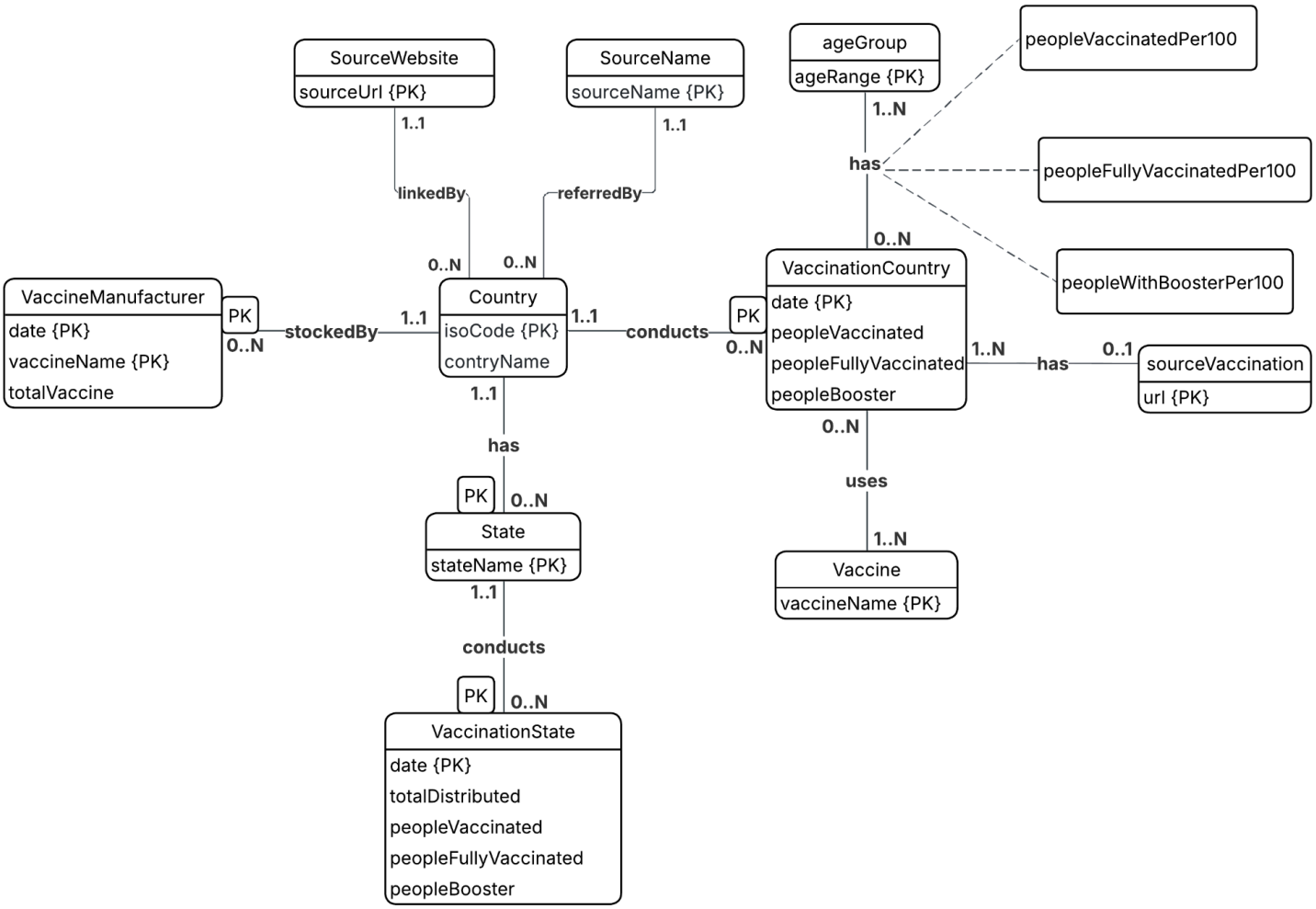
Part B: Designing the Database

Design ER diagram



Assumptions

In the dataset, each country has an isoCode that uniquely identifies it. This makes Country suitable to be modeled as a strong entity, with isoCode serving as its primary key.Furthermore, sourceName and sourceWebsite are not functionally dependent on one another, they can vary and repeat independently. To avoid redundancy and update anomalies, they should be modeled as separate entities (e.g., SourceName and SourceWebsite), each related to Country (1:N).
Because some countries have no state data, and the VaccinationCountry records are not related to states, the State should be modeled as a separate entity. Since different countries may share the same state names, each state requires the country's identifier to be uniquely distinguished. Therefore, model State as a weak entity owned by Country and a 1:N relationship from Country to State. For state data, vaccination records are reported separately per state. Accordingly, define a distinct VaccinationState entity. Because each record must be identified in the context of a specific state, model VaccinationState as a weak entity dependent on State.
Because VaccinationCountry includes age-disaggregated data, define a separate AgeGroup reference entity. Link VaccinationCountry to AgeGroup via an associative entity (e.g., VaccinationAgeGroup), identified by a composite key such as (isoCode, date, ageRange). This associative relation stores the age-specific metrics shown in the ER diagram. Each vaccination record in the dataset includes a sourceUrl, and these URLs are often repeated across records or may not exist in some record. To reduce redundancy and simplify maintenance, model sourceUrl as a separate SourceURL entity and reference it from VaccinationCountry. Use a 1:N relationship from SourceURL to vaccination records, with url as the primary key in SourceURL. This normalization prevents update anomalies and makes corrections to shared URLs straightforward. The dataset lists the vaccine names administered for each vaccination record but does not provide per-vaccine used quantities. Consequently, the model should capture only the association between a vaccination and the vaccines used without any allocation of totals to individual vaccines. Each vaccine can appear in multiple vaccination records. Therefore, define a separate Vaccine entity and model its relationship with VaccinationCountry as M:N.
Based on the provided dataset, vaccine-manufacturer information is not consistently linked to country-level vaccination records. For example, some countries that have vaccination data (e.g., Australia) lack any corresponding manufacturer entries. In addition, several countries do not include state/province data. As a result, Vaccination cannot be reliably related to VaccineManufacturer at the country (or state) level. Given these gaps, modeling this area as a fan-trap relationship does not pose a problem for this dataset.
Moreover, each vaccine-manufacturer record is associated with a specific country and depends on the country's information to be uniquely identified. Therefore, it is appropriate to model VaccineManufacturer as a weak entity dependent on Country.
Finally, some attributes in the dataset are derivable from others and can therefore be excluded to reduce redundancy. Examples include total_vaccination, vaccinated_per_hundred, and vaccinated_per_million. These can be computed on demand from base measures. For vaccination-related data, also remove data from a group of countries out of the database for keeping only data for individual countries.

Mapping ER diagram

1. Map Strong Entities
    - Country (isoCode , countryName)
    - Vaccine (vaccineName)
    - AgeGroup (ageRange)
    - SourceCountry (sourceName)
    - SourceWebsite (SourceUrl)
    - SourceVaccination (url)

2. Map Weak Entities
    - VaccinationCountry (isoCode*, date, peopleVaccinated, peopleFullyVaccinated, peopleBooster)
    - VaccineManufacturer (isoCode*, date , vaccineName , totalVaccine)
    - State (isoCode* , stateName)
    - VaccinationState (isoCode* , stateName* , date , totalDistributed , peopleVaccinated , peopleFullyVaccinated , peopleBooster)

Adding the primary key of the owner entity into the entity along with the partial key attributes form the primary key for the relation.

3. Map 1:1 Relationships
   No action needed (there is no 1:1 relationship)

4. Map 1:N Relationships
    - Country (isoCode , countryName , sourceName* , sourceUrl*)
    - VaccinationCountry (isoCode*, date, peopleVaccinated, peopleFullyVaccinated, peopleBooster , url*)
    - VaccineManufacturer (isoCode*, date , vaccineName , totalVaccine)
    - State (isoCode* , stateName)
    - VaccinationState (isoCode* , stateName* , date , totalDistributed , peopleVaccinated , peopleFullyVaccinated , peopleBooster)

There is no change in the relations (State , VaccinationManufacturer , VaccinationState) because the foreign key was already added in the second step.

5. Map M:N Relationships
    - VaccineAgeGroup (isoCode*, date*, ageRange*, peopleVaccinatedPer100, peopleFullyVaccinatedPer100, peopleWithBoosterPer100)
    - UsedVaccine (isoCode* , date* , vaccineName*)

Creating a new relation and copy the primary key of each of the participating entities to the new relation to form the primary key of the new relation and include all relation attributes

6. Multi-valued Attributes
   No action needed (there is no multi-valued attribute)

7. Map higher-degree relationships
   No action needed (there is no higher-degree relationships)

Final schema

- Country (isoCode , countryName , sourceName* , sourceUrl*)
- SourceCountry (sourceName)
- SourceWebsite (SourceUrl)
- Vaccine (vaccineName)
- VaccinationCountry (isoCode*, date, peopleVaccinated, peopleFullyVaccinated, peopleBooster , url*)
- SourceVaccination (url)
- UsedVaccine (isoCode* , date* , vaccineName*)
- AgeGroup (ageRange)
- VaccineAgeGroup (isoCode*, date*, ageRange*, peopleVaccinatedPer100, peopleFullyVaccinatedPer100, peopleWithBoosterPer100)
- VaccineManufacturer (isoCode*, date , vaccineName , totalVaccine)
- State (isoCode* , stateName)
- VaccinationState (isoCode* , stateName* , date , totalDistributed , peopleVaccinated , peopleFullyVaccinated , peopleBooster)

Normalization

Functional Dependencies

1. isoCode → countryName, sourceName, sourceUrl
2. isoCode, date → peopleVaccinated, peopleFullyVaccinated, peopleBooster, url
3. isoCode, date, ageRange → peopleVaccinatedPer100, peopleFullyVaccinatedPer100, peopleWithBoosterPer100
4. isoCode, date, vaccineName → totalVaccine
5. isoCode, stateName, date → totalDistributed, peopleVaccinated, peopleFullyVaccinated, peopleBooster

Based on the final schema and the functional dependencies analyzed and provided above, each attribute in every relation is considered a single-valued attribute. Therefore, the schema satisfies 1NF.

Moreover, since every composite primary key in each relation has full functional dependency, it also satisfies 2NF.

In addition, there is no transitive dependency in any relation, meaning the schema also satisfies 3NF. Therefore, this schema is already in Third Normal Form (3NF) and does not require further normalization.