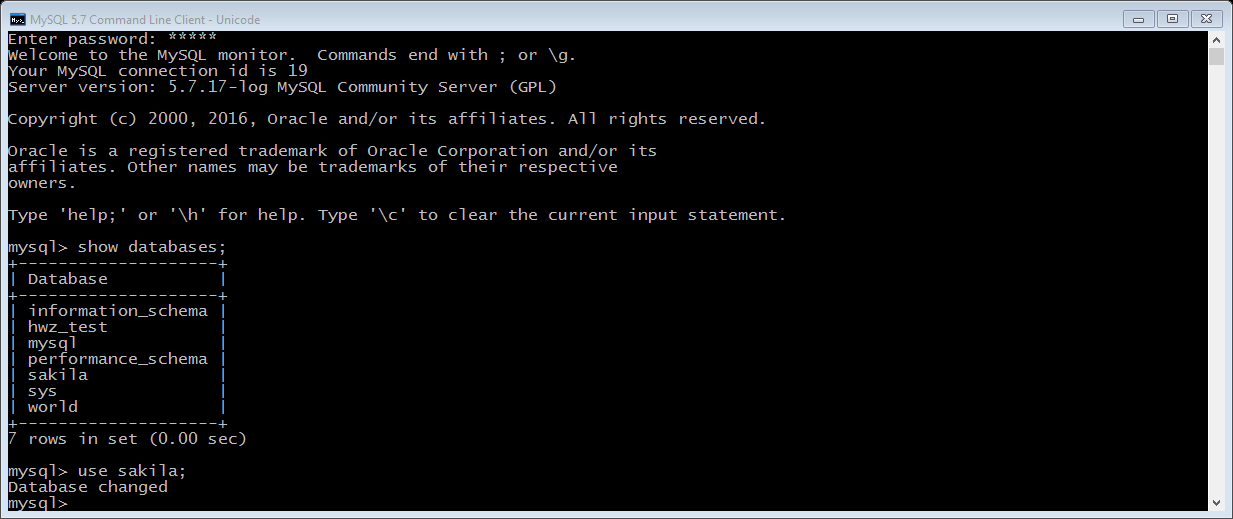
END//

call sakila.test\_searchCountry('GermanY', false);

# Using Command Line Client

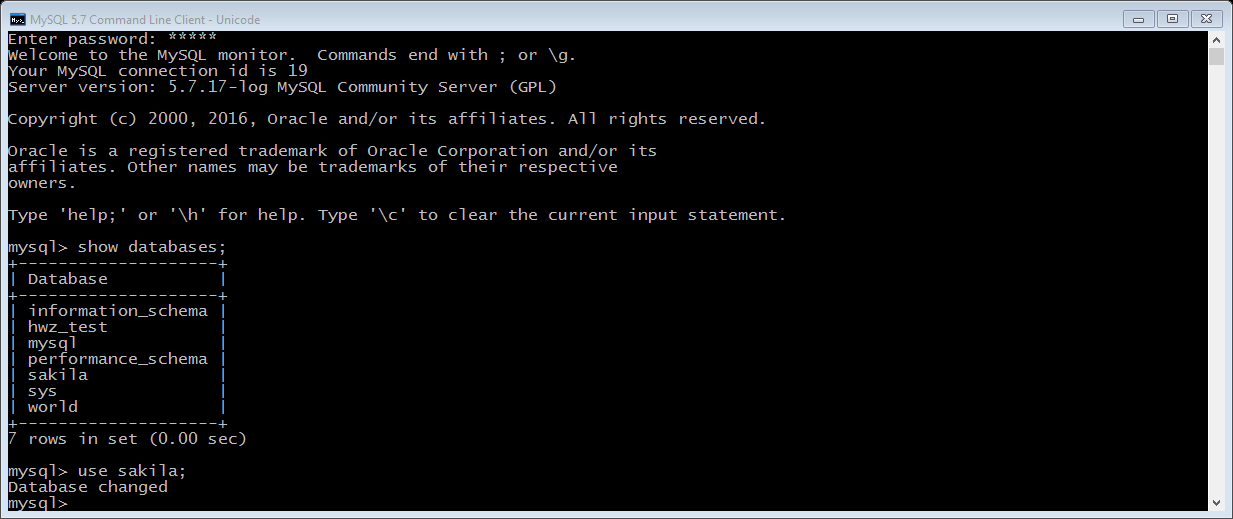
Start -> MySql Command Line Client - Unicode

MySQL Root Password: admin



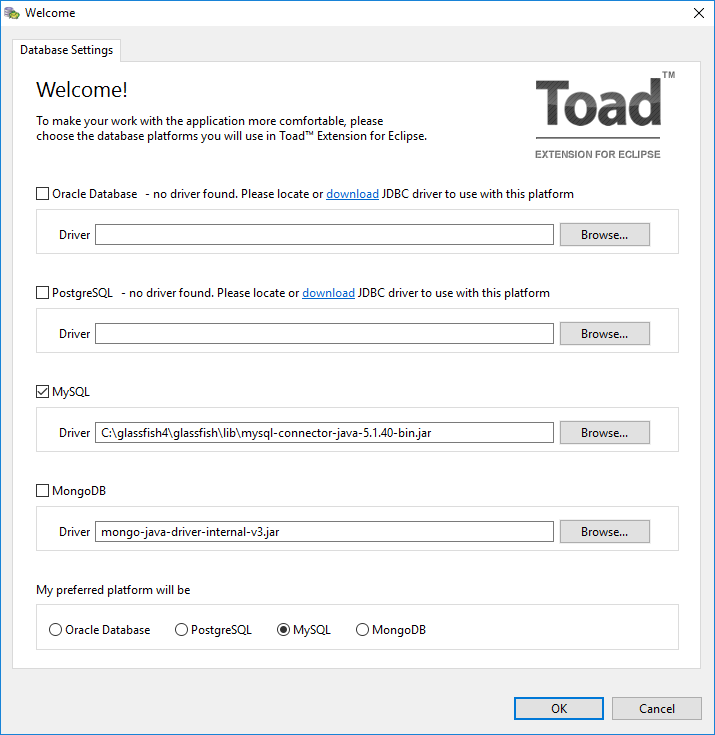
List all defined DBs: show databases;

Set a DB: use sakila;



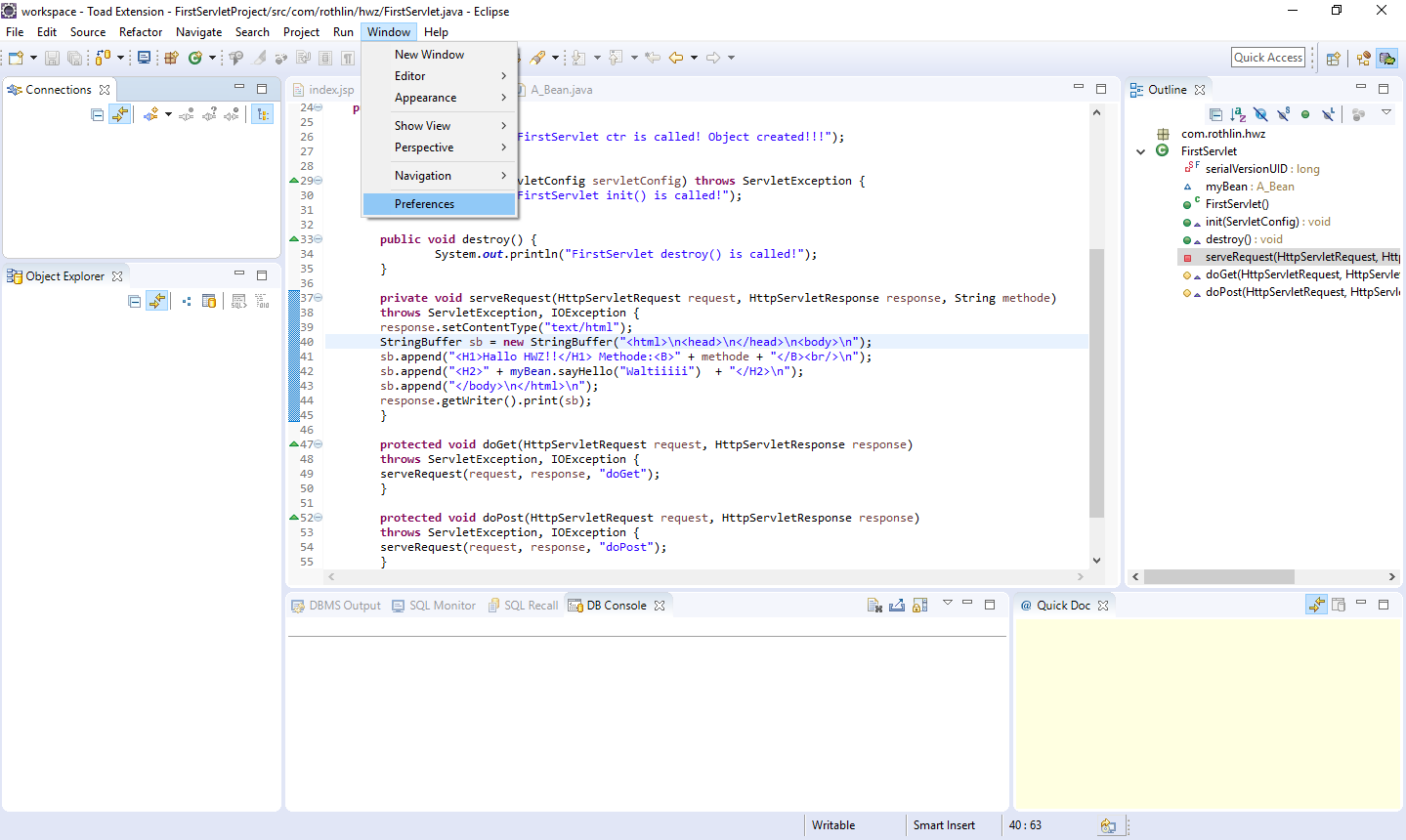
# Using Eclipse TOAD extension

## Add MySQL JDBC driver (Original in Anleitung\_Application\_Server\_Setup\_xx.docx)

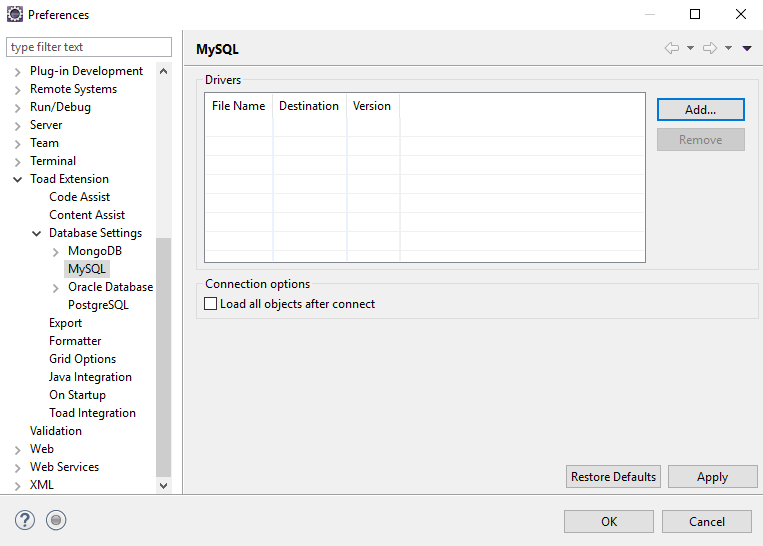


Oder falls das Window nach den restart nicht erscheint….

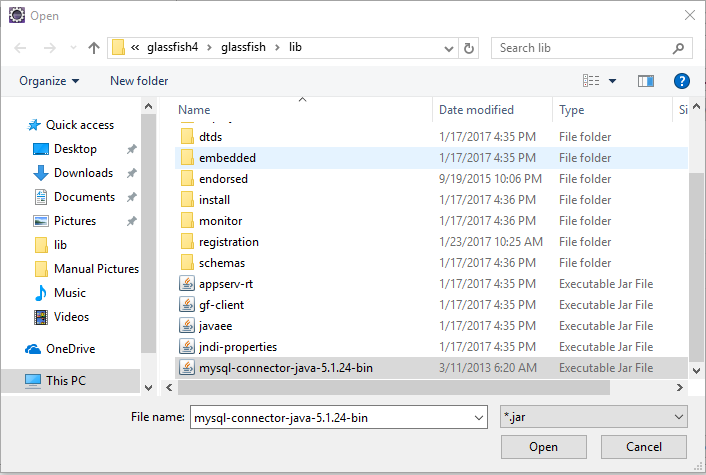
Eclipse -> Window -> Preferences



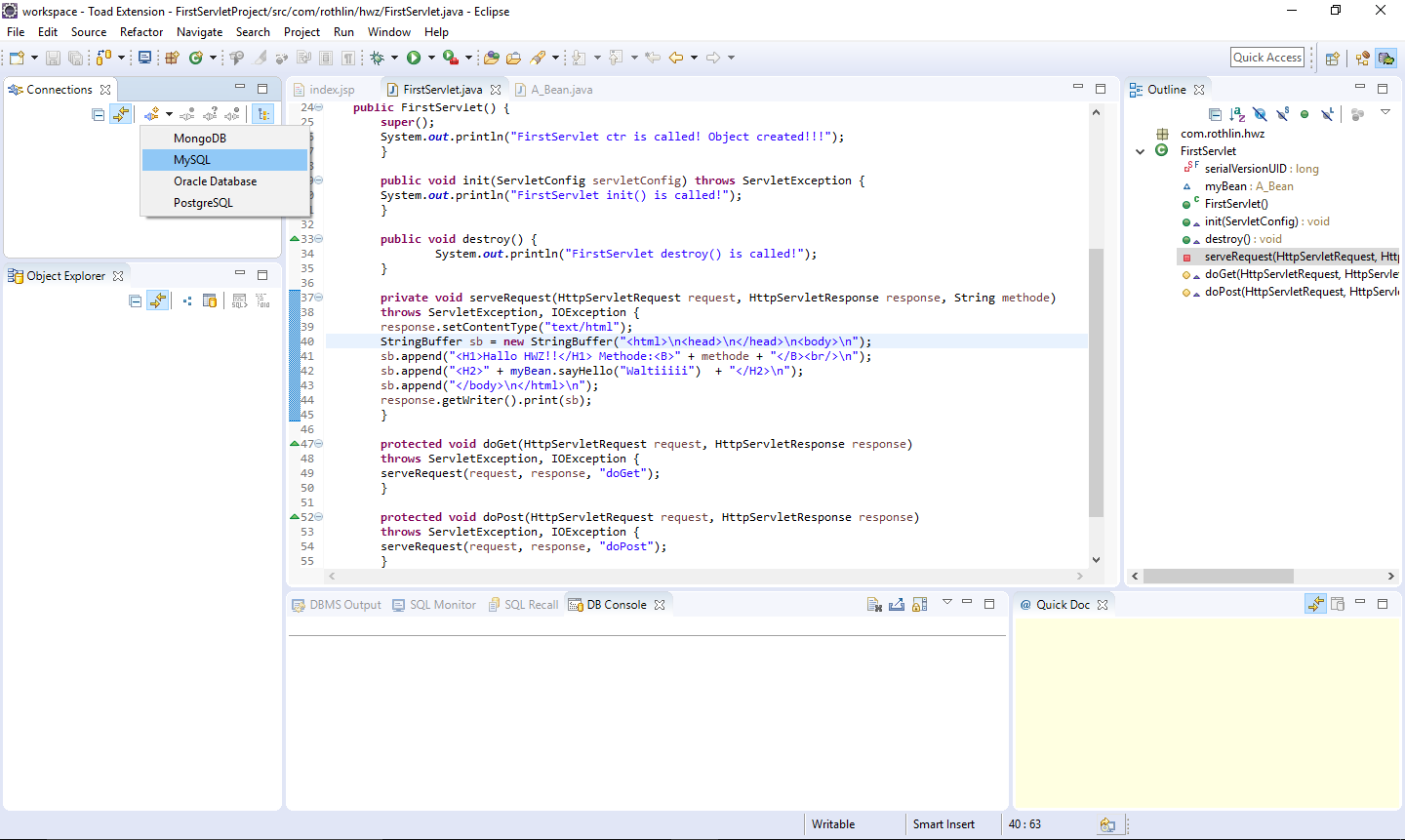
Toad Extension -> Database Settings -> MySQL -> Add



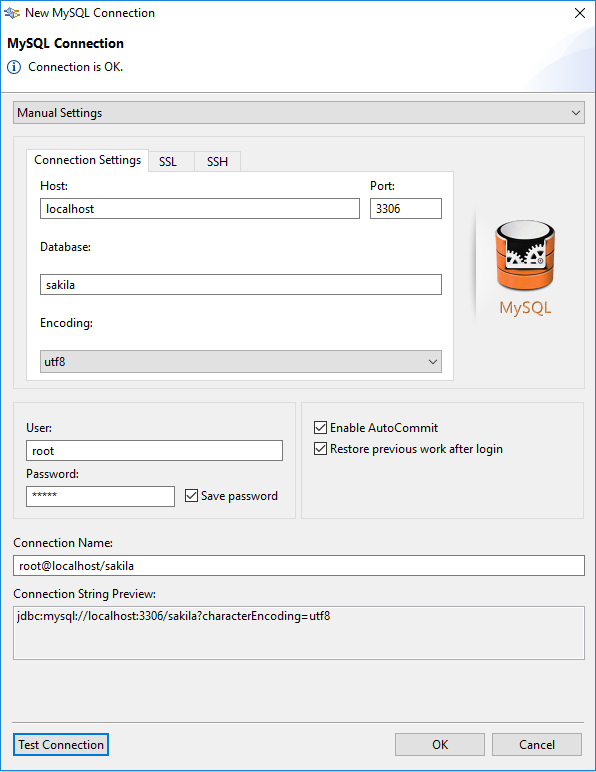
Add the MySQL JDBC driver, which is located at C:\glassfish4\glassfish\lib.



## Add MySQL connection

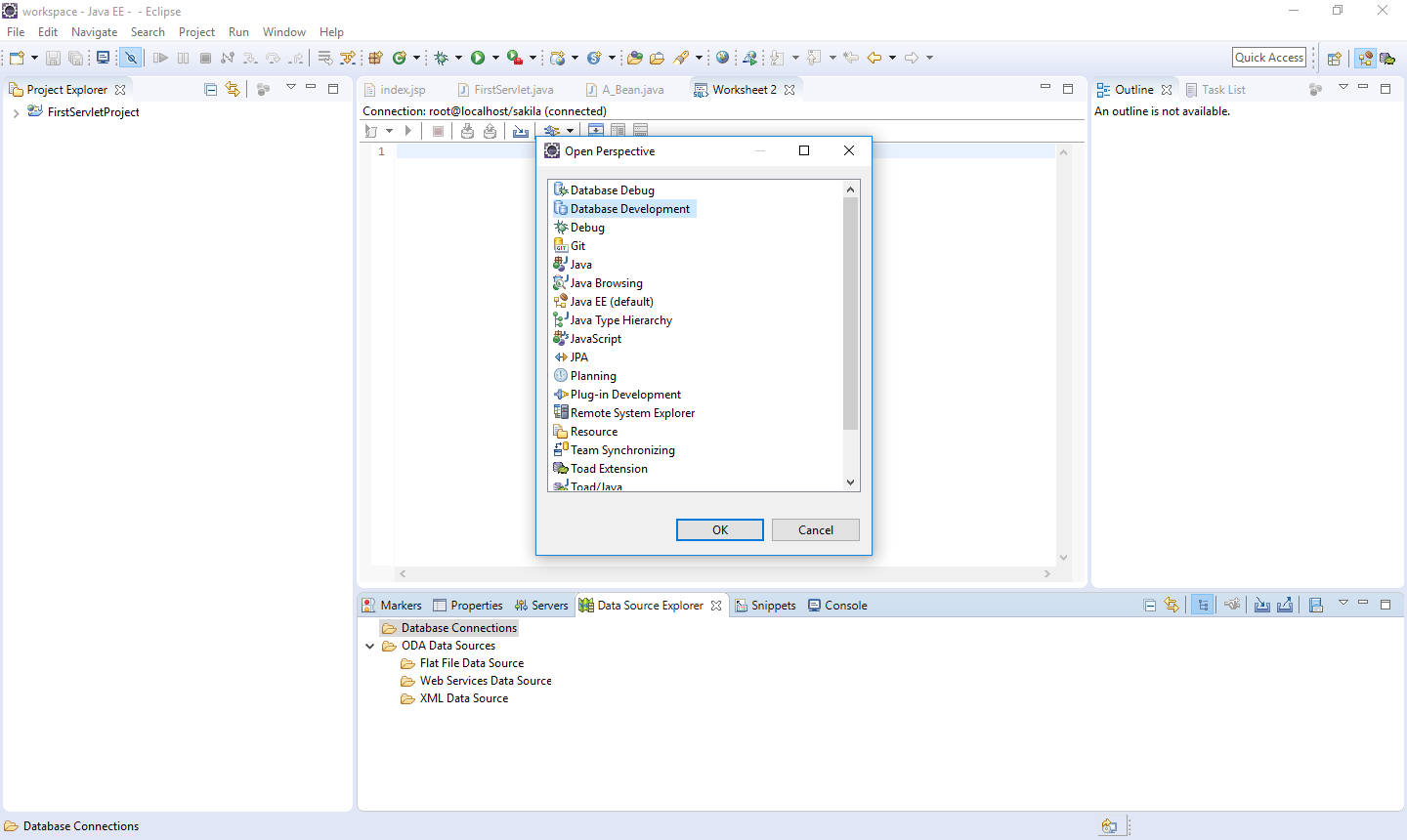


MySQL Root Password: admin

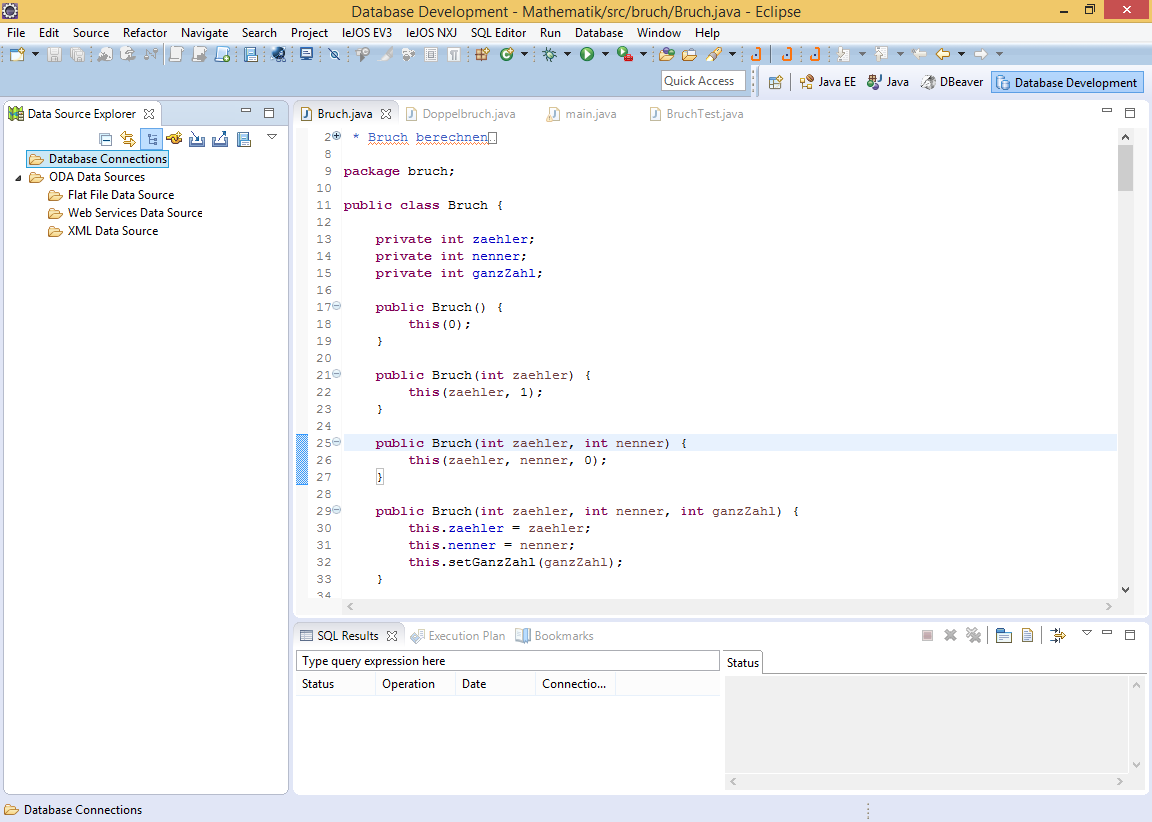


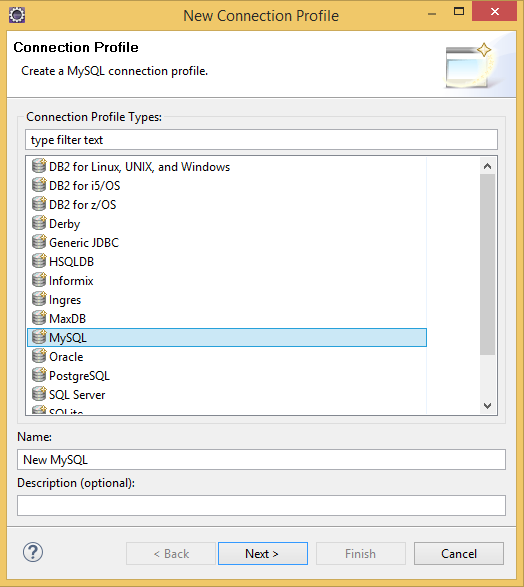
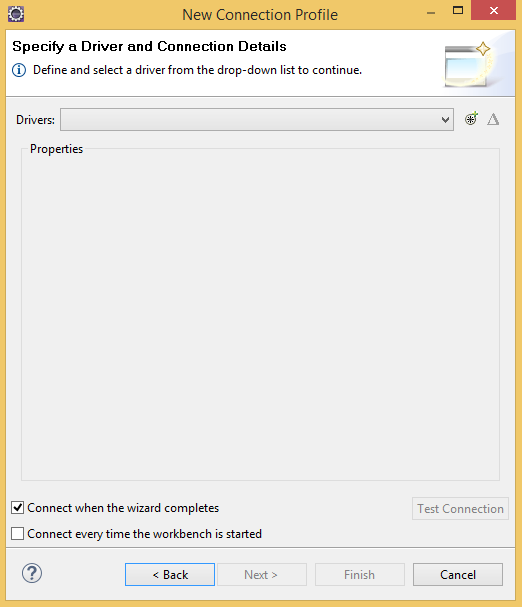
## Create DB-Connection from TOAD Plugin

Add new Database Development Interface

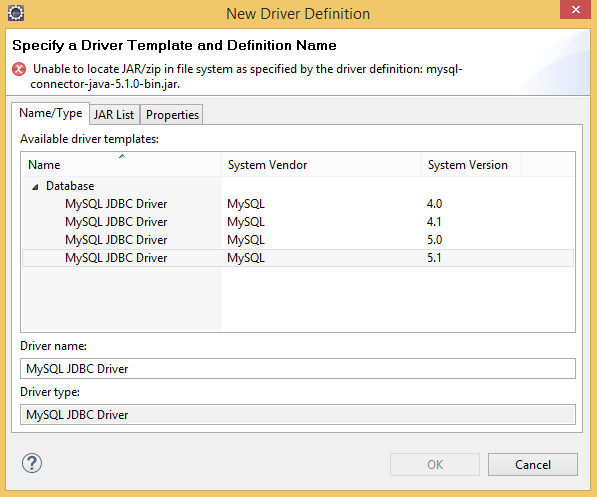


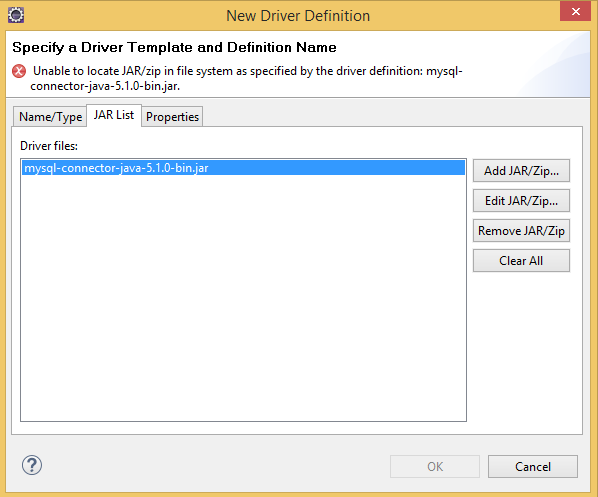
Right Mouse-Click on Database Connections 🡪 new….



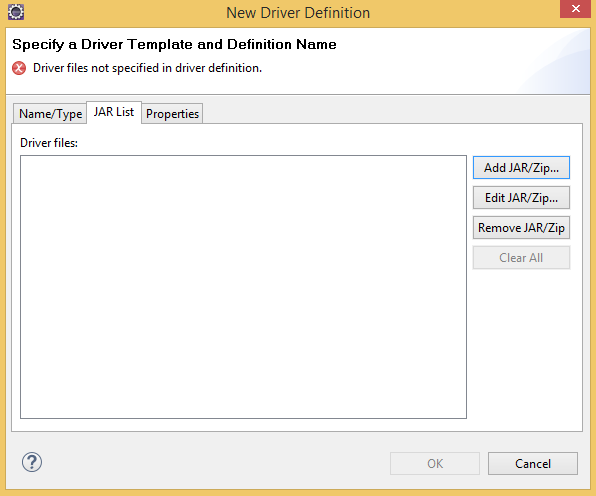
 

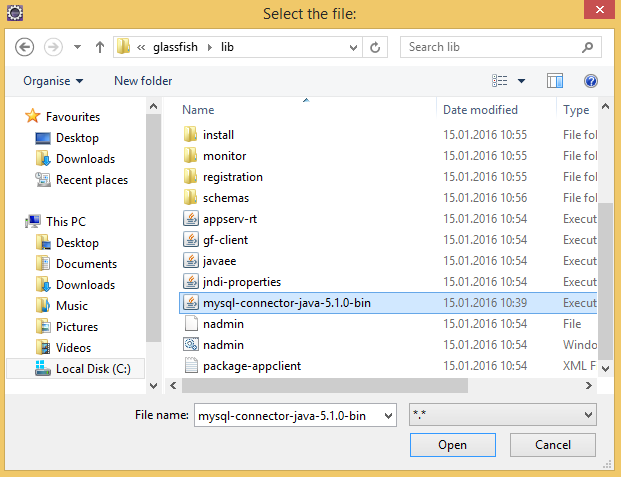
Select the MySQL JDBC Driver with the matching System Version (January 2015: 5.1) and select the «Jar List»-Tab. Remove the current Jar in the list and click on “Add JAR/Zip…”.

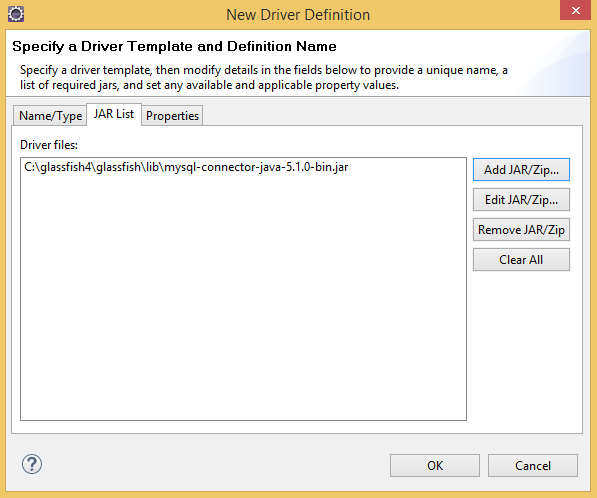




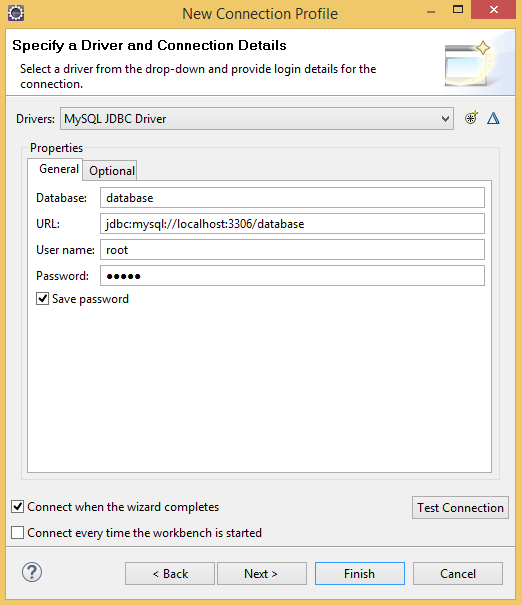
Select “Add JAR/Zip”. Then navigate to the installation directory of the glassfish-server and select the driver in the directory …/glassfish/lib/ select mysql-connector-java-5.1.0-bin and click on “Open”.







MySQL Root Password: admin



sakila

jdbc:mysql://localhost:3306/sakila

root

admin