Project 2 – Part 1

Malavika Vidyarthi **Tejaswi Singh**
1001398623 **1001387430**

High Level Specifications for the Geography Database are as follows:

1) The database will keep track of COUNTRIES, STATES, and CITIES. Each COUNTRY has a unique CountryName, CapitalCity, and NeighbouringCountries of each COUNTRY.
   In addition, the continent to which each country belongs is kept. The PopulationHistory of each country is kept. PopulationHistory is a list of (Year, Population) entries (not every year has to be on the list).
2) Countries are made up of STATES (sometimes, the divisions within a country (for example, in Canada) are called provinces, but we will call them states for uniformity). Assume that a STATE is uniquely identified by the StateName. For each STATE, the CapitalCity of the state is kept, and the database will keep track of the NeighbouringStates and NeighbouringCountries, if any. For some states, the population history is also kept.
3) The database will keep track of some major CITIES. A city is identified by the combination (CityName, StateName). For some cities, the database will keep track of the current Population for the city, and the Longitude/Latitude co-ordinates of the city center.
4) The database will keep track of some of the major WATER_BODIES, which can be RIVERS, LAKES, SEAS, CANALS, or OCEANS. A WATER_BODY is identified by a unique WBName. The

database will keep track of the Cities, Countries, States, or Continents that are adjacent to or intersect with a water body.

Other assumptions made:

1) Every **Continent** has at least one **Country**.
2) Every **Country** has at least one **State**.
3) Every **State** has at least one **City**.
4) Every **Country** has at least one neighboring **Country**.
5) Each **State** is neighbored by either a **State** of its own **Country** or a **State** of Another **Country** or **both**.
6) Each **Country** has exactly one **Capital**.
7) Each **State** has exactly one **Capital**.
8) A **Continent** is identified by a unique **ContinentName**.
9) Population history is a multivalued, composite attribute.
10) **Cities** is a weak entity and its identifying relationship is with entity **State**.
11) **Water_Bodies** is a **Super Class** which is divided into the following **sub classes** based on the **TYPE** of water_body:
    a) **RIVERS**
    b) **CANALS**
    c) **OCEANS**
    d) **LAKES**
    e) **SEAS**

Documentation of the EER Diagram:

1) **ENTITY TYPES – The EER Diagram consists of the following entities types:**

   a) **Countries**

      Countries Entity type has the following **attributes**:

i) CountryName – Primary key

ii) PopulationHistory – Multivalued, Composite attribute with Population and Year

b) States

States Entity type has the following **attributes**:

i) StateName – Primary key

ii) PopulationHistory – Multivalued, Composite attribute with Population and Year

c) Cities

Cities is a weak entity and its Identifying Relationship is with States entity type (Thus, exists only with Statename). Its **attributes** are as follows:

i) CityName – Primary Key

ii) Population

iii) Latitude

iv) Longitude

d) Continent

Continent Entity type has the following **attributes**:

i) ContinentName – Primary Key

e) Water_Bodies

Water_Bodies Entity type has the following **attributes**:

i) WBName – Primary Key

Water_Bodies is a **Super Class** which is divided into the following **sub classes** based on the **TYPE** of water_body:

a) **RIVERS**

b) **CANALS**

c) **OCEANS**

d) **LAKES**

e) **SEAS**

2) RELATIONSHIP TYPES - The EER Diagram consists of the following relationship types:

a) **Adjacent –** M:N recurring relationship with Countries
b) **Neighbor –** M:N relationship between Countries and States
c) **Neighbor –** M:N recurring relationship with States

All these relationship types have a similar role, where in one the behave as a neighbor of each other (neighbor of countries/states/each other)

d) **Of -** N:1 relationship between Countries and Continent (**total** relationship)
e) **Consists_Of -** 1:N relationship between Countries and States (**total** relationship)
f) **Has –** 1:N **identifying** relationship between States and Cities
g) **Has_Capital –** 1:1 relationship between Countries and Cities
h) **Has_Capital –** 1:1 relationship between Countries and Cities
i) **Intersect –** M:N relationship between Countires and Water_Bodies.
j) **ContinentsOn -** M:N relationship between Continents and Water_Bodies.
k) **StatesOn -** M:N relationship between States and Water_Bodies.
l) **CitiesOn -** M:N relationship between Cities and Water_Bodies. The above 4 relationship types represent continents/countires/states/cities that intersect with or on the the waterbodies.

EER Diagram is shown below in Figure1. (MS Paint was used to make the EER Diagram).

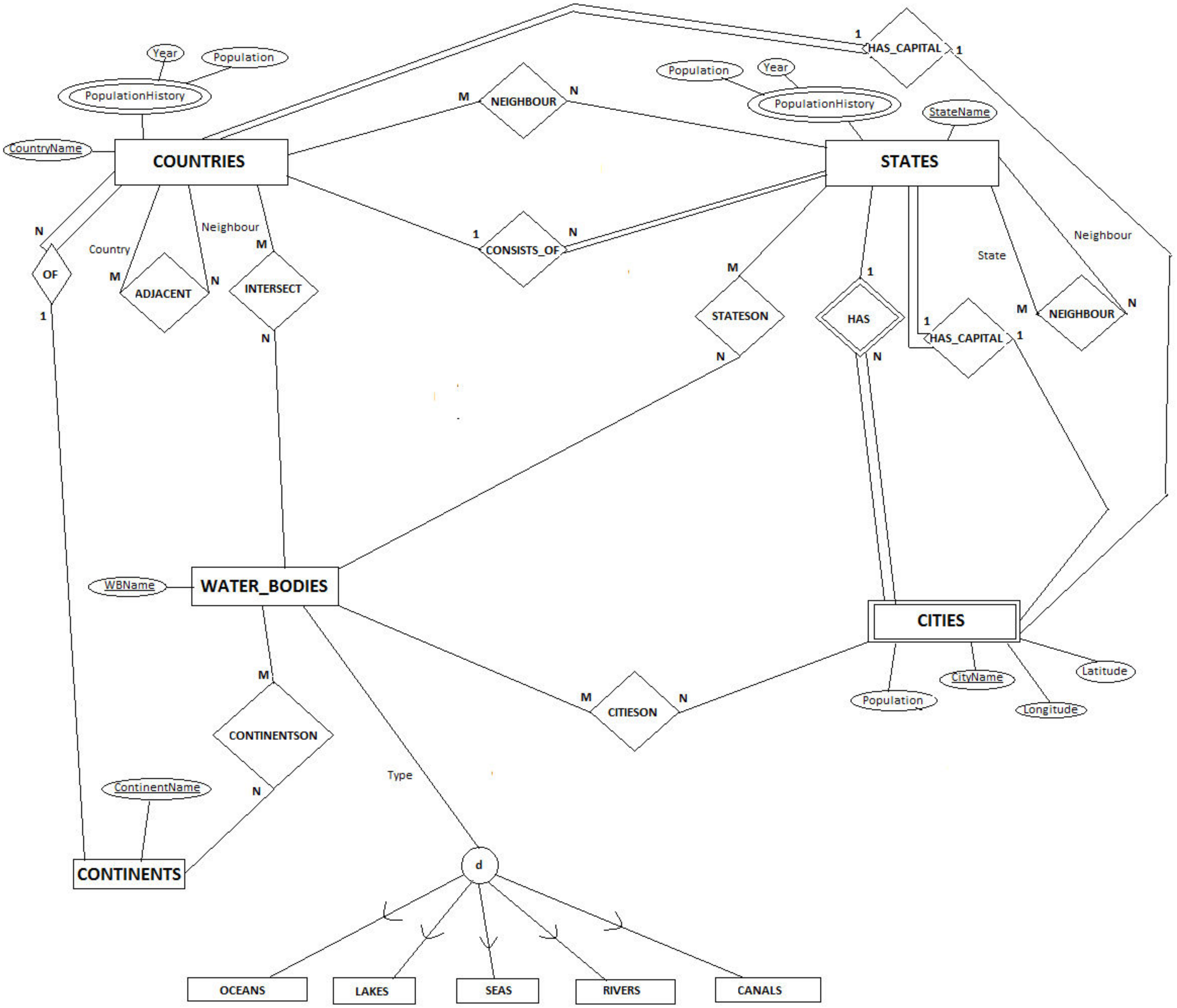


FIGURE 1. - ENHANCED ENTITY RELATIONSHIP DIGRAM FOR GEGRAPHY DATABASE

MAPPING OF EER TO RELATIONAL SCHEMA

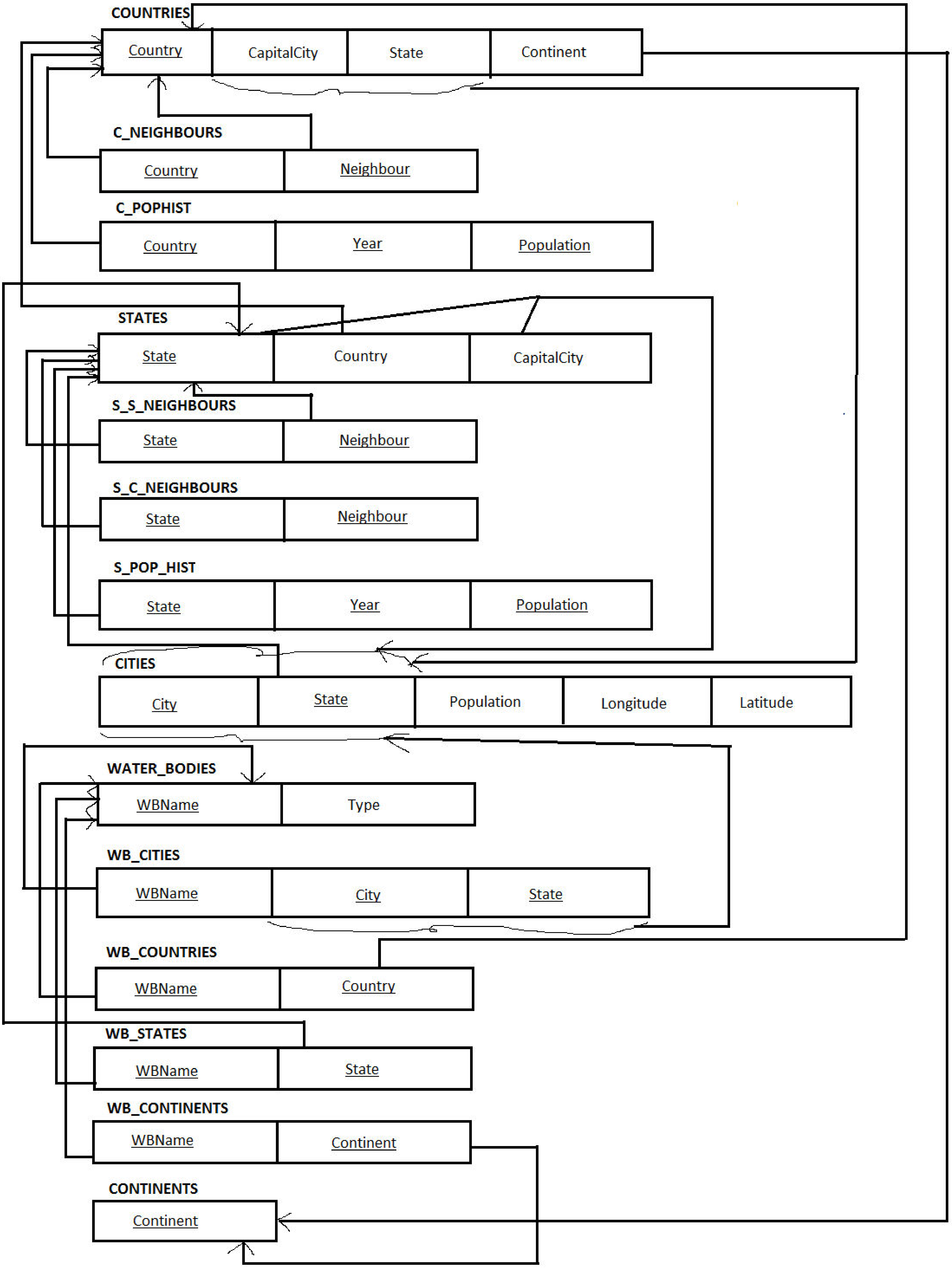
Strong entity types (COUNTRIES, STATES, WATER_BODIES, CONTINENT) are transformed to separate tables. In each, Primary key is set.

Weak entity type (CITIES) is transformed to a table. The primary key is partial key of CITIES plus primary key of the STATES.

1:N relationship types are identified and the primary key of entity type on the 1 side is included in the N side relation as the foreign key. The combination of both becomes the primary key of this relation.

M:N Relationship types are identified and a separate table is formed for each relationship type with combination of primary keys of the participating entity types as the primary key.

The specialization (Oceans, Rivers, Seas, Canals, Lakes) is transformed using a single table option. An attribute Type is included in the new relation.


COUNTRIES
Country
CapitalCity
State
Continent
C_NEIGHBOURS
Country
Neighbour
C_POPHIST
Country
Year
Population
STATES
State
Country
CapitalCity
S_S_NEIGHBOURS
State
Neighbour
S_C_NEIGHBOURS
State
Neighbour
S_POP_HIST
State
Year
Population
CITIES
City
State
Population
Longitude
Latitude
WATER_BODIES
WBName
Type
WB_CITIES
WBName
City
State
WB_COUNTRIES
WBName
Country
WB_STATES
WBName
State
WB_CONTINENTS
WBName
Continent
CONTINENTS
Continent

SQL STATEMENTS TO CONSTRUCT A DATABASE

```
CREATE TABLE CONTINENTS (Continent VARCHAR(40), CONSTRAINT c2pk
PRIMARY KEY(Continent));

CREATE TABLE COUNTRIES( Country VARCHAR(40) NOT NULL, CapitalCity
VARCHAR(40) NOT NULL, State VARCHAR(40), Continent VARCHAR(50),
CONSTRAINT COUPK PRIMARY KEY(Country), CONSTRAINT C1 FOREIGN
KEY(Continent) REFERENCES CONTINENTS(Continent));

CREATE TABLE C_NEIGHBOURS( Country VARCHAR(40) NOT NULL, Neighbour
VARCHAR(40), CONSTRAINT CNEPK PRIMARY KEY(Country, Neighbour),
CONSTRAINT COUFK FOREIGN KEY(Country) REFERENCES
COUNTRIES(Country), CONSTRAINT NEIFK FOREIGN KEY(Neighbour)
REFERENCES COUNTRIES(Country));

CREATE TABLE C_POP_HIST( Country VARCHAR(40) NOT NULL, Year
NUMBER(5) NOT NULL, Population
NUMBER(14), CONSTRAINT PHISPK PRIMARY KEY(Country, Year, Population),
CONSTRAINT PHISFK
FOREIGN KEY(Country) REFERENCES COUNTRIES(Country));


CREATE TABLE STATES( State VARCHAR(40) NOT NULL, Country
VARCHAR(40), CapitalCity VARCHAR(40), CONSTRAINT STAPK PRIMARY
KEY(State), CONSTRAINT COU_SFK FOREIGN KEY(Country) REFERENCES
COUNTRIES(Country) ON DELETE CASCADE);

CREATE TABLE S_S_NEIGHBOURS(State VARCHAR(40) NOT NULL, Neighbour
VARCHAR(40), CONSTRAINT SSSNPK PRIMARY KEY(State, Neighbour),
CONSTRAINT STSSFK FOREIGN KEY(State) REFERENCES STATES(State),
CONSTRAINT NEISSFK FOREIGN KEY(Neighbour) REFERENCES
STATES(State));

CREATE TABLE S_C_NEIGHBOURS(State VARCHAR(40) NOT NULL, Neighbour
VARCHAR(40), CONSTRAINT SCNEIPK1 PRIMARY KEY(State, Neighbour),
CONSTRAINT SCNBFK1 FOREIGN KEY(State) REFERENCES STATES(State),
CONSTRAINT NEIGHSFK1 FOREIGN KEY(Neighbour) REFERENCES
COUNTRIES(Country));

CREATE TABLE S_POP_HIST( State VARCHAR(40) NOT NULL, Year
NUMBER(5), Population NUMBER(10), CONSTRAINT SPHPK PRIMARY
KEY(State, Year, Population), CONSTRAINT SPHFK FOREIGN KEY(State)
REFERENCES STATES(State));

CREATE TABLE CITIES( City VARCHAR(40) NOT NULL, State VARCHAR(40),
Population NUMBER(10), Longitude DECIMAL(7,4), LATITUDE DECIMAL(7,3),
CONSTRAINT CITPK PRIMARY KEY(City, State), CONSTRAINT STAFK
FOREIGN KEY(State) REFERENCES STATES( State));

CREATE TABLE WATER_BODIES(WBName VARCHAR(40), Type
VARCHAR(40), CONSTRAINT WBPK PRIMARY KEY(WBName));

CREATE TABLE WB_COUNTRIES(WBName VARCHAR(40), Country
VARCHAR(40), CONSTRAINT WBPK1 PRIMARY KEY(WBName, Country),
CONSTRAINT COUFK1 FOREIGN KEY(Country) REFERENCES
COUNTRIES(Country), CONSTRAINT WB1FK FOREIGNKEY(WBName)
REFERENCES WATER_BODIES(WBName));

CREATE TABLE WB_CITIES(WBName VARCHAR(40), City VARCHAR(40), State
VARCHAR(40),CONSTRAINT WBPK2 PRIMARY KEY(WBName, City, State),
CONSTRAINT CITFK2 FOREIGN KEY(City, State) REFERENCES
CITIES(City,State), CONSTRAINT WBFK2 FOREIGN KEY(WBName)
REFERENCES WATER_BODIES(WBName));



CREATE TABLE WB_STATES(WBName VARCHAR(40), State VARCHAR(40),
CONSTRAINT WBPK3 PRIMARY KEY(WBName,State), CONSTRAINT STAFK3
FOREIGN KEY(State) REFERENCES STATES(State), CONSTRAINT WBFK3
FOREIGN KEY(WBName) REFERENCES WATER_BODIES(WBName));
CREATE TABLE WB_CONTINENTS(WBName VARCHAR(50), Continent
VARCHAR(50), CONSTRAINT WB_CONT_PK PRIMARY KEY(WBName,
Continent));

ALTER TABLE WB_CONTINENTS ADD CONSTRAINT TO_WB_FK FOREIGN
KEY(WBName) REFERENCES WATER_BODIES(WBName);

ALTER TABLE WB_CONTINENTS ADD CONSTRAINT TO_COUNTRY_FK
FOREIGN KEY(Continent) REFERENCES CONTINENTS(Continent);

ALTER TABLE STATES ADD CONSTRAINT STATE_TO_CITY_FK FOREIGN
KEY(CapitalCity,State) REFERENCES CITIES(City,State);

Alter table STATES disable constraint STATE_TO_CITY_FK;
```