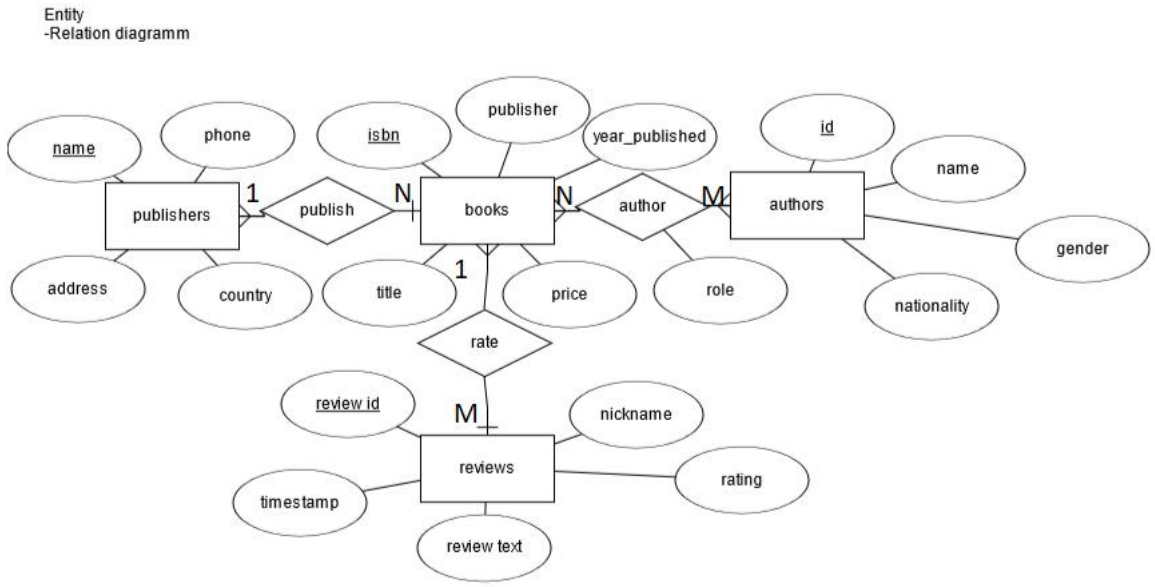
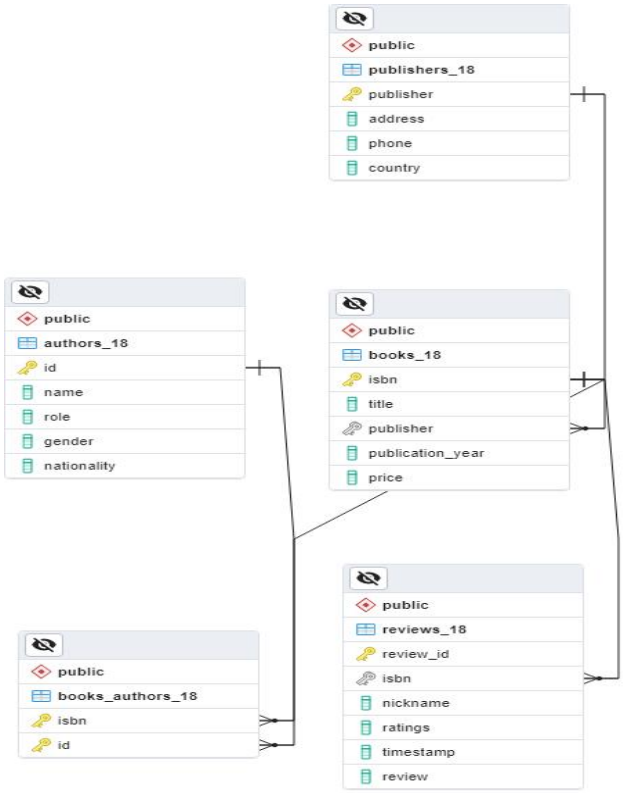
MSc in "Data Science"
Data Management
Student Name: Eleni Neti
Student ID: 2022202204018

We build our database based on the following E-R diagramm:



The above is easily turned into the ralational database:



In brief, 5 relations arise in our relational database.

'PUBLISHERS_18' The table has four columns: publisher, address, phone, and country. The publisher column is a VARCHAR(255) field that is also the primary key for the table, which means that it must contain a unique value for each row in the table and cannot be NULL. The address, phone, and country columns are also VARCHAR(255) fields, but they have default values of 'UNKNOWN' if no value is specified for a particular row.

'BOOKS_18' The columns are as follows:

- isbn: This column stores the ISBN of the book, which is a unique identifier for the book. The data type of this column is VARCHAR(10), which means it can store up to 10 characters of text. We chose this data type, in order to avoid troubleshooting, since some of the isbns contain a string character. This column is also marked as the PRIMARY KEY, which means that each value in this column must be unique and cannot be NULL.
- title: This column stores the title of the book. The data type of this column is VARCHAR(200), which means it can store up to 200 characters of text. This column is marked as NOT NULL, which means that each value in this column must be provided and cannot be NULL.
- publisher: This column stores the name of the publisher of the book. The data type of this column is VARCHAR(255), which means it can store up to 255 characters of text. This column is marked as NOT NULL, which means that each value in this column must be provided and cannot be NULL.
- publication_year: This column stores the year in which the book was published. The data type of this column is INT, which means it can store integer values. This column is marked as NOT NULL, which means that each value in this column must be provided and cannot be NULL.
- price: This column stores the price of the book. The data type of this column is NUMERIC(10,2), which means it can store numeric values with up to 10 digits, including 2 digits after the decimal point. The default value of this column is 0.00, which means that if no value is specified when inserting a new record, the price will be set to 0.00.

The table also has a foreign key constraint on the publisher column. This constraint specifies that the publisher values in the BOOKS_18 table must match the publisher values in the PUBLISHERS_18 table. This constraint also specifies that if the publisher value in the PUBLISHERS_18 table is updated, the corresponding value in the BOOKS_18 table will be updated automatically (ON UPDATE CASCADE), and if a publisher value is deleted from the PUBLISHERS_18 table, all corresponding values in the BOOKS_18 table will be deleted automatically (ON DELETE CASCADE).

'AUTHORS_18' The columns are as follows:

- id: This column stores a unique identifier for the author. The data type of this column is VARCHAR(20), which means it can store up to 20 characters of text. This column is also marked as the PRIMARY KEY.
- name: This column stores the name of the author. The data type of this column is VARCHAR(255)and is marked as NOT NULL.
- role: This column stores the role of the author, such as "writer" or "illustrator". The data type of this column is VARCHAR(255)and marked as NOT NULL.
- gender: This column stores the gender of the author. The data type of this column is VARCHAR(255). The default value of this column is 'UNKNOWN', which means that if no value is specified when inserting a new record, the gender will be set to 'UNKNOWN'. This column is also marked as NOT NULL.
- nationality: This column stores the nationality of the author. The data type of this column is VARCHAR(255), The default value of this column is also set to 'UNKNOWN'.

'BOOKS_AUTHORS_18' This relation will be used to establish a many-to-many relationship between books and authors by storing the ISBN and author ID for each book written by each author

The columns are as follows:

- isbn: This column stores the ISBN of a book. The data type of this column is VARCHAR(10), which means it can store up to 10 characters of text. This column has a foreign key constraint that specifies that the values in this column must match the values in the isbn column of the BOOKS_18 table.
- id: This column stores the unique identifier of an author. The data type of this column is VARCHAR(20), which means it can store up to 20 characters of text. This column has a foreign key constraint that specifies that the values in this column must match the values in the id column of the AUTHORS_18 table.

The table also has a primary key constraint on the isbn and id columns. This constraint specifies that each combination of values in these columns must be unique, which means that each book-author combination can only be represented once in the table. This is useful because it ensures that there are no duplicate records in the table and allows you to easily reference specific book-author combinations.

'REVIEWS_18' The table has several columns, each with a specific data type:

- review_id: a string with a maximum length of 255 characters, and is set as the primary key for the table.
- isbn: a string with a maximum length of 10 characters, and is used as a foreign key to reference the isbn column in the "BOOKS_18" table.
- nickname: a string with a maximum length of 255 characters, and has a default value of "USER" if no value is specified when inserting a new row.
- ratings: an integer for storing a user's ratings for a book.
- timestamp: a timestamp for storing the time at which a review is added to the table. If not avalible, the default value is the current time.This can be useful for tracking the time when a record was created or modified.
- review: a string for storing the text of a user's review. The default value is "not available" if no review is provided.

The table also has two "ON UPDATE" and "ON DELETE" cascade constraints for the isbn foreign key, which specifies how the database should handle updates or deletions in the "BOOKS_18" table that affect related rows in the "REVIEWS_18" table.