



WS 2012

LV Informatik-I für Verkehrsingenieure

6.2 Datenbanken

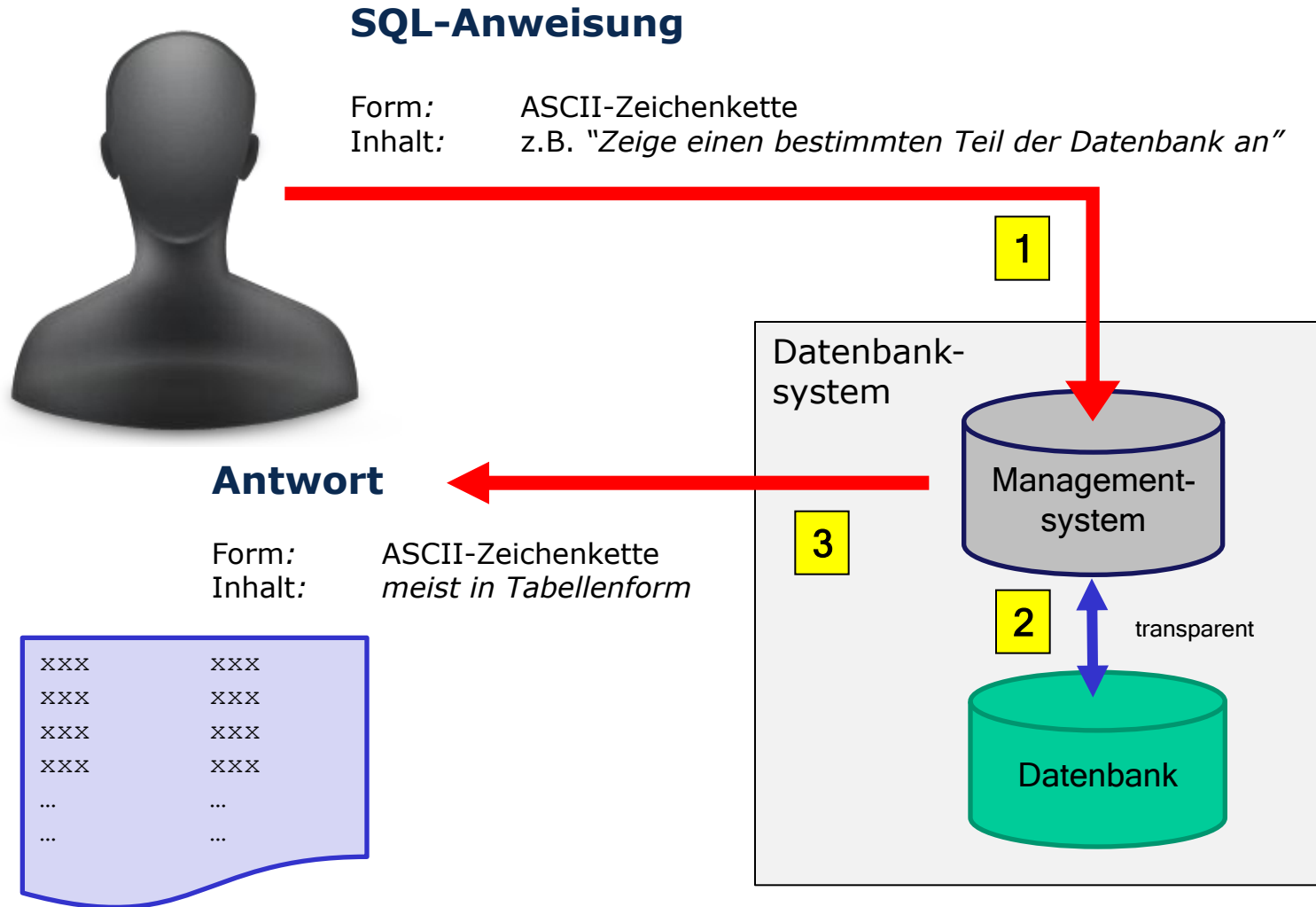
SQL - Einführung

Dr. rer.nat. D. Gütter

Mail: Dietbert.Guetter@tu-dresden.de

WWW: wwwpub.zih.tu-dresden.de/~guetter/

SQL - Sprache zur Datenbanknutzung



SQL (Structured Query Language)

Sprache zum Speichern, Bearbeiten und Abfragen von Daten in relationalen Datenbanken

- wird von vielen DBMS unterstützt, z.B. MySQL, MS-Access, ... Standardisierung aber unvollkommen (SQL-Dialekte)
- Klassen von Anweisungen zur

DDL (Data Definition Language) - Tabellendefinition

DCL (... Control ...) - Zugriffskontrolle

DML (... Manipulation ...) - Datenmanipulation

TCL (Transaction Control Language)

Absicherung der inneren Parallelarbeit vieler DML-Befehle („gleichzeitige“ Arbeit an mehreren Tabellen)

„alles oder nichts“ ausführen, sonst Inkonsistenz

Bezeichnungen / Notation

Auftreten

[] genau einmal oder keinmal

()+ mindestens einmal

- **Attribute** sind Elemente der Relationen (Spalten in der Tabellendarstellung)
- **Aliase** sind Alternativbezeichner für Attribute (z.B. um Tipparbeit bei sehr großen Abfragen einzusparen)
- **Quelle** sind Relationen als Relation gilt jede in einer Datenbank hinterlegte Relation, aber auch das Ergebnis einer Abfrage
- **Klauseln** sind Bedingungen an die auszuwählenden Elemente; sie schränken den Ergebnisraum ein und können Konstanten oder SFW-Blöcke sein.
- **Aggregatfunktionen** schränken den Ergebnisraum ein; sie ergänzen SQL um nicht in Klauseln darstellbare Bedingungen.

DDL: CREATE, (ALTER TABLE) DROP

Ohne Erklärung; nur zur Kenntnis nehmen, dass es das gibt.

Datenbanken und Relationen werden mittels `CREATE` erzeugt.

```
CREATE DATABASE Datenbank;  
  
CREATE TABLE Relation (  
    (Attribut Datentyp)+  
)
```

Löschen von Indizes, Relationen oder Datenbanken:

```
DROP INDEX Index ON Relation;           /* MS Access */  
  
ALTER TABLE Relation DROP INDEX Index; /* MySQL */  
  
DROP TABLE Relation;  
  
DROP DATABASE Datenbank;
```

DML - Data Manipulation Language

Verschiedene Blöcke

- INSERT Einfügen Spalte in Tabelle
- UPDATE Überschreiben von Attributen
- MERGE Mischen von Relationen
- DELETE Löschen von Tabellenzeile
- TRUNCATE leert Tabelle; Rücksetzen auf Standardwerte
- SFW („SELECT-FROM-WHERE“-Block)
liefert neue Relation aus einer Quell-Relation

Kernelement der Datenbankabfragen
→ in dieser LV Konzentration auf SFW-Blöcke

DML: INSERT

Einfügen von neuen Elementen in eine vorhandene Relation:

```
INSERT INTO Relation [(Attribut+)] VALUES ((Konstante+))+;
```

- Konstanten sind feste Werte,
z.B. 'Müller' (String), true (Boolean), 5 (Integer), ...
- Beim Konstrukt `(Attribut+) VALUES ((Konstante+))+` muss die Attributanzahl mit der Konstantenanzahl übereinstimmen.

z.B. `(m,n,p) VALUES ('a','b','c')`
 `(m,n,p) VALUES ('a','b','c'),('x','y','z')`

```
INSERT INTO Relation [(Attribut+)] SFW-Block;
```

SFW-Block liefert Relation, die bei Attribut-Gleichheit direkt in andere Relationen eingefügt wird.

DML: UPDATE

Aktualisieren vorhandener Elemente in einer vorhandenen Relation:

```
UPDATE Relation SET (Attribut=Ausdruck)+ [WHERE Klausel];
```

- Jedes Attribut (auch mehrere gleichzeitig) kann durch einen Ausdruck in Form von Konstanten oder einem SFV-Block neu beschrieben werden.
- Die `WHERE`-Klausel beschränkt die Aktualisierung auf entsprechende Elemente (beispielsweise `WHERE name='Dresden';`)
- Fehlt die `WHERE`-Klausel, werden **alle** (!) Elemente aktualisiert.

DML: DELETE, TRUNCATE

Löschen von (Zeilen-)Elementen aus einer Relation:

```
DELETE FROM Relation [WHERE Klausel];
```

Leeren einer Relation:

```
TRUNCATE Relation;
```

- Dies entspricht `DELETE FROM Relation;`
(also `DELETE ohne WHERE`)

DML: MERGE, JOIN

Ohne Erklärung; nur zur Kenntnis nehmen, dass es das gibt.

Fusion von Relationen:

```
MERGE INTO Relation USING Quelle ON Join-Klausel
WHEN MATCHED UPDATE SET (Attribut=Ausdruck)+
WHEN NOT MATCHED [BY TARGET] INSERT (Attribut+)
VALUES (Ausdruck+)
[WHEN NOT MATCHED BY SOURCE DELETE];
```

„SELECT-FROM-WHERE“-Block Auswahl einer Teilrelation aus einer Quell-Relation

```
SELECT [DISTINCT] Attribut [AS Alias]
FROM Quelle [AS Alias]
[WHERE Klausel]
[GROUP BY (Attribut)+ ]
[HAVING Aggregatfunktion]]
[ORDER BY (Attribut [ASC|DESC])+];
```

SQL-Demonstrationen

Beispiele basieren weitgehend auf

Geographiedatenbank „Mondial“

(<http://www.dbis.informatik.uni-goettingen.de/Mondial/>)

Mondial-**Abfragen** (nur lesend) über WWW (Uni Göttingen)

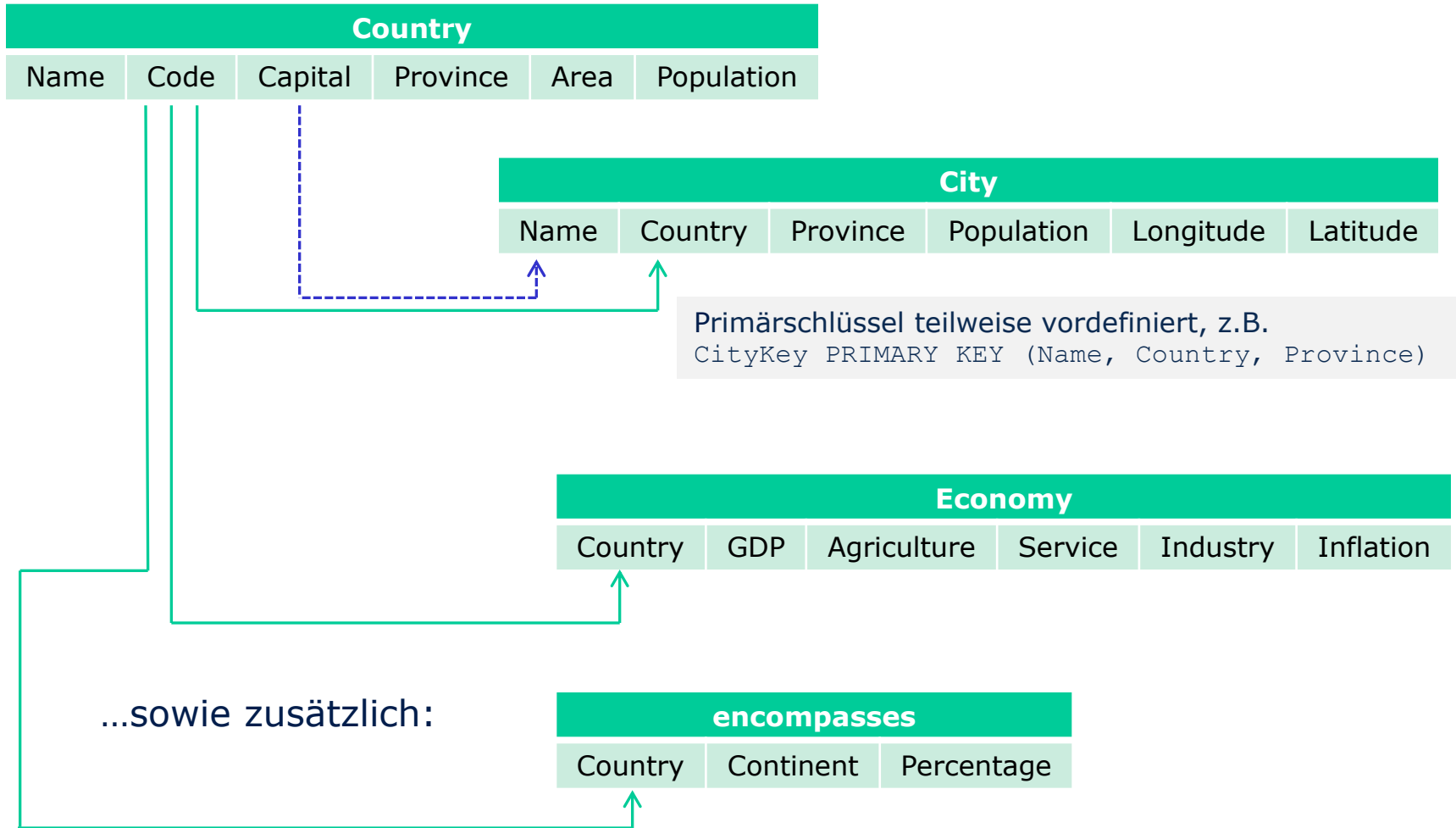
(<http://www.semwebtech.org/sqlfrontend/>)

Mondial-Datenbank (SQL-Version)

relationales Datenbanksystem

- enthält statistische Daten zu den Ländern der Welt auf Basis des Central Intelligence Agency World Factbook in vielen Relationen (Tabellen):
 - Country, City, Province, Continent, Borders,
 - Economy, Politics, Organization,
 - Population, Language, Religion, EthnicGroup,
 - Mountain, Desert, Island, Lake, Sea, River, ...
- in dieser Lehrveranstaltung Beschränkung auf wenige Tabellen
- frei zugängliche Didaktikversion, beschränkt auf Lesezugriffe, keine DDL- oder DCL-Anweisungen
DML-Anweisungen auf SFW-Blöcke eingeschränkt

Mondial – Datenbank



Mondial – Datenbankerzeugung mit DDL

```
CREATE TABLE Country (  
    Name VARCHAR2(35) NOT NULL UNIQUE,  
    Code VARCHAR2(4) CONSTRAINT CountryKey PRIMARY KEY,  
    Capital VARCHAR2(35),  
    Province VARCHAR2(35),  
    Area NUMBER CONSTRAINT CountryArea CHECK (Area >= 0),  
    Population NUMBER CONSTRAINT CountryPop CHECK (Population >= 0));
```

```
CREATE TABLE City (  
    Name VARCHAR2(35),  
    Country VARCHAR2(4),  
    Province VARCHAR2(35),  
    Population NUMBER CONSTRAINT CityPop CHECK (Population >= 0),  
    Longitude NUMBER CONSTRAINT CityLon  
        CHECK ((Longitude >= -180) AND (Longitude <= 180)) ,  
    Latitude NUMBER CONSTRAINT CityLat  
        CHECK ((Latitude >= -90) AND (Latitude <= 90)) ,  
    CONSTRAINT CityKey PRIMARY KEY (Name, Country, Province));
```

```
CREATE TABLE Economy (  
    Country VARCHAR2(4) CONSTRAINT EconomyKey PRIMARY KEY,  
    GDP NUMBER CONSTRAINT EconomyGDP CHECK (GDP >= 0),  
    Agriculture NUMBER,  
    Service NUMBER,  
    Industry NUMBER,  
    Inflation NUMBER);
```

...

Mondial-DB: Beispielaufgaben zu SELECT

Anzeige der (vollständigen) Tabelle Country

Eingabe in die Mondial-DB
über die Webseite

<http://www.semwebtech.org/sqlfrontend/>

SQL-Anweisung

```
SELECT * FROM Country;
```

Query Results - Mozilla Firefox

www.semwebtech.org/sqlfrontend/sqlfrontend?query-text=SELECT+*+From+Country%3B%0D%0A%0D

Homepage of Institute for Informatics, University of Göttingen

Playground for SQL Queries

With this form, you can state SQL queries against the [Mondial](#) Database. The database is used in the lectures

- [Databases](#)
- [Introduction to Databases and Database Programming in SQL/Oracle](#)

SELECT * From Country;

send reset

Results: 241

NAME	CODE	CAPITAL	PROVINCE	AREA	POPULATION
Albania	AL	Tirane	Albania	28750	3249136
Greece	GR	Athens	Attiki	131940	10538594
Macedonia	MK	Skopje	Macedonia	25333	2104035
Serbia	SRB	Belgrade	Serbia	77474	7379339
Montenegro	MNE	Podgorica	Montenegro	14026	672180
Kosovo	KOS	Pristina	Kosovo	10887	1804838
Andorra	AND	Andorra la Vella	Andorra	450	72766
France	F	Paris	Ile de France	547030	58317450
Spain	E	Madrid	Madrid	504750	39181114
Austria	A	Vienna	Vienna-Wien	83850	8023244
Czech Republic	CZ	Prague	Praha	78703	10321120
Germany	D	Berlin	Berlin	356910	83536115

Aufgabe (2a): Selektion nach Spalten

eingeschränkte Anzeige der Tabelle Country
(nur Stadtname und Einwohneranzahl)

SQL-Anweisung

```
SELECT Name, Population  
FROM Country;
```

Query Results - Mozilla Firefox

www.semwebtech.org/sqlfrontend/sqlfrontend?query-text=SELECT+Name%2C+Population+From+Cou

Homepage of Institute for Informatics, University of Göttingen

Playground for SQL Queries

With this form, you can state SQL queries against the [Mondial](#) Database. The database is used in the lectures

- [Databases](#)
- [Introduction to Databases and Database Programming in SQL-Oracle](#)

SELECT Name, Population From Country;

send reset

Results: 241

NAME	POPULATION
Albania	3249136
Greece	10538594
Macedonia	2104035
Serbia	7379339
Montenegro	672180
Kosovo	1804838
Andorra	72766
France	58317450
Spain	39181114
Austria	8023244
Czech Republic	10321120
Germany	83536115

Aufgabe (2b): Mehrfacheinträge eliminieren

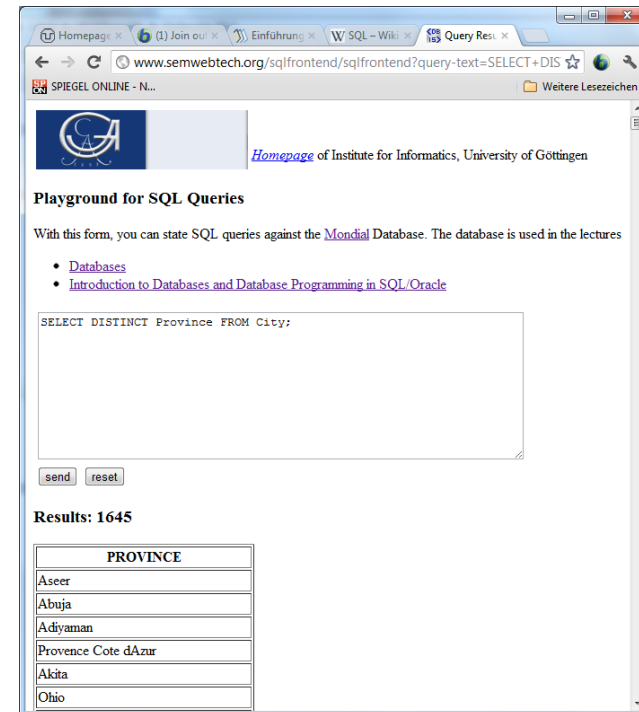
Für jede Stadt die Provinz anzeigen

```
SELECT Province FROM City;
```

(Mehrfach-Einträge möglich,
z.B. Mexico, Estado de)

Entfernen der Mehrfacheinträge

```
SELECT DISTINCT Province FROM City;
```



Homepage of Institute for Informatics, University of Göttingen

Playground for SQL Queries

With this form, you can state SQL queries against the [Mondial](#) Database. The database is used in the lectures

- [Databases](#)
- [Introduction to Databases and Database Programming in SQL/Oracle](#)

```
SELECT DISTINCT Province FROM City;
```

send reset

Results: 1645

PROVINCE
Aseer
Abuja
Adiyaman
Provence Cote dAzur
Akita
Ohio

Aufgabe (3a): Selektion nach Bedingungen

eingeschränkte Anzeige der Tabelle Country
(Länder mit mehr als 55 Millionen Einwohnern)

SQL-Anweisung

SFW-Block

```
SELECT Name, Population  
FROM Country  
WHERE Population > 55000000;
```



Registriert sind in der Datenbank
22 Länder mit
mehr als 55 Mio. Einwohnern.

Query Results

www.semwebtech.org/sqlfrontend/sqlfrontend?query-text=SELECT%09N

SPIEGEL ONLINE - N...

Diese Seite ist auf Englisch Soll sie übersetzt wer... Übersetzen Nein Optionen

Homepage of Institute for Informatics, University of Göttingen

Playground for SQL Queries

With this form, you can state SQL queries against the [Mondial](#) Database. The database is used in the lectures

- Databases
- Introduction to Databases and Database Programming in SQL/Oracle

```
SELECT Name, Population  
FROM Country  
WHERE Population > 55000000;
```

send reset

Results: 22

NAME	POPULATION
France	58317450
Germany	83536115
Italy	57460274
Russia	148178487
Turkey	62484478

Zeilenanzahl geringer (22)

Aufgabe (3b): Selektion nach Bedingungen

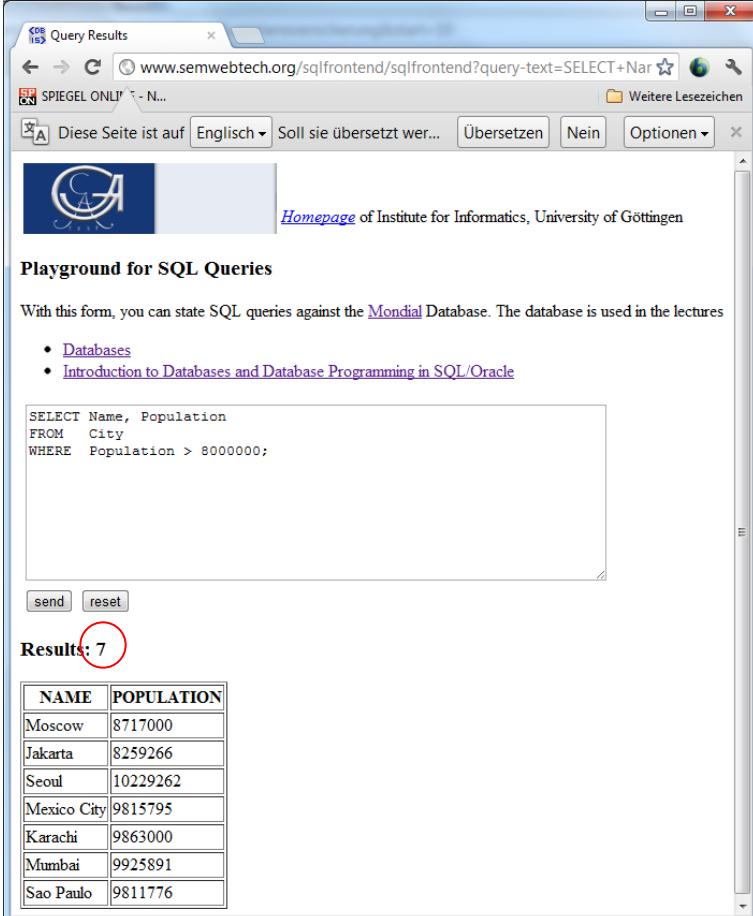
eingeschränkte Anzeige der Tabelle City
(Städte mit mehr als 8 Millionen Einwohnern)

SQL-Anweisung

```
SELECT Name, Population  
FROM City  
WHERE Population > 8000000;
```



Registriert sind in der Datenbank
7 Städte mit
mehr als 8 Mio. Einwohnern.



The screenshot shows a web browser window with the URL `www.semwebtech.org/sqlfrontend/sqlfrontend?query-text=SELECT+Nar`. The page title is "Query Results". The browser's address bar shows the URL. The page content includes a logo for the Institute for Informatics, University of Göttingen, and a section titled "Playground for SQL Queries". Below this, there is a text area containing the SQL query: `SELECT Name, Population
FROM City
WHERE Population > 8000000;`. Below the text area are "send" and "reset" buttons. The results are displayed as "Results: 7", with the number "7" circled in red. Below the results is a table with two columns: "NAME" and "POPULATION".

NAME	POPULATION
Moscow	8717000
Jakarta	8259266
Seoul	10229262
Mexico City	9815795
Karachi	9863000
Mumbai	9925891
Sao Paulo	9811776

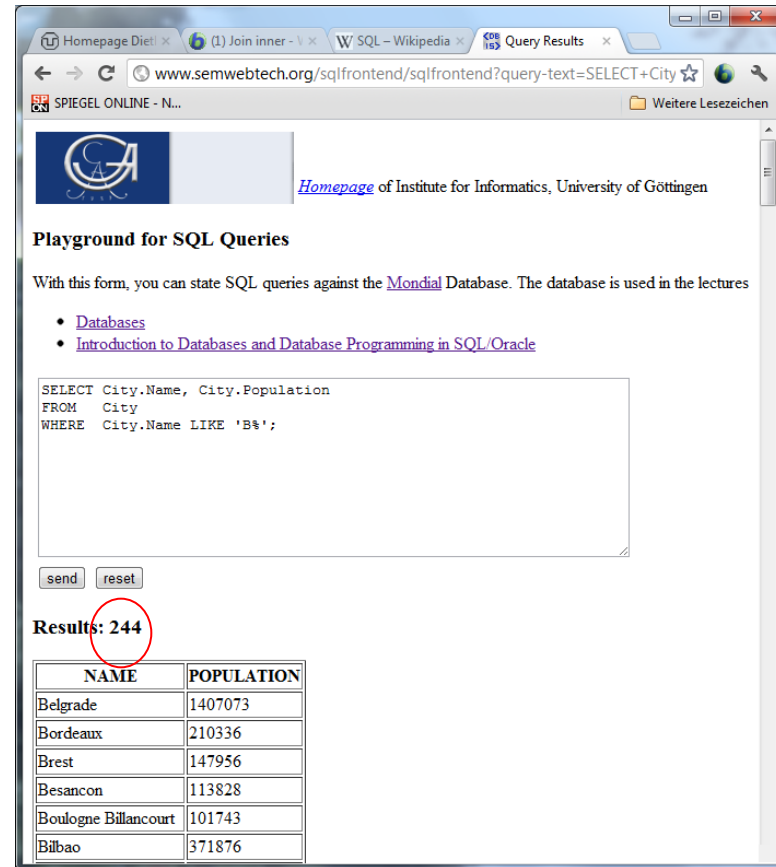
Aufgabe (3c): Selektion nach Bedingungen

eingeschränkte Anzeige der Tabelle City
(Städte mit Anfangsbuchstaben "B")

SQL-Anweisung

```
SELECT Name, Population
FROM City
WHERE Name LIKE 'B%';
```

→
244 Städte
beginnen mit Buchstabe „B“



www.semwebtech.org/sqlfrontend/sqlfrontend?query-text=SELECT+City

Homepage of Institute for Informatics, University of Göttingen

Playground for SQL Queries

With this form, you can state SQL queries against the [Mondial](#) Database. The database is used in the lectures

- [Databases](#)
- [Introduction to Databases and Database Programming in SQL/Oracle](#)

```
SELECT City.Name, City.Population
FROM City
WHERE City.Name LIKE 'B%';
```

send reset

Results: 244

NAME	POPULATION
Belgrade	1407073
Bordeaux	210336
Brest	147956
Besancon	113828
Boulogne Billancourt	101743
Bilbao	371876

Aufgabe (4): Aggregatfunktionen

Aggregatfunktionen können Attributwerte auswerten

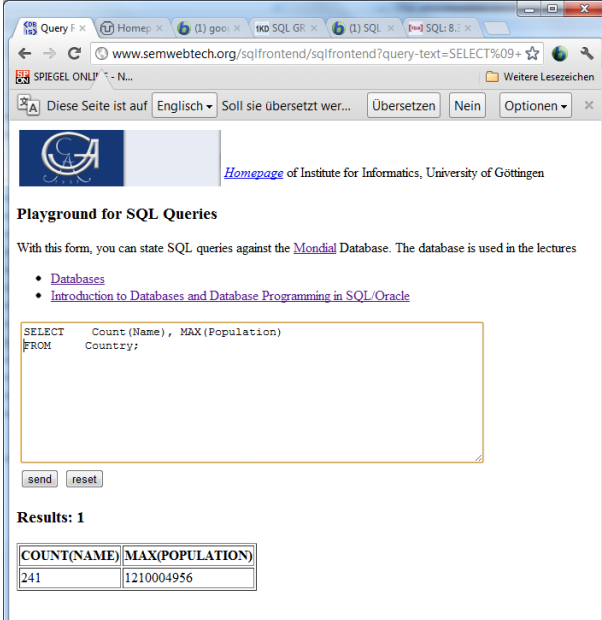
- MAX/MIN (Maximum/Minimum aller Werte)
- COUNT (Anzahl aller Werte)
- SUM (Summe aller Werte)
- AVG (Mittelwert aller Werte)
- ...

z.B. Anzeige Länderanzahl und Einwohneranzahl des größten Landes aus Tabelle "Country"

SQL-Anweisung

```
SELECT COUNT(Name), MAX(Population)
FROM Country;
```

Außer den Aggregatfunktionen besitzt SQL weitere Operationen für Addition, Subtraktion, Zeichenkettenoperationen, ... (in dieser LV nicht weiter behandelt).



Playground for SQL Queries

With this form, you can state SQL queries against the [Mondial](#) Database. The database is used in the lectures

- [Databases](#)
- [Introduction to Databases and Database Programming in SQL/Oracle](#)

```
SELECT Count (Name), MAX (Population)
FROM Country;
```

send reset

Results: 1

COUNT(NAME)	MAX(POPULATION)
241	1210004956

Verbundrelationen

Tabellengröße der Tabellen "Country" bzw. "City"

SELECT * From Country; → 241 Zeilen

SELECT * From City; → 3261 Zeilen

Verbund-Tabelle von "Country" und "City"

„Country“ Zeile 1	„City“ Zeile 1
„Country“ Zeile 1	„City“ Zeile 2
„Country“ Zeile 1	...
„Country“ Zeile 1	„City“ Zeile 3261
„Country“ Zeile 2	„City“ Zeile 1
„Country“ Zeile 2	...
„Country“ Zeile 2	„City“ Zeile 3261
...	...
„Country“ Zeile 241	„City“ Zeile 1
„Country“ Zeile 241	...

241*3261 Zeilen

(Kartesisches Produkt)

Verbundoperationen (JOIN)

Hintereinanderausführung

- Bildung der Verbundrelation zweier Relationen
- und deren Reduzierung durch Selektion

Selektionsbedingungen

z.B. Vergleiche von Attributen ($>$, $<$, $=$, \neq)

Effizienz von konkreten Datenbanksysteme

- möglichst nicht die gesamte Verbundrelation intern zwischenspeichern
- Einsparung von Speicherplatz und Rechenzeit

JOIN – Grobübersicht: Anweisungsvarianten

Syntax SELECT <Attributenliste> FROM <Hauptrelation>
 [<join-typ>] JOIN <verknüpfte Relation>
 ON <Bedingung>

In einer SELECT-Anweisung können mehrere JOINS auftreten (hintereinander oder geschachtelt)

JOIN-Typen

- INNER JOIN Ergebnistabelle ist der Teil des Tabellenverbundes (Kreuzprodukt), auf den die Bedingung zutrifft
- OUTER JOIN Ergebnistabelle enthält auch Datensätze, für die eine Vergleichsbedingung nicht erfüllt ist.

in SQL sind verschiedene Varianten von OUTER JOIN definiert.

Auswerten mehrerer Tabellen (5a)

Tabellengröße der Verbund-Tabelle von "Country" und "City"

SELECT * From Country, City; → 785901 Zeilen (= 241*3261)

Homepage of Institute for Informatics, University of Göttingen

Playground for SQL Queries

With this form, you can state SQL queries against the [Mondial](#) Database. The database is used in the lectures

- [Databases](#)
- [Introduction to Databases and Database Programming in SQL/Oracle](#)

```
SELECT * FROM Country, City;
```

send reset

Results: 785901

NAME	CODE	CAPITAL	PROVINCE	AREA	POPULATION	NAME	COUNTRY	PROVINCE	POPULATION	LONGITUDE	LATITUDE	ELEVATION
Albania	AL	Tirane	Albania	28750	3249136	Tirane	AL	Albania	192000	19.8	41.3	null
Albania	AL	Tirane	Albania	28750	3249136	Shkoder	AL	Albania	62000	19.2	42.2	null
Albania	AL	Tirane	Albania	28750	3249136	Durres	AL	Albania	60000	19.3	41.3	null
Albania	AL	Tirane	Albania	28750	3249136	Vlore	AL	Albania	56000	19.3	40.3	null

Aufgabe (5b): Auswerten mehrerer Tabellen

Gleiche Anfrage wie bei (3b), aber über Verknüpfung von **Country** und **City**

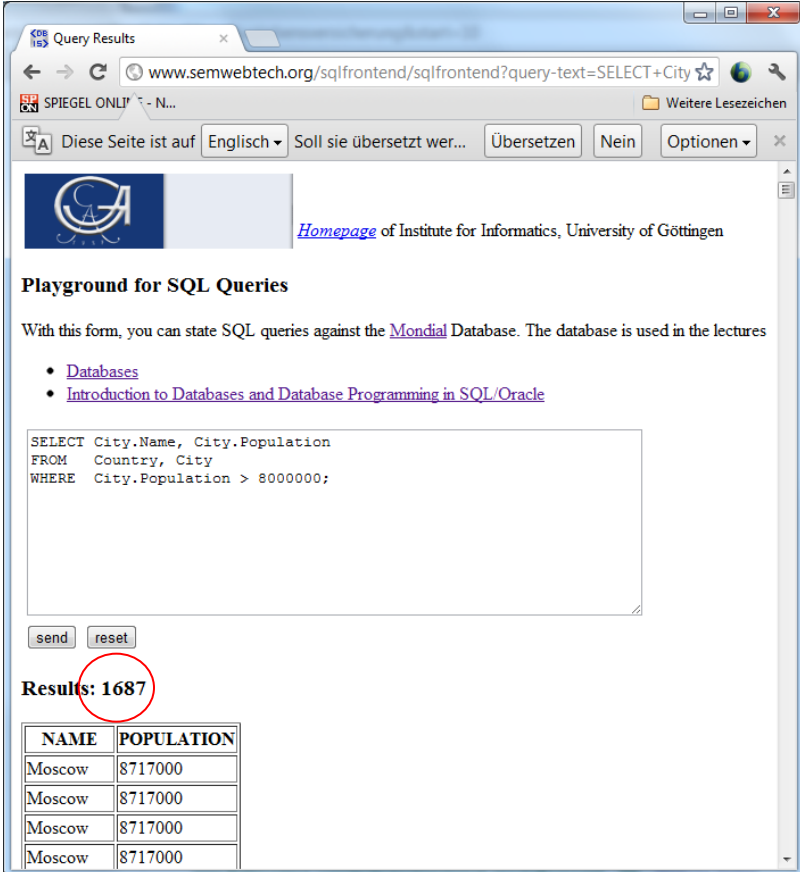
„Name“ und „Population“
kommen in beiden Relationen vor!

Deshalb Angabe
von Relation.Attribut erforderlich

```
SELECT City.Name, City.Population
FROM Country, City
WHERE City.Population > 8000000;
```

Anweisung ist sinnlos !

Für alle 7 Acht-Mio-Städte existieren
in Ergebnistabelle je 241 Zeilen;
insgesamt 1687 Zeilen.



Query Results

www.semwebtech.org/sqlfrontend/sqlfrontend?query-text=SELECT+City

SPIEGEL ONLINE - N...

Diese Seite ist auf Englisch Soll sie übersetzt wer... Übersetzen Nein Optionen

Homepage of Institute for Informatics, University of Göttingen

Playground for SQL Queries

With this form, you can state SQL queries against the [Mondial](#) Database. The database is used in the lectures

- [Databases](#)
- [Introduction to Databases and Database Programming in SQL/Oracle](#)

```
SELECT City.Name, City.Population
FROM Country, City
WHERE City.Population > 8000000;
```

send reset

Results: 1687

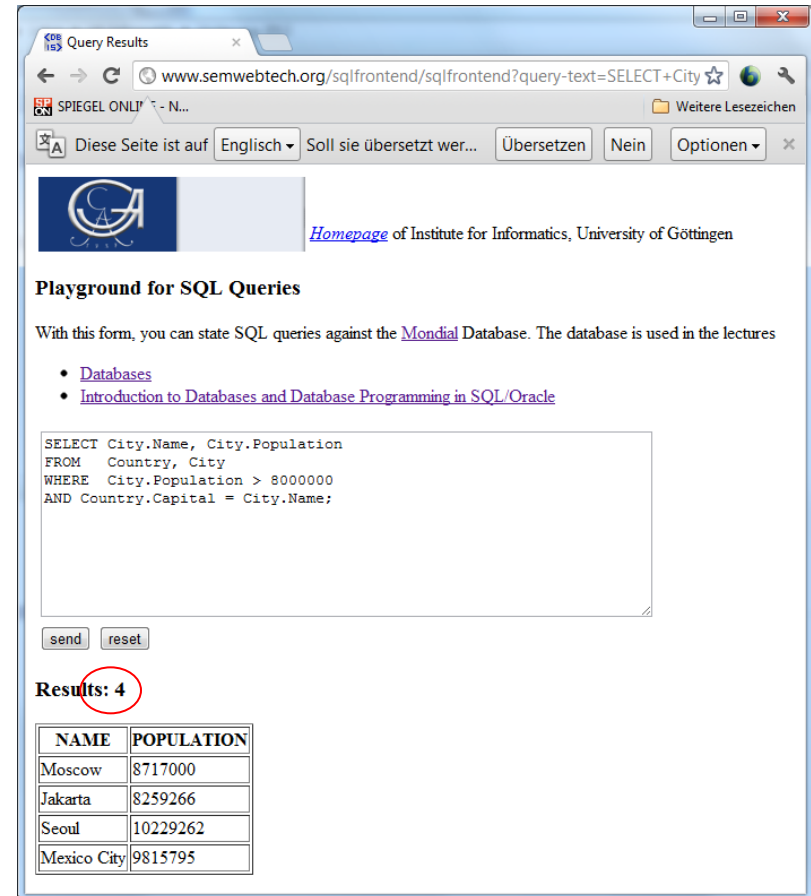
NAME	POPULATION
Moscow	8717000
Moscow	8717000
Moscow	8717000
Moscow	8717000

Aufgabe (5c): Sinnvolle Untermengen

Gleiche Anfrage wie bei (5b), aber mit Einschränkungsbedingung

Nur Hauptstädte berücksichtigen !

```
SELECT City.Name, City.Population
FROM Country, City
WHERE City.Population > 8000000
AND Country.Capital = City.Name;
```



The screenshot shows a web browser window titled "Query Results" with the URL `www.semwebtech.org/sqlfrontend/sqlfrontend?query-text=SELECT+City`. The page content includes a logo for the Institute for Informatics, University of Göttingen, and a section titled "Playground for SQL Queries". Below this, there is a text area containing the SQL query: `SELECT City.Name, City.Population FROM Country, City WHERE City.Population > 8000000 AND Country.Capital = City.Name;`. Below the text area are "send" and "reset" buttons. The results are displayed as a table with 4 rows, with the number "4" circled in red above the table.

Results: 4

NAME	POPULATION
Moscow	8717000
Jakarta	8259266
Seoul	10229262
Mexico City	9815795

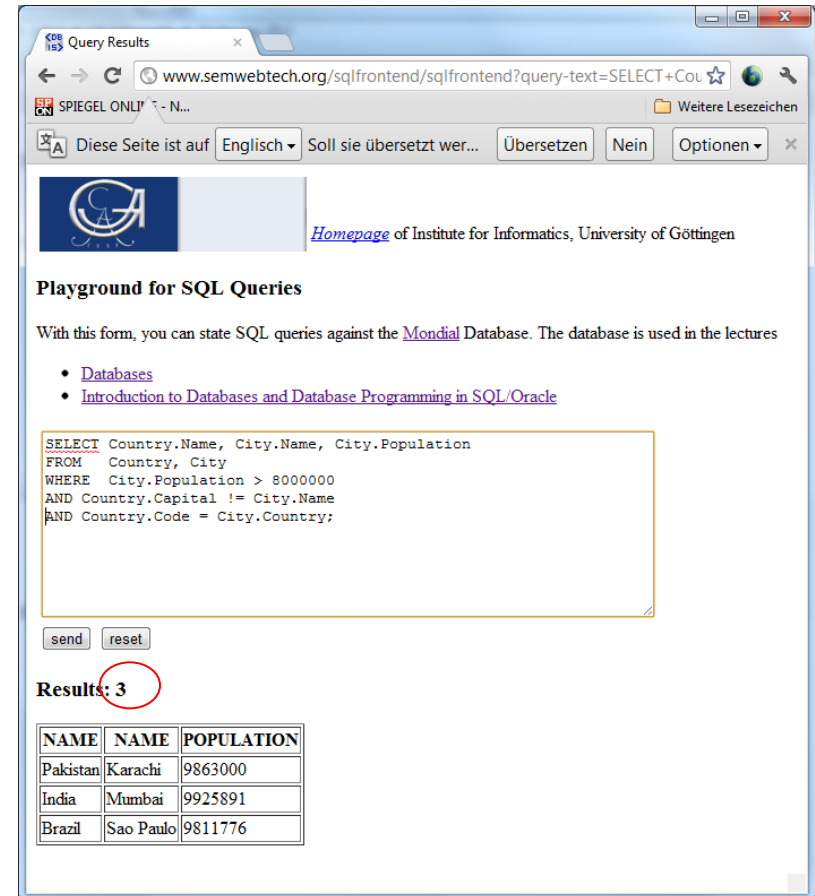
Aufgabe (5d): Sinnvolle Untermengen

Gleiche Anfrage wie bei (5b), aber mit Einschränkungsbedingung

Nur Nichthauptstädte berücksichtigen !

```
SELECT City.Name, City.Population
FROM Country, City
WHERE City.Population > 8000000
AND Country.Capital != City.Name
AND Country.Code = City.Country;
```

Was käme heraus,
wenn die zweite AND-Bedingung nicht existiert ?



Query Results

www.semwebtech.org/sqlfrontend/sqlfrontend?query-text=SELECT+Cou

SPIEGEL ONLINE

Diese Seite ist auf Englisch Soll sie übersetzt wer... Übersetzen Nein Optionen

GA
Homepage of Institute for Informatics, University of Göttingen

Playground for SQL Queries

With this form, you can state SQL queries against the [Mondial](#) Database. The database is used in the lectures

- [Databases](#)
- [Introduction to Databases and Database Programming in SQL/Oracle](#)

```
SELECT Country.Name, City.Name, City.Population
FROM Country, City
WHERE City.Population > 8000000
AND Country.Capital != City.Name
AND Country.Code = City.Country;
```

send reset

Results: 3

NAME	NAME	POPULATION
Pakistan	Karachi	9863000
India	Mumbai	9925891
Brazil	Sao Paulo	9811776

Anzeige ordnen durch ORDER BY

Aufgabe

Anzeige von "Land", "Hauptstadt" und "Einwohneranzahl"
mit folgender Darstellung:

Landesnamen alphabetisch aufsteigend und Bevölkerungsanzahl absteigend

Sortieranweisung ORDER BY .

```
SELECT Country.Name, City.Name, City.Population
FROM Country, City
WHERE City.Population > 8000000
      AND Country.Capital = City.name
```

} Kern des
SFW-Blocks

```
ORDER BY Country.Name ASC, City.Population DESC;
```

} Sortier-
anweisung

→ Ordne Ergebnis aufsteigend nach dem Attribut „Name“ aus „Country“
und absteigend nach dem Attribut „Population“ aus „City“.

Aliasnamen für Tabellen und Attribute

Gleiche Aufgabe (5c) unter Nutzung von Aliasnamen

- T1 für Tabelle "Country"
- T2 für Tabelle "City"
- Attributname "Name" in T1 durch Zeichenkette "Land" ersetzen
- Hauptstadt
- Einwohner

```
SELECT T1.Name AS Land,  
       T2.Name AS Hauptstadt,  
       T2.Population AS Einwohner  
  
FROM   Country T1, City T2  
  
WHERE  T2.Population > 8000000  
AND    T1.Capital = T2.Name;
```

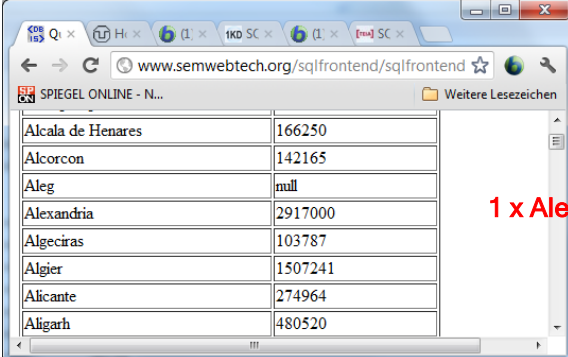
Gruppierung durch GROUP BY

Falls in SELECT-Anweisung
Angabe von Spalten **und** Aggregationsfunktion(en)
--> Auswahl von Gruppierungsspalte(n)

Beispiel: Anzeige aller Städtenamen aus Tabelle "City";
im Falle gleichnamiger Städte nur größte Stadt auswählen

SQL-Anweisung

```
SELECT Name, MAX(Population)
FROM City
GROUP BY Name
ORDER BY Name ASC;
```

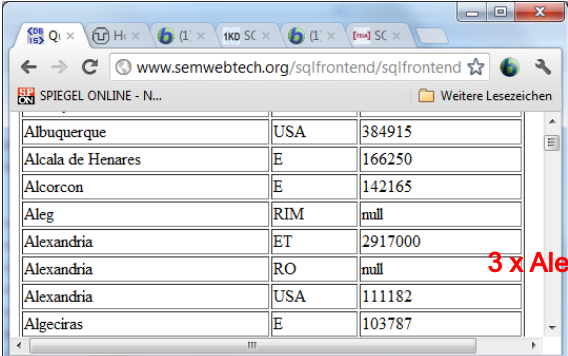


Alcala de Henares	166250
Alcorcon	142165
Aleg	null
Alexandria	2917000
Algeciras	103787
Algier	1507241
Alicante	274964
Aligarh	480520

1 x Alexandria

aber bei Gruppierung nach Name und Land

```
SELECT Name, Country, MAX(Population)
FROM City
GROUP BY Name, Country
ORDER BY Name ASC;
```



Albuquerque	USA	384915
Alcala de Henares	E	166250
Alcorcon	E	142165
Aleg	RIM	null
Alexandria	ET	2917000
Alexandria	RO	null
Alexandria	USA	111182
Algeciras	E	103787

3 x Alexandria

Aufgabe (5d): Notation mit INNER JOIN

Gleiche Anfrage wie bei (5b), aber mit Einschränkungsbedingung

Nur Nichthauptstädte berücksichtigen !

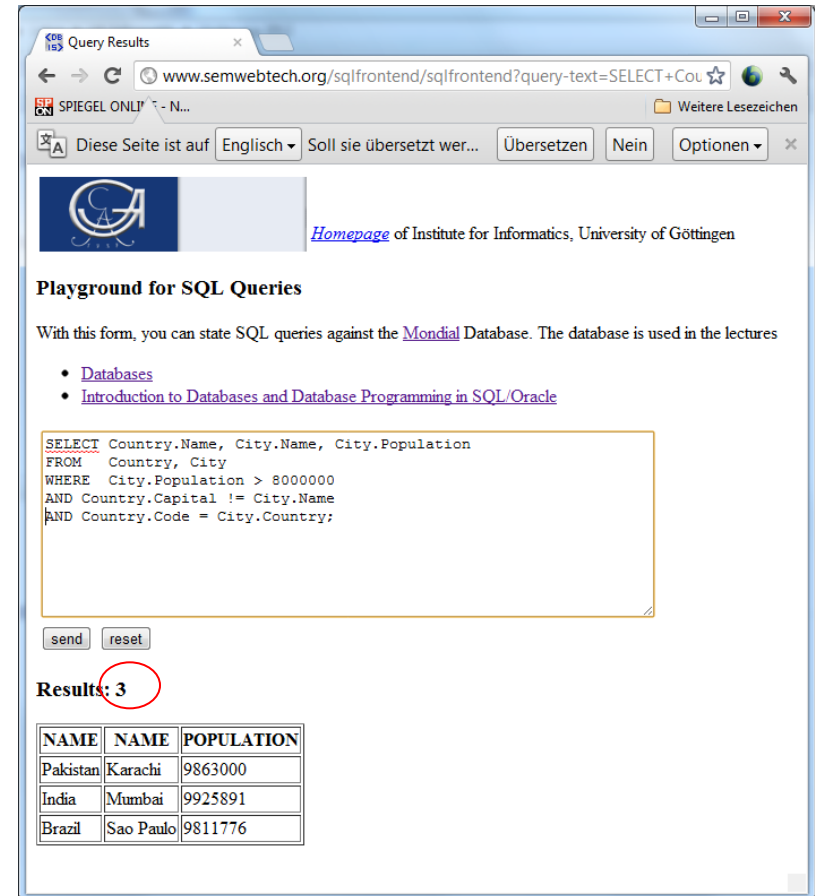
```
SELECT City.Name, City.Population
```

```
FROM Country INNER JOIN City
```

```
ON City.Population > 8000000
```

```
AND Country.Capital != City.Name
```

```
AND Country.Code = City.Country;
```



The screenshot shows a web browser window titled "Query Results" with the URL www.semwebtech.org/sqlfrontend/sqlfrontend?query-text=SELECT+Cou. The page content includes a logo for the Institute for Informatics, University of Göttingen, and a section titled "Playground for SQL Queries". Below this, there is a text area containing the following SQL query:

```
SELECT Country.Name, City.Name, City.Population
FROM Country, City
WHERE City.Population > 8000000
AND Country.Capital != City.Name
AND Country.Code = City.Country;
```

Below the query text are "send" and "reset" buttons. The results are displayed as "Results: 3" (circled in red) and a table with the following data:

NAME	NAME	POPULATION
Pakistan	Karachi	9863000
India	Mumbai	9925891
Brazil	Sao Paulo	9811776

Komplexere JOIN-Aufgaben

sprengen den Rahmen dieser Einführung

Literaturhinweis für freiwillige Weiterbildung

Dr. B. Keller: Lehrveranstaltung „Datenbanken (Lehramt)“

→ <http://wwwdb.inf.tu-dresden.de/lectures/ws-2011-2012/datenbanken-fuer-lehramt/>

SQL-Details

→ http://de.wikibooks.org/wiki/Einführung_in_SQL:_Ein_Einstieg

Frei verfügbare Tools

„H2 Database Engine“ (HSQLDB)

- in Java implementiert,
- besitzt WWW-Interface,
- Mondial steht als in-Memory-Datenbank für H2 zur Verfügung.

→ <http://www.h2database.com/html/main.html>

Microsoft Office Access

- Download kostenlos für Studierende
- Versionen 2007 und 2010 verfügbar

→ <http://www.fsr-verkehr.de/services/msdnaa.htm>