



CS 443

LAB 3

Malialosa Taupule

February 16, 2017



Questions:

Our database has the following table:

PERSON

Con ID	Con Name	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: Country
Pop: Population
Rgn: Stands for region (e.g. west, east, central, etc.)
Cty: City
Per: Person
DofB: Date of birth
Add: Address

Assumptions:

- 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
- Every person in the world has a different SSN
- Every person in the world has only one citizenship and only one address
- There's 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 the Person table? Why?

SSN should be the primary key of the Person table because it uniquely identifies each person (or instance of the person table) within the database.

2.) Explain the anomalies that exist in the Person table. Choose only one example of each anomaly: insert, delete, and modification.

Insert anomaly

You cannot add a new city into the database unless you also add a person who lives there. This is because PerSSN is the primary key of this table, thus its data value **cannot** be null.

Delete anomaly

Say there's only one person living in a city. If he/she dies and you want to delete him from the database, you would lose information pertaining to the city he/she lived in. Remember, PerSSN **cannot** be null because it is the primary key of this table.

Modification anomaly

If a person moves, you would have to update his/her information to match his new address. This would require the update of many of his/her associated data values (i.e. PerAdd, CtyName, StateName, ConID, etc.).

3.) **Normalize the table. Create as many tables as necessary such that all new tables are in third normal form. All the transitive and derived dependencies must be removed.**

- ✓ The current table is assumed to be in first normal form, therefore we can check if it's in second normal form.
- ✓ The current table is also in second normal form because PerSSN is the only primary key, making partial functional dependencies impossible.
- ✗ The current table is not in third normal form because there exists many derived dependencies. In order for it to be fully normalized, each derived dependency must be removed.

Derived dependencies:

These attributes were removed because each were derived dependencies and therefore, unnecessary.

- **PerAge:** I removed this attribute because you can derive a person's age from subtracting the current date and the person's PerDofB.
- **CtySize:** This attribute was removed because you can derive CtySize by calculating the city's population. You do this by counting each person who lives in each city.
- **StatePop & ConPop:** These attributes were removed for a similar reason to the removal of CtySize. You can calculate StatePop by totaling the population within each of the state's cities. Similarly, you can also calculate ConPop by totaling the population within each of the country's states.
- **StateSize & ConSize:** Like CtySize, these attributes were removed because you could derive both the StateSize and ConSize by calculating both the state population and country population.

New normalized tables:

Table 1: PERSON(PerSSN, PerName, PerDofB, PerAdd, CtyCode*)

Table 2: CITY(CtyCode, CtyName, StateCode*)

Table 3: STATE(StateCode, StateName, StateRgn, ConID*)

Table 4: COUNTRY(ConID, ConName)

4.) Draw your ERD based on the fully normalized table (i.e. Reverse Engineering)

