Thema:	Datenbanknutzung an der Hochschule und zu Hause	
Autor:	Prof. Dr. Stephan Kleuker	
Version / Datum:	5.0.3 / 1.9.2015	
Empfänger:	Teilnehmer der Lehrveranstaltung Datenbanken	

Diese Notiz fasst wichtige Informationen zur Nutzung der Oracle-Datenbank in der Lehrveranstaltung Datenbanksysteme zusammen. Für zu Hause wird die zu Lernzwecken freie Oracle XE-Variante vorgeschlagen, die sich nur in wenigen Details von der an der Hochschule installierten Variante unterscheidet.

Die Bilder in diesem Dokument können teilweise zu älteren Versionen gehören, wobei sich die Funktionalität nur unwesentlich geändert hat. Aktualisierungshinweise von Lesern sind erwünscht.

1	Installation von Oracle XE				
	1.1 Download				
	1.2	Einschub: Oracle-Account einrichten	9		
	1.3	Weiter mit Download	12		
	1.4	Installation	12		
	1.5	Dienste von Oracle XE	20		
2	Ein	nrichten eines Nutzers in Oracle XE	25		
3	Installation des SQL-Developers		34		
	3.1	Download	34		
	3.2	Installation	38		
4	Ers	te Nutzung des SQL Developers	40		
	4.1	Erster Start	40		
	4.2	Verbindung einrichten	43		
	4.3	Normaler Start des SQL Developers	47		
	4.4	SQL Developer zusammen mit der HS-Datenbank	48		
5	Einspielen von SQL-Skripten50				
6	6 Erzeugen und Bearbeiten von Tabellen				

6.1	Arbeitsblatt erstellen	54
6.2	SQL-Befehl eingeben und ausführen	55
6.3	Ausführung einzelner SQL-Anfragen	58
7 Ti	pps zum SQL-Developer	63
7.1	Arbeitsblätter	63
7.2	Datenbank administrieren	64
8 Er	stellung von PL/SQL-Funktionen und Prozeduren	71
8.1	Ausgaben sichtbar werden lassen	71
8.2	Prozedur erstellen	72
8.3	Prozedur ausführen	76
8.4	Ausführung mit EXECUTE	79
8.5	Nutzung des Debuggers	80
8.6	Einschub: Mangelnde Debug-Rechte	82
8.7	Einzelne Debug-Schritte	83
9 Er	stellung von PL/SQL-Triggern	85
10 JE	DBC	87
11 SC	QLite	96
11.1	Installation	96
11.2	Erste Nutzung	97
11.3	SQLite Database Browser	102
12 M	ongoDB	112
12.1	Installation	112
12.2	Start der Datenbank	113
12.3	Erste Nutzung	116
12	2.3.1 JavaScript	117

12.3.2 Anlegen der Datenbank	118
12.3.3 Ein- und Ausgabe einer Collection	118
12.3.4 Cursor	120
12.3.5 Find	122

1 Installation von Oracle XE

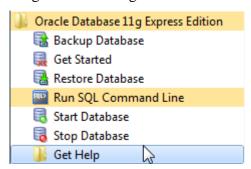
In der Hochschule steht eine Oracle-Datenbank zur Verfügung, die von jedem Rechner aus mit einem Client-Programm, z. B. SQL-Developer, genutzt werden kann. Dazu wird ein vom Datenbank-Administrator eingerichteter Nutzer-Zugang benötigt.

Möchte man die Praktikumsaufgaben auch zu Hause lösen, gibt es vielfältige Möglichkeiten, da es einige gute zum privaten Gebrauch freie Datenbanken (z. B. Postgres, MaxDB, Firebird, Apache Derby) gibt, die mit ähnlicher Funktionalität wie Oracle im Bereich der in der Veranstaltung genutzten Features, zur Verfügung stehen. Die drei großen Hersteller (Oracle, IBM und Microsoft) bieten mittlerweile auch freie Varianten ihrer professionellen Produkte an. Da an der Hochschule Oracle genutzt wird und Oracle einige Informationen zu den vielfältigen Nutzungsmöglichkeiten frei für den persönlichen Gebrauch anbietet, wird hier die Installation und Nutzung der freien Oracle XE-Version mit dem ebenfalls freien SQL-Developer vorgestellt.

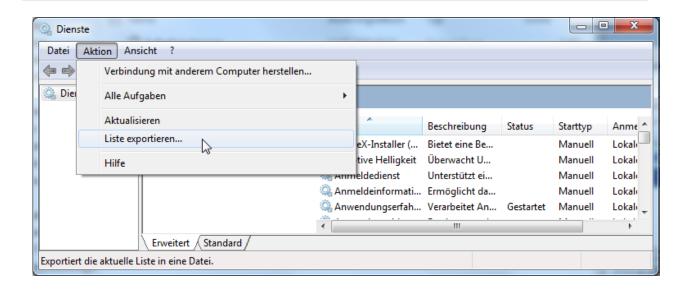
Generell ist bei der Installation einer größeren Datenbank zu beachten, dass das System in zwei Teile zerfällt. Das Erste ist die eigentliche Datenbank, genauer der Datenbank-Server, der die Zugriffe und die Daten verwaltet und sehr viel Speicher und Rechenzeit benötigt. Der zweite Teil ist der Client, mit dem der Nutzer auf die Datenbank zugreifen kann. Client und Server laufen typischerweise auf unterschiedlichen Rechnern, können aber auch auf dem gleichen Rechner installiert sein. Zu beachten ist weiterhin, dass der Datenbankserver einiges an Leistung eines Rechners in Anspruch nimmt. Oracle XE, hier die 10g-Variante, lässt sich z. B. nur ab 256 MB-Hauptspeicher installieren. Sonst erhält man folgende Information.



Für die genannten Systeme gilt allerdings, dass es immer Möglichkeiten gibt, den Server von Hand ein- bzw. auszuschalten. Dies ist meist unter Windows über den eigentlichen Programmstart möglich.

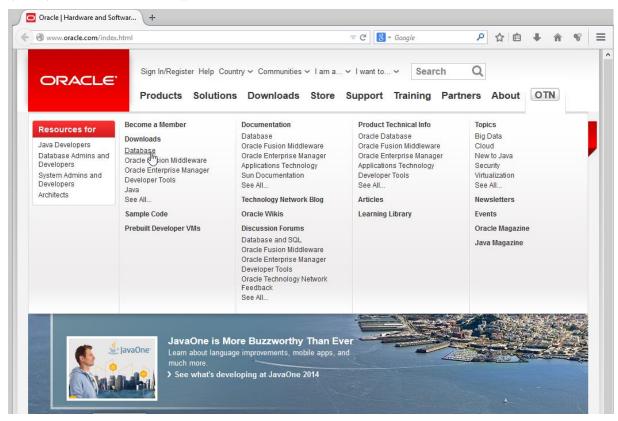


Trotzdem sollte man die Prozesse kennen, die auf dem eigenen Rechner laufen. Unter Windows kann man dies z. B. unter Start -> Systemsteuerung -> System und Sicherheit -> Verwaltung -> Dienste sichtbar machen und die Starteigenschaften der Prozesse verändern. Unter Win7 gibt es die Möglichkeit, die aktuellen Prozessnamen zu exportieren, so dass ein Vergleich vor und nach der Installation möglich ist.



1.1 Download

Ein Weg zum Download führt über die Hauptseite von Oracle http://www.oracle.com/index.html, man klappt rechts-oben "Oracle Technology Network" (OTN) herunter und wählt "Database".

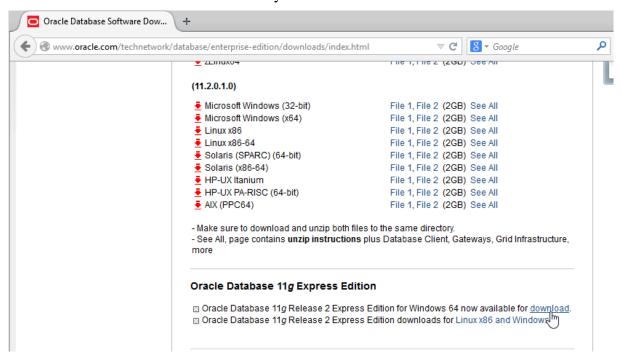


Auf der folgenden Seite kann man sich eine vollständige Oracle Database herunterladen. Das dauert sehr lange und ist nur sinnvoll, wenn man z. B. einen virtuellen Rechner oder eine zweite Betriebssystem-Installation hat, auf der diese getrennt gespeichert werden kann.

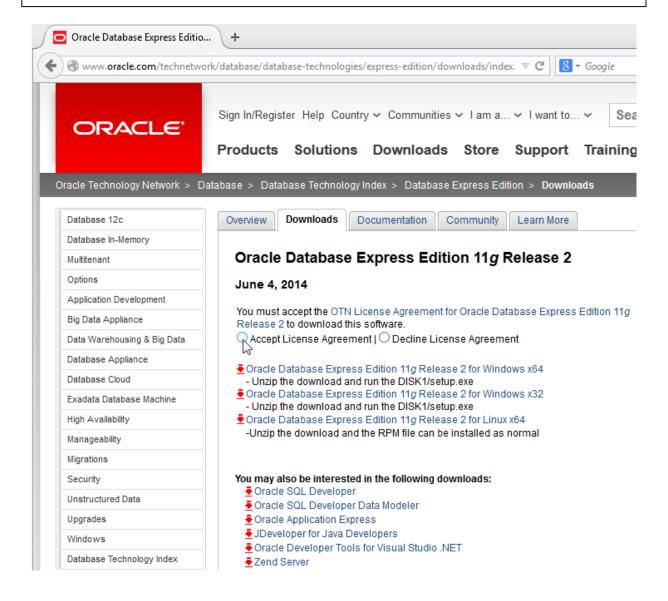
Unten auf der Seite kommt man zur Oracle Database 11g Express Edition, die zum Erlernen relationaler Datenbanken völlig ausreicht. Man beachte, dass man bereits oben auf der Seite die License Agreements akzeptieren muss.



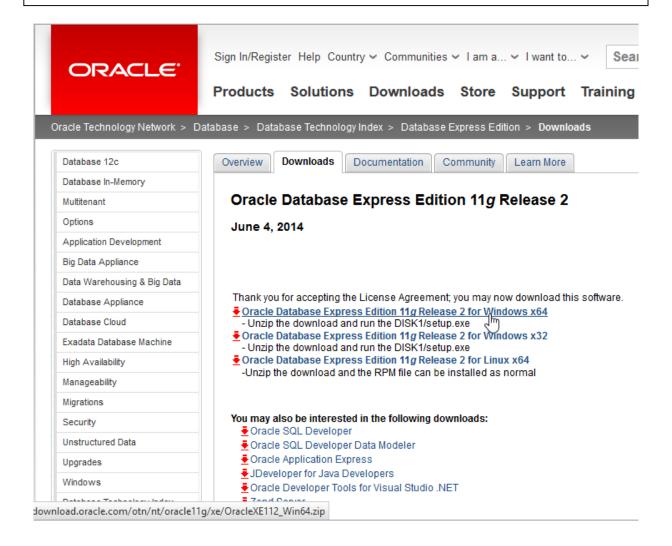
Den eigentlichen Download-Link findet man weiter unten. Man beachte, dass der obere Link für Windows 64 und der untere für x86-Systeme ist.



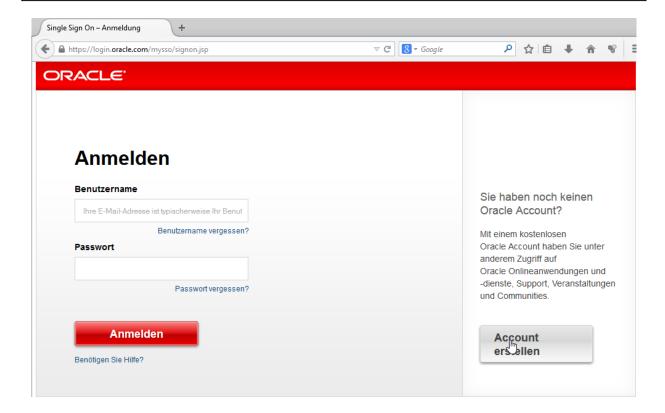
Bevor man eine passende Express Edition herunter laden kann, muss die Lizenzbedingung oben akzeptiert werden.



Nun kann man den Link zur Datenbank anklicken. Hier stehen Versionen für Windows x64 und x32 (x86) zur Verfügung. Man beachte, dass auf dieser Seite auch ein Link zum SQL Developer angeboten wird. Diese Software wird später zum Zugriff auf die Datenbank auch benötigt.



Danach muss man sich leider bei Oracle anmelden. Ein sehr unglücklicher Ansatz, dies erst hier einzufordern. Vielleicht sagt dies einiges über Oracle aus. Der Account ist ohne direkte Kosten.



1.2 Einschub: Oracle-Account einrichten

Die Account-Einrichtung wird hier gezeigt, birgt aber keine Überraschungen.



MeinProfil - Account anlegen

Haben Sie schon ein Konto? Anmeldung 👯

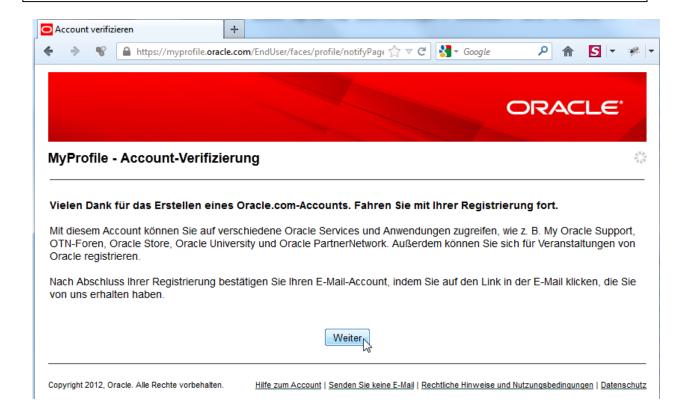
Bitte geben Sie die folgenden Informationen an, die zum Erstellen des Oracle.com-Kontos erforderlich sind. Geben Sie auch die Einstellungen für die E-Mail-Kommunikation an.

Mit dem Oracle.com-Account haben Sie Zugang zu einer Vielzahl von Online-Anwendungen und -Services, wie Oracle Store, My Oracle Support, Marketing-Veranstaltungen und Oracle PartnerNetwork. Wenn Sie sich für einen dieser Services registrieren, werden Sie weitergeleitet, wenn mehr Informationen erforderlich sind, um den Prozess abzuschließen.

* Pflichtfeld					
Account-Daten					
* E-Mail	@hs-osnabrueck.de	Bitte geben Sie eine gültige E-Mail-Adresse als Benutzernamen an. Wir empfehlen die Angabe Ihrer geschäftlichen E-Mail-Adresse. Passwörter müssen aus mindestens 8 Zeichen bestehen, Groß- und Kleinbuchstaben sowie mindestens eine Zahl enthalten.			
★ Passwort	•••••				
* Passwort bestätigen	•••••				
Persönliche Daten					
* Vorname (Rufname)	Stephan				
Zweiter Vorname					
* Nachname	Kleuker				
* Funktion	Lehrender				
* Land	Deutschland				
* Bundesland/Kanton	Niedersachsen ▼				
* Stadt	Osnabrück				
* Unternehmen	Hochschule Osnabrück				
* Adresse 1	Postfach 1940				
Adresse 2					
* Postleitzahl	49009				
* Telefon (geschäftlich)	015212345678				
	☐ Ja, ich möchte Mitglied im Oracle Technology Network (OTN) werden				
Ja, ich möchte Mitglied in der Oracle C-Central Community werden					
		Produkten, Services und Veranstaltungen , klicken Sie auf den Link "Senden Sie keine E-Mail"			
Sie sind sich darüber im Klaren und erklären sich damit einverstanden, dass die Verwendung der Oracle Website den Oracle.com-Nutzungsbedingungen und der Oracle Datenschutzrichtlinie unterliegt.					
	Anlegen				

Copyright 2012, Oracle. Alle Rechte vorbehalten.

 $\underline{\textit{Hilfe zum Account}} \mid \underline{\textit{Senden Sie keine E-Mail}} \mid \underline{\textit{Rechtliche Hinweise und Nutzungsbedingungen}} \mid \underline{\textit{Datenschutz}}$

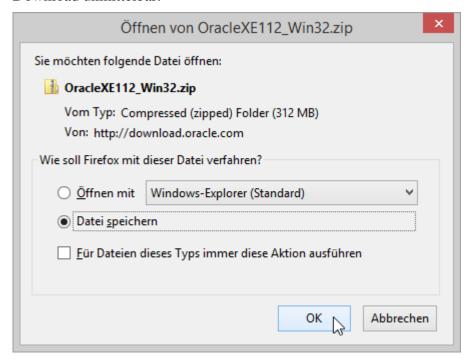


Man erhält recht schnell eine E-Mail, in der man mit Hilfe eines Links bestätigen kann, dass man angemeldet werden wollte. Hier trägt man seine Anmeldedaten ein und kommt auf eine Bestätigungsseite.



1.3 Weiter mit Download

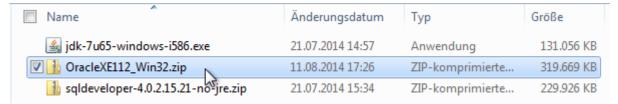
Ist man bereits angemeldet oder hat jetzt seine Account-Daten eingegeben, startet der Download unmittelbar.



Vor der Installation sollte man bedenken, dass es sich bei der Datenbank um einen Server-Prozess handelt, so dass installierte Firewalls zunächst abgeschaltet werden müssen. Nach der Installation der Datenbank kann die Firewall wieder eingeschaltet werden, wobei abhängig vom eingesetzten Produkt individuelle Einstellungen vorzunehmen sind. Dabei muss nur der lokale Zugriff auf den eigenen Datenbankserver (unter localhost oder 127.0.0.1) möglich sein.

1.4 Installation

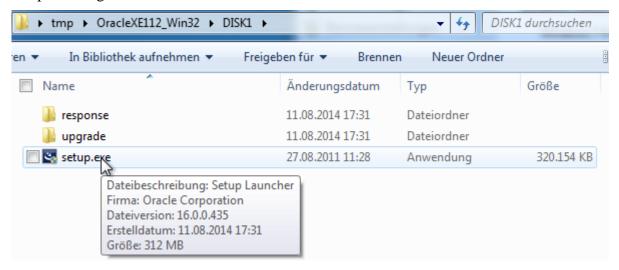
Die Installation nutzt die beiden folgenden Dateien, die zweite kann später geladen werden, siehe Kapitel 3.



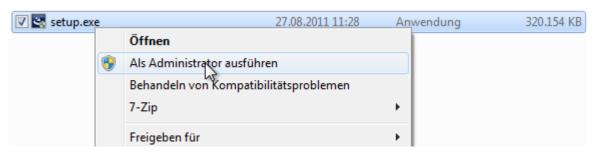
Diese können nach einer Anmeldung bei Oracle herunter geladen werden. Es wird natürlich nicht garantiert, dass der vorgestellte Weg auch für neuere Versionen funktioniert. Bei der hier genutzten SQL-Developer-Variante wird davon ausgegangen, dass eine Java-Version ab 1.7 installiert ist, die ebenfalls von Oracle erhältlich ist und in der oberen Abbildung gezeit wird. Hier sollte der Informatiker immer ein JDK (Java Development Kit) und nicht nur eine JRE (Java Runtime Environment) nutzen, da Java als Standardprogrammiersprache beherrscht

werden sollte. Eine Java-Installation ist z. B. in http://home.edvsz.hs-osnabrueck.de/skleuker/querschnittlich/NetbeansNutzung.pdf beschrieben.

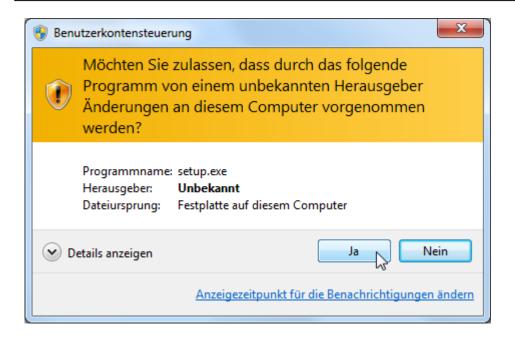
Das Zip-File der XE-Version wird ausgepackt und die darin im Ordner DISK1 enthaltene Datei setup.exe ausgeführt.



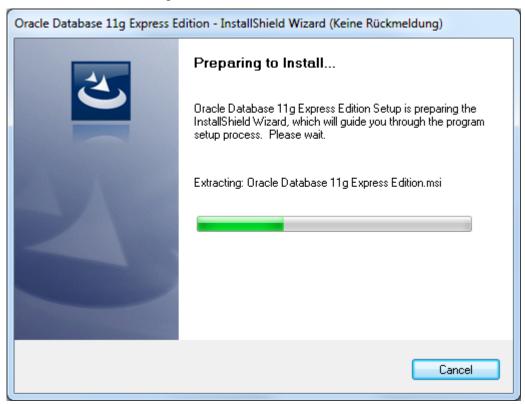
Wenn erlaubt, kann die Installation über einen Rechtsklick auch als Administrator ausgeführt werden.



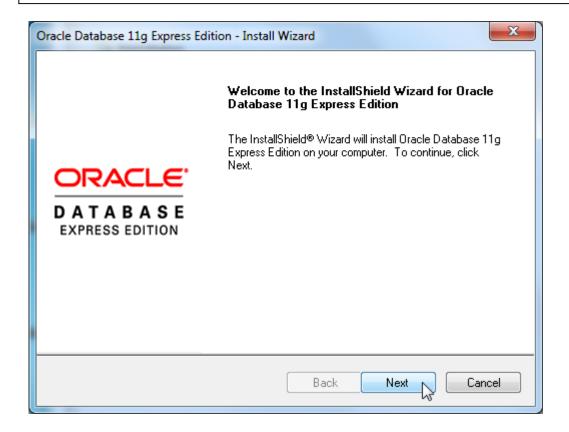
Auf Meldungen der Firewall oder Windows-Sicherheitsmeldungen wird hier nicht weiter eingegangen.



Die Installationsvorbereitungen dauern durchaus einige Zeit. In der Kopfzeile steht 11g obwohl eine Version 12 heruntergeladen wurde.



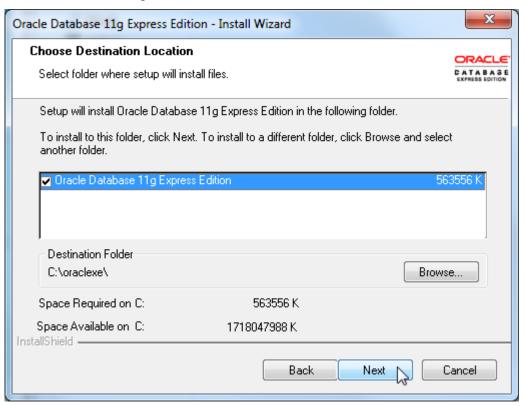
Dann kann die eigentliche Installation starten.



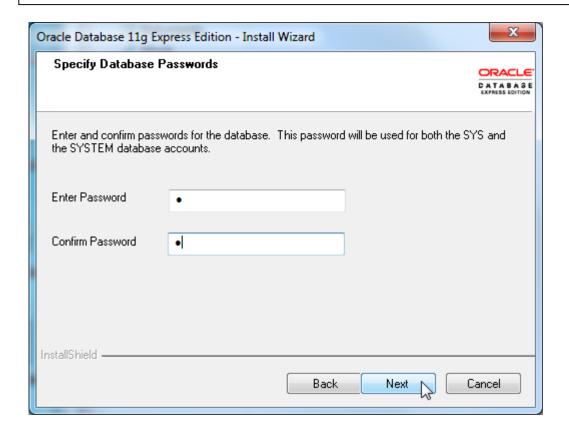
Die folgende Bestätigung muss zusätzlich angewählt werden.



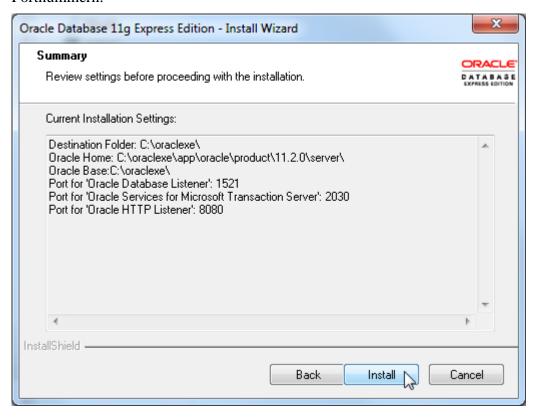
Der Installationsvorschlag wird übernommen, man kann den Pfad ändern, wobei es bei früheren Versionen Probleme gab, wenn sich ein Leerzeichen im Pfadnamen befand.



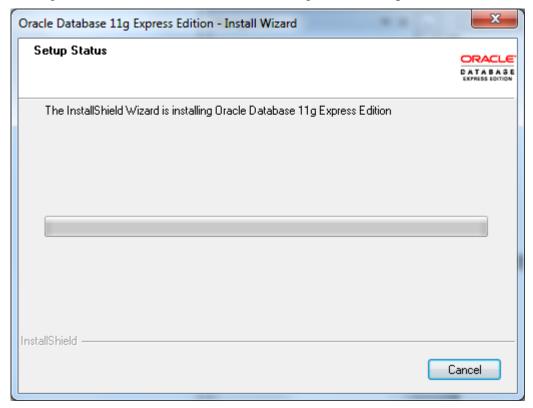
Dann muss das Passwort für die zentralen Administrationskonten mit den Nutzernamen SYS und SYSTEM festgelegt werden. Dieses Passwort ist später änderbar, sollte man es vergessen, ist die Wartung und Einrichtung weiterer Nutzer nicht mehr möglich. Ein Passwort wie x oder SYSTEM ist zum Experimentieren eventuell hilfreich, da sie hier schon in dieser Anleitung stehen.



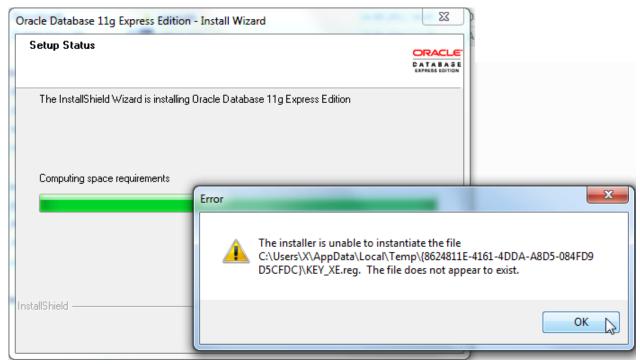
Danach kann die Installation begonnen werden. Man beachte die vergebenen, Oracle üblichen, Portnummern.



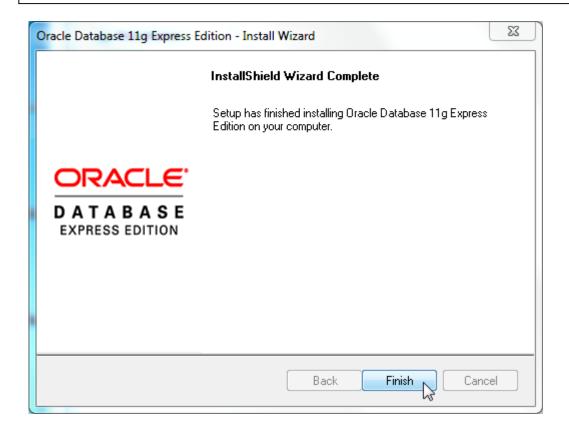
Die eigentliche Installation nimmt wieder einige Zeit in Anspruch.



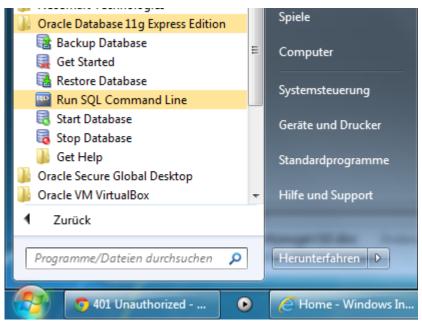
Einige Fehlermeldungen können anscheinend ignoriert werden (????).



Nach einiger Zeit und vielen Meldungen der Firewall ist die Installation dann erfolgreich abgeschlossen.



Unter den Programmen findet man dann einige Möglichkeiten zur Datenbanksteuerung. Sehr wichtig ist dabei, dass man die Datenbank starten und schließen kann. Nach der Installation ist die Datenbank automatisch gestartet.



1.5 Dienste von Oracle XE

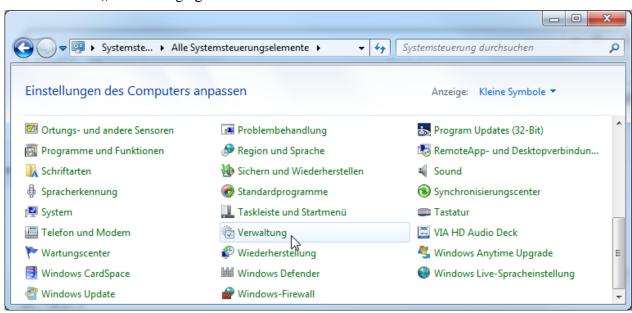
Ändert man nichts an den Einstellungen, wird die Datenbank mit jedem Rechnerstart ebenfalls gestartet. Dies Verhalten kann z. B. über die Dienst-Einstellungen geändert werden. Ein Weg diese zu erreichen ist über "Start -> Systemsteuerung".



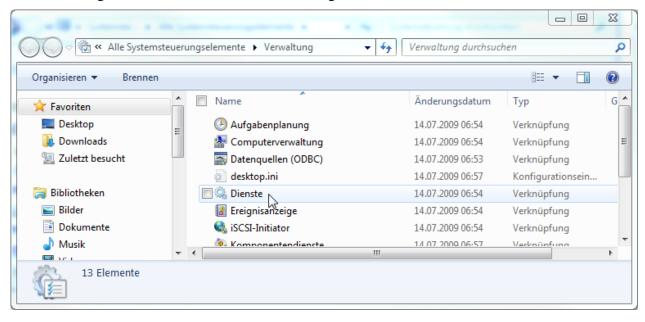
Es wird links oben, falls nicht bereits eingestellt, die Anzeige auf "kleine Symbole gesetzt".



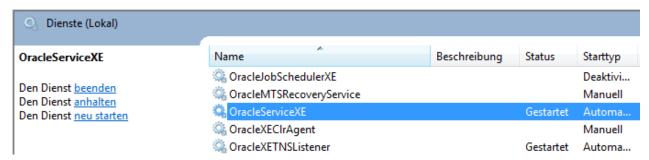
Dann wird auf "Verwaltung" geklickt.



Im sich öffnenden Fenster wird dann ein Doppelklick auf "Dienste" gemacht um zur Dienstverwaltung zu kommen. Man kann sich auch eine Verknüpfung mit "Dienste" auf die Oberfläche legen, wenn man die Datenbank häufiger ein- und ausschalten möchte.



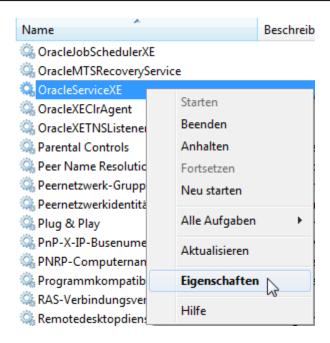
Die folgende Abbildung zeigt, dass zumindest zwei Oracle-Dienste automatisch gestartet werden.



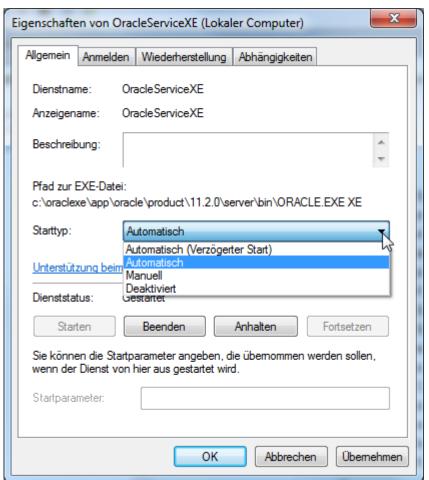
Zur Veränderung des Verhaltens wird ein Rechtsklick auf dem Dienst gemacht und "Eigenschaften ausgewählt.

OracleServiceXE

Den Dienst <u>beenden</u> Den Dienst <u>anhalten</u> Den Dienst <u>neu starten</u>



Die Einstellung des Startverhaltens erfolgt über "Starttyp".



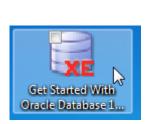
Bei der Einstellung "Automatisch" startet die Datenbank mit jedem Rechnerstart, was immer Zeit kostet und Hauptspeicher beansprucht. Sinnvoll ist die Einstellung "Manuell", da die Datenbank dann über "Start Database" gestartet werden kann. Bei "Deaktiviert" ist kein Start der Datenbank mehr möglich, bevor die Einstellung des Dienstes wieder geändert wird.

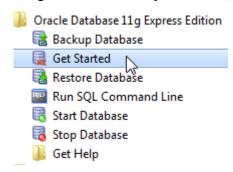


2 Einrichten eines Nutzers in Oracle XE

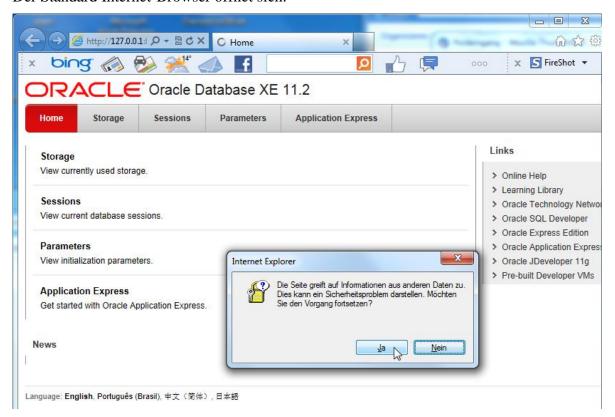
Bisher ist nur der Systemadministrator als Nutzer in Oracle XE eingetragen. Generell gilt, dass man zu Übungszwecken alle Schritte als Administrator durchführen kann, da er auch als normaler Nutzer tätig sein kann. Professioneller und zur Übung ist es aber sinnvoll, zumindest einen weiteren Nutzer auf seinem Rechner in der Datenbank einzurichten. Bei mehreren Nutzern können insbesondere die Übungen zur Transaktionssteuerung mit mehreren Datenbanknutzern zu Hause nachvollzogen werden, wobei hier kleine Unterschiede zur Version an der HS möglich sind. Bevor der SQL-Developer für die Administration genutzt wird, wird hier gezeigt, wie der zu Oracle XE gehörende Web-Client zur Einrichtung (und nicht gezeigt, zur weiteren Nutzung) eingesetzt werden kann.

Man findet nach der Installation auf der Oberfläche folgendes Icon, was jetzt mit einem Doppelklick genutzt wird. Im Programmordner entspricht dies "Get Startet".





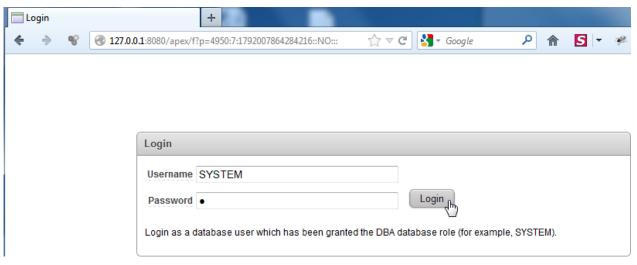
Der Standard Internet-Browser öffnet sich.



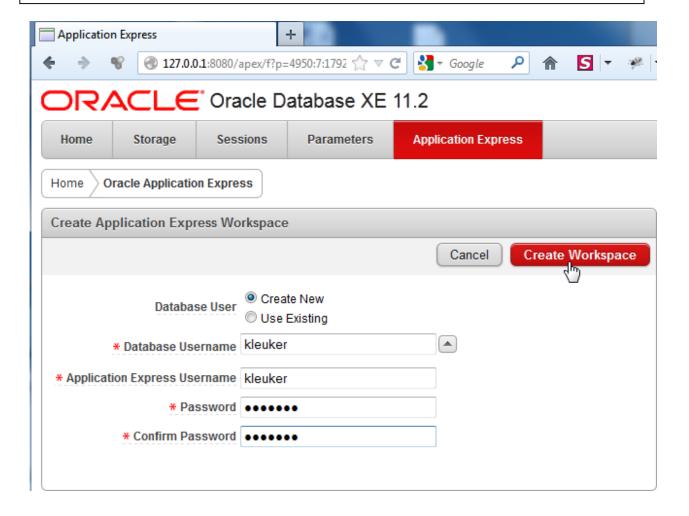
Zur Einrichtung eines neuen Nutzers wird auf "Application Express" geklickt.



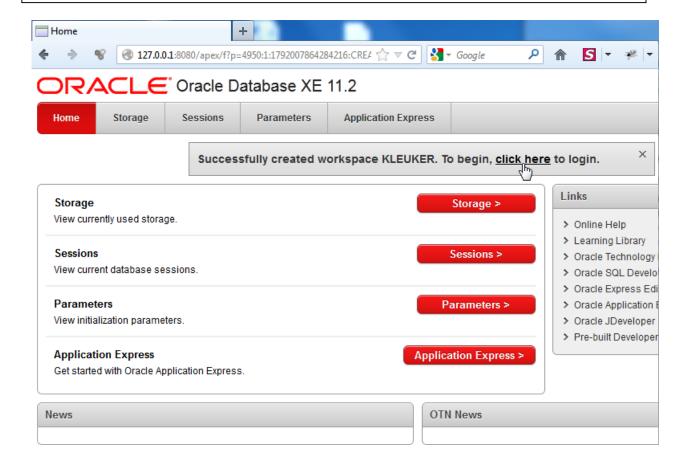
Man muss sich als Administrator SYSTEM beim System mit dem vorher selbst vergebenen Passwort anmelden. Dies ist leider nur möglich, wenn Cookies im Browser erlaubt sind.



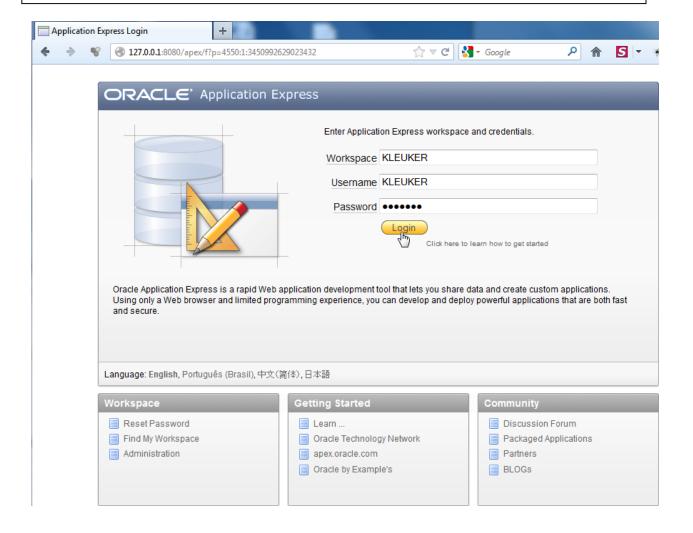
Nun kann direkt ein neuer Nutzer mit einem Workspace eingerichtet werden.



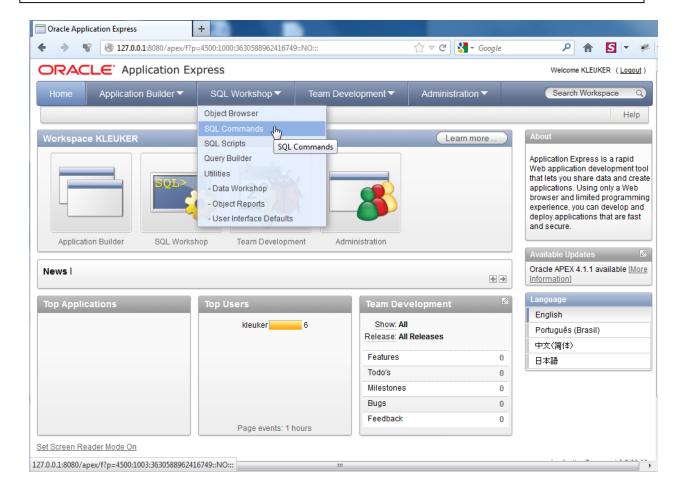
Man erhält eine Erfolgsmeldung und kann z.B. direkt versuchen, sich anzumelden.



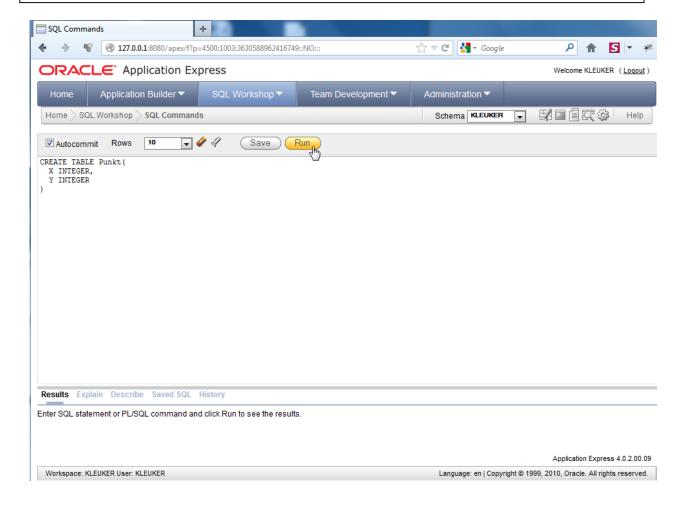
Man kommt zu einem neuen Anmeldeschirm und kann mit dem Passwort die Datenbank betreten, beim Nutzer spielt die Groß- und Kleinschreibung keine Rolle, anders beim Passwort.



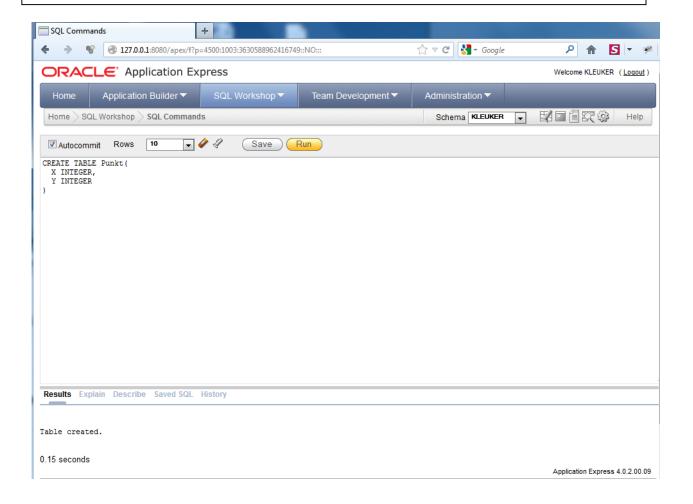
Hier stehen einige Möglichkeiten zur Verfügung, die aber bequemer mit dem im nächsten Kapitel vorgestellten SQL Developer bearbeitet werden können. Da diese aber immer installiert sein muss, kann diese Web-Lösung durchaus hilfreich sein. Als Beispiel kann über "SQL Workshop" eine Möglichkeit zur Eingabe von SQL-Befehlen geöffnet werden.



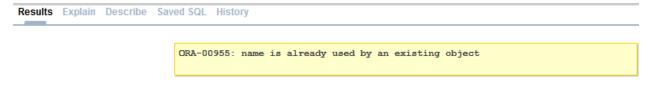
Befehle werden über "RUN" ausgeführt.



Im unteren Bereich erhält man dann eine Meldung, ob der Befehl erfolgreich war.

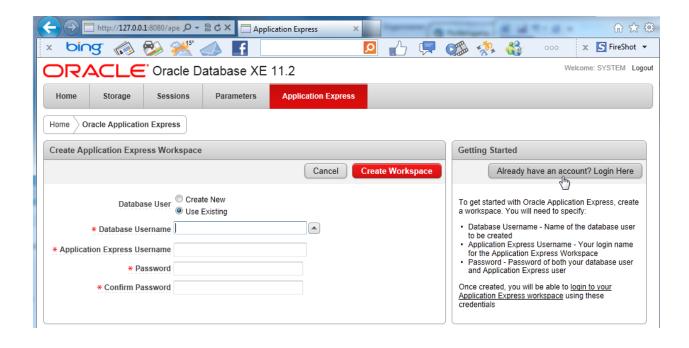


Drückt man nochmals auf RUN erhält man eine erwartete Fehlermeldung.



0.01 seconds

Möchte man als einfacher Datenbank-Nutzer den Web-Client nutzen, muss man sich trotzdem als Administrator SYSTEM zuerst einloggen, danach kann unter "Application Express" auf der rechten Seite "Login Here" angeklickt werden.



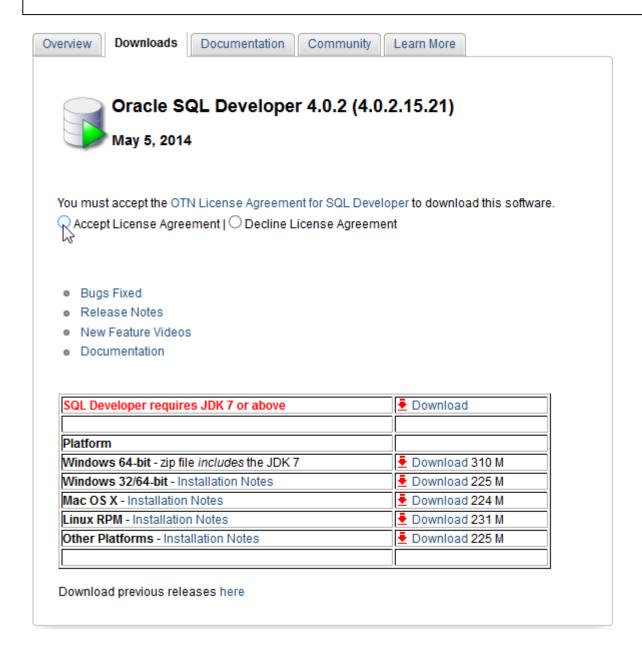
3 Installation des SQL-Developers

Die folgende Installation gilt nur für den eigenen Rechner, auf den Hochschulrechnern ist der SQL-Developer bereits installiert, beachten Sie die später folgenden Informationen zur Nutzung.

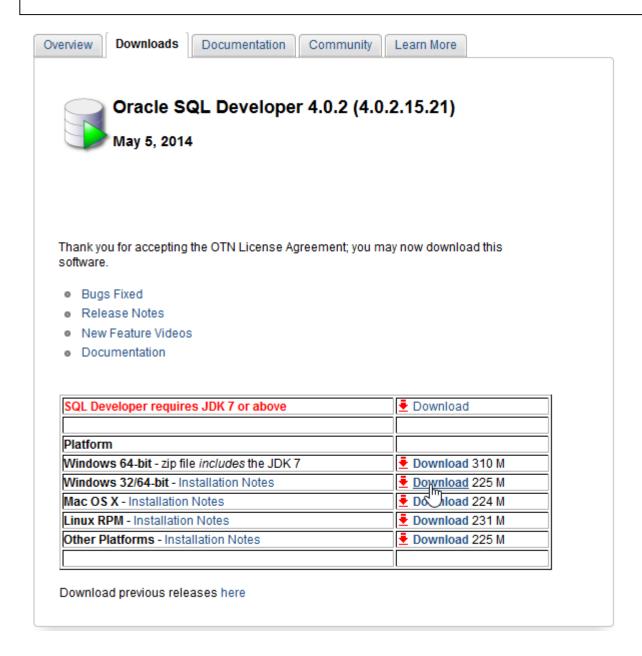
Generell benötigt der SQL Developer eine installierte Java-Version. Die hier betrachtete Version wünscht eigentlich Java 6, bisherige Analysen zeigten aber noch keine Probleme mit Java 7. Die Installation von Java ist nicht Bestandteil dieses Dokuments kann aber z. B. unter http://home.edvsz.hs-osnabrueck.de/skleuker/querschnittlich/NetbeansNutzung.pdf nachgelesen werden.

3.1 Download

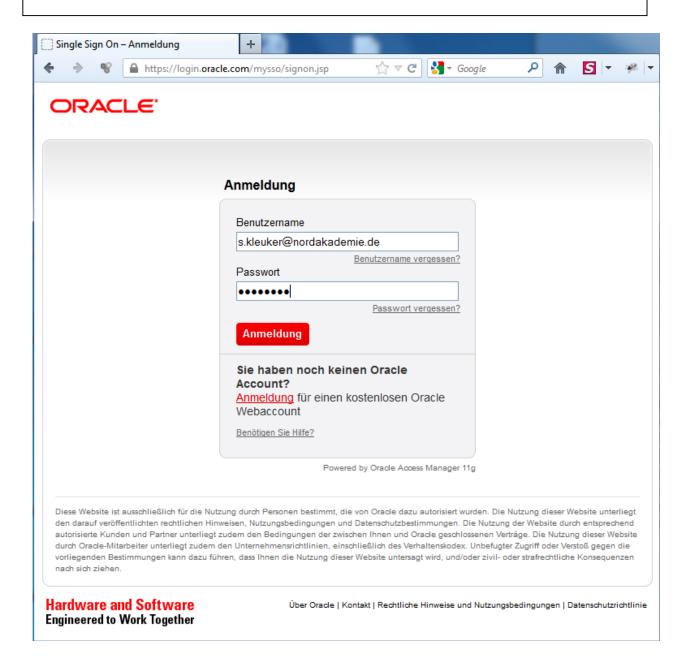
Ein Download ist von der Seite http://www.oracle.com/technetwork/developer-tools/sql-developer/downloads/index.html möglich, alternativ muss nach "SQL Developer" auf den Oracle-Seiten gesucht werden. Man beachte, dass man vor dem Download "Accept License Agreement" markiert.



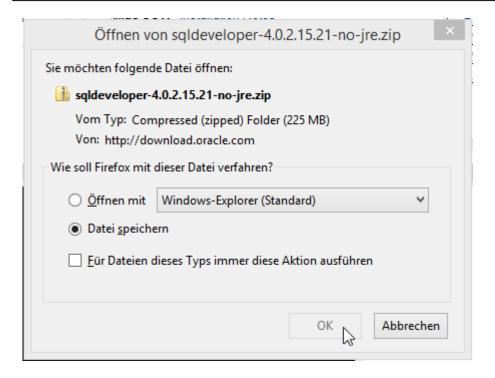
Weiterhin muss die zum eigenen Betriebssystem passende Variante ausgewählt und der Link angeklickt werden.



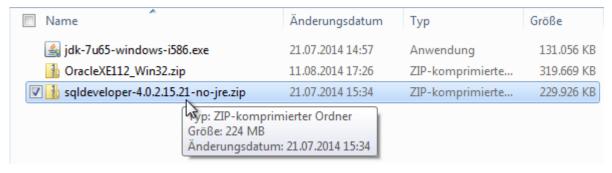
Man muss dann noch seine Anmeldungsdaten eingeben. Nach dem Klick auf "Anmeldung" startet der Download automatisch. Informationen zur Anmeldung findet man im Abschnitt "1.2 Einschub: Oracle-Account einrichten".



Der Download dauert etwas.

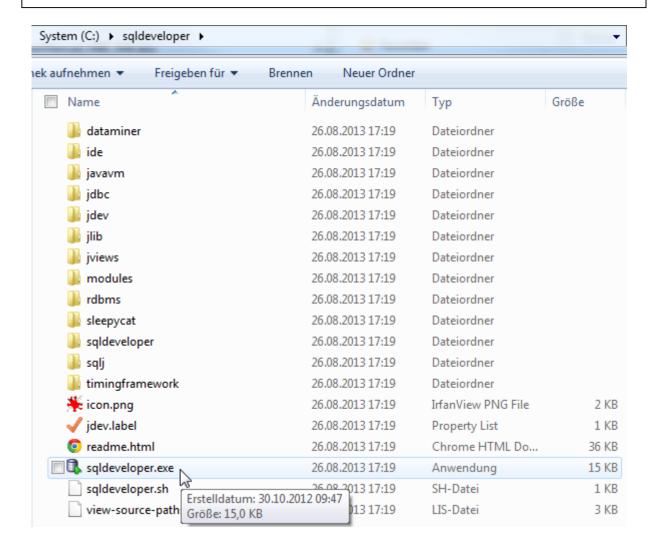


Man erhält folgende Datei.



3.2 Installation

Die Installation besteht ausschließlich aus dem Auspacken des zip-Files in einem beliebigen Installationsverzeichnis. Wird der SQL-Developer direkt unter C:\ ausgepackt, befindet sich das zu startende Programm im Verzeichnis C:\sqldeveloper und kann durch einen Doppelklick gestartet werden.



4 Erste Nutzung des SQL Developers

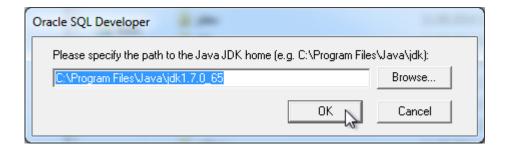
4.1 Erster Start

Zum Starten wird die in der vorherigen Abbildung markierte Datei sqldeveloper.exe ausgeführt. Auf etwaige Sicherheitsabfragen wird hier nicht eingegangen, da diese abhängig von den Systemeinstellungen und der installierten Firewall sind.



Abhängig von den installierten Programmen kann man beim ersten Start aufgefordert werden, den direkten Pfad zu java.exe (also dem zugehörigen bin-Verzeichnis) anzugeben, oder man kann man auch aufgefordert werden, den Pfad zum zu nutzenden JDK anzugeben (dem Verzeichnis oberhalb des Java-bin-Verzeichnisses). Wenn dies nicht passiert, hat der SQL-Developer automatisch den richtigen Pfad gefunden. Meldungen von Firewalls werden hier nicht angegeben, können aber mehrfach auftreten.

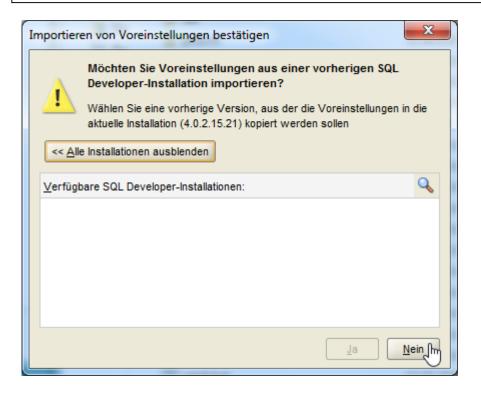




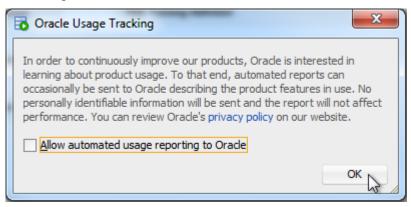
Dann findet der eigentliche Start statt.



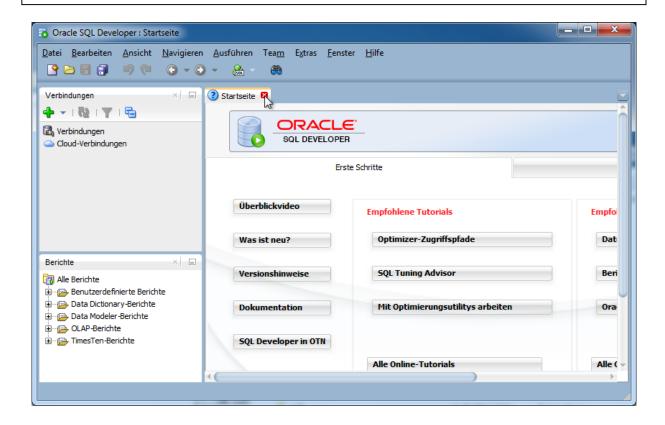
Hatte man bereits eine Variante des SQL-Developers installiert, kann man Voreinstellungen übernehmen. Dies muss man selbst entscheiden. Hier findet keine Übernahme statt, da der Dialog auch erscheint, wenn es keine weitere Installation auf dem Rechner gibt.



Ob man die dann gewünschte Usage-Tracking-Information liefern möchte, sollte man kritisch hinterfragen.



Die Startseite beinhaltet einige Informationsmöglichkeiten, die aber auch später nutzbar sind. Die Seite wird geschlossen.



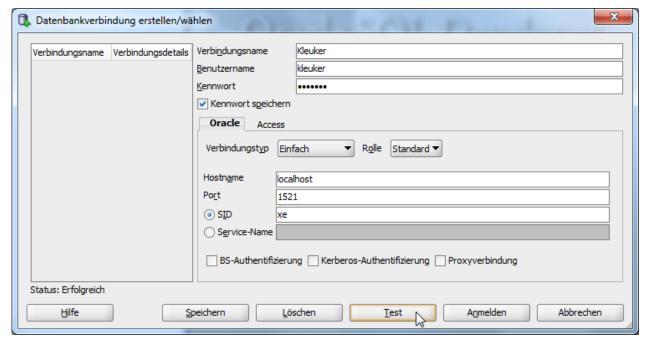
4.2 Verbindung einrichten

Nun soll für den vorher eingerichteten Nutzer eine Verbindung (Connection) zur Datenbank aufgebaut werden. Man erinnere sich, gegebenenfalls zuerst die Datenbank zu starten. Zum Verbindungsaufbau wird z. B. ein Rechtsklick auf "Verbindungen" auf der linken Seite gemacht und "Neue Verbindung …" gewählt.

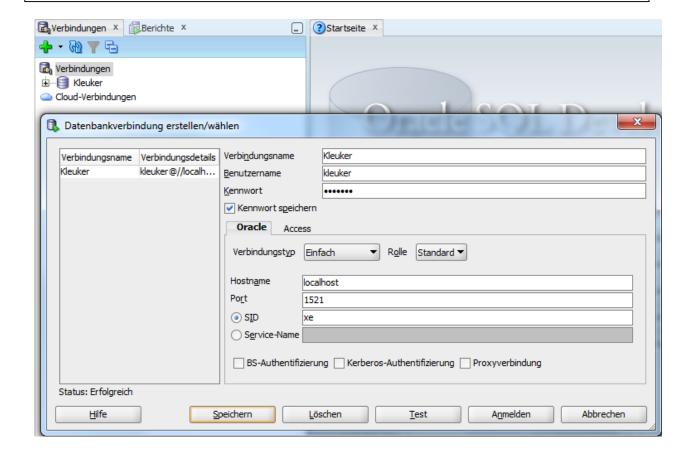


Der Name der Verbindung kann frei gewählt werden; weiterhin muss der Name des vorher eingerichteten Nutzers mit seinem Passwort in der Mitte oben eingegeben werden.

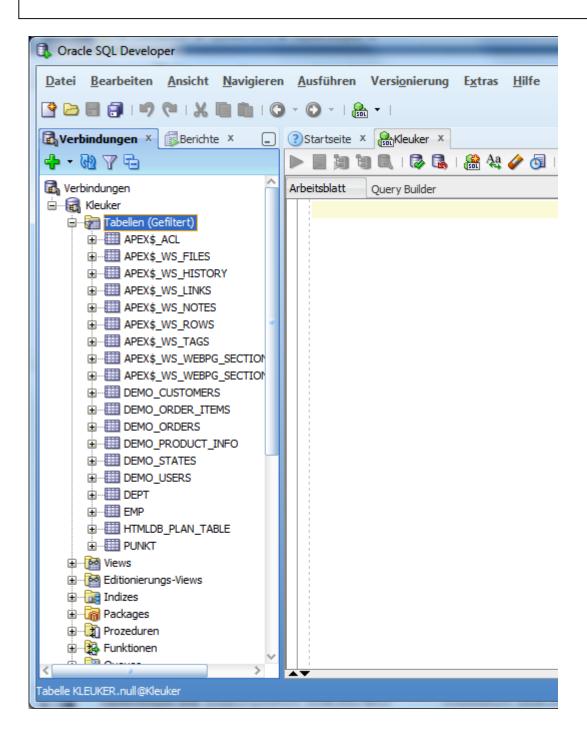
Weiter unten werden die Verbindungsdaten eingetragen. Der Hostname des eigenen Rechners ist "localhost" oder 127.0.0.1, es wird die Verbindung über den Standard-Port 1521 genutzt, der Name der Datenbank genauer der SID ist "xe". Mit dem Knopf "Test" kann überprüft werden, ob eine Verbindung zur Datenbank aufgebaut werden kann, die natürlich vorher gestartet sein muss.



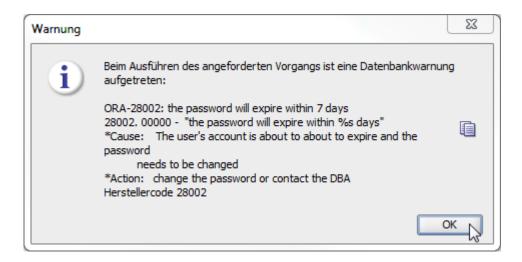
Durch Klicken auf "Speichern" wird die Datenbankverbindung eingerichtet, das Fenster kann dann geschlossen werden. Man sieht unter Verbindungen, dass die neue Verbindung eingerichtet wurde.



Nach dem erfolgreichen Verbindungsaufbau können dann alle Datenbankoperationen durchgeführt werden, die in der Veranstaltung gelehrt werden.

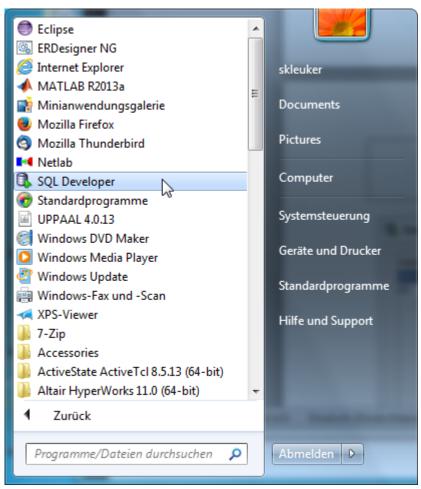


Eventuell wird man hingewiesen, dass man sein Passwort in einer Frist erneuern muss.



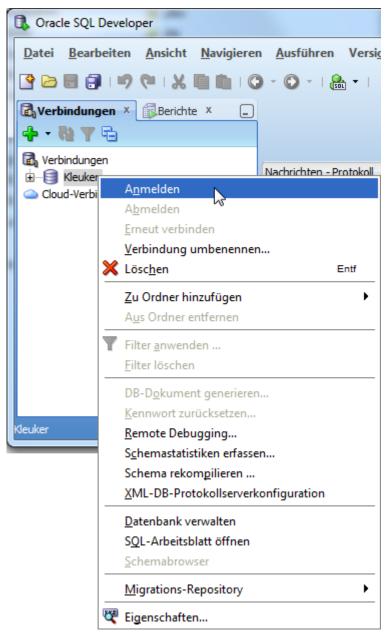
4.3 Normaler Start des SQL Developers

In der Hochschule ist der Developer unter Start->Alle Programme erreichbar.



Startet man den Developer, werden alle existierenden Verbindungen angezeigt. Durch einen Rechtsklick auf eine Verbindung, kann die Verbindung hergestellt werden. Abhängig davon,

ob man das Passwort hat speichern lassen, muss gegebenenfalls dieses Passwort noch eingegeben werden.



4.4 SQL Developer zusammen mit der HS-Datenbank

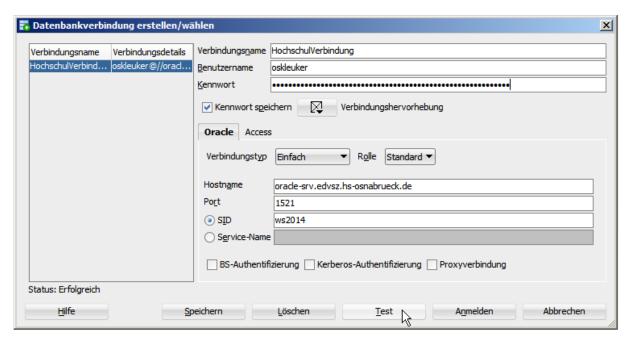
Der SQL Devekoper kann auch von Zuhause aus direkt mit der Datenbank der Hochschule genutzt werden. Die dazu benötigten Einträge stehen auch in der folgenden Abbildung.

Hostname: oracle-srv.edvsz.hs-osnabrueck.de

Port: 1521

SID: ws2014 Hinweis: Achtung, SID hängt vom Semester ab, in dem die

Veranstaltung stattfindet

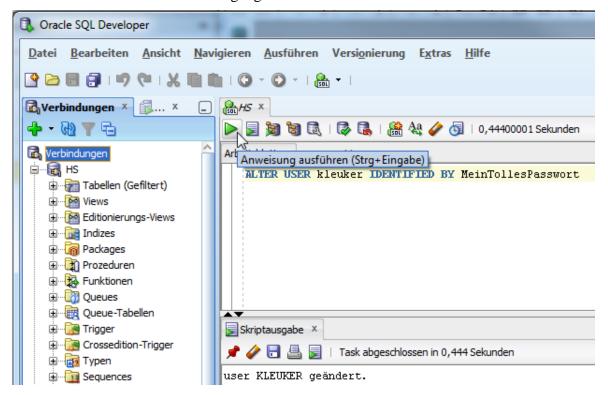


Der Test-Knopf kann zunächst zum Test der Verbindung genutzt werden.

Nach der ersten Anmeldung sollte das Passwort geändert werden. Dazu wird der folgende Befehl in das Bearbeitungsfenster gegeben. Aus Sicherheitgründen sollte nicht das hochschulweite Passwort genutzt, weiterhin muss auf Sonderzeichen verzichtet werden.

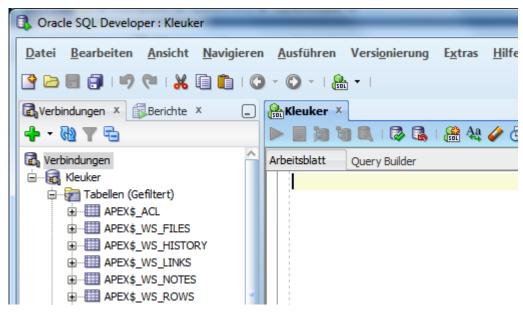
ALTER USER <benutzername> IDENTIFIED BY <neues passwort>

Danach wird, wie im Bild gezeigt, der grüne Pfeil links-oben zum Ausführen gedrückt. Unten erscheint eine Bestätigung. Man beachte, dass das Passwort unverschlüsselt angezeigt und nicht überprüft wird. Ein Tippfehler hier hat drastische Konsequenzen, ggfls. kann der DB-Adminstrator einen verlorenen Zugang wieder herstellen.

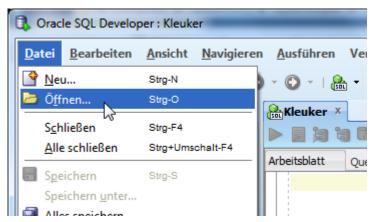


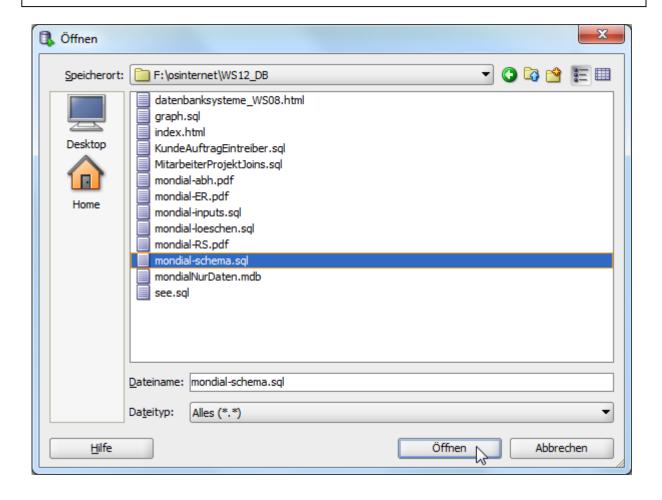
5 Einspielen von SQL-Skripten

Das typische Arbeitsfenster hat die folgende Form. Auf der linken Seite kann man zu den eigenen Datenbankinhalten manövrieren. Zentral zur Bearbeitung ist das Fenster rechts oben "Enter SQL Statement", da hier SQL-Befehle eingegeben werden können. Das Ergebnis der Aktionen ist im unteren rechten Fenster sichtbar, wobei es über die Reiter verschiedene Darstellungsmöglichkeiten gibt.



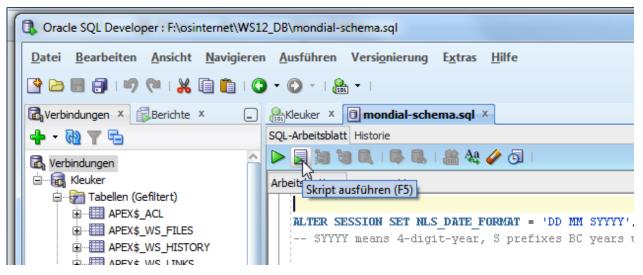
Möchte man eine Datei mit SQL-Befehlen laden, so erfolgt dies über das File-Menü, mit dem man zur SQL-Datei manövriert.



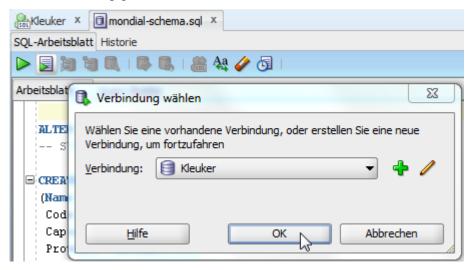


Die Datei wird in einem eigenen Fenster angezeigt, wobei die Befehle noch nicht ausgeführt werden.

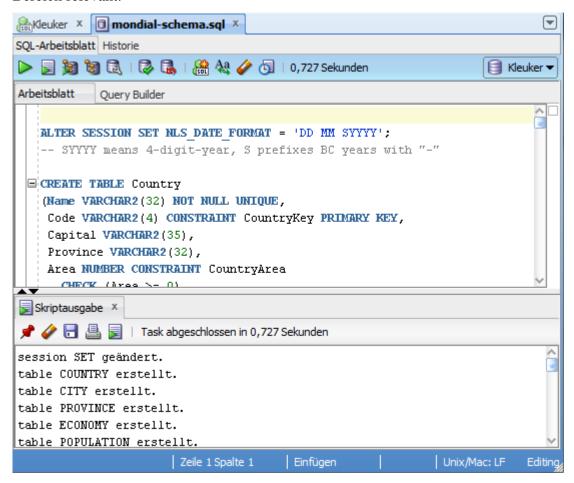
Generell werden Befehlssequenzen, also Mengen von Befehlen als Skript im Editor entweder über den grünen Pfeil auf einem Dokument oben oder über F5 ausgeführt. Einzelne Befehle werden über den grünen Pfeil ganz links , bzw. Strg+Return ausgeführt. Dabei wird immer der Befehl genutzt, auf dem der Cursor steht.



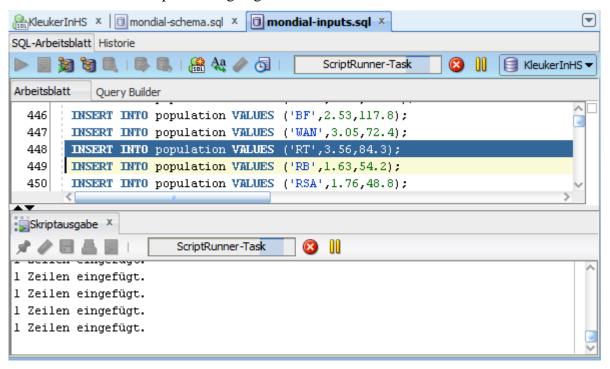
Beim Ausführen muss noch eine Verbindung ausgewählt werden. Hat man ein SQL-Fenster für eine Verbindung geöffnet, entfällt dieser Schritt.



Für das Ergebnis ist der sich spätestens jetzt öffnende Reiter "Skriptausgabe" im unteren Bereich relevant.



Bei Skripten, die länger Laufen, wird ein Laufbalken "ScriptRunner-Task" eingeblendet. Weiterhin kann die Skript-Nutzung abgebrochen und unterbrochen werden.



Am Ende wird immer die Laufzeit mit ausgegeben.

```
SQL-Arbeitsblatt Historie
🕟 🗾 📓 👸 👼 | 🔯 👢 | 🕍 🗛 🥢 👩 | 764,51501465 Sekunden

    KleukerInHS ▼

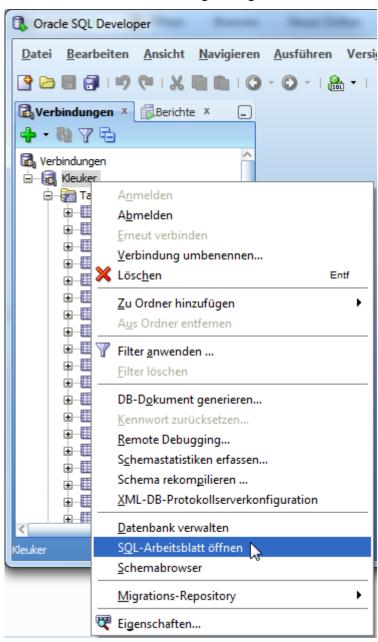
Arbeitsblatt
            Query Builder
    1
       (ALTER SESSION SET NLS DATE FORMAT = 'YYYY-MM-DD';
        INSERT INTO country VALUES ('Albania', 'AL', 'Tirane', 'Albania', 28750, 3249
        INSERT INTO country VALUES ('Greece', 'GR', 'Athens', 'Attiki', 131940, 10538
        INSERT INTO country VALUES ('Macedonia', 'MK', 'Skopje', 'Macedonia', 25333,
Skriptausgabe X
📌 🧽 🔡 📕 | Task abgeschlossen in 764,515 Sekunden
l Zeilen eingefügt.
l Zeilen eingefügt.
l Zeilen eingefügt.
festgeschrieben.
```

6 Erzeugen und Bearbeiten von Tabellen

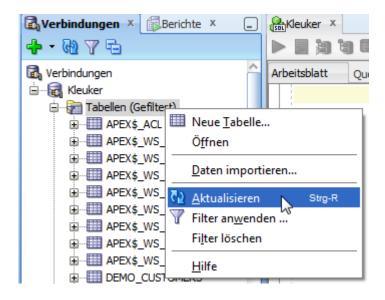
In der Veranstaltung wird gefordert, dass Tabellen direkt über SQL-Befehle definiert werden, über die direkte Nutzung des SQL-Developers zur alternativen Erstellung kann man sich selbst zusätzlich informieren.

6.1 Arbeitsblatt erstellen

Falls für eine genutzte Verbindung gerade kein "Arbeitsblatt" offen ist wird dies über ein Rechtsklick auf der Verbindung erzeugt.



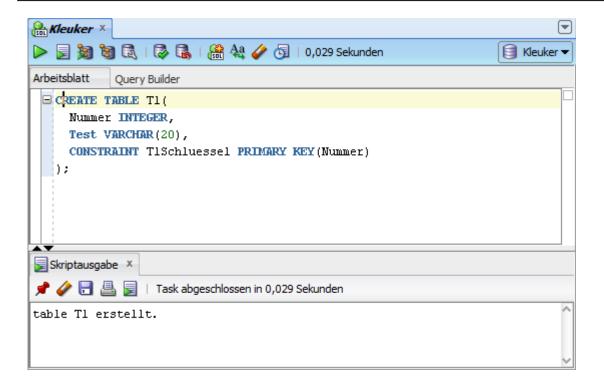
Danach kann man sich über die existierenden Tabellen links informieren. Dazu ist gegebenenfalls ein "Aktualisieren"-Befehl notwendig.



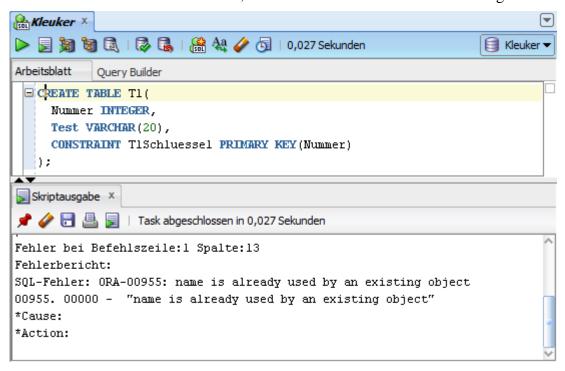
6.2 SQL-Befehl eingeben und ausführen

Der SQL-Befehl zur Tabellenerzeugung wird im Arbeitsblatt eingegeben und kann dann mit dem grünen Pfeil ausgeführt werden.

Eingabebefehle werden im Arbeitsblatt eingegeben. Man beachte, dass einzelne Befehle mit Strg+Return (Anweisung ausführen) und Gruppen von Befehlen mit F5 (Skript ausführen) ausgeführt werden. Fehlermeldungen stehen unten in der Skriptausgabe.



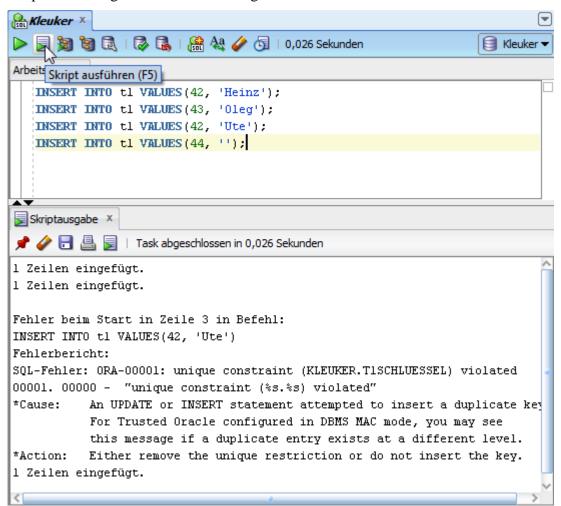
Führt man den Befehl nochmals aus, erhält man die erwartete Fehlermeldung.



Die vorherige Fehlermeldung deutet bereits an, dass es sinnvoll ist, Zeilennummern einzublenden. Ein Weg dazu ist ein Rechtsklick auf dem linken weißen Rand des Editor-Feldes.

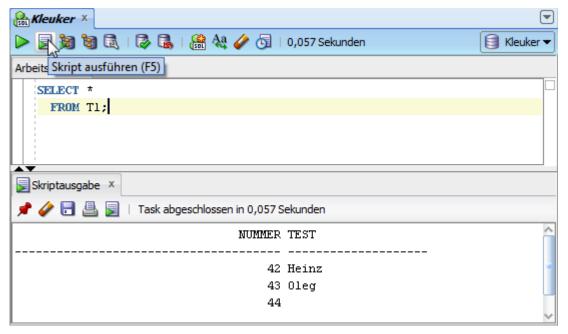


Etwaige Fehlermeldungen werden in der "Skriptausgabe" beschrieben, dabei wird das Skript nicht abgebrochen, die nachfolgenden Befehle werden ausgeführt. Man muss also den gesamten Output nach möglichen Fehlermeldungen durchsuchen.



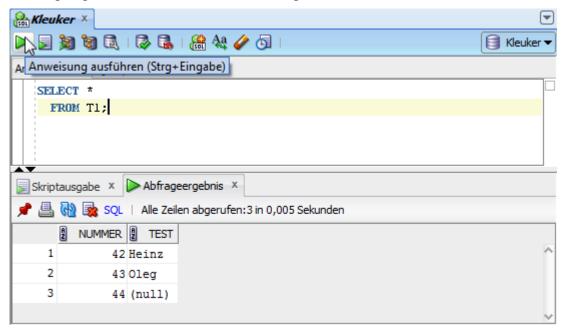
Um einzelne Befehle aus einem Skript auszuführen, wird der Cursor in der Zeile des Befehls platziert und der grüne Pfeil gedrückt.

SQL-Anfragen werden im oberen Fenster eingegeben. Das Ergebnis wird wieder unten ausgegeben.

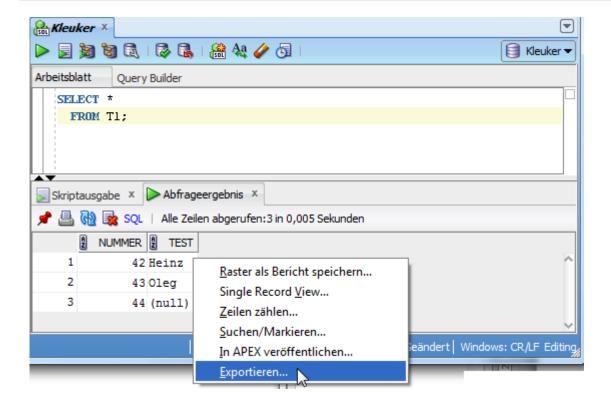


6.3 Ausführung einzelner SQL-Anfragen

Werden nur einfache Anfragen ausgeführt, erhält man mit "Anweisung ausführen" unter "Abfrageergebnis" eine schönere Darstellung.



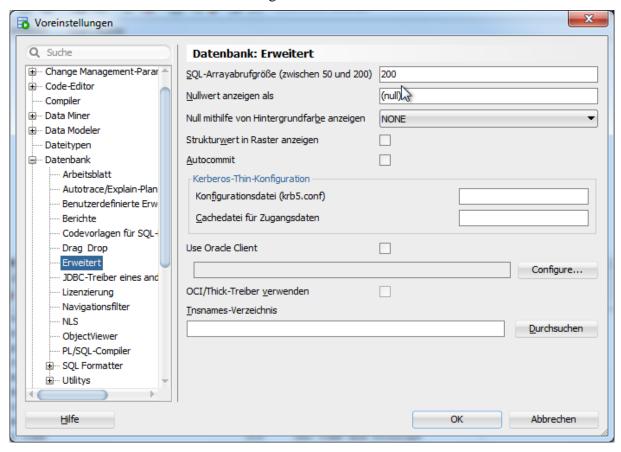
Durch einen Rechtsklick in der Tabelle ergeben sich dabei weitere Möglichkeiten, das Ergebnis zu exportieren.



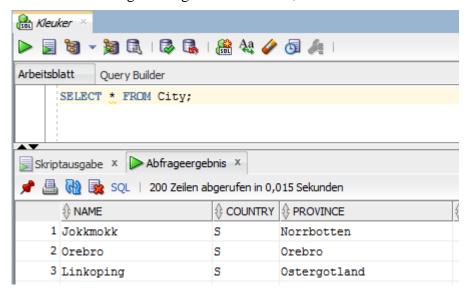
Mit den Standard-Einstellungen zeigt der Developer nur 50 Zeilen des Ergebnisses an, alternativ kann man auch "Skript ausführen" für nur einen Befehl nutzen und erhält immer alle Ergebnisse ausgegeben. Die maximale Grenze der angezeigten Werte ist auf 200 Zeilen änderbar. Möchte man wissen, wieviele Zeilen es wirklich gibt, kann der COUNT-Befehl aus SQL helfen. Um die maximale Grenze einzustellen, wählt man unter "Extras" dann "Voreinstellungen".



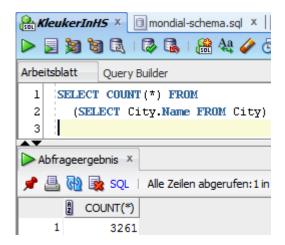
Hier wird links der der Punkt "Datenbank" aufgeklappt und "Erweitert" angeklickt. Rechtsoben kann dann die maximale Größe eingestellt werden.



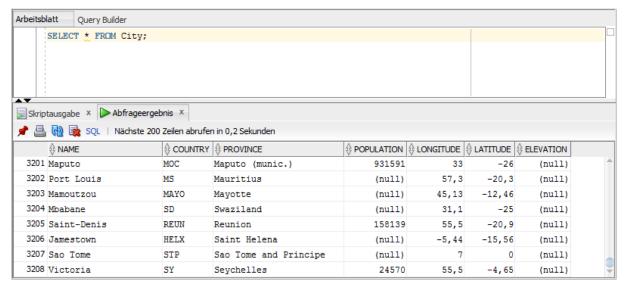
Liefert eine Anfrage dann genau 200 Zeilen, sollte man die Größe des Ergebnisses überpüfen.



Man kann die Ergebnistabelle z. B. in eine neue Anfrage einbetten.

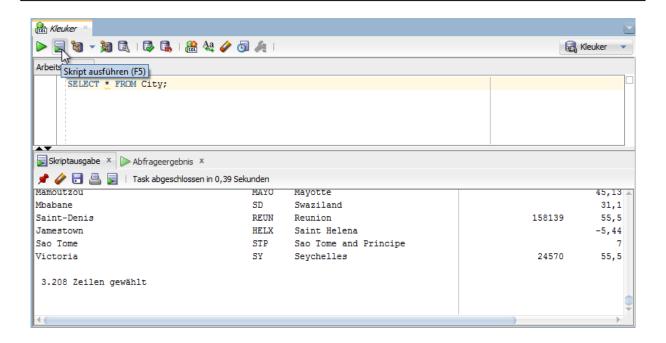


Abhängig von der SQL Developer-Version, kann man weitere Ergebnisse auch sehen, wenn man den Scrollbalken der Tabelle ganz nach unten zieht. Man sieht dann, dass die weiteren Ergebnisse nachgeladen werden. Bei älteren Versionen, bzw. Windows XP, sieht man zwar, dass der Scrollbalken nach oben springt, aber es werden keine Zeilen nachgeladen.



Noch eine weitere Alternative besteht unabhängig von der Version, wenn man in die Ergebnistabelle klickt und dort zusammen die "Tasten Strg+Bild nach unten" drückt.

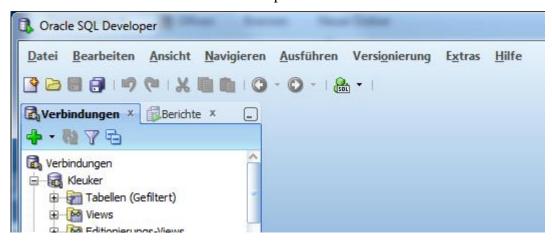
Möchte man sofort alle Zeilen sehen, sollte man den Befehl als Skript ausführen.



7 Tipps zum SQL-Developer

7.1 Arbeitsblätter

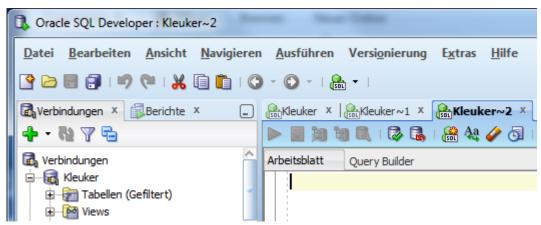
Falls man aus Versehen das Bearbeitungsfenster geschlossen hat, bekommt man folgende Ansicht mit einem leeren rechten Arbeitsplatz:



Um wieder arbeiten zu können, gibt es z. B. in der zweiten Reihe ein SQL-Icon, mit dem neue Arbeitsblätter (Worksheets) aufgemacht werden können.



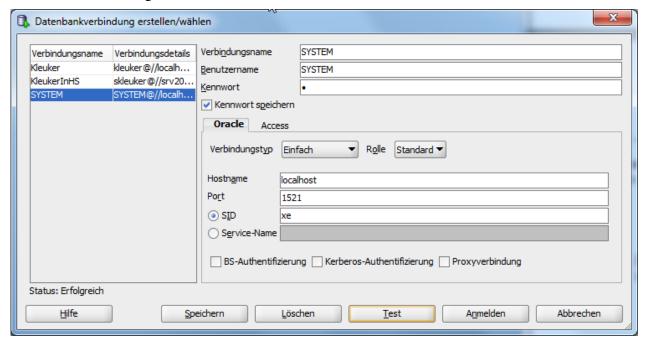
Interessant ist dabei weiterhin, dass man so auch mehrere Worksheets parallel nutzen kann.



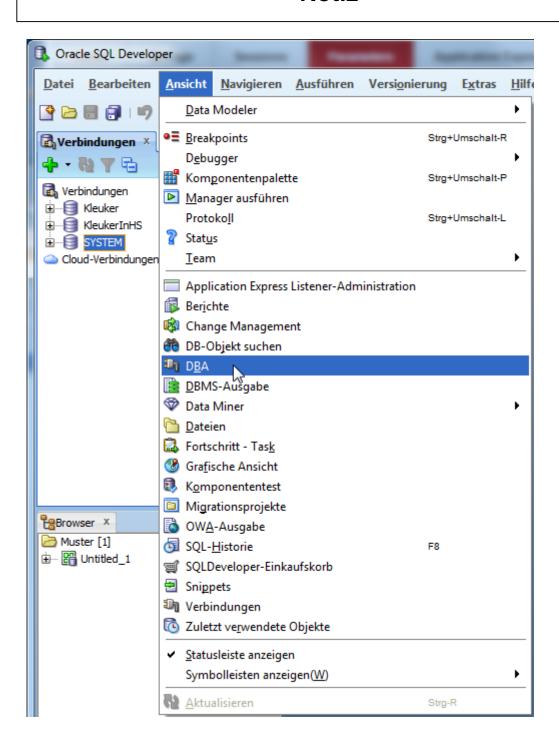
Falls man mit mehreren Datenbank-Verbindungen experimentieren möchte, muss man den SQL-Developer mehrfach starten und kann sich dann mehrfach mit dem gleichen Nutzer anmelden.

7.2 Datenbank administrieren

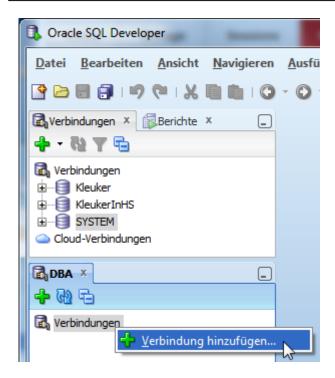
Mit dem SQL Developer kann man seine XE-Datenbank auch administrieren und z. B. neue Nutzer einrichten. Die Möglichkeiten werden hier kurz andiskutiert. Zunächst ist eine Datenbankverbindung als Administrator SYSTEM einzurichten.



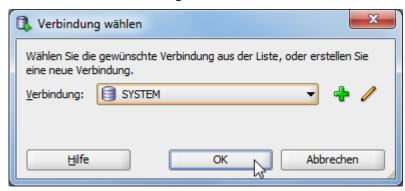
Im SQL Developer wird dann unter "Ansicht" der Punkt "DBA" ausgewählt.



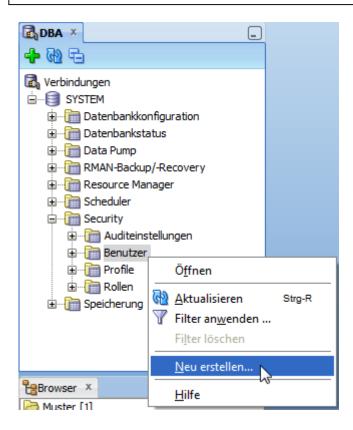
Standardmäßig wird auf der linken Seite ein kleines Fenster eingeblendet, bei dem man mit einem Rechtsklick eine Verbindung hinzufügt.



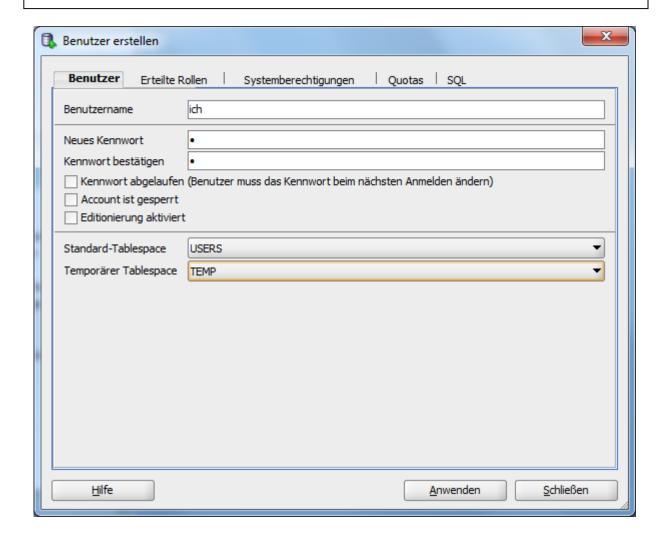
Es wird eine vorhandene, genauer die Administrator-Verbindung, ausgewählt.



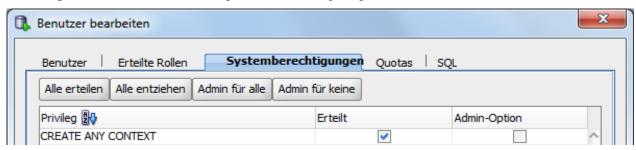
Dann kann man die ausgewählte Verbindung aufklappen und z. B. einen Nutzer unter "Security" hinzufügen.



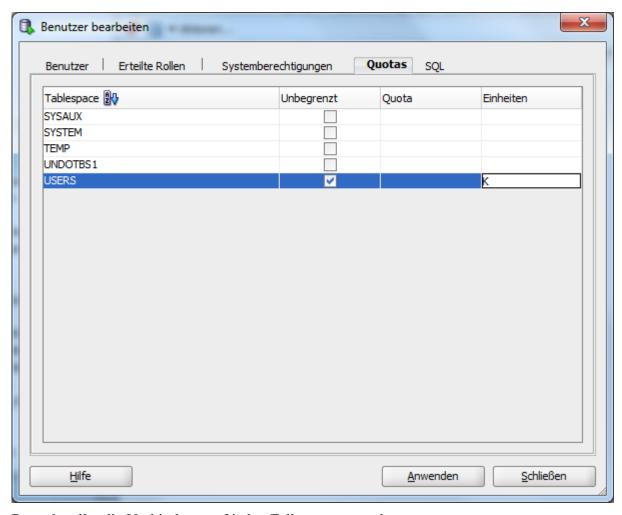
Danach müssen die einzelnen Reiter durchgegangen werden.



Zumindest müssen einige Systemberechtigungen eingestellt werden, damit sich der Nutzer überhaupt anmelden kann. Die folgende Abbildung zeigt eine solche Auswahl.



		CREATE CLUSTER	✓	^
		CREATE CUBE		
	Ш	CREATE CUBE BUILD PROCESS		
	Ш	CREATE CUBE DIMENSION		
		CREATE DATABASE LINK		
		CREATE DIMENSION	✓	
		CREATE EVALUATION CONTEXT		
		CREATE EXTERNAL JOB		
		CREATE INDEXTYPE	✓	
		CREATE JOB	✓	
		CREATE LIBRARY		
	Ш	CREATE MATERIALIZED VIEW	✓	
		CREATE MEASURE FOLDER		
		CREATE MINING MODEL		
	Ш	CREATE OPERATOR	✓	
		CREATE PROCEDURE	✓	
		CREATE SEQUENCE	✓	
		CREATE SESSION	V	
		CREATE SYNONYM	▽	
		CREATE TABLE	~	
		CREATE TABLESPACE		
		CREATE TRIGGER	✓	
		CREATE TYPE	✓	
		CREATE USER		
		CREATE VIEW	✓	
		DEBUG ANY PROCEDURE	✓	
		DEBUG CONNECT SESSION	✓	



Danach sollte die Verbindung auf jeden Fall getestet werden.

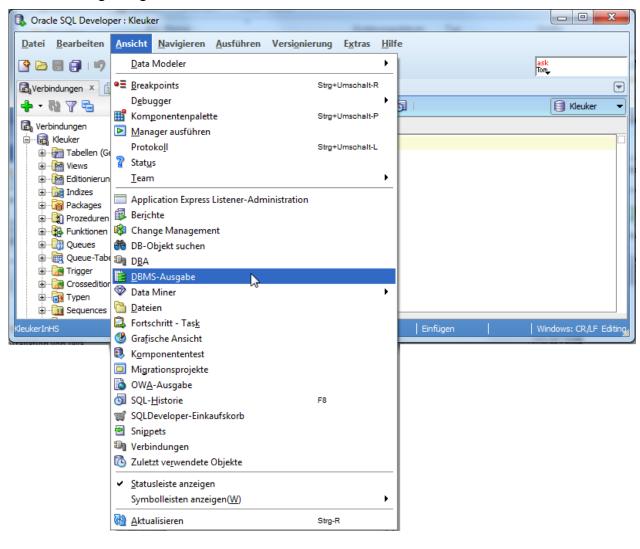
8 Erstellung von PL/SQL-Funktionen und Prozeduren

Für diese Notiz wird davon ausgegangen, dass Sie die Einführung in die Nutzung des SQL-Developers gelesen haben. Es wird hier eine mögliche Nutzungsweise gezeigt, es sind einige andere Ansätze möglich.

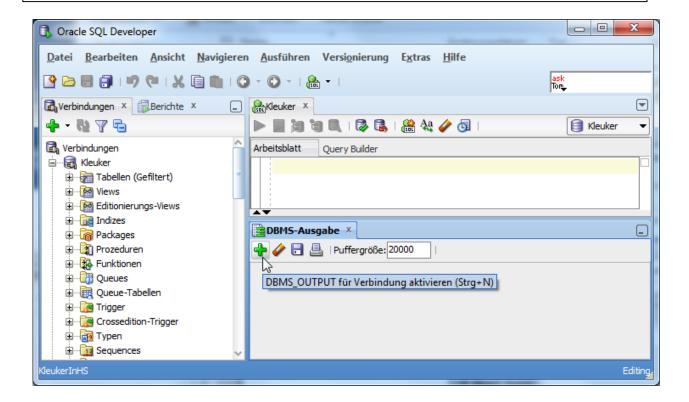
Als Beispiel wird eine einfache Prozedur entwickelt, die eine Begrüßung auf der Konsole ausgeben soll. Da die Entwicklung von Prozeduren und Funktionen identisch verläuft, wird hierzu kein zusätzliches Beispiel angegeben.

8.1 Ausgaben sichtbar werden lassen

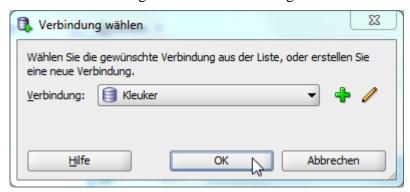
Um generell Programmausgaben sehen zu können, **muss** die Ausgabe erlaubt werden. Ein sicherer Weg ist es, das zugehörige Ausgabefenster gleich einzublenden. Dazu wird "Ansicht - > DBMS-Ausgabe" gewählt.



Für die garantierte Ausgabe muss links oben "DBMS_OUTPUT aktivieren" angeklickt werden.

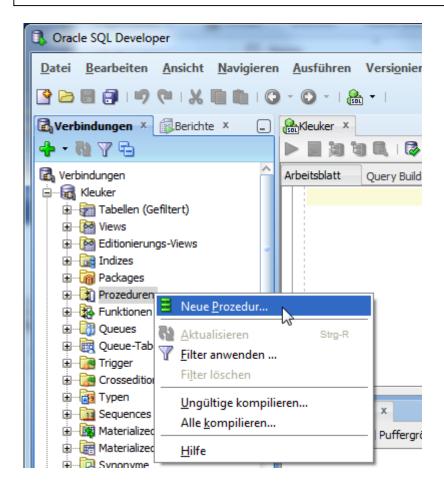


Man muss dann die gewünschte Verbindung auswählen.

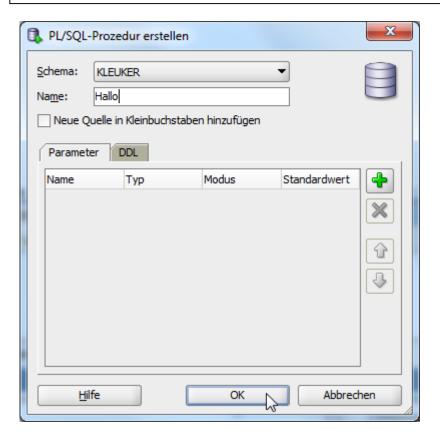


8.2 Prozedur erstellen

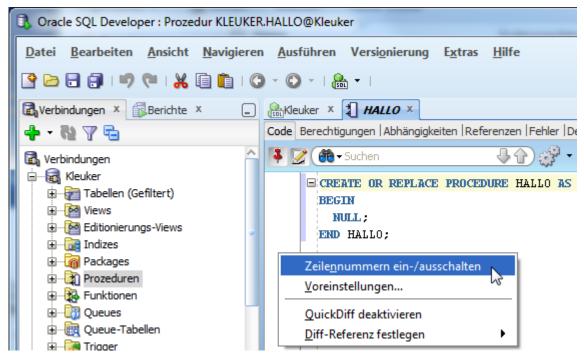
Zum Erstellen eine Prozedur wird ein Rechtsklick auf "Prozeduren" in der Auswahl am linken Rand ausgeführt und "Neue Prozedur..." ausgewählt.



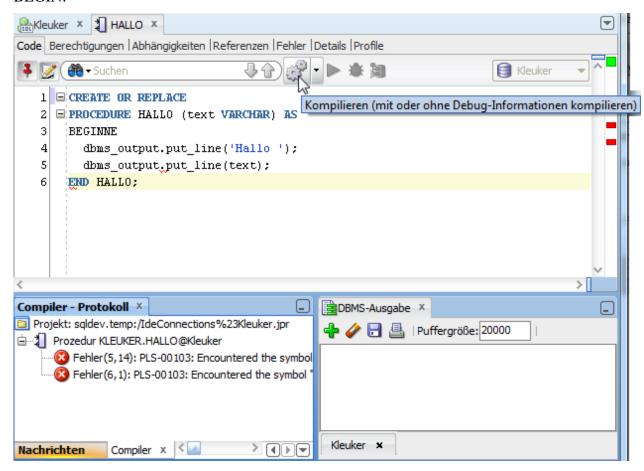
In dem sich dann öffnenden Dialog-Fenster können einige Einstellungen gemacht werden. Hier wird vorgeschlagen, nur den Prozedur-Namen einzutragen und dann die Erzeugung mit "OK" abzuschließen.



Es wird ein Editor-Fenster geöffnet, in dem die eigentliche Programmierung stattfindet. Bevor man anfängt, sollte man mit einem Rechts-Klick auf den ganz linken Rand des Editor-Fensters die Zeilennummerierung einschalten.



Danach findet die eigentliche Programmierung im Editor statt. In der Kopfzeile des Editors befindet sich ein Knopf "Kompilieren" um die Kompilierung der Prozedur zu starten, die Kompilierung zum Debuggen wird weiter unten behandelt. Etwaige Fehlermeldungen werden unterhalb des Editors eingeblendet. Man beachte im folgenden Fenster steht BEGINNE statt BEGIN.



Die durchaus häufiger wenig aussagekräftigen Fehlermeldungen sollten von oben nach unten abgearbeitet werden. Dabei ist die Angabe der Fehlerposition nur als der Versuch eines Hinweises zu deuten.

Nach einer erfolgreichen Korrektur steht die Prozedur zur Nutzung zur Verfügung, das "SYS." Kann weggelassen werden.

```
CREATE OR REPLACE

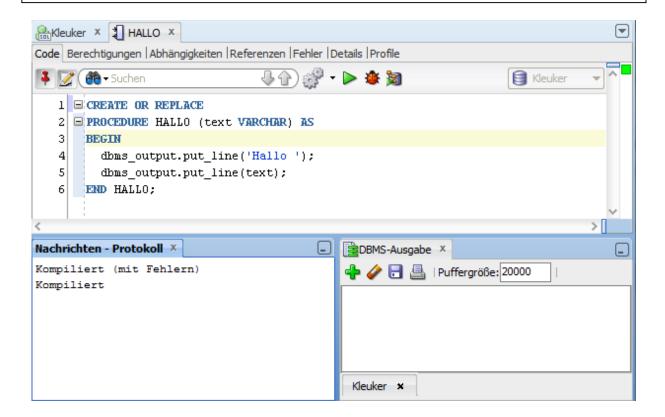
PROCEDURE HALLO (text VARCHAR) AS

BEGIN

SYS.DBMS_OUTPUT.PUT_LINE('Hallo ');

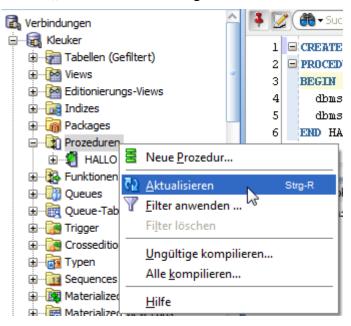
SYS.DBMS_OUTPUT.PUT_LINE(text);

END HALLO;
```

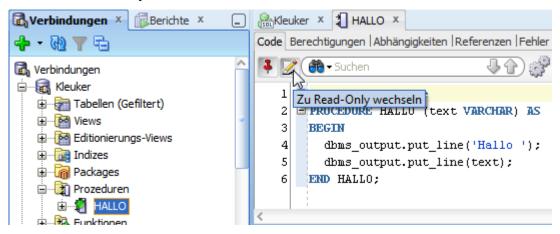


8.3 Prozedur ausführen

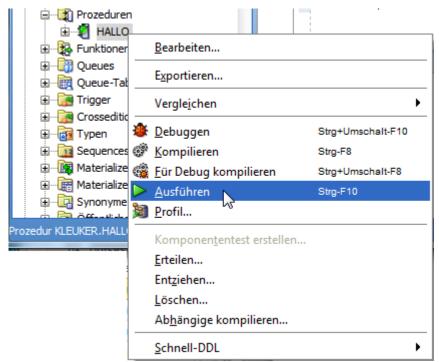
Wird auf der rechten Seite der Punkt "Prozeduren" aufgeklappt, kann man hier alle vorhandenen Prozeduren erkennen. Eventuell muss mit einem Rechtsklick auf "Prozeduren" erst ein "Aktualisieren" durchgeführt werden.



Durch einen einfachen Klick auf den Prozedur-Namen wird diese auf der rechten Seite angezeigt und der Editor geöffnet. Es gibt auch die Möglichkeit vor Veränderungen im Editor durch den Read-Only-Modus zu schützen.



Weiterhin kann die Prozedur mit einem Rechtsklick auch ausgeführt werden.



Alternativ dazu wird auf den grünen Pfeil geklickt.

```
Code Berechtigungen | Abhängigkeiten | Referenzen | Fehler | Details | Profile

| Code | Berechtigungen | Abhängigkeiten | Referenzen | Fehler | Details | Profile

| Code | Berechtigungen | Abhängigkeiten | Referenzen | Fehler | Details | Profile

| Code | Berechtigungen | Abhängigkeiten | Referenzen | Fehler | Details | Profile

| Code | Berechtigungen | Abhängigkeiten | Referenzen | Fehler | Details | Profile

| Code | Berechtigungen | Abhängigkeiten | Referenzen | Fehler | Details | Profile

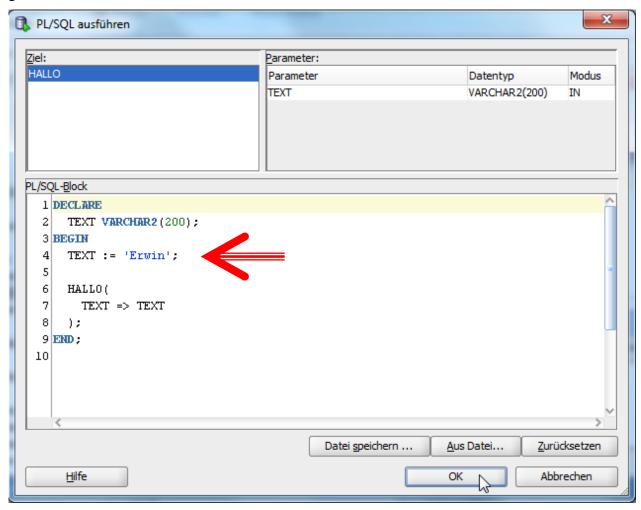
| Code | Berechtigungen | Abhängigkeiten | Referenzen | Fehler | Details | Profile

| Code | Berechtigungen | Abhängigkeiten | Referenzen | Fehler | Details | Profile

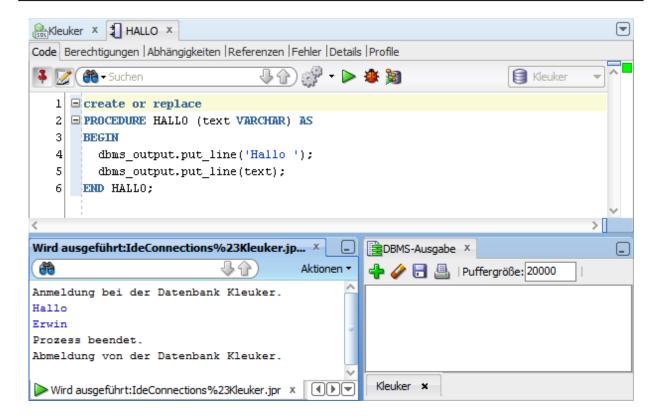
| Code | Berechtigungen | Abhängigkeiten | Referenzen | Fehler | Details | Profile

| Code | Code
```

Nach dem Start der Prozedur geht ein Fenster auf, in das man Parameter eintragen kann. Dazu wird der angezeigte PL/SQL-Block genutzt, der im nächsten Schritt ausgeführt werden soll. Im Beispiel wurde der Wert für die Variable text von Hand an der mit dem Pfeil markierten Stelle geändert.

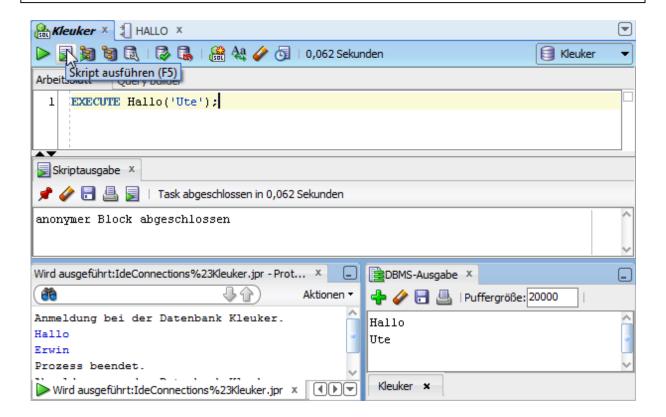


Danach wird das Programm mit "OK" ausgeführt.



8.4 Ausführung mit EXECUTE

Eine alternative Ausführung findet im eigentlichen Bearbeitungsfenster statt. Dieser Ansatz hat den Vorteil, dass er unabhängiger vom eigentlichen Werkzeug ist. Dazu wird der Befehl im Fenster eingegeben und muss dann mit "Skript ausführen" (F5) ausgeführt werden. Das Ergebnis kann wie folgt aussehen. Das wichtige Ausgabefenster sieht man rechts unten.



8.5 Nutzung des Debuggers

Neben der direkten Kompilierung besteht auch die Möglichkeit, zunächst den Debugger zu nutzen. Die hier betrachtete Prozedur sieht wie folgt aus.

```
CREATE OR REPLACE PROCEDURE ADEBUGTEST (wert INTEGER) AS BEGIN

DBMS_OUTPUT.PUT_LINE('Wert: ' || wert);
IF wert>0 THEN

DBMS_OUTPUT.PUT_LINE('Wert groesser 0');
IF wert>10 THEN

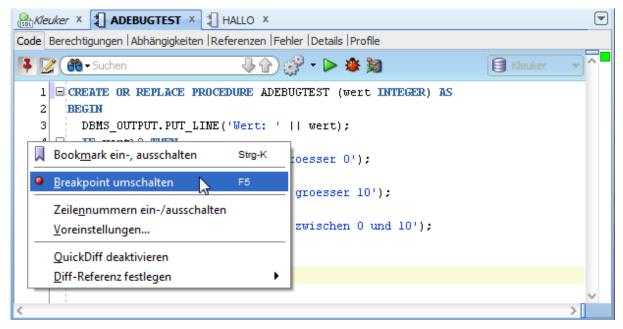
DBMS_OUTPUT.PUT_LINE('Wert groesser 10');
ELSE

DBMS_OUTPUT.PUT_LINE('Wert zwischen 0 und 10');
END IF;
END IF;
END ADEBUGTEST;
```

Dazu wird die Prozedur oder Funktion mit der zweiten Kompilierungsvariante übersetzt, dazu muss der kleine Pfeil neben den Zahnrädern zum Aufklappen der Varianten angeklickt werden.

```
Code Berechtigungen | Abhängigkeiten | Referenzen | Fehler | Details | Profile
                                                                          Kleuker
🧸 📝 ( 🍓 🕶 Suchen
   1 CREATE OR REPLACE PROCEDURE ADE 🙀 Für Debug kompilieren
                                          Kompilieren
          DBMS_OUTPUT.PUT_LINE('Wert:
   3
         IF wert>0 THEN
   4
   5
            DBMS_OUTPUT.PUT_LINE('Wert groesser 0');
   6
     IF wert>10 THEN
   7
              DBMS OUTPUT.PUT LINE('Wert groesser 10');
   8
              DBMS OUTPUT.PUT LINE('Wert zwischen 0 und 10');
   9
  10
            END IF:
  11
         END IF:
  12
       END ADEBUGTEST;
```

In der Spalte mit den Zeilennummern kann man mit der rechten Maustaste Breakpoints setzen. Im Debug-Modus bleibt das Programm an diesen Stellen stehen, so dass der Nutzer den weiteren Programmverlauf Schritt für Schritt nachverfolgen kann. Die Breakpoints sind vor "Für Debug kompilieren" zu setzen.

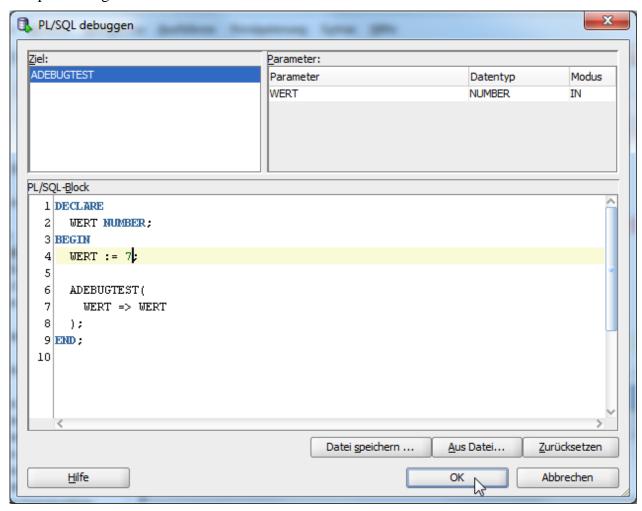


Das eigentliche Debuggen wird durch einen Druck auf den Marienkäfer in der oberen Zeile gestartet.

```
Code Berechtigungen | Abhängigkeiten | Referenzen | Fehler | Details | Profile

| Code | Profile | Code | C
```

Danach steht das gleiche Fenster wie für Testläufe zur Verfügung. Es müssen für die Parameter Beispielwerte gesetzt werden.



8.6 Einschub: Mangelnde Debug-Rechte

Sollte man eine Meldung der folgenden Form erhalten, sind die Session-Rechte nicht ordentlich gesetzt.

```
Debugging: IdeConnections%23Kleuker.jpr - Protokoll x

Anmeldung bei der Datenbank Kleuker.

PL/SQL wird ausgeführt: ALTER SESSION SET PLSQL_DEBUG=TRUE

PL/SQL wird ausgeführt: CALL DBMS_DEBUG_JDWP.CONNECT_TCP( '127.0.0.1', '49441' )

ORA-01031: insufficient privileges

ORA-06512: at "SYS.DBMS_DEBUG_JDWP", line 68

ORA-06512: at line 1

Diese Session erfordert DEBUG CONNECT SESSION- und DEBUG ANY PROCEDURE-Benutzerberechtigungen.

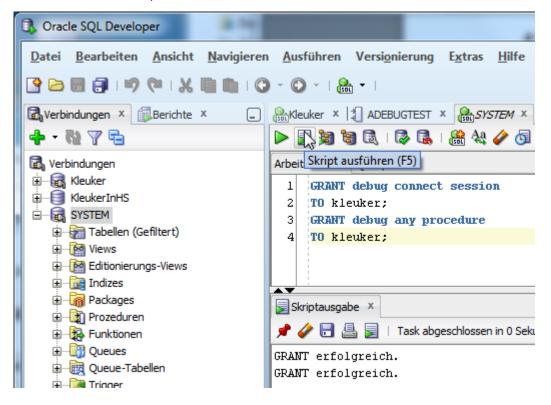
Prozess beendet.

Abmeldung von der Datenbank Kleuker.

usgeführt:IdeConnections%23Kleuker.jpr x Ausgabevariablen x Debugging: IdeConnections%23Kleuker.jpr x
```

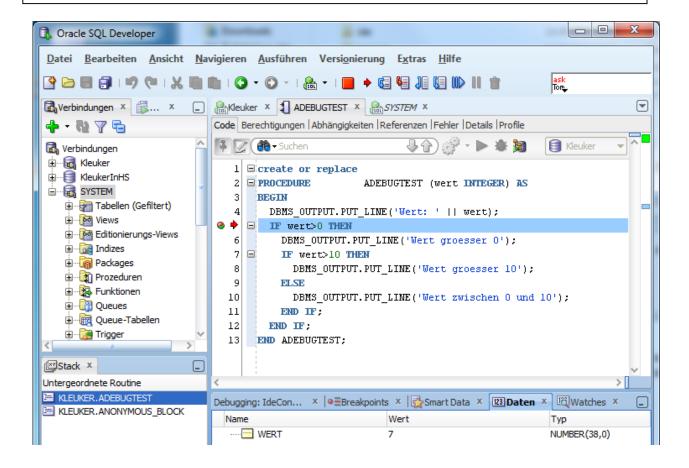
Dies kann man nachholen, in dem man sich als Administrator SYSTEM einloggt, also einfach eine weitere Verbindung für diesen nutzer herstellt. Der Nutzer kann dann direkt diese Rechte an den Nutzer geben, konkret:

```
GRANT debug connect session
TO kleuker;
GRANT debug any procedure
TO kleuker;
```

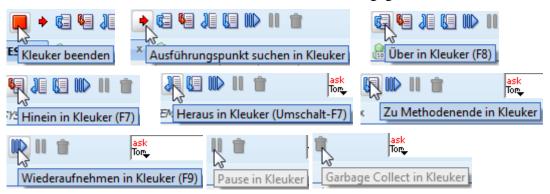


8.7 Einzelne Debug-Schritte

Nach der Klärung der Rechte wird der Debugger erneut gestartet. Die Ausführung stoppt beim ersten Breakpoint, danach kann die Oberfläche wie folgt aussehen.



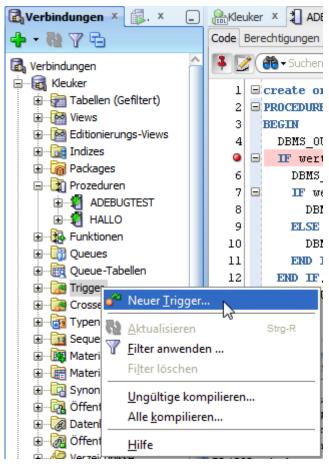
In der zweiten Zeile, ab dem roten Quadrat befinden sich die Steuermöglichkeiten für den Debugger. Die Bedeutung der einzelnen Knöpfe kann man den Beschriftungen entnehmen, wobei statt dem Procedure-Namen der Schema-Name angegeben wird.



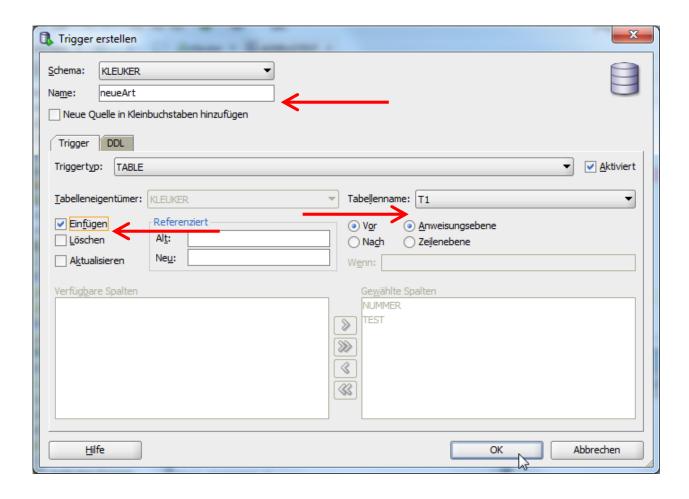
Typisch ist die Nutzung von "Step Over", deutsch "Über", wenn keine Prozeduren aufgerufen werden und von "Step Into", deutsch "Hinein", um den Aufruf von Prozeduren und Funktionen zu verfolgen.

9 Erstellung von PL/SQL-Triggern

Bei der Erstellung von Triggern kann man entweder den vom Werkzeug gewünschten Arbeitsweg gehen oder flexibler die Programmierung nutzen. Zum Anlegen eines neuen Triggers wird zunächst ein Rechtsklick auf Triggers ausgeführt und die Erstellung eingeleitet.

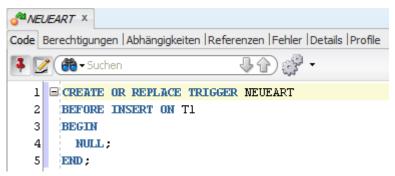


Danach erscheint die folgende Eingabemaske.



Um flexibel arbeiten zu können, kann man hier nur die minimal notwendigen Eingaben, den Namen, die Art des Triggers und die betroffene Tabelle eingeben bzw. auswählen.

Nach einem OK wird der Trigger-Editor angezeigt. Die Steuerung bei der späteren Veränderung von Triggern erfolgt analog zu den von Prozeduren.

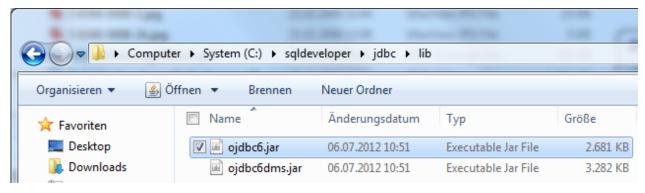


Generell ist zu beachten, dass ein Trigger nicht einfach getestet werden kann, zwar kann er zum Debuggen übersetzt werden, allerdings ist er logischerweise ohne ein eintretendes Event nicht ausführbar. Man sollte sich deshalb frühzeitig eine Testumgebung ausdenken.

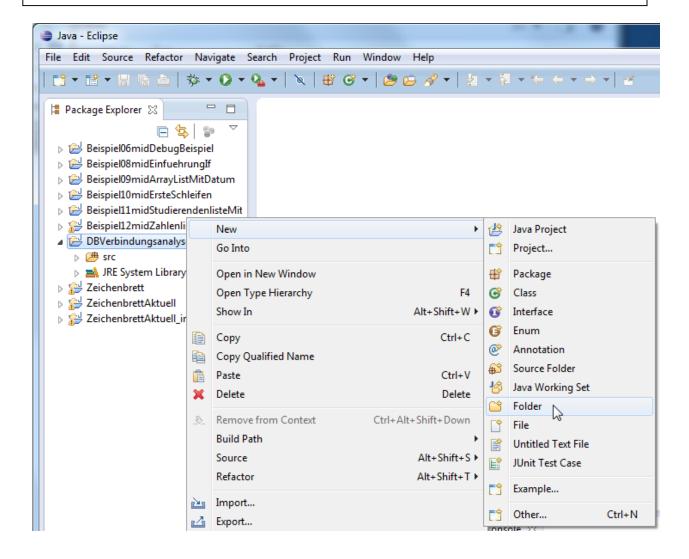
10 JDBC

Dieses Kapitel zeigt, wie eine Verbindung von Datenbanken mit Hilfe von JDBC aufgebaut werden kann. Es wird davon ausgegangen, dass eine lauffähige Eclipse-Entwicklungsumgebung für Java vorliegt. Details dazu können z. B. http://home.edvsz.hs-osnabrueck.de/skleuker/querschnittlich/SEU.pdf entnommen werden.

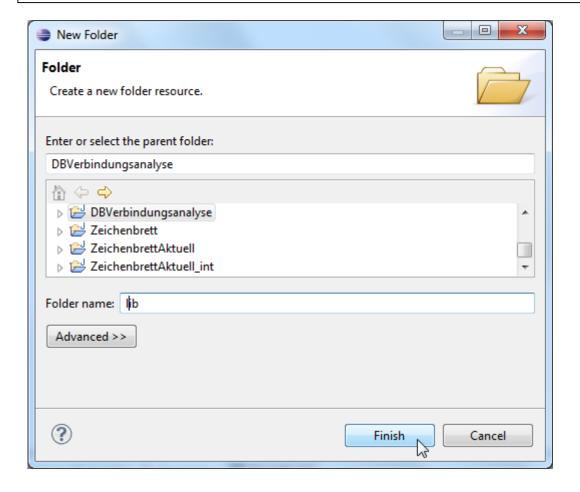
Zum Aufbau der Verbindung wird ein JDBC-Treiber benötigt, der vom jeweiligen Datenbankhersteller zu beziehen ist. Statt ihn hier von Oracle herunter zu laden, wird hier ausgenutzt, dass der JDBC-Treiber mit dem SQL Developer ausgeliefert wird. Ist dieser unter C:\sqldeveloper installiert, findet man ihn im Unterordner jdbc\lib.



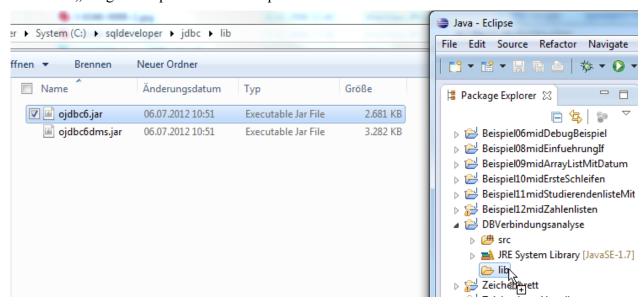
Nun wird in Eclipse ein Projekt, z. B. "DBVerbindungsanalyse" angelegt. In dem Projekt wird ein Ordner lib mit einem Rechtsklick auf dem Projekt und der Auswahl "New" und "Folder" erzeugt.



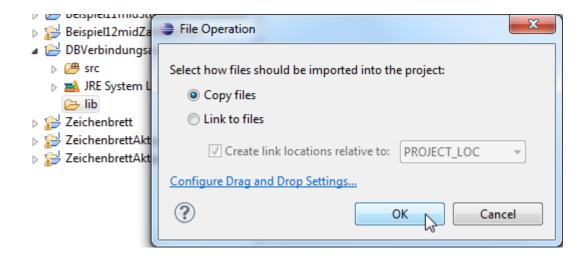
Hier muss nur der Name eingegeben werden.



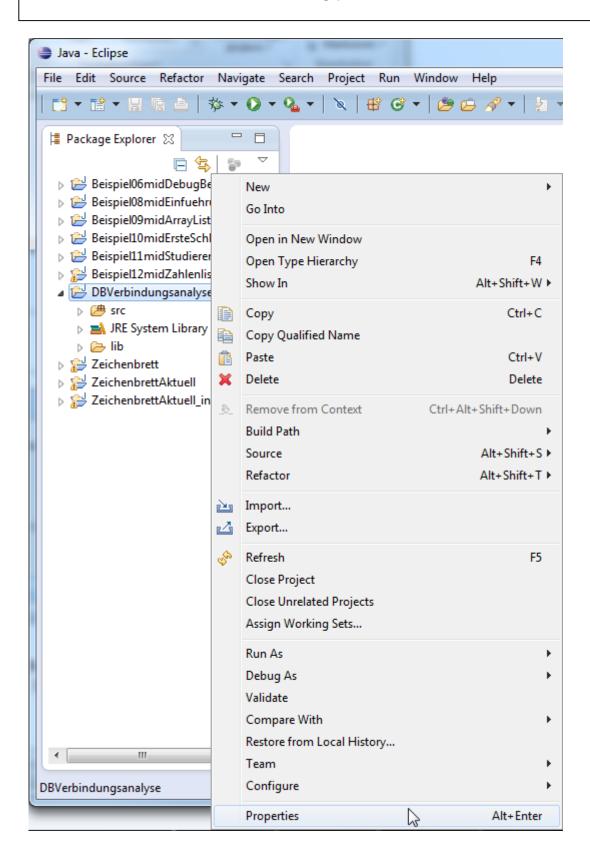
Im nächsten Schritt wird der JDBC-Treiber in diesen Ordner lib kopiert, dies kann z. B. durch einfaches "Drag & Drop" mit der Maus passieren.



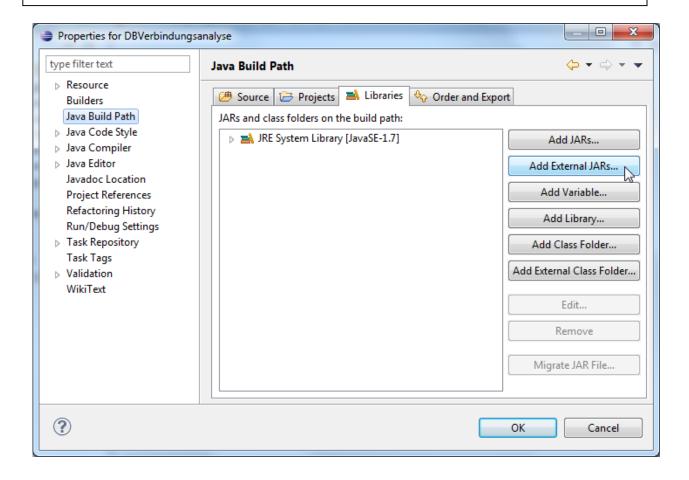
Man sollte "Copy Files" auswählen.



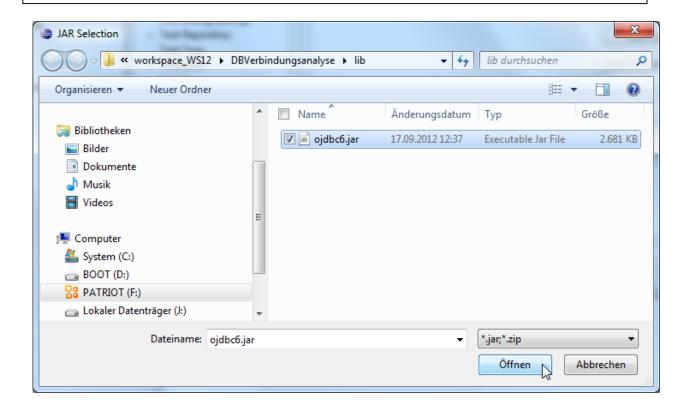
Im nächsten Schritt wird der JDBC-Treiber für das Projekt zur Verfügung gestellt, so dass man von Java aus, auf die Klassen zugreifen kann. Hierzu wird ein Rechtsklick auf dem Projekt gemacht und unten "Properties" gewählt.



Rechts wird "Java Build Path", dann der Reiter "Libraries" und der Knopf "Add External JARs" angeklickt.



Nun wird zu dem lib-Verzeichnis des Projekts gesteuert, wobei der genaue Pfad vom Pfad des Workspaces abhängt, und die jar-Datei über "Öffnen" hinzugefügt und die "Properties" über "OK" verlassen.



Nun kann ein normales Java-Programm unter Nutzung des Triebers entstehen. Die Projektstruktur sieht wie folgt aus.

```
■ DBVerbindungsanalyse

■ # src
■ db
□ DBVerbindung.java
□ MRE System Library [JavaSE-1.7]
■ Referenced Libraries
□ □ ojdbc6.jar - F:\workspaces\workspace_V
■ ib
■ ojdbc6.jar
```

Das Programm hat folgenden Aufbau, dabei wird eine Verbindung zu der vorher erstellten XE-Version aufgenommen. package db;

```
import java.sql.Connection;
import java.sql.DatabaseMetaData;
import java.sql.DriverManager;
import java.sql.SQLException;

public class DBVerbindung {
   private Connection con=null;
   private String dbAdresse="127.0.0.1"; //localhost
   private String dbInstanz="xe"; //XE-Version

   public DBVerbindung(){}
```

```
public void verbinden(String nutzer, String passwort){
    DriverManager.registerDriver(
        new oracle.jdbc.driver.OracleDriver());
    con = DriverManager.
    getConnection("jdbc:oracle:thin:@"+dbAdresse+":1521:"
        + dbInstanz,nutzer,passwort);
  } catch (SQLException e) {
    ausnahmeAusgeben(e);
  }
}
public void verbindungAnalysieren(){
  if (con==null){
    System.out.println("keine Verbindung vorhanden");
    return;
  try {
    DatabaseMetaData dbmd = con.getMetaData();
    System.out.println("DB-Name: " + dbmd.getDatabaseProductName()
        +"\nDB-Version: " + dbmd.getDatabaseMajorVersion()
        +"\nDB-Release: " + dbmd.getDriverMinorVersion()
        +"\nTransaktionen erlaubt: " + dbmd.supportsTransactions()
        +"\nbeachtet Großklein :" + dbmd.storesMixedCaseIdentifiers()
        +"\nunterstützt UNION :" + dbmd.supportsUnion()
        +"\nmax. Prozedurname: " + dbmd.getMaxProcedureNameLength());
  } catch (SQLException e) {
    ausnahmeAusgeben(e);
  }
}
public void verbindungTrennen(){
  if (con == null){
    System.out.println("eh keine Verbindung vorhanden");
    return;
 try {
    con.close();
  } catch (SQLException e) {
    ausnahmeAusgeben(e);
}
private void ausnahmeAusgeben(SQLException e){
 while (e != null){
        System.err.println("ORACLE Fehlercode: " + e.getErrorCode());
        System.err.println("SQL State: " + e.getSQLState());
        System.err.println(e);
        e = e.getNextException();
      }
}
public static void main(String[] s){
  DBVerbindung db= new DBVerbindung();
  db.verbinden("kleuker", "kleuker");
  db.verbindungAnalysieren();
```

```
db.verbindungTrennen();
}

Das Programm liefert die folgende Ausgabe.
DB-Name: Oracle
DB-Version: 11
DB-Release: 2
Transaktionen erlaubt: true
beachtet Großklein :false
unterstützt UNION :true
max. Prozedurname: 30
```

Soll eine Verbindung mit der Hochschuldatenbank aufgebaut werden, sind folgende zwei Variablen zu ändern.

```
private String dbAdresse="oracle-srv.edvsz.hs-osnabrueck.de"; // in HS
private String dbInstanz="Ora11"; //SID
```

Die Ausgabe entspricht der vorherigen Ausgabe.

11 SQLite

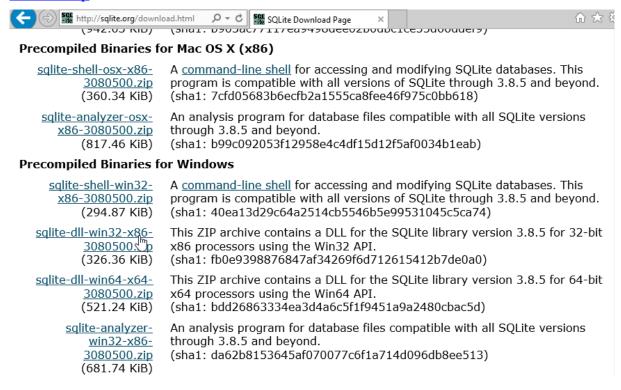
Oftmals benötigen auch sehr kleine Programme und Devices Datenbanken zur effizienten Verwaltung von Daten. Hierzu werden eingebettete Datenbanken genutzt, die die Kernfunktionalität relationaler Datenbanken anbieten, allerdings nur wenig Speicherplatz und Rechenzeit benötigen. Neben SQLite (http://sqlite.org/, sprich "Es Ku El leit") sind weitere Beispiele:

- Berkeley DB (http://www.oracle.com/technetwork/database/databasetechnologies/berkeleydb/overview/index.html)
- embedded Apache Derby (http://db.apache.org/derby/)

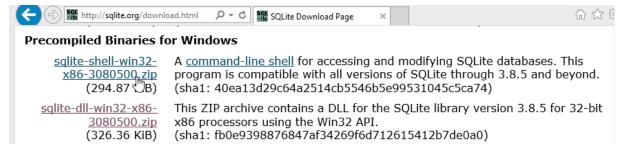
SQLite hat den Reiz, dass es bereits direkt in Android und auch in iOS zur einfachen Nutzung eingebaut ist. Weiterhin ist die Datenbank auch einfach in Windows Phone 8 nutzbar und kann in fast jedem Betriebssystem genutzt werden. Minimale Voraussetzung ist ein C-Compiler und ein "echtes" Betriebssystem.

11.1 Installation

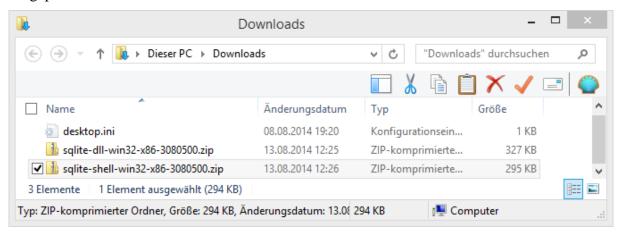
Hier werden nur die ersten Schritte mit SQLite unter Windows beschrieben. Der Download kann über die Seite http://sqlite.org/download.html starten, es wird hier ein "Precompiled Binary for Windows" genutzt, dass zur x86- oder x64-Variante des Betriebssystems passen sollte. Die benötigte Datei ist hier für x86 http://sqlite.org/2014/sqlite-dll-win32-x86-3080500.zip.



Zu Arbeit mit SQLite stehen vergleichbar zum SQL Developer einige Werkzeuge zur Verfügung. Eines davon wird später noch kurz vorgestellt. Für die ersten Schritte ist aber das Command-Line-Werkzeug ("DOS-Box") sehr nützlich, das ebenfalls von der Webseite, unmittelbar über dem vorher genutzten Link, heruntergeladen werden kann (http://sqlite.org/2014/sqlite-shell-win32-x86-3080500.zip).

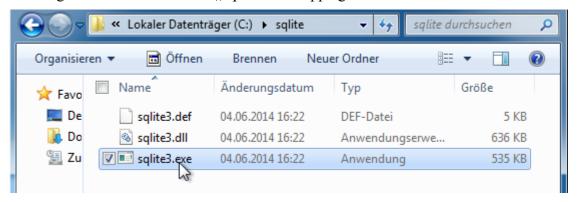


Die beiden erhaltenen Zip-Dateien können in einem beliebigen Ordner, im Beispiel C:\sqlite, ausgepackt werden. Die Datenbank ist damit installiert und nutzbar.

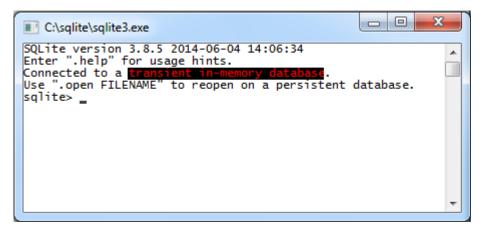


11.2 Erste Nutzung

Für SQLite muss kein Datenbank-Server gestartet werden, da die gesamte Datenbankfunktionalität auf Basis einer einfachen Datei funktioniert. Zur ersten Datenbank-Nutzung wird einfach die Datei "sqlite.exe" doppelgeklickt.



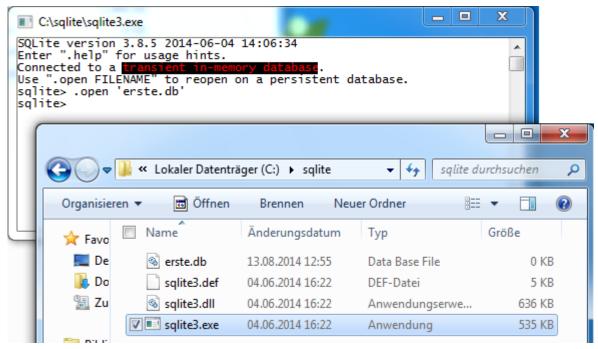
Die Startmeldung deutet bereits an, dass aktuell mit einer internen Datenbank gearbeitet wird, was ein typischer Ansatz für eingebettete Datenbanken ist. Diese interne Datenbank kann normal genutzt werden, ist aber mit Beendung der Datenbanknutzung verloren. Die Datenbank ist damit für interne Datenverwaltungen nutzbar, deren Daten nicht persistiert werden sollen. Die Möglichkeit, die Datenbank später zu persistieren, ist gegeben, wird hier aber nicht betrachtet.



Eine neue Datenbank wird mit dem gleichen Befehl angelegt, der auch zum Öffnen einer existierenden Datenbank genutzt wird. Dies ist

.open <meinRechtWillkürlichGewählterDatenbankname>

Betrachtet man das Dateiverzeichnis, sieht man, dass eine neue Datei, die Datenbank, angelegt wurde. Generell können auch Pfadnamen mit übergeben werden, so dass die Datenbank nicht im gleichen Verzeichnis liegen muss. Enthält der Name, wie im Beispiel, keine Leerzeichen, können die Anführungsstriche weggelassen werden.



Generell beginnen Befehle des Kommandozeilen-Werkzeugs mit einem Punkt. Eine relative Kurzübersicht ist über den Befehl "help" erreichbar.

```
\Sigma S
                                                                                C:\sqlite\sqlite3.exe
sqlite> .help
.backup ?DB? FILE
                           Backup DB (default "main") to FILE
.bail on|off
                           Stop after hitting an error.
                                                              Default OFF
                           Clone data into NEWDB from the existing database
List names and files of attached databases
.clone NĖWDB
.databases
                           Dump the database in an SQL text format
If TABLE specified, only dump tables matching
.dump ?TABLE? ...
                             LIKE pattern TABLE.
.echo on|off
                           Turn command echo on or off
                           Exit this program
.exit
.explain ?on|off?
                           Turn output mode suitable for EXPLAIN on or off.
                           With no args, it turns EXPLAIN on.
Turn display of headers on or off
.headers on|off
.help
                           Show this message
.import FILE TABLE
                           Import data from FILE into TABLE
                           Show names of all indices
If TABLE specified, only show indices for tables
.indices ?TABLE?
                             matching LIKE pattern TABLE.
                           Load an extension library
load FILE ?ENTRY?
.log FILE|off
                           Turn logging on or off.
                                                        FILE can be stderr/stdout
                           Set output mode where MODE is one of:
.mode MODĖ ?TABLE?
                             CSV
                                        Comma-separated values
                             column
                                                                    (See .width)
                                        Left-aligned columns.
                             html
                                        HTML  code
                                        SQL insert statements for TABLE
                             insert
                              line
                                        One value per line
                             list
                                        Values delimited by .separator string
                           tabs Tab-separated values
tcl TCL list elements
Use STRING in place of NULL values
.nullvalue STRING
.once FILENAME
                           Output for the next SQL command only to FILENAME
.open ?FILENAME?
                           Close existing database and reopen FILENAME
.output ?FILENAME?
                           Send output to FILENAME or stdout
.print STRING...
                           Print literal STRING
.prompt MAIN CONTINUE Replace the standard prompts
.quit
                           Exit this program
.read FILENAME
                           Execute SQL in FILENAME
                           Restore content of DB (default "main") from FILE
.restore ?DB? FILE
                           Write in-memory database into FILE
Show the CREATE statements
.save FILE
.schema ?TABLE?
                             If TABLE specified, only show tables matching
                             LIKE pattern TABLE.
                           Change separator used by output mode and .import
.separator STRING
                           Run CMD ARGS... in a system shell
Show the current values for various settings
.shell CMD ARGS...
.show
.stats on|off
                           Turn stats on or off
                           Run CMD ARGS... in a system shell
List names of tables
If TABLE specified, only list tables matching
.system CMD ARGS...
.tables ?TABLE?
                             LIKE pattern TABLE.
.timeout MS
                           Try opening locked tables for MS milliseconds
.timer on|off
                           Turn SQL timer on or off
.trace FILEloff
                           Output each SQL statement as it is run
                           Print the name of the VFS stack
Set column widths for "column" r
.vfsname ?AUX?
.width NUM1 NUM2 ...
                             Negative values right-justify
sqlite>
```

SQLite unterstützt fast alle klassischen SQL-Befehle. Die einzige Ausnahme ist, dass die Datenbank keine Rechteverwaltung enthält und so GRANT sowie REVOKE nicht nutzbar sind. Die Software, die die Datenbank nutzt, muss also sicherstellen, dass nur berechtigte Personen

und Programme die Datenbank nutzen. Natürlich kann auf Grundlage von SQLite eine eigene Nutzerverwaltung entwickelt werden.

Die SQL-Befehle werden einfach eingegeben und müssen mit einem Semikolon enden. Wird kein Semikolon eingegeben, kann man einfach einen Befehl in der nächsten Zeile fortsetzen, was auch durch das veränderte Prompt-Zeichen am linken Rand deutlich wird. Sollte man das Semikolon einfach vergessen haben, kann man es direkt in der nächsten Zeile eintippen und "Return" drücken. Auf Formatierungsmöglichkeiten für die Ausgabe wird nicht eingegangen.

Auf Groß- und Kleinschreibung muss nur innerhalb von Strings geachtet werden. Das folgende kleine Beispiel zeigt eine Besonderheit, dass Spalten mit dem Typ INTEGER, die als PRIMARY KEY definiert werden, automatisch mit einem "Autoincrement" versehen sind. Dies bedeutet, dass automatisch ein neuer Wert für diesen Schlüssel, durch den alten Maximalwert erhöht um eins, berechnet wird.

```
\Sigma S
                                                                                                                 C:\sqlite\sqlite3.exe
sqlite> CREATE TABLE Punkt(
                 ID INTEGER,
     ...>
                   X INTEGER,
      ...>
                   Y INTEGER
      ...>
                   PRIMARY KEY(ID)
sqlite> INSERT INTO Punkt VALUES(1,42,43);
sqlite> INSERT INTO Punkt(X,Y) VALUES(98,99);
sqlite> INSERT INTO Punkt(5,17,17);
Error: near "5": syntax error
sqlite> INSERT INTO Punkt VALUES(5,17,17);
sqlite> INSERT INTO Punkt VALUES(NULL,25,26);
sqlite> SELECT * FROM Punkt
1|42|43
2|98|99
5|17|17
6|25|26
sqlite>
```

Möchte man Befehle wiederholen oder Eingaben korrigieren, sind vorher eingegebene Zeilen über die Pfeil-Nach-Oben- und die Pfeil-Nach-Unten-Tasten erreichbar. Innerhalb einer Zeile kann durch die Pfeil-Nach-Links- und Pfeil-Nach-Rechts-Tasten einfach manövriert werden. Änderungen sind dann auch möglich. Am Rande sei angemerkt, dass die Fehlermeldungen von SQLite generell sehr schwach und unpräzise sind.

Sollen die bisher eingegebenen Befehle nachträglich in einer Datei gespeichert werden, ist dies mit zwei Befehlen möglich. Mit Hilfe des Befehls "dump" werden die bisher eingegeben Befehle ausgeben. Genau wird um die Aktionen herum eine Transaktion mit Start und Ende ausgegeben. Weiterhin wird ein PRAGMA-Befehl an den Anfang gesetzt, wobei generell mit PRAGMA-Befehlen Einstellungen der Datenbank vorgenommen werden können und Eigenschaften der Datenbank abfragbar sind. Im konkreten Fall sieht man, dass explizit angegeben wird, dass die Datenbank im Standardverhalten, zumindest bis zur Version 3.8, FOREIGN KEY-Beziehungen ignoriert. Dies ist durch einen vergleichbaren PRAGMA-Befehl leicht änderbar.

Aktuell finden alle Ausgaben in der Konsole statt. Durch den Befehl "output <Dateiname»" werden alle folgenden Ausgaben in die genannte Datei umgelenkt. Mit dem Befehl "output stdout" finden die nachfolgenden Ausgaben wieder in der Konsole statt. Möchte man Befehle aus einer Datei ausführen, ist der Befehl "read <Datei»" zu nutzen.

```
\Sigma S
                                                                               C:\sqlite\sqlite3.exe
sqlite> .dump
PRAGMA foreign_keys=OFF;
BEGIN TRANSACTION;
CREATE TABLE Punkt(
  ID INTEGER,
   X INTEGER,
Y INTEGER,
   PRIMARY KEY(ID)
INSERT INTO "Punkt" VALUES(1,42,43);
INSERT INTO "Punkt" VALUES(2,98,99);
INSERT INTO "Punkt" VALUES(5,17,17);
INSERT INTO "Punkt" VALUES(6,25,26);
COMMIT;
sqlite> .output 'punkt.sql'
sqlite> .dump
sqlite> .output stdout
sqlite> SELECT * FROM Punkt
           WHERE X=Y-1;
1|42|43
2 | 98 | 99
6 | 25 | 26
sqlite>
   🔼 | 📘 🤚 🌈 🖚 | punkt.sql - WordPad
              Start
                       Ansicht
             Ausschneiden
                                                - 11 - A A
                               Courier New
             Kopieren
                               F K U abe X X A A
                                                                                           Bil
    Einfügen
         Zwischenablage
                                            Schriftart
                                                                           Absatz
               3 - 1 - 2 - 1 - 1 - 1 - 2 - 1 - 1 - 1 - 2 - 1 - 3 - 1 - 4 - 1 - 5 - 1 - 6 - 1 - 7 - 1 - 8 - 1 - 9
                                   PRAGMA foreign keys=OFF;
                                   BEGIN TRANSACTION;
                                   CREATE TABLE Punkt (
                                      ID INTEGER,
                                       X INTEGER,
                                       Y INTEGER,
                                       PRIMARY KEY(ID)
                                   INSERT INTO "Punkt" VALUES (1, 42, 43);
                                   INSERT INTO "Punkt" VALUES (2,98,99);
                                   INSERT INTO "Punkt" VALUES (5, 17, 17);
                                   INSERT INTO "Punkt" VALUES (6, 25, 26);
                                   COMMIT;
```

SQLite hat ein eigenes schwaches Typsystem, was es ermöglicht, fast beliebige Typen für Attribute anzugeben, da vereinfachend die Typen bei der Deklaration der Tabelle und auch bei INSERT-Befehlen nicht berücksichtigt werden. Wer nur mit stark getypten Sprachen, wie C, C++ und Java gearbeitet hat, wird dies irritierend finden, der schwach getypte Ansatz bietet aber neben kleinen Nachteilen auch einige Vorteile, wie die sehr flexible Nutzung von SQL-Skripten anderer Datenbanken.

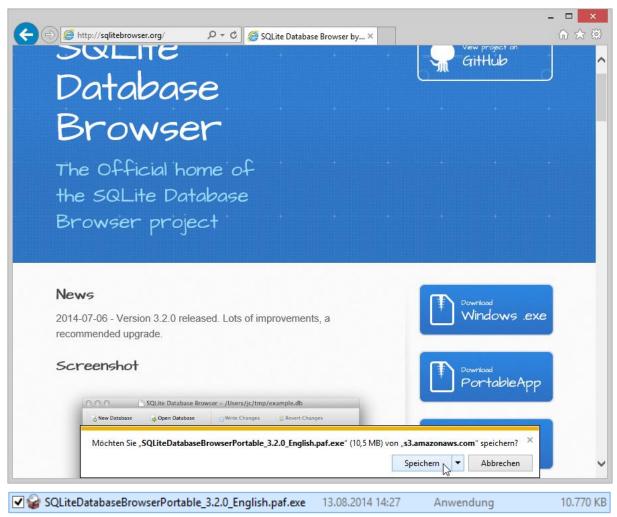
```
\Sigma S
                                                                                                                                                                                                                                  C:\sqlite\sqlite3.exe
 salite> CREATE TABLE WILD (
                                  A VARCHAR(42) PRIMARY KEY,
            ...>
                                  B reallyTineDouble,
            ...>
                                  C DATE,
                                 D WulliWulliVoodoo,
            ...>
                                 Ε,
            ...>
            ...>
                                 CONSTRAINT Wild_E CHECK(E < 42)
               ..>);
 sqlite> INSERT INTO Wild VALUES(1,2,3,4,5,6);
 sqlite> INSERT INTO Wild VALUES(1,2,3,4,5,7);
Error: UNIQUE constraint failed: WILD.A
sqlite> INSERT INTO Wild VALUES(2,2,3,4,42,7);
Error: UNIQUE CONSCIANCE
sqlite> INSERT INTO Wild VALUES(2,2,3,4,42,7);
Error: CHECK constraint failed: Wild_E
sqlite> INSERT INTO Wild VALUES(2,2,3,4,41,7);
sqlite> INSERT INTO Wild VALUES('2',2,3,4,41,7);
Error: UNIQUE constraint failed: WILD.A
sqlite> INSERT INTO Wild VALUES(NULL,2,3,4,41,7);
sqlite> INSERT INTO Wild VALUES(NULL,2,3,4,41,7);
sqlite> INSERT INTO Wild VALUES(NULL,'2','3','4','41','7');
Error: CHECK constraint failed: Wild_E
sqlite> INSERT INTO Wild VALUES(NULL,'2','3','4','0','7');
sqlite> INSERT INTO Wild VALUES(NULL, '2', '3', '4', '0', '7');
Error: CHECK constraint failed: Wild_E
sqlite> INSERT INTO Wild VALUES(NULL, '2', '3', '4', 41, '7');
sqlite> INSERT INTO Wild VALUES(NULL, NULL, '1.1.1970', NULL, NULL, NULL, '1.1.1970', NULL, NULL, NULL);
sqlite> INSERT INTO Wild VALUES('Hei', '2', '3', '4', '0', '7');
Error: CHECK constraint failed: Wild_E
sqlite> INSERT INTO Wild VALUES('Hei', '2', '3', '4', 40, '7');
sqlite> INSERT INTO Wild VALUES('Hei', '2', '3', '4', 40, '8');
Error: UNIQUE constraint failed: WILD.A
sqlite> INSERT INTO Wild VALUES('Hei', '2', '3', '4', 40, '8');
sqlite> SELECT # FROM Wild
...>;
 1|2.0|3|4|5|6
  2 2.0 3 4 41 7
  ||2.0|3|4|41|7
   2.0 3 4 41 7
     ||1.1.1970|||
  Hei | 2.0 | 3 | 4 | 40 | 7
  Hei |2.0|3|4|40|8
  sqlite> SELECT A+100 FROM Wild;
                                                                                                                                                                                                                                                            Ξ
 101
 102
 100
  100
 sqlite>
```

11.3 SQLite Database Browser

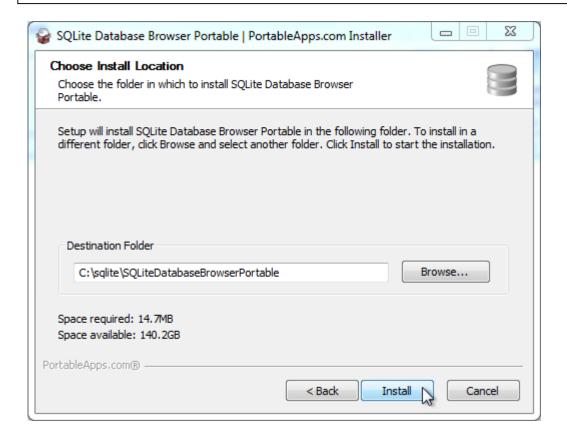
Der SQLite Database Browser (http://sqlitebrowser.org/) ist ein Beispiel für ein graphisches Werkzeug, das die Arbeit mit SQLite-Datenbanken etwas vereinfacht. Weitere Beispiele sind (unvollständige Liste):

- SQLiteStudio (http://sqlitestudio.pl/)
- sqlite-manager (Firefox-Plugin, https://code.google.com/p/sqlite-manager/)
- SQLitebrowser (http://www.cenote.de/sqlitebrowser/)

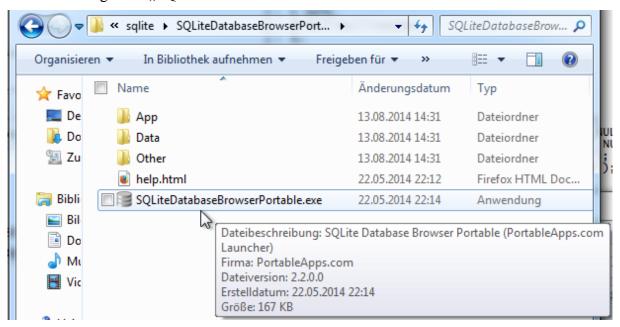
Der Browser existiert in verschiedenen Varianten, hier wird die portable Version genutzt, die ohne besondere Rechte auf jedem Rechner nutzbar ist. Die Version ist direkt über die Seite http://sqlitebrowser.org/ am rechten Rand erhältlich, es startet direkt der Download.



Die exe-Datei muss einmal ausgeführt werden. Der einzig relevante Schritt ist die Auswahl des Verzeichnisses, in das das Werkzeug kopiert wird. Da das Programm portabel ist, kann der Ordner dann in jedes beliebige Verzeichnis, z.B. C:\sqlite, verschoben werden.

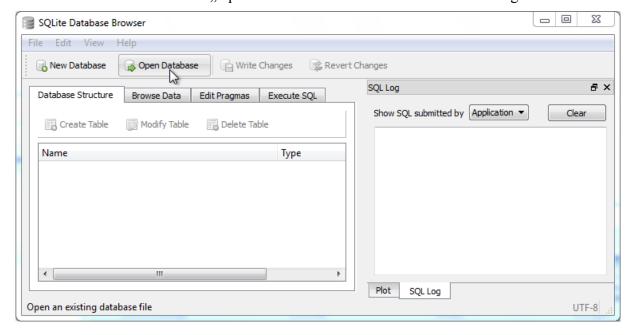


Der Start erfolgt über "SQLiteDatabaseBrowserPortable.exe".

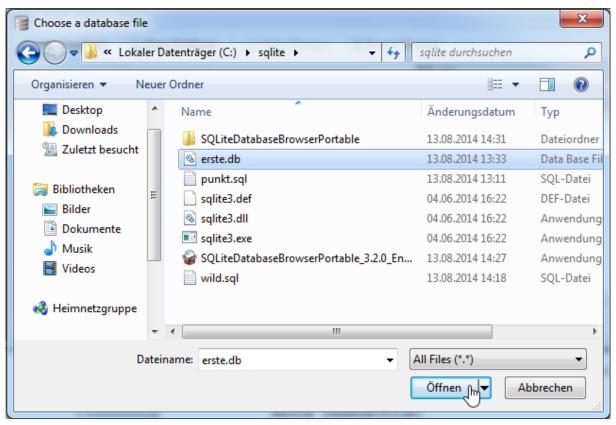


Das Werkzeug besteht im Wesentlichen aus vier Bereichen. In der obersten Zeile befindet sich das Menü, das u.a. unter File Möglichkeiten zum Import und Export bietet. In der Zeile darunter befinden sich vier zentrale Steuerungsknöpfe mit denen eine neue Datenbank erstellt, also die zugehörige Datei angelegt und eine existierende Datei geöffnet werden kann. Weiterhin gibt es zwei Knöpfe zur Transaktionssteuerung, mit denen die letzen Änderungen übernommen (Write

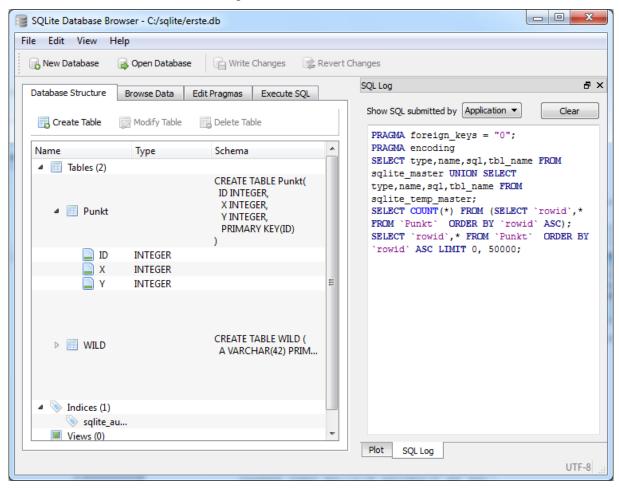
Changes, COMMIT) oder verworfen (Revert Changes, ROLLBACK) werden können. Am Rechten Rand kann man sich alternativ zwei Log-Dateien ansehen, in denen die letzten ausgeführten Befehle dieses Programms (Application) und die des Nutzers (User) sichtbar sind. Links befinden sich vier zentrale Reiter, mit denen die eigentliche Datenbanknutzung stattfindet. Zunächst soll über "Open Database" eine existierende Datenbank geöffnet werden.



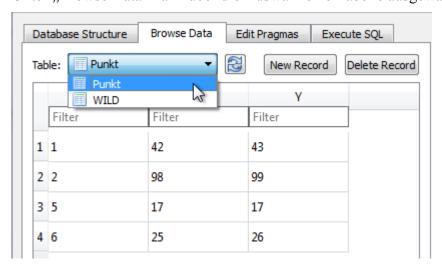
Die gewünschte Datenbank wird selektiert.



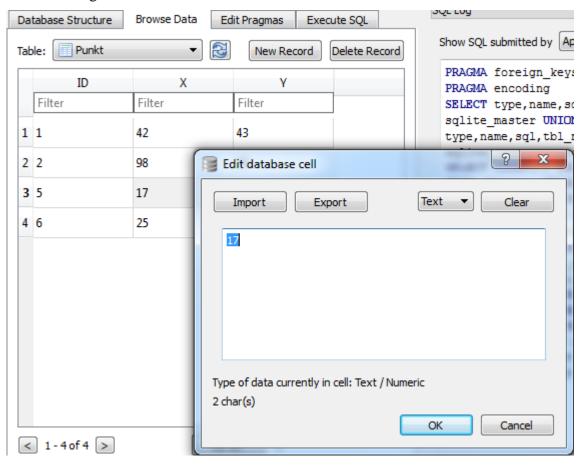
Unter dem Reiter "Database Structure" werden die enthaltenen Tabellen angezeigt. Sie können ausgeklappt werden, so dass alle Details sichtbar sind. Über die enthaltenen Menü-Punkte können z. B. Tabellen sehr einfach gelöscht werden.



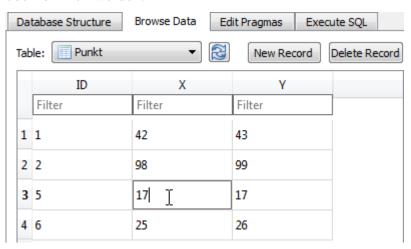
Unter "Browse Data" kann über die Auswahl eine Tabelle ausgewählt werden.



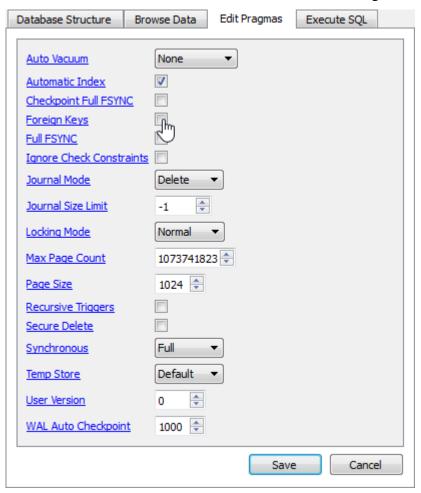
Weiterhin sind Änderungen der Tabelleninhalte möglich, dazu muss ein Doppelklick auf einem Feldelement gemacht werden.



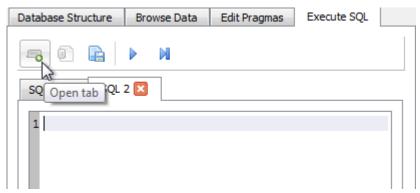
Durch zwei einfache Klicks mit etwas Zeitabstand sind Werte direkt in den einzelnen Zellen änderbar. Man beachte, dass erst durch "Write Changes" die Änderungen endgültig übernommen werden.



Über "Edit Pragmas" sind Datenbankeinstellungen änderbar. Hier kann z. B. die Nutzung von FOREIGN KEYs eingeschaltet werden. Man beachte, dass die Änderung gespeichert ("Save") werden muss und erst mit der nächsten Datenbanknutzung stattfindet.

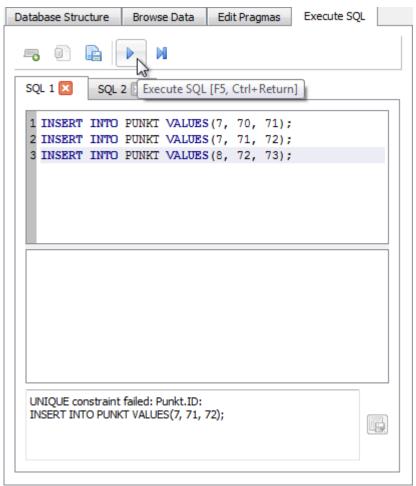


Unter dem Reiter "Execute SQL" können SQL-Befehle ausgeführt werden. Mit dem Icon linksoben sind weitere Reiter zur Ausführung von SQL-Befehlen hinzufügbar. Die Knöpfe daneben ermöglichen das Abspeichern und Laden von Dateien.

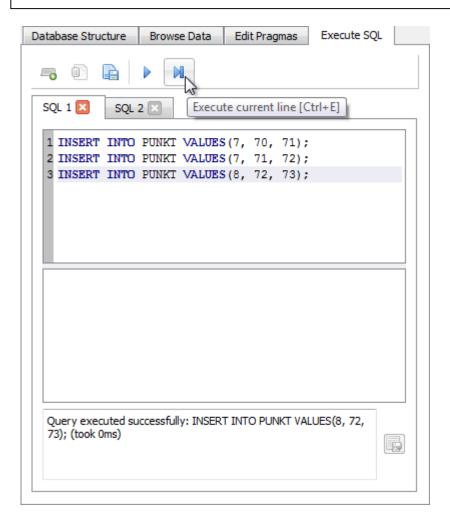


Zur Ausführung gibt es wieder zwei Möglichkeiten. Das einfache Dreieck führt den gesamten Inhalt des Fensters aus. Man beachte, dass bei Fehlern die nachfolgenden Zeilen nicht

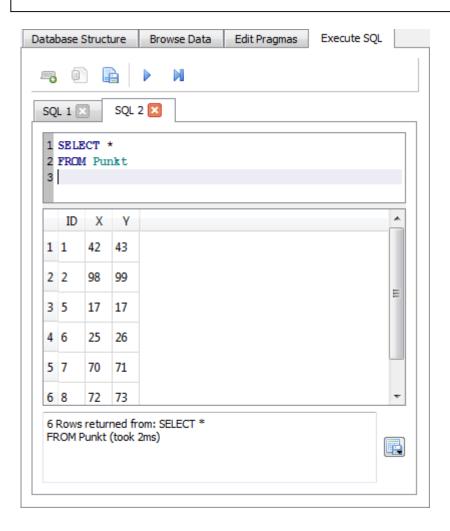
ausgeführt werden. Das Ergebnis der nachfolgenden Abbildung ist, dass nur eine neue Zeile eingetragen wird. Die zugehörige Fehleranzeige gibt darüber leider keine Auskunft.



Mit dem zweiten Pfeil wird nur ein Befehl ausgeführt. Wenn nichts markiert ist, wird nur die Zeile ausgeführt in der Cursor steht. Steht hier nur ein Teil eines Befehls, gibt es eine Fehlermeldung. Ist in der Eingabe ein Text markiert, wird der markierte Text als ein Befehl ausgeführt.



Bei SELECT-Befehlen erfolgt eine formatierte Ausgabe. Dabei wird in der linken Spalte ein Zähler ausgegeben.

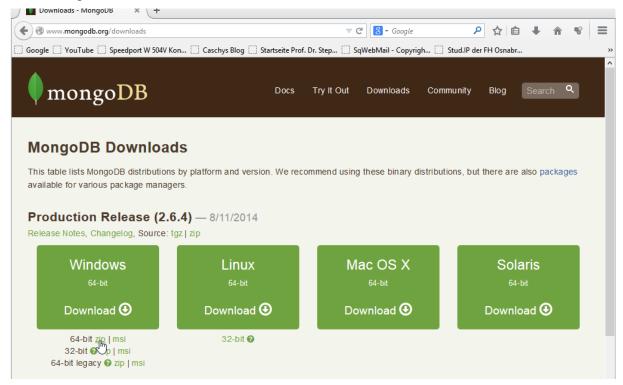


12 MongoDB

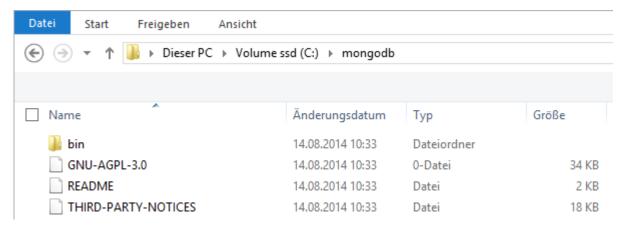
Neben den klassischen relationalen Datenbanken gibt es eine heterogene Gruppe von sogenannten NoSQL-Datenbanken. Dabei steht "NoSQL" für "not only SQL", was bereits andeutet, dass Konzepte von SQL keine oder eine geringe Rolle spielen. Eine wesentliche Motivation von NoSQL-Datenbanken ist, dass zusammengehörige Daten in relationalen Datenbanken oft über viele Tabellen verteilt sind. Der zunächst willkürliche Begriff "zusammengehörig" ist dabei etwas durch "Daten, die sehr häufig immer zusammen genutzt werden" präzisierbar. Durch die Idee, zusammengehörige Daten auch zusammen kompakt zu speichern, kommt es zu einem Ansatz von NoSQL-Datenbanken, deren prominenteste Vertretung die MongoDB (http://www.mongodb.org/) ist, deren Name sich aus dem englischen Begriff "humongous" für "gigantisch" ableitet. Generell stellen damit NoSQL-Datenbanken eine Alternative zu relationalen Datenbanken dar, die für bestimmte Aufgaben klare Vorteile bieten, aber generell nicht das Ziel der Verdrängung relationaler Datenbanken haben. In großen Projekten mit fast disjunkten Datenbereichen muss individuell über die Nutzung der passenden Datenbankfunktionalität nachgedacht werden, da ein Zusammenspiel verschiedener Datenbanken mit unterschiedlicher Technik möglich ist und sinnvoll sein kann.

12.1 Installation

Hier werden nur die ersten Schritte mit MongoDB unter Windows beschrieben. Der Download kann über die Seite http://www.mongodb.org/downloads starten, es wird hier ein "Zip-File für Windows" genutzt, dass zur 32-bit- oder 64-bit-Variante des Betriebssystems passen sollte. Die benötigte Datei ist hier für 64-bit https://fastdl.mongodb.org/win32/mongodb-win32-x86_64-2008plus-2.6.4.zip. Man beachte, dass mit der 32-bit-Variante nur deutlich kleinere Datenmengen verwaltbar sind.



Die heruntergeladene Datei kann in einem beliebigen Verzeichnis, hier C:\mongodb, entpackt werden.



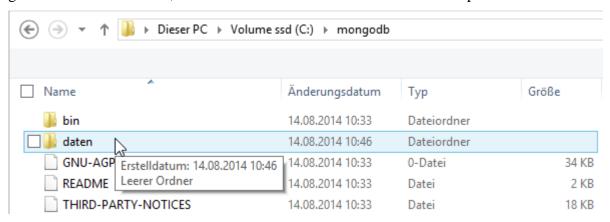
Die Datenbank-Software ist damit nutzbar.

12.2 Start der Datenbank

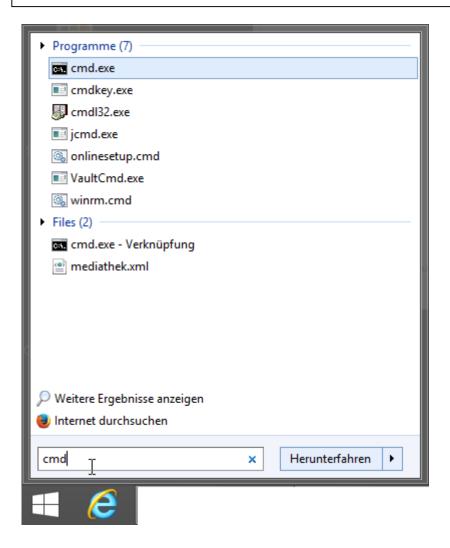
Es gibt eine größere Anzahl von Werkzeugen, die die Arbeit mit MongoDB erleichtern, die allerdings hier nicht vorgestellt werden. Der Fokus liegt auf dem Kommandozeilen-Werkzeug, mit dem erste Schritte sehr einfach ausführbar sind.

Als Vorbereitung muss ein neues Verzeichnis angelegt werden, in dem MongoDB die Daten speichert. Dieses Verzeichnis sollte am Anfang leer sein und für keine anderen Daten genutzt werden.

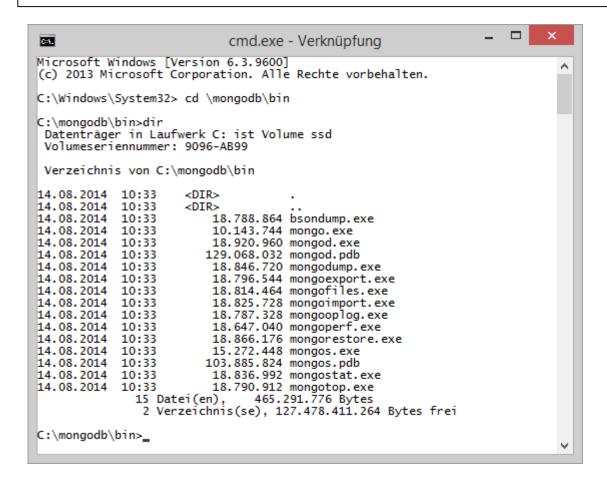
Im Beispiel wird zum Experimentieren ein Ordner C:\mongodb\daten angelegt, wobei es generell sehr unüblich ist, diesen Ordner im Installationsverzeichnis zu platzieren.



Zum Start wird ein Kommandozeilenfenster benötigt, was man z. B. unter dem Start-Knopf bekommt, wenn "cmd" in der unteren Zeile eingetippt und dann "cmd.exe" angeklickt wird.



Es wird in das Installationsverzeichnis von Mongodb, genauer das zugehörige bin-Verzeichnis gesteuert.



Zum Start der Datenbank wird das Programm mongod aufgerufen und als einziger Parameter mit "-- dbpath" das Verzeichnis mit den Daten angegeben. Die Anführungsstriche können weggelassen werden, wenn keine Leerzeichen im Pfad enthalten sind.

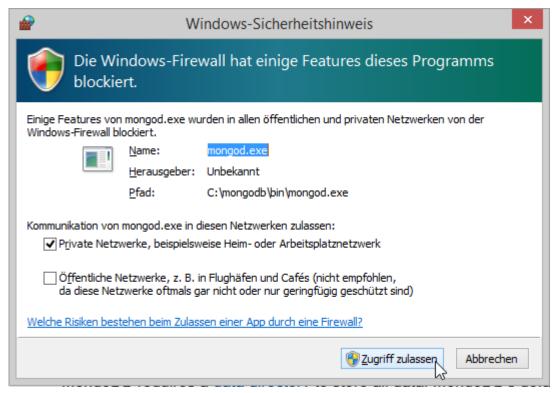
mongod --dbpath "c:\mongodb\daten"

```
microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. Alle Rechte vorbehalten.

C:\Windows\System32>cd \mongodb\bin

C:\mongodb\bin>mongod --dbpath "c:\mongodb\daten"
2014-08-14T10:52:21.110+0200 [initandlisten] MongoDB starting : pid=2780 port=27017 dbpath=c:\mongodb\daten 64
-bit host=x
2014-08-14T10:52:21.111+0200 [initandlisten] targetMinOS: Windows 7/Windows Server 2008 R2
2014-08-14T10:52:21.111+0200 [initandlisten] db version v2.6.4
2014-08-14T10:52:21.111+0200 [initandlisten] git version: 3a830be0eb92d772aa855ebb711ac91d658ee910
2014-08-14T10:52:21.112+0200 [initandlisten] build info: windows sys.getwindowsversion(major=6, minor=1, build =7601, platform=2, service_pack='Service Pack 1') B005T_LIB_VERSION=1_49
2014-08-14T10:52:21.112+0200 [initandlisten] allocator: system
2014-08-14T10:52:21.112+0200 [initandlisten] options: { storage: { dbPath: "c:\mongodb\daten" } }
2014-08-14T10:52:21.116+0200 [initandlisten] recover: no journal files present, no recovery needed
2014-08-14T10:52:21.1148+0200 [initandlisten] waiting for connections on port 27017
```

Abhängig von der installierten Sicherheits-Software müssen ggfls. einige Rechte eingeräumt werden.



Nun ist die Datenbank gestartet. Das Fenster muss, solange die Datenbank genutzt wird, geöffnet bleiben. Bei einem professionelleren Einsatz kann der Datenbankstart als Windows-Service konfiguriert werden, so dass auch ein unmittelbarer Start in Verbindung mit dem Hochfahren des Rechners möglich ist. Weiterhin ist es sinnvoll, das bin-Verzeichnis der MongoDB in die Pfad-Variable des Rechners einzutragen.

12.3 Erste Nutzung

Zur ersten Nutzung wird wieder ein Kommandozeilenfenster (cmd.exe) geöffnet. Da im Beispiel das bin-Verzeichnis nicht im Pfad eingetragen ist, wird wieder in das bin-Verzeichnis gesteuert und die Verbindung mit "mongo" gestartet. Das Fenster wartet jetzt auf Eingaben zur Interaktion mit der Datenbank.

```
cmd.exe - Verknüpfung - mongo

Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. Alle Rechte vorbehalten.

C:\Windows\System32>cd \mongodb\bin

C:\mongodb\bin>mongo
MongoDB shell version: 2.6.4
connecting to: test
Welcome to the MongoDB shell.
For interactive help, type "help".
For more comprehensive documentation, see
    http://docs.mongodb.org/
Questions? Try the support group
    http://groups.google.com/group/mongodb-user
>
```

Wenn man das eigentliche Fenster der Datenbank beobachtet, sieht man in der letzten Zeile, dass eine Verbindung akzeptiert wurde.

```
cmd.exe - Verknüpfung - mongod --dbpath "c:\mongodb\daten" - \( \text{\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\text{$\t
```

Eine Übersicht über die Datenbankbefehle gibt der Befehl "help" aus.

```
> help
         db.help()
db.mycoll.help()
                                          help on db methods
                                          help on collection methods
         sh.help()
                                          sharding helpers
         rs.help()
                                          replica set helpers
         help admin
                                          administrative help
         help connect
                                          connecting to a db help
         help keys
                                          key shortcuts
         help misc
                                          misc things to know
         help mr
                                          mapreduce
         show dbs
                                          show database names
         show collections
                                          show collections in current database
         show users
                                          show users in current database
         show profile
                                          show most recent system.profile entries wit
h time >= 1ms
         show logs
                                          show the accessible logger names
show log [name]
y, 'global' is defaul+
                                          prints out the last segment of log in memor
         use <db_name>
                                          set current database
                                         list objects in collection foo
list objects in foo where a == 1
result of the last line evaluated; use to f
         db.foo.find()
         db.foo.find( { a : 1 } )
         it
urther iterate
         DBQuery.shellBatchSize = x
                                         set default number of items to display on s
hell
         exit
                                          quit the mongo shell
```

12.3.1 JavaScript

Diese MongoDB-Shell nutzt die Programmiersprache JavaScript als Hilfsmittel zur Kommunikation mit der Datenbank. Die folgenden Zeilen zeigen, wie eine einfache Variable und ein zusammengesetztes Objekt deklariert werden. Für das Objekt wird die JSON-Notation genutzt, die mittlerweile sehr weit über die Grenzen von JavaScript verbreitet ist und als kompaktere Alternative zu XML genutzt wird. MongoDB nutzt JSON-Strukturen, genauer die sehr eng verwandten BSON-Strukturen oder Objekte, um Daten abzuspeichern. Dies JSON-Objekte werden oft auch als "Dokumente" bezeichnet.

Man sieht, dass bei nicht abgeschlossenen JavaScript-Befehlen weitere zum Befehl gehörende Eingaben in der Folgezeile stehen. Man kann Befehle mit einem Semikolon abschließen, muss es aber nicht.

```
var wert = 42;
print(wert)
var pkt1 = { id: 1, x: 0, y: 0 }
pkt1.x
```

12.3.2 Anlegen der Datenbank

Bisher wurden nur JavaScript-Befehle eingegeben. Nun soll eine Datenbank angelegt und genutzt werden. Der Befehl "show" zeigt dabei alle vorhandenen Datenbanken. Mit "use <Datenbankname>" wird zur Datenbank mit diesem Namen gewechselt, falls sie noch nicht existiert, wird sie angelegt. Man beachte, dass Tippfehler hier zu drastischen Problemen führen können, da eventuell in einer neuen falschen Datenbank gearbeitet wird. In der Variablen db steht der Name der aktuell genutzten Datenbank, die am Anfang auf "test" steht.

Im folgenden Beispiel fällt auf, dass die neu angelegte Datenbank nicht sofort bei den existierenden Datenbanken aufgeführt wird. Dies passiert erst nach der ersten Nutzung der Datenbank.

```
db
     show dbs
     use db01
     show dbs
      db
db
test
> show dbs
admin
       (empty)
local
       0.078GB
> use db01
switched to db db01
show dbs
admin
       (empty)
local
       0.078GB
db01
```

Das Grundkonzept von MongoDB ist, dass Dokumente, also JSON-Objekte, in Sammlungen, also Collections, verwaltet werden. Dies ist durchaus durch die Vorstellung einer Liste in C++ oder Java zu veranschaulichen.

12.3.3 Ein- und Ausgabe einer Collection

Eine Collection wird durch ihre erste Nutzung angelegt, wobei der Zugriff immer über <Datenbank>.<CollectionName> erfolgt. Im folgenden Beispiel werden mehrere Dokumente in eine Collection eingefügt und mit Hilfe des Befehl "show collections" die Existenz der Collection überprüft. Da MongoDB schemafrei ist, es also keine direkte Möglichkeit gibt, für Dokumente eine feste Struktur vorzugeben, können Dokumente mit einer beliebigen Struktur in einer Collection enthalten sein. Ob dies sinnvoll ist, entscheidet dann das Programm, das die

Datenbank nutzt, da man natürlich da für einen einheitlichen Aufbau der Dokumente sorgen kann. Die Schemafreiheit ist ein zentrales Konzept, da so Dokumente mit den einfachsten Mitteln flexibel verändert werden können. Das Beispiel zeigt, dass das pkt-Objekt sehr einfach eine neue Eigenschaft bekommt. Was dann passiert, wenn das Objekt mehrfach in die Datenbank eingetragen wird, soll danach geklärt werden.

```
wert
                               pkt1
                               db.punkte.insert(pkt1);
                               db.punkte.insert({id:2, x:42, y:42});
                               db.punkte.insert({id:2, x:wert+1, y:wert-1});
                               pkt1.z = 1;
                               pkt1
                               db.punkte.insert(pkt1);
                               db.punkte.insert(pkt1);
                               db.punkte.insert({id:8, x:"Hai", fisch:"Fisch"});
                                show collections
 > wert
  42
42
> pkt1
{ "id" : 1, "x" : 0, "y" : 0 }
> db.punkte.insert(pkt1);
WriteResult({ "nInserted" : 1 })
> db.punkte.insert({id:2, x:42, y:42});
WriteResult({ "nInserted" : 1 })
db.punkte.insert({id:2, x:wert+1, y:wert+1, y:we
 > db.punkte.insert({id:2, x:wert+1, y:wert-1});
WriteResult({ "nInserted" : 1 })
  > pkt1.z = 1;
pkt1
{ "id" : 1, "x" : 0, "y" : 0, "z" : 1 }
> db.punkte.insert(pkt1);
WriteResult({ "nInserted" : 1 })
> db.punkte.insert(pkt1);
WriteResult({ "nInserted" : 1 })

WriteResult({ "nInserted" : 1 })
 > db.punkte.insert({id:8, x:"Hai", fisch:"Fisch"});
WriteResult({ "nInserted" : 1 })
   > show collections
  punkte
 system.indexes
```

Betrachtet man nebenbei die eigentliche Datenbank-Konsole, erkennt man, dass die Datenbank und dann die Collection angelegt wurde.

```
cmd.exe - Verknüpfung - mongod --dbpath "c:\mongodb\daten"
2014-08-14T11:04:12.681+0200 [initandlisten] connection accepted from 127.0.0.1:55562 #1 (1 connection now ope
2014-08-14T11:08:21.217+0200
2014-08-14T11:08:21.217+0200
2014-08-14T11:08:21.217+0200
                                                                                           mem (MB) res:37 virt:298
mapped (incl journal view):160
connections:1
                                                         [clientcursormon]
[clientcursormon]
                                                          [clientcursormon]
[clientcursormon]
                                                                                            mem (MB) res:37 virt:298
mapped (incl journal view):160
connections:1
2014-08-14T11:13:21.242+0200
2014-08-14T11:13:21.242+0200
2014-08-14T11:13:21.242+0200
2014-08-14T11:13:21.243+0200
2014-08-14T11:18:21.265+0200
2014-08-14T11:18:21.265+0200
2014-08-14T11:18:21.266+0200
2014-08-14T11:23:21.292+0200
2014-08-14T11:23:21.292+0200
2014-08-14T11:23:21.292+0200
2014-08-14T11:28:21.315+0200
2014-08-14T11:28:21.315+0200
2014-08-14T11:28:21.315+0200
                                                          [clientcursormon]
                                                                                            mem (MB) res:37 virt:298
mapped (incl journal view):160
connections:1
mem (MB) res:37
                                                          [clientcursormon]
                                                         [clientcursormon]
[clientcursormon]
[clientcursormon]
[clientcursormon]
[clientcursormon]
                                                                                            mem (MB) res:37 virt:298
mapped (incl journal vi
                                                                                           connections:1
mem (MB) res:37 virt:298
mapped (incl journal view):160
connections:1
mem (MB) res:37 virt:298
mapped (incl journal view):160
connections:1
llocating new datafile comments:
                                                                                              connections:1
                                                          [clientcursormon]
[clientcursormon]
2014-08-14T11:28:21.315+0200
2014-08-14T11:33:21.341+0200
                                                          [clientcursormon]
[clientcursormon]
[clientcursormon]
2014-08-14T11:33:21.341+0200
2014-08-14T11:33:21.341+0200 [clientcursormon] connections:1
2014-08-14T11:35:31.447+0200 [FileAllocator] allocating new datafile c:\mongodb\daten\db01.ns, filling with ze
roes...
2014-08-14T11:35:31.447+0200 [FileAllocator] creating directory c:\mongodb\daten\_tmp
2014-08-14T11:35:31.450+0200 [FileAllocator] done allocating datafile c:\mongodb\daten\db01.ns, size: 16MB, t
ook 0.001 secs
2014-08-14T11:35:31.451+0200 [FileAllocator] allocating new datafile c:\mongodb\daten\db01.0, filling with zer
2014-08-14T11:35:31.453+0200 [FileAllocator] done allocating datafile c:\mongodb\daten\db01.0, size: 64MB, to
2014-08-14T11:35:31.454+0200 [conn1] build index on: db01.punkte properties: { v: 1, key: { _id: 1 }, name: "_
id_", ns: "db01.punkte" }
2014-08-14T11:35:31.454+0200 [conn1] added index to empty collection
2014-08-14T11:36:21.355+0200 [clientcursormon] mem (MB) res:37 virt:458
2014-08-14T11:36:21.355+0200 [clientcursormon] mapped (incl journal view):320
2014-08-14T11:36:21.355+0200 [clientcursormon] connections:1
```

Die zentrale Funktion, um systematisch in Collections zu suchen, heißt find, die mit verschiedenen Parametern aufgerufen werden kann. Ohne Parameter werden alle Elemente der betrachteten Collection ausgegeben. Das folgende Beispiel zeigt, dass jedes Dokument automatisch einen eigenen eindeutigen Schlüssel mit dem Attribut _id enthält. Die Ausgabe macht auch deutlich, dass bei der mehrfachen Nutzung der Variablen pkt1 einfach Kopien des jeweiligen Objekts angelegt werden, so dass Änderungen an pkt1 keine Auswirkungen auf die Daten in der Datenbank haben.

```
db.punkte.find();
```

```
> db.punkte.find();
{ "_id" : ObjectId("53ec82e3ac6feb03e7070258"), "id" : 1, "x" : 0, "y" : 0 }
{ "_id" : ObjectId("53ec834cac6feb03e7070259"), "id" : 2, "x" : 42, "y" : 42 }
{ "_id" : ObjectId("53ec8362ac6feb03e707025a"), "id" : 2, "x" : 43, "y" : 41 }
{ "_id" : ObjectId("53ec837cac6feb03e707025b"), "id" : 1, "x" : 0, "y" : 0, "z" : 1 }
{ "_id" : ObjectId("53ec837dac6feb03e707025c"), "id" : 1, "x" : 0, "y" : 0, "z" : 1 }
{ "_id" : ObjectId("53ec837dac6feb03e707025d"), "id" : 8, "x" : "Hai", "fisch" : "Fisch" }
>
```

12.3.4 Cursor

Die find()-Funktion liefert einen Cursor, ein Objekt, das schrittweise durchlaufen werden kann. Dabei wird die Funktion next() genutzt, um den Wert an der aktuellen Cursor-Position auszugeben und den Cursor einen Schritt weiter zu setzen. Mit der Funktion hasNext() kann abgefragt werden, ob es noch einen Wert gibt, der beim nächsten next()-Aufruf ausgegeben wird. Das folgende Beispiel zeigt auch, dass zur Ausgabe von JSON-Objekten die Funktion printjson() genutzt werden kann. Es ist zu beachten, dass die Groß- und Kleinschreibung bei Befehlen eine Rolle spielt.

```
var cur = db.punkte.find();
while (cur.hasNext()) {
```

Für einen Cursor gibt es die Funktion limit(anz), mit dem maximal eine Anzahl von anz Objekten als Cursor zurückgegeben wird. Weiterhin gibt es die Funktion skip(anz) für einen Cursor, der die ersten anz-Elemente überspringt. Die Anzahl aller Elemente eines Cursors wird mit der Funktion count() zurückgegeben.

```
db.punkte.find().count();
db.punkte.find().limit(2).skip(3);
db.punkte.find().skip(3).limit(2);
db.punkte.find().count();
```

```
> db.punkte.find().count();

db.punkte.find().limit(2).skip(3);
{ "_id" : ObjectId("53ec837cac6feb03e707025b"), "id" : 1, "x" : 0, "y" : 0, "z" : 1 }
{ "_id" : ObjectId("53ec837dac6feb03e707025c"), "id" : 1, "x" : 0, "y" : 0, "z" : 1 }
> db.punkte.find().skip(3).limit(2);
{ "_id" : ObjectId("53ec837cac6feb03e707025b"), "id" : 1, "x" : 0, "y" : 0, "z" : 1 }
{ "_id" : ObjectId("53ec837dac6feb03e707025c"), "id" : 1, "x" : 0, "y" : 0, "z" : 1 }
> db.punkte.find().count();
```

12.3.5 Find

Eine erste detailliertere Anfragemöglichkeit besteht darin, der Funktion find() ein JSON-Objekt zu übergeben. Es werden dann alle Elemente der Collection übergeben, die in allen Werten mit dem übergebenen Objekt übereinstimmen.

```
db.punkte.find( {id:1} );
   db.punkte.find( {id:1, z:1} );

db.punkte.find( {id:1} );
{ "_id" : ObjectId("53ec82e3ac6feb03e7070258"), "id" : 1, "x" : 0, "y" : 0 }
{ "_id" : ObjectId("53ec837cac6feb03e707025b"), "id" : 1, "x" : 0, "y" : 0, "z" : 1 }
{ "_id" : ObjectId("53ec837dac6feb03e707025c"), "id" : 1, "x" : 0, "y" : 0, "z" : 1 }
} db.punkte.find( {id:1, z:1} );
{ "_id" : ObjectId("53ec837cac6feb03e707025b"), "id" : 1, "x" : 0, "y" : 0, "z" : 1 }
{ "_id" : ObjectId("53ec837cac6feb03e707025c"), "id" : 1, "x" : 0, "y" : 0, "z" : 1 }
} "_id" : ObjectId("53ec837dac6feb03e707025c"), "id" : 1, "x" : 0, "y" : 0, "z" : 1 }
```

Detailliertere Anfragen können z. B. Größenvergleiche bei Werten enthalten. Dabei wird wieder ein JSON-Anfrage-Objekt übergeben, dass dann aber spezielle Attribute enthält. Konkret steht \$lt für "less than" (kleiner) oder \$gt für "greater than" (größer). Mit Gleichheit heißen die Attribute oder Vergleichsoperatoren \$lte und \$gte.

```
db.punkte.find({x: {$gt:1, $1t:43}});
> db.punkte.find({x: {$gt:1, $1t:43}});
{ "_id" : ObjectId("53ec834cac6feb03e7070259"), "id" : 2, "x" : 42, "y" : 42 }
>
```