

Report

190022658

March 4, 2022

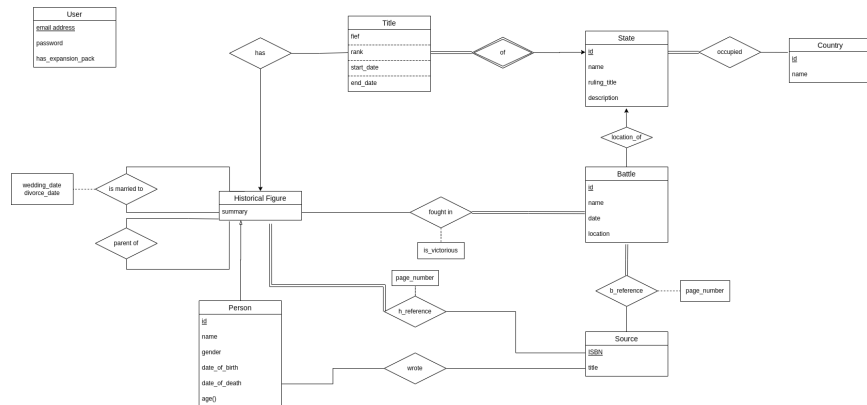
1 Overview

In the practical we were given a scenario and were required to design an ER model for it. We then needed to convert the model into a relational schema and identify all the full functional dependencies in the relational schema. Using that we needed to show which normalisation form was achieved.

All of the listed requirements have been done.

2 ER Diagram

The ER diagram is in the same folder in two formats, it can be opened in the browser in the svg format and in diagrams.net in the drawio format.



3 Relational schema

- person (id: integer, name: string, gender: string, date_of_birth: date, date_of_death: date, age: integer)
- historical_figure (id*: integer, summary: string)
- state (id: integer, name: string, description: string, ruling_title: string)
- country (id: integer, name: string)
- battle (id: integer, name: string, date: date, location: (float, float), state_id*: integer)
- source (ISBN: string, title: string)
- user (email_address: string, password: string, has_expansion_pack: boolean)
- title (state_id*: integer, fief: string, rank: string, start_date: date, end_date: date, historical_figure_id*: integer)
- is_married_to (person_id*: integer, person2_id*: integer, wedding_date: date, divorce_date: date)
- parent_of (parent_id*: integer, child_id*: integer)
- occupied (state_id*: integer, country_id*: integer)
- b_reference (battle_id*: integer, ISBN*: string, page_number: integer)
- h_reference (historical_figure_id*: integer, ISBN*: string, page_number: integer)
- fought_in (historical_figure_id*: integer, battle_id*: integer, is_victorious: boolean)
- wrote (person_id*: integer, ISBN*: string)

4 Design

4.1 ER Model

For the ER model I first identified nouns and then the attributes as follows.

- Person
 - name, gender, date_of_birth, date_of_death, age()
- Historical figure inherited from person
 - summary
- Title (weak set)
 - fief, rank, duration
- State
 - name, description, ruling title
- Modern country
 - name
- Battle
 - name, date, location
- Source
 - title, ISBN
- User
 - email, password, has_expansion_pack

Then I figured the relationships they should have between them as follows.

- Person and Person (person is parent of person)
- Person and Person (person is married to person)
- Person and Title (person has titles)
- Title and State (title are of a state)

- State and Modern Country (state occupied countries)
- Battle and Person and State (battle between persons in state)

A historical figure and a battle can only have a book as a reference for a single page. This constraint is added because of the specification. This can be changed by making the page number attribute of the relationship multi-valued.

A battle is fought in a single state or no state.

I used a superclass “person” for historical figure as it allows non-historic figures to be authors of sources as well.

Title is a weak entity set because there is candidate key and so that no information is lost.

The user entity can have bookmarks and other features depending on the scenario but this is enough for the base scenario.

Relationship attributes was necessary because of many-to-many relationship and made the diagram consise and easy to understand.

The only derived attribute is the age of a person, and while it can be a simple attribute it makes more sense this way as the information needed to compute it is already available.

The battle name could be the primary key but that would require the assumption that each battle name is unique which might or might not be true.

4.2 Relational Schema

Here I simply transferred the attributes to the schema. In cases of one-to-many or many-to-one relations, I added a foreign key of the “many” entity set to the “one” entity set. In the case of many-to-many relations, I added relational schemas with private keys of both entity sets along with relation attributes where applicable.

I chose a tuple of two floats for the location to be specific.

4.3 Normalisation

The full functional dependencies are as follows

4.3.1 person

id -> name
id -> gender
id -> date_of_birth
id -> date_of_death
id -> age
{ date_of_birth, date_of_death } -> age

4.3.2 historical_figure

id -> summary

4.3.3 state

id -> name
id -> description
id -> ruling_title

4.3.4 country

id -> name

4.3.5 battle

id -> name
id -> date
id -> location
id -> state_id

4.3.6 source

ISBN -> title

4.3.7 user

email_address -> password
email_address -> has_expansion_pack

4.3.8 title

{ state_id, fief, rank, start_date, end_date } -> historical_figure_id

4.3.9 Conclusion

It is 1NF because there is no searching using a name given in the scenario and all relevant relational access is atomic.

It is 2NF because all non-prime attributes are fully functionally dependent on the candidate keys.

Not a 3NF or BCNF because a derived attribute exists. A derived attribute is a transitive dependency. In this case in the relation, person, there exists

id -> date_of_birth
id -> date_of_death
{ date_of_birth, date_of_death } -> age
id -> age

Thinking about it, the id in battle might be dependent on the attributes - name, date, location.

5 Conclusion

This practical reinforced the theory learnt and helped practice building database models.