

Final Project, Stage 2

Step 1-3: (ERD is at the bottom)

Our group chose to do an ER diagram out of the simplicity of our data and maintaining a more cleaner visual of our data.

The ERD has 5 entities: *User*, *Country*, *Continent*, *US States*, and *YearlyMax*.

User: The attributes are CountryName, Username, and Password. *User* stores the user login information, which is based on the Username, Password, and specified CountryName. CountryName is the foreign key, as it will be used to correlate *User* to *Country*. Overall, *User* is the entity, as it oversees individual attributes of a user like their country of choice, username, and password.

Country: The attributes are CountryName, Year, Population, Military Size, GDP, CO2 Emissions, Birth Rate, and Continent. The CountryName and Year are the Composite Primary Key because each country and year correspond to a population, military size, GDP, CO2 emissions, birth rate, and continent. The continent is the foreign key because it will be used to correlate *Country* to *Continent*. Overall, *Country* is the entity as it oversees individual attributes like the population size, military size, GDP, CO2 emissions, birth rate, and continent in any given country in any given year.

Continent: The attributes are the ContinentName, Year, Population, Military Size, GDP, CO2 Emissions, and Birth Rate. The ContinentName and Year are the Composite Primary Key, where each continent and year can correspond to a population, military size, GDP, CO2 emissions, birth rate. ContinentName and Year are the primary key because it will be used to correlate the *Country* and *Continent*. Overall, *Continent* is the entity as it oversees individual attributes like the continent, year and the respective population size, military size, GDP, CO2 emissions, birth rate.

US States: The attributes are the State, Year, Population, Military Size, GDP, CO2 Emissions, Birth Rate, and CountryName. The State and Year are the Composite Primary Key, where each state and year can correspond to a population, military size, GDP, CO2 emissions, birth rate, and a country it resides in. State and Year are the primary key because it will be used to correlate the *US States* and *Country*. Overall, *US States* is the entity as it oversees individual attributes like the state, year and the respective population size, military size, GDP, CO2 emissions, birth rate, and country.

YearlyMax: The attributes are the Year, Population, Military Size, GDP, CO2 Emissions, Birth Rate. Year is the primary key, where each year corresponds to the country with the highest population size, military size, GDP, CO2 emissions, birth rate. Year is the foreign key because it will be used to correlate the *YearlyMax* and *Country*. Overall, the *YearlyMax* is the entity as it

oversees individual attributes like the max population size, military size, GDP, CO2 emissions, birth rate in a given year.

Side Note: What you will see amongst all of these entity-attribute relationships, is that they follow a similar format: the entity is the overall structure with the attributes serving as the individual components that make up the whole structure.

Relationships: *Country* has a many-many relationship with the *Continent* as there are instances where multiple countries have multiple continents, like Russia or Turkey. In general, each country is mapped to a singular continent, where a continent can have more than one country. Since there are some countries with multiple continents, you have a many-many relationship. *Country* has a 1-many relationship with *US States* because 1 country (the US) has a link to multiple states. All the US States will map to one country, which is the USA. *Country* has a many-1 relationship with *User*, each user has 1 country mapped to a country, but since multiple users can have the same country, you see a many-1 relationship. Lastly, *Country* has a 1-many relationship with *YearlyMax* because a singular country can have many different records, however, a record will only be linked to one country.

Step 4 - Normalizing Data with 3NF:

Country:

CountryName*Year - > Population
CountryName*Year - > MilitarySize
CountryName*Year - > CO2 Emissions
CountryName*Year - > BirthRate
CountryName*Year - > GDP
CountryName -> ContinentName

The table satisfies 3NF because there are no transitive dependencies or partial dependencies.

Continent:

ContinentName*Year - > Population
ContinentName*Year - > MilitarySize
ContinentName*Year - > CO2 Emissions
ContinentName*Year - > BirthRate
ContinentName*Year - > GDP

The table satisfies 3NF because there are no transitive dependencies or partial dependencies.

US States:

USStates*Year - > Population
USStates*Year - > MilitarySize
USStates*Year - > CO2 Emissions
USStates*Year - > BirthRate
USStates*Year - > GDP
USStates -> CountryName

The table satisfies 3NF because there are no transitive dependencies or partial dependencies.

YearlyMax:

Year - > Population
Year - > MilitarySize
Year - > CO2 Emissions
Year - > BirthRate
Year - > GDP

The table satisfies 3NF because there are no transitive dependencies or partial dependencies.

User:

Username -> Password
Username -> Country

The table satisfies 3NF because there are no transitive dependencies or partial dependencies

Step 5 - ER to Logical Design (Relational Schema)

Country(CountryName : VARCHAR(32) [PK], Year : INT, Population : INT, MilitarySize : INT, GDP : Decimal, C02 Emissions : Decimal, BirthRate : Decimal, ContinentName : VARCHAR(20) [FK to Continent.ContinentName])

Continent(ContinentName : VARCHAR(20) [PK], Year : INT, Population : INT, MilitarySize : INT, GDP : Decimal, C02 Emissions : Decimal, BirthRate : Decimal)

State(StateName : VARCHAR(12) [PK], Year : INT, Population : INT, MilitarySize : INT, GDP : Decimal, C02 Emissions : Decimal, BirthRate : Decimal)

YearlyMax(Year : INT [PK], CountryName : VARCHAR(32), Population : INT, MilitarySize : INT, GDP : Decimal, CO2 Emissions : Decimal, BirthRate : Decimal)

User(Username : VARCHAR(20) [PK], Password : VARCHAR(20), CountryName : VARCHAR(32) [FK to Country.CountryName])

