

Gestionarea unei biblioteci

ADRIAN MANEA, 510

11 ianuarie 2020

Cuprins

DE ADĂUGAT/CLARIFICAT	1
1 Descrierea aplicației	3
1.1 Utilizatori și atribute	3
1.2 Procesele din aplicație	3
1.3 Matricea proces-utilizator	6
1.4 Entitățile și modelarea datelor	6
1.5 Schemele relaționale	8
1.6 Matricea entitate-proces	8
1.7 Matricea entitate-utilizator	9
1.8 Utilizatori și conturi	9
2 Implementare în PostgreSQL	10
2.1 Instalare și configurare inițială	10

DE ADĂUGAT/CLARIFICAT

cum țin număr de împrumuturi & termen? 8

SECȚIUNEA 1 _____

DESCRIEREA APLICAȚIEI

Aplicația servește la gestionarea unei biblioteci, în relația cu cititorii, oferind roluri speciale pentru clienți din mediul academic.

1.1 Utilizatori și atribute

Utilizatorii aplicației, identificați prin IP-uri specifice sînt:

- **Administratori** ai bazei de date;
- **Personal educațional** (EDU);
- **Cititori** din afara mediului educațional (NEDU);
- **Bibliotecari**;
- **Cititori înregistrați, dar neabonați** (PUBLIC).

Considerăm că studenții și profesorii fac parte din personalul educațional (Edu), statut pe care trebuie să-l verifice periodic. De asemenea, un cititor oarecare poate deveni Edu în timp, dacă adaugă verificarea.

1.2 Procesele din aplicație

(P1) Vizualizarea cărților în stoc;

(P2) Vizualizarea bibliotecarilor, cu specializările lor;

(P3) Adăugarea unei cărți;

- (P4) Adăugarea unui cititor;
- (P5) Adăugarea unui abonat Edu;
- (P6) Înregistrarea unui cont public;
- (P7) Actualizarea stocului unor cărți;
- (P8) Verificarea statutului Edu;
- (P9) Administrarea abonamentului;
- (P10) Vizualizarea cărților de o anumită specializare;
- (P11) Vizualizarea revistelor de o anumită specializare;
- (P12) Adăugare reviste;
- (P13) Administrare abonament revistă;
- (P14) Administrare abonament la newsletter;
- (P15) Împrumut carte;
- (P16) Feedback bibliotecar.

De exemplu, **descompunerea funcțională** a procesului (P14), de administrare a abonamentului la o revistă, poate fi reprezentat ca în figura 1.1.

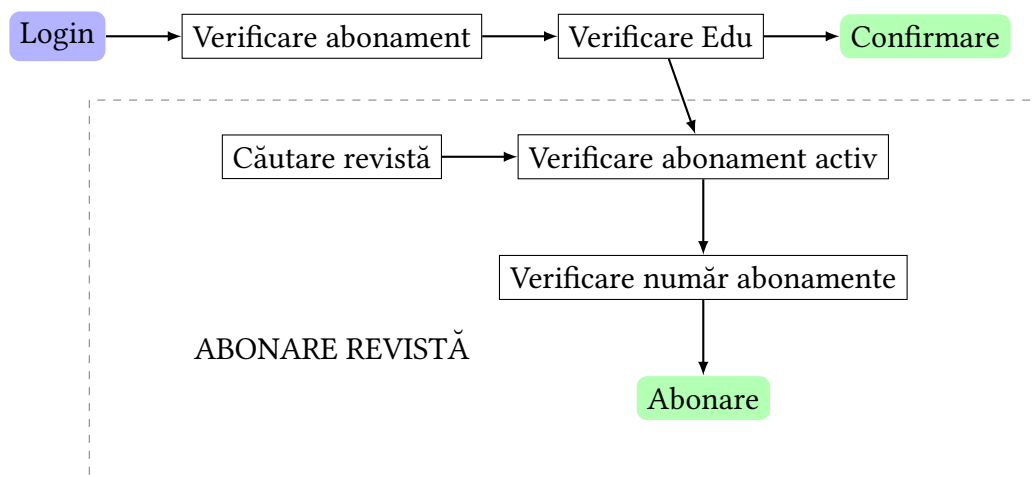


Figura 1.1: Descompunerea funcțională a procesului de abonare la revistă

Similar, procesul (P9) de verificare a statutului Edu se poate reprezenta ca în figura 1.2.

Și un ultim exemplu pe care îl prezentăm este acela al procesului (P16), de scriere a feedback-ului pentru un bibliotecar. Descompunerea funcțională este prezentată în figura 1.3.

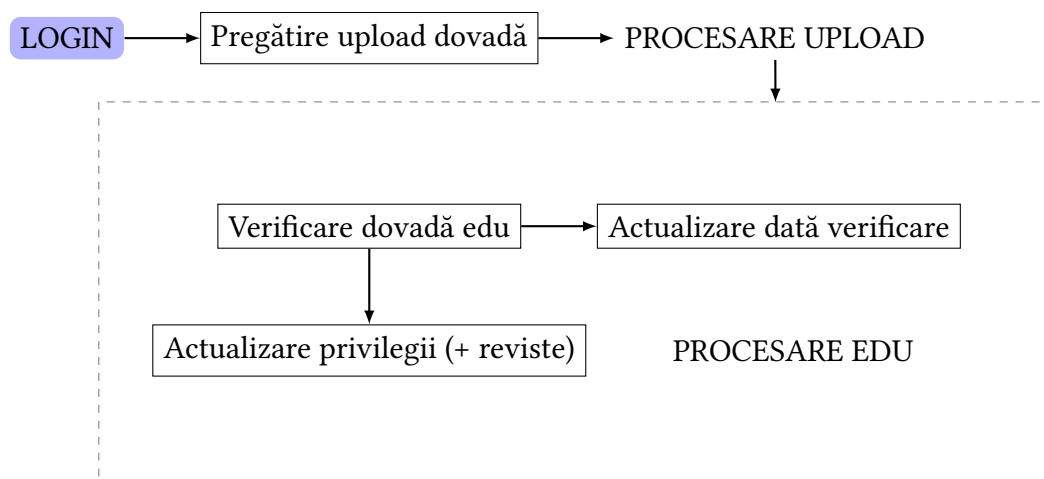


Figura 1.2: Descompunerea funcțională a procesului de verificare Edu

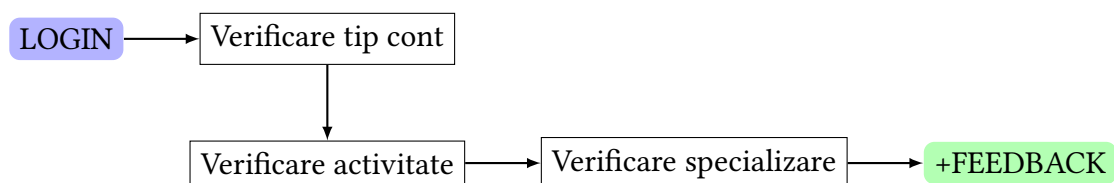


Figura 1.3: Descompunerea funcțională a procesului de adăugare feedback

1.3 Matricea proces-utilizator

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	P16
Administrator	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Edu	X	X						X	X	X	X			X	X	X
NEdu	X	X						X	X	X				X	X	X
Bibliotecar	X	X	X	X	X				X	X	X	X	X	X		
Public	X									X				X		

1.4 Entitățile și modelarea datelor

Biblioteca este un centru de împrumut care permite și abonamentul la reviste pentru cei din mediul academic.

În bibliotecă se pot adăuga *cărți* și *reviste* de diverse specializări, în funcție de contractele cu furnizorii. Fiecare carte și revistă aparțin unei singure specializări.

Personalul Edu se poate abona și la reviste de specialitate, în baza unei verificări actualizate. Ceilalți cititori cu abonament pot deveni Edu printr-o verificare. Publicul larg poate doar să vadă stocul de cărți, general sau pe specializări și să adauge cărțile dorite în wishlist. Toți utilizatorii se pot abona la newsletter pentru a afla când se actualizează stocul.

De asemenea, bibliotecarii sînt asociați specializărilor, fiind responsabil de publicațiile dintr-o anumită specializare.

Cititorii abonați (EDU sau NEDU) pot lăsa feedback bibliotecarilor cu care au lucrat anterior într-o anumită specializare.

Diagrama ER se poate prezenta ca în figura 1.4.

De menționat că toate relațiile M:M ($\infty : \infty$) vor fi rezolvate cu *tabelele asociative* BIB_SPEC, BIB_CIT și PUB_CARTE, listate la §1.5.

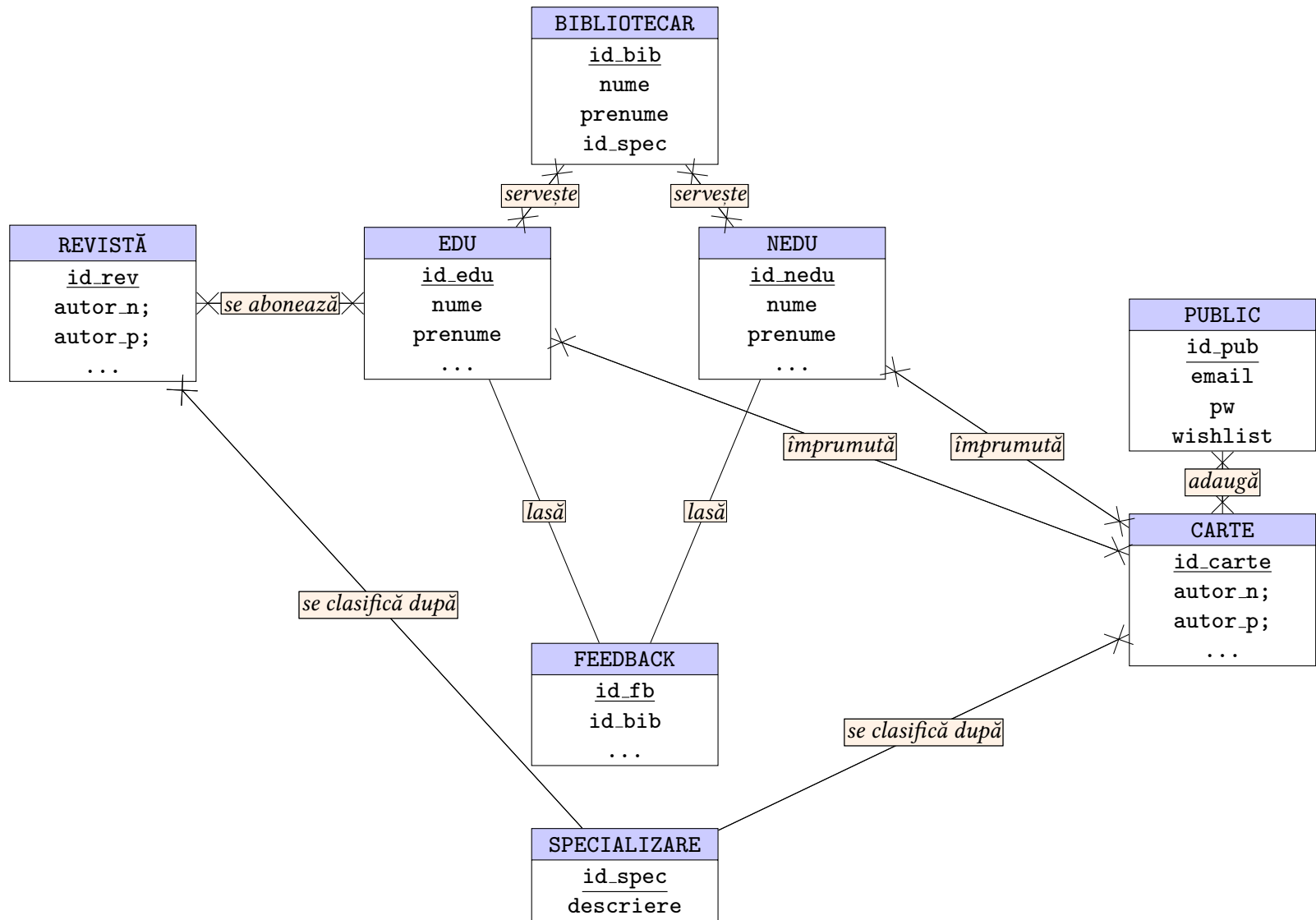


Figura 1.4: Diagrama ER

1.5 Schemele relaționale

```

BIBLIOTECAR      (id_bib#, nume, prenume, id_specializare);
NEDU              (id_cit#, nume, prenume, email, newsletter, abonament_activ,
                  id_carte, id_spec);
EDU               (id_edu#, nume, prenume, email, newsletter, abonament_activ,
                  id_carte, id_rev, id_spec);
PUBLIC            (id_pub#, email, pw, wishlist, newsletter);
CARTE             (id_carte#, autor_n, autor_p, titlu, an, id_spec, stoc);
REVISTA           (id_rev#, autor_n, autor_p, titlu, numar, id_spec, stoc);
SPECIALIZARE      (id_spec#, descriere);
FEEDBACK          (id_fb#, id_bib, id_cit, id_edu, id_spec, rating, text, data);
BIB_SPEC          (id_bib#, id_spec#);
BIB_CIT           (id_bib#, id_cit#);
PUB_CARTE         (id_pub#, id_carte#);

```

cum țin număr de împrumuturi & termen?

1.6 Matricea entitate-proces

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	P16
BIBLIOTECAR	S	S	IUD	IUD	IUD		U	IUD	U	S	S	IUD	IUD	U	U	
NEDU	S	S							IUD	IUD	S				U	
EDU	S	S							U	IUD	S	S		U	U	
PUBLIC	S					IUD						S		IUD		
CARTE	S		IUD								S				U	
REVISTA												S	I			
SPECIALIZARE											S	S				
FEEDBACK																I

Legenda: I = Insert, U = update, D = delete, S = select.

1.7 Matricea entitate-utilizator

	Edu	NEdu	Bibliotecar	Admin	Public
BIBLIOTECAR	S	S	S	S, I, U, D	
NEDU		S, I, U, D	S, I, U, D		
EDU		S, I, U, D	S, I, U, D		
PUBLIC					
CARTE	S	S	S, I, U, D	S, I, U, D	S
REVISTA	S		S, I, U, D	S, I, U, D	S
SPECIALIZARE	S	S	S	S, I, U, D	S
FEEDBACK	I	I	S	S,I,U,D	

Legenda: I = Insert, U = update, D = delete, S = select.

1.8 Utilizatori și conturi

Conturile utilizatorilor vor fi definite individual pentru utilizatorii identificați de bibliotecă.

În această etapă a dezvoltării aplicației, vor fi disponibile maximum 10000 de conturi Edu, 10000 de conturi de cititori (înregistrați), 10 conturi de bibliotecar, 1000 conturi de utilizator neabonat (PUBLIC).

SECȚIUNEA 2 _____

IMPLEMENTARE ÎN POSTGRESQL

2.1 Instalare și configurare inițială

Pentru implementare, vom folosi sistemul de baze de date PostgreSQL, disponibil gratuit și open source. Implementarea se va face pe laptopul personal, folosind:

- OS: Manjaro Linux i3;
- Emacs 26 pentru editare text și comenzi shell;
- `st` pentru comenzi avansate de terminal, dacă este necesar.

Programul se instalează folosind managerul de pachete din Manjaro, cu comanda:

```
$ sudo pacman -S postgresql
```

După instalare, putem consulta consulta detaliile de pe pagina ArchWiki.

Pe scurt, PostgreSQL creează automat un utilizator cu numele `postgres`, care este proprietarul implicit al bazelor de date. Astfel, pentru a lansa aplicația și a face modificări, trebuie să ne identificăm ca utilizatorul `postgres`, cu una dintre comenzile:

```
$ sudo -iu postgres
$ su -l postgres
```

Inițializarea cluster-ului de baze de date, unde se vor crea bazele și tabelele se face în calitatea de utilizator `postgres`, de exemplu, în locația implicită, cu comanda:

```
[postgres]$ initdb -D /var/lib/postgres/data
```

pentru a forța utilizarea limbii engleze și a codării UTF-8, folosim comanda

```
[postgres]$ initdb --locale=en_US.UTF-8 -E UTF8 -D /var/lib/postgres/data
```

```
# ... output ...
```

```
# finalizat cu ... ok
```

Folosind `systemd`, trebuie să activăm și să pornim daemon-ul `postgresql`, cu comenzile:

```
$ sudo systemctl enable postgresql
```

```
$ sudo systemctl start postgresql
```

Pentru început, creăm un utilizator nou, căruia îi putem da ce atribuții dorim, precum și o bază de date pe care utilizatorul respectiv să o poată accesa (sau administra, în funcție de rolul dat):

```
# devenim utilizatorul postgres mai întâi
```

```
$ sudo -iu postgres
```

```
# creăm utilizatorul (e.g. theUser) cu dialog pas cu pas, pentru a alege rolul
```

```
[postgres]$ createuser --interactive
```

```
# creăm o bază de date pentru el (e.g. theDatabase)
```

```
[postgres]$ createdb -O theUser theDatabase
```

```
# dacă theUser nu are rol de creare, putem crea cu postgres pentru el
```

```
[postgres]$ createdb -U postgres -O theUser theDatabase
```

Acum putem porni subshell-ul `psql` și să ne conectăm la baza de date creată mai sus:

```
$ sudo -iu postgres
```

```
[postgres]$ psql -d theDatabase
```

```
# câteva dintre meta-comenzile pentru subshell-ul psql:
```

=> \help	# accesează help
=> \c <database>	# conectează-te la baza de date <database>
=> \du	# afișează utilizatorii și permisiunile
=> \dt	# afișează tabelele și permisiunile
=> \q	# închide subshell-ul
=> \?	# afișează toate meta-comenzile