

Fragen, Antworten, Kommentare

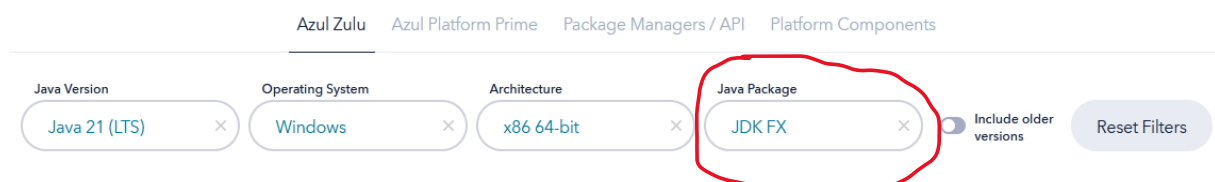
Hinweis: Generell macht die Formatierung und gegebenenfalls sinnvolle Umbenennung SQL-Anfragen deutlich lesbarer.

```
SELECT DISTINCT City1.Name, Country1.Name, Country2.Name
FROM Country Country1
  JOIN City City1
    ON (City1.Country = Country1.Code)
  JOIN City City2
    ON (City1.Name=City2.Name)
  JOIN Country Country2
    ON (City2.Country = Country2.Code)
WHERE (City1.Country!= City2.Country
      OR City1.Province != City2.Province)
      AND Country1.Name <= Country2.Name
ORDER BY City1.Name, Country1.Name, Country2.Name;
```

oder

```
SELECT DISTINCT City1.Name, Country1.Name, Country2.Name
FROM Country Country1, Country Country2, City City1, City City2
WHERE City1.Name = City2.Name
  AND City1.Country = Country1.Code
  AND City2.Country = Country2.Code
  AND (City1.Country!= City2.Country
      OR City1.Province != City2.Province)
  AND Country1.Name <= Country2.Name
ORDER BY City1.Name, Country1.Name, Country2.Name;
```

Hinweis: Gerade bei Aufgabe 26 ist die Nutzung der KleukerSEU sinnvoll, da in der benutzten Java-Version bereits JavaFX/OpenJFX installiert ist. Sollten Sie eine eigene SEU nutzen, beachten bzw. prüfen Sie, dass Java diese Bibliotheken bereits enthält, da anders als bei den meisten Bibliotheken die einfache Einbindung von jar-Dateien nicht ausreicht. Nutzen Sie sonst eine Java-Version mit OpenJFX, z. B. von <https://www.azul.com/downloads/#zulu> (Java Package JDK FX)



Frage: Ich habe eine Anfrage, die mit Derby läuft, aber im SQLChecker nicht.

Antwort: Dies kann passieren, da der SQLChecker nicht Derby sondern SQLite nutzt. Dies ist ein gutes Beispiel, dass SQL-Datenbanken sehr große Gemeinsamkeiten haben, sich aber in der Unterstützung von Sprachdetails unterscheiden. Wer mit SQLite etwas spielen möchte, kann z. B.

<http://kleuker.iui.hs-osnabrueck.de/querschnittlich/sqliteTool.html> nutzen. Laden Sie die Datenbank mondial_sqlite.db aus der SQLChecker-Version aus dem Hauptverzeichnis mit „Browse..“ in das

Browser-Tool. SQLite erlaubt z. B. auch Anfragen, die kritisch als falsch einzustufen sind, wie die folgenden Bilder zu SQLite und Derby zeigen.

SQLite Abfrage im Browser

Browse... mondial_sqlite.db

```
SELECT *
FROM Country
GROUP BY Name;
```

Ausführen

Vorherige Befehle auswählen

Name	Code	Capital	Province	Area	Population
Afghanistan	AFG	Kabul	Afghanistan	647500	26023100
Albania	AL	Tirana	Albania	28750	2821977
Algeria	DZ	Algiers	Algeria	2381740	37062820

Tabellenübersicht

- [Country](#)
- [City](#)
- [Province](#)
- [Economy](#)
- [Population](#)
- [Politics](#)
- [Language](#)
- [Religion](#)
- [EthnicGroup](#)
- [Countrypops](#)
- [Countryothername](#)
- [Countrylocalname](#)
- [Provpops](#)
- [Provinceothername](#)
- [Provincelocalname](#)
- [Citypops](#)
- [Cityothername](#)
- [Citylocalname](#)
- [Continent](#)

```
17 SELECT *
18 FROM Country
19 GROUP BY Name;
```

Messages

An error occurred when executing the SQL command:

```
SELECT *
FROM Country
GROUP BY Name
```

Column reference 'COUNTRY.CODE' is invalid, or is part of an invalid expression. For a SELECT list with a GROUP BY, the columns and expressions being selected may only contain valid grouping expressions and valid aggregate expressions.

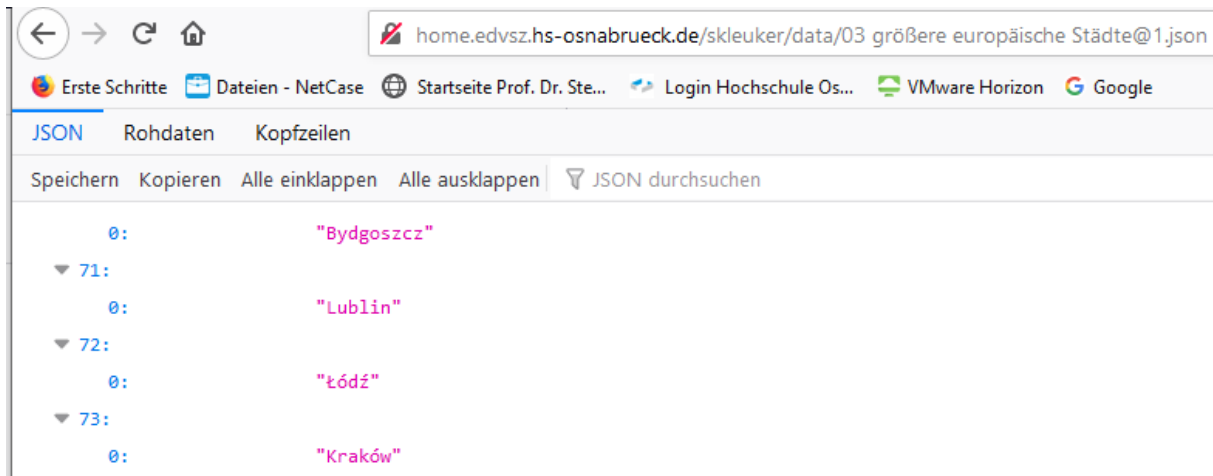
1 statement failed.

Sie können versuchen Ihr korrektes Ergebnis umzustrukturieren, so dass es auch in SQLite läuft, sie können die Anfrage aber auch so im Praktikum vorzeigen.

nochmal Frage: Der SQLChecker läuft auf meinem PC problemlos, aber auf meinem Laptop gab es Probleme

Der SQLChecker lädt die Aufgaben immer am Anfang einmal aus dem Internet, genauer dem Verzeichnis <http://home.edvsz.hs-osnabrueck.de/skleuer/data/>. Die Daten sind UTF8-codiert. Dies ist aber nicht die Windows-StandardEinstellung. Dies bedeutet, dass der SQLChecker ohne Zusatzangabe beim Schreiben von Dateien den Default-Zeichensatz von Windows nutzt, was zu Problemen führen kann. Sie haben damit seit dem Start korrupte Daten. Der Fehler ist bemerkenswert schwer zu lokalisieren, wie folgende Geschichte zeigt. Die Ergebnis-Datei wird

zunächst korrekt erzeugt und auf dem Server abgelegt, wie es mit "Łódź" im Datensatz 72 zu erkennen ist.



Auch selbst im Browser ausprobieren: <http://home.edvsz.hs-osnabrueck.de/skleuker/data/03%20gr%C3%B6%C3%9Fere%20europ%C3%A4ische%20St%C3%A4dte@1.json>

Der einfachste Workaround besteht darin, (1) dass data-Verzeichnis umzubenennen, um alte Lösungen zu behalten und dann (2) den SQLChecker neu mit der Ergänzung unten zu starten, um ein neues data-Verzeichnis anzulegen. Der Startaufruf in der bat-Datei muss wie folgt ergänzt werden.

```
java -Dfile.encoding=UTF-8 -jar SQLChecker.jar
```