Name: Omid Azodi

Date: 02/25/2016

Class: CS443

Professor: Dr. Hadaegh

CS443-Lab 3

**Question 1:**

Suppose that our database has the following table.

Person

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Con  ID | Con  Name | Con  Pop | Con  Size | State  code | State  Name | State  Rgn | State  Size | State  Pop | Cty  Code | Cty  Name | Cty  Size | Per  SSN | Per  Name | Per  Age | Per  DofB | Per  Add |

**Field Explanation:**

Con: Stands for Country

Pop: Stands for population

Rgn: Stands for region (like west, east, central, etc.)

Cty: Stands for City

Per: Stands for Person

DofB: Stands for date of birth

Add: Stands for Address

**It is assumed that**

* Every country in the world has a different country ID
* Every city in the world has a different city code
* Every state in the world has a different state code, and
* Every person in the world has a different SSN
* Every person in the world has only one citizenship and has only one address

**Other Assumptions:**

* All relationships are total participation
* There is no village, county, area, etc. A country consists of several states and each state has several cities

1. **Based on the above assumptions, what do you choose to be the primary key of Person table? Why?**

* I chose the primary key for the Person table to be PerSSN. I chose this, because every person in the world has a different SSN, therefore this means it would be something to be unique.

1. **Explain the anomalies exist in the Person table. Choose only one example of insert anomaly, one example of delete anomaly and one example of update anomaly. Note that update does not mean adding or deleting records. It only refers to modifications of values in some rows of the table.**

* Insert anomaly: An insert anomaly that exists in the Persons table would be if you wanted to add/insert a new country, you couldn’t be able to unless you had a person living in the country, before adding that country.
* Delete anomaly: An delete anomaly that exists in the Persons table would be if you wanted to erase/delete a person and that person was the last one living in a country you would have to remove the country and all information in regards to that country such as city, state, etc.
* Update anomaly: An update anomaly that exists in the Persons table would be if you wanted to update/change a Country name you would have to change all people living in that country to the new country name.

1. **Normalize the table; create as many as tables necessary such that all new tables are in third normal form. All the transitive and derived dependencies must be removed. You may add more assumptions as you wish; however, your assumptions must be realistic.**

The original table is in 1st normal form, because there is one value/instance in each cell/field of the table.

The original table is also in 2nd normal form, because there is no partial functional dependency, because there is no composite primary key for the table.

I will then throughout explain what my steps are to make it 3rd normal form. The way to tell if it is in third normal form is to make sure it is in 2nd normal form, which I did, and then make sure there is no transitive or derived dependency. So below the steps show that I took out the transitive dependencies and derived dependencies.

Original Table:

Person(PerSSN, PerName, PerAge, PerDofB, PerAdd, ConId, ConName, ConPop, ConSize, StateCode, StateName, StateRgn, StateSize, StatePop, CityCode, CityName, CitySize)

Derived:

* The reason why I am taking out ConPop is because ConPop can be derived by adding up all the people that live in the country.
* The reason why I am taking out ConSize is because ConSize can be derived from CitySize by adding up all the city sizes.
* The reason why I am taking out StatePop is because StatePop can be derived by adding up all the people that live in the state.
* The reason why I am taking out StateSize is because StateSize can be derived from CitySize by adding all the city sizes inside that state.
* The reason why I am taking out PerAge is because Age can be derived from PerDofB by subtracting current year minus year born.

Table after derived attributes are removed:

Person(PerSSN, PerName, PerDofB, PerAdd, CountryId, CountryName, StateCode, StateName, StateReg, CityCode, CityName, CitySize)

//\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

We split the table above because there is a transitive dependency and I will split them up according below:

Country(CountryId, CountryName)

In the person table below I will take out CountryName but leave CountryId to be a foreign key that refers to the primary key of the table above. Because with the CountryName we can find the CountryId.

Person(PerSSN, PerName, PerDofB, PerAdd, CountryId, StateCode, StateName, StateReg, CityCode, CityName, CitySize)

//\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

We split the table above because there is a transitive dependency and I will split them up according below:

Country(CountryId, CountryName)

State(StateCode, StateName, StateRegion, CountryId)

In the person table below I will take out CountryId and put it in the State table, because the countryId will now be a foreign key that refers to the Country Table. I will take out StateName, StateRegion from Person table, but leave StateCode in the Person table and that will be a foreign key in the Person table that will refer to the State table.

Person(PerSSN, PerName, PerDofB, PerAdd, StateCode, CityCode, CityName, CitySize)

//\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

City(CityCode, CityName, CitySize, StateCode)

State(StateCode, StateName, StateRegion, CountryId)

Country(CountryId, CountryName)

Person(PerSSN, PerName, PerDofB, perAdd, CityCode)

1. **Draw your ERD based on fully normalized table (Reverse Engineering)**

**1**

City

State

**M**

**1**

R2

R1

**M**

R3

**M**

**1**

Country

Person