

# Nutzerdokumentation SQLChecker

Wintersemester 2018/2019



Databases and Information Systems (DBIS)  
Johann Wolfgang Goethe-Universität Frankfurt am Main

# Inhaltsverzeichnis

<b>1</b>	<b>Übersicht</b>	<b>2</b>
<b>2</b>	<b>Quickstart</b>	<b>2</b>
2.1	Installation . . . . .	2
2.2	Benutzung . . . . .	2
2.3	Hinweise . . . . .	3
<b>3</b>	<b>Installation</b>	<b>4</b>
<b>4</b>	<b>Das Programm starten</b>	<b>4</b>
4.1	zu alte JRE Version . . . . .	5
<b>5</b>	<b>Die grafische Oberfläche</b>	<b>6</b>
5.1	Die Toolbar . . . . .	7
5.1.1	Datei . . . . .	7
5.1.2	Bearbeiten . . . . .	8
5.1.3	Ausführen . . . . .	8
5.1.4	Olat . . . . .	8
5.1.5	? . . . . .	8
5.2	Übung . . . . .	9
5.2.1	Aufgaben . . . . .	9
5.2.2	SQL . . . . .	9
5.2.3	Menüleiste . . . . .	9
5.3	Einstellung . . . . .	10
5.3.1	Datenbank . . . . .	10
5.3.2	Student . . . . .	11
5.4	Ausgabe . . . . .	12

# 1 Übersicht

Der *SQLChecker* ist ein Open-Source-Java-Programm, entwickelt als Unterstützung für die Datenbanken-Vorlesung der Goethe Universität Frankfurt am Main. Die Anwendung ermöglicht Studenten das Erstellen und Testen von SQL-Queries für vorgegebene Aufgaben. Mit Hilfe des Programms können die Statements einfach automatisiert getestet und ausgewertet werden. Der *SQLChecker* nutzt intern das Datenbanken Testing-Framework *DBFit*. Für die Verbindung mit der Datenbank ist *MySQL* oder *MariaDB* notwendig. Das Projekt wird auf *GitHub* zur Verfügung gestellt.

## 2 Quickstart

### 2.1 Installation

- Laden Sie sich die aktuelle Version des *SQLCheckers* von der Webseite von DB1.
- Stellen Sie sicher, dass sie mindestens *JRE 10* richtig installiert haben.  
3
- Starten Sie *MariaDB* oder *MySQL* und statten Sie einen Nutzer mit benötigten Rechten aus.
- Starten Sie ein Systemterminal und wechseln Sie in den Ordner des *SQLCheckers*.
- Führen Sie in diesem Terminal

```
$ java -jar SQLChecker-<Versionsnummer>.jar
```

aus, um den *SQLChecker* zu starten.

### 2.2 Benutzung

Sie brauchen für jedes Projekt eine SQLChecker-Template-Datei (\*.sqlt) und optional ein Reset-Skript (\*.sql). Diese werden pro Aufgabe zur Verfügung gestellt.

- Legen Sie ein neues Projekt an. Klicken Sie dazu auf den Reiter 📁 Datei und wählen sie 📄 Neu ...
- Während des Dialogs wählen Sie eine SQLChecker-Template-Datei aus und setzten den Pfad zu ihrem Projekt.

- Tragen Sie nun die Angaben Ihrer Person unter `Einstellung - Student` ein. Abgaben ohne diese Angaben werden mit Null Punkten bewertet.
- Im Reiter `Datenbank` unter `Einstellung` tragen Sie bitte unbedingt alle Angaben außer das Reset-Skript ein. Dieses ist optional.
- Im WS18/19 wird ausschließlich die Datenbank airport verwendet. Benutzen Sie diese.
- Wollen sie das Rücksetzen der Datenbank nutzen, tragen Sie auch den Pfad zu einem gültigen Reset-Skript ein.
- Nun können Sie Aufgaben im Reiter `Übung` auswählen und im SQL-Bereich SQL-Statements eintragen. Tragen Sie nur ein SQL-Statement pro Aufgabe ein.
- Sie können die Statements testen, indem Sie `Ausführen` klicken.
- Die Datenbank können sie über den entsprechenden Button zurücksetzen.
- Sind Sie mit der Bearbeitung der Aufgabe fertig, können Sie die Abgabe-Datei über `Olat - Exportieren ...` erzeugen.

### 2.3 Hinweise

- Sie können die Einstellung ihres aktuellen Projektes exportieren, indem Sie `Datei - Exportiere Konfigurationsdatei` anklicken. Die erstellte Datei endet auf \*.conf.  
Sie können diese Datei über `Datei - Importiere Konfigurationsdatei` wieder laden und ihre Einstellung wiederherstellen. Wenn Sie ein neues Projekt anlegen, kann automatisch ihre alte Konfiguration wieder geladen werden. Stellen Sie dazu sicher, dass sich die exportierte Konfigurationsdatei im selben oder dem direkten Eltern-Ordner des neuen Projektes befindet, wenn Sie das neue Projekt anlegen.
- Das Reset-Skript wird automatisch geladen, wenn es sich im selben Ordner des Projektes befindet. Das Reset-Skript muss dazu folgenden Namen tragen: `Name der Übung_reset.sql`. Standardmäßig sollte das mitgelieferte Reset-Skript den richtigen Namen tragen. Nennen Sie es also bitte nicht um.
- Benutzen Sie bitte ausschließlich UTF-8 gültige Zeichen.

- Ändern Sie die Abgabe-Datei nicht. Insbesondere ändern Sie nicht ihre Zeichenkodierung.
- Benutzen Sie pro einzelner Aufgabe nur ein SQL-Statement.
- Sie können ein Projekt direkt beim Programmaufruf laden, indem Sie  

```
$ java -jar SQLChecker-<Versionsnummer>.jar -s <Pfad>
```

aufrufen.
- Config-Dateien können Sie entsprechend über das Attribut `-c` laden.
- Falls Sie Anregungen oder Fragen haben, können Sie sich uns gerne direkt über das GitHub kontaktieren.

### 3 Installation

Nutzen Sie die im Rahmen der Vorlesung vorgestellte Version des *SQLCheckers*. Diese finden sie auf der Seite von DB1. Alternativ laden Sie die aktuelle Version des *SQLCheckers* von unserem GitHub. Der *SQLChecker* besteht aus einem einzelnen Java-Archiv (\*.jar).

*SQLChecker* wurde mit Java 10 entwickelt. Die aktuelle SQL-Checker-Version braucht zur Ausführung mindestens Java SE Runtime Environment 10 (JRE 10). Die zum Stand der Erstellung dieser Anleitung aktuelle Version Java 11 findet sich unter oracle.com. Um ihre derzeitige Standard-Java-Version festzustellen, führen Sie in einem Systemterminal

```
$ java --version
```

aus.

### 4 Das Programm starten

Um die grafische Oberfläche des *SQLCheckers* zu starten, öffnen Sie zunächst ein Terminal ihres Betriebssystems. Wechseln Sie in den Ordner des *SQLCheckers* und führen Sie folgendes aus:

```
$ java -jar SQLChecker-<Versionsnummer>.jar
```

Beachten Sie, dass die richtige Path-Umgebungsvariable ihres Betriebssystems für Java gesetzt sein muss. In der Regel wird diese bei der Installation von

Java automatisch erstellt. Sollte dies nicht der Fall sein, schauen Sie bitte auf diese Seite: [java.com/de/download/help/path.xml](http://java.com/de/download/help/path.xml).

Befindet sich die JAR-Datei in einem anderen Verzeichnis, muss der Pfad explizit angegeben werden:

```
$ java -jar path/SQLChecker-<Versionsnummer>.jar
```

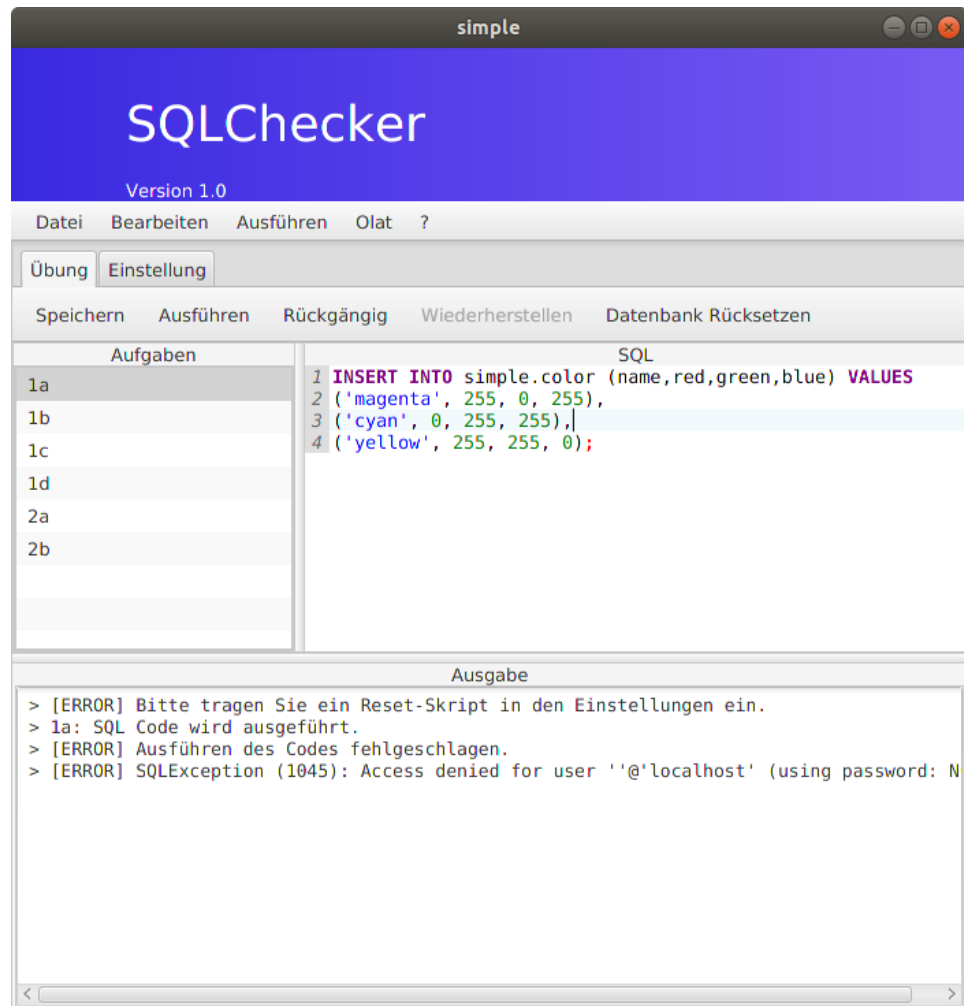
Die Darstellung von Pfaden ist abhängig von ihrem Betriebssystem. Passen Sie den Pfad entsprechend an.

#### 4.1 zu alte JRE Version

Die folgende oder ähnliche Fehlermeldung weist daraufhin, dass eine aktuellere Version des Runtime Environment benötigt wird (siehe 3).

```
Error: A JNI error has occurred, please check your installation
↳ and try again
Exception in thread "main"
↳ java.lang.UnsupportedClassVersionError:
↳ de/unifrankfurt/dbis/Runner has been compiled by a more
↳ recent version of the Java Runtime (class file version
↳ 54.0), this version of the Java Runtime only recognizes
↳ class file versions up to 52.0
```

## 5 Die grafische Oberfläche



Das Hauptfenster besteht aus folgenden Bereichen:

- 5.1. Die Toolbar
- 5.2. Der Übungsreiter
- 5.3. Der Einstellungsreiter
- 5.4. Die Ausgabe

## 5.1 Die Toolbar



Die Toolbar besteht aus diesen Menüpunkten:

5.1.1. Datei

5.1.2. Bearbeiten

5.1.3. Ausführen

5.1.4. Olat

5.1.5. ?

### 5.1.1 Datei

Im Menüpunkt Datei finden sich folgende Funktionen:

**Neu ...** Erstellen Sie ein neues Projekt. Sie werden zunächst aufgefordert eine Aufgaben-Template auszuwählen. Die Template-Dateien enden auf \*.sqlt und werden mit den Aufgaben mitgeliefert.

Wählen Sie als nächstes den Ort für ihr Projekt. Die Projekt-Datei endet auf \*.sqlc.

**Öffnen ...** Öffnen Sie ein vorhandenes Projekt. Projekt-Dateien enden auf \*.sqlc. Befindet sich im gleichen Ordner ein Reset-Script (\*.sql), wird es automatisch mitgeladen.

**Speichern** Speichern Sie das aktuelle Projekt am ausgewählten Projekt-Pfad.

**Speichern Unter ...** Wählen sie einen neuen Projekt-Ort. Die Projekt-Datei endet auf \*.sqlc. Das Projekt wird am neuen Ort gespeichert. Das Projekt am alten Ort bleibt bestehen.

**Lade Konfigurationsdatei ...** Wählen Sie eine Konfigurationsdatei. Konfigurationsdateien enden auf \*.conf. Die Konfigurationdatei wird geladen und die aktuelle Konfiguration wird überschrieben.

**Exportiere Konfigurationsdatei ...** Exportieren Sie die aktuelle Konfiguration um sie für ein anderes Projekt wieder zu verwenden. Wählen Sie den gewünschten Ort. Konfigurationsdateien enden auf \*.conf. Die Konfiguration wird automatisch als Teil ihres Projektes gespeichert.

**Schließen** Schließen Sie das aktuelle Projekt.



### 5.1.2 Bearbeiten

**Rückgängig** Die letzte ausgeführte Veränderung im aktiven SQL-Feld wird widerrufen.

**Wiederherstellen** Die letzte rückgesetzte Veränderung im aktiven SQL-Feld wird wiederhergestellt.

### 5.1.3 Ausführen

Um die folgenden Punkte ausführen zu können müssen die Datenbank-Informationen 5.3.1 richtig gesetzt worden sein und MySQL oder MariaDB müssen laufen.

**Datenbank Rücksetzen** Das Reset-Skript wird ausgeführt. Bei fehlerlosen Durchlauf ist die Datenbank wieder im Initial-Zustand. Im Ausgabe-Feld erhalten sie weitere Information zum Status des Rücksetzens.

**Aufgabe ausführen** Das SQL-Statement der aktiven Aufgabe wird ausgeführt. Das Ergebnis wird im Ausgabe-Feld angezeigt.

**Vollständig Ausführen** Die SQL-Statements aller Aufgaben werden nacheinander ausgeführt. Dies ist sinnvoll, wenn die Aufgaben voneinander abhängig sind.

### 5.1.4 Olat

**Exportieren ...** Erstellen Sie die Abgabe-Datei am gewünschten Ort. Laden Sie diese in Olat unverändert hoch.

### 5.1.5 ?

**Über** Zeige das Über-Fenster an.

## 5.2 Übung

The screenshot shows a web interface for an exercise. At the top, there are two tabs: 'Übung' (selected) and 'Einstellung'. Below these are five buttons: 'Speichern', 'Ausführen', 'Rückgängig', 'Wiederherstellen', and 'Datenbank Rücksetzen'. The main area is divided into two panels. The left panel, titled 'Aufgaben', contains a list of tasks: 1a, 1b, 1c, 1d, 2a, and 2b. The right panel, titled 'SQL', contains an SQL editor with the following text:

```
1 INSERT INTO simple.color (name,red,green,blue) VALUES
2 ('magenta', 255, 0, 255),
3 ('cyan', 0, 255, 255),
4 ('yellow', 255, 255, 0);
```

Hier werden die Lösungen der Aufgaben erstellt. Auf der linken Seite sehen Sie die Liste aller zu bearbeitenden Aufgaben 5.2.1 Auf der rechten Seite werden die zu erstellenden SQL-Statements im SQL-Feld 5.2.2 eingetragen. Am oberen Rand des Reiters sehen sie eine Reihe von Menüpunkten 5.2.3.

### 5.2.1 Aufgaben

In diesem Feld steht eine Liste aller Aufgaben. Klicken Sie die Aufgabe an, die Sie bearbeiten wollen. Die Lösung zu dieser Aufgabe tragen Sie im SQL-Feld ein. Sie sollen die Aufgaben von oben nach unten bearbeiten, da tiefere Aufgaben von früheren Aufgaben abhängig sein können.

### 5.2.2 SQL

Tragen Sie hier die Lösung für die im Aufgaben-Feld ausgewählte Aufgabe ein. Eine Lösung besteht in der Regel aus einem SQL-Statement. Sollten weitere erwünscht sein, wird dies explizit gefordert. Wird mehr als ein SQL-Statement eingetragen, obwohl nicht gefordert, wird die Aufgabe mit Null Punkten bewertet.

### 5.2.3 Menüleiste

**Speichern** Siehe 5.1.1.

**Ausführen** Siehe 5.1.3.

**Rückgängig** Siehe 5.1.2.

**Wiederherstellen** Siehe 5.1.2.

**Datenbank Rücksetzen** Siehe 5.1.3.

### 5.3 Einstellung

Der Reiter Einstellung besteht aus den beiden Punkten Datenbank und Student.

#### 5.3.1 Datenbank

The screenshot shows a software interface with two tabs: 'Übung' and 'Einstellung'. The 'Einstellung' tab is active and contains a section for 'Datenbank'. This section has several input fields: 'Name der SQL-Datenbank' with the value 'airport', 'Name des Nutzers' with 'db1', 'Passwort' which is masked with four dots, 'Host' with 'localhost', 'Port' with '3306', and 'Reset Skript' with the path '~/DB1/Blatt1/dump.sql'. To the right of the 'Reset Skript' field is a button labeled 'Öffnen ...'. At the bottom of the 'Einstellung' tab, there is a link to the 'Student' section.

Tragen Sie hier die Datenbank-Informationen ein. Achten Sie bitte darauf, dass der Nutzer angelegt wurde, das Passwort stimmt und der Nutzer über die nötigen Rechte der Datenbank verfügt.

**Name der SQL-Datenbank** Der Name der Datenbank auf der die SQL-Statements ausgeführt werden soll. Nachträgliches Wechsel der aktiven Datenbank mittels `USE name;` ist nicht möglich. Sollte die Datenbank noch nicht existieren, wird Ihnen angeboten, diese neu anzulegen.

**Name des Nutzers** Der Nutzernamen für die Datenbankverbindung.

**Passwort** Das Passwort des Nutzers.

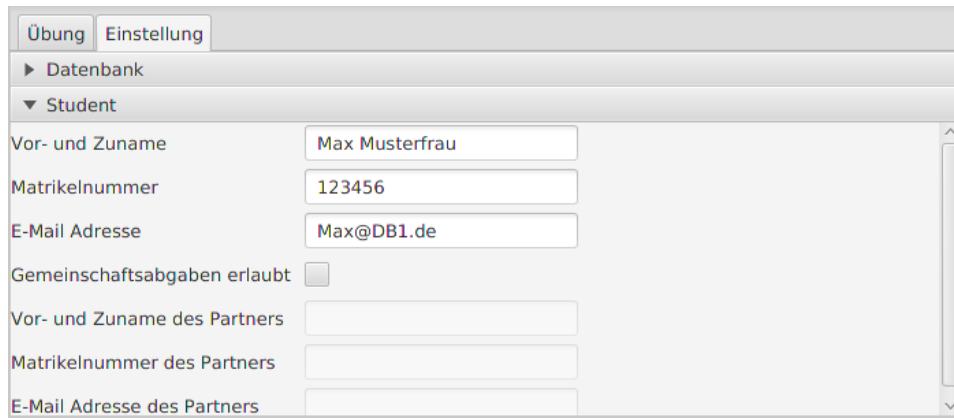
**Host** Der Host des Datenbank-Servers. Wird die Datenbank auf Ihrem Rechner ausgeführt, belassen Sie den Eintrag auf localhost.

**Port** Der Port der Datenbank. Der Standardwert ist 3306.

**Reset Skript** Der Pfad zum Reset-Skript. Das Reset-Skript besteht aus einer Reihe von SQL-Befehlen, welche die Datenbank wieder in den für

die Aufgabe vorgesehenen Initial-Zustand bringt. Eine Reset-Skript-PasswortsDatei endet auf \*.sql.

### 5.3.2 Student



Hier werden die Information zu den Autoren der Abgabe eingetragen. Die Angaben sind für die Abgabe zwingend erforderlich. Ohne diese Einträge wird die Abgabe mit Null Punkten bewertet.

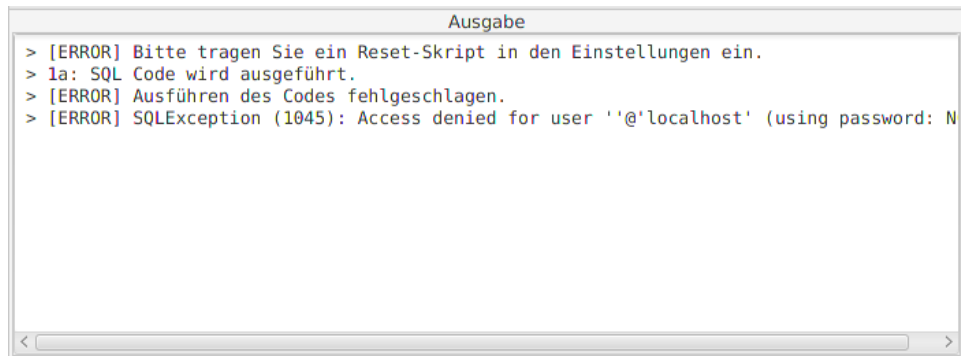
**Vor- und Zuname** Tragen Sie hier Ihren Namen ein.

**Matrikelnummer** Hier wird Ihre Matrikelnummer erwartet.

**E-Mail Adresse** Der Ort für Ihre E-Mail Adresse.

**Gemeinschaftsabgaben erlaubt** Setzen Sie den Haken, wenn Ihnen gestattet wurde Abgaben mit einem Partner zu bearbeiten. Vergessen Sie nicht die Angaben für Ihren Partner.

## 5.4 Ausgabe



```
> [ERROR] Bitte tragen Sie ein Reset-Skript in den Einstellungen ein.  
> 1a: SQL Code wird ausgeführt.  
> [ERROR] Ausführen des Codes fehlgeschlagen.  
> [ERROR] SQLException (1045): Access denied for user '@'localhost' (using password: N
```

Hier werden alle produzierten Informationen ausgegeben. Insbesondere sind dies Informationen von der Datenbank und Ergebnisse nachdem Sie ein SQL-Statement ausgeführt haben.

Sie können die aktuelle Ausgabe löschen, indem Sie [Rechtsklick - löschen](#) ausführen.