RQ

> These measurements were taken in a MacBook Air, 2.2 GHz Dual-Core Intel Core i7 with 8 GB 1600 MHz DDR3 with 500GB Flash Storage running macOS Big Sur V 11.3 and SQLite version 3.36.0 2021-06-18 18:36:39.

Consider the relational version of the Mondial database.

Task 1

Write tuple-relational calculus (TRC) expressions that, upon evaluation, return the data characterized by each of the following English-language specifications:

(1) Return the name of any country that has a lake.

$$\{A|\exists C \in Counry\ \exists L \in geo\_Lake\ (C.Code = L.Counry \wedge A.Country\_Name = C.Name)\}$$

(2) Return all the available attributes on cities whose population is between 3 and 5 million inhabitants.

$$\{A|A \in City\ \wedge\ A.Population > 3000000 \wedge\ A.Population < 5000000\}$$

(3) Return the country code and the continent of every country not in Europe or in Australia/Oceania.

$$\{A|\exists C \in Country\ \exists T \in Continent\ \exists E \in encompasses(C.code = E.country \wedge T.Name \wedge \neg\ (T.Name = "Europe" \wedge\ T.Name = "Australia/Oceania") \wedge A.Country\_Code = C.code \wedge\ A.Continent = T.Name)\}$$

(4) Return the names of countries that also give their name to one of its own provinces.

$$\{A|\exists C \in Country\ \exists P \in Province\ (C.Code = P.Country \wedge C.Name = P.Name\ \wedge A.Country\_Name = C.Name)\}$$

(5) Return the names of countries that are not landlocked (i.e., have a sea coast).

$$\{A|\exists C \in Country\ \exists S \in geo\_Sea\ (C.Code = S.country \\ \wedge A.Country\_Name = C.Name)\}$$

Task 2

Write relational-algebraic (RA) expressions that, upon evaluation, returns the data characterized by each of the following English-language specifications.

(6) Return the names of countries that are not landlocked (i.e., have a sea coast).

```
//Q6
\rename_{Country_Name} \project_{Name}(
    Country \join_{Code=Country} geo_Sea
);
```

(7) Return the names of all lakes, rivers and seas.

```
//Q7
\rename_{Name} \project_{Lake}(
    geo_Lake \union geo_Sea \union geo_River
);
```

(8) Return the name of the country and the name of the organization of countries with Buddhist populations.

```
//Q8
\rename_{country_Name,org_Name} \project_{C_Name,O_Name}(
  \rename_{C_Name,C_Code,C_Capital,C_Province,C_Area,C_Popultion} Country
\join_{C_Code = R_Country and R_Name = "Buddhist"}
\rename_{R_Country,R_Name,R_Percentage} Religion \join_{C_Code = M_Country}
\rename_{M_Country,M_Organization,M_Type} isMember \join_{O_Abbreviation =
M_Organization}
\rename_{O_Abbreviation,O_Name,O_City,O_Country,O_Province,O_Established}
Organization
);
```

(9) Return the names of countries that also give their name to one of its own provinces.

```
//Q9
\rename_{Country_Name} \project_{Name}(
  \select_{Name = p_Name}(
    Country \join_{Code = p_Country}
\rename_{p_Name,p_Country,p_Population,p_Area,p_Capital,p_CapProv} Province
  )
);
```

(10) Return, for every river in Great Britain, the length of that river.

```
//Q10
\rename_{river_Name,Length} \project_{Name,Length}(
  \select_{r_Country = "GB"}(
    River \join_{Name = r_River} \rename_{r_River,r_Country,r_Province}
geo_River
  )
);
```

(11) Return the names of all cities that have a population larger than that of the capital city of the country.

```
//Q11
\rename_{city_Name} \project_{CI2_Name}(

\rename_{CI1_Name,CI1_Country,CI1_Province,CI1_Population,CI1_Latitude,CI1_Long
itude,CI1_Elevation} City \join_{CI1_name = CO_Capital and CI1_Country =
CO_Code} \rename_{CO_Name,CO_Code,CO_Capital,CO_Province,CO_Area,CO_Population}
Country \join_{CO_Code = CI2_Country and CI2_Population > CI1_Population}
\rename_{CI2_Name,CI2_Country,CI2_Province,CI2_Population,CI2_Latitude,CI2_Long
itude,CI2_Elevation} City
);
```

Task 3

Write SQL expressions that, upon evaluation, returns the data characterized by each of the following English-language specifications. Use duplicate removal where appropriate (e.g., when a duplicate is not required in the intended answer).

(12) Return the names of up to 10 countries and the value corresponding to half the country’ s population.

```
--Q12
select C.Name as country_Name,C.Population/2 as HalfPopulation
from Country C
LIMIT 10;
```

(13) Return all the information available about cities whose name is Manchester.

```
--Q13
select *
from City
where Name = "Manchester";
```

(14) Return the name of cities whose name starts with the substring 'Man’ .

```
--Q14
select Name as city_Name
from City C
where C.Name like 'Man%';
```

(15) Return the name of the country and the name of the organization of countries with Buddhist populations that are members of organizations established after 1st December 1994.

```
--Q15
select BCM.Name as country_Name,O.name as org_Name
from Organization O,(
  select BC.Name,M.Organization
  from isMember M,
    (select C.Code,C.Name
    from Country C, Religion R
    where C.Code = R.Country and R.Name = "Buddhist") BC
  where BC.Code = M.Country) BCM
where O.Abbreviation = BCM.Organization and Established > "1999-12-01"
;
```

(16) Return the name of each country with the number of islands in it.

```sql
--Q16
select C.Name as country_Name,L.island_Num
from Country C,(
select I.Country, count(Country) as island_Num
from geo_Island I
group by Country
) L
where C.Code = L.Country
order by C.Name ;
```