Produce an ER diagram for a relational database.

Assumptions:

(1) After carefully reviewing the values recorded in the table, I discovered that many values have a calculative relationship with each other, such as:

total_vaccinations<- total_vaccinations_per_hundred,

total_vaccinations = people_vaccinated + people_fully_vaccinated + total_boosters,

people_vaccinated <- people_vaccinated_per_hundred,

people_fully_vaccinated <- people_fully_vaccinated_per_hundred,

total_boosters <- total_boosters_per_hundred,

total_distribute <- total_distribute_per_hundred,

total_vaccination <- daily_vaccinations_raw, daily_vaccinations

daily_vaccinations_raw, daily_vaccinations < daily_vaccinations_per_million,

total_vaccinations/ total_distributed = share_doses_used.

Therefore, in the database ER model, I do not need to store all the original data columns in each table; I **only need to retain the key columns**, such as:

people_vaccinated, people_fully_vaccinated, total_boosters, total_distribute.

(2) **iso_code** is the unique code of each country.

(3) Each country has a vaccination record and vaccine production record for a period of time. A country may use different brands of vaccines on the same date.

(4) The **states_info** table not only record the US states, but also include other states in different countrys. **Some countries might share the same state name.** Each country could have at least 0 states (provinces may be used instead of states), and at most have many states. Each state has a vaccination record for a period of time.

(5) The age groups were divided into several groups, and each age group in each country had a record of vaccination over a period of time.

Explanation of normalization challenges and the resulting changes.

Mapping ER models to relational database schema:

Step1: Mapping strong entity

Country_Info (iso_code, country_name)

Source (source_name, source_url)

Vaccine_Brand (vaccine_brand)

Manufacturer_Location (location)

Date (year, month, day)

Age_group (age_range)

Step2: Mapping weak entity and its identified relationship.

States_Info (iso_code*, states_name)

Country_Vaccines_Brand (iso_code*, year*, month*, day*, vaccines_brand*)

Vaccinations_by_Manufacturer (vaccine_brand*, location*, year*, month*, day*, total_vaccination_number)

Age_Vaccinated_Record (age_range* ,iso_code*, year*, month*, day*, people_vaccinated_per_hundred, people_fully_vaccinated_per_hundred, people_with_booster_per_hundred)

Step3: Mapping 1:1 relationships

Not existing.

Stelp4: Mapping 1:N relationships.

Country_Sources (iso_code, source_name*, source_url*)

Step5: Mapping M:N relationships.

I introduced 3 new table to represent these relationships. Primary keys of participating entities together become the new primary key of this relation.

Country_Vaccinated _Record (iso_code*, year*, month*, day*, people_vaccinated, people_fully_vaccinated, total_boosters)

States_Vaccinated_Record (iso_code*, state_name, year*, month*, day*, people_vaccinated, people_fully_vaccinated, total_boosters, total_distribute)

Country_States (iso_code*, state_name).

Country_States is the same schema as **States_info**. I just keep States_info.

Step6: Multi-valued Attributes [1..N]

Not existing.

Step7: Mapping higher degree relationships

We have 3 ternary relationship in this model (Vaccinations_by_Manufacturer, Age_Vaccinated_Record). The schemas are the same as step2.

Country_Vaccines_Brand (iso_code*, vaccines_brand*, year*, month*, day*)

Vaccinations_by_Manufacturer (vaccine_brand*, location*, year*, month*, day*, total_vaccination_number)

Age_Vaccinated_Record (age_range*, iso_code*, year*, month*, day*, people_vaccinated_per_hundred, people_fully_vaccinated_per_hundred, people_with_booster_per_hundred)

Apart from **Source** (source_name, source_url), all of above relations are in 3NF as they have no transive or partical dependencies.

Source is in 2NF, because source_url <- source_name. It's is a tansitive dependency in Source relation. I need to separate Source into two relations:

Source_url (source_url)

Source_name (source_url*, source_name)

Now, they are both in 3NF. Besides, the relation Country_Source should also be changed as: Country_Source (iso_code*, source_url*). It's also in 3NF now.

Database schema.

In summary, the final schema is:

(1) **Source_url** (source_url)

(2) **Source_name** (source_url*, source_name)

(3) **Country_Info** (iso_code, country_name)

(4) **Country_Source** (iso_code, source_url*)

(5) **Vaccine_Brand** (vaccine_brand)

(6) **Manufacturer_Location** (location)

(7) **Date** (year, month, day)

(8) **States_Info** (iso_code*, states_name)

(9) **Age_group** (age_range)

(10) **Country_Vaccines_Brand** (iso_code*, year*, month*, day*, vaccines_brand*)

(11) **Country_Vaccinated_Record** (iso_code*, year*, month*, day*, people_vaccinated, people_fully_vaccinated, total_boosters)

(12) **Vaccinations_by_Manufacturer** (vaccine_brand*, location*, year*, month*, day*, total_vaccination_number)

(13) **Age_Vaccinated_Record** (age_range* ,iso_code*, year*, month*, day*, people_vaccinated_per_hundred, people_fully_vaccinated_per_hundred,

people_with_booster_per_hundred)

(14) **States_Vaccinated_Record** (iso_code*, state_name, year*, month*, day*, people_vaccinated, people_fully_vaccinated, total_boosters, total_distribute,)

The above relations are all in 3NF as they have no transitive or partial dependencies.