



This model describes a **public library**.

Table Patron contains library users. A patron may be

- Adult - then the e-mail address and phone no. must be filled in; parent_id is null
- Child – e-mail address and phone no. are null, parent_id must be filled in and must refer to an adult patron (it is **not required** to enforce this business rule as a data constraint in DB)

The following attributes can hold null values: Patron.e_mail, Patron.phone_no, Patron.parent_id, Checkout.end_date.

Implement the data model using ORM

- Use Python SQLAlchemy ORM with **type annotations**
- Use file-based SQLite DB
- Create Table object representing the join table book_author
- Define mapped class for each DB table (except join table)
 - Define attribute for each DB column using "Mapped" type annotation and, if needed, also "mapped_column()" function
 - Define attributes for both directions of each relationship in the model (including many-to-many relationship represented by join table) using "relationship" function
- Create the model within your DB

Insert data using ORM

- Adult patrons (full name, e-mail, phone no.)
 - Spongebob Squarepants, spongebob@sqlalchemy.org, 111111
 - Sandy Cheeks, sandy_cheeks@sqlalchemy.org, 222222
- Child patrons (full name, parent's name)
 - Eddie Cheeks, Sandy Cheeks
 - Patricia Cheeks, Sandy Cheeks

- Books & authors (authors, title, year)
 - Neil Gaiman, Terry Pratchett: Good Omens, 1990
 - Stephen Baxter, Terry Pratchett: The Long Earth, 2012
 - Terry Pratchett: The Colour of Magic, 2022
 - In all three cases “Terry Pratchett” is the same author and must be represented by a single object/row
 - Insert one extra book of your choice and its author(s)
- Checkouts (patron full name, book title, start date, expiry date, end_date)
 - Eddie Cheeks, The Colour of Magic, 2023-10-20, 2023-11-20,
 - Eddie Cheeks, Good Omens, 2023-10-08, 2023-11-08,
 - Patricia Cheeks, The Long Earth, 2023-09-02, 2023-12-02,
 - Spongebob Squarepants, The Long Earth, 2023-05-20, 2023-08-20, 2023-08-15
 - Insert additional checkout for your extra books. Assign it to any of the existing patrons.

Create queries using ORM

- Define function **patron.display_checkouts(self)** that writes to stdout list of current checkouts and their children's current checkouts (current = end_date is NULL)
 - For each checkout display autor(s) name, title and expiry_date; if expiry_date < current_date, display info "EXPIRED"
 - If patron/child has no checkouts, display info "No checkouts"
 - In the main program, call the function for the patron with full_name “Sandy Cheeks”. It should give result:

```
Patron: Sandy Cheeks
No checkouts
```

```
Child: Eddie Cheeks
Terry Pratchett: The Colour of Magic, until: 2023-11-20
Neil Gaiman, Terry Pratchett: Good Omens, until 2023-11-08 - EXPIRED
```

```
Child: Patricia Cheeks
Stephen Baxter, Terry Pratchett: The Long Earth, until 2023-12-02
```

- Define function **book.display_checkout_history(self)** that writes to stdout list of all checkouts for given book together with patron full names and start date + end date. If end date is NULL, info “Active checkout” is displayed.
 - In the main program, call the function for the book with the title “The Long Earth”. It should give result:

```
Stephen Baxter, Terry Pratchett: The Long Earth
```

```
Checkout history:
Patricia Cheeks, from: 2023-09-02 - Active checkout
Spongebob Squarepants, from: 2023-05-20, to: 2023-08-15
```

Modify data using ORM

- Define function **patron.prolong_all(self, new_ex_date)** that sets expiry_date of all current patron's checkouts (current = end_date is null) to new_ex_date
- In the main program call this function for Eddie Cheeks and new_ex_date = 2024-01-20. Persist changes in DB.