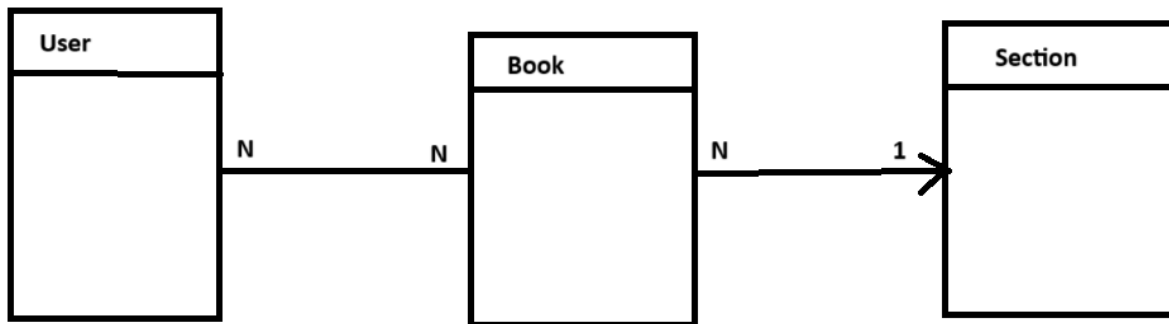


Models for Library Management System

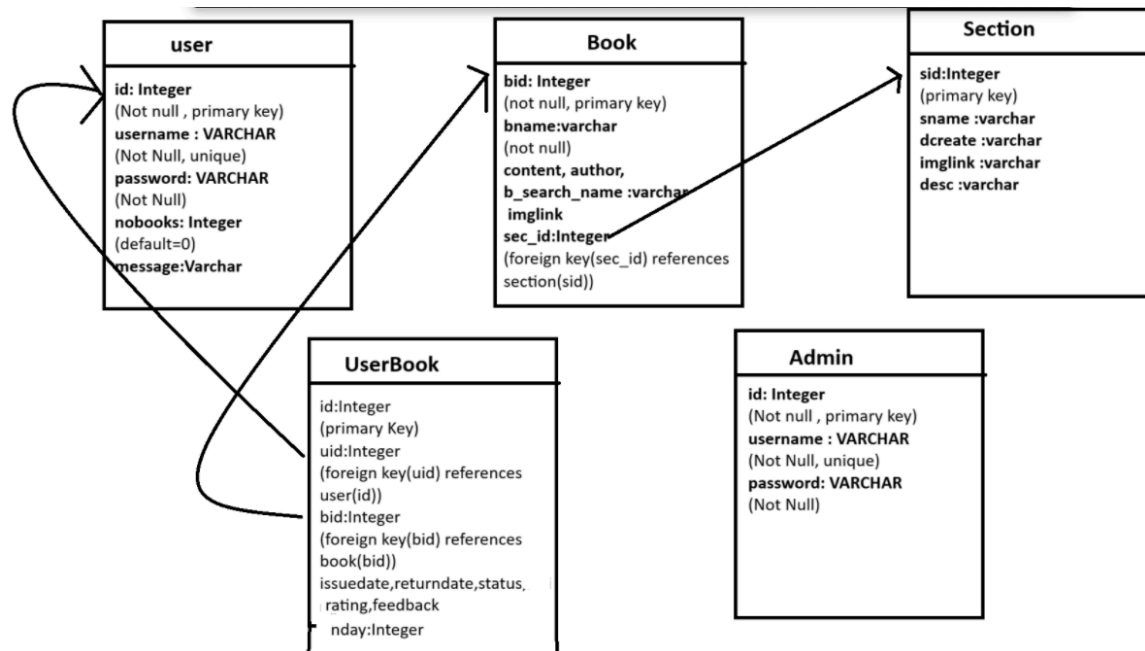
Presentation video link:- <https://youtu.be/B433bATgT8g>

- Basic Idea



There is a many to many relationship between user and book
There is many to one relationship between books and section

- Actual sqlite implementation



Here the UserBook is the associative table between book and user

The arrows representing the attribute having foreign keys and taking values from other tables

- FlaskSQLAlchemy Models

```
class User(db.Model):
    id=db.Column(db.Integer(),primary_key=True)
    username=db.Column(db.String(),nullable=False,unique=True)
    password=db.Column(db.String(),nullable=False)
    nobooks=db.Column(db.Integer(),default=0)
    message=db.Column(db.String(),default="nomessage")
    user_b=db.relationship("UserBook",back_populates="user")
```

```
class Admin(db.Model):
    id=db.Column(db.Integer(),primary_key=True)
    username=db.Column(db.String(),nullable=False,unique=True)
    password=db.Column(db.String(),nullable=False)
```

```
class Section(db.Model):
    sid=db.Column(db.Integer(),primary_key=True)
    sname=db.Column(db.String(),nullable=False,unique=True)
    dcreate=db.Column(db.String(),nullable=False)
    imglink=db.Column(db.String())
    desc=db.Column(db.String())
    sec_b=db.relationship("Book",backref="book_s")
```

```
class Book(db.Model):
    bid=db.Column(db.Integer(),primary_key=True)
    bname=db.Column(db.String(),nullable=False)
    content=db.Column(db.String(),nullable=False)
    author=db.Column(db.String(),nullable=False)
    b_search_name=db.Column(db.String(),nullable=False)
    imglink=db.Column(db.String())
    sec_id=db.Column(db.Integer(),db.ForeignKey("section.sid"))
    book_u=db.relationship("UserBook",back_populates="book")
```

```
class UserBook(db.Model):
    id=db.Column(db.Integer(),primary_key=True)
    uid=db.Column(db.Integer(), db.ForeignKey('user.id'))
    bid=db.Column(db.Integer(), db.ForeignKey('book.bid'))
    issuedate=db.Column(db.String(), nullable=False)
    returndate=db.Column(db.String(), default="to be updated")
    nday=db.Column(db.Integer(),nullable=False)
    status=db.Column(db.String(),nullable=False,default="Requested")
    rating=db.Column(db.String(),default="unknown")
    feedback=db.Column(db.String(),default="unknown")
    user=db.relationship("User",back_populates="user_b")
    book=db.relationship("Book",back_populates="book_u")
```