**1.4.1.2 Verknüfung von Daten**

1.4.1.2.1

Aufgabe 1 : der 1. Ist der richtige. Um ein genaueres Ergebnis zu erzielen ist es möglich noch das Land in dem die Hauptstadt liegt als Bedingung anzugeben.

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) \* FROM `ort` WHERE Name = "Berlin" [AND](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/logical-operators.html#operator_and) Land="Deutschland"

Aufgabe 2: Mann kann nach keiner Hauptstadt suchen, da zu wenig Werte in der Tabelle ort vorhanden sind. Man kann höchstens nach der Einwohnerzahl sortieren.

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) Name FROM ort WHERE Land="Vereinigte Staaten von Amerika" ORDER BY Einwohner desc

**Ein Primärschlüssel einer Tabelle ist eine Kombination von Attributen, die eine Identifizierung eines Datensatzes eindeutig ermöglicht. Der Schlüssel ist minimal, das heißt, man kann kein Attribut weglassen, um den Datensatz eindeutig zu finden.**

**Wenn mehrere Attribute benötigt werden (z.B. Name, Laenge, Breite), dann wird in der Regel ein künstlicher Primärschlüssel (z.B. eine Ortsnummer ONR) eingefügt.**

1.4.1.2.3

Aufgabe 1 : SELECT \* FROM land Where Kontinent ="Europa" Order by Einwohner desc

Es fehlt Russland, da Russland nicht komplett in Europa liegt.

Aufgabe 2 : Die Einwohneranzahl ist nicht genauer definiert d. h. man weiss nicht genau wie viele Einwohner in dem Land wohnen. Die Werte von Kontinentfläche und KontinentEinwohner werden ab Zeile 2 nicht mehr richtig angezeigt.

Aufgabe 3: Das liegt daran, dasss der Artikel evtl in einer Art Datenbank abgespeichert wurde und diese Nummer ist ID für den Artikel

**Ein Fremdschlüssel ist ein Attribut in einer Tabelle, das auf den Primärschlüssel einer anderen Tabelle verweist.**

**Über den Fremdschlüssel werden zwei Tabellen miteinander verbunden. Dies stellt eine Beziehung zwischen zwei Tabellen dar. Beziehungen sind nach den Tabellen das zweite wichtige Konzept von relationalen Datenbanken.**

1.4.1.2.5

Aufgabe 1: Es hat alle Spalten aus beiden Tabellen ausgegeben und nicht zusammengepackt

Aufgabe 2: KNR ist genauer definiert aus welcher Tabelle er jetzt welches KNR nehmen soll. Richtig wäre es land.KNR=kontinent.KNR

1.4.1.2.6

Will man in SQL Daten aus mehreren Tabellen ausgeben, so werden sog. **"Joins"** gebildet. Dabei muss der Programmierer angeben, wie die Tabellen miteinander verbunden werden sollen.

Um zwei Tabellen miteinander zu verbinden (engl. to join), muss im **WHERE**-Teil des SQL-Befehls eine **Joinbedingung** angegeben werden.  
Diese legt fest, dass nur "passende" Datensätze ausgegeben werden, bei denen der Fremdschlüssel der abhängigen Tabelle zum Primärschlüssel der übergeordneten Tabelle passt (sog. **inner join**).

1.4.1.2.7

Aufgabe 1:

Ich würde beim 1. Und 2. Einen Primärschlüssel erstellen welcher nicht künstlich ist und beim 3. Und 4. Einen künstlichen Primärschlüssel.

Aufgabe 2:

a: Man kann somit den Schüler alle seine Noten eindeutig zuweisen.

Dies könnte zur Verwechslung führen da für die Namen in der Notentabelle „Schüler“ benutzt wurde

b: Schüler

c: Schüler

d: Schüler aus der Notentabelle muss auch in Name umgenannt werden

Aufgabe 3:

Man muss einen Join zwischen den beiden Tabellen bilden. Und die doppelten Werte zusammenpacken.

Beispiel:

SELECT \* FROM ort, land WHERE land.LNR = ort.LNR

Aufgabe 4:

a: [SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) land.Name, land.Einwohner FROM kontinent, land WHERE land.KNR = kontinent.KNR [and](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/logical-operators.html#operator_and) land.KNR = "AS“

b: SELECT land.Name, land.Einwohner FROM kontinent, land WHERE land.KNR = kontinent.KNR and land.KNR = "AS" ORDER BY land.Einwohner desc

c: SELECT land.Name, land.Einwohner FROM kontinent, land WHERE kontinent.KNR = land.KNR AND land.KNR = „EU“ AND land.Einwohner >= 300000

Aufgabe 5:

a: SELECT ort.Name, ort.LNR, ort.Einwohner FROM kontinent, land, ort WHERE ort.ONR = land.HauptONR AND ort.LNR = land.LNR AND land.KNR = kontinent.KNR AND ort.Einwohner >= 10 ORDER BY ort.Einwohner desc

b: SELECT ort.Name, ort.LNR, ort.Einwohner FROM kontinent, land, ort WHERE ort.ONR = land.HauptONR AND ort.LNR = land.LNR AND land.KNR = kontinent.KNR AND ort.Einwohner >= 1000000 AND land.KNR = "EU" ORDER BY ort.Einwohner desc

c: SELECT ort.Name, ort.LNR, ort.Einwohner FROM kontinent, land, ort WHERE ort.ONR = land.HauptONR AND ort.LNR = land.LNR AND land.KNR = kontinent.KNR AND ort.Einwohner >= 1000000 AND land.Einwohner <= 10000000 ORDER BY ort.Einwohner desc

d: SELECT ort.Name, ort.LNR, land.Name FROM kontinent, land, ort WHERE ort.ONR = land.HauptONR AND ort.LNR = land.LNR AND land.KNR = kontinent.KNR