

### Aufgabe 21 (5 Punkte)

Gegeben seien die Mondial-Tabellen, formulieren Sie folgende SQL-Anfragen und überprüfen Sie Ihre Ergebnisse mit dem SQLChecker im Zweig „Fortgeschritten“.

- Geben Sie den Ländernamen, den Namen der zugehörigen Hauptstadt und die Einwohnerzahl dieser Hauptstadt aller Länder in Amerika (Kontinent) sortiert nach den Ländernamen aus.
- Geben Sie die Namen aller Länder in Europa aus, für die weder ein Fluss (GEO\_RIVER) noch eine Wüste (GEO\_DESERT) eingetragen sind.
- Geben Sie die Namen aller Länder aus, für die bei jeder Stadt dieses Landes keine Einwohnerzahl (also NULL) eingetragen ist.
- Geben Sie die Namen aller Länder aus, deren Hauptstadt weniger als 500000 Einwohner hat und für die mehr als fünf Städte in der Datenbank eingetragen sind.
- Geben Sie die Namen der Länder an, die an kein Meer (SEA) grenzen und deren Hauptstadt an einem Fluss liegt (LOCATED).
- Geben Sie alphabetisch sortiert die Namen aller Städte aus, deren Breitengrad maximal einen Grad Differenz zur geographischen Breite von Berlin hat (Berlin selbst soll sich im Ergebnis befinden).
- Geben Sie einmalig die Namen der Länder aus, von denen eine Stadt am Atlantik (Atlantic Ocean) und eine Stadt am Mittelmeer (Mediterranean Sea) liegen (LOCATED).

### Aufgabe 22 (3 Punkte)

Gegeben seien die folgenden Tabellen mit folgenden Inhalten.

Z	MATNR	Z	NAME
	42		Ute
	43		Ulf
	44		Uwe

Z	MODNR	Z	NAME
	4711		Prog1
	4712		Math1
	4713		DB

Z	MATNR	Z	MODNR	Z	VERSUCH	Z	NOTE
	42		4711		1		5
	42		4711		2		2,7
	42		4713		1		1,7
	43		4711		1		5
	43		4711		2		5
	43		4713		1		5

Schreiben Sie zu jeder der folgenden Anfragen a) – g), wie die Aufgabenstellung ausgesehen haben könnte und geben Sie genau das Ergebnis an, das Sie als Ergebnis erwarten.

a) 

```
SELECT ZStudi.name
FROM ZStudi, ZModul ZM1, ZModul ZM2, ZPruefung ZP1, ZPruefung ZP2
WHERE ZStudi.matnr = ZP1.matnr
AND ZStudi.matnr = ZP2.matnr
AND ZM1.name = 'Prog1'
AND ZM2.name = 'DB'
AND ZP1.modnr = ZM1.modnr
AND ZP2.modnr = ZM2.modnr
AND ZP1.note = 5.0
AND ZP2.note = 5.0;
```

b) 

```
SELECT ZModul.name
FROM ZModul
WHERE NOT EXISTS (
SELECT *
FROM ZPruefung
WHERE ZModul.modnr = ZPruefung.modnr);
```

- c) 

```
SELECT ZStudi.name
FROM ZStudi, ZPruefung
WHERE ZStudi.matnr = ZPruefung.matnr
AND ZPruefung.versuch = 2
AND ZPruefung.note = 5.0
AND NOT EXISTS (
    SELECT *
    FROM ZPruefung ZP2
    WHERE ZStudi.matnr = ZP2.matnr
    AND ZP2.versuch = 3
    AND ZP2.modnr = ZPruefung.modnr);
```
- d) 

```
SELECT ZStudi.name
FROM ZStudi, ZPruefung, ZModul
WHERE ZStudi.matnr = ZPruefung.matnr
AND ZPruefung.modnr = ZModul.modnr
AND ZModul.name = 'Prog1'
AND ZPruefung.note = 5.0
INTERSECT
SELECT ZStudi.name
FROM ZStudi, ZPruefung, ZModul
WHERE ZStudi.matnr = ZPruefung.matnr
AND ZPruefung.modnr = ZModul.modnr
AND ZModul.name = 'DB'
AND ZPruefung.note = 5.0;
```
- e) 

```
SELECT *
FROM ZStudi LEFT JOIN ZPruefung
ON (Zstudi.matnr = ZPruefung.matnr
AND ZPruefung.NOTE < 5);
```
- f) 

```
SELECT *
FROM ZStudi LEFT JOIN ZPruefung
ON Zstudi.matnr = ZPruefung.matnr
WHERE ZPruefung.NOTE < 5;
```
- g) 

```
SELECT ZStudi.name, ZModul.name, ZPruefung.note
FROM ZStudi NATURAL JOIN ZPruefung
NATURAL JOIN ZModul;
```
- h) Geben Sie eine zu e) äquivalente Anfrage mit gleichem Ergebnis an, die ohne besondere JOIN-Operationen wie LEFT JOIN auskommt. Hinweis: Um neue NULL-Werte in einer SELECT-Zeile in Derby zu erzeugen, muss ein Typ dazu angegeben werden, z. B. cast(NULL as INTEGER)